C Tech Help System
for
EVS & MVS 9.88

EVS / MVS HELP SYSTEM

EVS and our flagship MVS are standalone Windows programs which offers the ultimate in speed, power and flexibility. C Tech unites advanced volumetric gridding, geostatistical analysis, and 4D visualization tools into a software system developed to address the needs of all Earth science disciplines. The graphical user interface is integrated with modular analysis and graphics routines which can be customized and combined to satisfy the analysis and visualization needs of any application. EVS and MVS can be used to analyze all types of analytical and geophysical data in any environment (e.g. soil, groundwater, surface water, air, noise, resistivity, etc.). Our integrated geostatistics provides quantitative evaluation of the quality of a site assessment and identification of locations at sites that require additional investigation.

For more information visit ctech.com

Hardware & Operating System Requirements

EVS & MVS have very few specific hardware requirements, and has been designed to run on a wide range of PC hardware. The software will run on virtually any windows computer.

The following table gives **Recommended** and **Ideal** system configurations. Of these requirements, CPU speed, number of cores and memorty speed have the greatest impact on calculation time, while graphics resolution and number monitors affects the appearance and usability the greatest. All system hardware must meet Windows 7, Vistas or XP hardware requirements.

Hardware Item	Recommended Configuration	Ideal Configuration
Operating System	Windows Vista 64 bit or higher	Windows 7 - 64 bit
CPU	2 or more cores	4 or more physical cores
RAM	8+ Gb - 1,066 Mhz or higher	12+ Gb - 1,600 Mhz or hig
Hard Disk	275 Mb installation 5+ Gb free	275 Mb installation 50+ Gb free
Graphics Card	ATI or NVIDIA Gaming cards	High end gaming or profes
Graphics Resolution	1920 x 1080 Normal fonts	1920 x 1080 or higher Normal fonts
Monitor	19 inch color	Dual monitors of 24 inch o
e-mail & web access	Required for software downloads and technical support	Required for software dow

Operating System Requirements

EVS & MVS are compatible with Windows XP, Vista, and Windows 7. Windows 7, 64 bit is our preferred operating system and unsupported operating systems (XP and Vista) may be dropped in a future release.

Please download and install all critical updates from:
http://windowsupdate.microsoft.com

Installing C Tech Software

Get ready to ***Visualize a New Approach*** using EVS or MVS. We are confident that you will find EVS is an easy to use, yet extremely powerful software tool. C Tech is continually developing new modules that dramatically enhance the utility of EVS. Customer suggestions and a creative development team have made, and will continue to make EVS the premier geologic, environmental, and mining analysis and visualization software system.

Installation Overview

The most current released version of our software may be downloaded from http://www.ctech.com if you first register and login. After downloading, run the program Install_EVS_MVS.exe and respond to the prompts.:

Detailed Installation Instructions are here.

Copy Protection

EVS & MVS is distributed as a demonstration version, which is converted to a working system by a software (hardware) key that plugs into the parallel port or by connecting to a computer running our Floating License Server Software. The demonstration version is limited to allow reading of only the specially encrypted data files that are supplied with the program. No changes or additions can be made to these data files or they will no longer operate with the program. The hardware key is activated using a key driver program that is installed using the "Install Key Driver" icon in the Tools sub-folder in the C Tech program group. The software protection is transparent to the user when properly installed. When the software starts, a splash screen will declare the mode that the software is running under (Demo, Full Version or Floating Version).

Installation

EVS is installed using the following procedure:

Installation of EVS for Windows Vistas and XP involves the following steps:

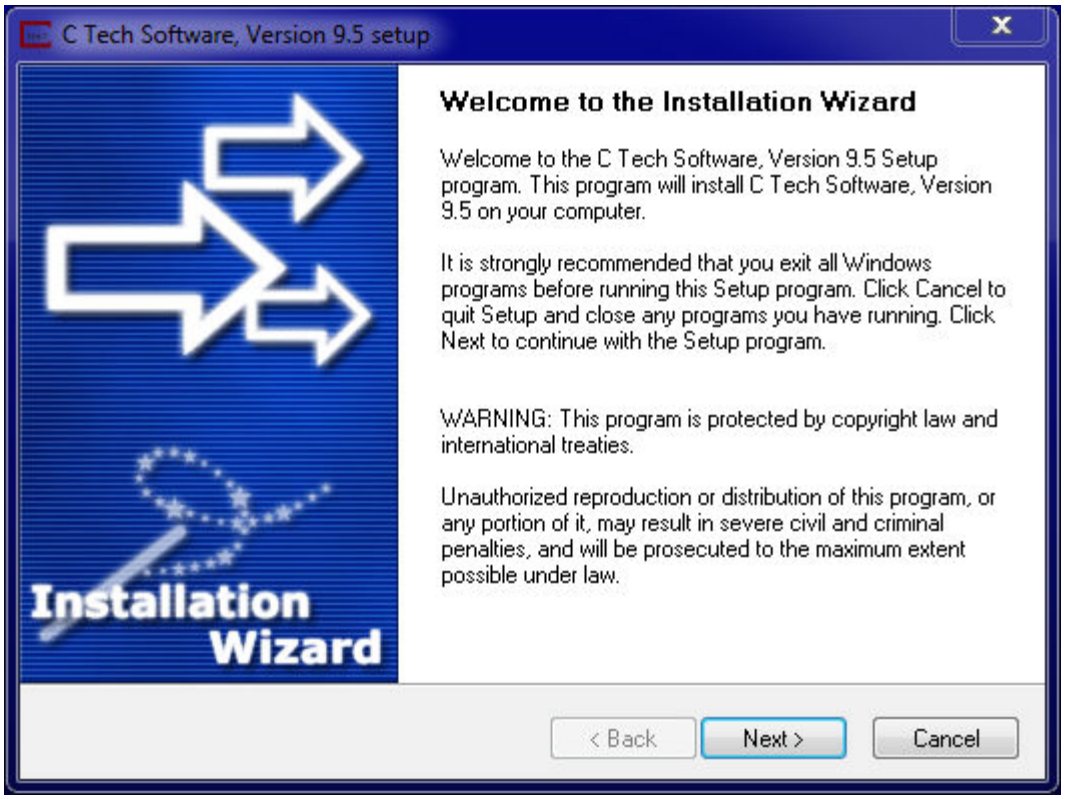
i. Insert the EVS DVD.

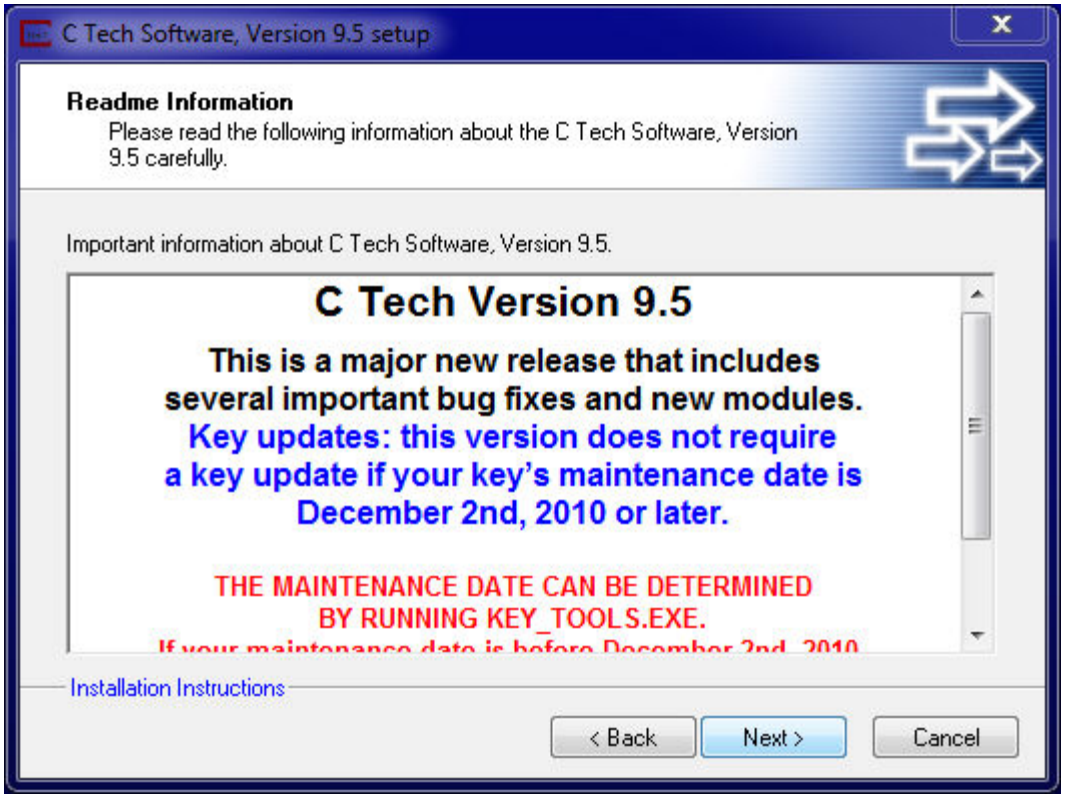
ii. If it does not autostart. Execute setup.exe on the EVS DVD(by clicking on Start > Run > drive:setup.exe > OK) and follow instructions.

iii. The install program will prompt you to specify the destination drive and directory where you wish to have EVS installed. The default is C:\CTECH.

Everything else is automatic.

Detailed Installation Instructions

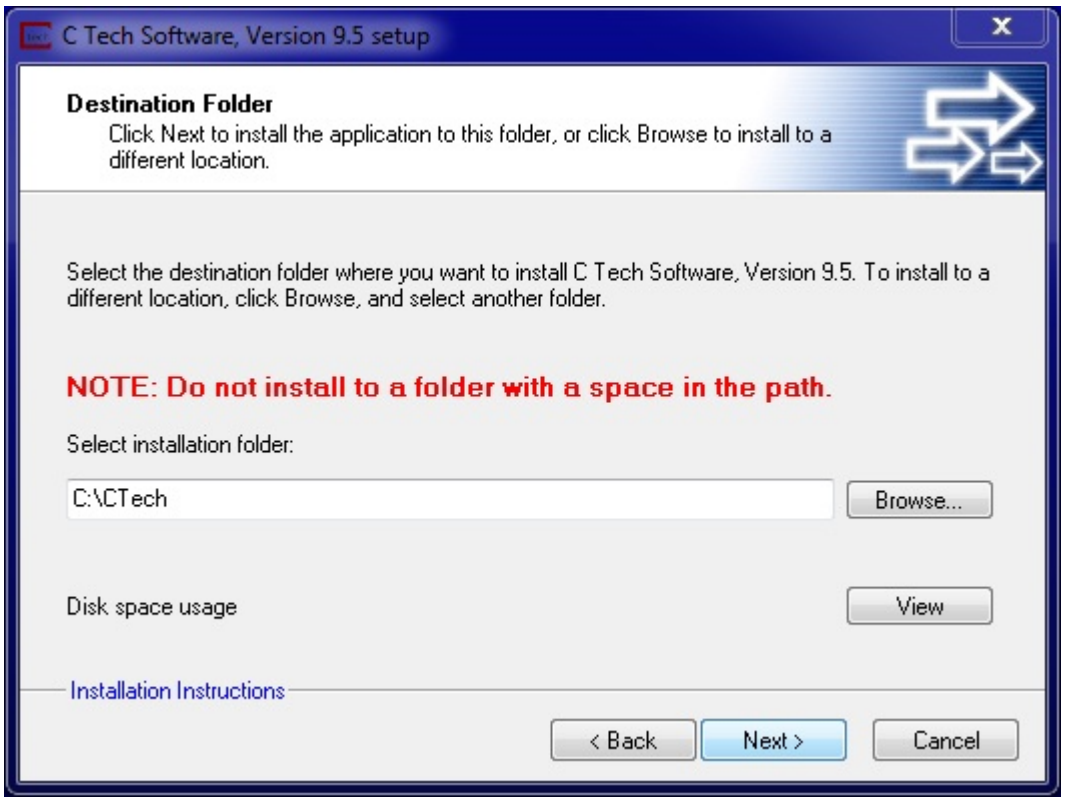




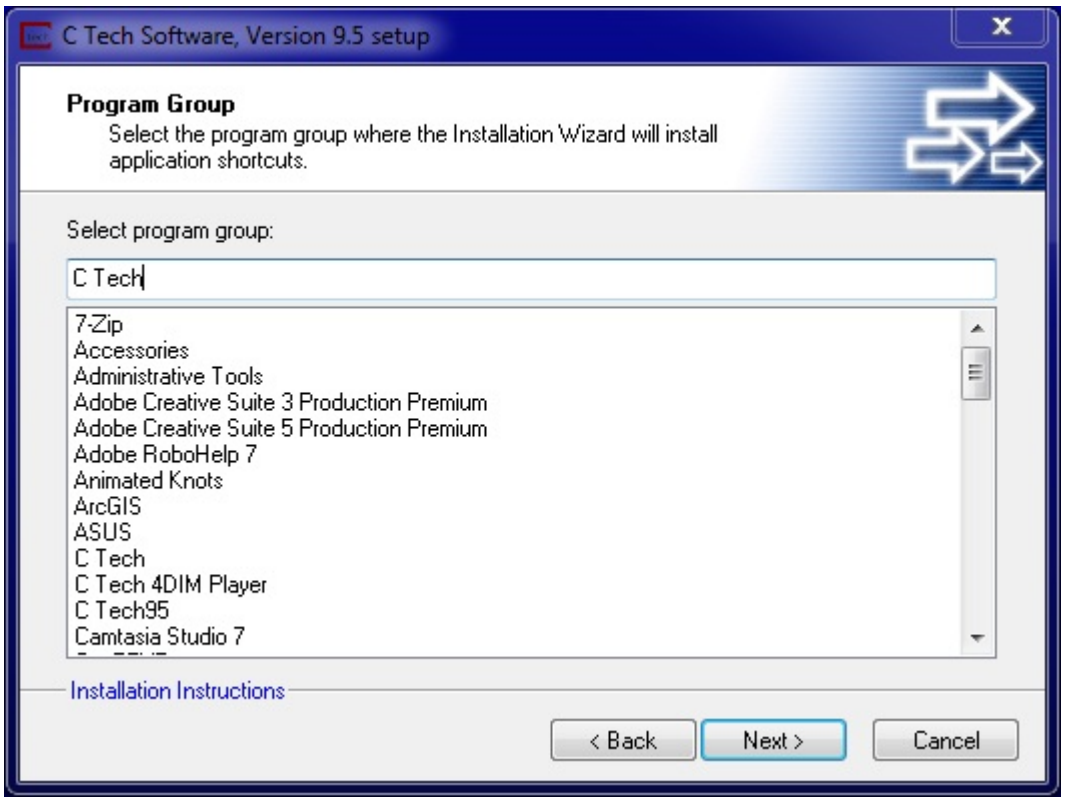
Continue through the welcome screen and release notes until the installer proceeds to a window with C Tech's software license.

Until you toggle "Yes, I agree to all the terms of this license agreement", the installer will not proceed.

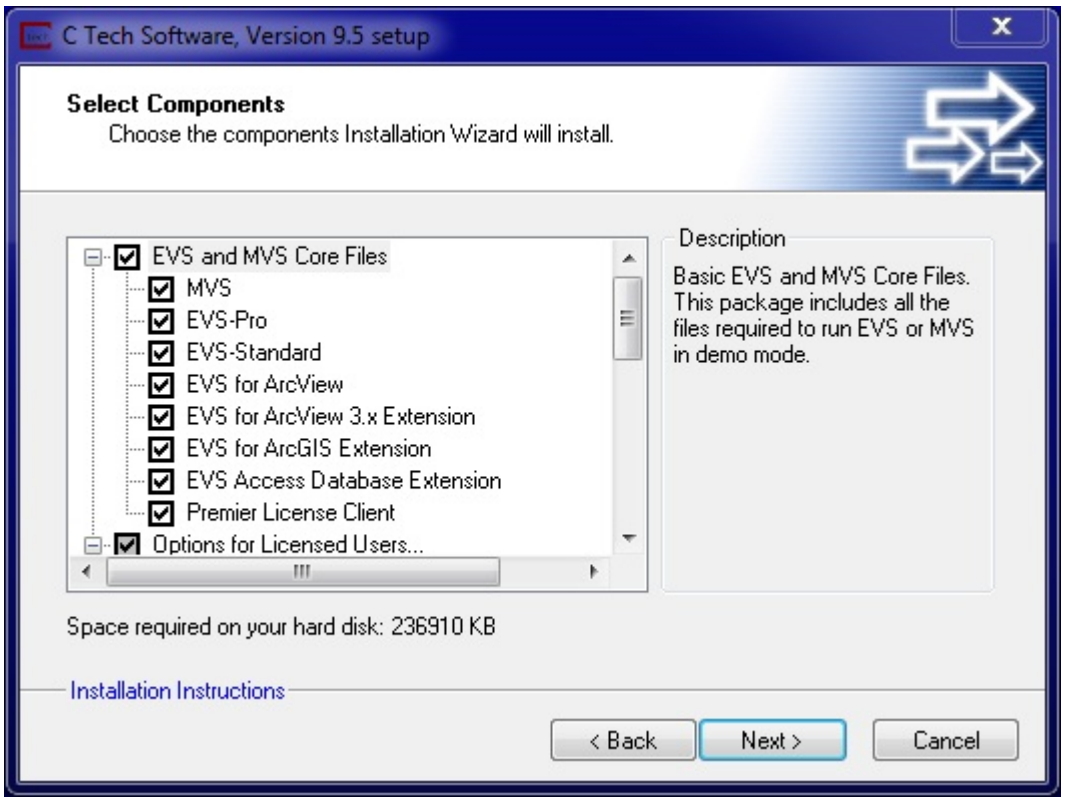
Next specify the folder and drive for installation. We strongly recommend that you install to the root level of one of your hard drives. **You cannot install to folders with a "space" in the name.**



The installer will then prompt you to install to a program group (we recommend you install to the "C Tech" program group.



The next window allows you to select which software products and/or components you wish to install.



The *Demo EVS and MVS* option installs:

1. Preview and Launch Menu
2. EVS-PRO and MVS shortcuts (all shortcuts are always in the Versions subfolder)
3. ESRI Extensions for geology and analyte (e.g. chemistry) data creation
4. EVS Access Database Launcher
5. Imagen animation player
6. Full set of sample data and applications.
7. Setup to use help online at www.ctech.com.

The *Premier License Client* option installs all appropriate software for Premier clients.

With all other windows choose next or OK to proceed. One exception is that if the installer would install system files that are older than your existing files it will prompt you. We recommend that you choose "Yes to All" which will **keep your newer system files.**

Installing C Tech Floating License Server

Installation Overview

C Tech Software is provided on DVD or via download, and is automatically installed using the install program setup.exe (which will autorun if your system is set up to do so). C Tech's Floating License Server is compatible with Windows Vistas, XP, Server 2003 or newer. EVS/MVS (the client software) is compatible only with Windows Vistas and XP.

NOTE: When performing key update operations and using the Key_tools.exe program shown below, you must be logged into the console (physically at the server machine). Remote Desktop and Terminal Services will not run properly nor detect the hardware key.

Windows Vistas, XP, Server 2003 or newer Service

The floating license server software is provided in two different versions (forms). For Windows Vistas, XP, Server 2003 or newer there is a "service" that will auto-install. The service has the advantage that is starts automatically on boot, does not require any user to log in, and uses very little system resources. This is ideal when the server software is to be installed on a company's computer server.

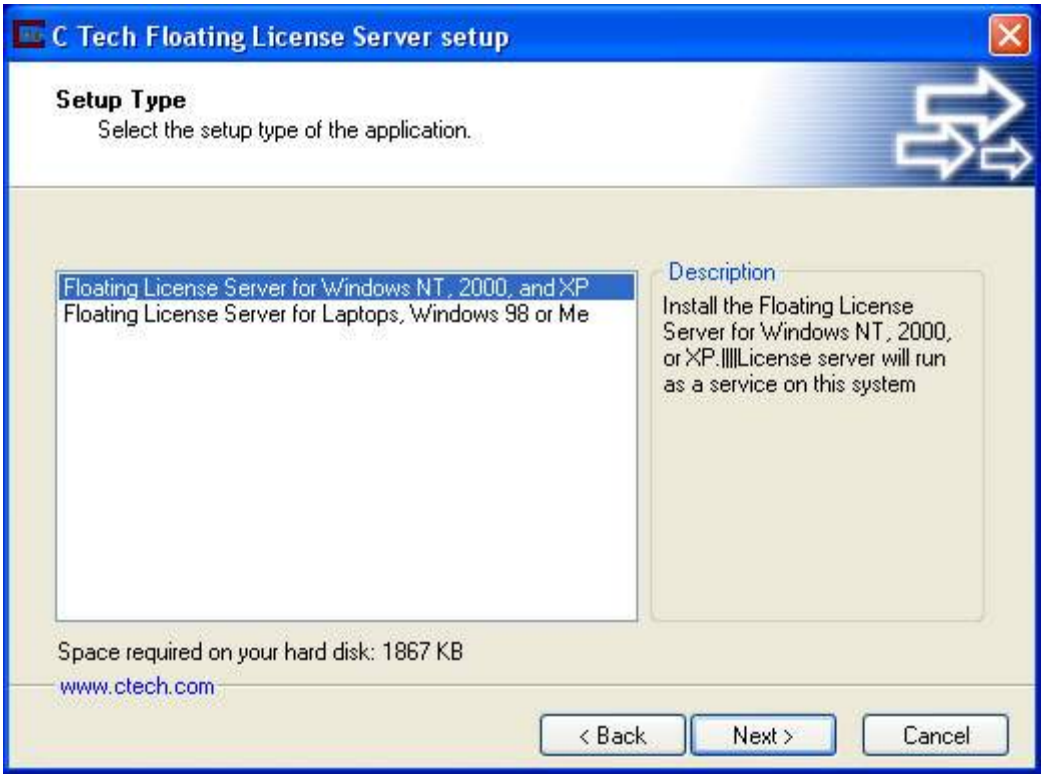
To install the service:

- First run the Floating License Server installation.
- Do not attach your hardware key to the USB port until the installation completes

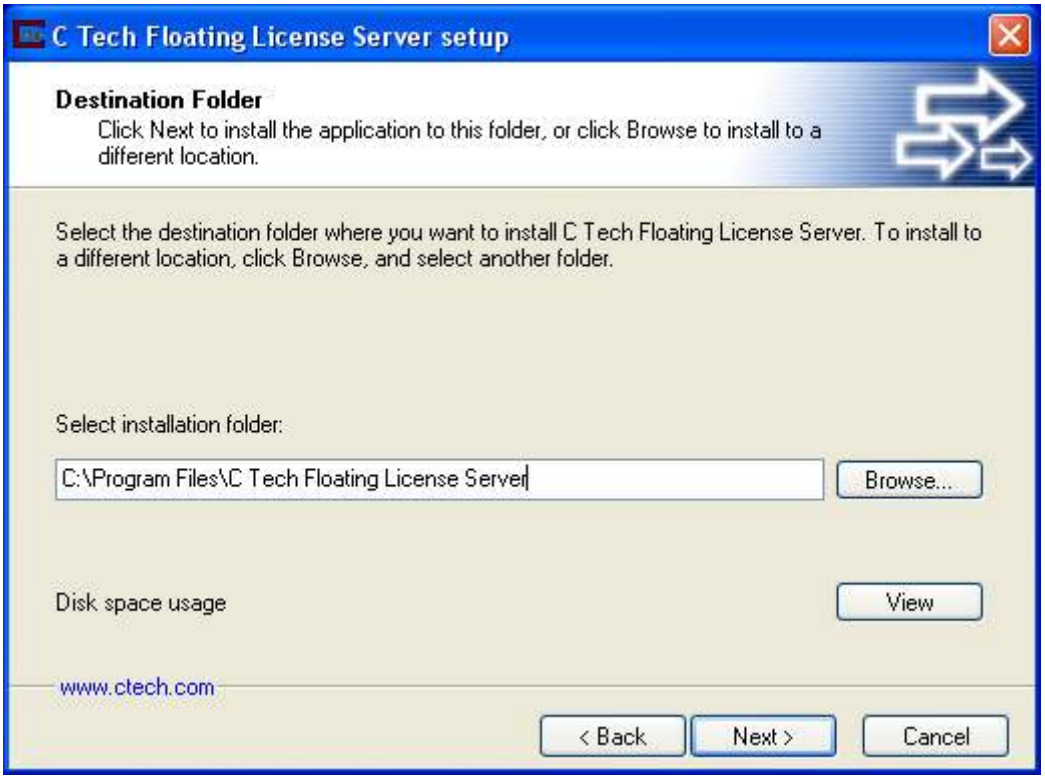
The installation welcome screen will appear next, please read the disclaimers and then select Next to continue with the installation.

Please read through the license agreement on the following screen and if you accept of the terms as they are listed select the checkbox labeled "Yes I agree with all the terms of this license agreement." Select Next to continue.

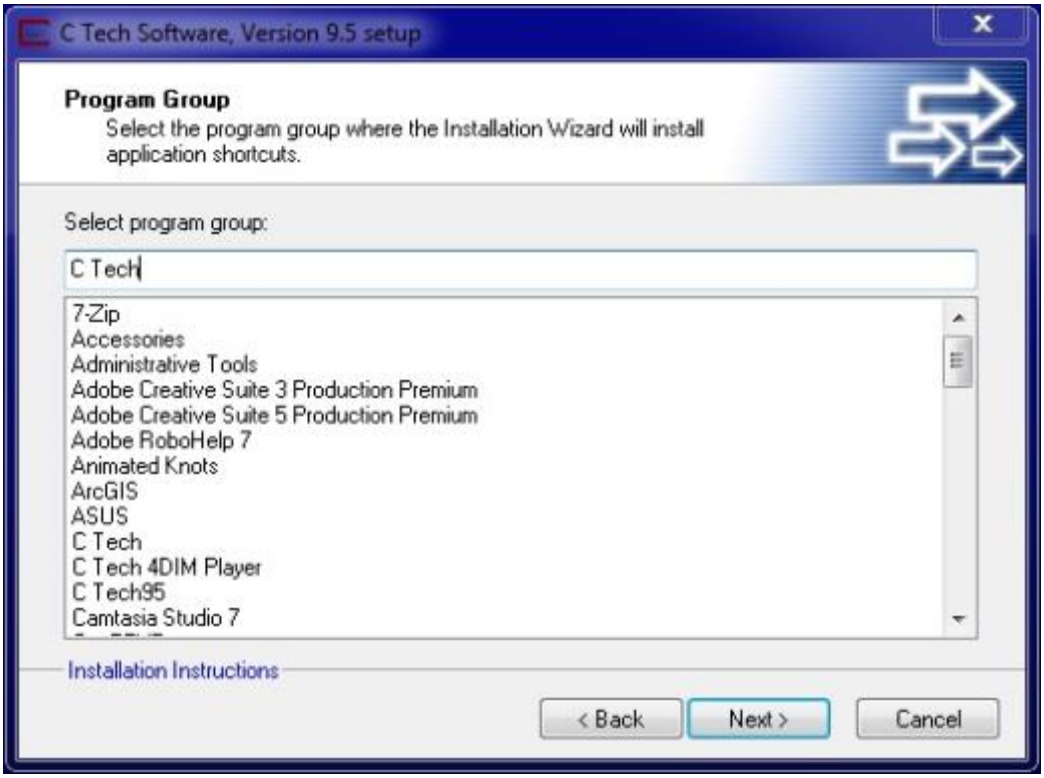
On the "Setup Type" screen select the Floating License Server forWindows Vistas, XP, Server 2003 or newer, once again select Next to continue.



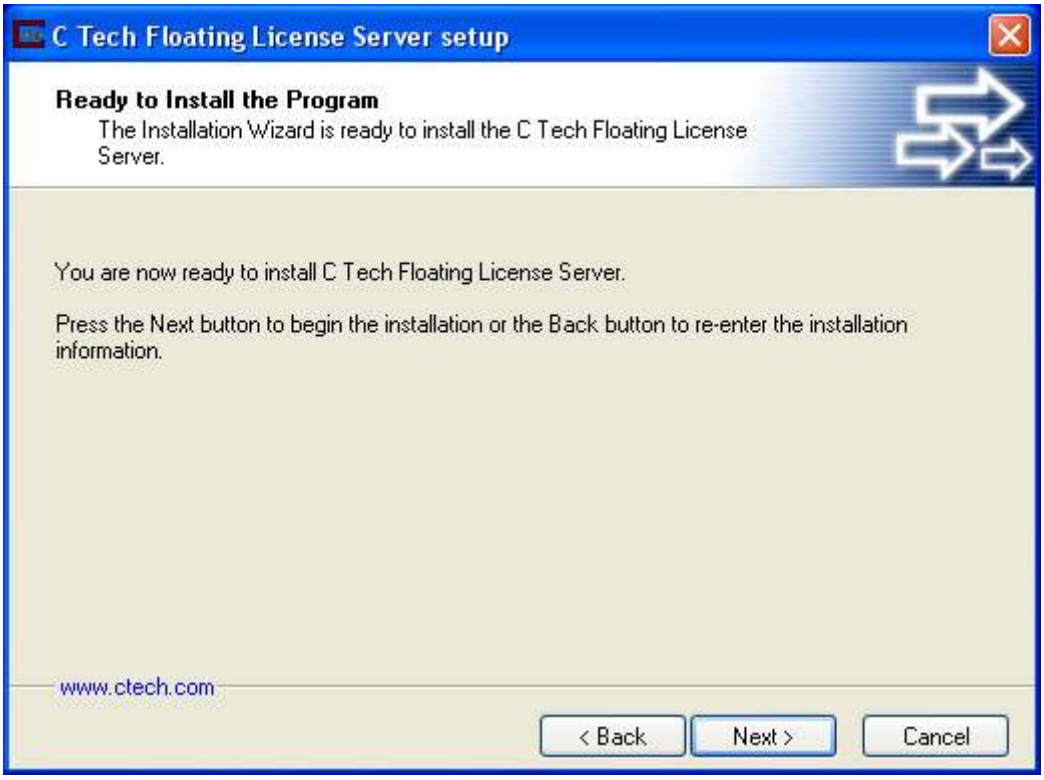
Next select the destination folder for Floating License Server to be installed to, and then select Next to continue.



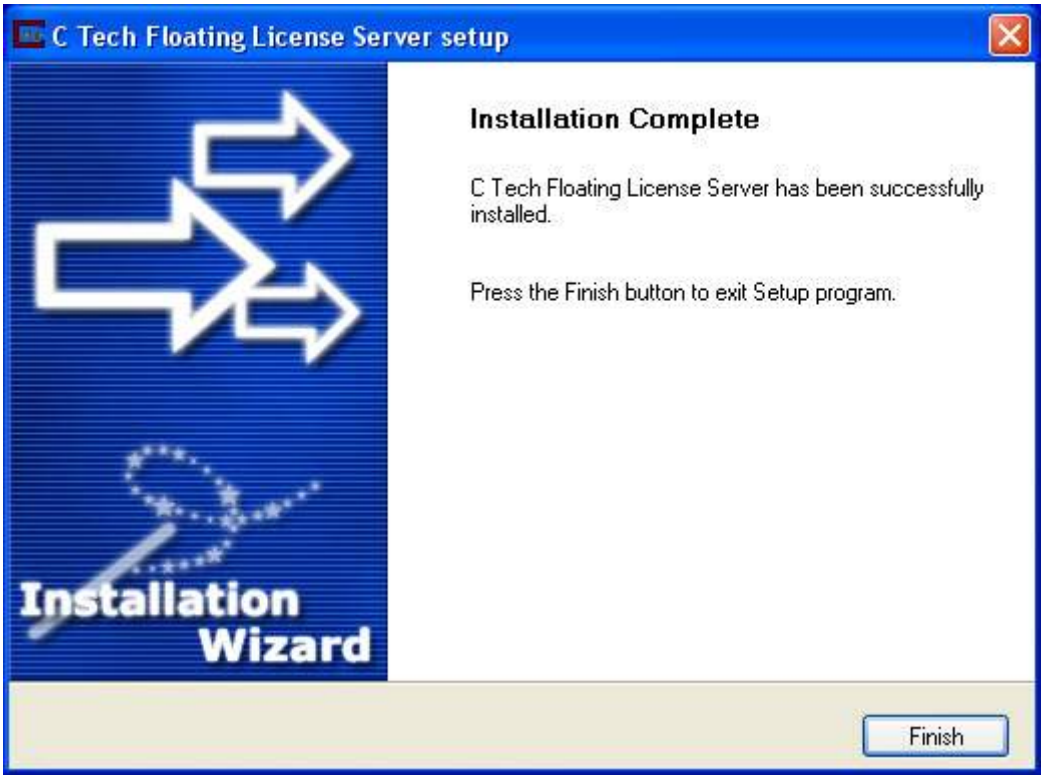
Select the program group you wish to use for the Server. Click Next to continue.



Congratulations you are ready to install the server software! Click Next to continue.



Click finish to exit the installation program.



NOW YOU MUST INSTALL THE CLIENT SOFTWARE

Installing C Tech Floating License Client Software

Installation Overview

C Tech Client Software is provided on CD-ROM, and is automatically installed using the install program setup.exe (which will autorun if your system is set up to do so). EVS/MVS floating license client software is compatible with Windows Vistas and XP. The installation procedure is identical on all operating systems.

Run the EVS or MVS setup program and select the option for the software version you have purchased. See the Detailed Installation Instructions for further information.

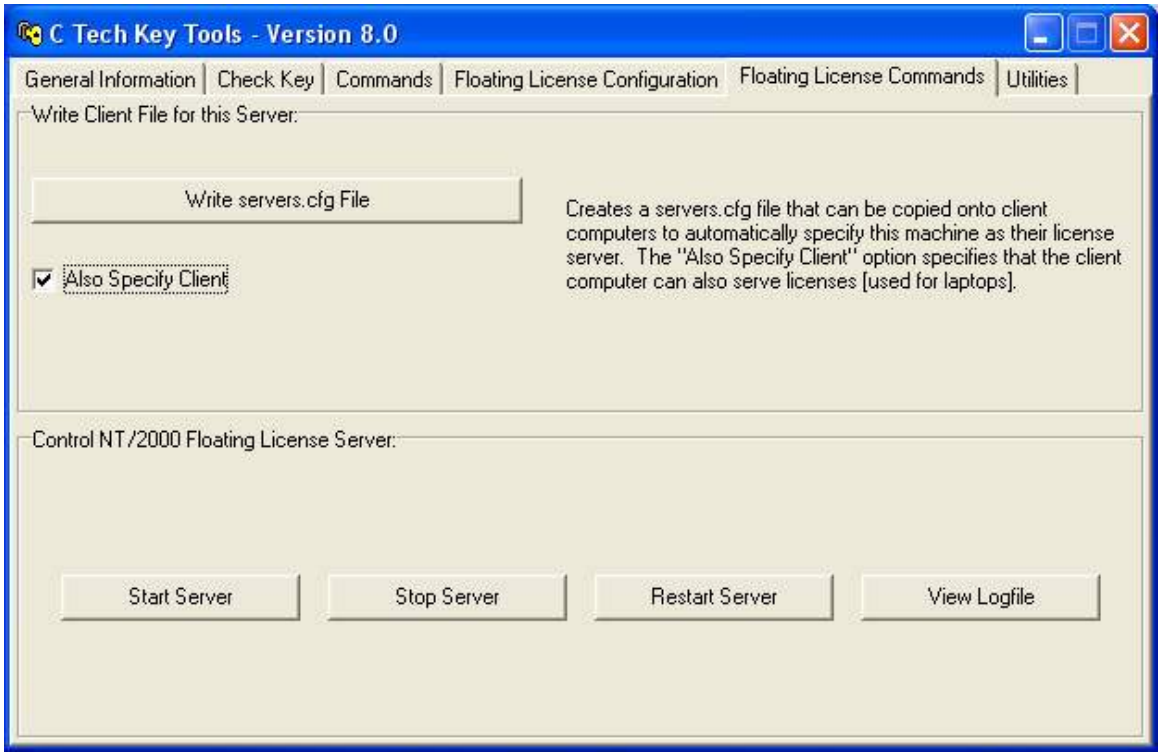
Respond to the prompts of the installation program.

You may uncheck the option to *Install Key Driver*. The hardware key is only installed on the server. If you are installing the server and client software on a computer, the key driver software need only be installed one time. (installing the key driver is harmless if not needed)

Following installation run the KeyTools.exe program in the tools subfolder of the C Tech program group ON THE SERVER MACHINE (the one with the hardware key).

Click on the "Floating License Commands" tab.

If you plan to occasionally run the client software in a standalone mode (by also running the standalone floating license server) so that computer (such as a laptop) can act as a server when separated from the network, check the "Also Specify Client" toggle as shown below.

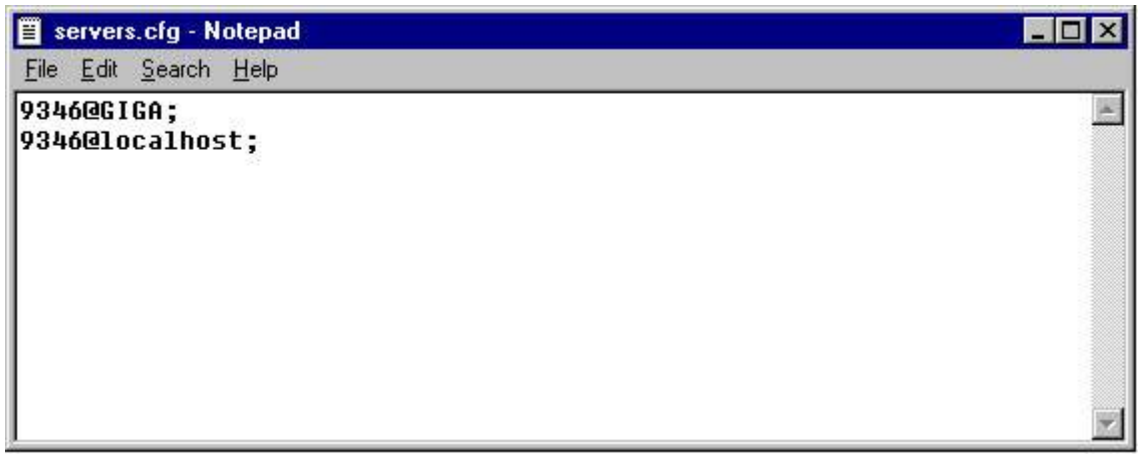


Click on the "Write servers.cfg File" button.

At this point, specify the location of the file servers.cfg. This file is used on all client machines (including the server if it will also be running the software) to specify the location of the server on your network.

The following message will appear to remind you to copy this file to the appropriate folder on each client machine. For EVS-PRO (float) the location would be ctech\bin\pro\servers.cfg.

If you open the file in an editor like notepad you will see the following (in this example the server name was GIGA).



localhost specifies that licenses may be obtained from the same machine upon which the client software is running.

Installing C Tech Premier Floating License Server

Installation Overview

C Tech Premier Floating License Software is provided to the C Tech contact for your company, and is automatically installed using the install program setup.exe (which will autorun if your system is set up to do so). The Premier Floating License Server is compatible only with Windows Vistas, XP, Server 2003 or newer.

MVS Premier (the client software) is compatible only with Windows Vistas and XP.

NOTE: When performing key update operations and using the Key_tools.exe program shown below, you must be logged into the console (physically at the server machine). Remote Desktop and Terminal Services will not run properly nor detect the hardware key.

Windows Vistas, XP, Server 2003 or newer Service

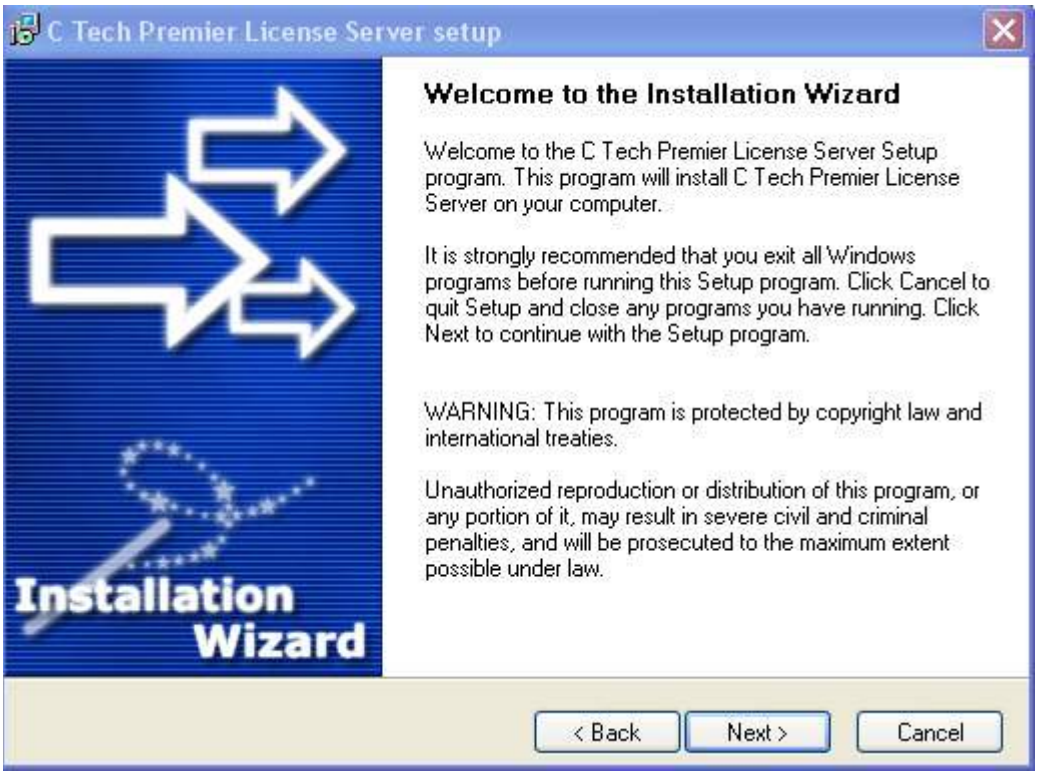
The floating license server software is provided in two different versions (forms). For Windows Vistas, XP, Server 2003 or newer there is a "service"

that will auto-install. The service has the advantage that is starts automatically on boot, does not require any user to log in, and uses very little system resources. This is ideal when the server software is to be installed on a company's computer server.

To install the service:

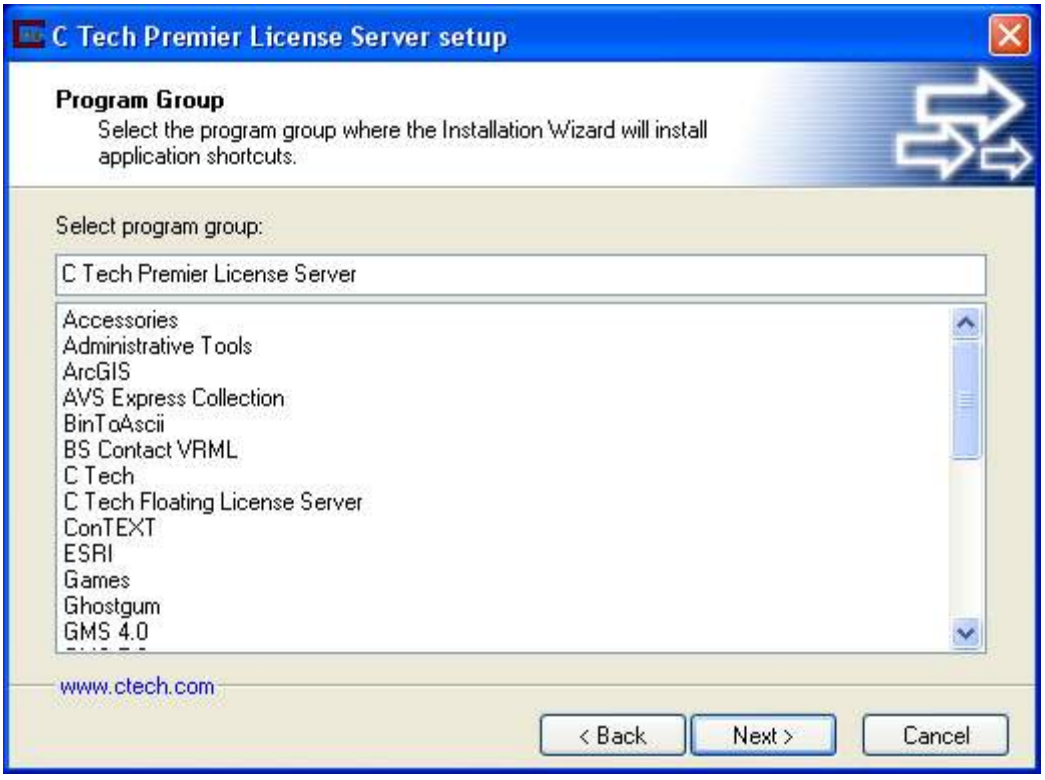
- First run your company's Premier Server installation.
- Do not attach your hardware key to the USB port until the installation completes

The installation welcome screen will appear next, the text "My Company Name" will be replaced with the name of your company, please read the disclaimers and then select Next to continue with the installation.

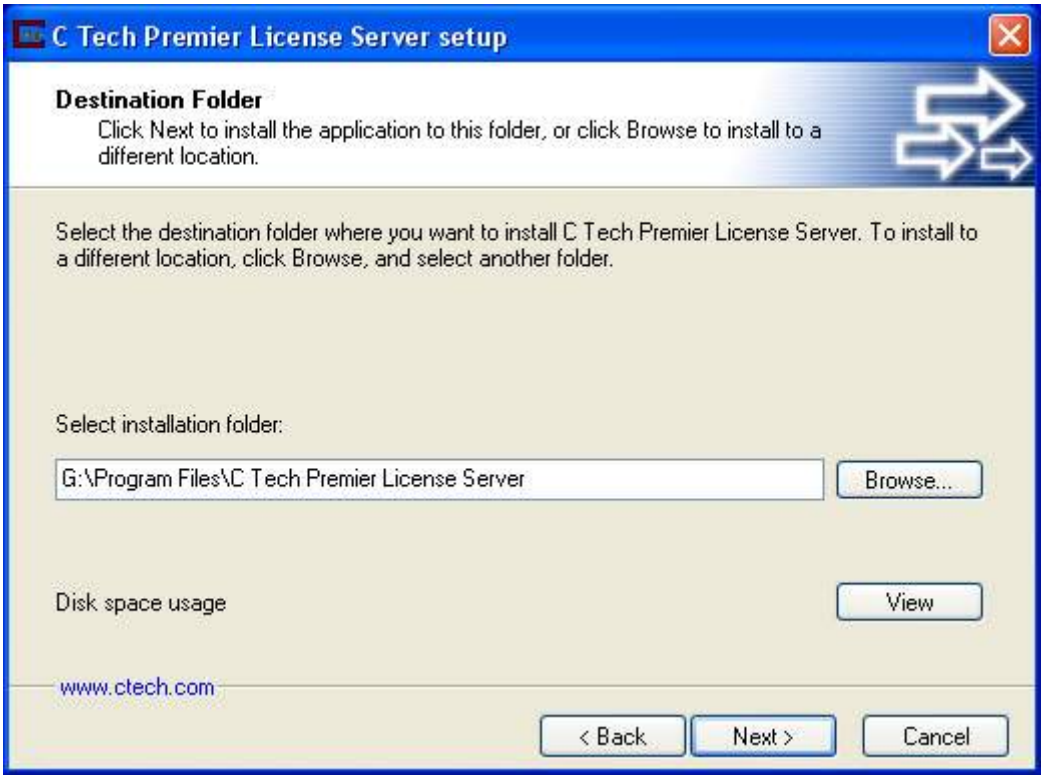


Please read through the license agreement on the following screen and if you accept of the terms as they are listed select the checkbox labeled "Yes I agree with all the terms of this license agreement." Select Next to continue.

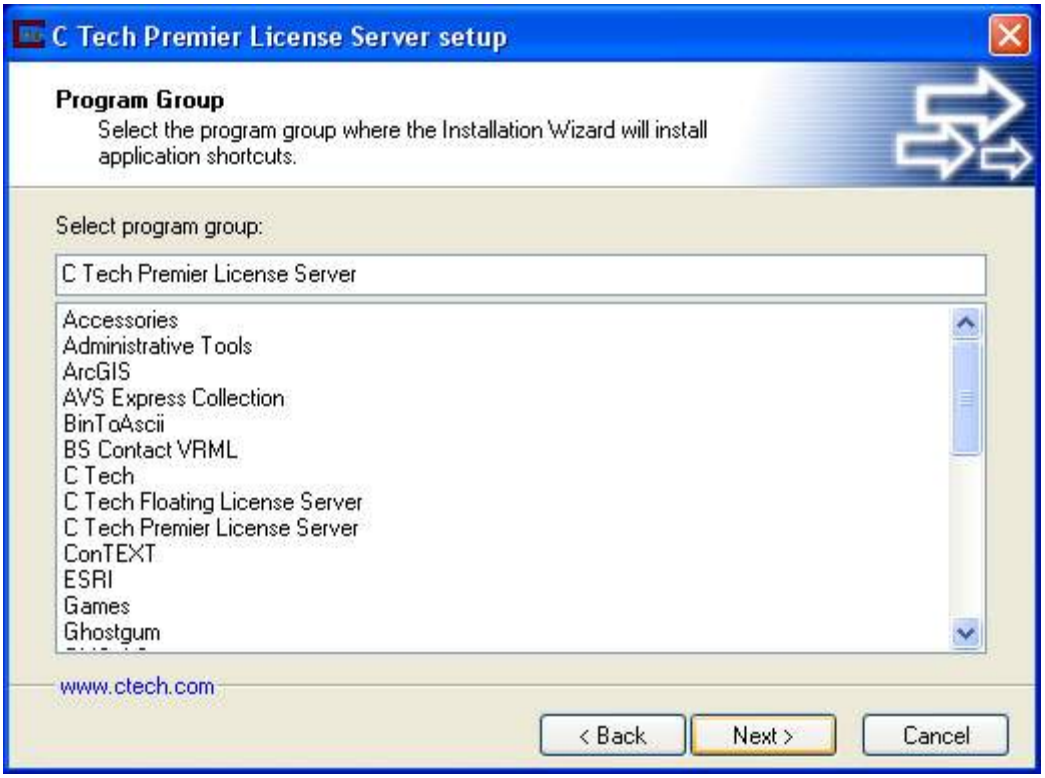
On the "Setup Type" screen select the Floating License Server for Windows Vistas, XP, Server 2003 or newer, once again select Next to continue.



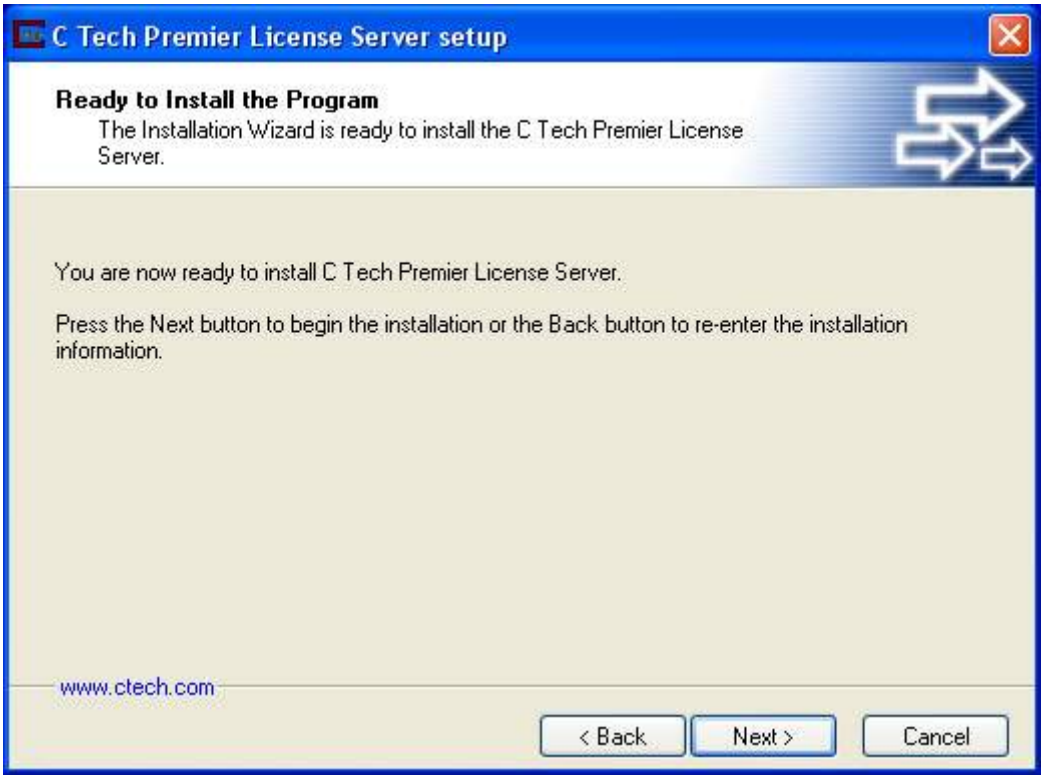
Next select the destination folder for Floating License Server to be installed to, and then select Next to continue.



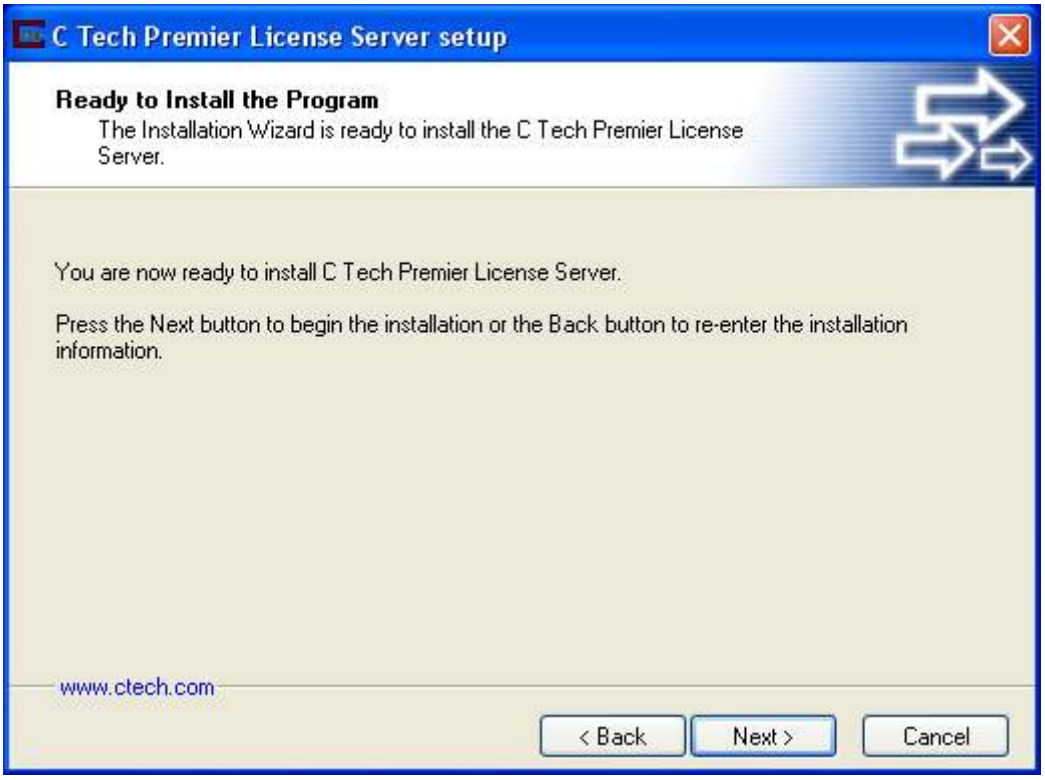
Select the program group you wish to use for the Server. Click Next to continue.



Congratulations you are ready to install the server software! Click Next to continue.



Click finish to exit the installation program.



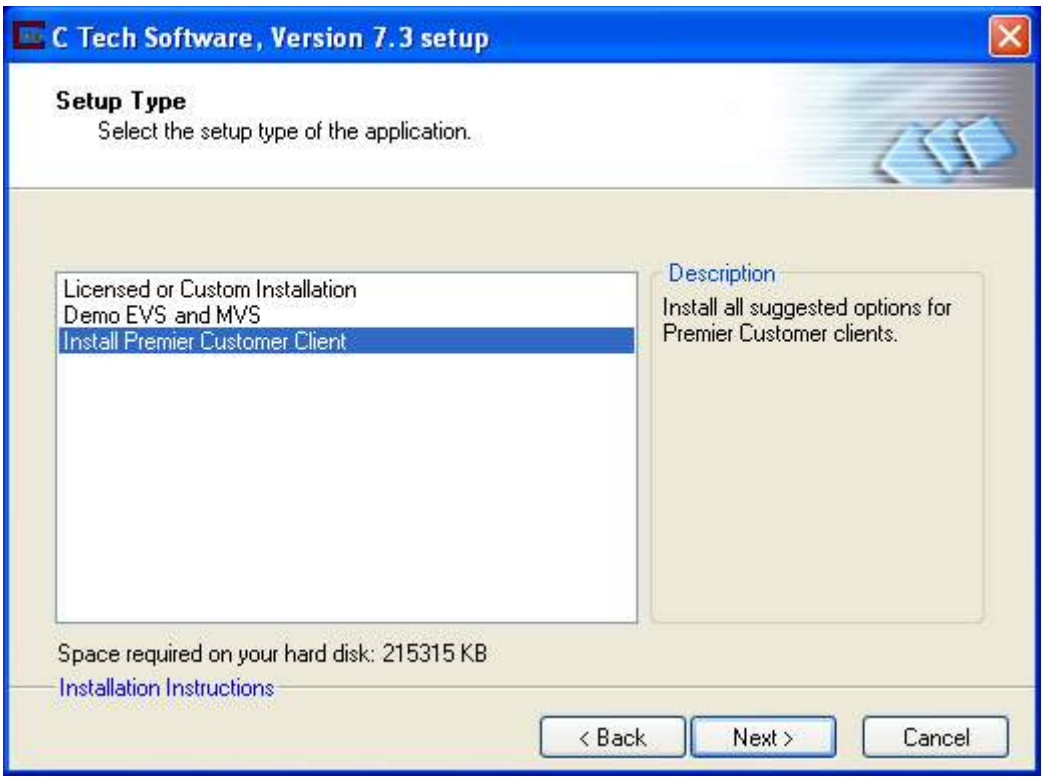
NOW YOU MUST INSTALL THE CLIENT SOFTWARE

Installing Premier Floating License Client Software

Installation Overview

C Tech Premier Floating License Software is provided to the C Tech contact for your company, and is automatically installed using the install program setup.exe (which will autorun if your system is set up to do so). The Premier Floating License Client software is compatible only with Windows Vistas and XP.

Run the EVS/MVS setup program and select the option for "Install Premier Customer Client" in the Setup Type screen.

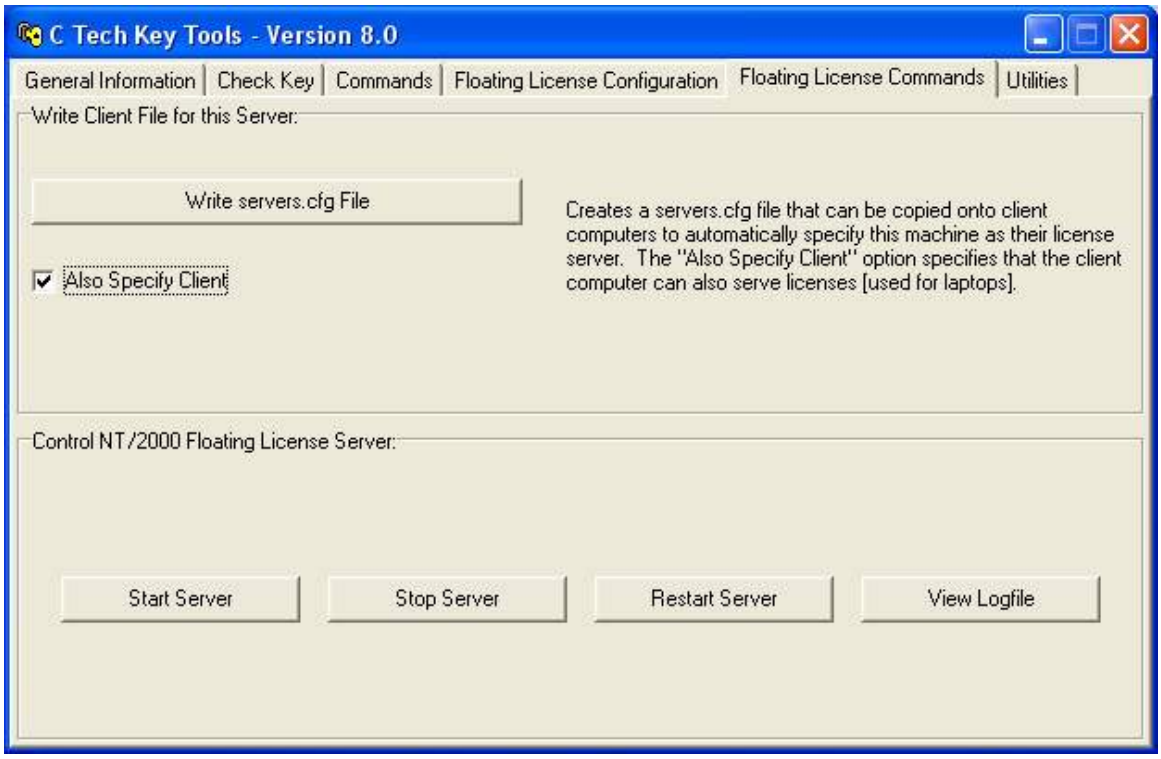


Respond to the prompts of the installation program.

Following installation run the KeyTools.exe program in the tools subfolder of the C Tech program group ON THE SERVER MACHINE (the one with the hardware key).

Click on the "Floating License Commands" tab.

If you plan to occasionally run the client software in a standalone mode (by also running the standalone floating license server) so that computer (such as a laptop) can act as a server when separated from the network, check the "Also Specify Client" toggle as shown below.



Click on the "Write servers.cfg File" button.

At this point, specify the location of the file servers.cfg. This file is used on all client machines (including the server if it will also be running the software) to specify the location of the server on your network.

The following message will appear to remind you to copy this file to the appropriate folder on each client machine. For MVS Premier, the location would be ctech\servers.cfg or ctech\bin\premier\servers.cfg.

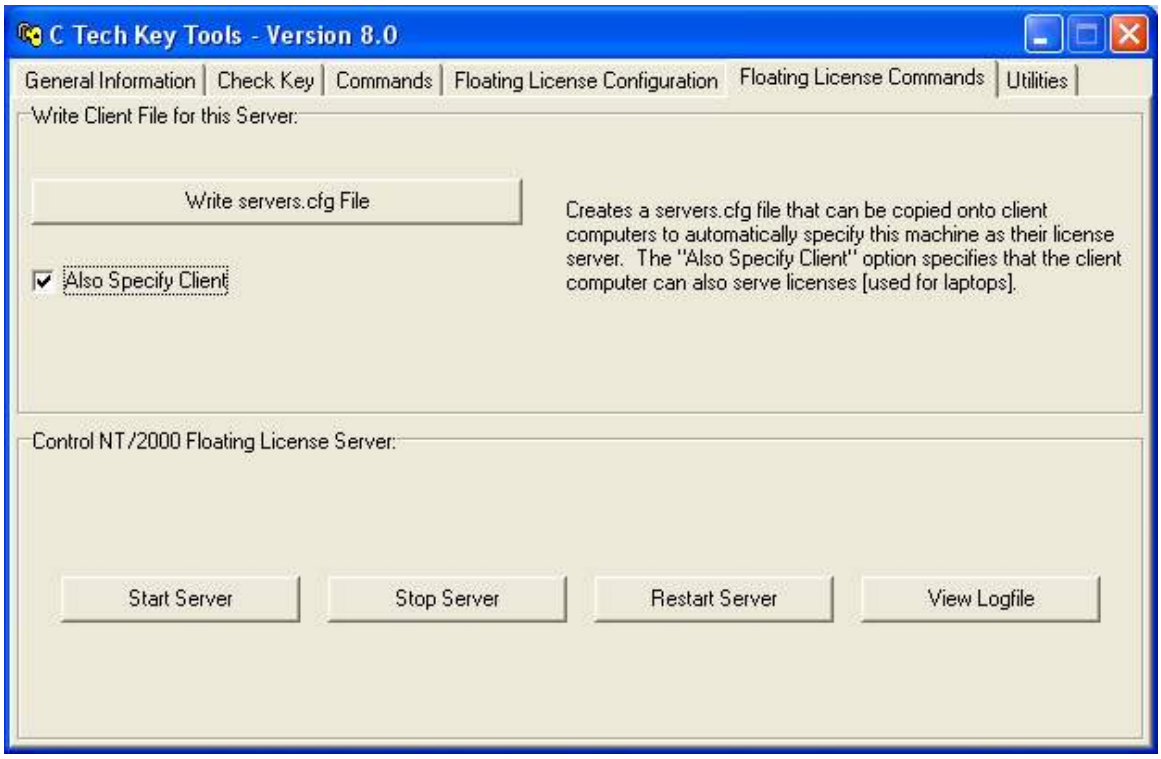
Installing Floating License Standalone on a Laptop

The EVS and MVS floating license versions can be setup to be used in normal floating license mode as well as standalone on a laptop or other computer not connected to a network. The installation differs slightly from the normal client installation of the C Tech Client Software.

First, install the C Tech Floating License Server on your primary server (can be any computer or server on your LAN/WAN/VPN) as well as on your laptop, as instructed. Then, install the C Tech Client Software on the laptop machine. C Tech Client Software is provided on DVD or by download.

EVS/MVS floating license client software is compatible with Windows Vistas and XP. The installation procedure is identical on all operating systems.

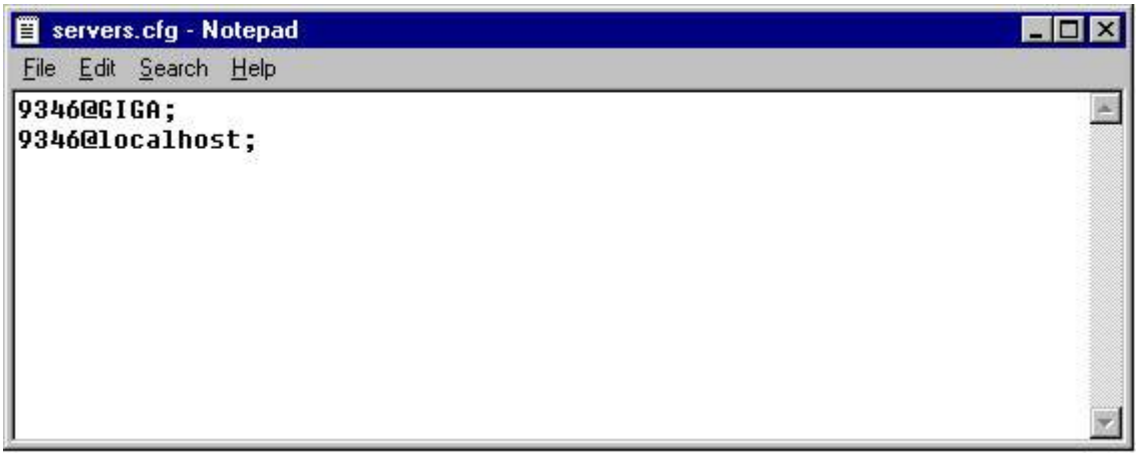
1. Run the setup program and select the software version(s) desired.
2. Respond to the prompts of the installation program.
3. Following installation run the KeyTools.exe program in the C Tech program group ON THE PRIMARY, NETWORKED SERVER MACHINE (with the hardware key attached and the server started).
4. Click on the "Floating License Utilities" tab.
5. To occasionally run the client software in a standalone mode, check the "Also Specify Client" toggle as shown below.



6. Click on the "Write servers.cfg File" button.
7. At this point, specify a location to save servers.cfg. This file will be used on any/all laptops or computers which will run the software without network access. Note: When you wish to run without access to the server, you must move the hardware key to the laptop (disconnected computer) thereby eliminating access for others on your network.

8. You must now copy this file to the appropriate folder on your laptop machine. This should always be the ctech install folder (default would be (c:\ctech\servers.cfg)

If you open the file in an editor like notepad you will see the following (in this example the server name was GIGA).



localhost specifies that licenses may be obtained from the same machine upon which the client software is running, in this case, the laptop.

9. The software should now run normally, provided the laptop is on the same network as the server machine (with the hardware key). In order to use the software on the laptop when it is not connected to the network, you simply need to do a couple of small steps.

a. Turn off the laptop.
b. Move the hardware key onto the laptop's parallel or USB port.
c. Boot the laptop disconnected from the office network. You should now be able to run EVS or MVS.

10. When you are finished, you can return the hardware key to the server. Be sure to always power computers off before removing or attaching keys.

Important Features in C Tech's Software

Fundamentals

Visualization of environmental data in MVS/EVS involves six fundamental steps:

1) Read and pre-process the geologic and/or analytical data to clip the data values into the ranges of interest, and to log transform the data when appropriate;

2) Krige the data to produce two- or three-dimensional grids of data values;

3) Filter the data to produce subsets which contain the regions and ranges of interest, and map the filtered data to a color range appropriate for the visualization objective;

4) Apply mapping and coloring techniques to display data as slices, edges, faces, isolines, surfaces, plumes, glyphs, streamlines, stream-ribbons, etc.;

5) View the output in 3-dimensional space with dynamic rotation, translation and zooming to visualize and emphasize the characteristics of interest in the data set;

6) Output selected views to digital image files (4DIMs, images and animations) for production of hardcopy or video animations (Animation is only available in EVS-PRO and MVS.).

MVS/EVS has been designed to provide streamlined reproducible methods to complete visualization and analyses. The modular structure of the program allows the user to graphically construct their own visualization programs, which can be saved as applications for subsequent use with the same or different data sets.

Data Pre-Processing

MVS/EVS provides automated data preprocessing in all of its kriging modules, and in stand-alone modules, which allow the user to clip a data set to within specified limits, and to take the log base 10 of the data values. Many environmental data sets contain a mixture of geologic and hydrologic data, which are generally smooth and continuous throughout a domain, and chemistry data commonly varies over several orders of magnitude across short distances. Therefore, MVS/EVS provides the capability to quickly preprocess these data sets, thus greatly enhancing the scientist's ability to analyze and visualize the data.

Expert Systems in MVS/EVS

MVS/EVS utilizes expert systems to analyze the input data, construct a multidimensional variogram which is a best fit to the dataset being analyzed, and then perform kriging in the domain to be considered in the visualization. The user is provided the option to specify values for parameters which control the variogram\kriging procedure, and the subsequent display and analysis of the data. One of the fundamental design criteria used in developing MVS/EVS's variogram and kriging algorithms was to produce kriged distributions that honor the measured distributions as closely as possible, and to provide the user with a valid mechanism to compare the modeled and measured domains. MVS/EVS's data posting and surfacing capabilities allow the user to make these comparisons with a minimum amount of effort. This section provides a description of the algorithms used in the variogram computation and kriging modules in MVS/EVS, along with general guidelines for setting parameter values for the variogram and kriging procedures.

Variogram Production

MVS/EVS employs an expert system variogram analysis procedure that examines the spatial distribution and number of points in the input data set,

and calculates a variogram that is a best fit to the data under the constraints imposed upon it by the user. In all of MVS/EVS's variogram algorithms, if a parameter has a default value of 0 and the user does not change it, then no constraints are being placed on the procedure and the algorithm will calculate, use, and return those parameters which provide the best fit of the variogram to the data. For many data sets, the unconstrained analyses will provide a good first cut model of the data, which might be improved by placing some constraints on the procedure. However, in many cases the scientist has additional knowledge of the data which should be appropriately considered in the variogram modeling procedure by constraining certain input parameters. Discussions of the significance of each of the general variogram parameters, and some guidelines that can be used in setting their values, is provided in the following module sections, and in the help pages of each of the kriging modules.

Most of MVS/EVS's kriging modules allow the user to specify the variogram pair search range, variogram range, and minimum range. MVS/EVS's variogram modules utilize a nugget of zero, which cannot be changed, and which basically require the calculated value to be equal to the known value of data points that fall exactly on a grid point in the modeled domain. For most environmental applications, this restriction of the nugget provides much more representative results than allowing the nugget to be greater than zero, and thus allowing estimated data points to be different than the measured data points when they coincide. For some specialized applications however, the user may want to consider a finite nugget, and in these cases MVS/EVS can not be used as it is currently implemented.

Kriging

MVS/EVS’s capabilities to grid and interpolate sparse measured data in three dimensions are unparalleled. MVS/EVS performs all interpolation using an accurate and geostatistically defensible process called Kriging. Kriging is a mathematical process recognized by the EPA as the best and standard means for interpolation and extrapolation of measured data. MVS/EVS provides a user-friendly expert system to drive its Kriging modules lifting the burden of determining optimal variogram parameters from the user. With MVS/EVS, the user can rely on expert system calculated default values to provide quality answers in minimal time.

Kriging is the only data estimation method which also provides statistical measures of goodness. MVS/EVS provides statistical confidence and uncertainty with all estimated parameters. These additional statistical measures are extremely useful in guiding additional site investigation. Our experience is that by using maximum uncertainty to guide site investigations, a 30% reduction in sampling locations can be realized for an equivalent quality of assessment.

As a user defined alternative to confidence and uncertainty, MVS/EVS will provide statistical bounds on the parameter estimate (This feature available only in EVS-PRO& MVS). In other words, MVS/EVS will determine the nominal, minimum, and maximum estimated distribution based on a user specified confidence level. With MVS/EVS, you can now directly answer the

question: *With my limited measured data, to an 80% confidence level, what is the largest and smallest plume I can expect?*

MVS/EVS's Kriging modules utilize a highly efficient algorithm which provides very fast and robust, interpolation and extrapolation of measured data sets. The employment of highly optimized matrix solution methods provides the capabilities to krige very large data sets quickly and easily.

MVS/EVS provides a full spectrum of three-dimensional gridding options, including: rectilinear grids with uniform spacing in x, y, & z directions; rectilinear grids with uniform spacing in x & y directions with z spacing determined by geologic layers; finite difference type grids with variable spacing in x & y directions and z spacing determined by geologic layers; convex hull bounded gridding with z spacing determined by geologic layers; and adaptive gridding which automatically refines gridding in the cell(s) surrounding measured samples to ensure that the interpolated results and isosurfaces accurately honor measured sample data. Adaptive gridding provides an effective resolution that cannot be approximated by any other method. It often provides more accurate results than increasing the number of elements by 100 to 1000 times.

Data Filtering

MVS/EVS provides a rich library of data filtering and subsetting modules. The options include filtering the data based on volumetric subsetting of the data range, performing mathematical manipulation of multiple data fields, and slicing, cutting, and presenting isolines on all or the subsetted regions of the data.

MVS/EVS quickly and easily extracts surface and volumetric subsets of the Kriged (gridded) data. MVS/EVS's modular structure allows the user to perform multiple, serial, subsetting operations using any of a number of nodal data parameters. For example, this functionality provides a mechanism to determine the volumetric subset which is those regions of the domain where cesium concentrations are above 50 pCi/g, soil porosity is above 12%, elevations range from 530 to 595 feet, and the statistical confidence in the cesium concentration is above 60%.

MVS/EVS provides the ability to include any number of arbitrarily placed and oriented slice and cutting planes within the three-dimensional Kriged data domain. Planes are positioned by the user by controlling rotations (about three axes) and position (distance of plane from the domain centroid).

Three-Dimensional Viewing

MVS/EVS's three dimensional viewer allows the user to perform real time rotations and manipulations on the image displayed in the viewer. Selection and manipulation of the color and material properties which control the rendering of individual objects is also allowed

MVS/EVS can display an unlimited number of objects simultaneously. All objects are truly three-dimensional and objects to be displayed together must be in the same coordinate system. MVS/EVS has the ability to display boreholes colored according to measured concentration, or to display boreholes whose color alternates according to a user defined depth interval

and to display the measured data samples as sized and colored spheres. Spheres and boreholes can also be exploded by geologic layer when geology information is also available.

MVS/EVS provides complete interactive control over viewing perspective, azimuth, elevation, scale and background color. Numerous other viewing parameters can also be controlled such as object rendering method (shading, outlining, etc.), object transparency, lighting (number and color and type of lights), background color, and more. Isosurface level is user controllable (and can be animated), and plots can be labeled within MVS/EVS or bitmap images can be exported to other applications.

Output & Image Production

Because MVS/EVS runs under all versions of Microsoft Windows operating systems, there are numerous options for printing output. MVS/EVS renders each scene in a user defined resolution. The Viewer window may be "captured" to the clipboard and pasted into another application. MVS/EVS allows printing directly from the program at a user specified resolution. MVS/EVS also includes an Output_Images module which will create images in many image formats, including Windows Bitmap (.bmp) files, Portable Network Graphic (.png) files, and many other compressed and uncompressed formats. All bitmap images can be imported into other applications such as CorelDraw™, or Adobe Illustrator ™ to add additional annotation and to print.

MVS/EVS also provides the ability to output any object (isosurfaces, labeled isolines, axes, etc.) in industry standard files, including DXF and Shapefiles. This output format provides a vector (versus raster) definition of each object. This provides the ability to integrate MVS/EVS output with CAD drawings or to plot objects in large scale (e.g. poster size) with vector resolution, or to bring output into GIS programs.

In addition to printing, MVS and EVS-Pro provide easy to use, powerful capabilities to create animation sequences. These animations can be produced as Windows Audio-Visual Interleaved (.avi) data files and (with special hardware) written directly to NTSC or PAL video formats such as DVD. The ability to produce animations showing rotation of 3D objects, or variations in subsetting level provides an invaluable capability that printed graphics cannot equal.

MVS and EVS-Pro also provide the ability to create C Tech's 4DIM files, which are fully interactive 3D animations. These can be viewed with the Playback_4DIM module in MVS/EVS, or with a standalone 4DIM player.

Automation

Because MVS/EVS provides a mechanism to save, and subsequently load, networks of modules (as a custom application), and also has a powerful scripting (macro) language, the process of generating initial figures can be fully automated. Because of MVS/EVS's expert system driven Kriging, no user intervention is required to produce reasonable results.

Important Features in C Tech's Software

Fundamentals

Visualization of environmental data in MVS/EVS involves six fundamental steps:

1) Read and pre-process the geologic and/or analytical data to clip the data values into the ranges of interest, and to log transform the data when appropriate;

2) Krige the data to produce two- or three-dimensional grids of data values;

3) Filter the data to produce subsets which contain the regions and ranges of interest, and map the filtered data to a color range appropriate for the visualization objective;

4) Apply mapping and coloring techniques to display data as slices, edges, faces, isolines, surfaces, plumes, glyphs, streamlines, stream-ribbons, etc.;

5) View the output in 3-dimensional space with dynamic rotation, translation and zooming to visualize and emphasize the characteristics of interest in the data set;

6) Output selected views to digital image files (4DIMs, images and animations) for production of hardcopy or video animations (Animation is only available in EVS-PRO and MVS.).

MVS/EVS has been designed to provide streamlined reproducible methods to complete visualization and analyses. The modular structure of the program allows the user to graphically construct their own visualization programs, which can be saved as applications for subsequent use with the same or different data sets.

Data Pre-Processing

MVS/EVS provides automated data preprocessing in all of its kriging modules, and in stand-alone modules, which allow the user to clip a data set to within specified limits, and to take the log base 10 of the data values. Many environmental data sets contain a mixture of geologic and hydrologic data, which are generally smooth and continuous throughout a domain, and chemistry data commonly varies over several orders of magnitude across short distances. Therefore, MVS/EVS provides the capability to quickly preprocess these data sets, thus greatly enhancing the scientist's ability to analyze and visualize the data.

Expert Systems in MVS/EVS

MVS/EVS utilizes expert systems to analyze the input data, construct a multidimensional variogram which is a best fit to the dataset being analyzed, and then perform kriging in the domain to be considered in the visualization. The user is provided the option to specify values for parameters which control the variogram\kriging procedure, and the subsequent display and analysis of the data. One of the fundamental design criteria used in developing MVS/EVS's variogram and kriging algorithms was to produce kriged distributions that honor the measured distributions as closely as possible, and to provide the user with a valid mechanism to compare the modeled and measured domains. MVS/EVS's data posting and surfacing

capabilities allow the user to make these comparisons with a minimum amount of effort. This section provides a description of the algorithms used in the variogram computation and kriging modules in MVS/EVS, along with general guidelines for setting parameter values for the variogram and kriging procedures.

Variogram Production

MVS/EVS employs an expert system variogram analysis procedure that examines the spatial distribution and number of points in the input data set, and calculates a variogram that is a best fit to the data under the constraints imposed upon it by the user. In all of MVS/EVS's variogram algorithms, if a parameter has a default value of 0 and the user does not change it, then no constraints are being placed on the procedure and the algorithm will calculate, use, and return those parameters which provide the best fit of the variogram to the data. For many data sets, the unconstrained analyses will provide a good first cut model of the data, which might be improved by placing some constraints on the procedure. However, in many cases the scientist has additional knowledge of the data which should be appropriately considered in the variogram modeling procedure by constraining certain input parameters. Discussions of the significance of each of the general variogram parameters, and some guidelines that can be used in setting their values, is provided in the following module sections, and in the help pages of each of the kriging modules.

Most of MVS/EVS's kriging modules allow the user to specify the variogram pair search range, variogram range, and minimum range. MVS/EVS's variogram modules utilize a nugget of zero, which cannot be changed, and which basically require the calculated value to be equal to the known value of data points that fall exactly on a grid point in the modeled domain. For most environmental applications, this restriction of the nugget provides much more representative results than allowing the nugget to be greater than zero, and thus allowing estimated data points to be different than the measured data points when they coincide. For some specialized applications however, the user may want to consider a finite nugget, and in these cases MVS/EVS can not be used as it is currently implemented.

Kriging

MVS/EVS’s capabilities to grid and interpolate sparse measured data in three dimensions are unparalleled. MVS/EVS performs all interpolation using an accurate and geostatistically defensible process called Kriging. Kriging is a mathematical process recognized by the EPA as the best and standard means for interpolation and extrapolation of measured data. MVS/EVS provides a user-friendly expert system to drive its Kriging modules lifting the burden of determining optimal variogram parameters from the user. With MVS/EVS, the user can rely on expert system calculated default values to provide quality answers in minimal time.

Kriging is the only data estimation method which also provides statistical measures of goodness. MVS/EVS provides statistical confidence and uncertainty with all estimated parameters. These additional statistical

measures are extremely useful in guiding additional site investigation. Our experience is that by using maximum uncertainty to guide site investigations, a 30% reduction in sampling locations can be realized for an equivalent quality of assessment.

As a user defined alternative to confidence and uncertainty, MVS/EVS will provide statistical bounds on the parameter estimate (This feature available only in EVS-PRO& MVS). In other words, MVS/EVS will determine the nominal, minimum, and maximum estimated distribution based on a user specified confidence level. With MVS/EVS, you can now directly answer the question: *With my limited measured data, to an 80% confidence level, what is the largest and smallest plume I can expect?*

MVS/EVS's Kriging modules utilize a highly efficient algorithm which provides very fast and robust, interpolation and extrapolation of measured data sets. The employment of highly optimized matrix solution methods provides the capabilities to krige very large data sets quickly and easily.

MVS/EVS provides a full spectrum of three-dimensional gridding options, including: rectilinear grids with uniform spacing in x, y, & z directions; rectilinear grids with uniform spacing in x & y directions with z spacing determined by geologic layers; finite difference type grids with variable spacing in x & y directions and z spacing determined by geologic layers; convex hull bounded gridding with z spacing determined by geologic layers; and adaptive gridding which automatically refines gridding in the cell(s) surrounding measured samples to ensure that the interpolated results and isosurfaces accurately honor measured sample data. Adaptive gridding provides an effective resolution that cannot be approximated by any other method. It often provides more accurate results than increasing the number of elements by 100 to 1000 times.

Data Filtering

MVS/EVS provides a rich library of data filtering and subsetting modules. The options include filtering the data based on volumetric subsetting of the data range, performing mathematical manipulation of multiple data fields, and slicing, cutting, and presenting isolines on all or the subsetted regions of the data.

MVS/EVS quickly and easily extracts surface and volumetric subsets of the Kriged (gridded) data. MVS/EVS's modular structure allows the user to perform multiple, serial, subsetting operations using any of a number of nodal data parameters. For example, this functionality provides a mechanism to determine the volumetric subset which is those regions of the domain where cesium concentrations are above 50 pCi/g, soil porosity is above 12%, elevations range from 530 to 595 feet, and the statistical confidence in the cesium concentration is above 60%.

MVS/EVS provides the ability to include any number of arbitrarily placed and oriented slice and cutting planes within the three-dimensional Kriged data domain. Planes are positioned by the user by controlling rotations (about three axes) and position (distance of plane from the domain centroid).

Three-Dimensional Viewing

MVS/EVS's three dimensional viewer allows the user to perform real time rotations and manipulations on the image displayed in the viewer. Selection and manipulation of the color and material properties which control the rendering of individual objects is also allowed

MVS/EVS can display an unlimited number of objects simultaneously. All objects are truly three-dimensional and objects to be displayed together must be in the same coordinate system. MVS/EVS has the ability to display boreholes colored according to measured concentration, or to display boreholes whose color alternates according to a user defined depth interval and to display the measured data samples as sized and colored spheres. Spheres and boreholes can also be exploded by geologic layer when geology information is also available.

MVS/EVS provides complete interactive control over viewing perspective, azimuth, elevation, scale and background color. Numerous other viewing parameters can also be controlled such as object rendering method (shading, outlining, etc.), object transparency, lighting (number and color and type of lights), background color, and more. Isosurface level is user controllable (and can be animated), and plots can be labeled within MVS/EVS or bitmap images can be exported to other applications.

Output & Image Production

Because MVS/EVS runs under all versions of Microsoft Windows operating systems, there are numerous options for printing output. MVS/EVS renders each scene in a user defined resolution. The Viewer window may be "captured" to the clipboard and pasted into another application. MVS/EVS allows printing directly from the program at a user specified resolution. MVS/EVS also includes an Output_Images module which will create images in many image formats, including Windows Bitmap (.bmp) files, Portable Network Graphic (.png) files, and many other compressed and uncompressed formats. All bitmap images can be imported into other applications such as CorelDraw™, or Adobe Illustrator ™ to add additional annotation and to print.

MVS/EVS also provides the ability to output any object (isosurfaces, labeled isolines, axes, etc.) in industry standard files, including DXF and Shapefiles. This output format provides a vector (versus raster) definition of each object. This provides the ability to integrate MVS/EVS output with CAD drawings or to plot objects in large scale (e.g. poster size) with vector resolution, or to bring output into GIS programs.

In addition to printing, MVS and EVS-Pro provide easy to use, powerful capabilities to create animation sequences. These animations can be produced as Windows Audio-Visual Interleaved (.avi) data files and (with special hardware) written directly to NTSC or PAL video formats such as DVD. The ability to produce animations showing rotation of 3D objects, or variations in subsetting level provides an invaluable capability that printed graphics cannot equal.

MVS and EVS-Pro also provide the ability to create C Tech's 4DIM files, which are fully interactive 3D animations. These can be viewed with the Playback_4DIM module in MVS/EVS, or with a standalone 4DIM player.

Automation

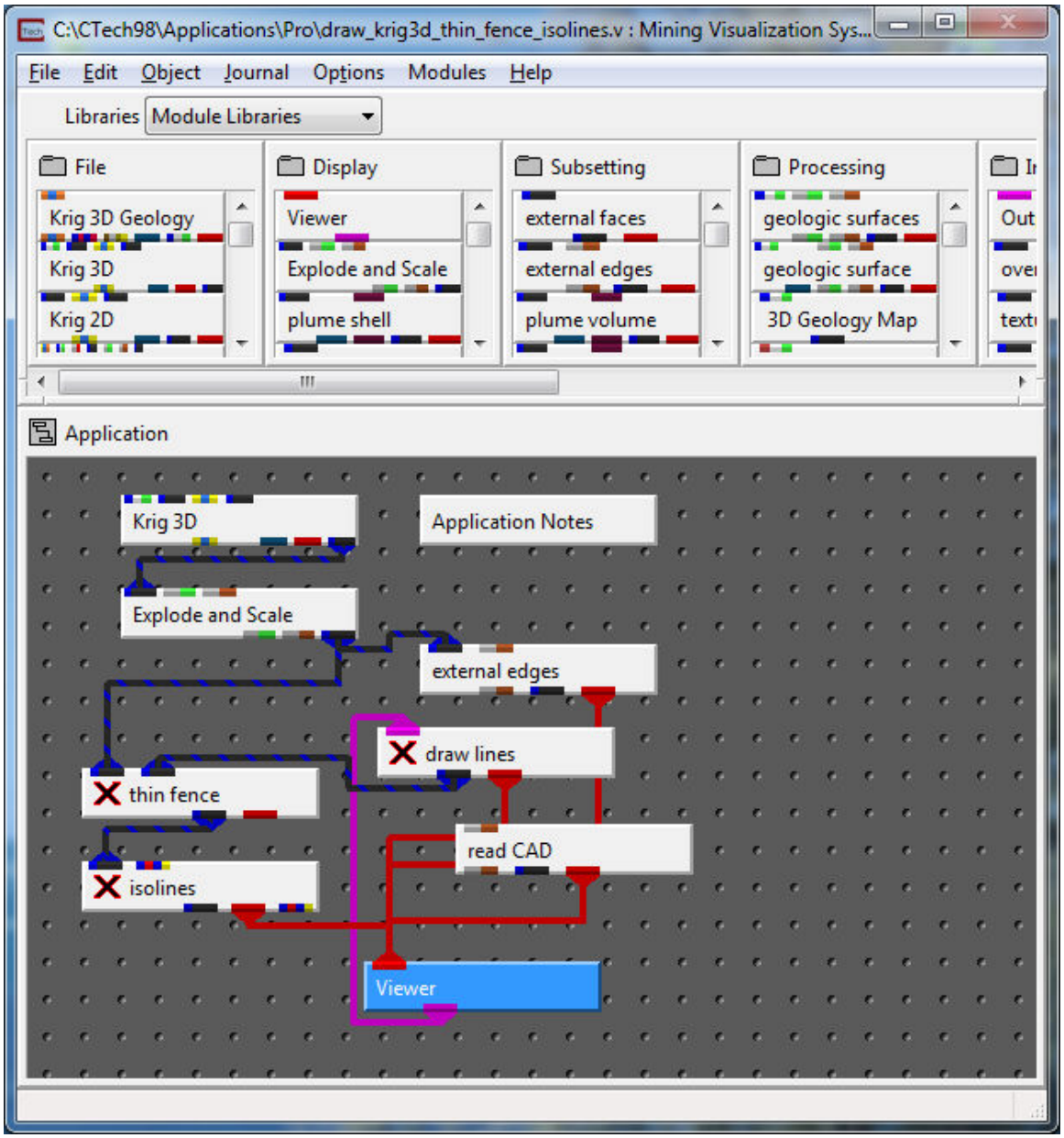
Because MVS/EVS provides a mechanism to save, and subsequently load, networks of modules (as a custom application), and also has a powerful scripting (macro) language, the process of generating initial figures can be fully automated. Because of MVS/EVS's expert system driven Kriging, no user intervention is required to produce reasonable results.

Getting Familiar With The MVS/EVS Environment

Starting MVS/EVS

If you have not already done so, start MVS/EVS at this time. To start MVS/EVS, double-click on the *appropriate* MVS or EVS icon located in the programs listing of your Windows Start Menu.

Visual Programming

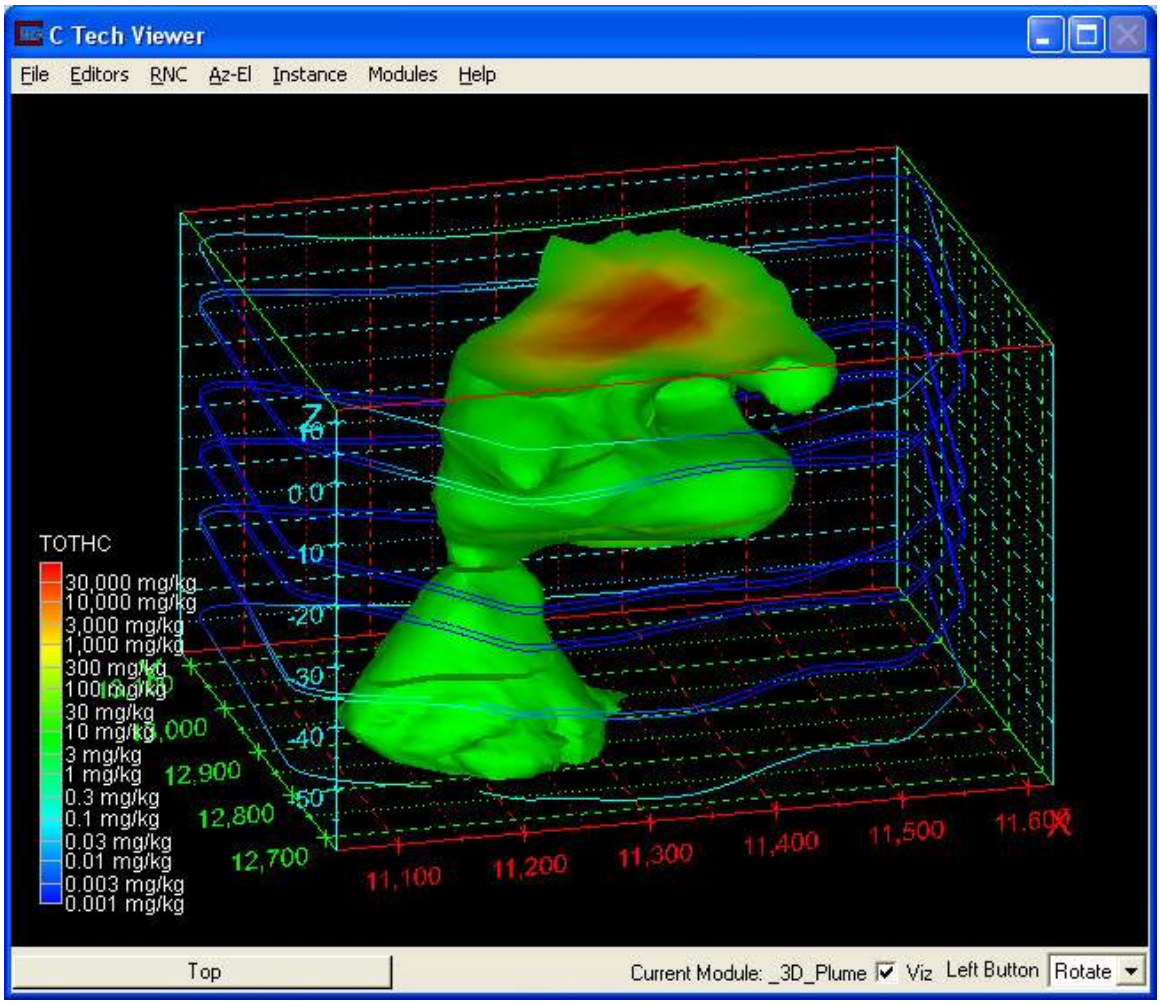


MVS/EVS uses a visual programming paradigm. The MVS/EVS Main Window has two primary parts: the *Module Libraries*, and the *Applicationarea*, as shown in the figure above. The *Applicationarea* is the primary tool that is used to build customized networks of modules (applications), and those networks (applications) are constructed from modules (modular programs) which are located in the MVS/EVS Module Library. Each module is represented by a small rectangular object that you drag into the *Applicationarea* and connect to other modules to construct *applications.*

Each module in MVS/EVS has its own set of user controls based on a graphical user interface. Modules are interconnected by mouse actions by making graphical "connections" between output and input ports of like color. Individual port colors represent different data structures which are passed

between modules. The collection of modules forms a network which is essentially a custom application. When a network is saved as an application it defines those modules which are to be included in the application and how those modules are to be used. MVS/EVS's network editor, although very easy to use, provides a graphical object oriented programming environment to create custom applications without the need for "programming" skills. Saved applications provide all the functionality of a custom application to a user without his/her needing to learn or understand program creation techniques.

The *Applicationarea* is the workspace where you use the modules to build networks. The MVS/EVS Module Libraries are like a toolbox, in that it there are a number of different tools that serve different purposes, but cannot be used until they are taken out of the toolbox. Similarly, the *Applicationarea* is similar to a workbench: it is the place where the tools are used to create visualizations. Most modules in the library have *input* and/or *output ports*. These are colored regions (ports) on the top and/or bottom of the module objects which represent the pipelines through which data flows to and from each module. A collection of modules that have their ports connected by pipelines comprise an MVS/EVS Application (network). The figure above shows a moderate complexity application which produced the figure below.



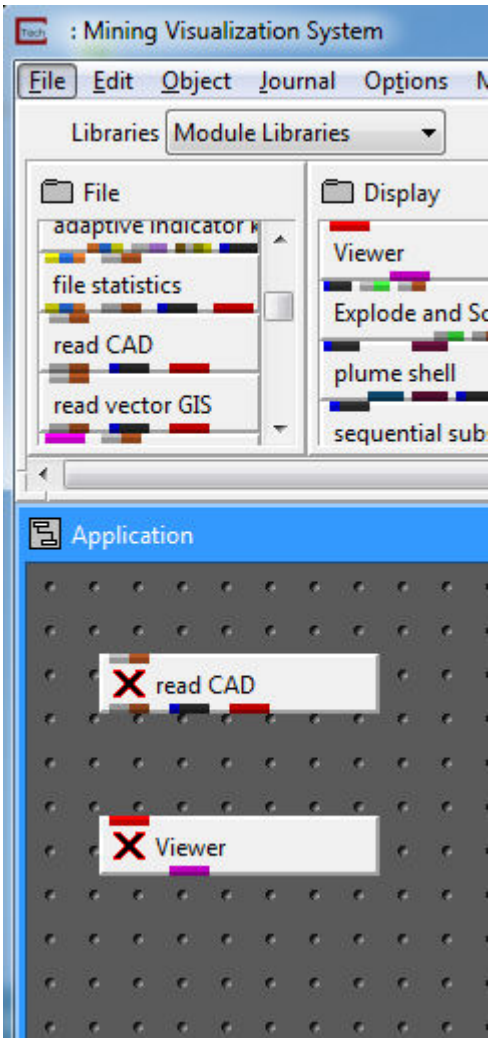
Creating a Simple Network

Let's build a simple network just to get an overview of how the MVS/EVS's visual programming environment works. In the MVS/EVS Module Library, go to the sublibrary (first column of modules), labeled **File**, and select **read_CAD**. You will have to scroll down in the library to find **read_CAD**. Use the standard windows scroll bar to the right of these modules, scrolling until the desired library is displayed.

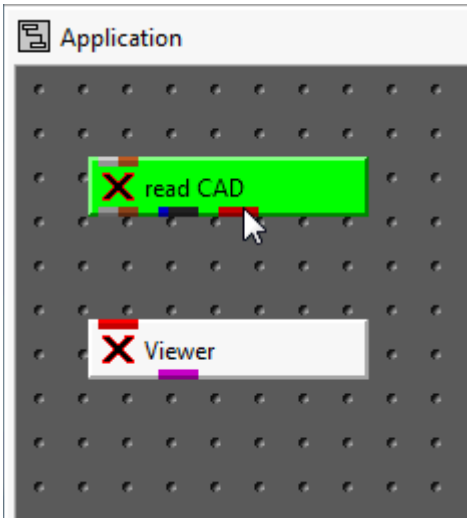
Drag the read_CAD module from the library to the Application area. Dragging is done by clicking and holding down the left mouse button. The Application area is located below the libraries, and is the large dark-gray with dimples area surrounded by a border; the top of the border says *Application*.

Next, go to the second column of modules shown above, called **Display**. Select the top module, labeled **Viewer** and drag it to the Application area directly below read_CAD. It may take a couple seconds for this process to be completed because the Viewer is the most comprehensive (and biggest) of all of the modules. When it completes, there should be another window in the upper left hand corner of your display, called *C Tech Viewer*.

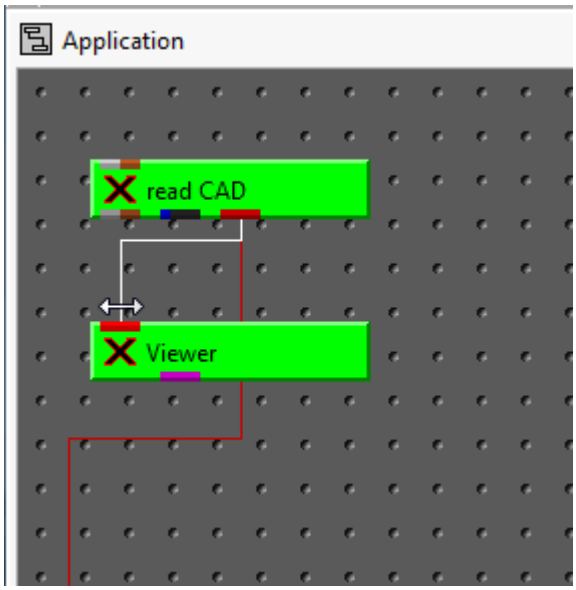
At this point your window should look like the figure below.



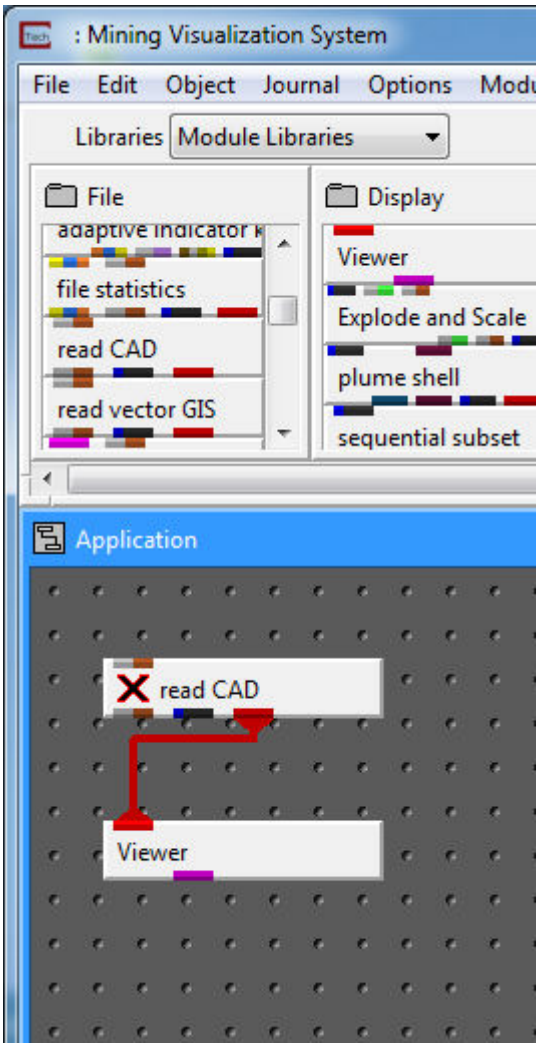
In the Application area, place the mouse over the red output port on the bottom of **read_CAD** (as in the figure above), and press the left mouse button. At this point **read_CAD** should turn green as shown below.



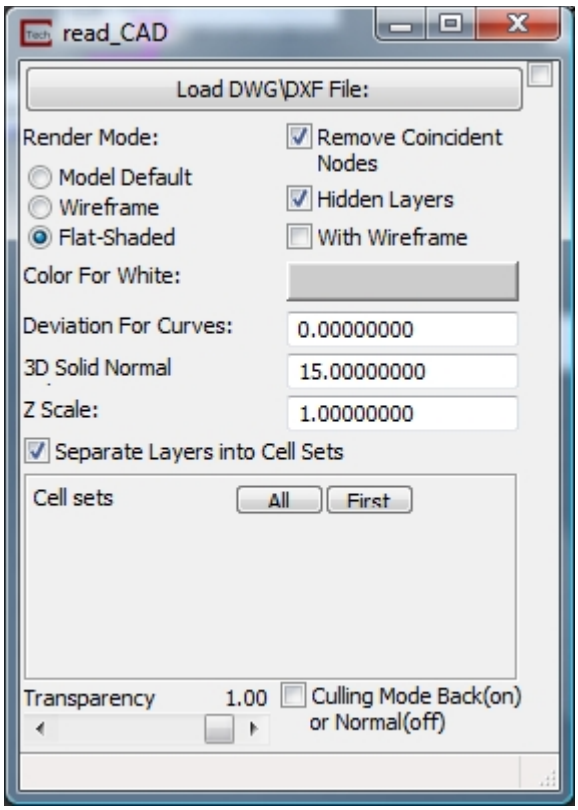
If you move your mouse (while holding the left mouse button down) towards the Viewer, a white line will appear going from that port to the left (red) input port on the **Viewer** (with a red line going as well as to the bottom of the Application area).



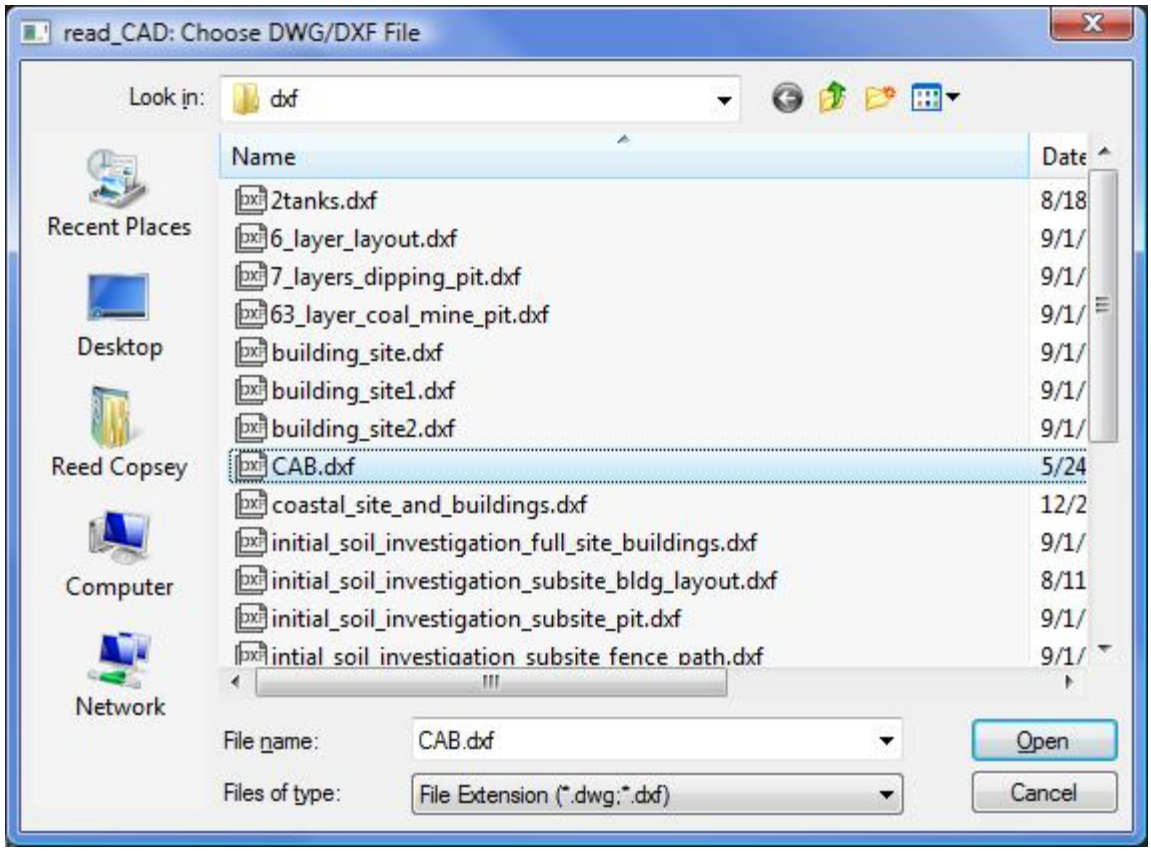
Move the mouse until it is over the left-hand input port on the top of **Viewer**, and release the left mouse button. At this point your window should match the figure below.



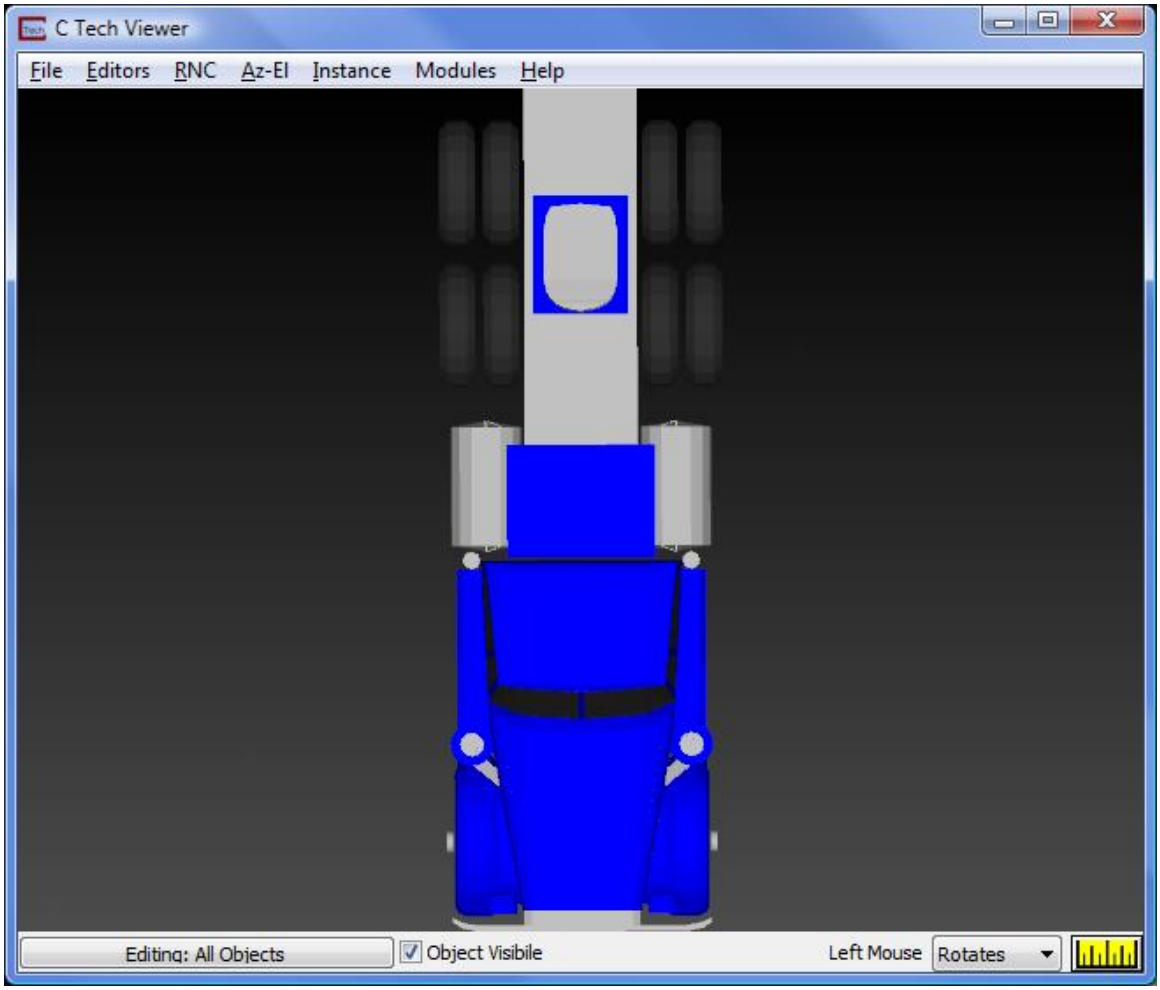
A thick red line will appear, connecting the output port of **read_CAD** to the input port of **Viewer**, as shown in the figure above. This graphical pipeline represents the data flow connection that MVS/EVS will use when you use **read_CAD** to load a file from in this application. To open the read_CAD module's user interface, merely double click on the read_CAD module and it will open (you can also use the mouse to click on the Modules pull-down list in the upper right hand portion of the Main MVS/EVS window). A browse dialogue box will appear.



Click on the **Read DXF File** button and a file browser will open.



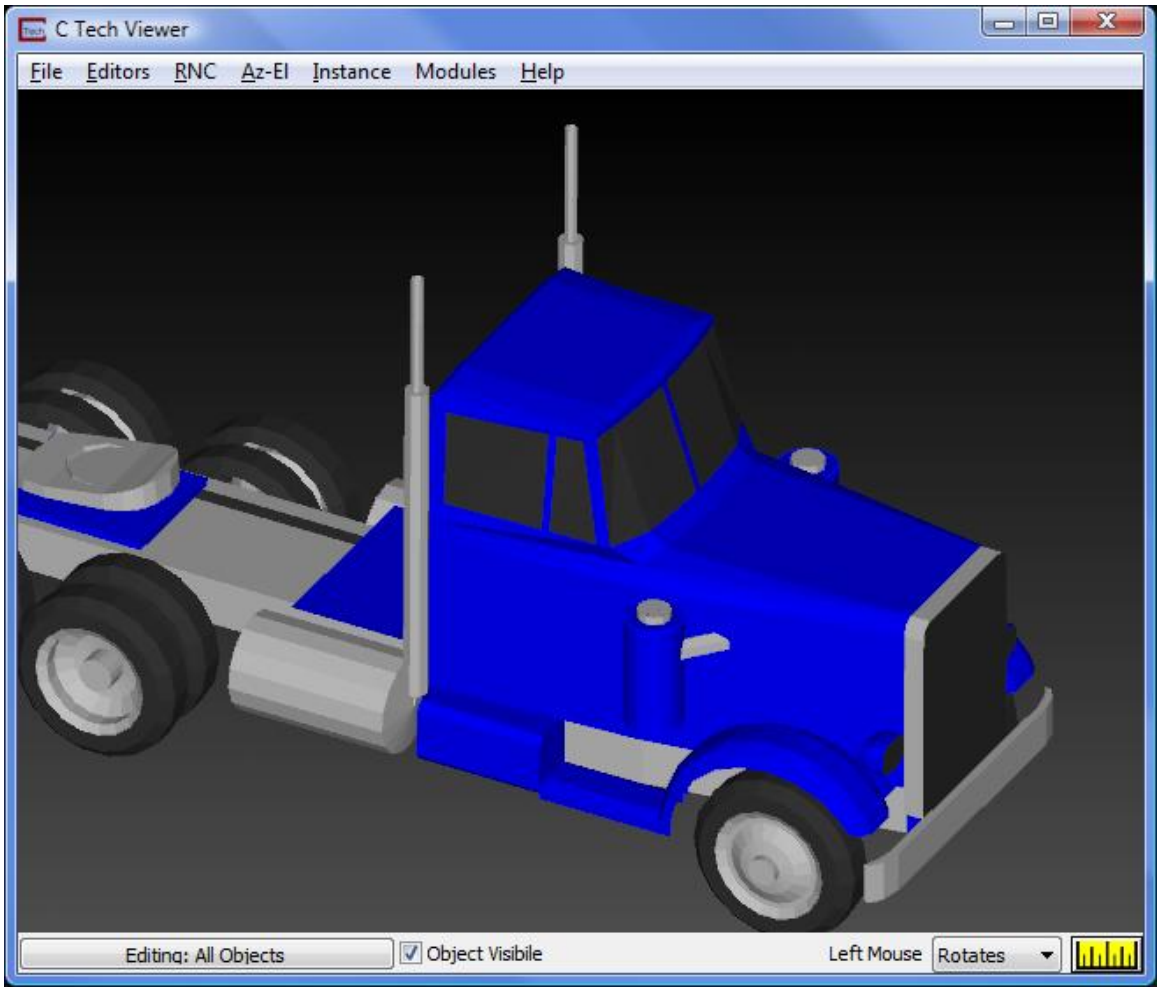
Select the file *CAB.dxf*, and click on **OK**. The Viewer module renders the geometry and then displays it in a three dimensional viewer, as shown below. At this point our viewer is showing us a TOP view of this object.



Manipulating Objects in the Viewer

MVS/EVS automatically normalizes and centers the object in the view. Once the object is centered in the view, it can be rotated by clicking and dragging on any part of the object with the left mouse button (see figure below). The object can be translated in the x-y plane by clicking and dragging it with the right mouse button. Try it now until you are comfortable with rotating, translating, normalizing and centering objects in the viewer.

Zooming of the object in the viewer is accomplished by depressing the shift and the left button (or a middle mouse button or wheel button without shifting) while clicking and dragging the mouse cursor towards the upper right corner of the viewer window (zoom in), or towards the lower left corner of the viewer (zoom-out). Try it now until you are comfortable with the zooming functions.



Workbook 1:covers these fundamentals in more detail and will introduce the process of 2D kriging of analytical (e.g. chemical) datasets.

Learning to Use C Tech's Software

Workbooks

MVS/EVS includes 13 workbooks to help you quickly learn how to use the software and its' powerful modular architecture. Additionally, C Tech offers regular training classes which address advanced topics and offer an accelerated path to visualization expertise.

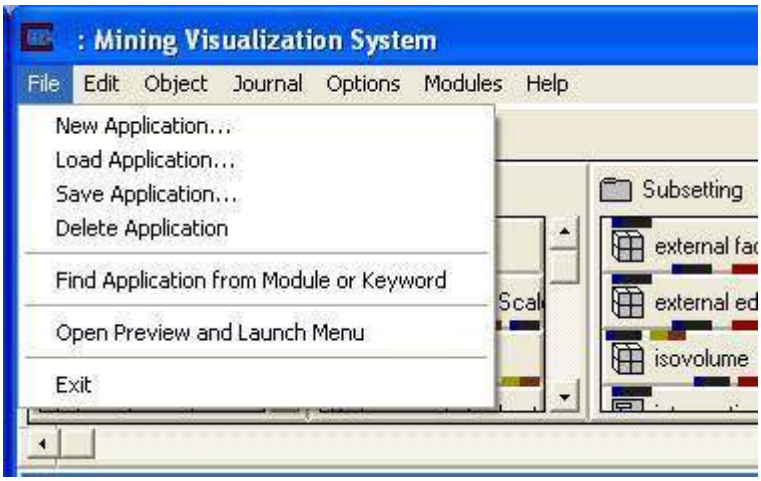
- **Workbook 1 Fundamentals and Two-Dimensional Kriging:**
- **Workbook 2 DrillGuide© Analytically Guided Site Assessment:**
- **Workbook 3 Creating A Geologic hierarchy:**
- **Workbook 4 Three-Dimensional Geologic Modeling:**
- **Workbook 5 Three-Dimensional Kriging:**
- **Workbook 6 Three-Dimensional Fence Diagrams:**
- **Workbook 7 Visualizing Groundwater Modeling Results:**
- **Workbook 8 Animation Using EVS-PRO & MVS:**
- **Workbook 9 Geostatistics in EVS:**

- **Workbook 10 Finite Difference Gridding:**
- **Workbook 11 Advanced Geologic Modeling Concepts:**
- Workbook 12 Controlling Geologic Hierarchy:
- **Visualization Fundamentals**
- **C Tech Main Help**
  - **Visualization Fundamentals**

Quick Start & No Limits

MVS/EVS's modular approach allows users to become productive with MVS/EVS in one or two days, yet provides tools which can be employed in such advanced ways that it is not easily outgrown. C Tech is continually developing new modules, which dramatically enhance the utility of MVS/EVS. Customer suggestions and a creative development team have made and will continue to make MVS/EVS the premier geologic, environmental, and mining analysis and visualization software system.

File menu commands



New Application (File menu)

Use this command to clear your existing application.

Shortcuts

Keys: ALT+F, then N

Load Application (File menu)

Use this command to load an application (network) file supplied with MVS/EVS or previously created in the network editor (Application) area.

Shortcuts

Keys: ALT+F, then L

Load Application dialog box

Files with the .v extension are shown for loading.

List Files of Type

Select the type of file you want to open:

Drives

Select the drive for loading or saving.

Directories

Select the directory which contains the file.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

Save Application (File menu)

Use this command to save an application file created in the network editor (Application) area. To save modules in an application (network), highlight them by sequential selection using the shift left mouse button or by lassoing. Then select File, Save Application and provide a file name. All application files should end in the suffix .v.

Shortcuts

Keys: ALT+F, then S

Delete Application (File menu)

Use this command todelete your currentapplication.

Find Application from Module or Keyword

C Tech's Find Application Toolprovides a quick way tofind sample applications illustrating the use of a module or keyword.

Open Preview and Launch Menu (File menu)

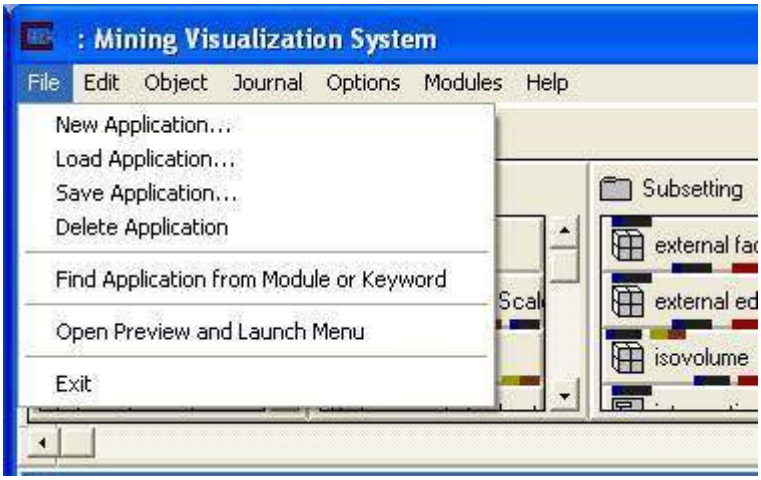
C Tech's Application Preview & Launch Menuprovides a quick way to select and access commonly used menus.

Exit command (File menu)

Use this command to end your MVS/EVS session.

Keys: ALT+F then X

File menu commands



New Application (File menu)

Use this command to clear your existing application.

Shortcuts

Keys: ALT+F, then N

Load Application (File menu)

Use this command to load an application (network) file supplied with MVS/EVS or previously created in the network editor (Application) area.

Shortcuts

Keys: ALT+F, then L

Load Application dialog box

Files with the .v extension are shown for loading.

List Files of Type

Select the type of file you want to open:

Drives

Select the drive for loading or saving.

Directories

Select the directory which contains the file.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

Save Application (File menu)

Use this command to save an application file created in the network editor (Application) area. To save modules in an application (network), highlight them by sequential selection using the shift left mouse button or by lassoing. Then select File, Save Application and provide a file name. All application files should end in the suffix .v.

Shortcuts

Keys: ALT+F, then S

Delete Application (File menu)

Use this command todelete your currentapplication.

Find Application from Module or Keyword

C Tech's Find Application Toolprovides a quick way tofind sample applications illustrating the use of a module or keyword.

Open Preview and Launch Menu (File menu)

C Tech's Application Preview & Launch Menuprovides a quick way to select and access commonly used menus.

Exit command (File menu)

Use this command to end your MVS/EVS session.

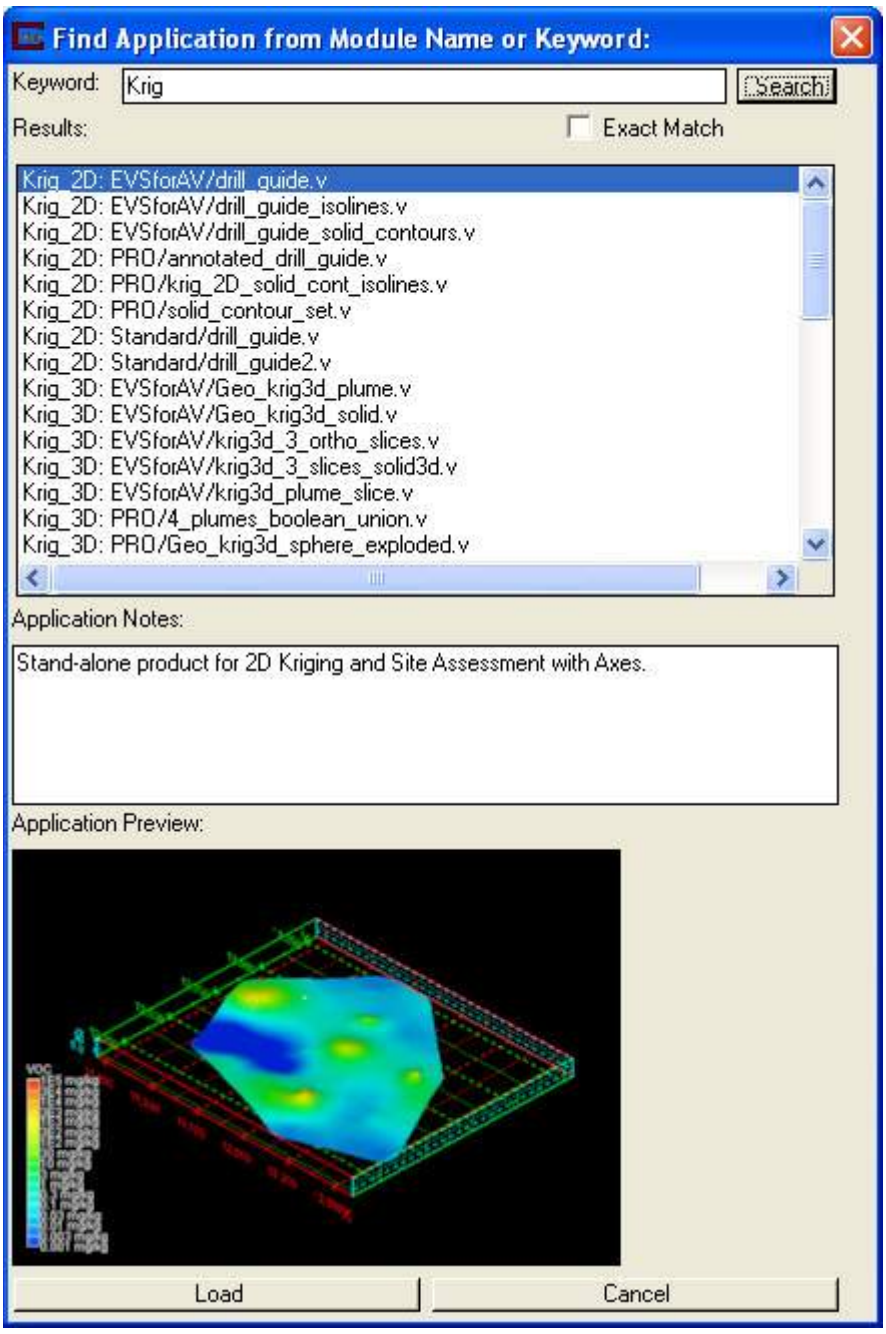
Keys: ALT+F then X

Find Application from Module or Keyword

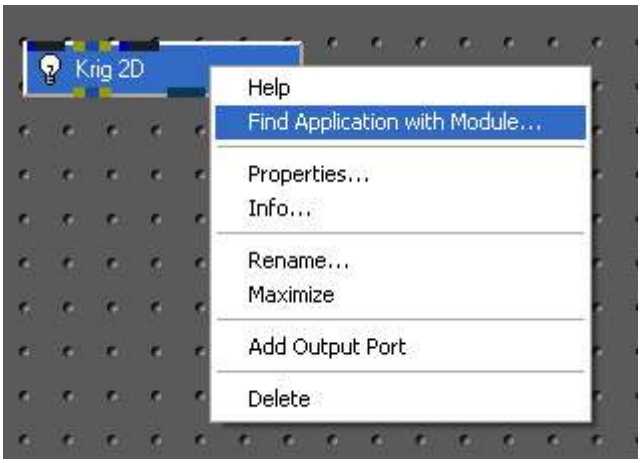
The *Find Application from Module Name or Keyword* tool provides a convenient way to find a sample application demonstrating the use of a module. It can be accessed in two ways.

First, in the Network Editor's File Menu, there is an option to open this window. This will allow you to type in a module name or keyword, or a portion thereof, and find any sample application demonstrating this module.

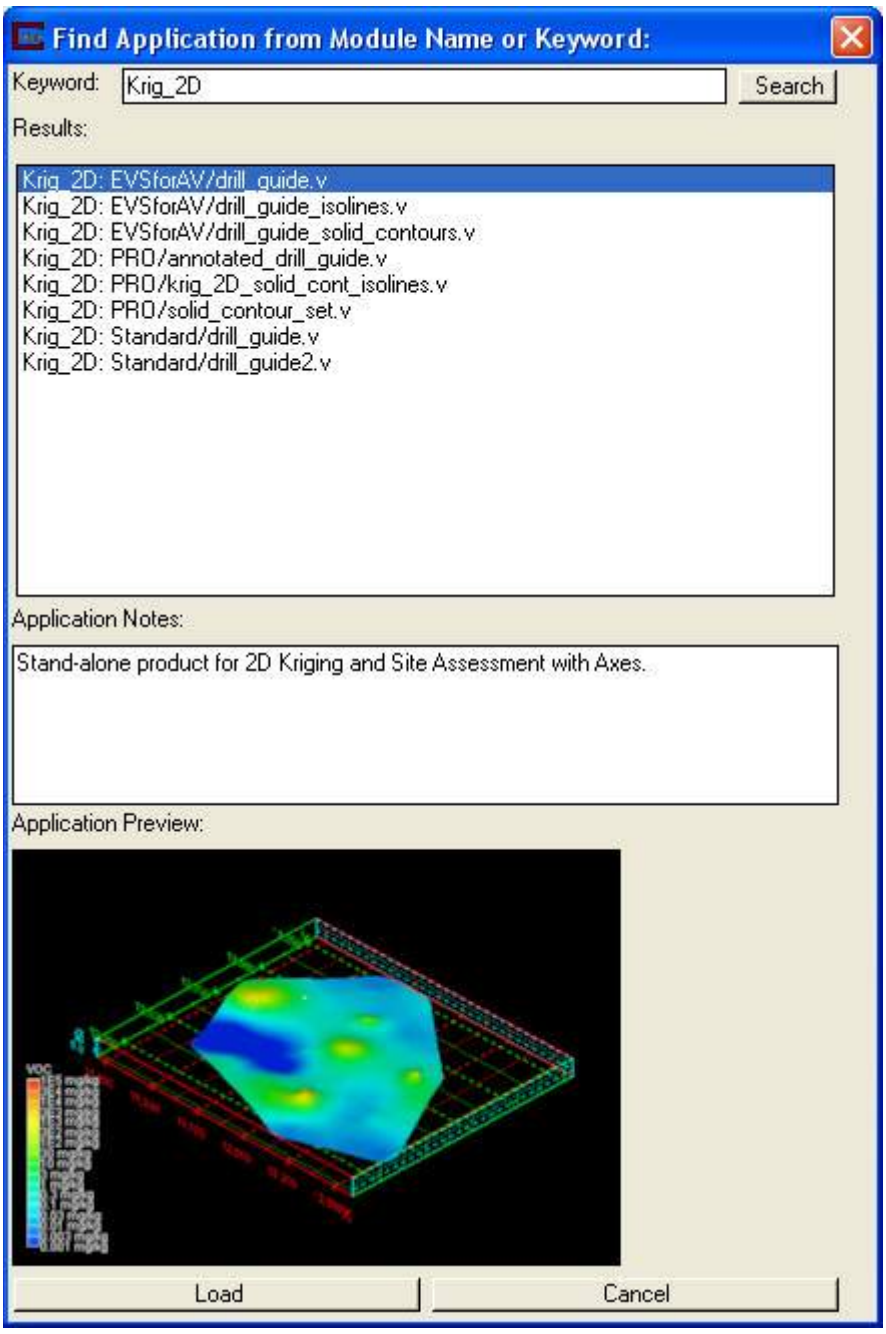
For example, in the image below, by typing "Krig" into the Keyword section, all sample applications featuring Krig_2D, Krig_3D, Krig_3D_Geology, and Krig_Fence are listed, along with their application notes and preview images when available.



The tool can also be accessed by right-clicking on a module, and choosing the "Find Application with Module" option.



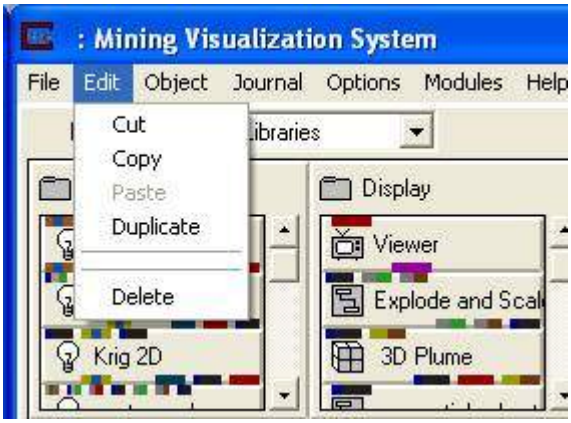
In this case, only applications demonstrating that specific module are listed.



Once an application is found, it can be loaded directly by pressing "Load" at the bottom of the Find Application window.

If the Exact Match toggle is selected the results will list only applications containing the exact Keyword. For example with the toggle off, the keyword Krig_3D will bring up applications containing both the Krig_3D module and the Krig_3D_Geology module. With the toggle on only Krig_3D applications will be displayed.

Edit menu commands



Cut command (Edit menu)

Use this command to remove the currently selected modules from the network editor and put them on the MVS/EVS clipboard. This command is unavailable if there is no data currently selected.

Cutting data to the clipboard replaces the contents previously stored there.

Shortcut

Keys: ALT+E then u

Copy command (Edit menu)

Use this command to copy the currently selected modules from the network editor and put them on the MVS/EVS clipboard. This command is unavailable if there is no data currently selected.

Copying data to the clipboard replaces the contents previously stored there.

Shortcut

Keys: ALT+E then c

Paste command (Edit menu)

Use this command to insert a copy of the clipboard modules into the network editor. This command is unavailable if the clipboard is empty.

Shortcut

Keys: ALT+E then p

Duplicate command (Edit menu)

Use this command to copy (duplicate) the currently selected modules from the network editor and put them into the network editor. This command is unavailable if there is no module(s) currently selected.

Shortcut

Keys: ALT+E then l

Delete command (Edit menu)

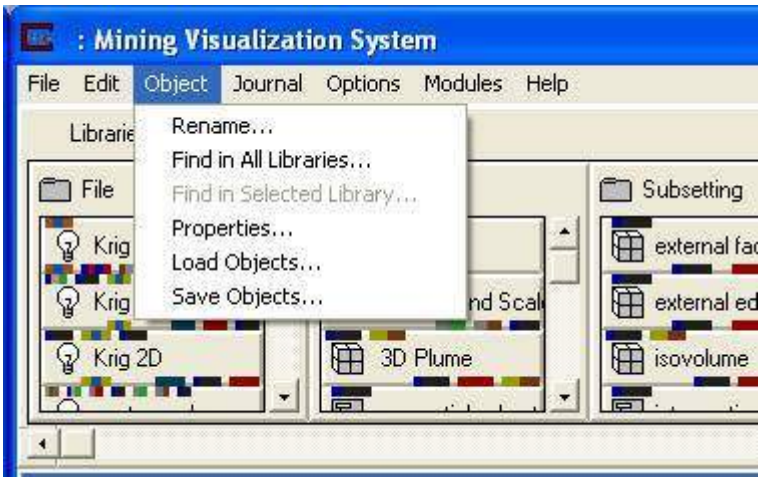
Use this command to delete selected modules...

Shortcuts

Keys: ALT+E then d

Object Menu Commands

The Object menu options allow you to quickly find modules, rename modules currently in use and save portions of networks for later use in MVS/EVS applications.



Rename

Use this command to rename the currently selected module within the network editor. This command is useful for customizing applications with module names specific to a project. For example you might rename a slice module in your application cross-section.

Shortcut

Keys: ALT+O then r.

Find in All Libraries/Find in Selected Library

Use this command to find any module in the MVS/EVS Module Library. The functionality supports the use of standard wildcard search syntax (such as: iso* to find isolines, plume_volume, isosurface, etc). After typing a search pattern (By Name) click Find, then select the desired module from the resulting list in the objects box and click Show. The appropriate module will be immediately displayed in its library position. Finally, click Cancel to close the object finder and you are ready to instance your newly found module.

Properties

This feature is intended for complex debugging operations only. It is not recommended for MVS/EVS customer use.

Load Objects/Save Objects

Similar to Load and Save Applications, these commands are useful for saving 'portions' of a network, which can then be brought up at a later time for use in an MVS/EVS application. To Load an object, you must select the application. Do this by clicking on the border surrounding the application in

the Network Editor (labeled "Application"). It will turn blue when selected. Load Objects will then work correctly.

Journal Menu Commands

The Journal menu options (This feature available only in EVS PRO) allow you to record and play a journal file. Journals record all of your actions such as changing filenames or module parameters.

Journal "Show Playback Controls"

Show Playback Controls modifies the main EVS/MVS window and allows for playing script (journal) files which can control every aspect of EVS. Playback Controls provides the ability to **Play, Pause andStop** journal playback and gives a display of playback status. Journals can change any parameter, modify the view, and cause files to be read or saved. EVS's Animator creates journal files which provide a mechanism to create seamless complex animation sequences. Be sure to keep in mind when you play back a journal, that the same modules must be present in the Modules - pull-down menu.

Journal Record

Journal Record creates a script (journal) file which can control every aspect of EVS. Journals can change any parameter, modify the view, and cause files to be read or saved. To create a journal, choose Journal.Record and provide a file name for the journal file. Until you select Journal.Stop, every action in EVS will be recorded and can be played back at a later time. Be sure to keep in mind when you play back a journal, that the same modules must be present in the Modules - pull-down menu.

Journal Record Stop

Journal Record creates a script (journal) file which is terminated using Journal.Stop.

Options Menu Commands

The Option menu options allow you to control the behaviour of several aspects of MVS and EVS applications.

Backup Application Before Save

Use this command to automatically create a backup of your applications prior to saving. Can help you recover your work should you overwrite an important application or crash during saving.

Show Multiple Modules

Allows you to select and display the user interfaces for multiple modules simultaneously.

Module Flashing

Allows you to see which modules are running be "flashing" changing their color when the module executes. Though on by default. turning it off will speed things up a bit. Especially real time animations like fly-throughs.

Verbose Functions

This feature is intended for complex debugging operations only. It should usually be used by MVS/EVS customers only when directed by Technical Support.

Adjust Volume

Allows you to control the volume of soundtracks in 4DIM files. Only applies to the Playback_4DIM module.

Modules Pull Down (List)

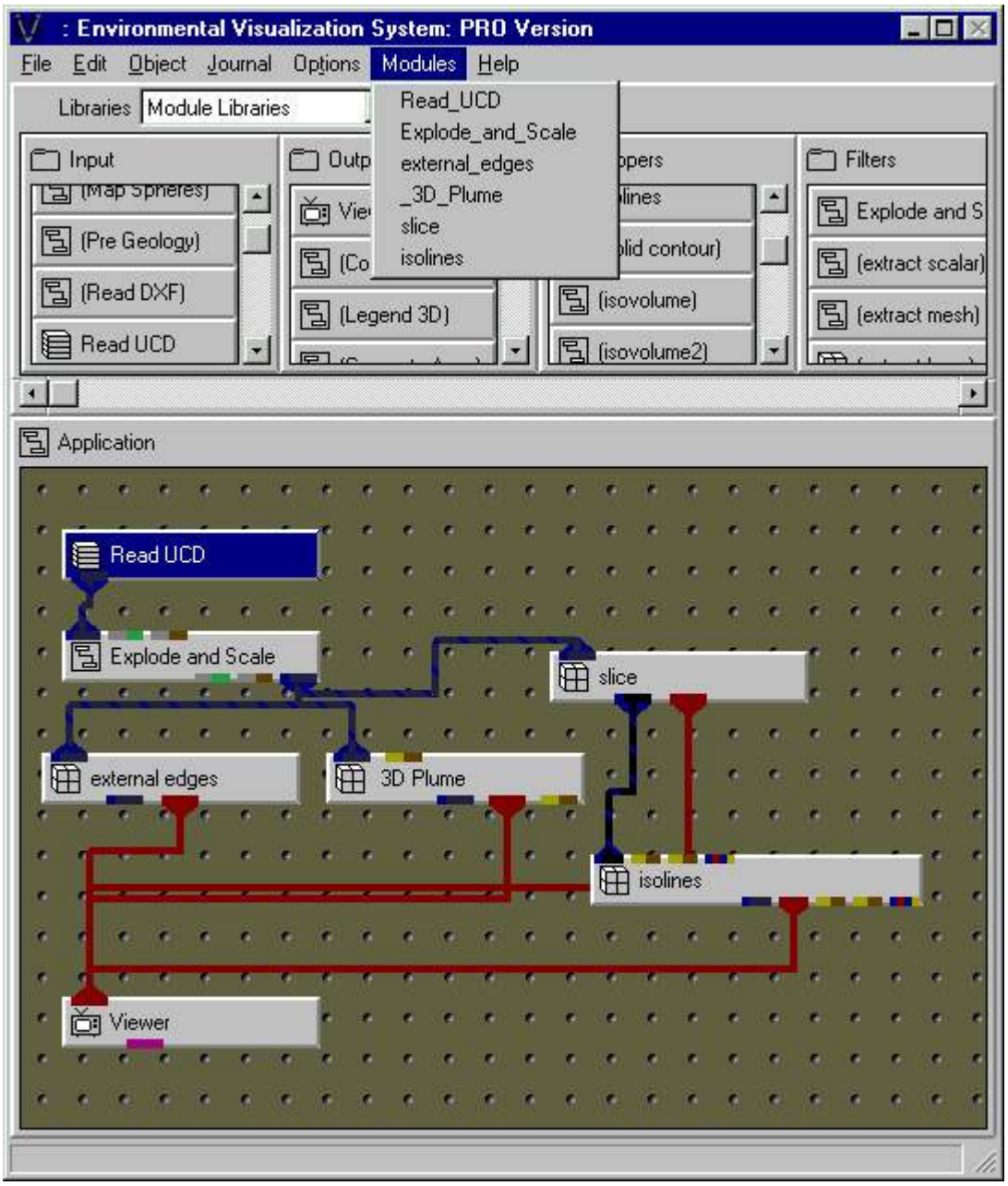
The Modules pull-down list provides a list of all modules in the application and allows you to access the module's user interface. Selecting a module here will make its user interface window visible. Note that the normal behavior in EVS is to allow only one module user interface to be visible at a time.

To view more than one module at a time either:

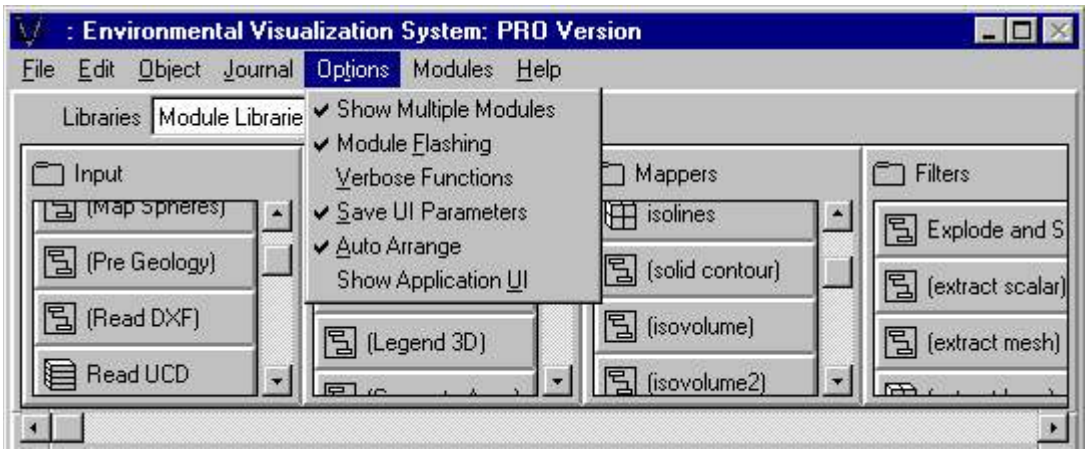
a) Turn on *Show Multiple Modules* in the Options Pull Down list or

b) Pin the module open (the little check box that forces the module's interface to stay visible)

Note that each module in the list exists in the EVS Application (Network Editor) portion of the main window.



However there is an option that allows selecting Multiple Modules. It is shown below selected.



Also, virtually every module's user interface (window) has a toggle below the "close" – X in the upper right hand corner. This toggle keeps those windows (or sub-windows) open.

Another possible location of the Modules pull-down is on the Viewer.

Help menu commands

The Help menu offers the following commands, which provide you assistance with this application:

Help...Contents

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using EVS and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

Help...On Selected Module

By selecting a module and then this menu option, you can obtain detailed help for that module. You can also click with the right mouse button and select **Help**. This is called Selecting a pop-up command

Network-Editor operations

- **Mouse Operations in the Network Editor**
- **Selecting a pull-down command**
- **Selecting a pop-up command**
- **Selecting modules**
- **Connecting and Disconnecting Modules**

Network-Editor operations

Mouse Operations in the Network Editor

MOUSE OPERATION BASICS

Mouse buttons

A mouse has at least two buttons. The primary buttons are left and right.

Mouse pointer

The mouse controls an arrow-shaped cursor on the screen. The cursor is called the mouse pointer. As you move the mouse, the mouse pointer moves accordingly.

Pointing

To point to an item, move the mouse pointer to the item. The mouse pointer's location is the tip of the pointer's arrow.

Selecting

To select an item:

1. Point to the item.
2. Click the indicated mouse button once.

Double-clicking

To double-click on an item:

1. Point to the item.
2. Click the indicated mouse button twice fairly rapidly.

Dragging

Dragging an item means moving it from one location to another.

To drag an item:

1. Point to the item.
2. Press and hold down the indicated mouse button.
3. Keeping the button held down, move the mouse pointer. Some operation takes place as you move the pointer.
4. When you have completed the operation, release the mouse button.

ADVANCED OPERATIONS

Scale (Zoom) in Applications

a. Holding down **Ctrl, Right Click** in the gray-bump region in the Application window (not on a module or connection).

b. Move the mouse pointer downward or to the left to zoom out or opposite to zoom in.

c. OR..**Right Click** in the gray-bump region and select "Zoom to Fit" from the pop up window.

d. **Right Click** in the gray-bump region and select "Reset Scaling" to return to the original zoom and tranlation.

Move (Translate or Pan) in Applications

a. Holding down **Shift, Right Click** in the gray-bump region in the Application window (not on a module or connection).

b. Move the mouse pointer to translate the modules in your application.

c. **Right Click** in the gray-bump region and select "Reset Scaling" to return to the original zoom and tranlation.

Selecting a pop-up command

Definition

A *pop-up command* instructs EVS to perform some operation. Pop-up commands appear when you press the right mouse button.

Where they apply

You can select a pop-up command for any of the following:

* A module
* A module's port
* An opened window
* A module connection line

Selecting a pop-up command

To select a pop-up command:

1. For a module, move the mouse pointer to the module. If the module is open, point to the module's title bar or window frame.

For a port, move the mouse pointer to the port.

For an opened window, move the mouse pointer to an unoccupied area of the window.

For a connection line, move the mouse pointer to the line.

2. Press down the right mouse button. A pop-up command list appears.

3. Keeping the mouse button pressed down, move the mouse pointer to the desired pull-down command.

4. Release the mouse button. The Network Editor executes the command.

You do not have to select a command from a popped-up command list. To remove the command list without selecting anything, move the pointer completely out of the command list, then release the mouse button.

Selecting Modules

Background

Most pull-down commands, as well as certain other Network Editor activities, operate on one or more *selected* modules. For example, the Edit->Delete pull-down command deletes the modules that you have selected.

Selecting a single module

To select a single module, select it with the left mouse button. The Network Editor selects the module and deselects all previously selected modules. If the module is open, select its title bar or frame. Do not select the module's ports.

Appearance

A selected module's color is blue.

Lasso

You can select several modules at once by *lassoing* the modules. Lassoing means using the mouse to outline a rectangular region, or lasso. The Network Editor selects the modules that are completely or partially inside the region.

To select one or more modules by lassoing:

1. Point to one corner of the rectangular region you intend to outline. The pointer must be over an unoccupied portion of the window.

2. Drag the mouse pointer with the left mouse button to the diagonally opposite corner of the region. As you do, the Network Editor displays the rectangular region's outline.

3. Release the mouse button.

The Network Editor selects the modules that are completely or partially inside the rectangular region. It removes the outline and deselects all previously selected modules.

Shift key

When you select one or more modules, the Network Editor by default immediately deselects the previously selected modules.

You can alter this behavior by holding down the Shift key when you select a module. Here's what happens:

***** If the module is currently not selected, the Network Editor selects it, *adding* it to the selection list. The selection list now consists of that module and the previously selected modules.

***** If the module is currently selected, the Network Editor deselects it, *removing* it from the selection list.

You can use the Shift key with a lasso operation, too.

Modules in the selection list must have the same parent module. If you attempt to add a module to the selection list, but the module's parent is

different, the Network Editor selects the module and deselects the previously selected modules.

Deselecting all modules

You can deselect all modules by selecting an unoccupied area in an opened window.

Connecting & Disconnecting Modules

To Connect Modules:

1. Make sure the modules have ports at the same level in the module hierarchy, and that both ports are visible.

2. Point to one of the two ports you intend to connect.

3. Drag the mouse pointer with the left mouse button toward the other port. When you press the left mouse button, the Network Editor shows all of the allowable connections. As you move the mouse pointer, the Network Editor guesses which connection you want to make and highlights it.

4. When the Network Editor has highlighted the correct connection, you can release the mouse button.

Canceling a connection in-progress

While dragging the mouse pointer, you may decide that you do not want to make a connection. To cancel a connection operation in progress, drag the mouse pointer back to the port so that none of the possible connections is highlighted, then release the mouse button.

To Disconnect Modules:

If we make an incorrect connection, we can remove the connection. This is called disconnecting the modules.

We can disconnect modules in two ways:

1) One way is to repeat the connection. That is, we point to one of the connected ports, then, holding down the left mouse button, drag the pointer to the other port. We then release the mouse button.

2) Another way is with the Delete Connection pop-up command. With the mouse, we point to the connection line, but not on a port, then hold down the right mouse button. A pop-up command list appears. Still holding down the mouse button, we move the pointer to the Delete Connection command, then release the mouse button.

Practice disconnecting and reconnecting the modules using the techniques described above. When done, be sure the modules are reconnected.

(The pop-up command list for a connection line includes a command called Insert Link. If you inadvertently select this, a link module gets created. You can delete the module by pointing to it, then selecting the Delete pop-up command.)

Workbooks

These workbooks are designed to help guide you from a novice to an intermediate user at your own pace.

We strongly recommend that you do the first 5 workbooks in order. After that, you'll have the fundamentals to allow you to investigate the others as your project needs dictate.

Workbooks

These workbooks are designed to help guide you from a novice to an intermediate user at your own pace.

We strongly recommend that you do the first 5 workbooks in order. After that, you'll have the fundamentals to allow you to investigate the others as your project needs dictate.

Workbook 1: Fundamentals and Two-Dimensional Kriging

- **Fundamentals and 2D Kriging**
- **Start EVS OR MVS**
- **A first look at the Network Editor**
- **Instance Krig_2D**
- **Instance Viewer**
- **Connect the modules**
- **Run the network**
- **Instance plume_area**
- **Experiment with plume_area**
- **Add external_edges**
- **Azimuth and Elevation Panel**
- **Advanced Azimuth and Elevation Options**
- **Mouse Transformations**
- **Manipulate Objects**
- **Explore the Viewer**
- **Save the network**
- **Exit EVS OR MVS**

Visualization Fundamentals
C Tech Main Help

Fundamentals and Two-Dimensional Kriging

In this Workbook, we use the Network Editor to build a network to perform data analysis and visualization. The application:

* Reads an EVS analyte (e.g. chemistry) data file.
* Performs data estimation (Kriging) of the sparse measured analyte (e.g. chemistry) data.
* Provides coordinates within the domain of the measured data where the uncertainty is highest. This location is the optimal position for the next borehole.
* Uses plume_area to subset the data and select data components for coloring
* Uses external_edges module to provide a visualization of the grid.
* Renders the resulting model in the Viewer.

Start EVS OR MVS

Start EVS OR MVS:

Double click on the EVS Pro OR MVS icon in the C Tech program group

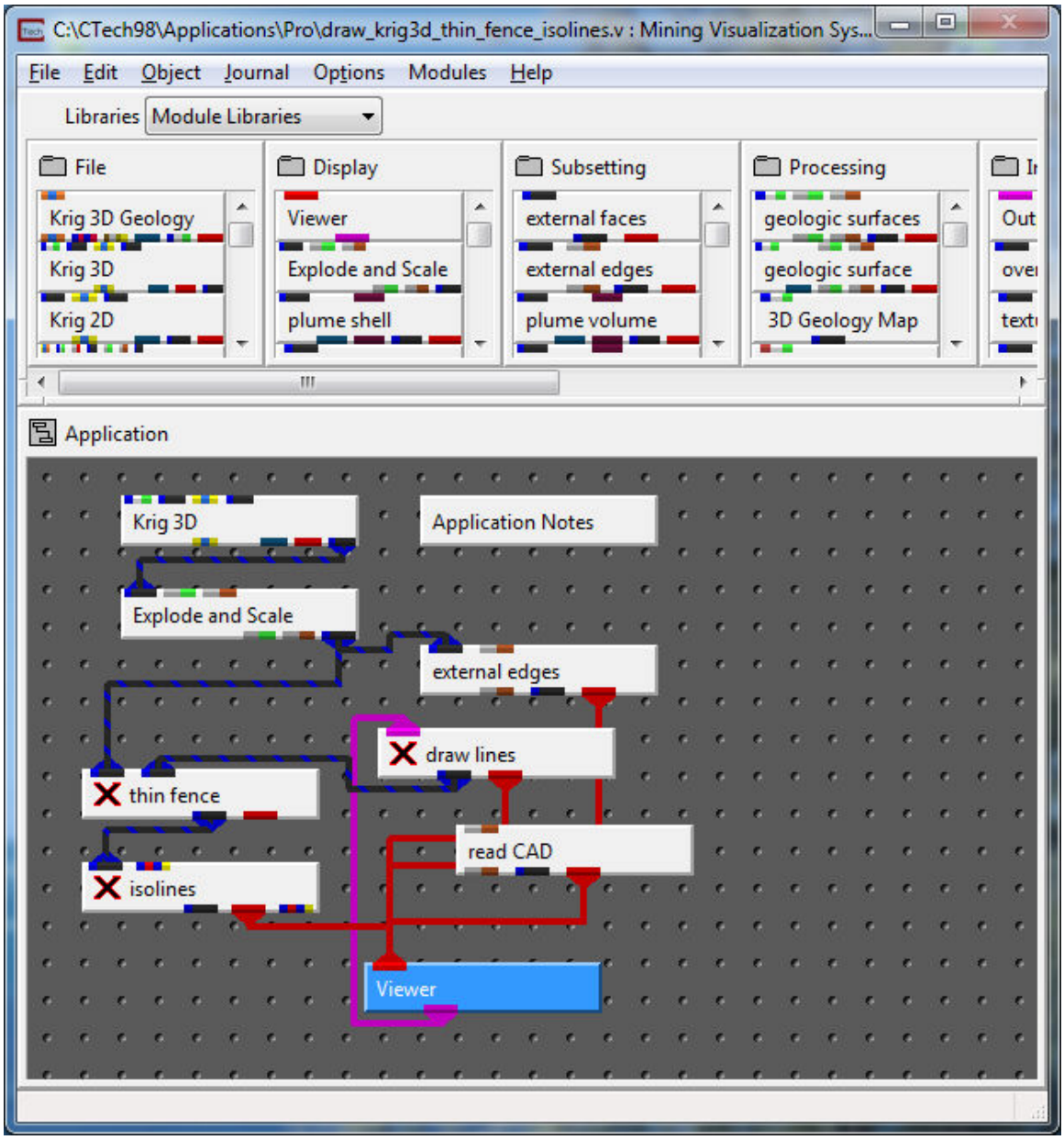
Several things happen:

* The Network Editor appears, occupying much of the right side of your desktop.
* The Status Window appears in a window.

A first look at the Network Editor

Here is the Network Editor. This graphic shows a more complex network than the one we will produce in this Workbook, but it is simple compared with many.

Note that in the application below, the modules with the ✗ have not yet run. In this application, these module are all dependent on the draw_lines module which allows you to draw the path which will be used to create fence diagrams.



Instance Krig_2D

Krig_2D is a module that reads an EVS .apdv (analyte (e.g. chemistry)) data file. It uses Kriging, a geostatistical method for estimating the concentrations at points on a "regular" grid.

Notice that we spell Krig_2D with an underscore, even though it appears as Krig 2D in the Network Editor. Krig_2D's real spelling is with an underscore. The Network Editor replaces underscores with spaces.

Instance Krig_2D

a. With the mouse, point to Krig_2D in File library.

b. Press and hold down the left mouse button. This selects the module, turning it blue.

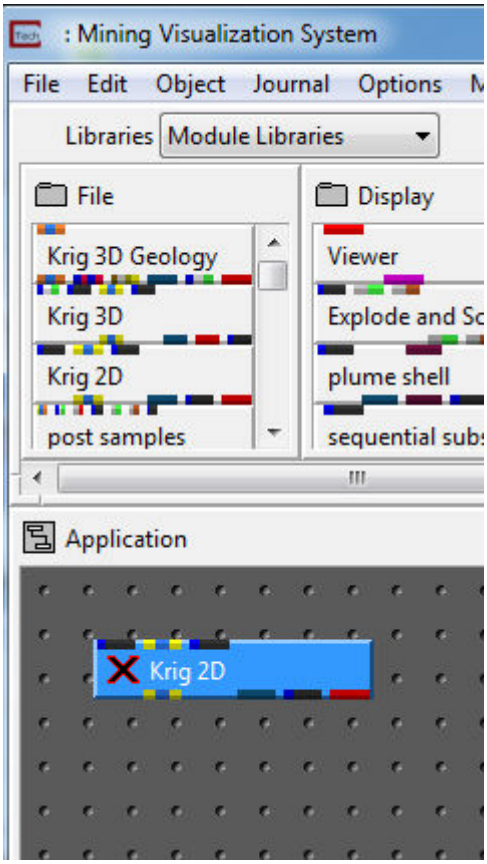
c. Keeping the button held down, move the mouse pointer into the the Application window workspace. This action is called *dragging*.

d. When we have dragged Krig_2D into the Application window , two things happen: an outline of Krig_2D appears; and the Application window 's title bar and frame turn pink, indicating that the Application window is where the module will be placed if we release the mouse button.

e. If the module is not dragging, release the mouse button and try again.

f. Release the mouse button.

The *Application* window should now look something like this:



Note that Krig_2D is preceded by an X because the module has not yet run.

Repositioning Krig_2D

We can easily reposition modules in Application window .

To reposition Krig_2D in The Application window :

a. With the mouse, point to Krig_2D in The Application window . Do not point to the module's ports (the colored rectangular strips at the top and bottom of Krig_2D).

b. Holding down the left mouse button, drag Krig_2D to a new position in the Application window , then release the mouse button.

Real time execution

EVS or MVS executes some aspects of our application *as we build it*. This helps us prototype our application and verify the work we have done. Krig_2D includes a user interface for specifying a .apdv (analyte (e.g. chemistry)) filename and provides many user adjustable parameters. When we instance Krig_2D into the Application window, one thing that happens is that Krig_2D's user interface becomes selectable in the Module menu, located at the top of the Network Editor.

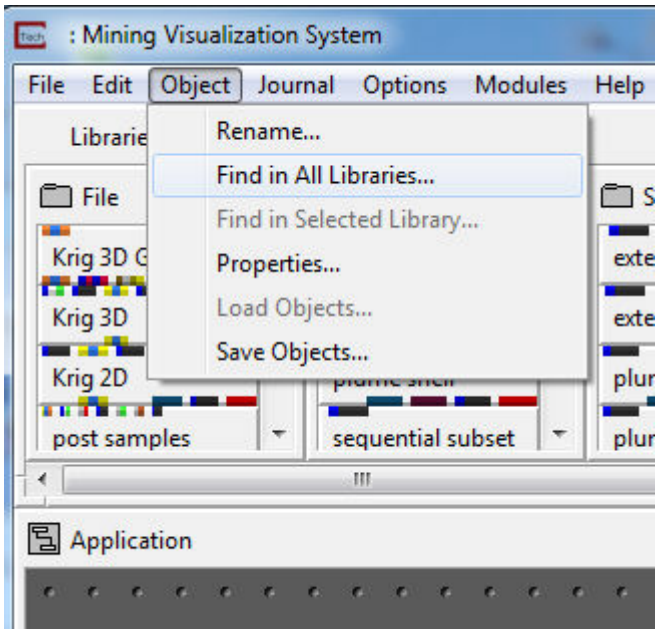
Select the Krig_2D pull-down command from the Modules pull-down menu in the Network Editor window.

Krig_2D's main user interface appears.

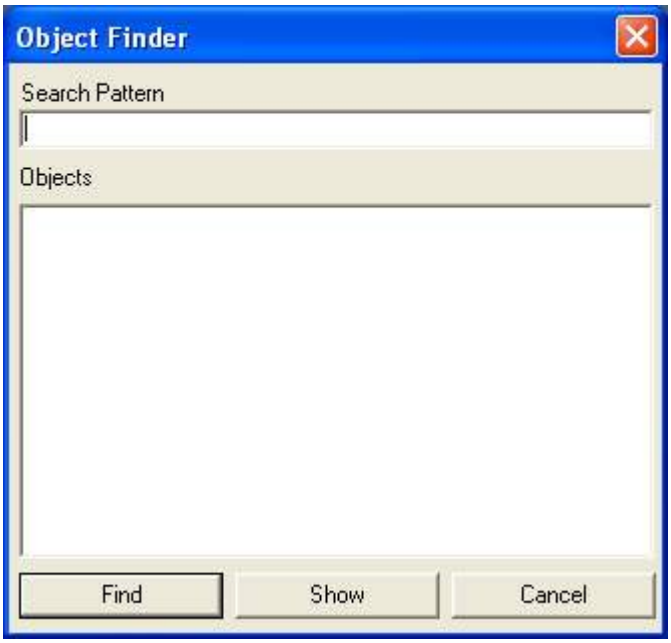
We won't select a file or modify any parameters... yet.

Finding modules using*ObjectFinder*

EVS OR MVS has a tool called an object finder that helps you to find any module in the libraries.

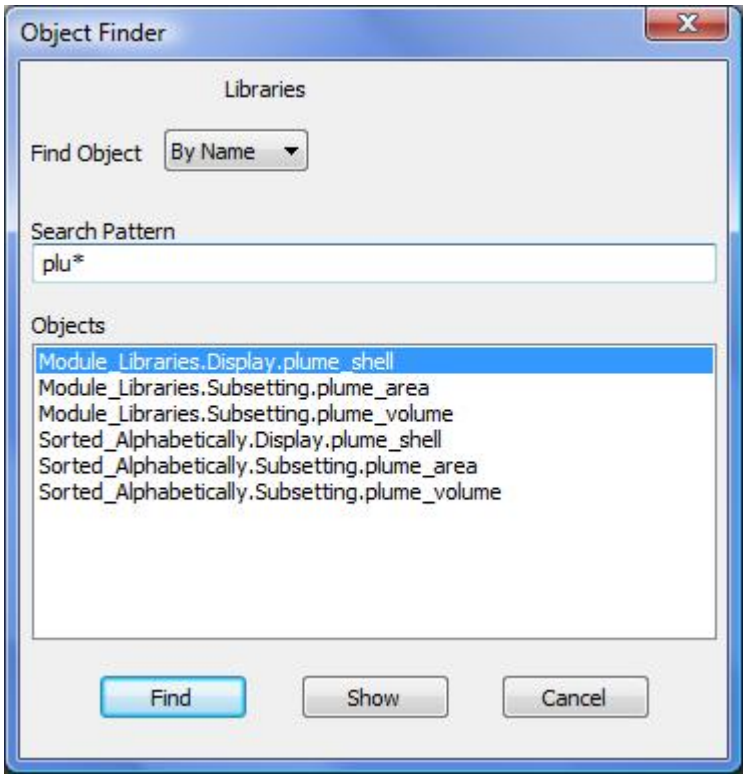


click on the Find in All Libraries in the Object button as seen above. Then, Object finder will come out.



Now let's search for the *plume_area* module. We'll be using it a bit later in this workbook. You don't need to remember the module name exactly. You can type letters that are in the name of the module and type * at the end.

For example, if you type ***plu**** and hit the **Find** button, you'll see all the modules that begin with "plu". If you wanted to find them by searching for "lum" you would need to type "*lum*" This search is not case sensitive.



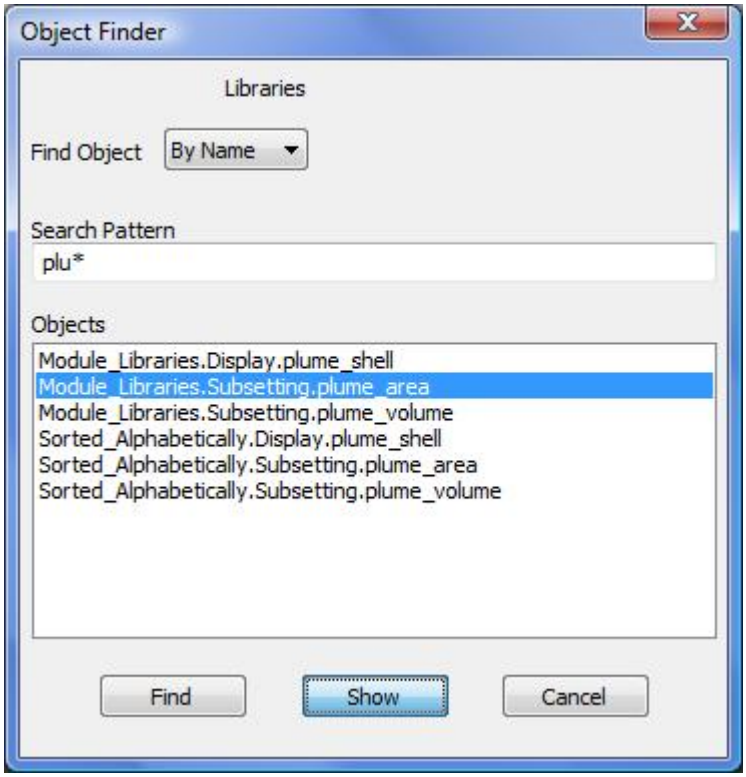
Notice that there are **many** modules with "plu" in the name. Although plume_area only shows up in two libraries, depending on the module name you type, you may notice that there are three different libraries. They are:

a. Deprecated_Modules
b. Module_Libraries and
c. Sorted_Alphabetically

Generally you should NEVER use deprecated modules. These are kept in this library for backwards compatibility, but these modules have been superceded by newer modules and will eventually be dropped.

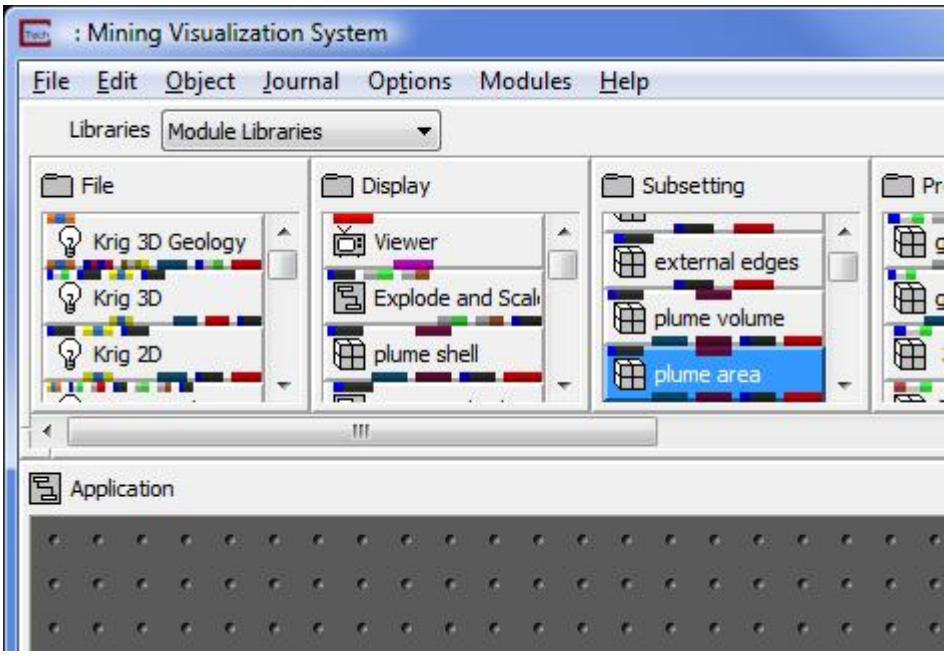
The other two libraries (b & c) are identical in content. These include all of the modules for the version you are running (e.g. MVS, EVS-PRO, EVS-Standard), but they are listed in the sublibraries (e.g. File, Display, etc.) in a different order. Module_Libraries are listed (sorted) by frequency of use and is the recommended (and default) library. Sorted_Alphabetically are sorted alphabetically with the exception that modules with capitilized names are listed first.

Select the second item in the list (Module_Libraries.Subsetting.plume_area) and hit the **Show** button.



This will cause plume_area to be highlighted.

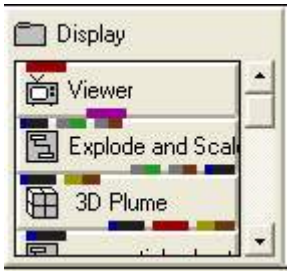
Once you click on Show you can close the Object Finder window. You should see plume_area highlighted as shown below:



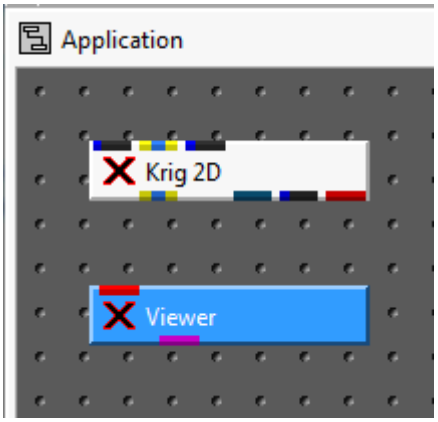
In this particular case, plume_area was the fourth module down in the Subsetting sub-library which required it to scroll down to display the module.

Instance Viewer

The Viewer is the most feature rich module in the C Tech Module Libraries. (It also takes the longest time to load) It includes a viewing window, editors, and mouse interactors. Viewer is located at the top of the Display sublibrary.

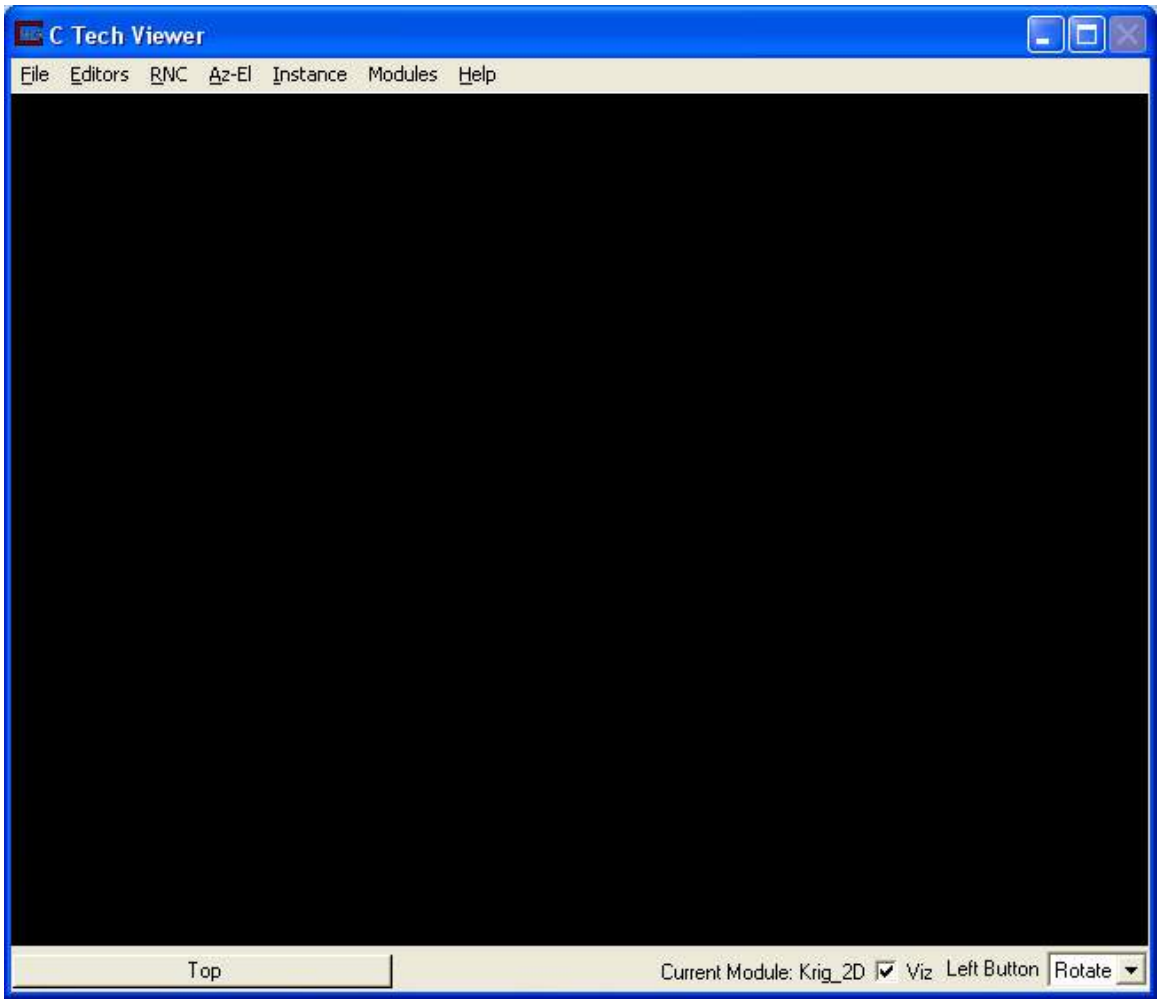


Instance Viewer and position it as shown in the diagram below.



Viewer is one of the most complex modules in EVS OR MVS. When it is instanced, a rendering window, appears on the screen in the upper left-hand corner.

Note that Viewer is preceded by an X because no modules are yet connected to it.



Connect the modules

Connections

We now connect these two modules. Connections determine how data flows or is shared among modules, and affects the modules' order of execution.

(Note: the order in which we instance and connect modules is, with the exception of certain array connections, unimportant. We could have instanced and connected these modules in any order.)

To connect modules, we link an output port of one of the modules to an input port of the other. For example, the image below shows what Krig_2D and Viewer look like connected. The connection means that Viewer gets its data to display by pointing to the data created by Krig_2D.

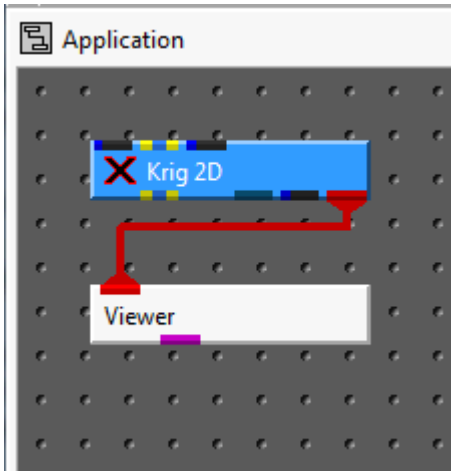
Connect Viewer to Krig_2D

a. With the mouse, point to Viewers only input port (red).

b. Hold down the left mouse button. Thin lines the same color as the port will appear, indicating all of the possible connections. EVS OR MVS allows us to connect only compatible modules. At this time, no connection is chosen.

c. Keeping the mouse button depressed, drag the pointer to Krig_2D's output port. As we move the pointer, the Network Editor highlights in white the connection it will make if we release the mouse button.

d. When the Network Editor highlights the connection to Krig_2D's output port, release the mouse button.



We can connect the modules in either direction. We could have used the same technique to connect the modules in reverse order; i.e., by dragging the mouse pointer from Krig_2D's output port to Viewer's input port. The direction a connection is made makes no difference.

Removing a connection

If we make an incorrect connection, we can remove the connection. This is called disconnecting the modules.

We can disconnect modules in two ways:

* One way is to repeat the connection. That is, we point to one of the connected ports, then, holding down the left mouse button, drag the pointer to the other port. We then release the mouse button.

* Another way is with the Delete Connection pop-up command. With the mouse, we point to the connection line, but not on a port, then hold down the right mouse button. A pop-up command list appears. Still holding down the mouse button, we move the pointer to the Delete Connection command, then release the mouse button.

Practice disconnecting and reconnecting the modules using the techniques described above. When done, be sure the modules are reconnected.

(The pop-up command list for a connection line includes a command called Insert Link. If you inadvertently select this, a link module gets created. You can delete the module by pointing to it, then selecting the Delete pop-up command.)

Canceling a connection in progress

Let's say we point to a port, press down the left mouse button, and start dragging the pointer towards another port. We then realize that we started on the wrong port, so we want to cancel the connection operation we started. But if we release the mouse button, EVS OR MVS will create the currently highlighted connection.

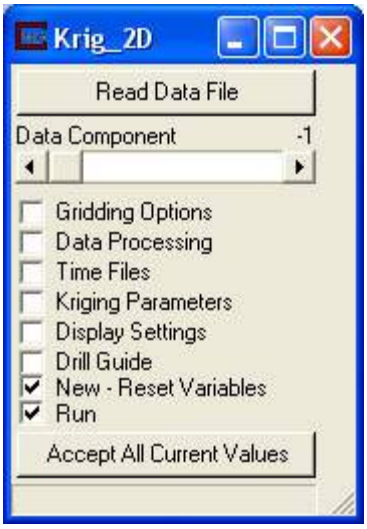
We can cancel a connection operation in progress by returning the pointer to the port we started from, so that no possible connections are highlighted, then releasing the mouse button.

Run the Network

Let's execute the analysis module, Krig_2D, in order to produce a model based on the data file we have selected

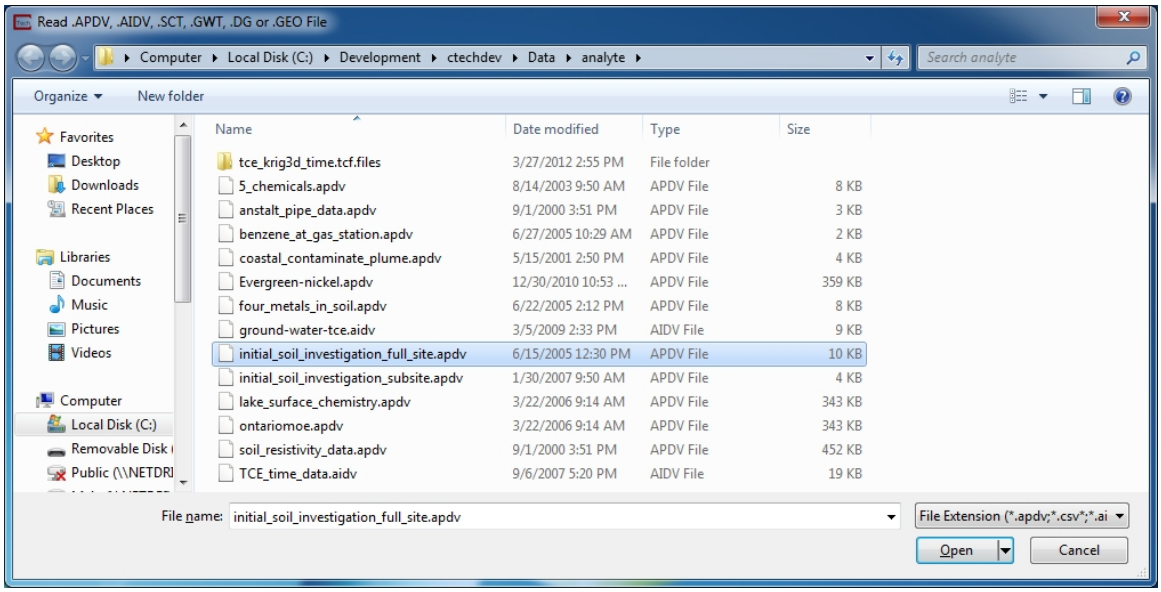
Read a file

a. Double click on Krig_2D in the Application area to open its user interface (shown below). Alternatively, select the Krig_2D pull-down command from the Modules - pull-down menu. The Krig_2D Main menu window appears.



b. Select the **Read Data File** button.

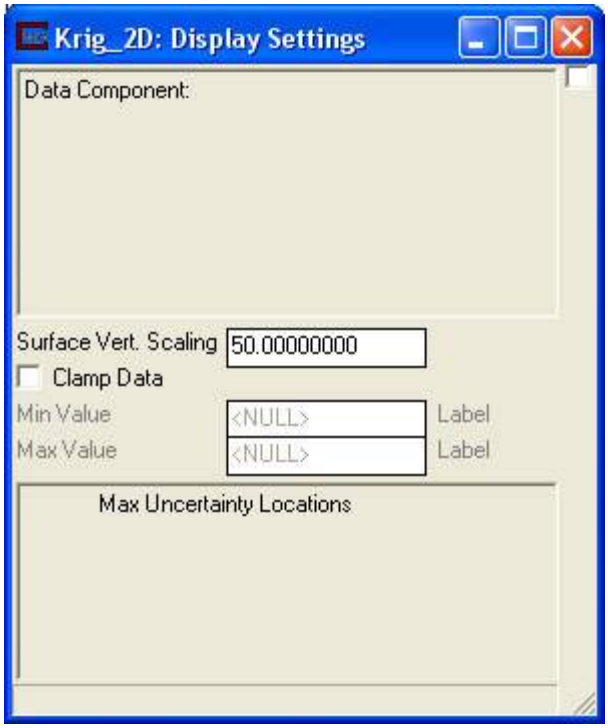
c. Select initial_soil_investigation_full_site.apdv from the Files list box.



d. Select the **Open** button.

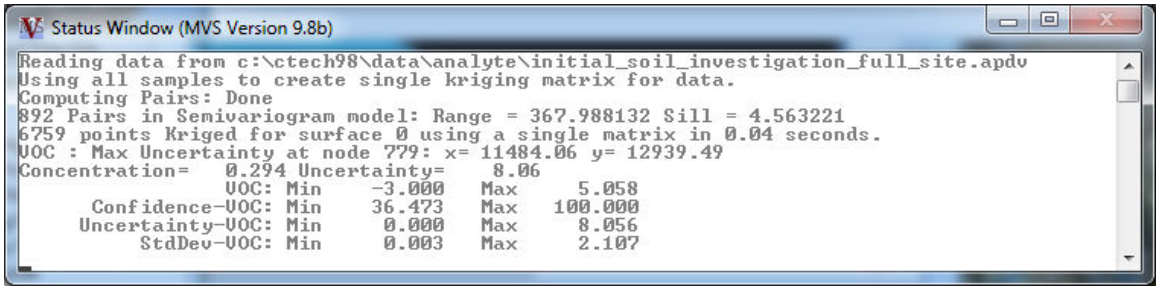
e. Before we run, let's modify one of the parameters. Open up the Krig_2D Display Settings subwindow by clicking on the "Display Settings" toggle.

f. Set the "Surface Vert. Scaling" to 50.0. Before going on to the next step, close the Krig_2D Display Settings subwindow by clicking on the toggle again. After the module runs, this subwindow will be updated and has many powerful options (which we will not address in this workbook).

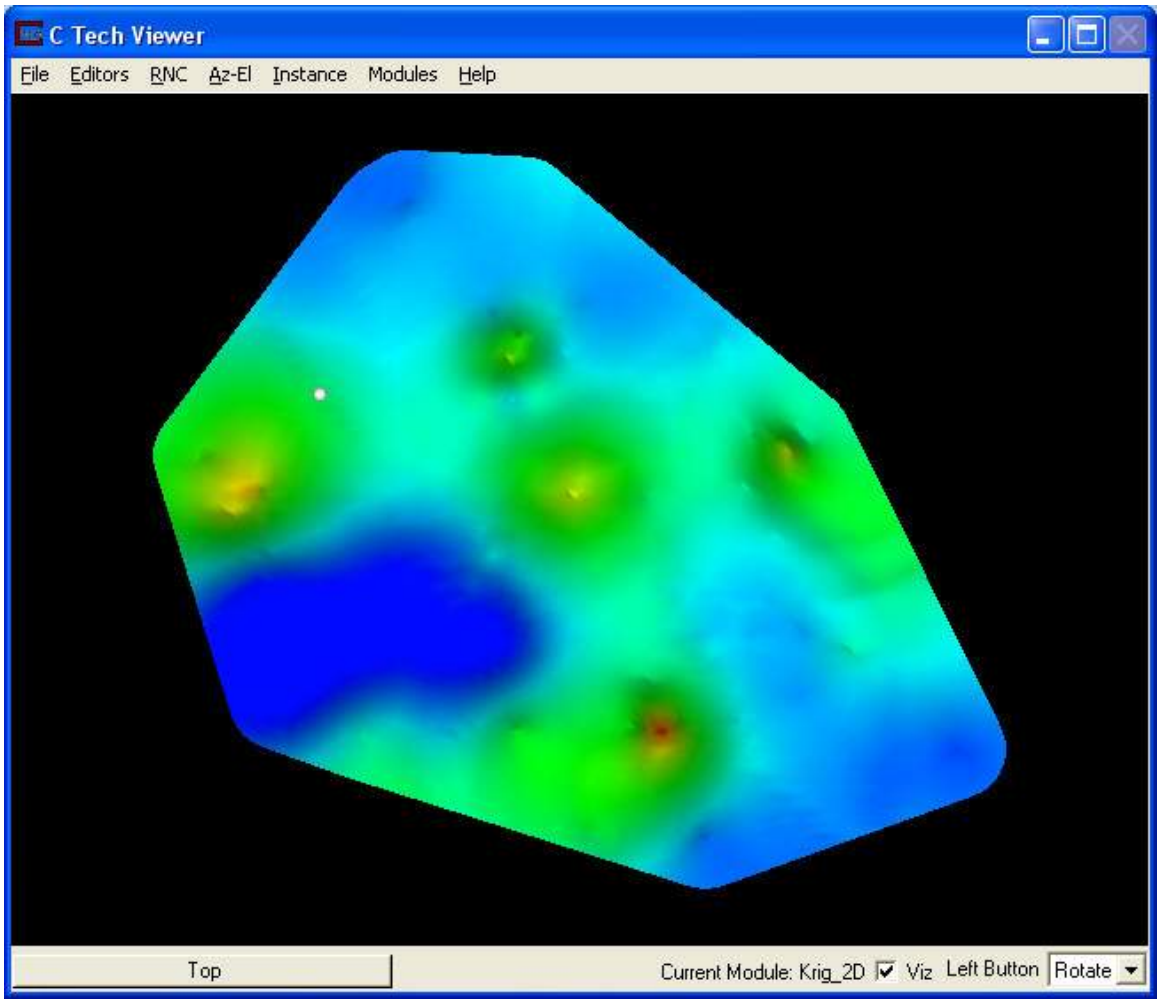


g. Select the Accept All Current Values button in the main Krig_2D window.

Krig_2D reads the analyte (e.g. chemistry) data and begins the kriging process. In a very short time, it calculates the estimated concentrations for the grid we selected. While it runs, Krig_2D prints messages to the console such as percentage completion. When it is done, the console message should look like this:



The viewer will promptly display a top view of the surface we have estimated. Your viewer should look like this:



Please note the gray sphere in the upper left region. This sphere denotes the location of highest uncertainty at this site. A detailed discussion of this subject is presented in Workbook 2 - DrillGuide©. In subsequent topics of this workbook this sphere will not be visible.

Instance plume_area

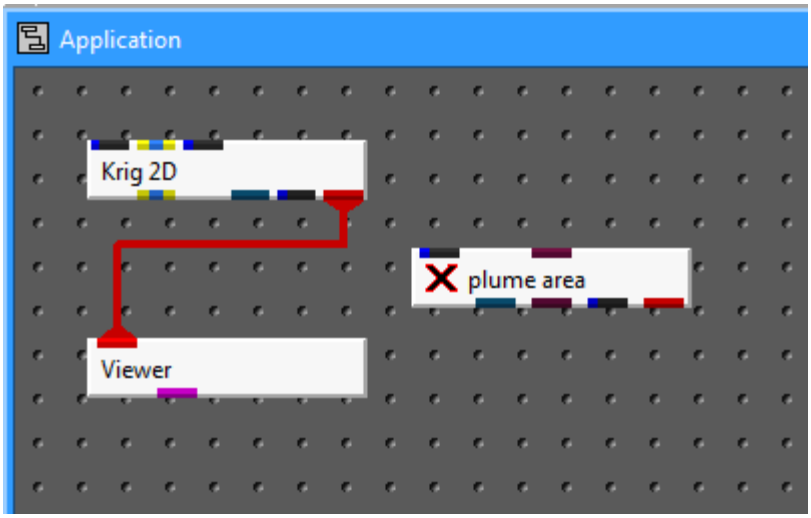
plume_area is a module that allows us to subset the output of Krig_2D to include only those regions having nodal data values above (or below) a specified iso-level. It also allows us to select the specific scalar nodal data component we wish to use for coloring the graphical objects.

Locating modules

plume_area is the third module in the Subsetting library in the top section of the Network Editor.

Instance plume_area

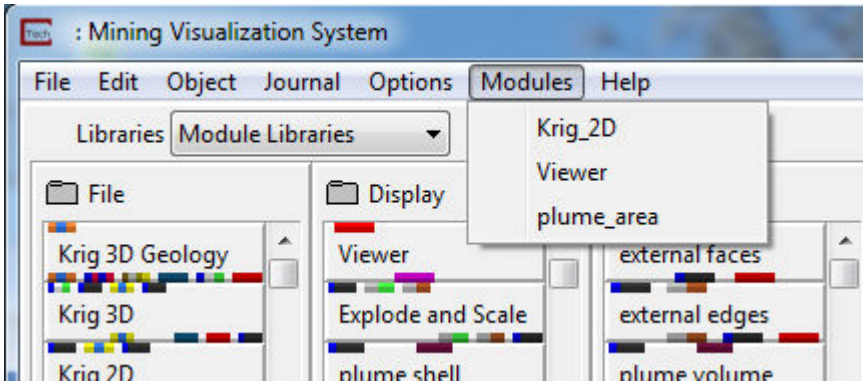
Point to plume_area; then, holding down the left mouse button, drag plume_area into the Application window .



EVS OR MVS doesn't care how we place modules in the Application window . But a good placement of modules results in cleaner looking connection lines, as we'll see in a later step. What you should note here is that plume_area in NOT in the network. The data flow is from Krig_2D to Viewer and is not passing through plume_area.

Modules List

The Modules List now shows Krig_2D, Viewer and plume_area:

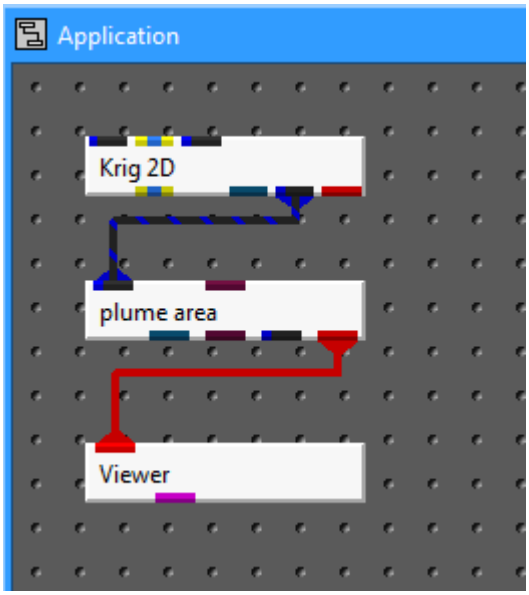


However, you don't really ever need to use the Modules List, since you can open any module's user interface (Control Panel) by double clicking on that module.

Insert plume_area into the current network

Let's now disconnect Krig_2D and Viewer and create a network from Krig_2D to plume_area to Viewer.

First, With the mouse, point to plume_area's leftmost input port (called input port 1). Next, connect the pointer to Krig_2D's blue-black output port. Now connect Viewer's input port to plume_area's second (red) output port, as shown in the diagram below.



What each connection means

Let's look briefly at what each connection means:

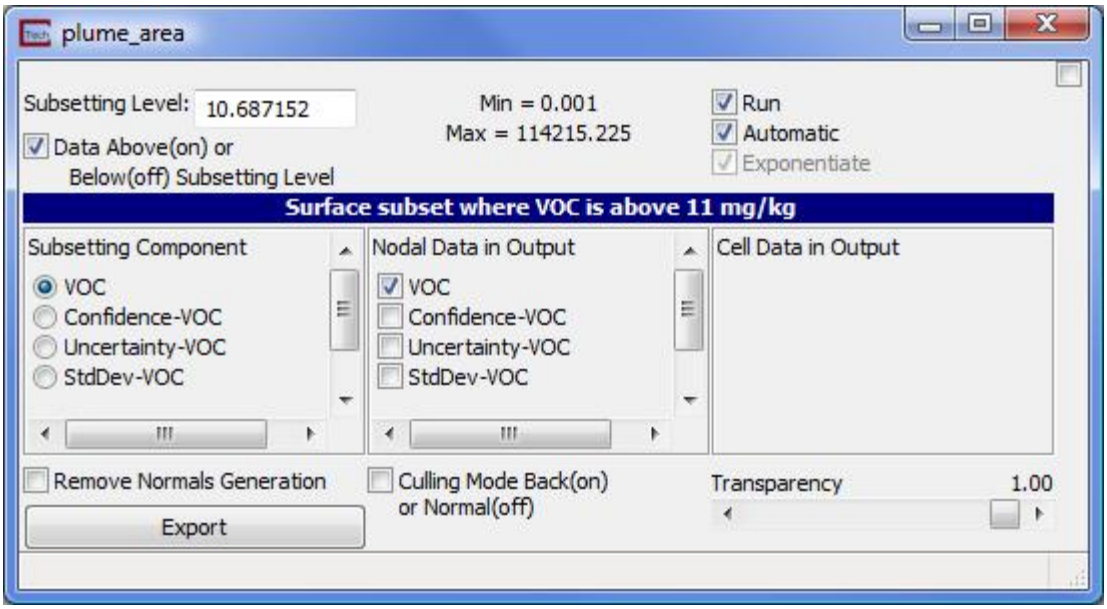
Connection	Meaning
plume_area to Krig_2D	plume_area creates a subset of the output of Krig_2D. Krig_2D reads an EVS analyte (e.g. chemistry) file. Its third output port (blue-black) makes the data available as an EVS field.
Viewer to plume_area	Viewer defines a full-featured viewer. It renders the graphics-display objects passed to its input port. plume_area makes its output available both as a field (blue-black output port 1) and as a graphics-display object suitable for rendering (red output port 2).

Experiment with plume_area

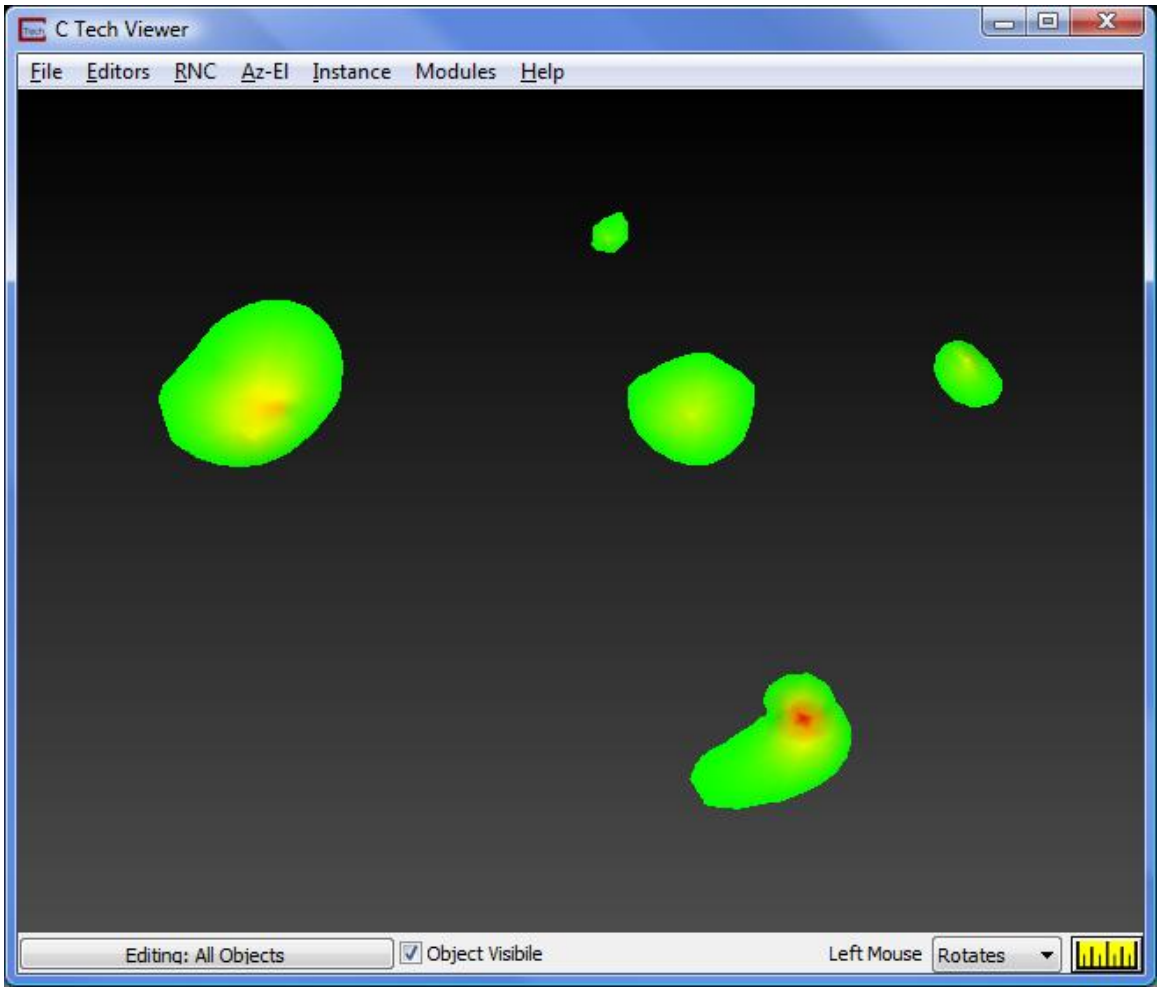
The plume_area module creates a surface whose data lies above a constant value. Let's experiment with plume_area's user interface, accessible through the Modules - pull-down menu.

Work with plume_area

a. Select the plume_area module's user interface from the Modules - pull-down menu or by double clicking on the module in the application area. The user interface for plume_area appears.

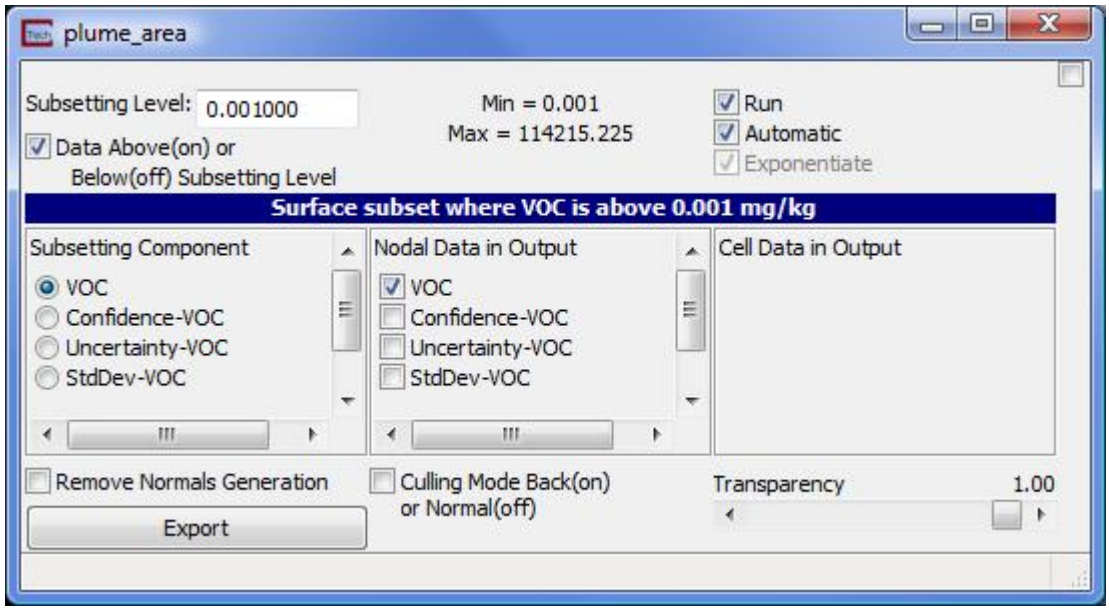


Notice that the current Subsetting Level is 10.687152 This means that we're producing a subset of the data for all values greater than 10.687152 mg/kg. Because only small regions of the domain are above 10.687152, the image we produce has several disjoint colored regions.

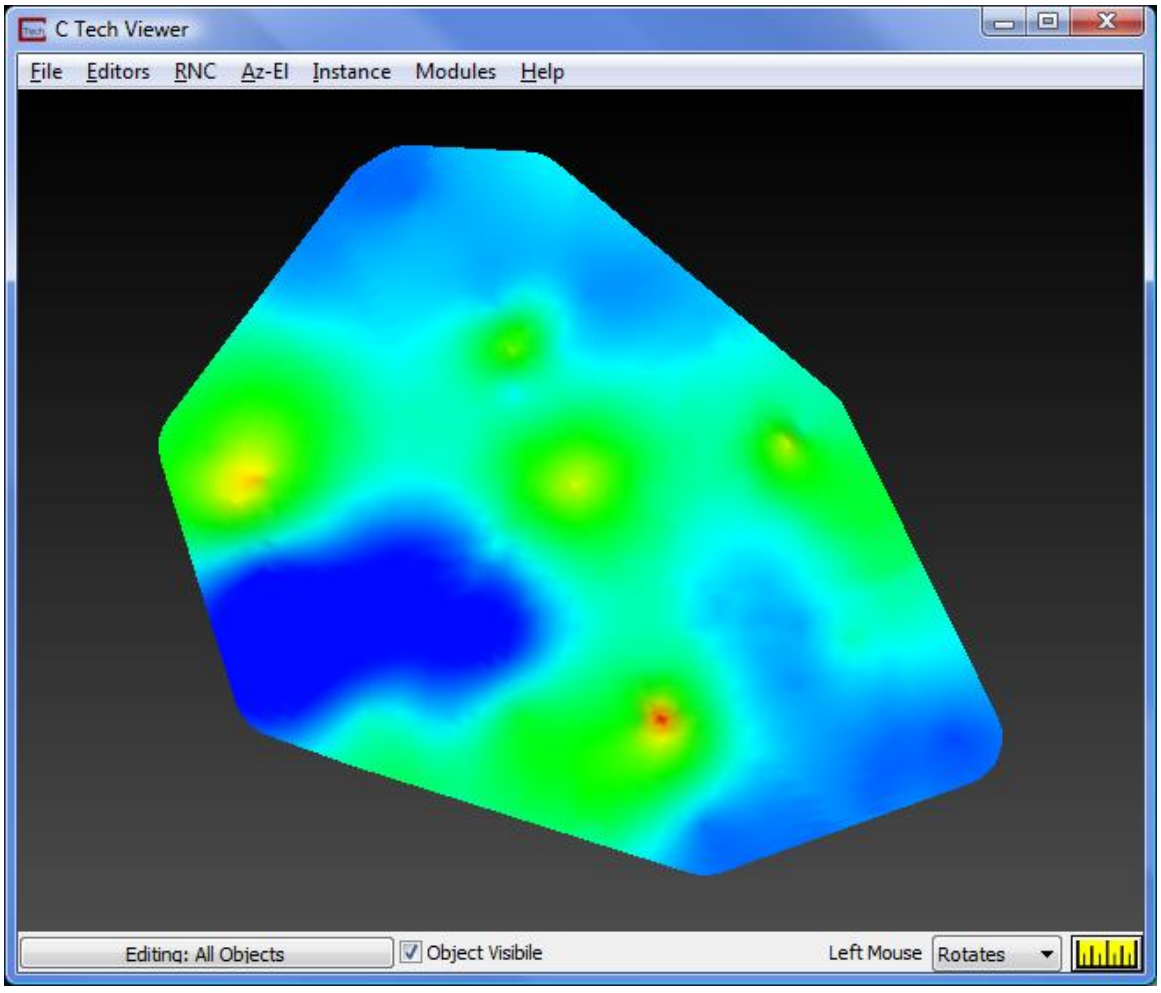


b. Actually, our data was "kriged" after being "log" processed. However, the displayed value of 10.687152 corresponds to an actual concentration just over 10 mg/kg.

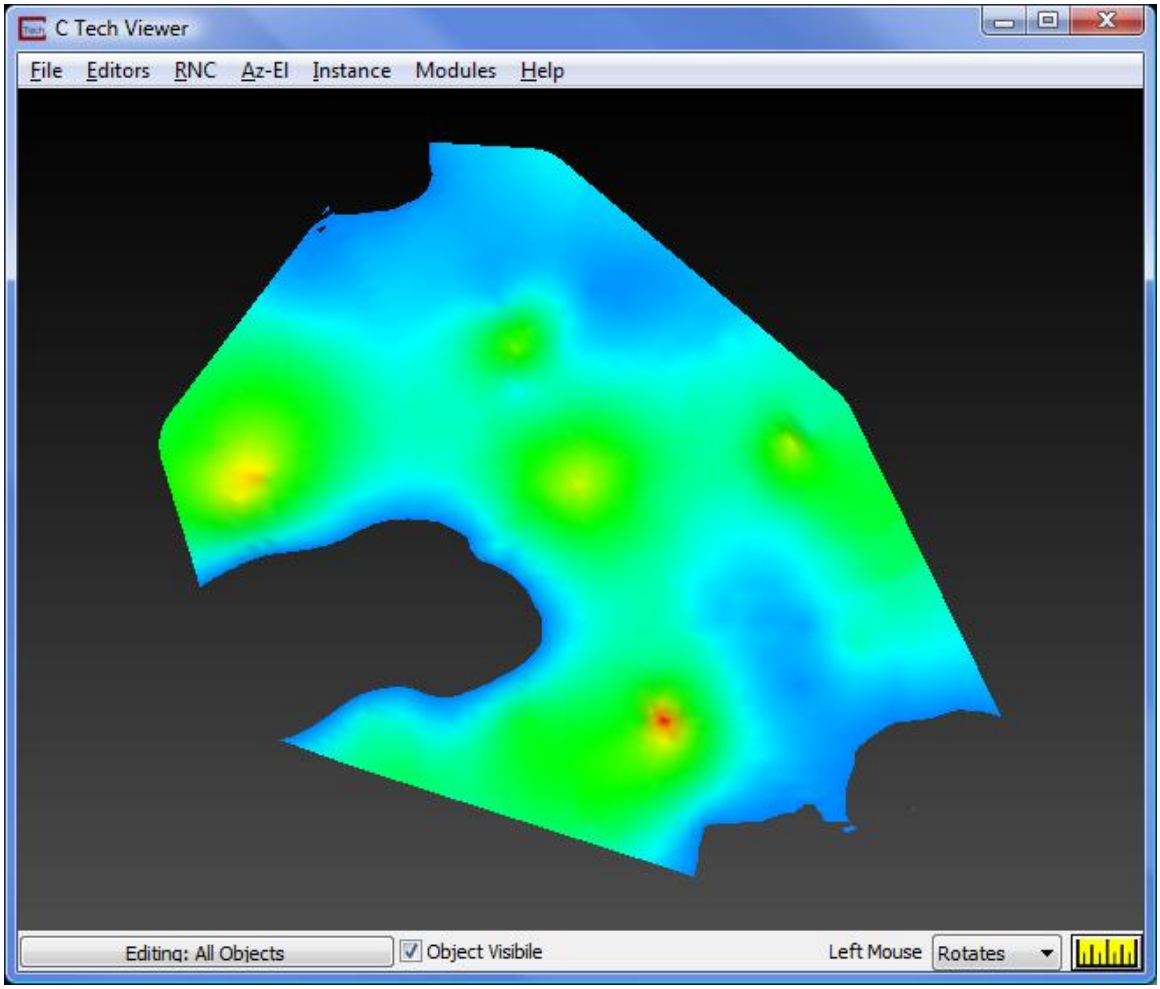
If we want to see the entire kriged area, we would set the Subsetting Level to the Min(imum) value for our selected Subsetting Component (VOC). That value is 0.001



Because the plume_area previously gave us only a portion of the view, the viewer will automatically normalize and center the object and you should get the following view.



Let's go back to a subsetting level that will give us only a portion of the model and we'll fill in the rest with lines corresponding to the entire model's grid. The value must be within the displayed min and max values (0.001 to 114,215.225). For this example we will choose 0.01



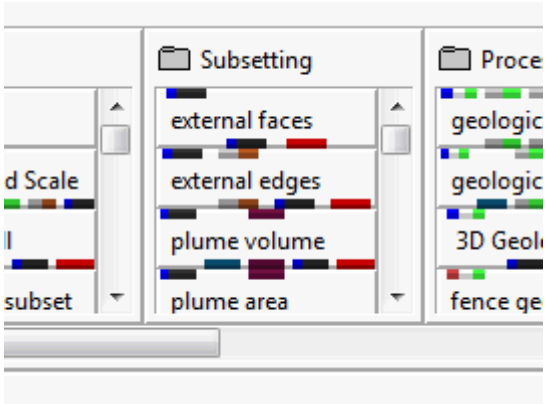
Add external_edges

external_edges is a module that draws lines corresponding to the edges of an object. The user interface panel for external edges has a parameter called edge angle. This angle is defined as the angle between two faces or cells in the model. If the angle is greater than the set edge angle, the edge is drawn. External edges are always drawn and setting the edge angle to 0 (zero) causes all cell boundaries to be drawn.

Locating the external_edges module

Scroll up (if necessary) to external_edges in Subsetting by clicking the up-arrow on the scroll bar. Scroll to the top of the list and you will see external_edges is the second module down in the list.

We can now see external_edges:

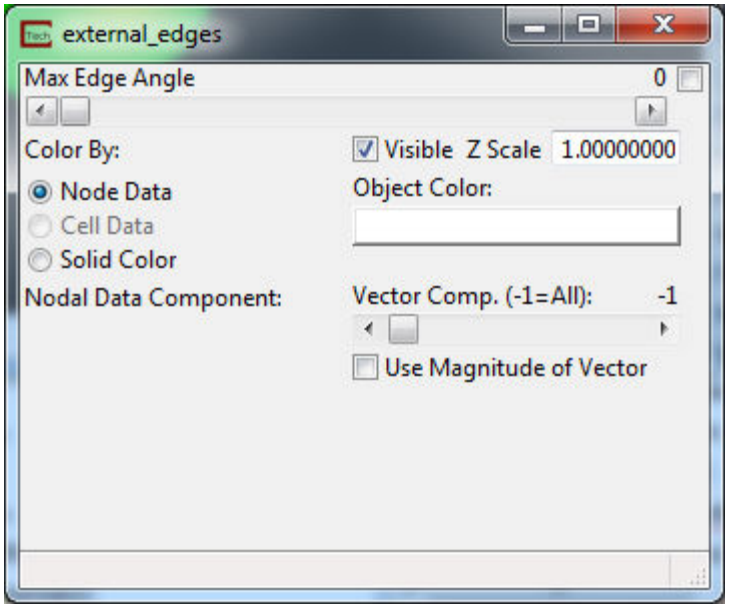


Instance external_edges

Instance external_edges. Point to external_edges; then, holding down the left mouse button, drag external_edges into the Application window .

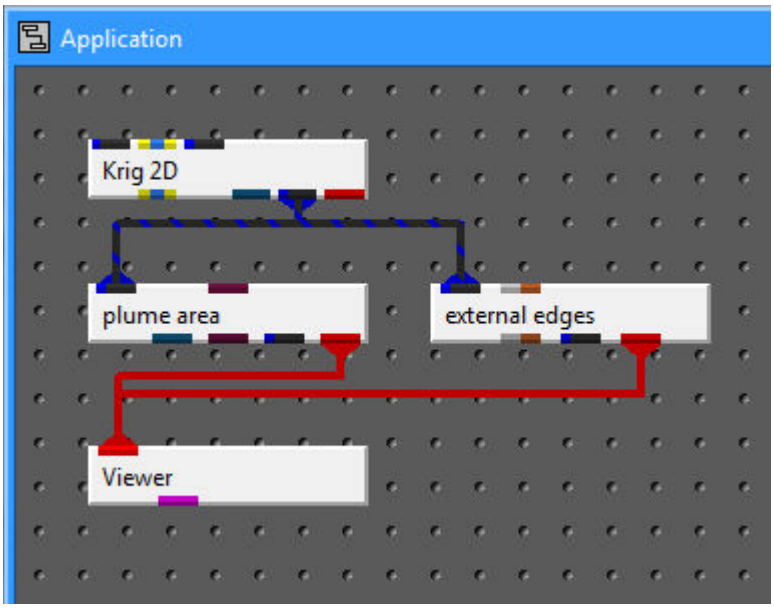
Double click on external_edges to make its control panel visible. We'll modify the external edge angle before connecting it. (There is no real advantage to doing it first because this module is very fast, however for more complex modules and for larger models, we will avoid computing the output twice by selecting the parameter values before connection.)

Change the edge angle to 0 (zero) as in the example.

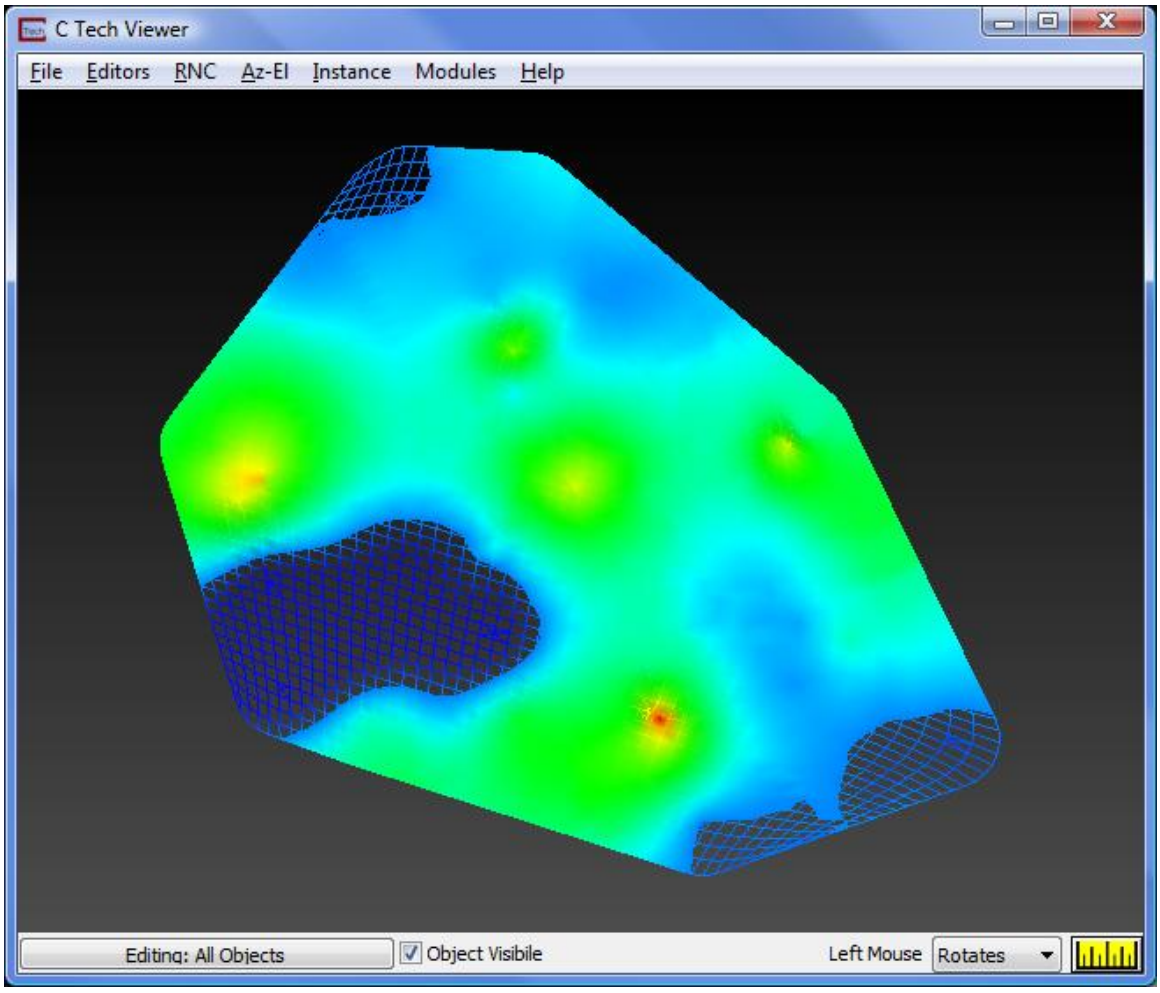


Connect external_edges to Krig_2Dand Viewer

Connect external_edges to Krig_2D using the same Krig_2D (blue-black) output that is connected to plume_volume. All output ports can have multiple connections, but only a few input ports can. Viewer's red input port can have any number of inputs. Connect external_edges' red output port to the Viewer. At this point your network should look like this:



and your view should have changed

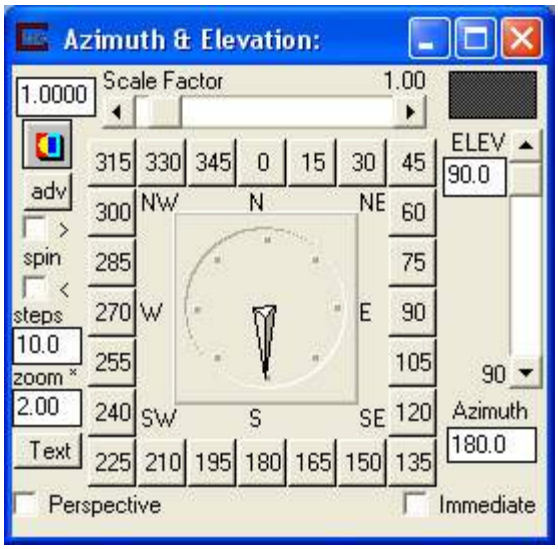


Transformations with the Az-El Panel

Viewer gives us another more precise way to transform (scale, and rotate) an object: through the Az-El panel.

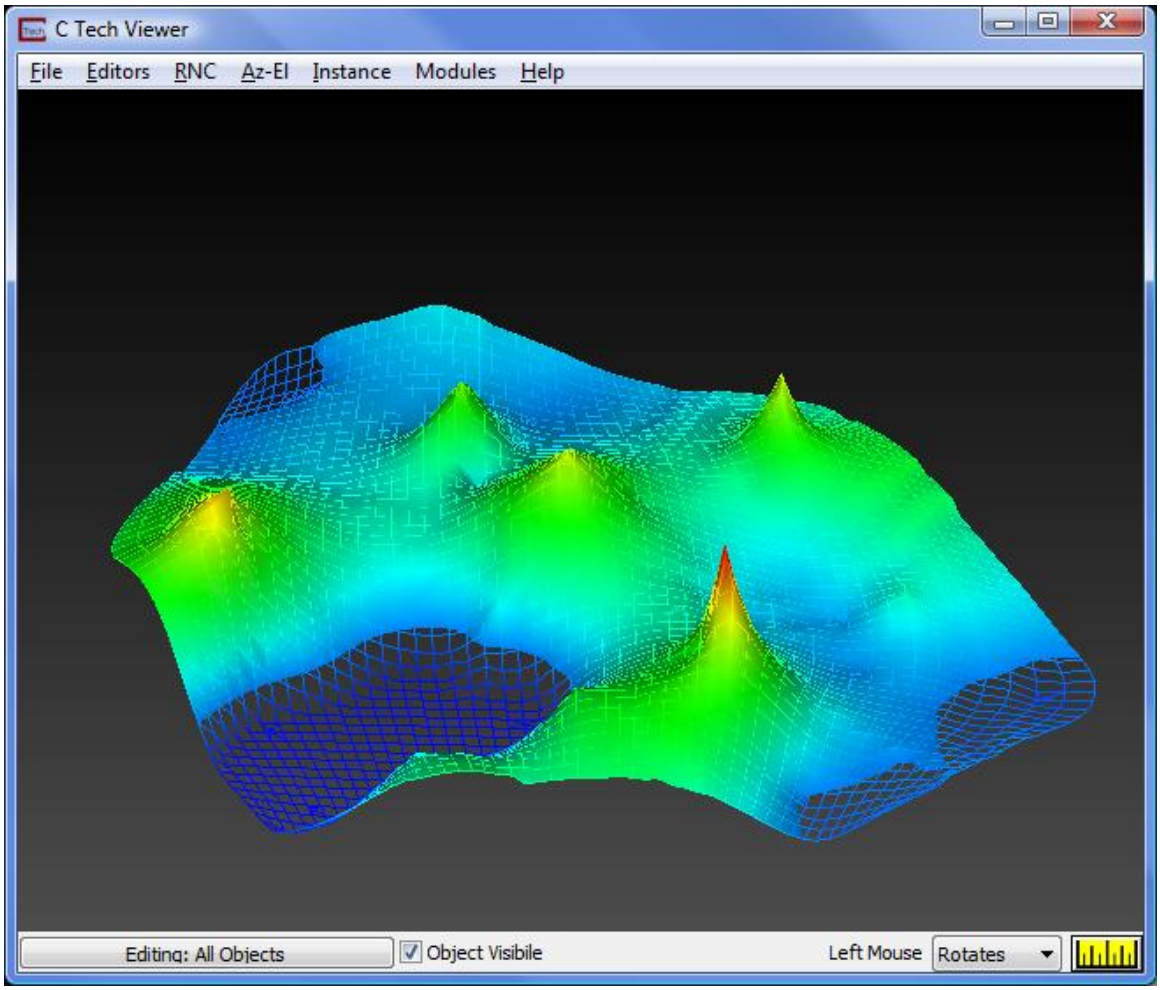
Transform the object with the Azimuth and Elevation Panel

a. Click on Az-El on the Viewer's main menu.



b. A panel pops up below the viewer with two sliders and an array of buttons. This panel allows you to instantly select a view from any azimuth and elevation. For a given (positive) elevation, selecting different azimuth buttons is equivalent to choosing a different compass heading and flying there in a helicopter to view the objects. The azimuth buttons are the direction from which you view your objects. (i.e. 180 degrees views the objects **from** the south). An elevation of 90 degrees corresponds to a view from directly overhead, 0 degrees is a view from the horizontal plane (side view) and -90 degrees is a view from the bottom.

c. Use the Azimuth and Elevation Panel to obtain a specific view by setting the scale slider and elevation slider to desired settings and click once on the desired azimuth button. If you choose a scale of 1.1, an elevation of 45 degrees and an azimuth of 195 degrees you should get something like this:

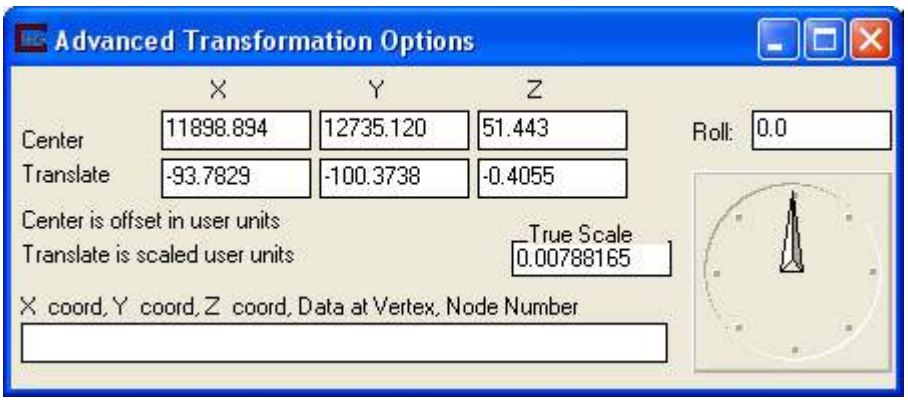


d. To obtain a sequence of views try clicking on all the buttons in sequence (quickly, without waiting for the viewer to redraw). All of your mouse actions are queued creating a "script" for a series of views. These views are equivalent to flying around your objects at a constant radius and elevation. While you're at it, try toggling the spin toggles on the Az_El panel (either '<' or '>. These will activate sequential views around the compass whose separation matches the specified type-in 'steps' value (the default here is 10 degrees).

After the next two topics we will adjust the line width for external edges. This will remove the broken looking lines in this figure which are caused by Z-Buffer ambiguities.

Advanced Transformations with the Az-El Panel

Further options with the Az-El panel are available by clicking on the "adv" button (just below the three color reset/normalize/center button) near the upper left corner of the panel. This pops up the window shown below:



The advanced transformation options are provided to allow rotations about a user defined center, as opposed to the default center of the objects, which is chosen by EVS OR MVS. The default center of an object shown in our viewer is midway between the min-max of the x, y and z dimensions. First, observe the default rotation center by clicking one of the spin buttons in the Az-El panel. Now let's rotate the object about a point that is located to the west of the object's center to demonstrate the use of the advanced transformation option.

a. Change the Az-El settings to a scale of 0.9, an elevation of 45 degrees and an azimuth of 180 degrees. Now, observe the default rotation center by clicking one of the spin buttons in the Az-El panel.

b. Now let's adjust the center of rotation by using the Alt Left-mouse action. First, depress the Alt button on the keyboard, then (while continuing to hold the Alt button) point to the anywhere on the right side of the mesh and press the left-mouse button. This action "selects" the object, AND activates the SEEK button in the upper right corner of the Az_El panel. Now press the SEEK button and your view will 'zoom-in' on this spot, thus establishing a NEW center of rotation. You may not have noticed this, but the center and translate values in the Advanced Transformations dialogue box were automatically updated. Although you could have manually entered values, the above approach is the most intuitive and efficient means of changing the center of rotation in the Viewer.

Observe your new center of rotation by clicking on one of the spin buttons in the Az-El panel and notice that the object is now rotating about a new center point. Also note that part of the object rotates outside the viewer window. The Translate options are provided to address this issue. The XY Translate dialogue boxes move (translate) the camera view when using the Az-El panel. Test this now by typing in a value of -108.0 into the Y Translate type-in box.

d. Wherever you select with an Alt-Left Mouse Click will appear in the bottom of this window. It will provide you with both coordinates AND the data value on the point on the surface.

e. Again Click on one of the spin buttons in the Az-El panel and notice that the object is still rotating about the new center point, but the object has been shifted over by the Translate function so the object is more visible in the viewer window during rotation. For further experimentation with these

options click on the three colored button (reset/normalize/center) and repeat the above steps as desired.

For details and specifications on the advanced transformation options please read:

- Module Libraries Help on Viewer

Transformations with the Mouse

Rotatethe model

Move the mouse to a location within the main portion of the Viewer's window.

Hold down the left mouse button and move the mouse pointer in various directions. The model rotates.

Release the mouse button.

NOTE: The best way to understand mouse rotations is to consider the viewer screen as a ball floating in a bowl of water. Where you start with the mouse and where you end (release) determine the angle of rotation.

Scalethe model

a. With the mouse, point to and select the object in the View window.

b. Hold down both the Shift key and the left mouse button (or the middle button alone).

c. Keeping the Shift key and mouse button held down, move the mouse pointer downward or to the left. As we do, the model scales down. Moving the mouse pointer upward or to the right scales up.

d. Release the Shift key and mouse button (or the middle button).

e. The wheel on wheel mice also zooms in and out.

Move(Translate or Pan) the model

a. With the mouse, point to the model in the View window.

b. Hold down the right mouse button and drag the object up, down, and around, then center the model.

c. Release the mouse button.

Mouse-controlled operations	What to do
Translate	Drag the object with the right mouse button (RMB)
Rotate	Drag the object with the left mouse button. (LMB)
Scale	Hold down the Shift key and drag the object with the left mouse button. (Shift-LMB) or Use the middle mouse button or wheel as a button without Shift

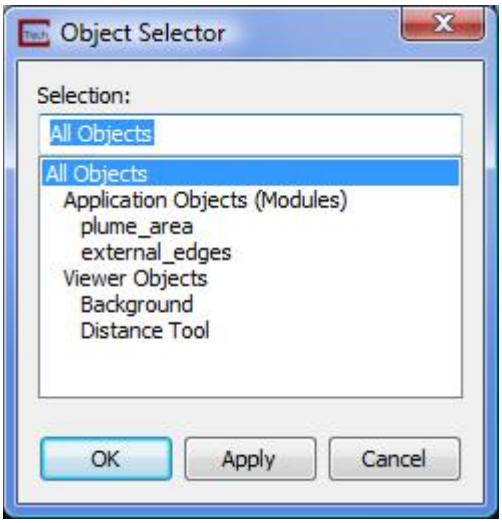
or

Use the Wheel (on mice so equipped)

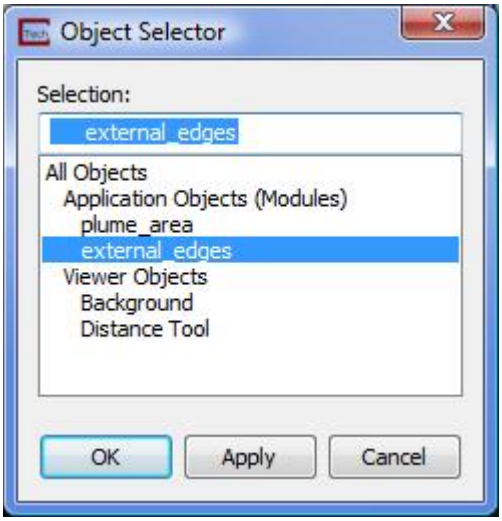
Manipulate the objects

Use Viewer's capabilities to manipulate the objects. For example, here is a series of steps for testing the inheritance of properties:

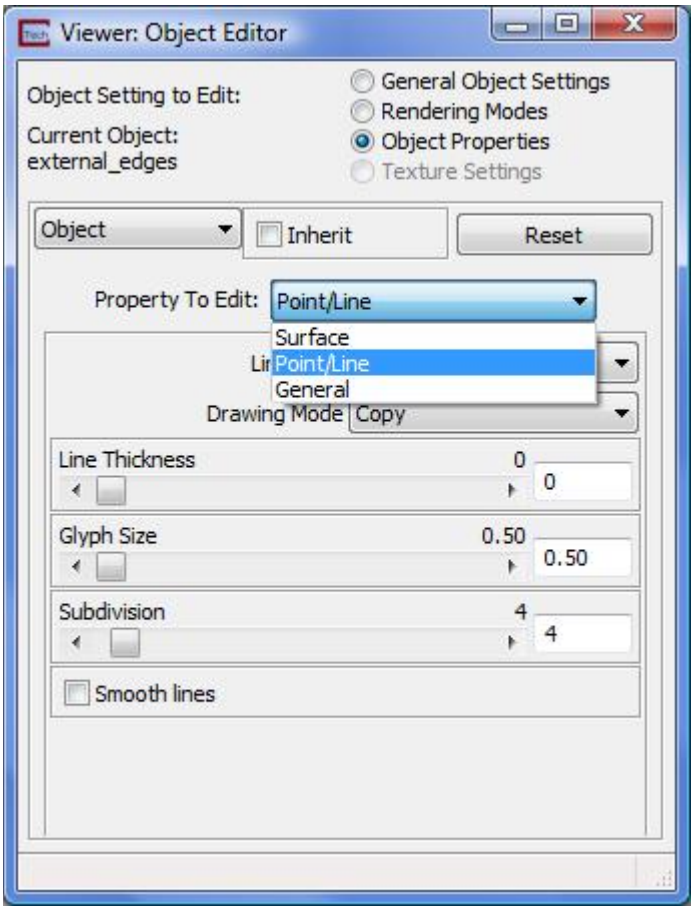
a. Display the object selector by clicking the 'Editing:All Objects' button on the Viewer. The button's name reflects the object currently selected, as we will see in the following steps. Once you click this button



b. Select the external edge object and press the Apply button. Do not use OK, because it will close the object selector and we do not want to close it unless the Top object is selected.

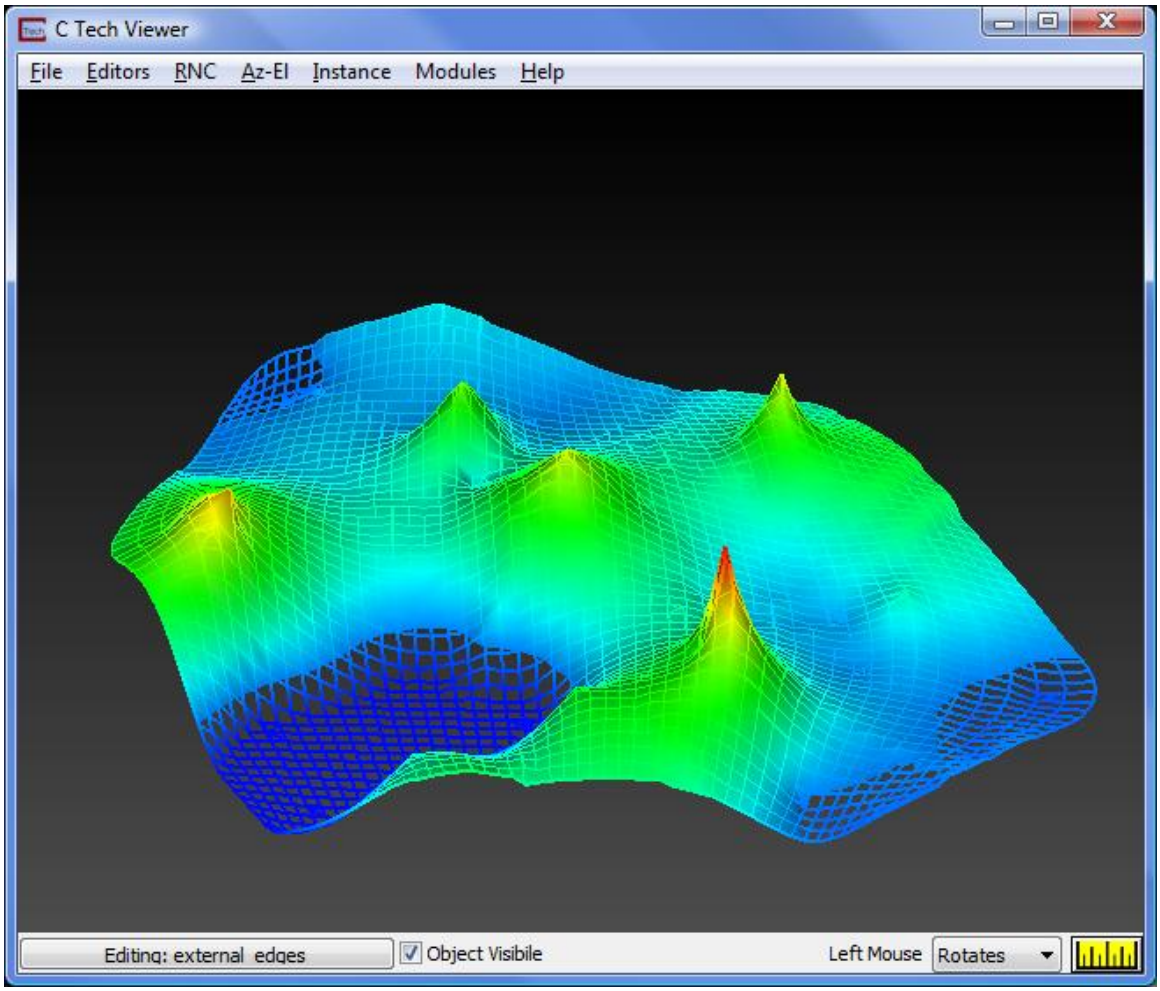


c. Now, from the pull-down menus on the Viewer, select Editors->Object->Advanced Settings. Inside the Object Editor select Object Setting to Edit: 'Object Properties' and select Property to Edit: 'Point/Line'. Your panel should look like this:



d. Use the line thickness slider to change the external_edges' line width from 0 to 2. (Note: 0 is a special case that actually produces a line width of 1, but uses the fastest drawing routines) Actually, we could have changed line width for All Objects because by default, an object inherits the properties of its parent, so when we change the top-level object, all objects change.

At this point, your Viewer should look just like this:



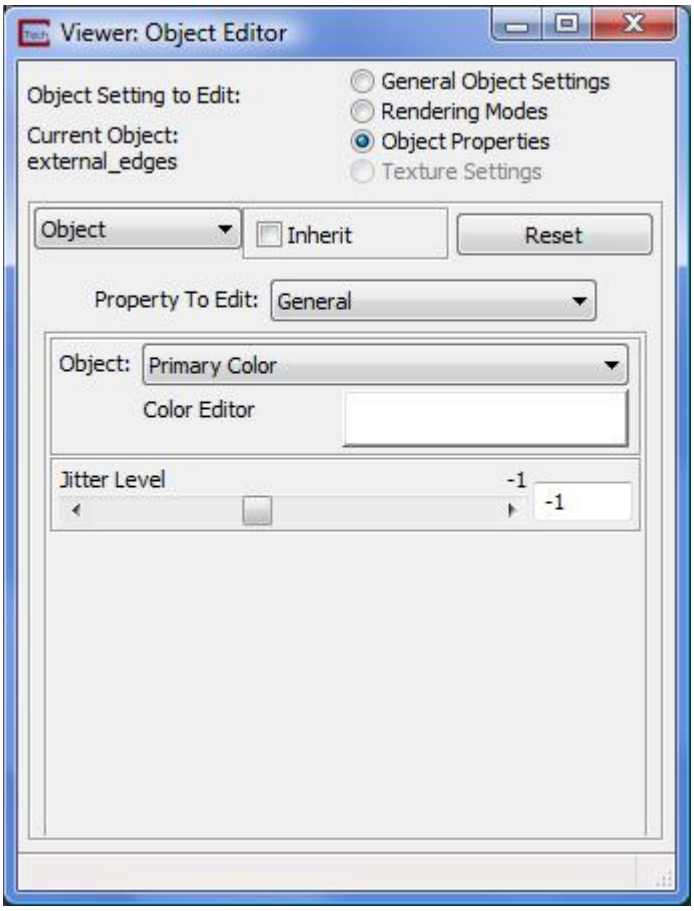
e. There is a completely different way to handle the broken lines that actually directly addresses the problem. The single pixel wide lines were broken because their location in the viewer was coincident with the underlying surface (created by plume_area). In the real world we never encounter situations where two objects occupy exactly the same space. However, in the world of computer graphics this is a common problem.

Normally, the process of deciding which objects we should see in the viewer is determined by an algorithm called "Z Buffering". This is simply the ordering of all lines, points and triangles so that the ones that are furthest from the observer are drawn first. When two objects are in exactly the same space, it is AMBIGUOUS which one is closer.

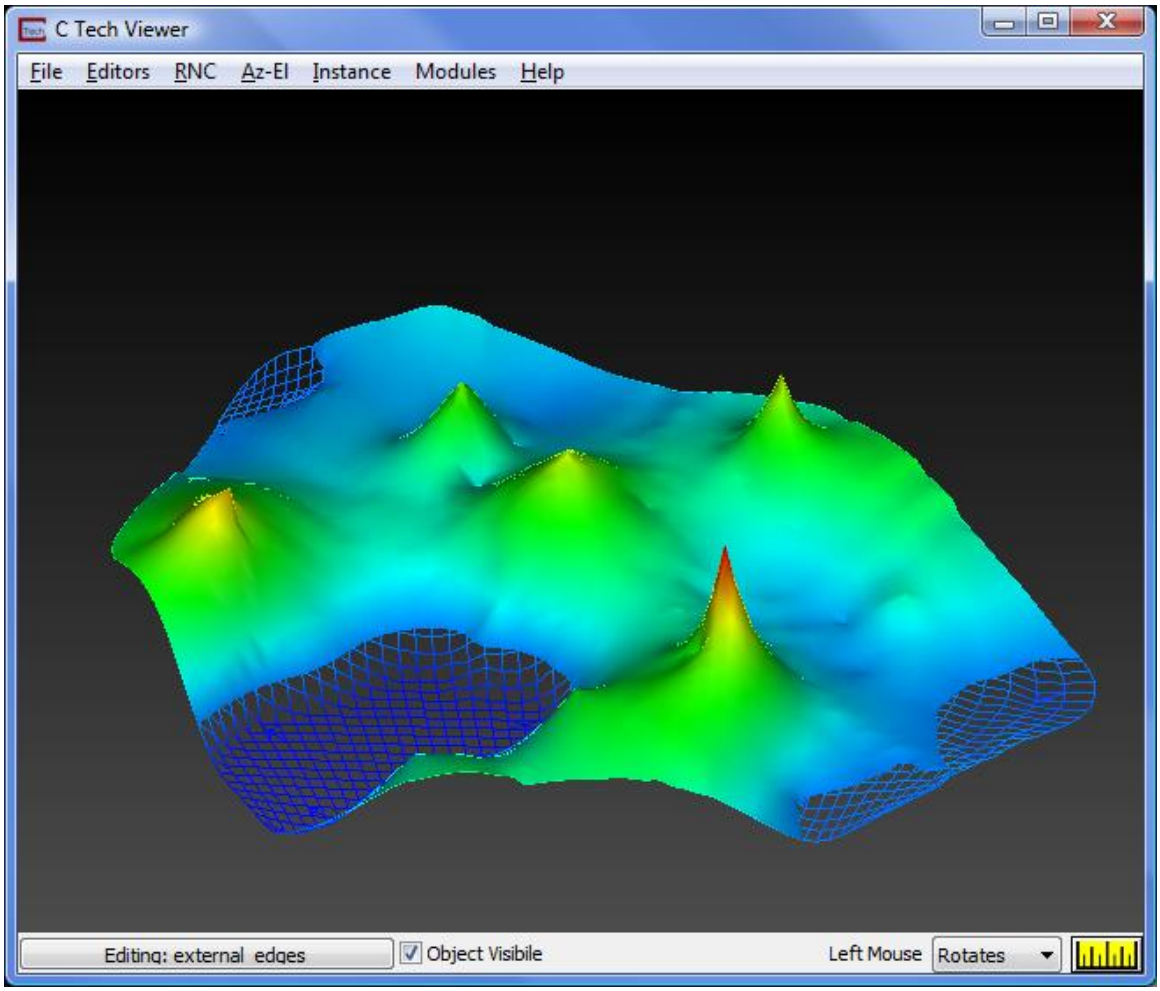
When this occurs, we can utilize another property called Jitter. Objects with higher jitter are preferentially seen over ones with lower jitter. Jitter allows us to solve this ambiguity problem in one of two ways.

f. First, let's set the line width back to zero.

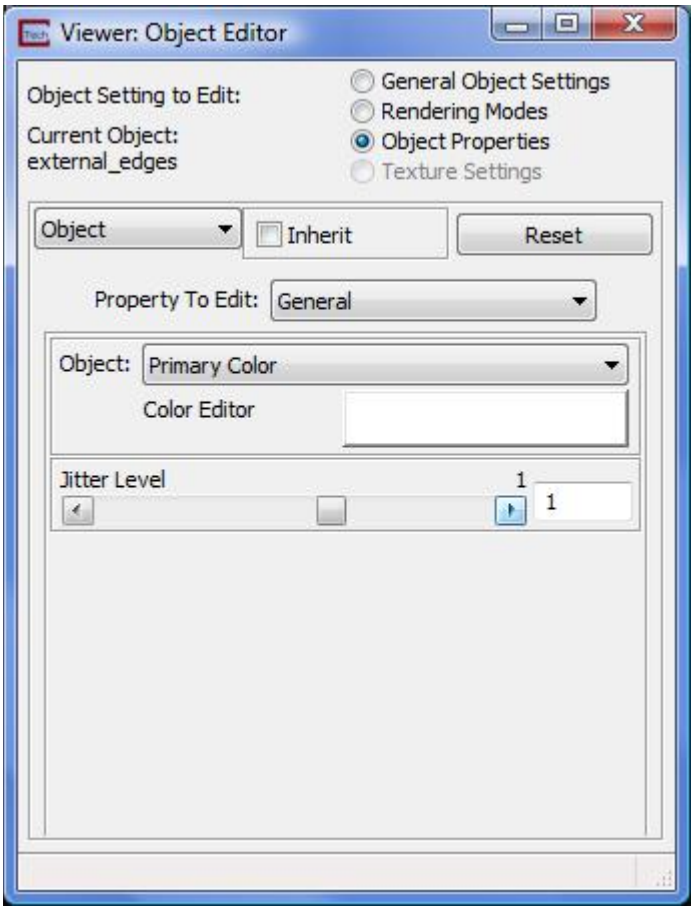
g. Now change the "Property to Edit:" to be General and set the Jitter Level to -1.



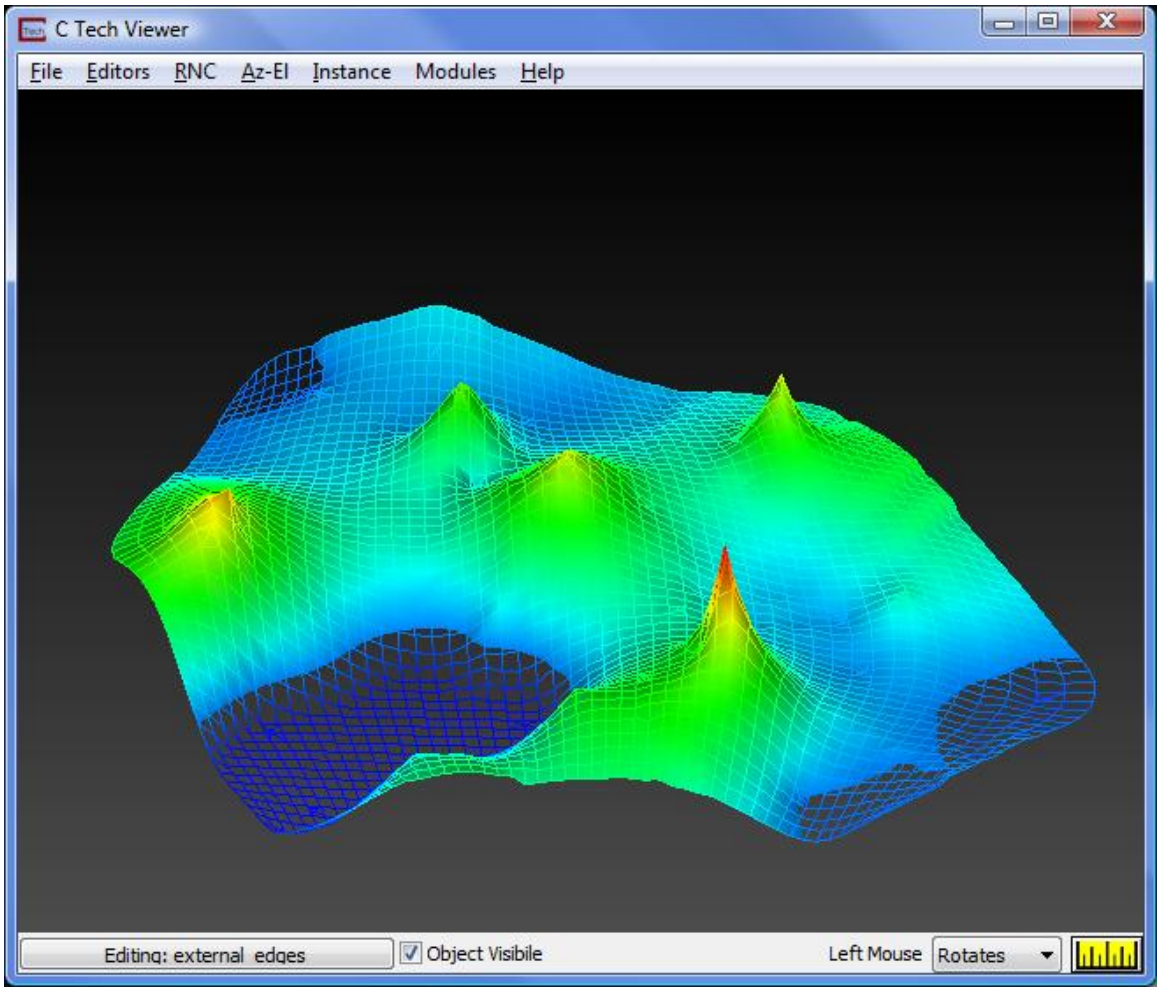
This will change our picture to something quite different. Here we'll have the cell outlines (external_edges) only where there is NO surface displayed.



h. However, if we set the Jitter to 1



We'll display the edges everywhere



i. Now go back to the Object_Selector and click on **All Objects** and select the OK button.

j. Now close the Object Editor window.

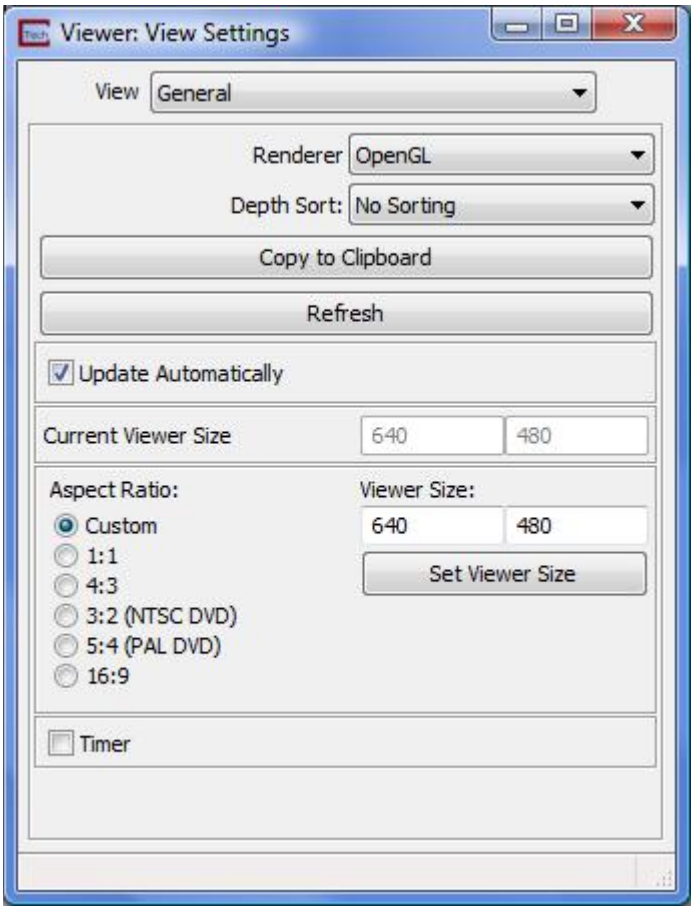
Explore Viewer's capabilities

Explore Viewer's editors and mouse operations. The tables and figures below summarize Viewer's capabilities.

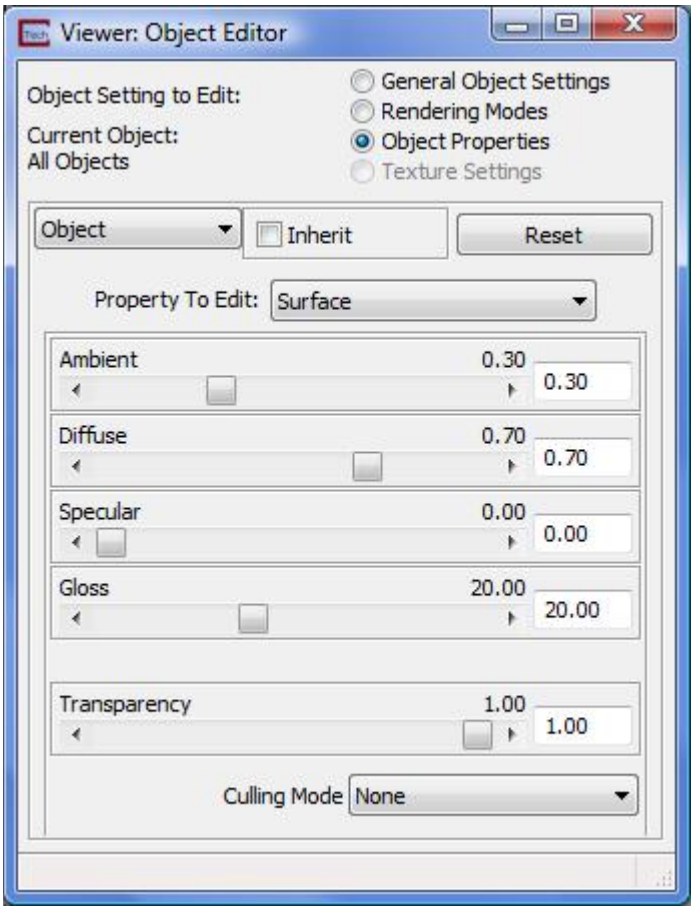
Editors under Viewer	Description
Object...	***Transparency, Culling Mode and Normals Generation***
	Advanced Settings
	options for changing properties, modes, etc:
	Note that most of the Object editing operations will effect only the selected object and those objects inheriting the settings. This feature is

	NOT under the editors pull-down, but instead is located in the bottom lefthand corner of the viewer. The button activates the dialog box for selecting the object of interest from a list of objects in the view.
View...	Lets you size the viewer in pixels, lets you choose software or OpenGL rendering and more. Settings for the viewer's background color are located here. The default is a gradient shading from black to medium gray, but any color can be obtained.
Camera...	An editor for the view's camera. Adjust the normalize and center preferences, lens settings (such as perspective w/ field of view), depth cueing, etc.
Light...	An editor for the view's light information. Note that the editor contains controls for settings such as a 'general' bi-directional light centered on the current object, AND an ambient light associated with the camera view. The widgets below the slider control the aspects of the current light, such as whether it is on or off, its type, and so forth.

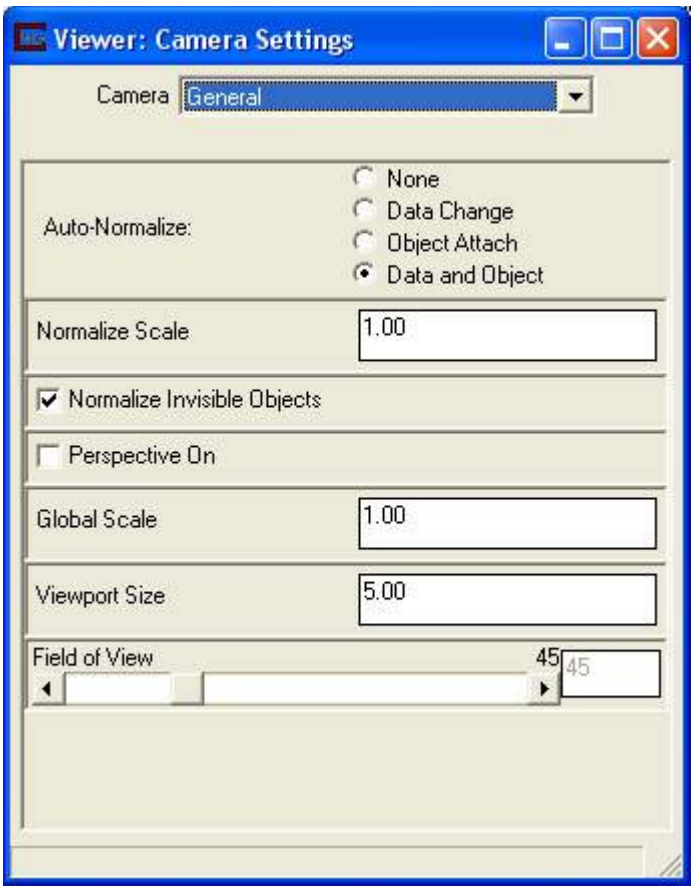
The View Editor ('General' panel only)



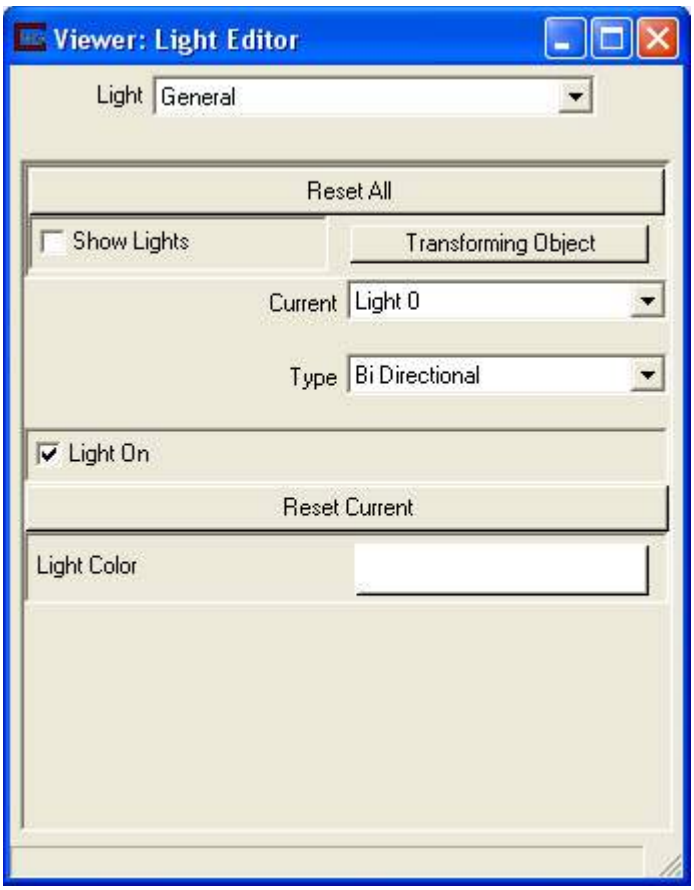
The Object Editor ('Properties' panel only)



The Camera Editor ('General panel only)



The Light Editor ('General' panel only)



Save the network

Let's now save our application, which we can do at any time by saving the network which includes specifications of the modules involved and the manner in which they are connected.

Save the application

a. Select the File->Save Application pull-down command.

The Save Application dialog box appears.

b. At the end of the Selection field, type Work_Book_1.v. (or any other name ending in .v.)

Exit EVS OR MVS

Let's exit EVS OR MVS.

Select the File->Exit pull-down command.

EVS OR MVS exits after displaying a confirmation message.

If you close the Status Window window, it will immediately exit EVS OR MVS without a confirmation message (but may have an error message). Be careful not to close this window by accident.

You may minimize the Status Window to get it out of your way. However, there are many useful messages that are sent to the console, so we highly

recommend that you keep it on your desktop when running the software (not minimized).

You have now completed Workbook 1.

Workbook 2 DrillGuide© Outline

DrillGuide© Analytically Guided Site Assessment

The EVS modules Krig_2D and Krig_3D provide the ability to perform Analytically Guided Site Assessment. We refer to this process as DrillGuide©. DrillGuide© allows you to use the trends in your existing soil or groundwater contamination data to determine the optimal location for additional sampling. This is accomplished by focusing on those areas of your site with high uncertainty. Uncertainty is computed using a proprietary C Tech algorithm that gives high values where concentrations are predicted to be high, but simultaneously where the confidence in that prediction is low.

DrillGuide© also allows you to determine when a site is adequately characterized.

Ask yourself - how you determine if you have collected enough data for a site? C Tech would answer that a site is adequately characterized when the statistical variation in predicted plume area (or volume) is acceptable.

This workbook will address these issues in detail for a particular real data set and hopefully teach you how to apply these concepts to your own sites.

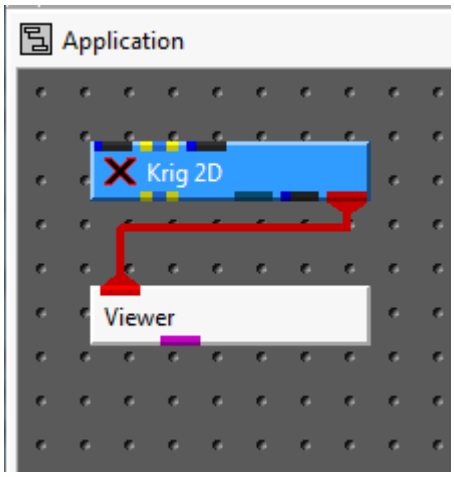
Important note: The purpose of this workbook is to teach you a new approach to sampling and site assessment. The focus is less on how to use the C Tech modules and more on ***why*** to use them. We recommend that you read through this entire workbook before trying to follow along on your computer. Focus first on understanding the concepts and later verify that you have the mechanics under your belt.

Two-Dimensional Kriging

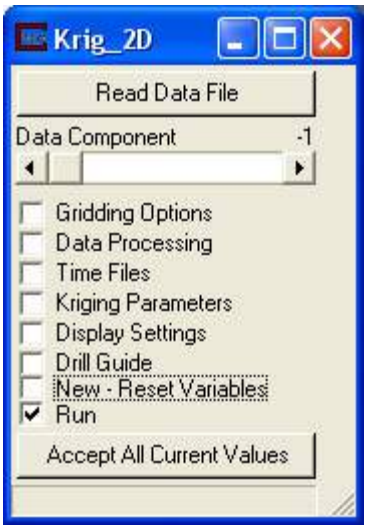
Most data collection methods require drilling (or punching) a hole in the ground at a particular x-y (Easting-Northing) location. Since there is not a practical way to sample at a particular x-y-z location without drilling/boring/punching from the surface, it makes sense to determine the x-y location of sampling and collect data at multiple elevations (depths).

For that reason, we generally use Krig_2D's two dimensional kriging to determine new sampling locations. The same techniques used in this workbook are also available with Krig_3D, however. You can perform the same DrillGuide© calculations in three dimensions by using Krig_3D.

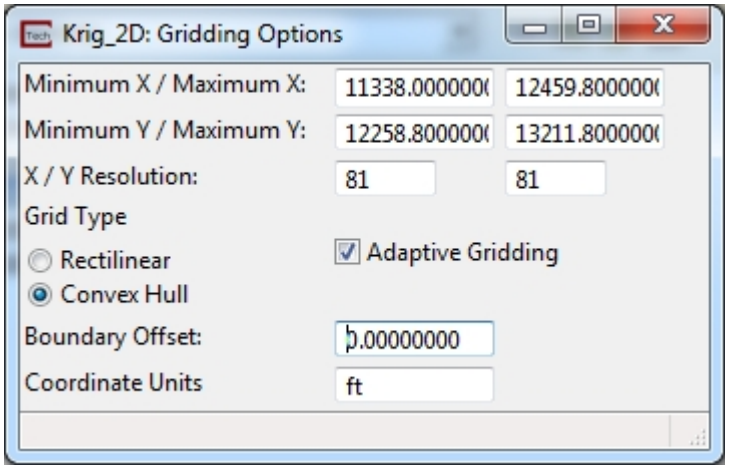
The first step is to krige your data. Begin by building the simple application shown below.



First, we'll turn OFF the "New – Reset Variables" toggle on the main window. This allows us to change values that the expert system would otherwise reset every time you run.

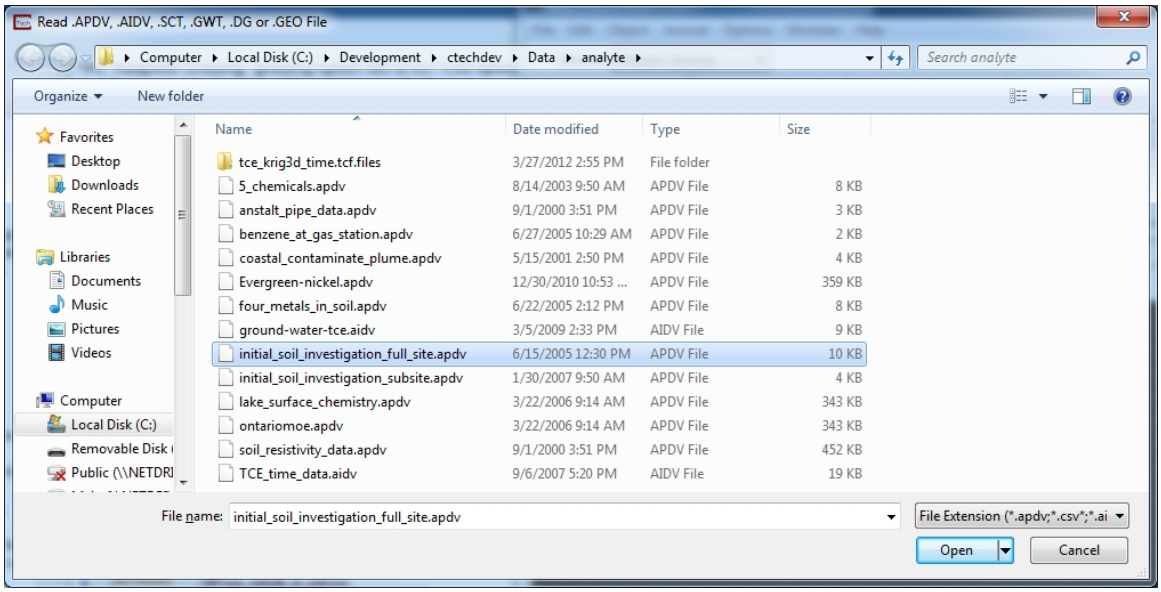


Open the Gridding Options window and note the "Boundary Offset:" parameter for the convex hull algorithm is 0.1. This would create a grid that was 10% larger than the convex hull of our data. Set this parameter to 0. This will force our grid to be exactly the convex hull of our data set and not offset beyond that boundary. This will ensure later, when we choose new locations, that they will be constrained to the convex hull of the original data. Sometimes you may want to have the new locations expand your site boundaries, but we will not do that for this workbook.

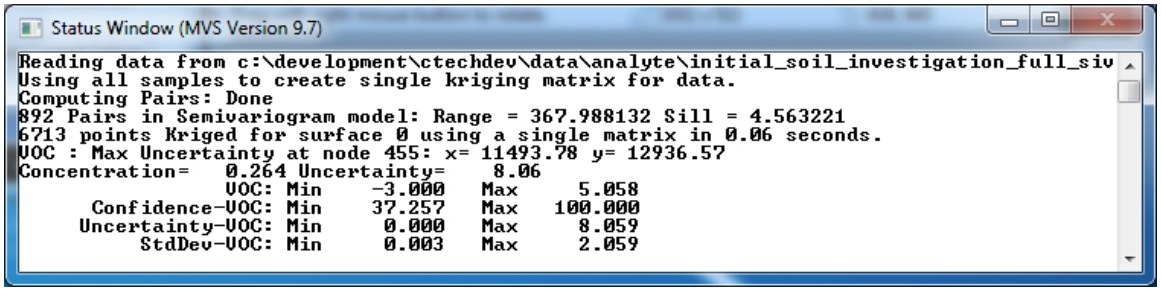


Close the Gridding Options and click on the "Read Data File" button:

and choose initial_soil_investigation_full_site.apdv as the analyte (e.g. chemistry) data file for this workbook.

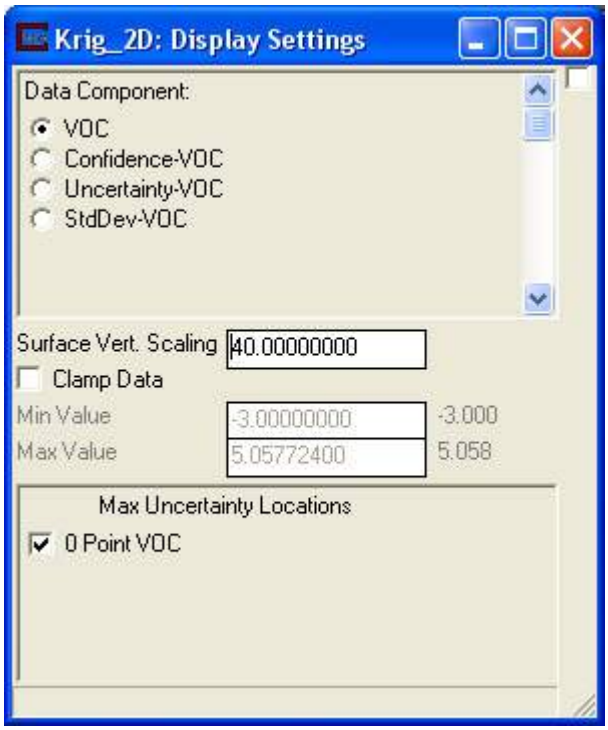


Click on the "Accept All Current Values" button and your message console should show the following:

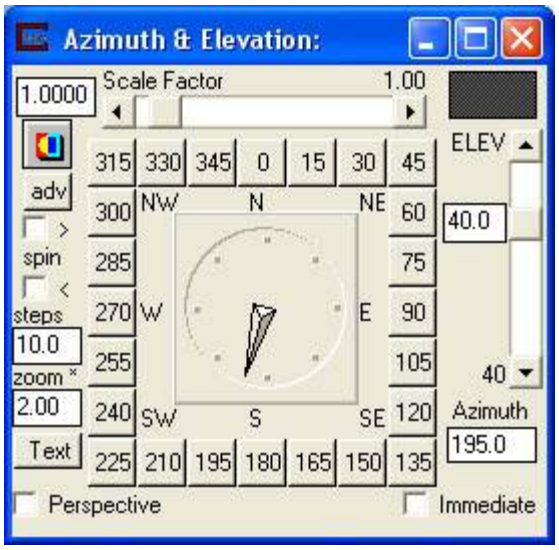


Please note that in the console above, this analysis was run with the demo version of EVS-PRO. There are messages here that apply specifically to the demo version. *"File check passed*" tells us that the file does have a valid password and will only be seen if you are running in demo mode.

Click on the "Display Settings" toggle on the main window to see the following:

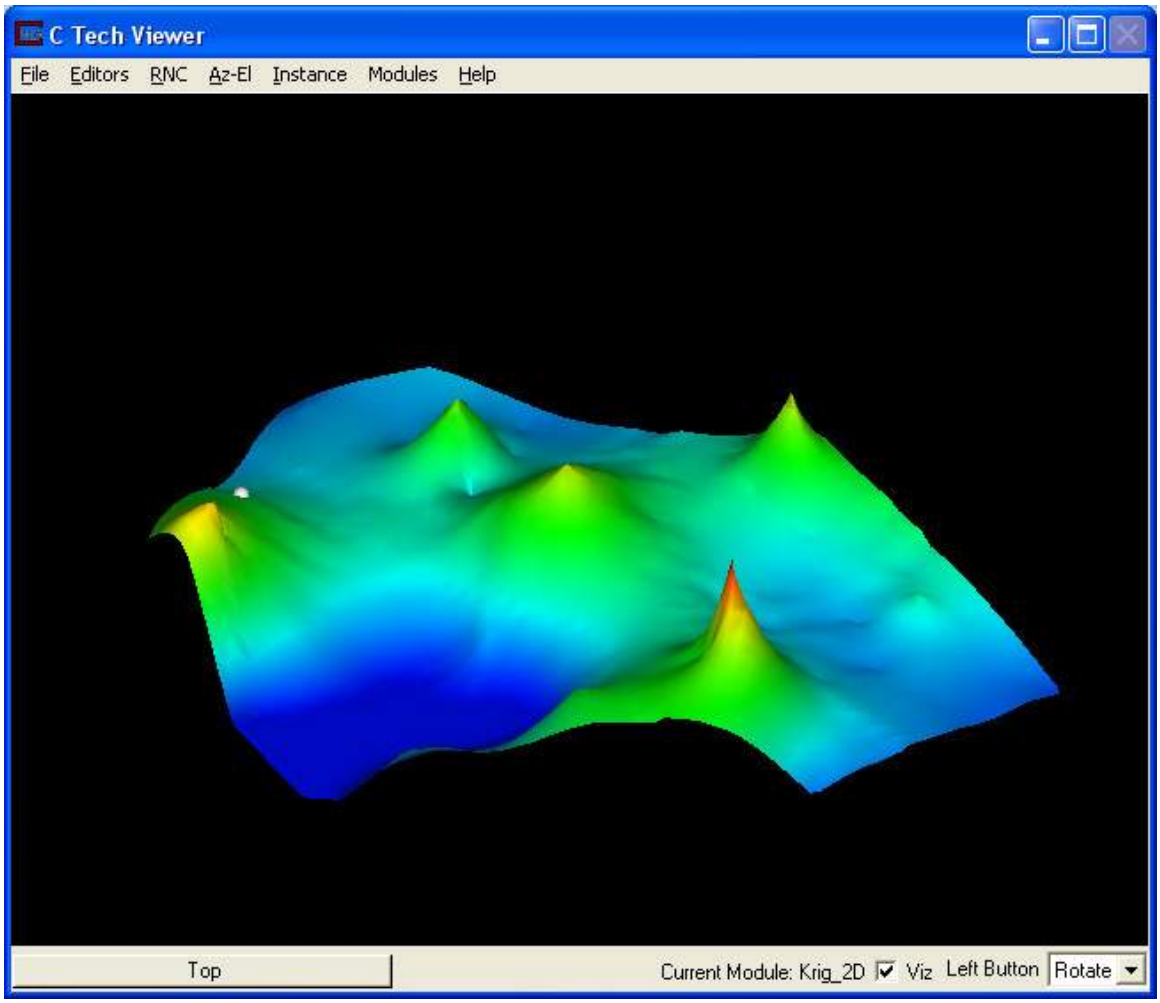


Change the "Surface Vert. Scaling" parameter above to 40.0 (default is 10.0) and set your view in the Az-El Selector as shown below.



At this point, your Viewer should look like the picture below. Note that the surface has both colors and elevation that correspond to concentration. High concentration regions are green or yellow or red and correspond to mountain peaks, whereas low concentration regions are cyan or blue and correspond to valley floors.

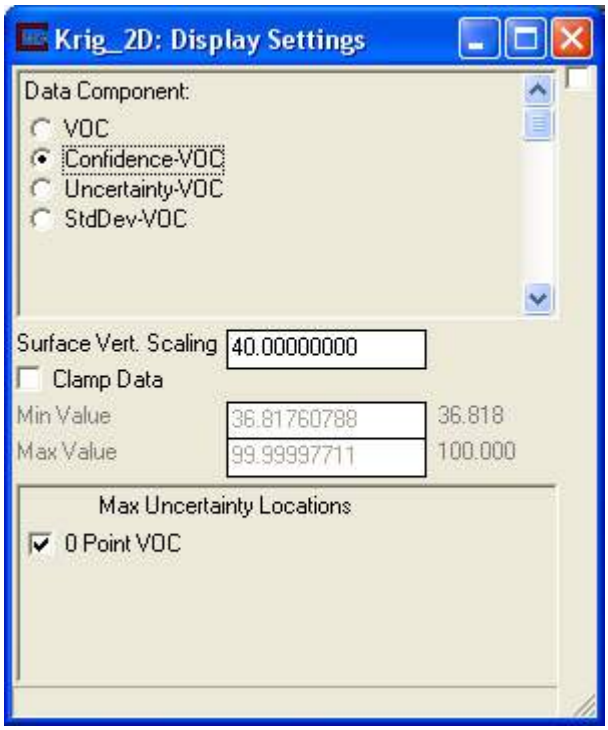
Additionally there is a gray sphere whose purpose will be discussed in subsequent topics.



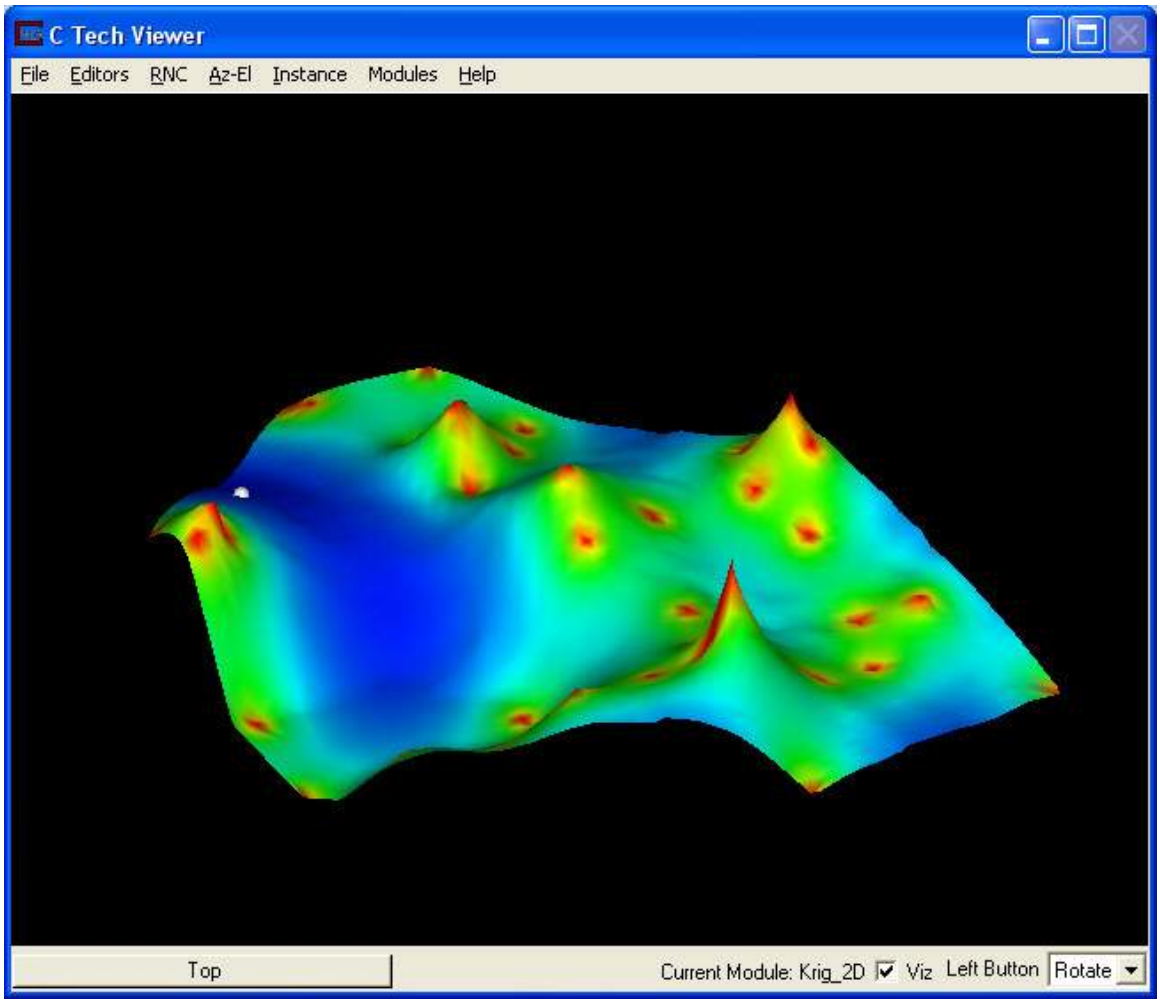
Confidence

Once again we pose the question of how to determine the optimal location for additional sampling. One answer ***might be*** to focus on locations in the site where the confidence in our prediction is low.

Change the "Data component" to *Confidence-VOC* as shown below.



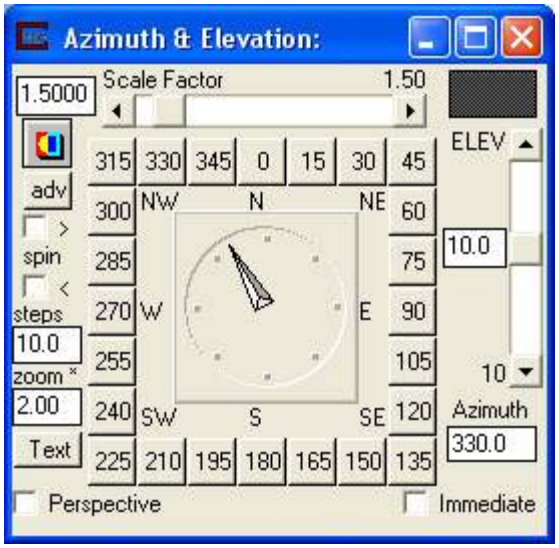
When you do, you should see the surface in the Viewer colored like a bad case of chicken pox. Actually, each red spot is one of the sampling locations. These locations are areas of high confidence (100%), since we **know** the concentration at those places. Everywhere else in the domain, we are **estimating** the values.



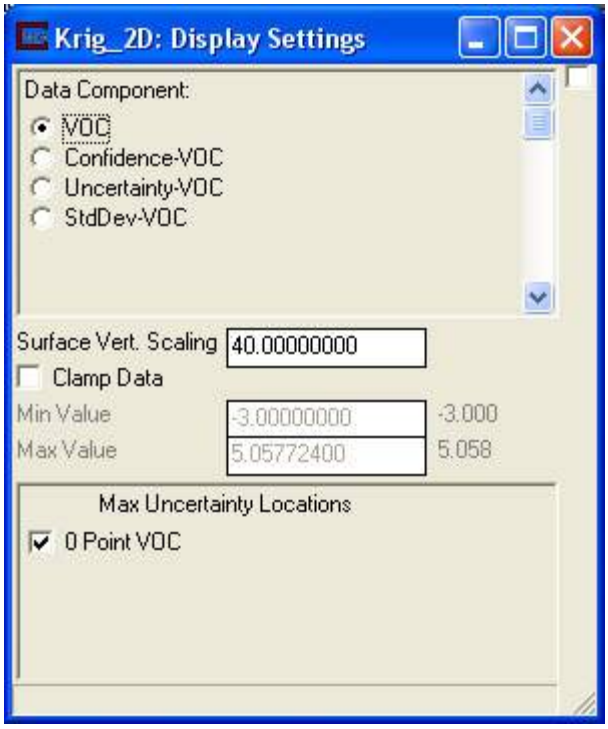
However, confidence alone is not the best measure of where to sample because it does not take concentration into consideration.

Max Uncertainty Location

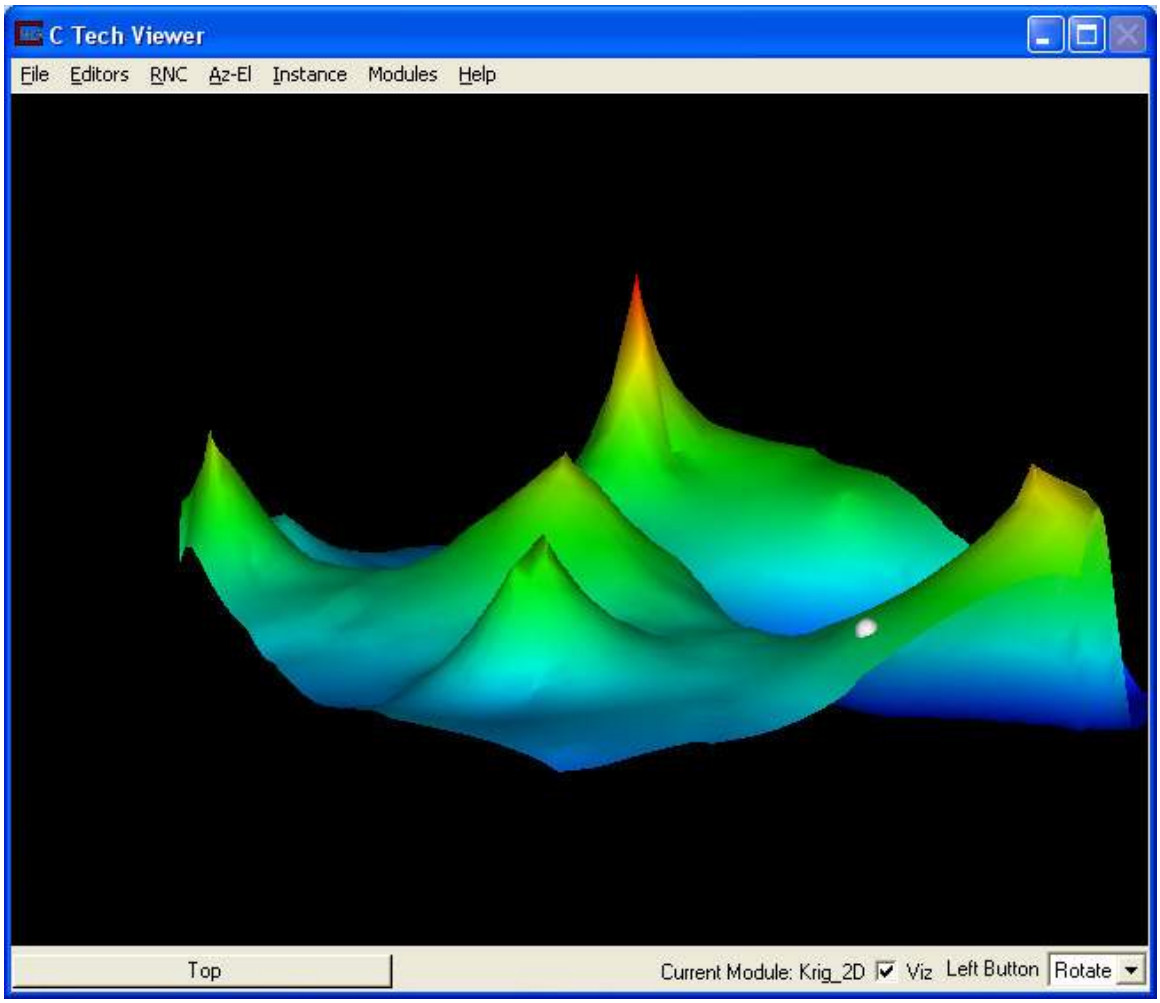
Change your view to match the setting below.



Also, change the Data component back to VOC.

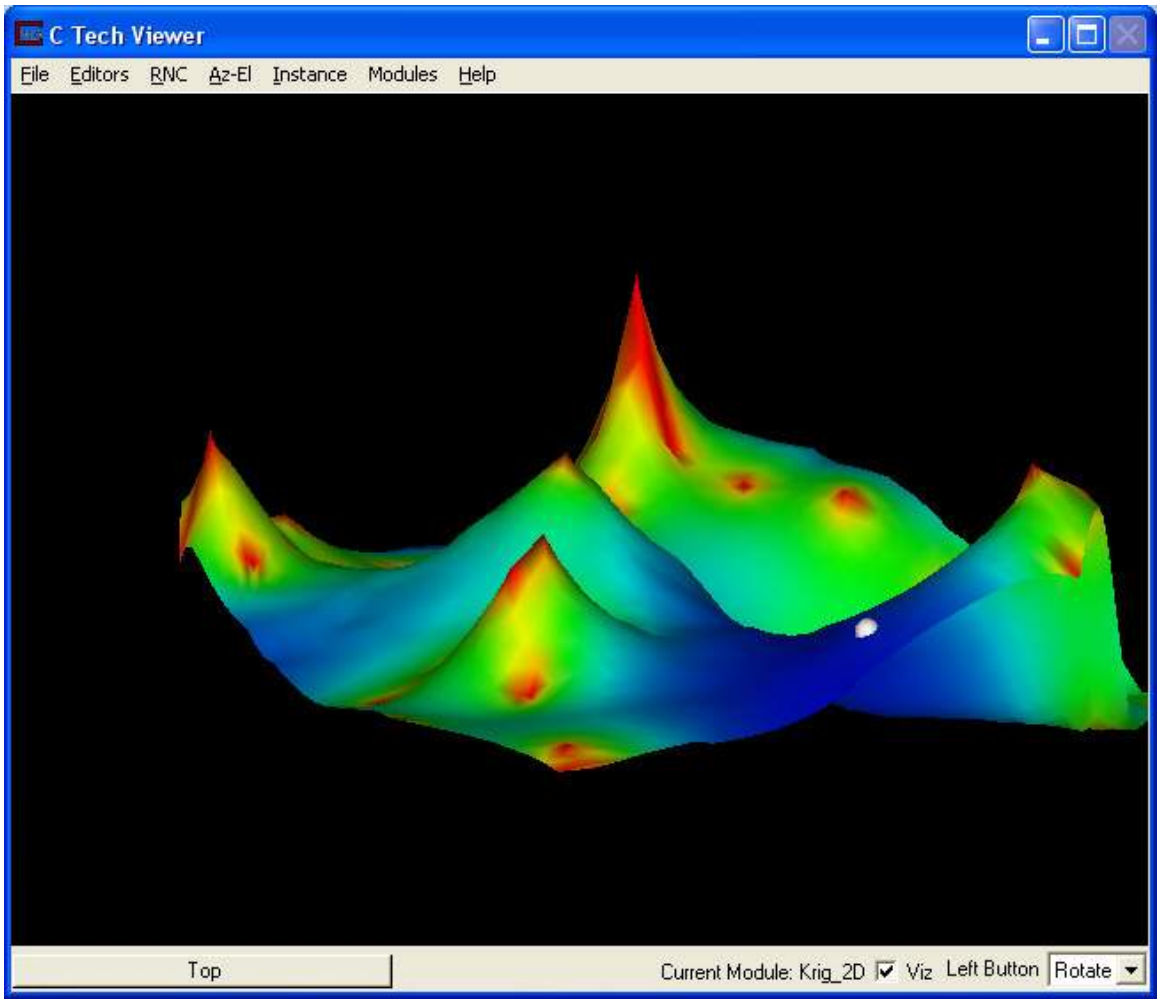


Your viewer should now show:



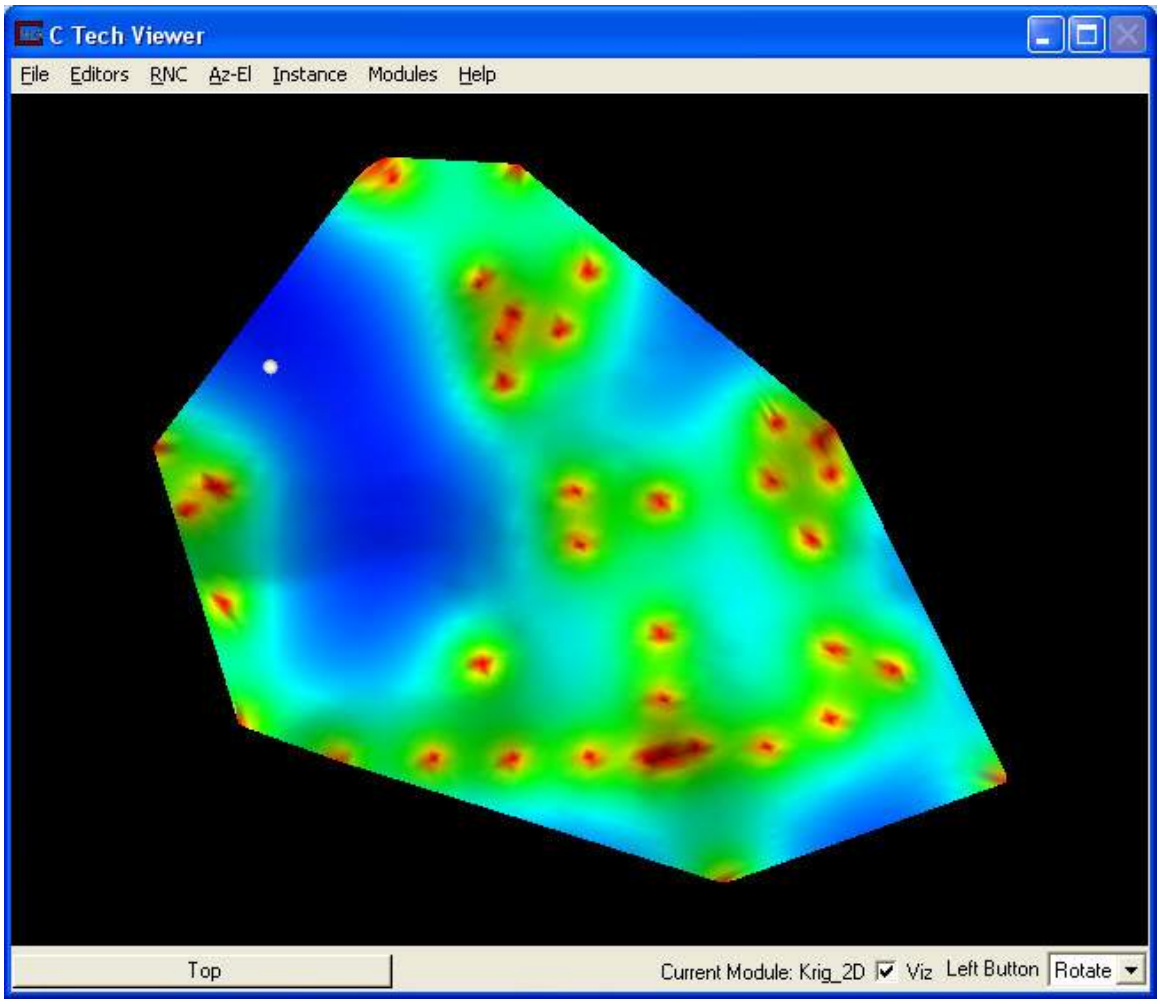
We choose this view because it provides a good look at the location that DrillGuide© has selected for a new sampling event. Note that it lies on the slope of a hill where the concentrations are well above our minimum values.

Switch the Data component to Confidence-VOC to show:



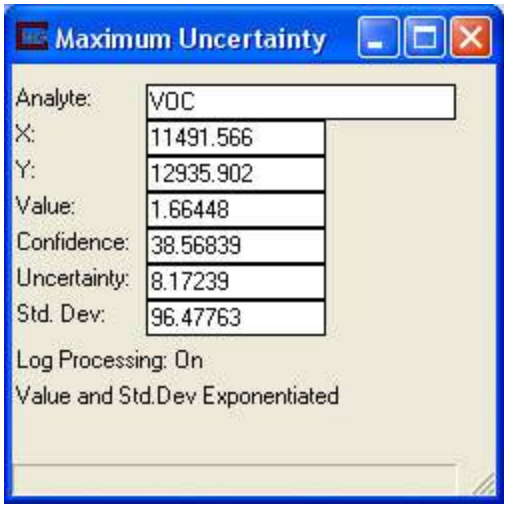
Note that the entire region around the sphere is low confidence, but the lowest values are slightly down slope from the sphere.

Select Top View from the Viewer drop down RNC menu to get the following view.

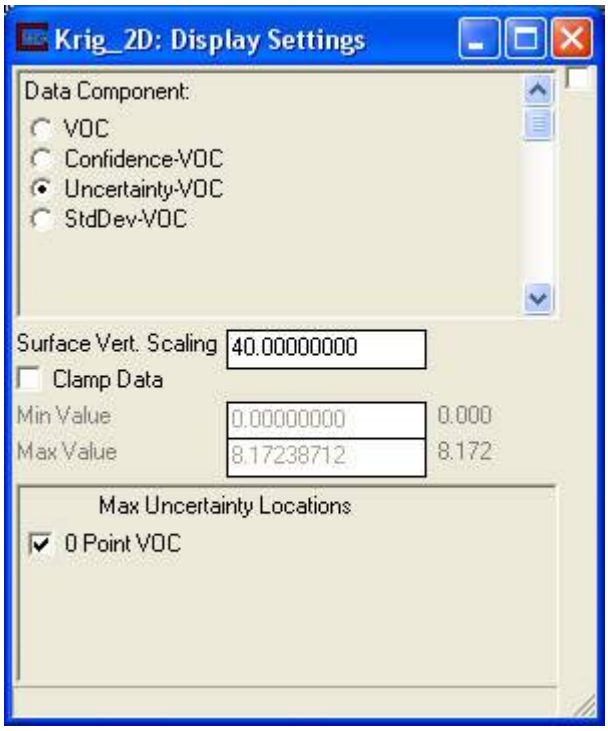


Note that from this view, the sample is also not at the centroid of the lowest confidence region.

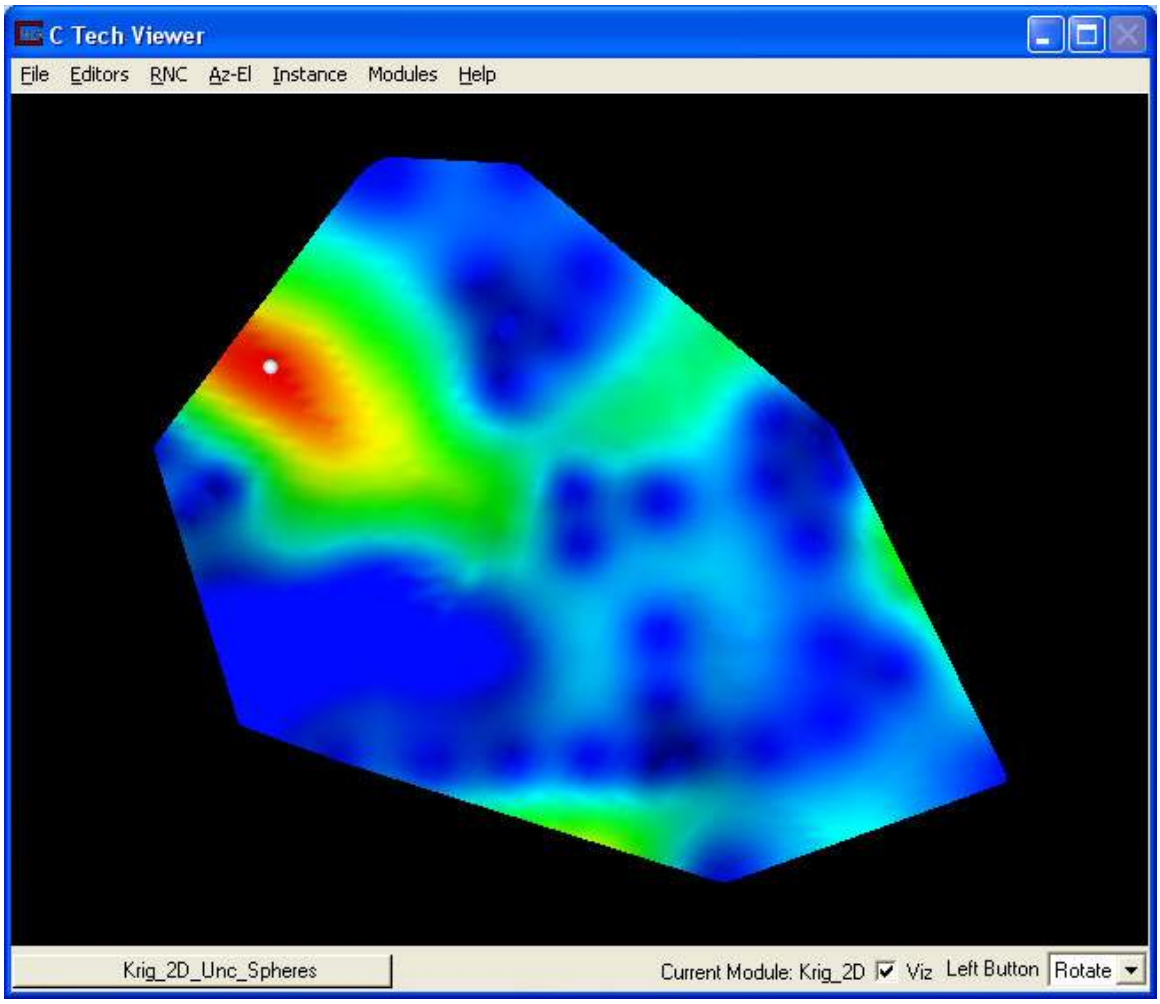
If you hold down the "Alt" key and click exactly on the sphere with the left mouse button, a window will pop-up in the upper right corner of your desktop with detailed information about the Maximum Uncertainty Location.



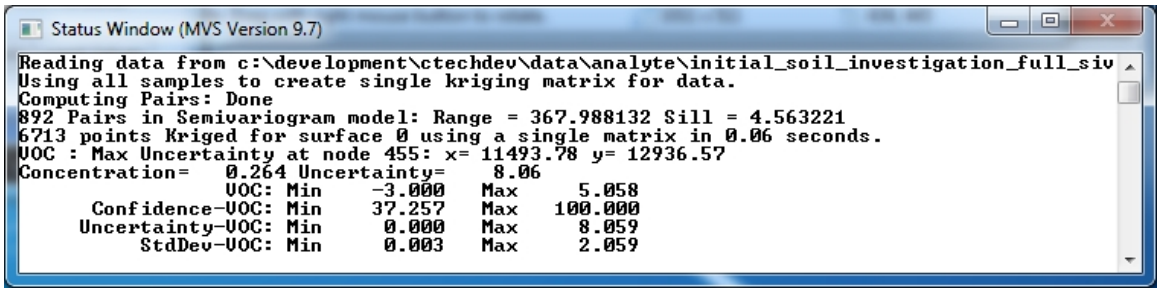
The location that DrillGuide© has selected is the location with the optimal combination of high predicted concentration and low predicted confidence.



Choose the Data component *Uncertainty-VOC* to see the following in your Viewer.



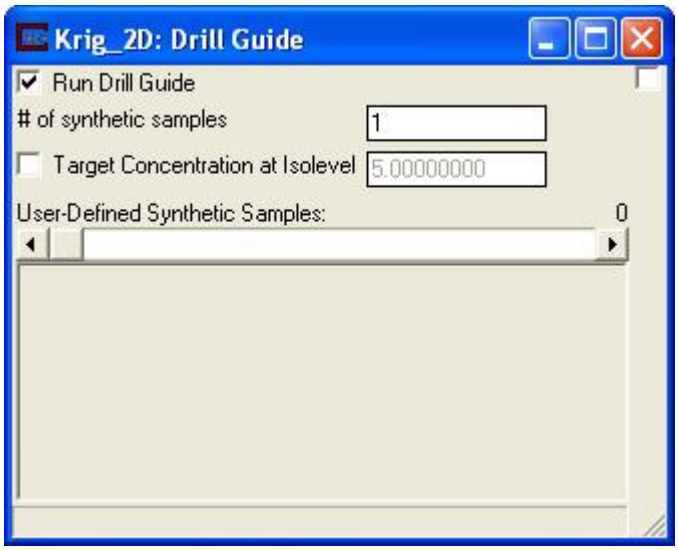
Note that the sphere is located in the centroid of the red region denoting highest uncertainty. Once Krig_2D computes the concentration and confidence at every grid node, it computes the uncertainty there also. It then searches the entire grid to find the node with the highest uncertainty and places the sphere at that location and prints it to the console as shown:



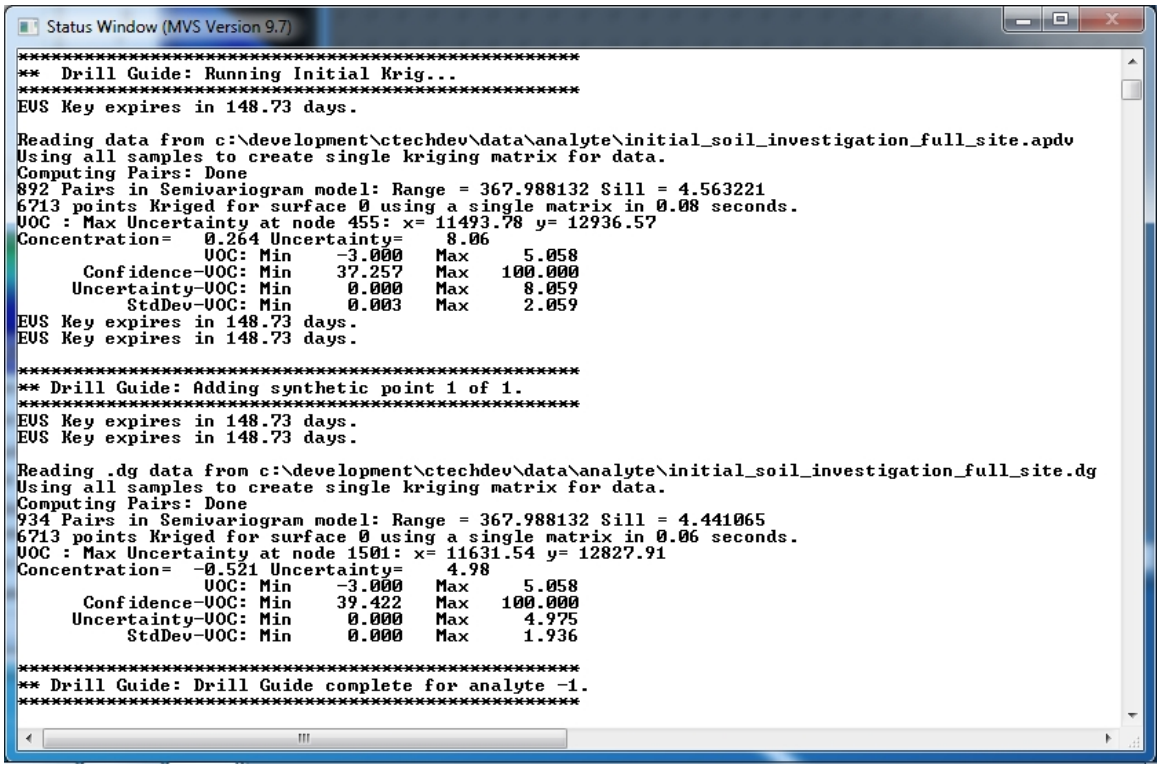
Creating A New Synthetic Sample

In addition to displaying the location of Max Uncertainty, which is the suggested location for a new sampling event, DrillGuide© can create a data file with the synthetic samples added.

We will begin by closing the Display Settings window in Krig_2D and opening the Drill Guide window. Turn on the "Run Drill Guide" toggle as seen in the image below.



Now choose "Accept All Current Values" in the main Krig_2D window. Krig_2D will now run, and display a different set of output from before.

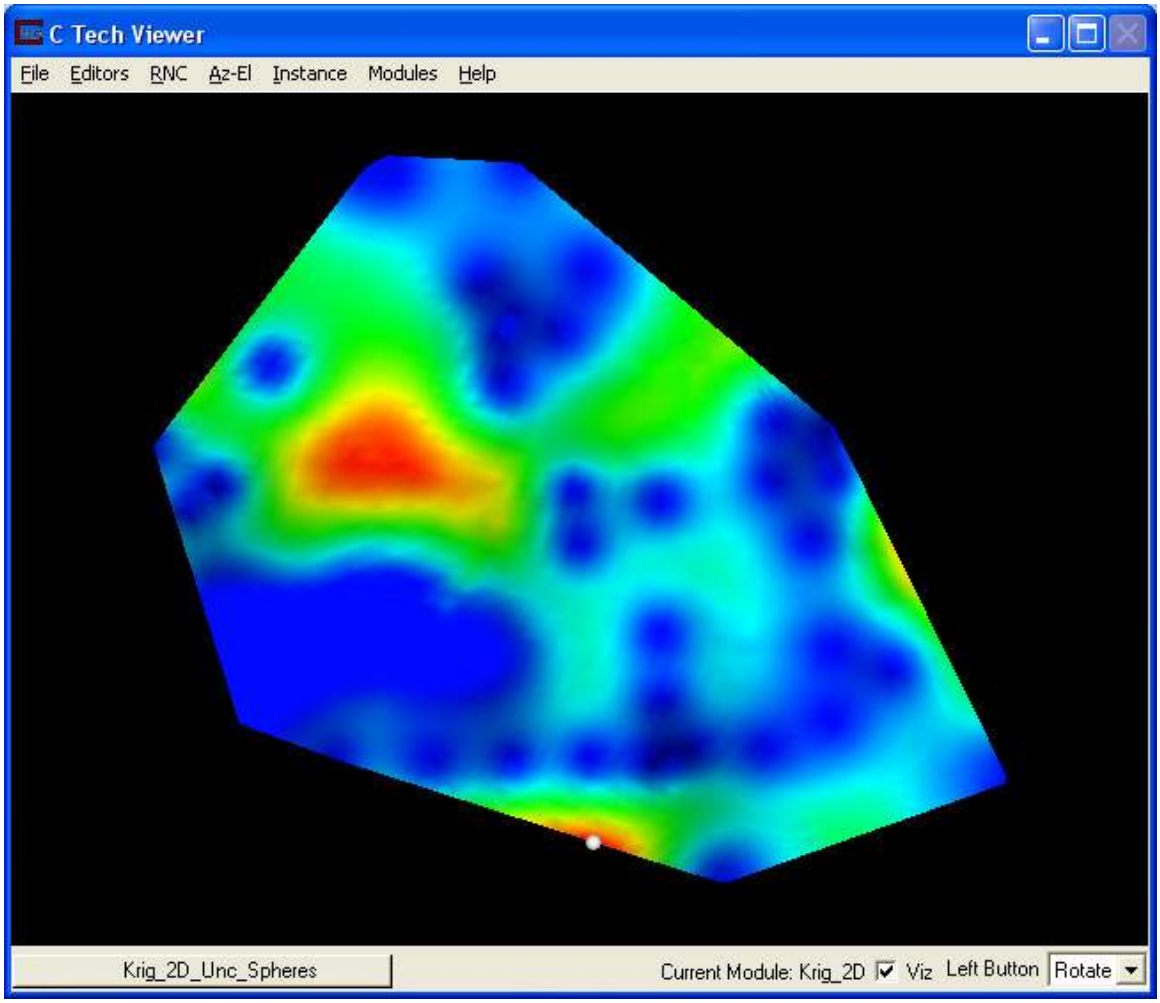


When a drill guide cycle is run, it creates a new data file (*.dg) that contains a synthetic sample. Each cycle adds one new synthetic sample. Below is a list of the file initial_soil_investigation_full_site.dg that was created from this first cycle. Note that the new sample is at the bottom of the file

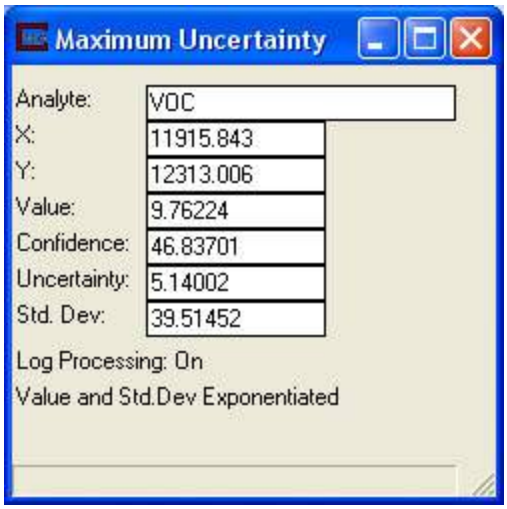
"initial_soil_investigation_full_site.apdv" -1
11491.566406 12935.902344 0.000000 1.664481 "new_1[0]"

This sample has the same x-y coordinate and predicted concentration as the Maximum Uncertainty Location. The z coordinate assigned to 0.0.

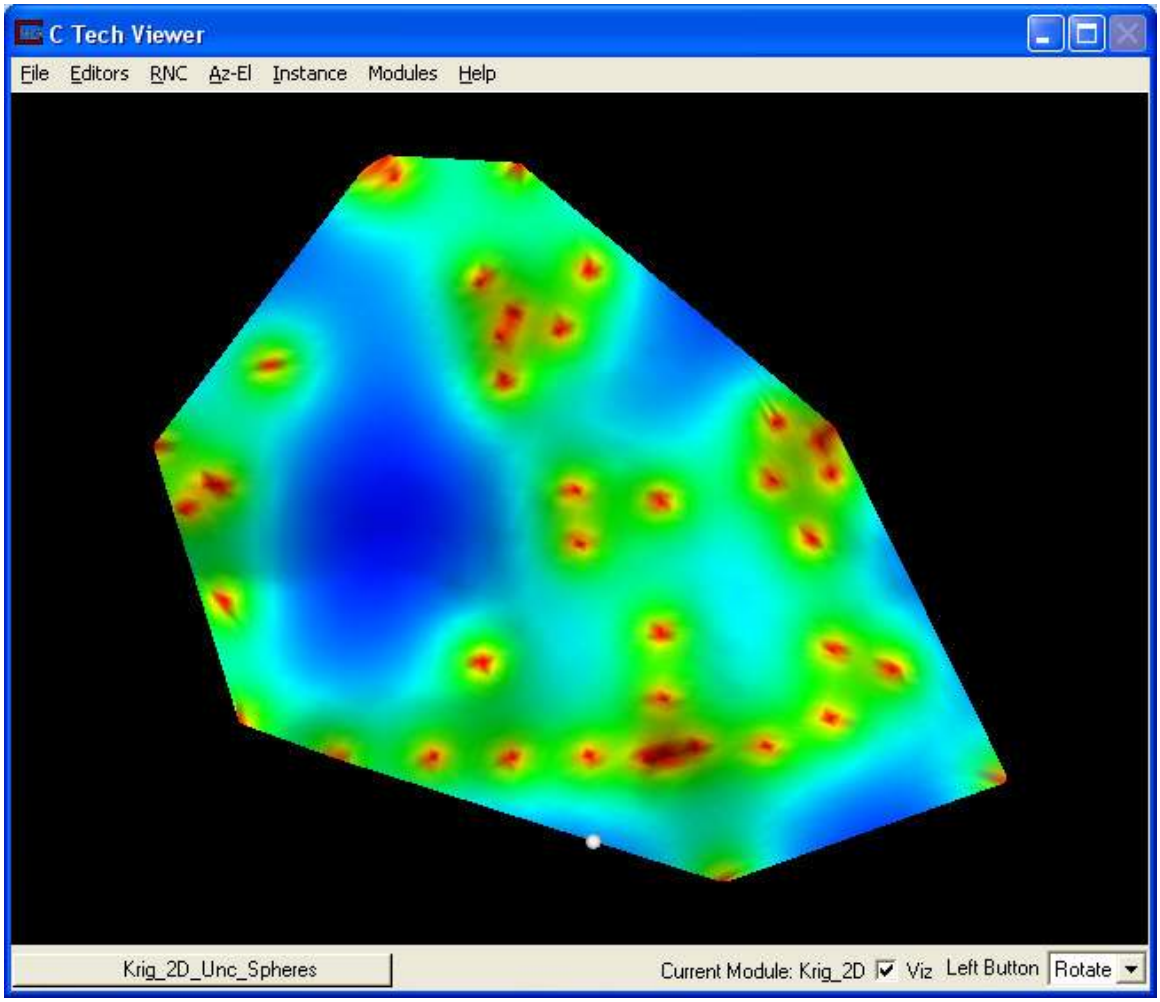
When we ran, notice that the Viewer's picture updated as well. The large red region which used to contain our small grey sphere is now blue, and two new red regions have appeared.



Your Viewer should look like the picture above. Notice now that the new location of Max Uncertainty lies far away from the original location. Probe the sphere with your Alt+Left Mouse Button.



If you switch the Data component in the Display Settings back to confidence, you will be able to clearly see that the confidence is now high in the location where the synthetic sample was located.

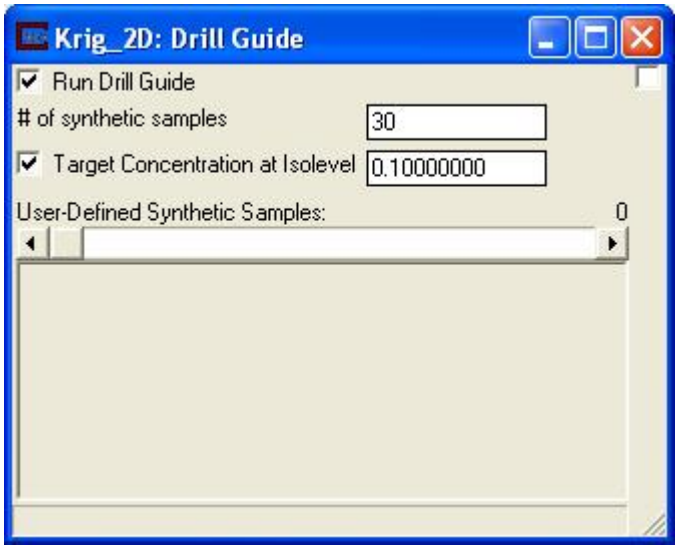


Iterative Analyses

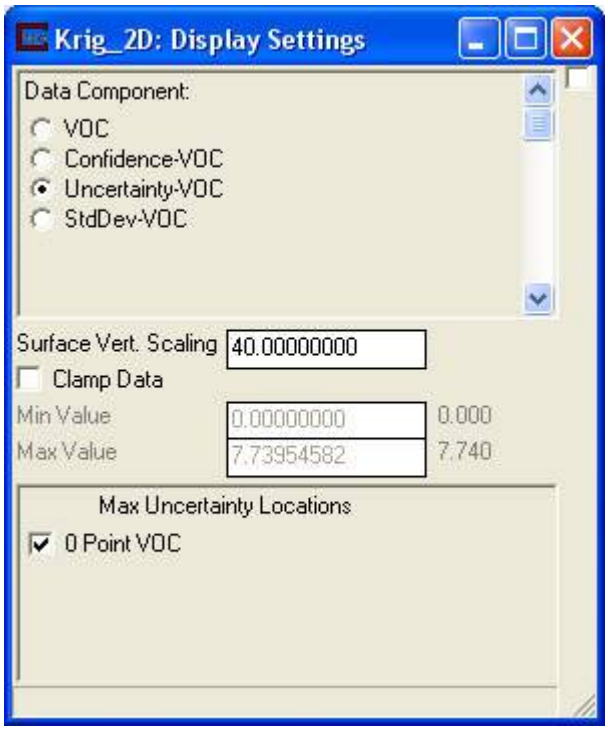
We could continue selecting each new data file that is created and run as many cycles as we wish. However, we have added the ability to run any number of cycles in a fully automated fashion. Open the Drill Guide settings window and change the value of *# of Synthetic Samples* to 30.

This will allow us to automatically cycle (read the new data file and run) until thirty new samples have been created appending each new synthetic sample point to the (*.dg) file.

Also, turn on the *Target Concentration at subsetting level* and change the target subsetting level 0.1. What this does is focus the optimization of sample location selection towards the goal of refining the extent of the 0.1 mg/kg VOC plume. The units are mg/kg since we are dealing with soil contamination data. The value is input in normal units even if you have the Log Processing option selected.

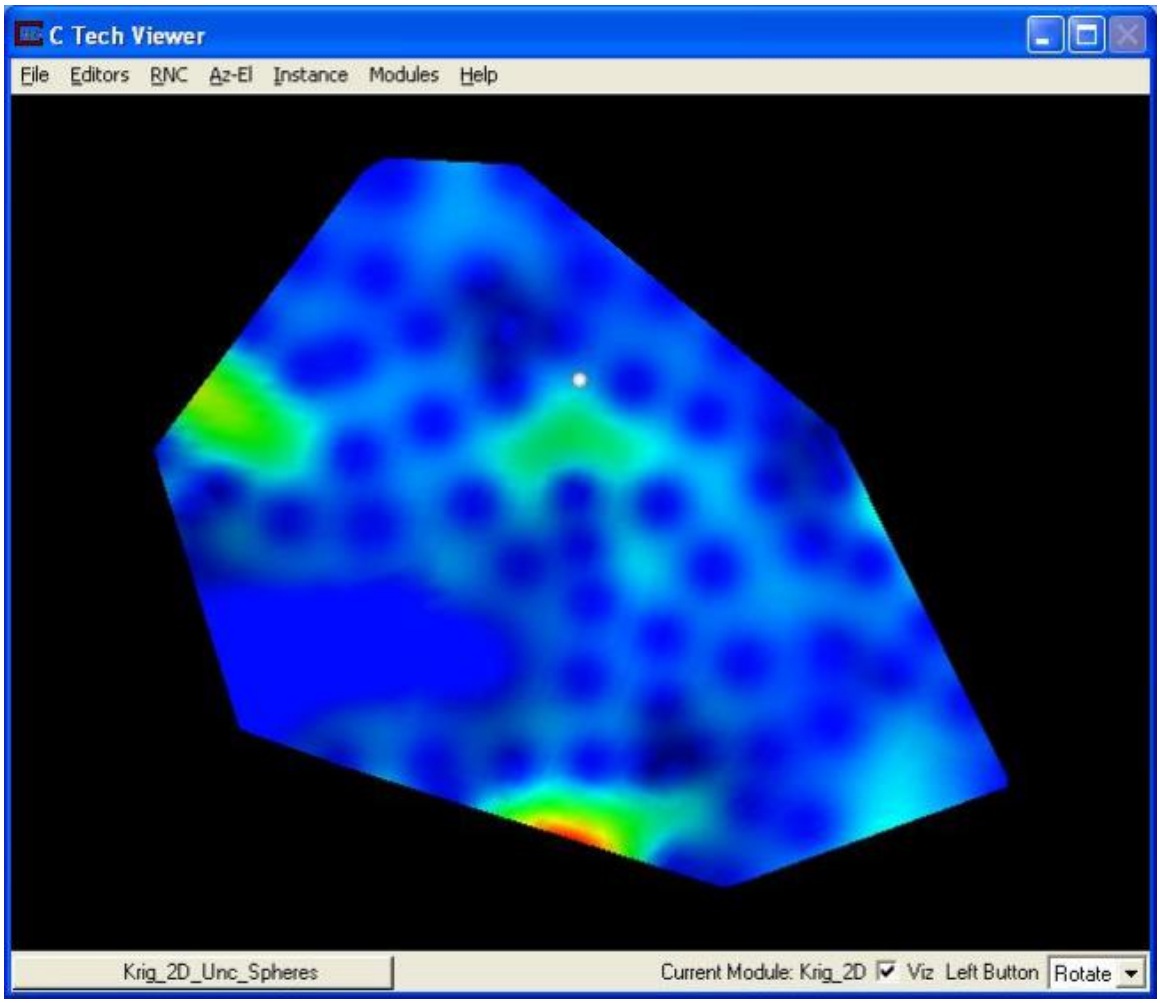


Since the Viewer will update during each cycle, let's select Uncertainty in the Display Settings window so we can see how the Uncertainty distribution changes over time. Remember that during each cycle, the scale will reset to reflect red areas as the current maximum uncertainty value.



Now, click on *Accept All Current Values* again.

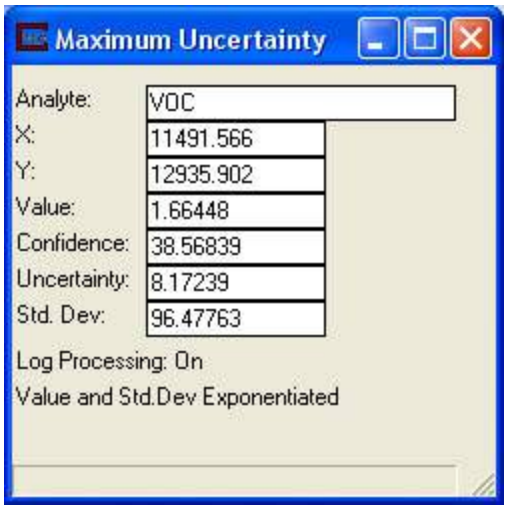
DrillGuide© will run for a few minutes and you should see your Viewer update with each new cycle. Messages will be printed to the console during each cycle, and when it is done you should see:



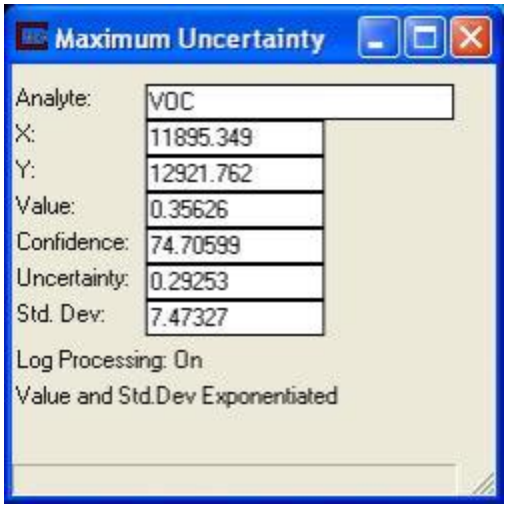
Note that in this figure some of the areas of the site are solid red and yellow. These solid regions of **constant** uncertainty are caused by modifications in the uncertainty algorithm that treat areas where there is high confidence that the concentration is much higher or lower than our target concentration (0.1 mg/kg for this example). In other words, areas where the concentration is much higher or lower than 0.1 are not treated preferentially. This results in the large areas of constant color.

Reduction in Uncertainty

When we ran our first cycle, the maximum uncertainty was over 8.



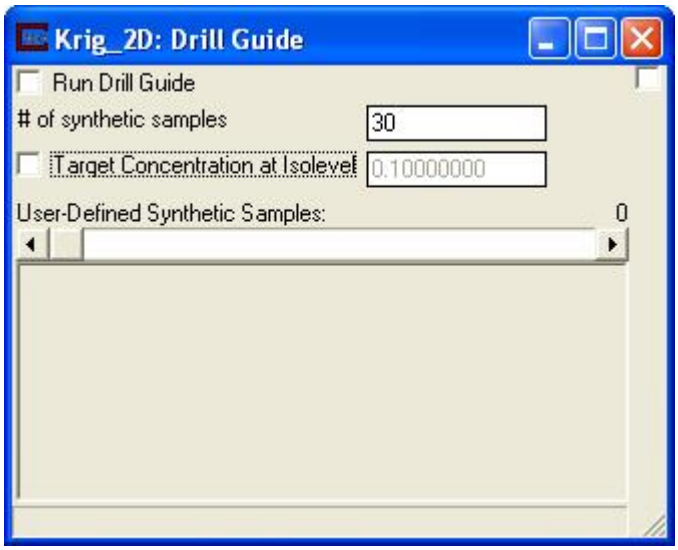
After 31 cycles, our maximum uncertainty is 0.29253, a huge reduction!



Min-Max Plume Technology

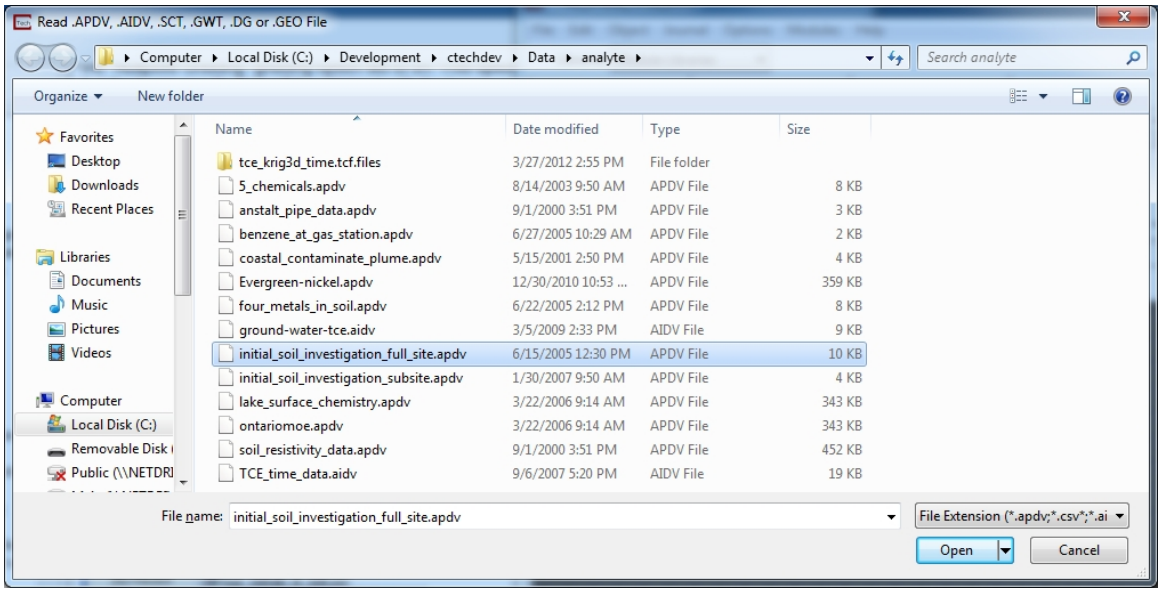
Sometimes confidence and uncertainty just won't answer your (or your customer's) question. Just how big (or small) might the contaminant plume be? EVS displays concentration, but what are the limits? The Min-Max Plume option directly answers those questions.

Before we begin, we will need to change a few settings. First, turn off the *Run Drill Guide* toggle in Krig_2D's Drill Guide window. Also turn off the *Target Concentration at subsetting level* toggle. This will cause Krig_2D to treat uncertainty in the default manner.

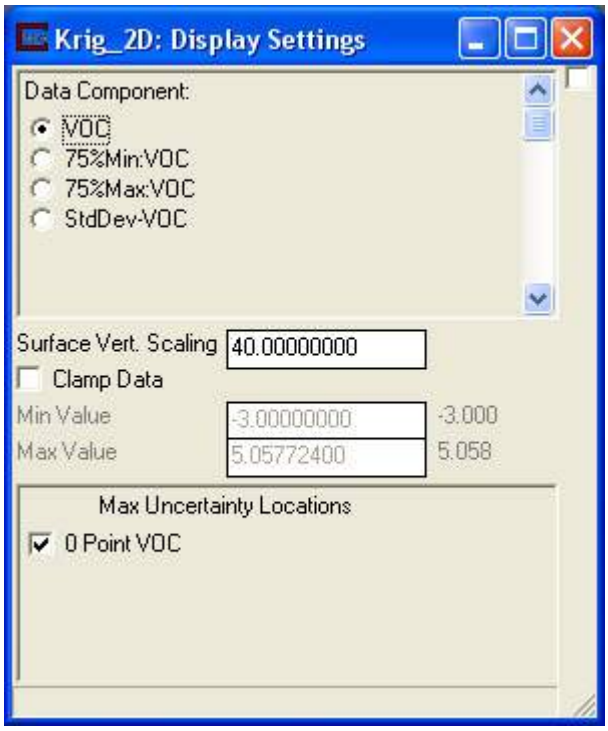


Open up the Kriging Parameters and choose Min-Max Plume for the Interpolation Method. Note that the default Confidence parameter is set to 60 (%). We will change it to 75 (%) for this workbook.

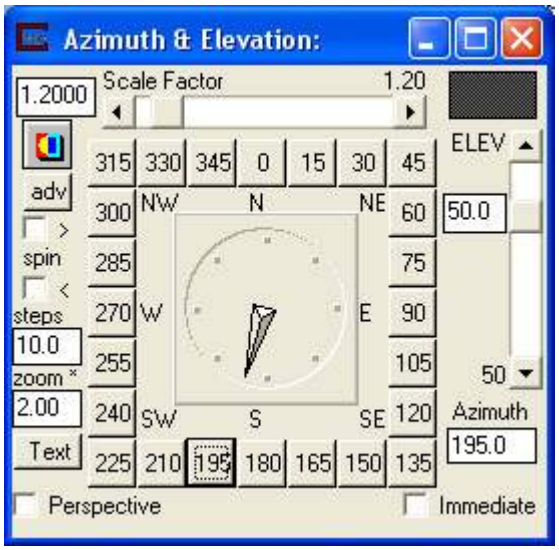
In order to assess how much we've improved things with our 31 Drill Guide cycles, let's go back to the original data file initial_soil_investigation_full_site.apdv.



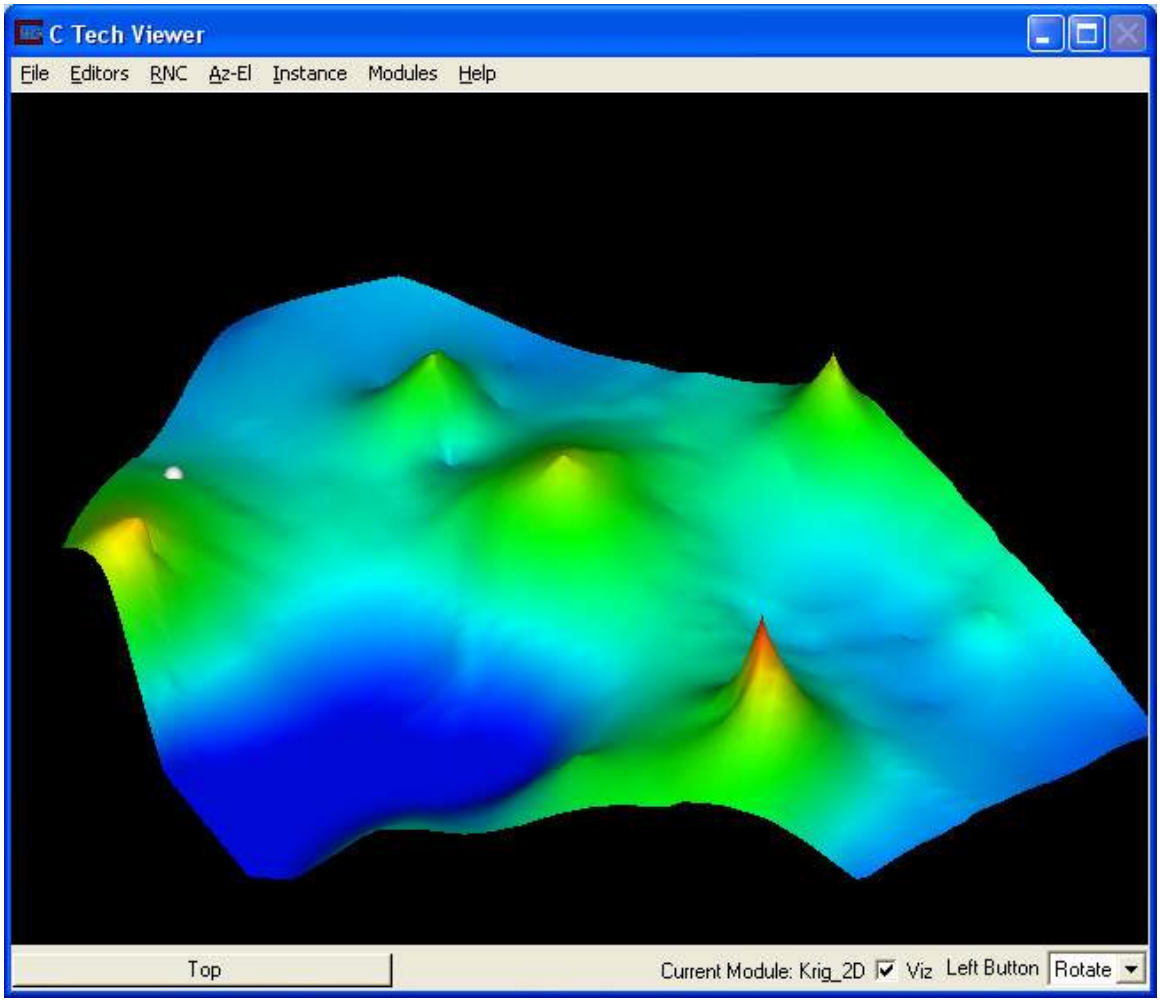
and click on Accept All Current Values again.
Set your Display parameters back to "VOC".



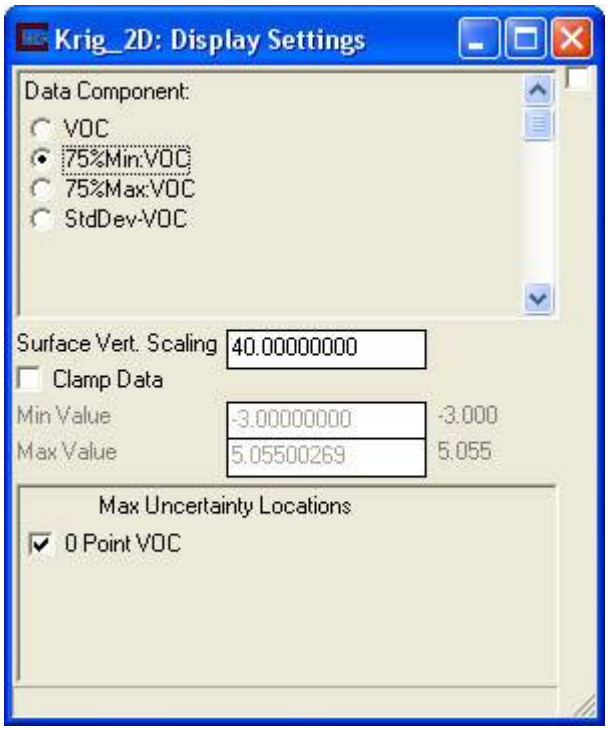
Change your view to match the Az-El setting below.



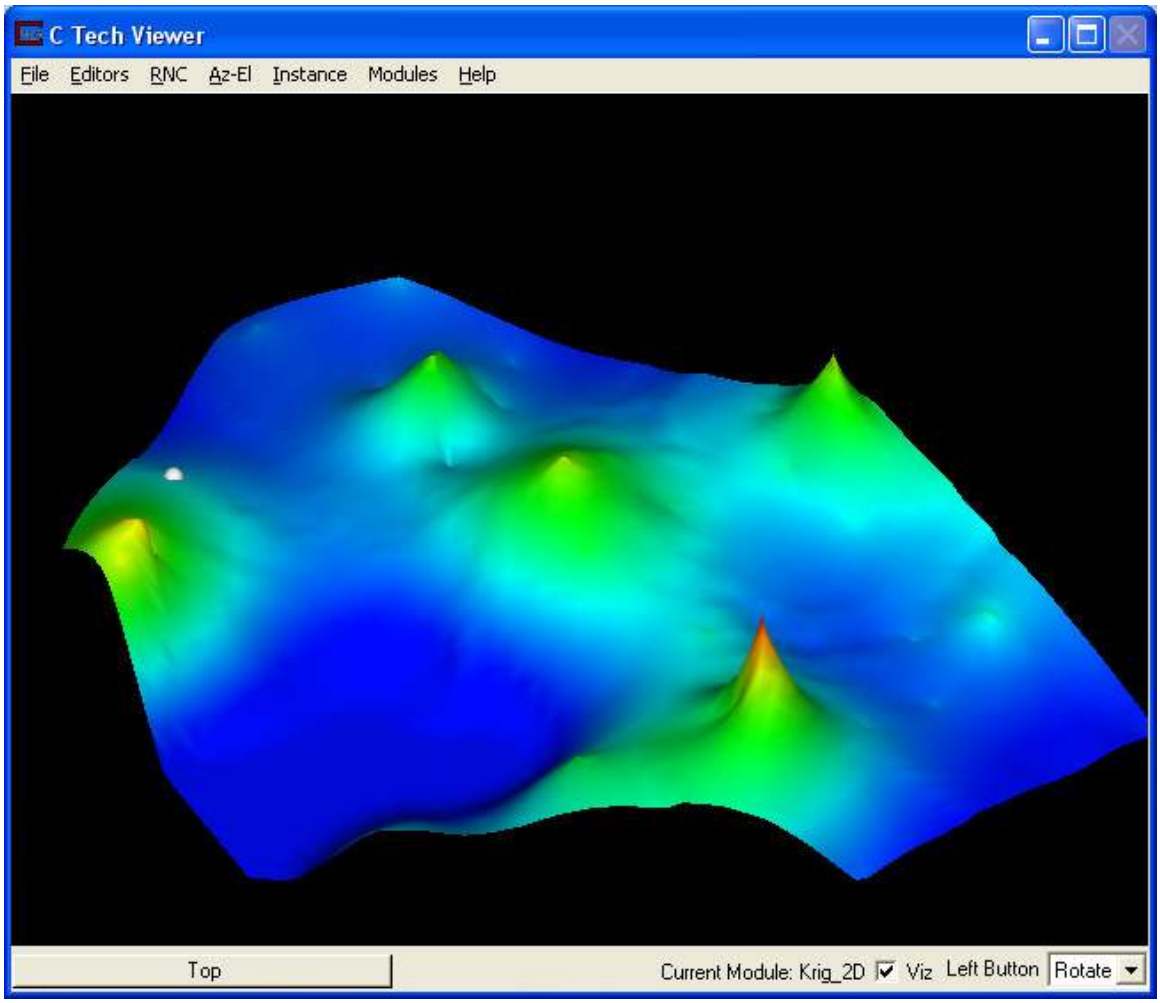
and your Viewer should show the original VOC distributions.



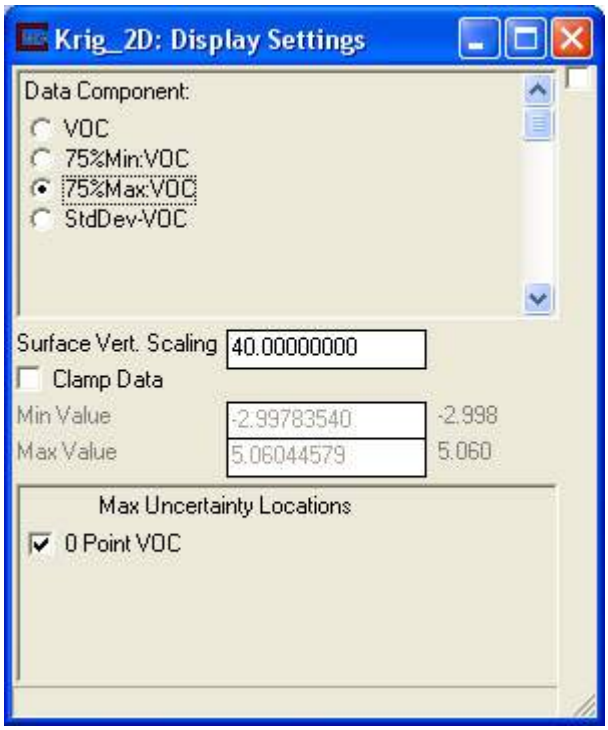
Switching to 75%Min:VOC will display the smallest that plume should be with a 75% confidence.



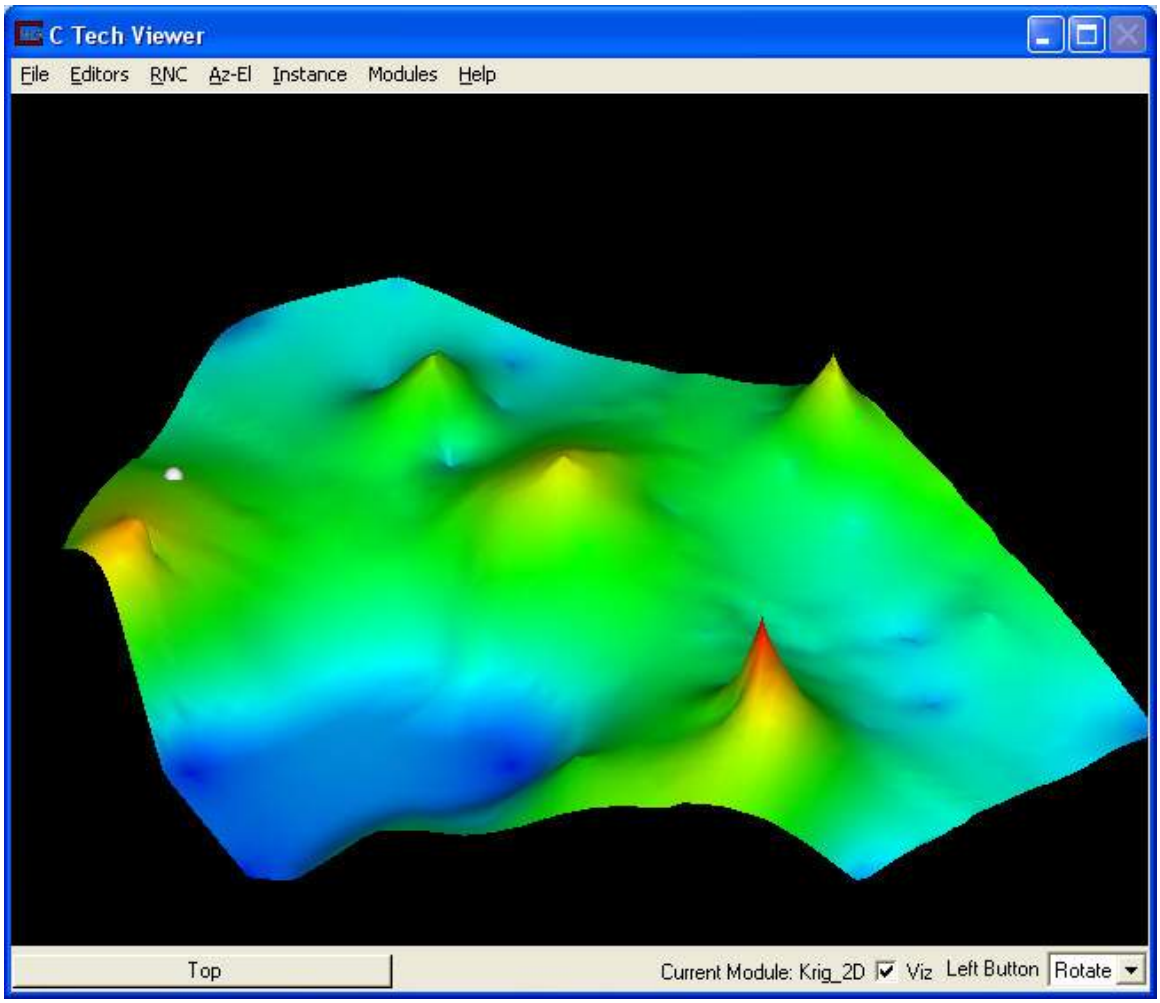
Note that the extent of the light blue and green areas is reduced.



Switching to 75%Max:VOC will display the largest that plume should be with a 75% confidence.



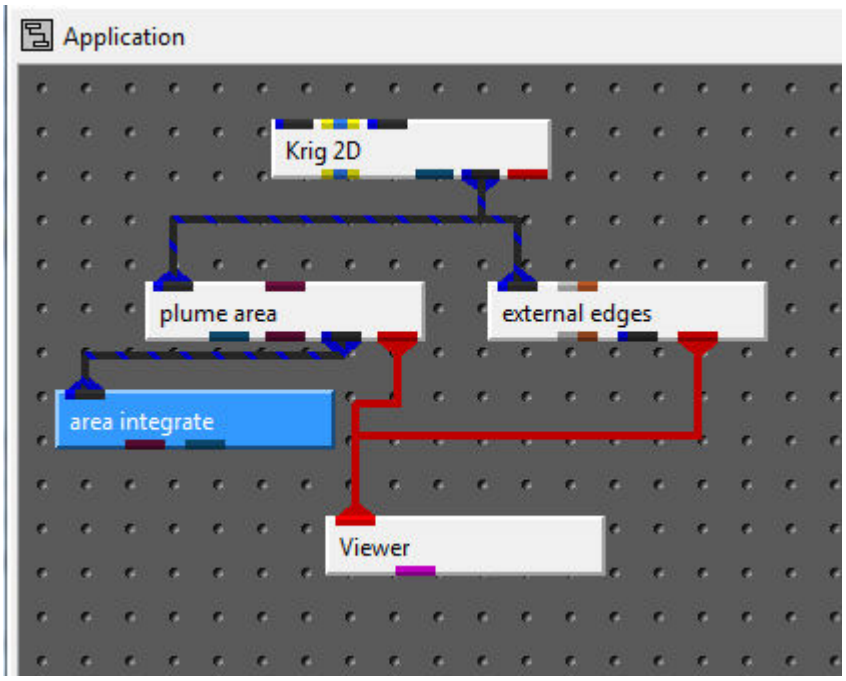
Note that the extent of the light blue and green areas is significantly increased.



It is obvious that there is a significant difference between the Min and Max values, but we should quantify that difference.

Computing Contaminant Area

In order to quantify that difference, we modify our network as shown below.



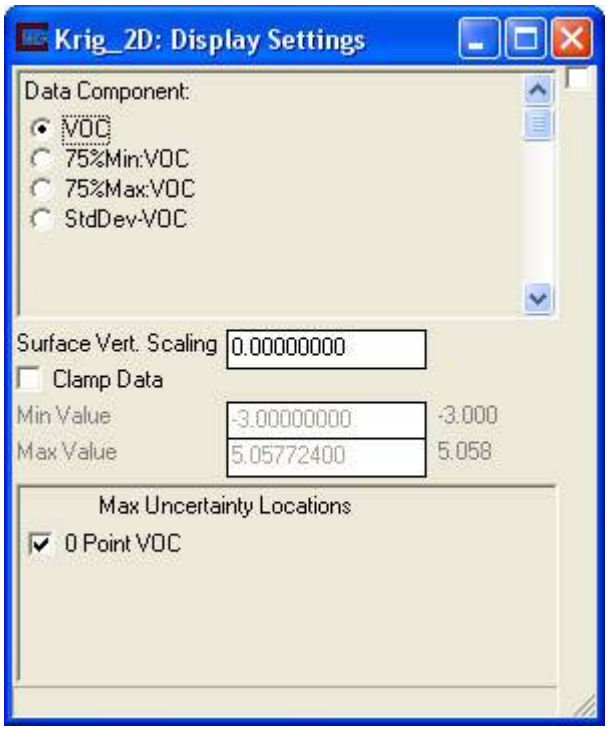
Each new module in the above application plays an important role.

plume_area allows us to subset the domain and display only those portions above (or below) some threshold concentration (the subsetting level).

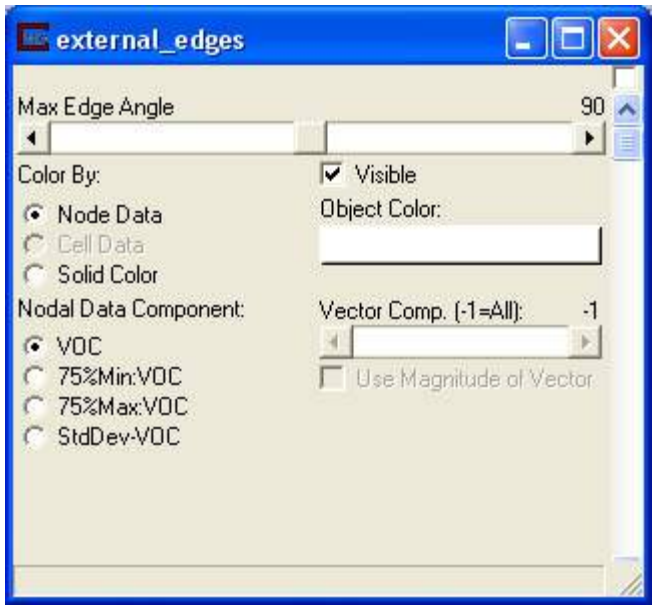
external_edges is somewhat optional, but gives us a display of the outline of the entire extent of the domain.

area_integrate computes the surface area of all portions of the domain meeting the plume_area subsetting criteria.

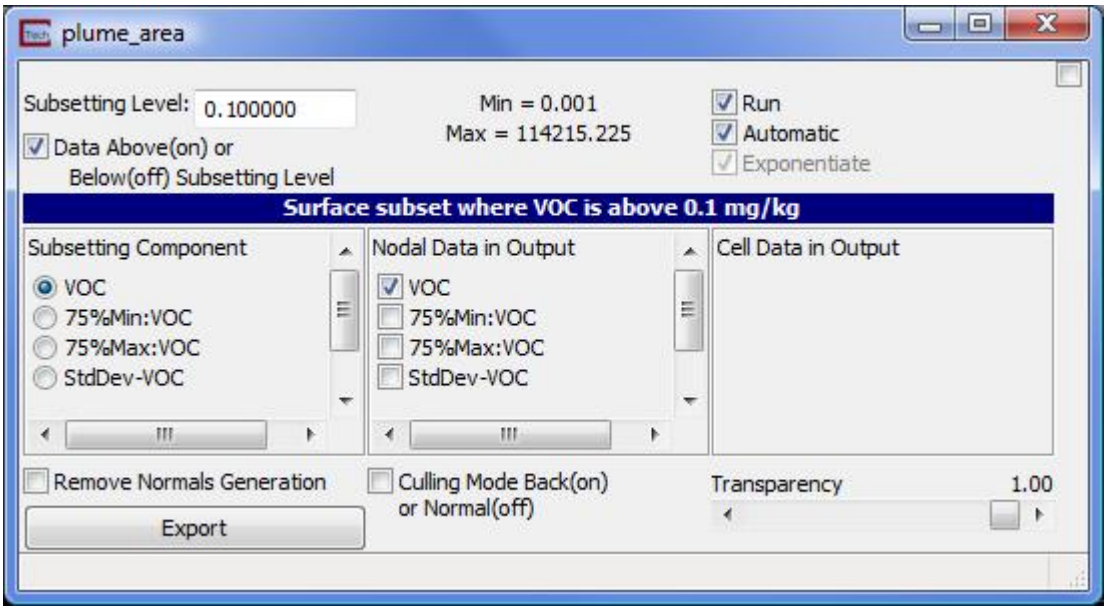
Since area_integrate computes the total surface area, we want to eliminate the *Surface Vert.Scaling*, in the Krig_2D Display Settings window, by setting it to 0.0. Otherwise the computed area will be artificially increased by the topography. We also will go back to viewing our nominal plume by setting the Data Component to VOC.



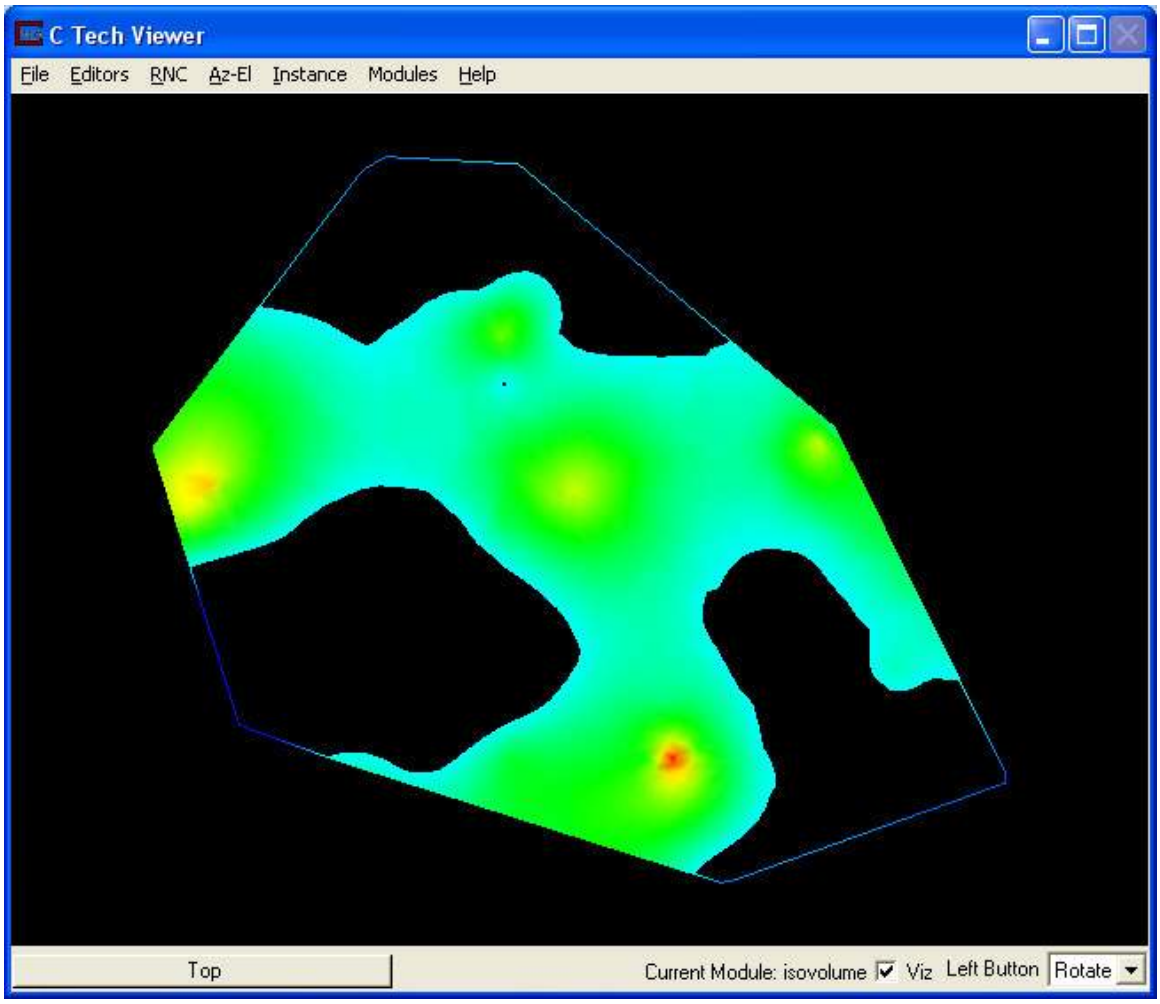
Next, set the external_edges' Max Edge Angle to 90 to show only the outside edges of the domain.



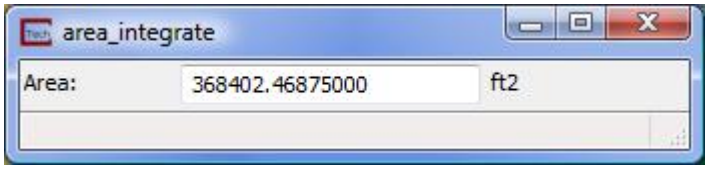
Then, open plume_area and set the Subsetting Level to 0.1 ppm, with the other settings as shown below.



Since we are looking at a flat 2D display, let's go back to a top view by selecting Top View from the RNC drop down menu on the Viewer. This will give us a display of the Nominal contaminated area for initial_soil_investigation_full_site.apdv.



This area (as shown below) is 368,402 sq. ft.



Determining Adequate Characterization

Our original data file initial_soil_investigation_full_site.apdv has 273 samples from 43 borings. This is a very under characterized site, especially considering the extremely high spatial variation in data. There are locations on this site where concentrations vary by orders of magnitude in just a few feet. Even when we take the average concentration over each boring, the variation is extreme. The table below shows a few borings and computes the distance between them and shows how dramatically the concentrations vary.

X	Y	Average-VOC	Boring	VOC-Ratio	Distance
11983.5	12423.6	3.586	CSB-7		
12007.8	12430.6	686.1	CSB-39	191.3	25.29

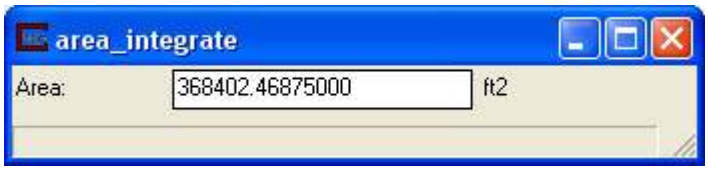
12019.8	12422.9	37307.8	CSW-10	54.4	14.25
12004.8	12499.4	0.237	CSB-9		
12007.8	12430.6	686.1	CSB-39	2895.0	68.86

In the table above, *VOC-Ratio* is the ratio of the concentration in that row divided by the concentration in the row above. Similarly the *Distance* is the distance in feet between those borings. What this shows is that moving away from CSW-10, in only 14 feet, the concentration changed by a factor 54. Only 25 feet from CSB-39 it dropped by a factor of nearly 200 and moving in another direction towards CSB-9 it dropped by a factor of nearly 3,000!

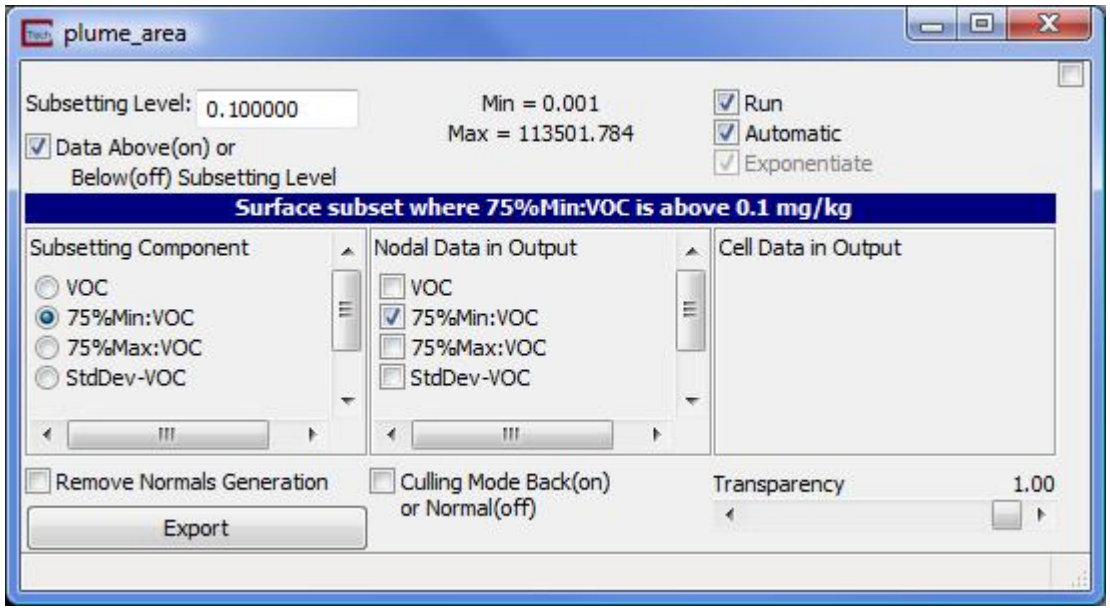
We can estimate the average spacing between our borings by taking the square root of: the total domain area divided by the number of borings. This data set has borings that are clustered, so we know in advance that this is only an estimate of average spacing.

Our total site area is 669,509 square feet. Given that our initial data file had 43 borings that would be an average spacing of about 125 feet. However, our table above shows that in about half that distance we could have a VOC variation of nearly 3,000 mg/kg!

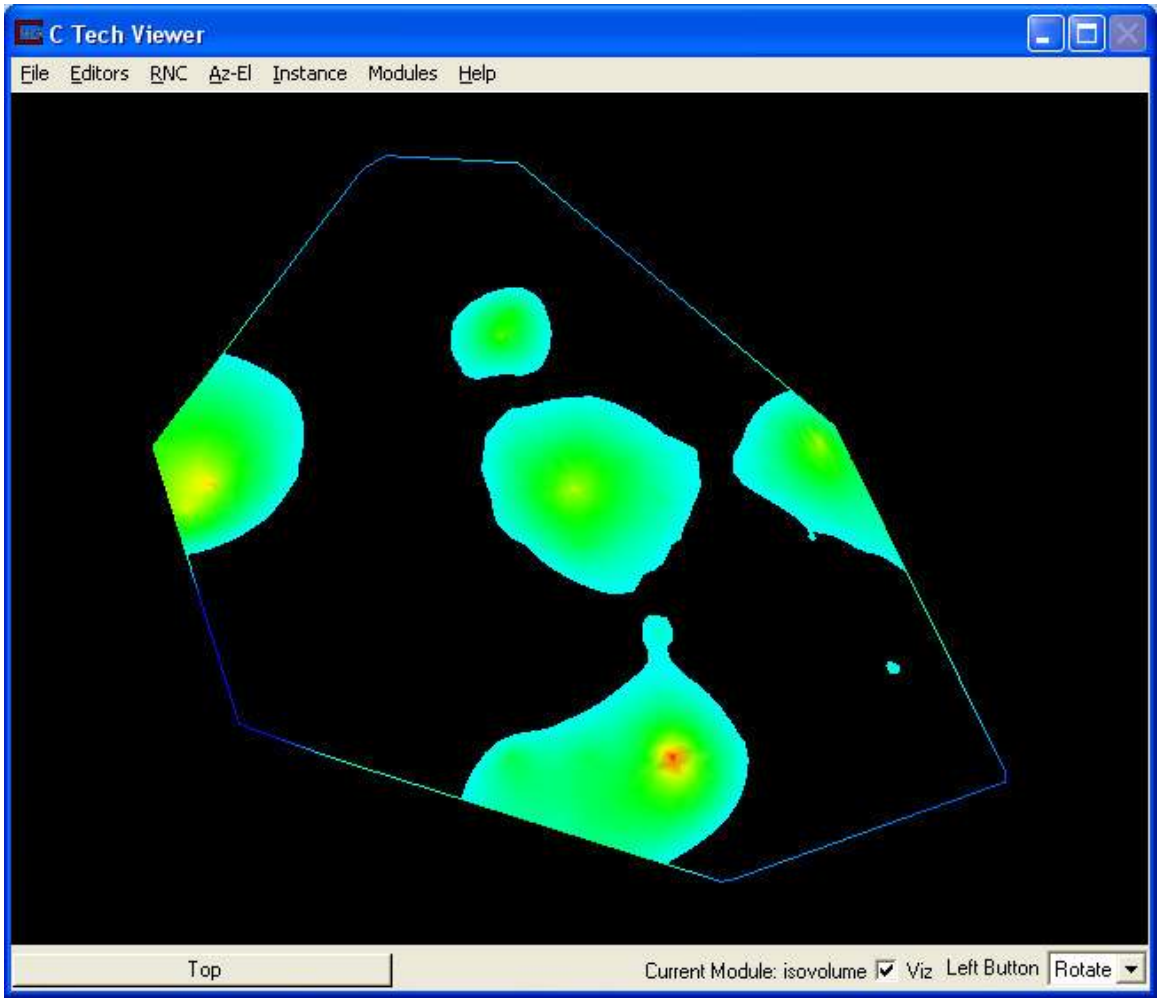
If you open the integrate_area window you will see that the nominal area contaminated above 0.1 ppm is 368,402 sq. ft.



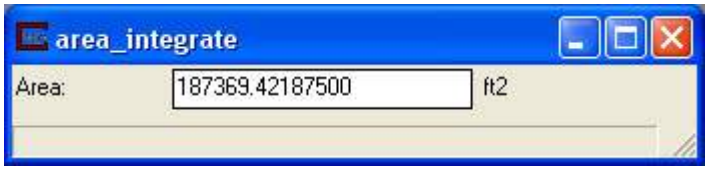
If we switch plume_area to compute the minimum expected area



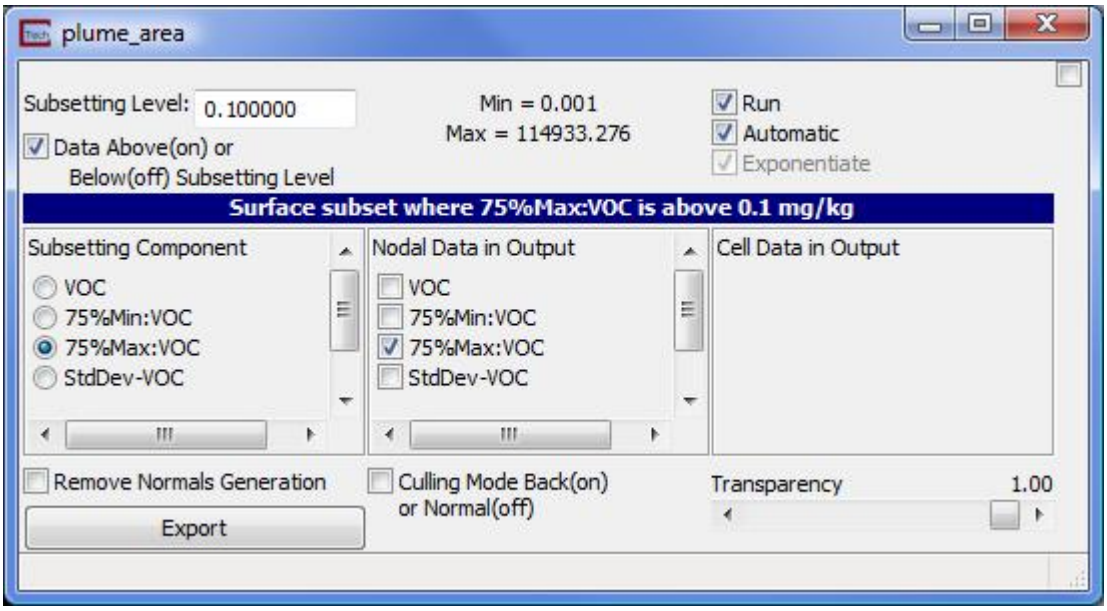
our display changes to:



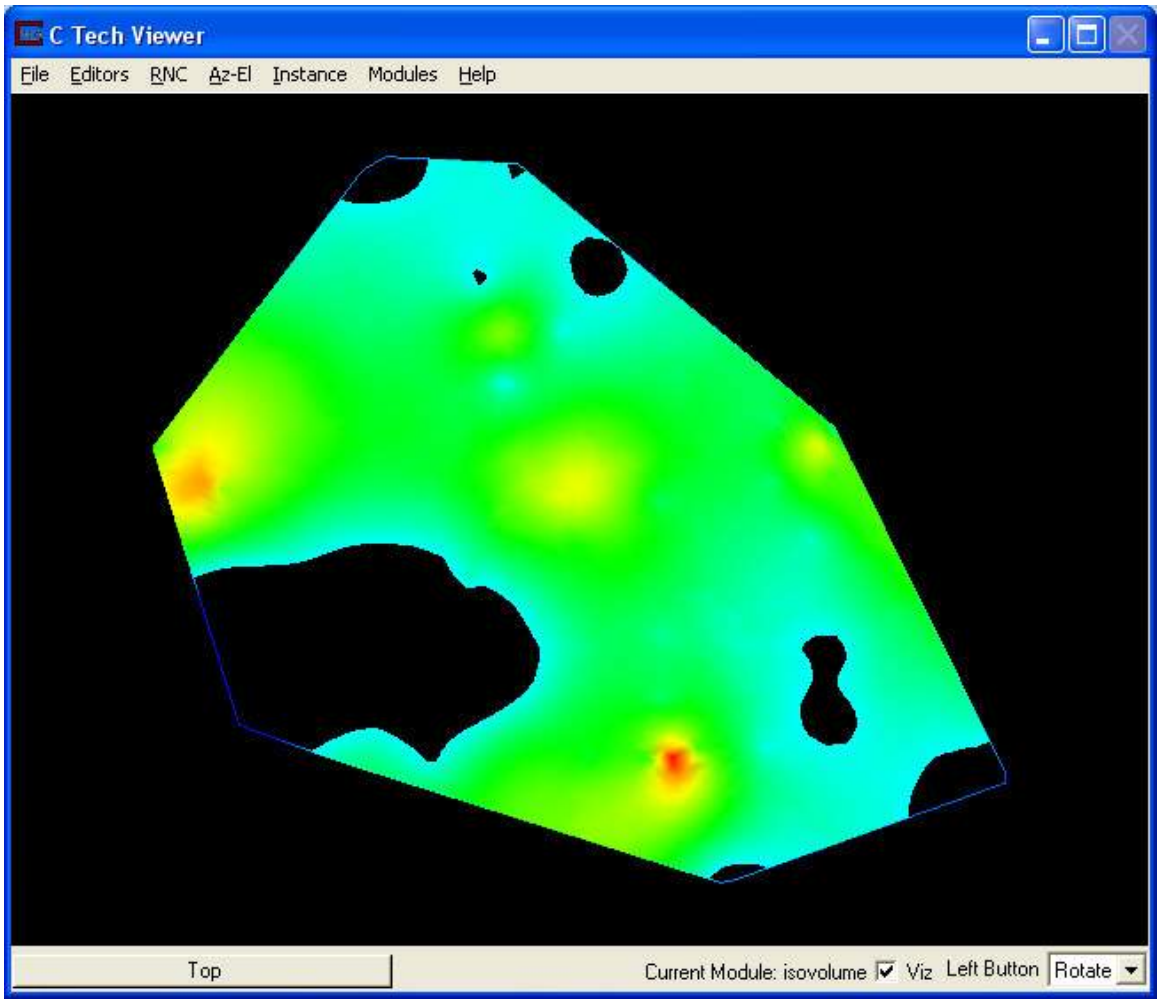
and the computed area is down to 187,369 sq. ft. (about half of the original!).



Similarly, our maximum expected area can be determined.



The maximum area display shows a markedly greater contaminated area.



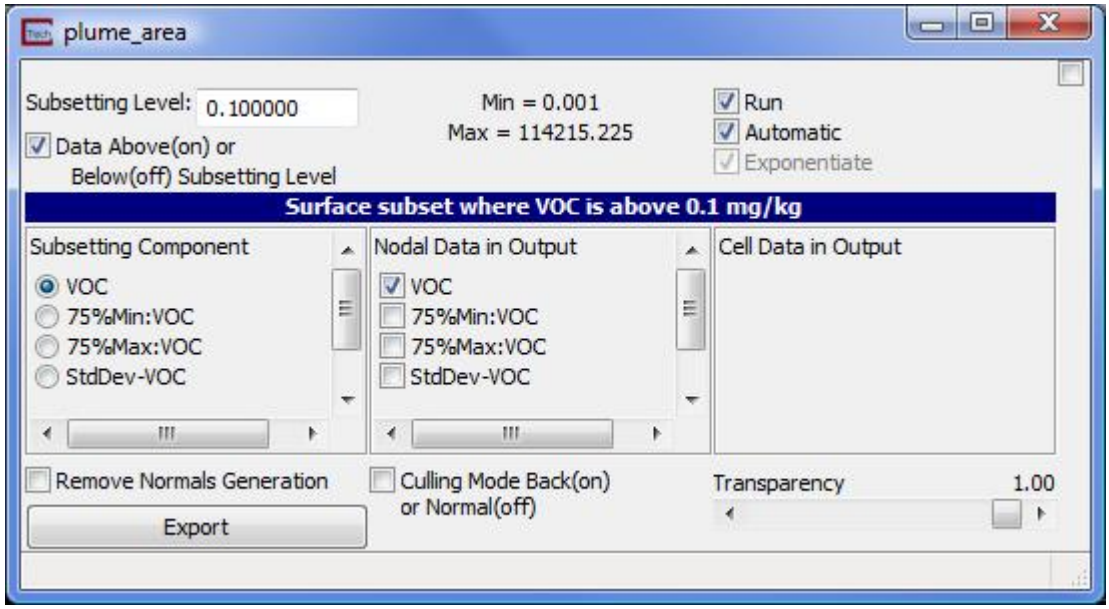
The maximum area is 553,695 square feet.

The table below computes some simple measures of the quality of the site assessment for the initial case. Note that the ratio between the maximum and minimum expected area is nearly 3.0.

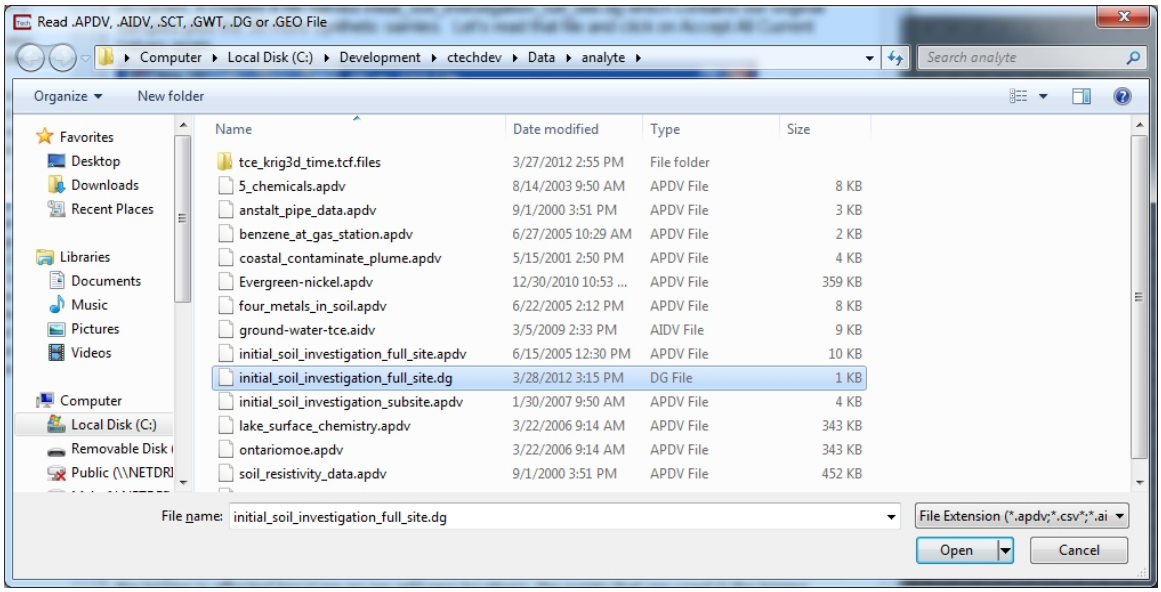
Nominal area	368402	Deviation
Minimum area	187369	-49.1%
Maximum area	553695	50.3%
Max / Min area ratio	2.96	

Note: for the calculations above and later in this topic, the normalized Deviation is computed as Delta / Nominal, where Delta is (Minimum - Nominal) or (Maximum – Nominal).

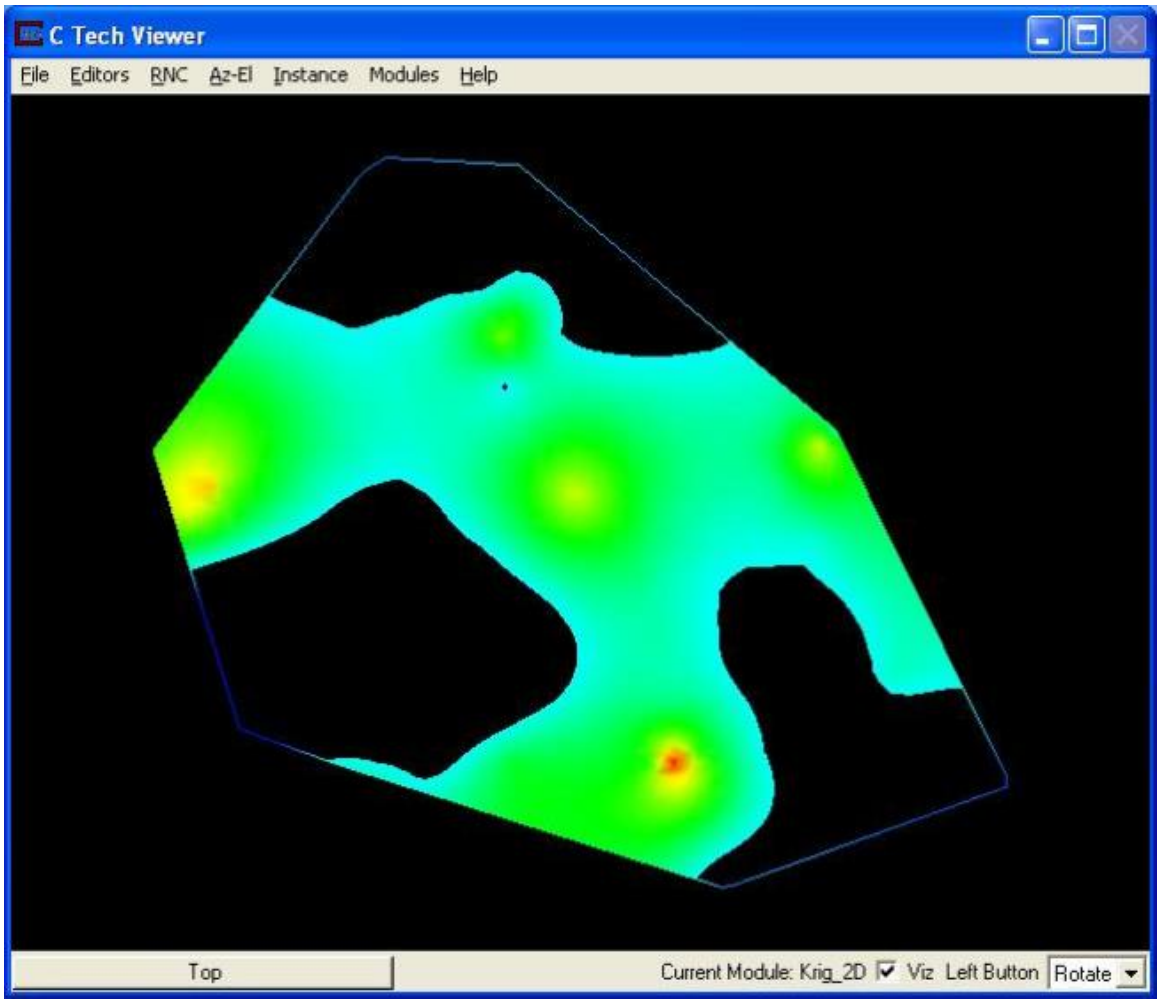
First, let's set plume_area to go back to viewing our nominal plume.

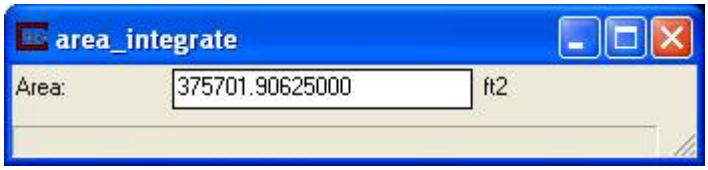


Now let's analyze the quality of fit after our 30 cycles. Since our last DrillGuide© run was set to run for 30 cycles, it created a file named initial_soil_investigation_full_site.dg which contains our original samples plus the 30 more synthetic samles. Let's read that file and click on Accept All Current Values again.



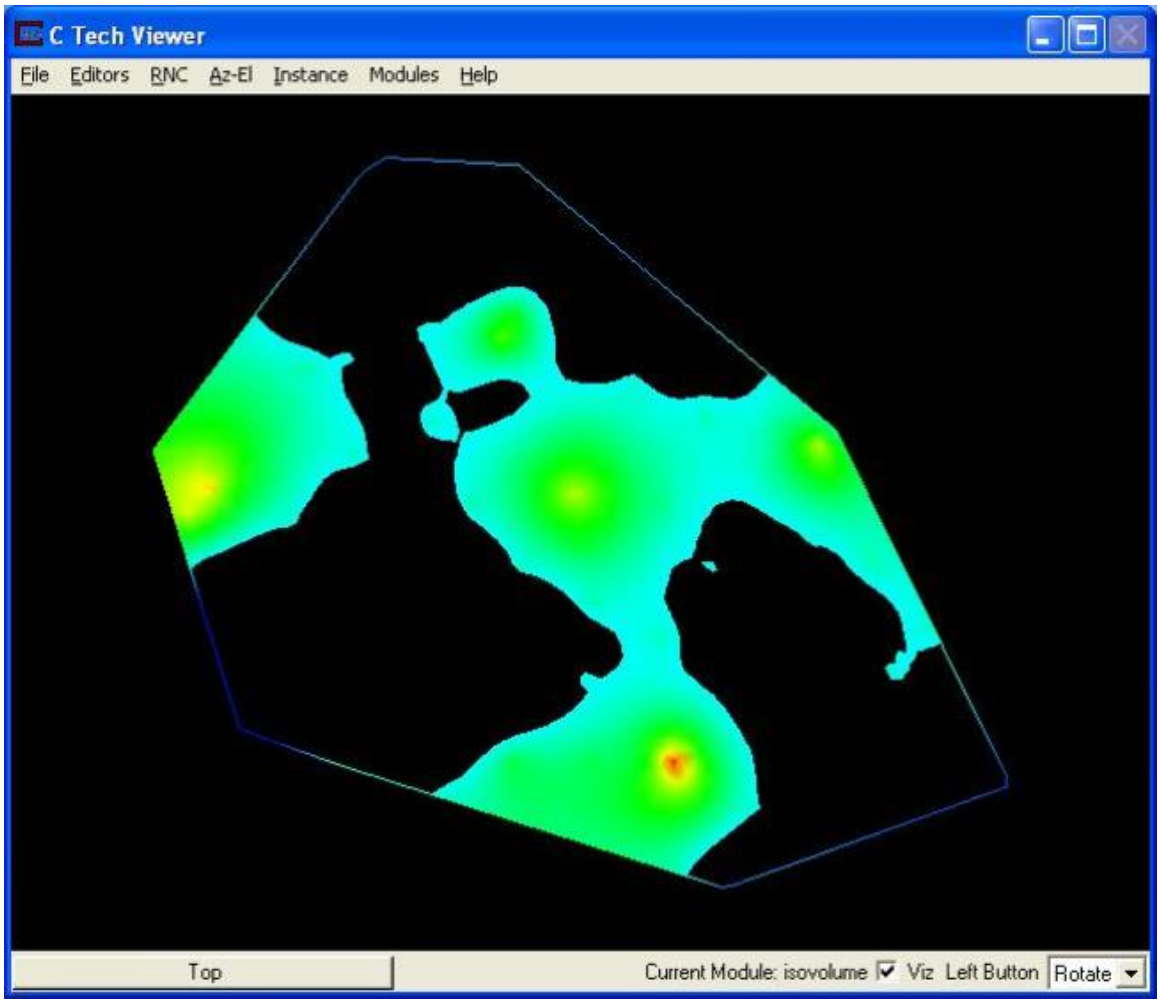
Note that the general distribution has not changed dramatically, but the nominal area has increased. Although each predicted (synthetic) sample has a concentration that matches the predicted values, the kriging is affected because as we add new locations, the points that are used in the kriging algorithm are affected.



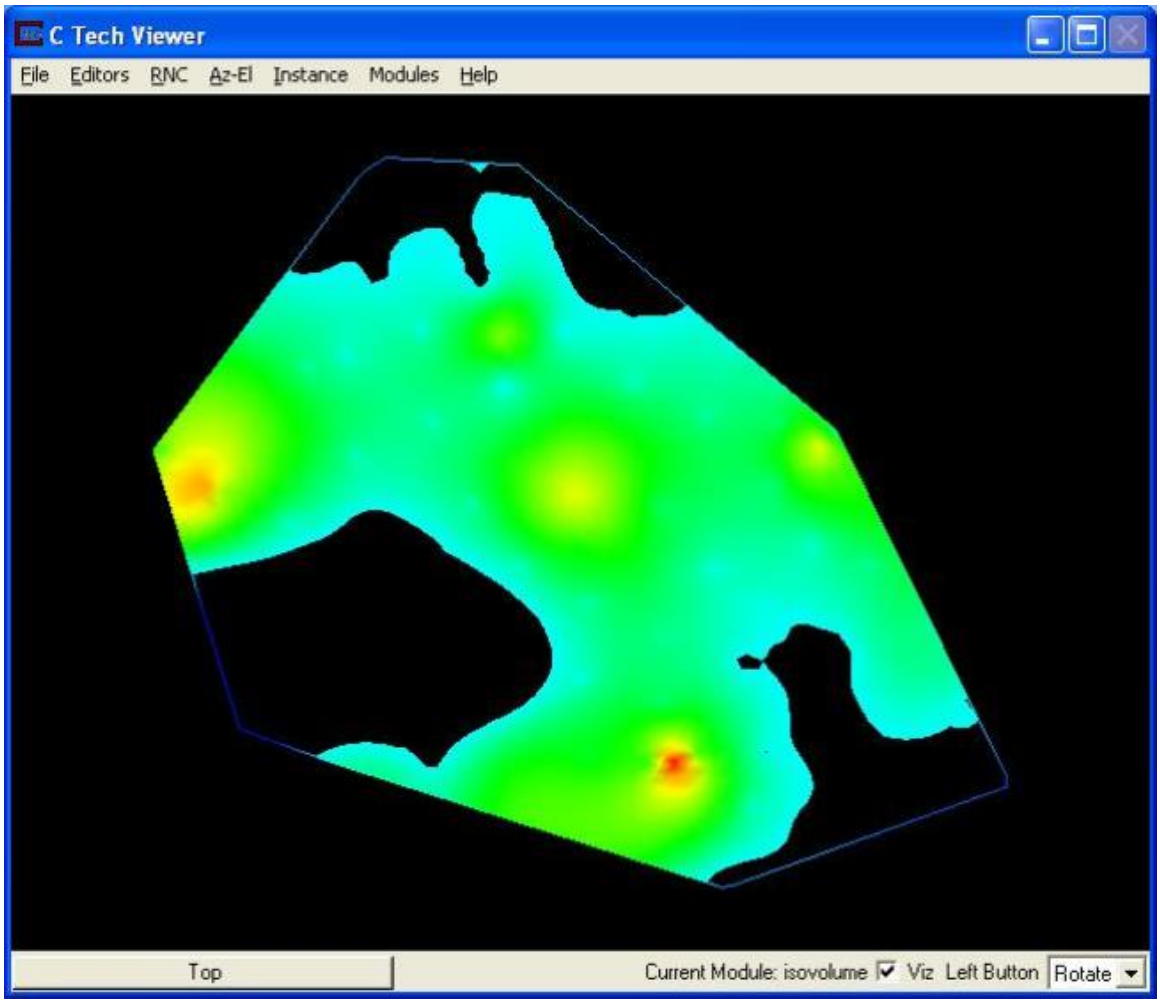


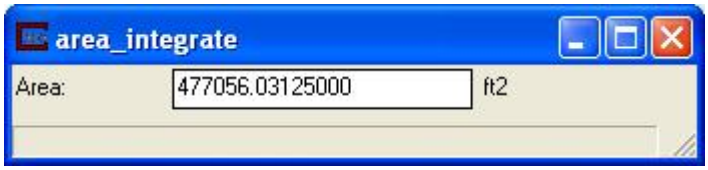
Our nominal affected area is now 375,701.9 sq. ft.

If we switch to display minimum it will show:



And the maximum affected area is:





Clearly we would like a ratio of 1.0, which would signify that there is no variation in the statistically determined contaminant area. The reduction to 1.89 represents a significant decrease.

Nominal area	375701.9	Deviation
Minimum area	266852.06	-29 %
Maximum area	477056.03	27 %
Max / Min area ratio	1.79	

The most logical way to assess our improvement is to look at the reduction in deviations between nominal and min or max. Ideally we would like deviations of zero (0.00%). What we are seeing is a reduction in Min deviation of 41%

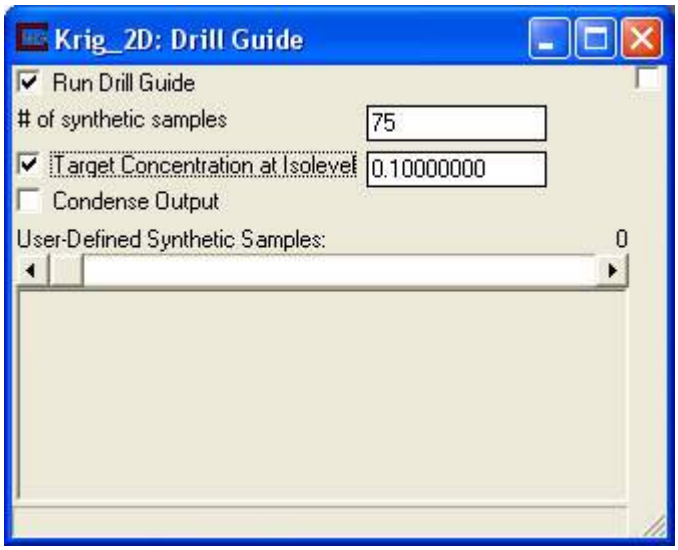
and reduction in Max deviation of 46%. Given that there were 99 borings initially and we have only added 30 synthetic borings, this is a significant improvement.

Clearly, a ratio of 1.79 for the largest to smallest expected plume area is excessive and still probably quite unacceptable.

To demonstrate that additional improvement is possible, we will run a total of 75 Drill Guide Cycles.

Open the Kriging Parameters window and switch back to Statistics.

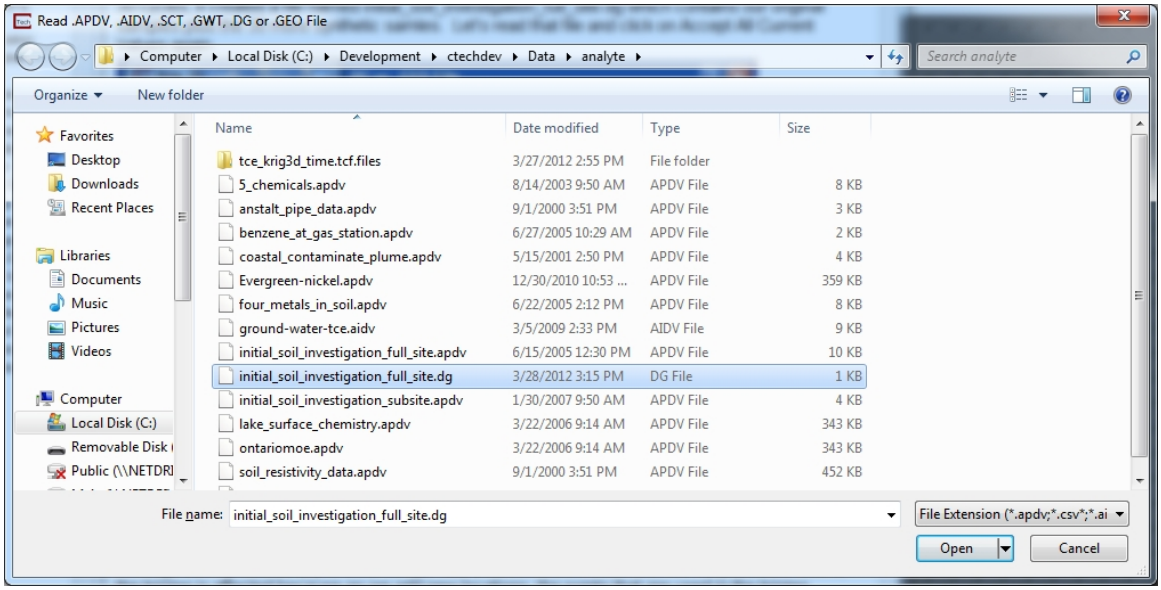
We also need to set our Drill Guide settings back to where we had them before. Open the Drill Guide window and turn on the Run Drill Guide and Target Concentration at subsetting level toggles. Also verify that the Target Concentration is set to 0.1 and set the number of Drill Guide Cycles to 75.



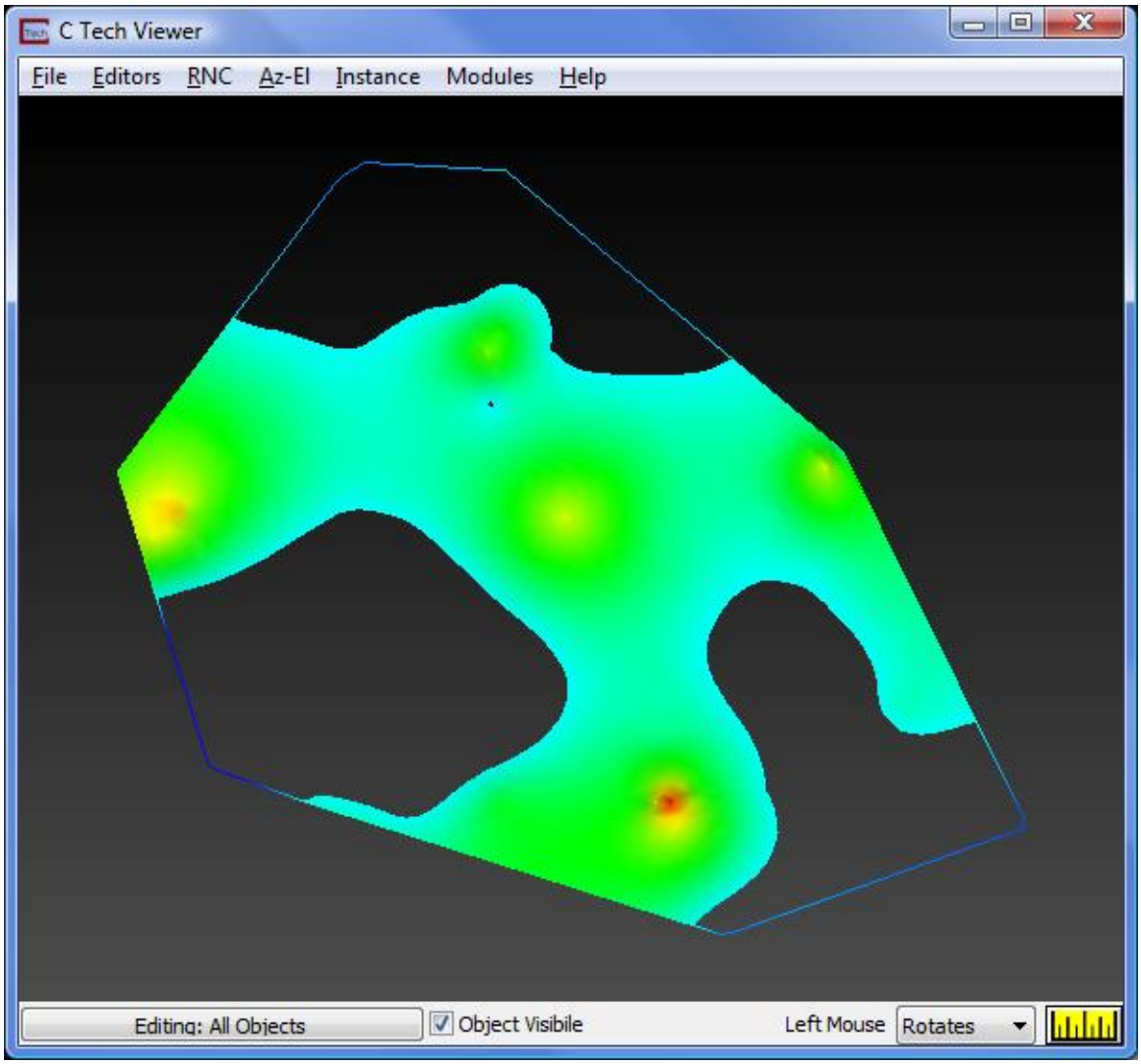
Be sure to select the original input data file initial_soil_investigation_full_site.apdv and click *Accept All Current Values*.

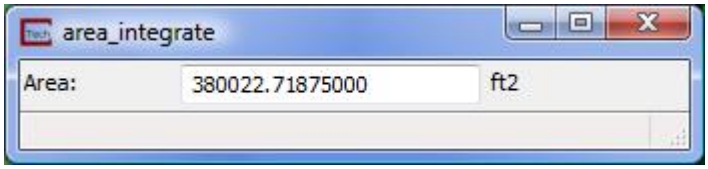
Once all the cycles have run, the (*.dg) data file created (initial_soil_investigation_full_site.dg) will automatically be the selected file. Switch the Kriging Parameter's option back to Min-Max Plume. Also, we will need to turn off the Run Drill Guide and Target Concentration at subsetting level toggles in the Drill Guide window..

Remember to click on *Accept All Current Values* one last time.

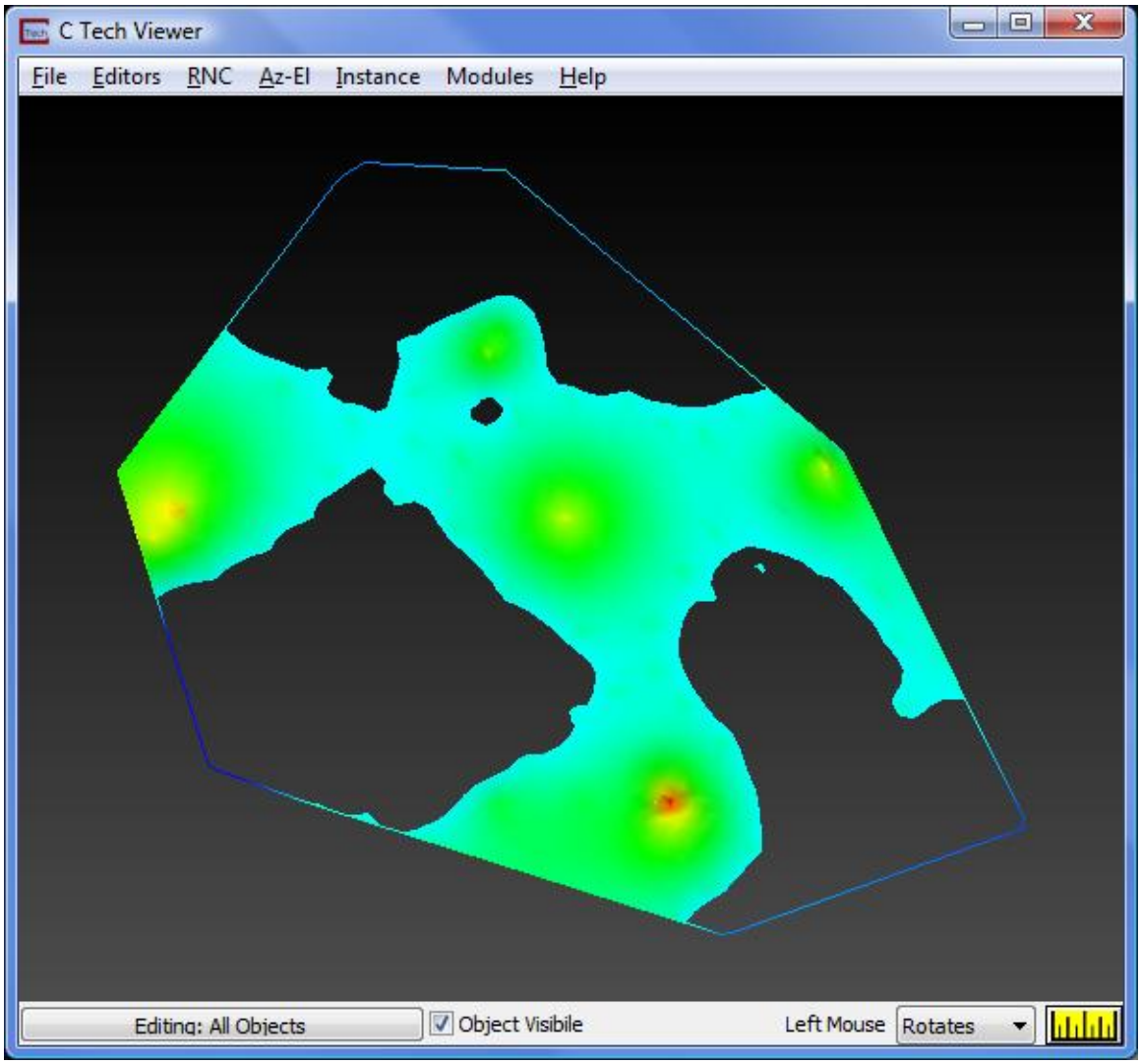


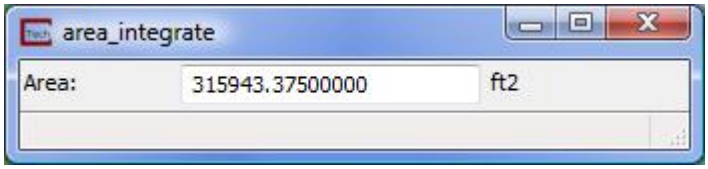
Disconnect Krig_2D's red port from the Viewer. Also, change plume_area back to displaying the nominal plume and set the *Exponentiated Level* to 0.1. After running the 45 additional cycles the Nominal area display should be:



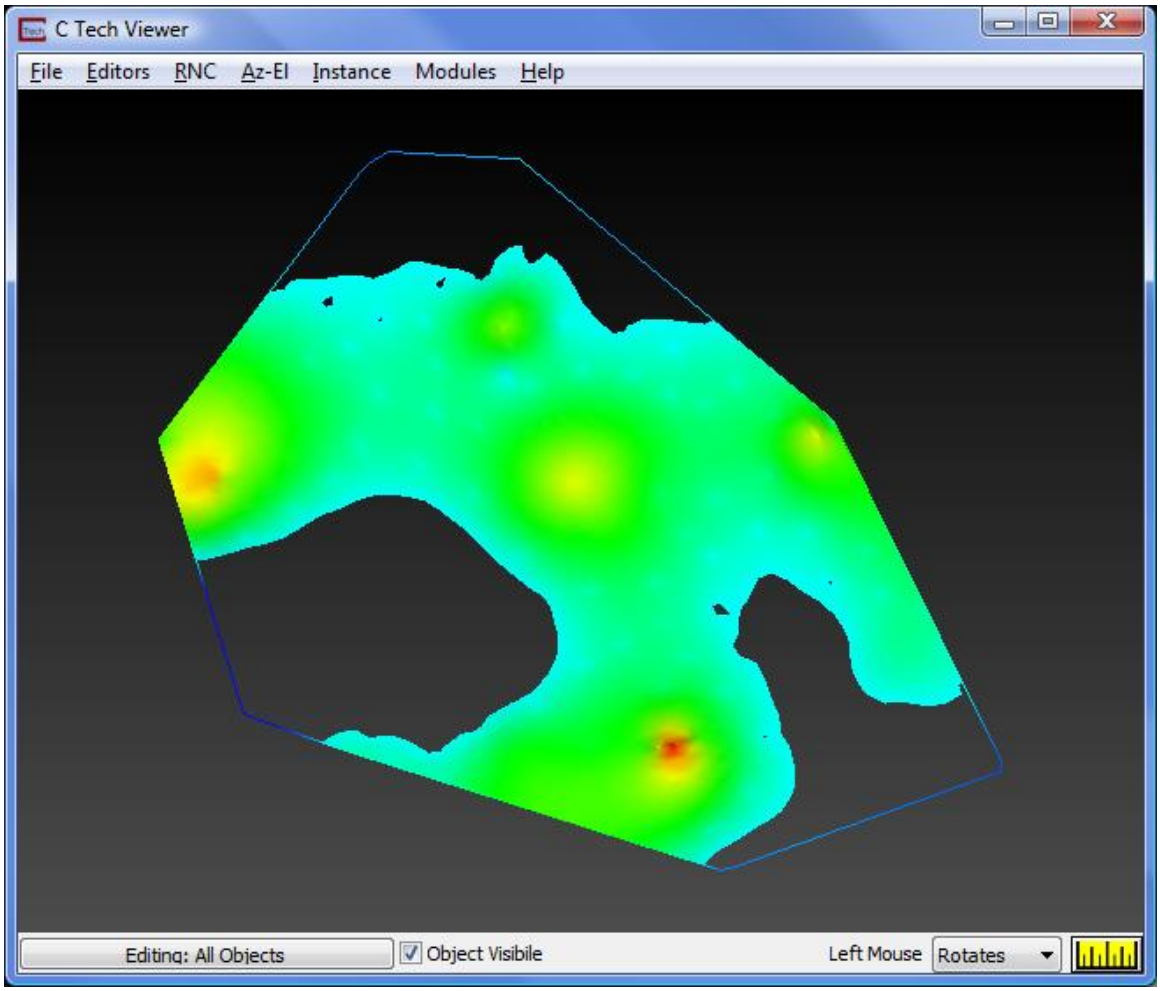


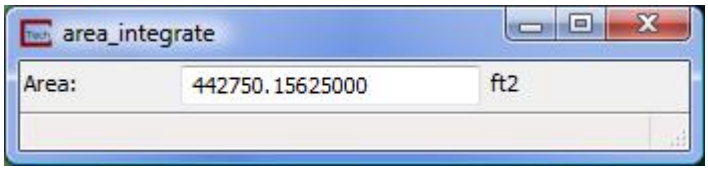
The Minimum area display should be:





and the Maximum area should be





Nominal area	371091.28	Deviation
Minimum area	307840.25	-17.0%
Maximum area	433038.78	16.7%
Max / Min area ratio	1.41	

Originally we had 43 samples. We now have 75 additional synthetic samples for a total of 118.

Once again, we can estimate the average spacing between our borings by taking the square root of: the total domain area divided by the number of borings. The square root of the total site area of 669,690 sq. ft. divided by 118, yields an average spacing of 75 feet (down from 125).

We have reduced the average spacing by 40% and achieved a substantial reduction in area variation.

DrillGuide© Conclusion

DrillGuide© technology provides a more defensible and cost optimized approach for determining new sampling locations for site assessments involving all types of analytes and conditions. Whether your site involves soil or groundwater contamination or gold concentrations, this technology will reduce the number of samples needed and provide a superior assessment of the analyte distributions.

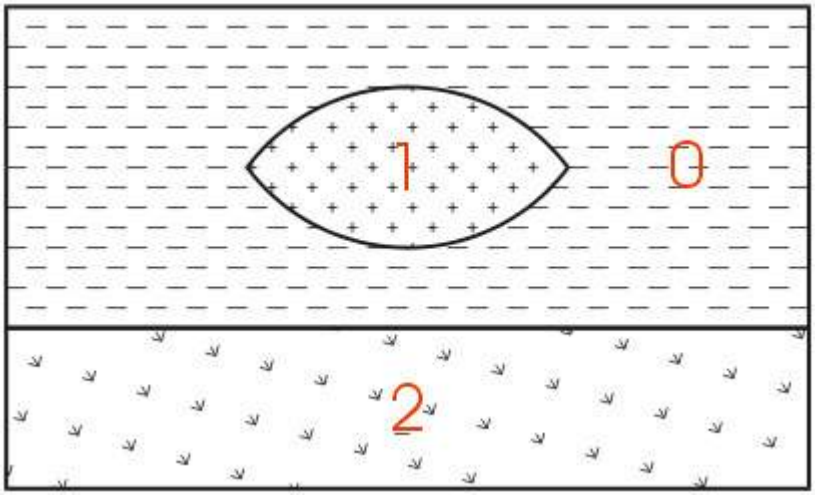
Workbook 3: Creating A Geologic Hierarchy

Geologic Modeling Concept Basics

Geologic Hierarchy

C Tech's original method for 3D geologic modeling utilizes one of two different ASCII file formats which contain "interpreted" geologic information. These two file formats both describe points on each geologic surface (ground surface and bottom of each geologic layer), based on the assumption of a geologic hierarchy. Simply stated, layer hierarchy requires that all geologic layers throughout the domain be ordered from top to bottom and that a consistent hierarchy be used for all borings. At first, it may not seem possible

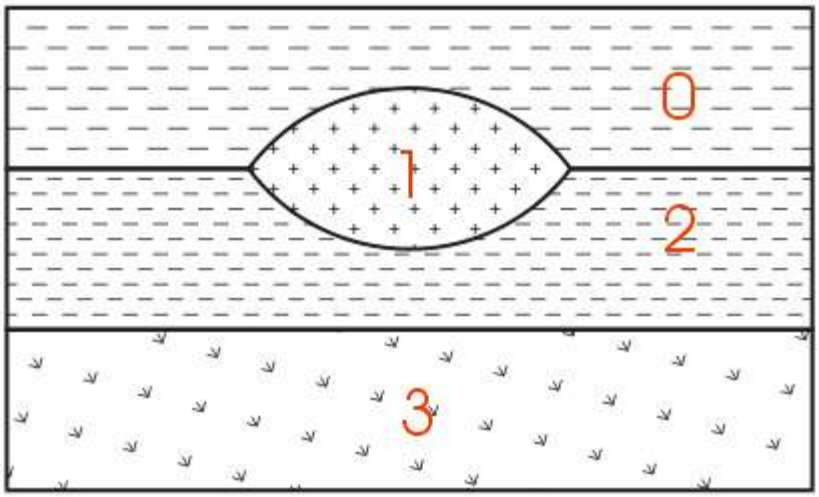
for a uniform layer hierarchy to be applicable for all borings. Layers often pinch out or exist as localized lenses. Also layers may be continuous in one portion of the domain, but are split by another layer in other portions of the domain. However, all of these scenarios and many others **can** be easily modeled using a hierarchical approach.



The easiest way to describe geologic hierarchy is with an example. Consider the example above of a clay lens in sand with gravel below.

Imagine borings on the left and right sides of the domain and one in the center. Those outside the center would not detect the clay lens. On the sides, it appears that there are only two layers in the hierarchy, but in the middle there are three materials and four layers.

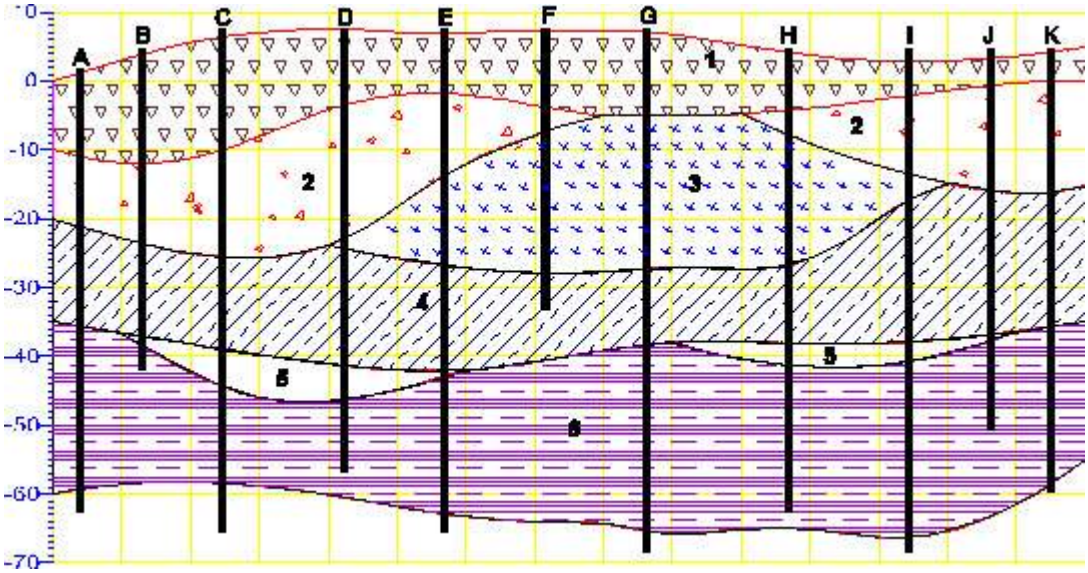
EVS's & MVS's hierarchical geologic modeling approach accommodates the clay lens by treating every layer as a sedimentary layer. Because we can accommodate "pinching out" layers (making the thickness of layers ZERO) we are able to produce most geologic structures with this approach. Geologic layer hierarchy requires that we treat this domain as 4 geologic layers. These layers would be Upper Sand (0), Clay (1), Lower Sand (2) and Gravel (3).



If desired, both Upper and Lower Sand can have identical colors or hatching patterns in the final output.

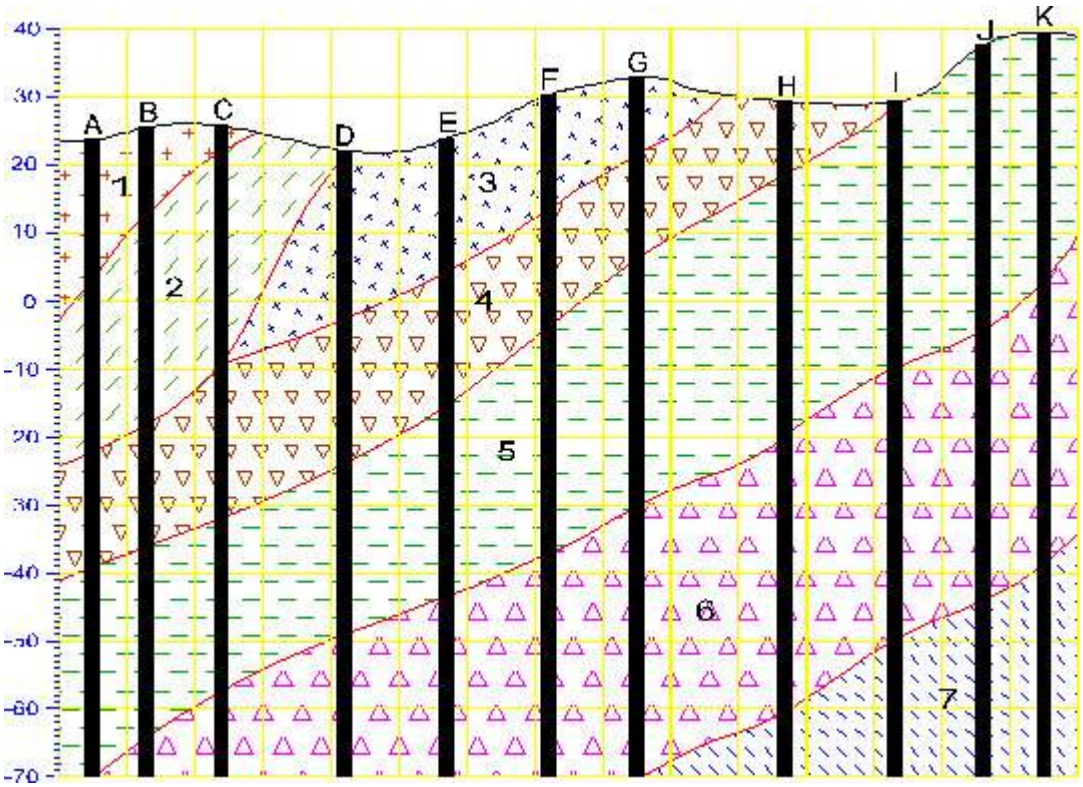
For those sites that can be described using the above method, it remains the best approach for building a 3D geologic model. Each layer has smooth boundaries and the layers (by nature of hierarchy) can be exploded apart to reveal the individual layer surface features. In the above example, the numbers represent the layer numbers for this site (even though layers 1 and 3 are both sand). Two examples of much more complex sites that are best described by this original approach are shown below.

Geologic Example: Sedimentary Layers and Lenses



Geology Example & Figure: Outcrop of Dipping Strata

EVS is not limited to sedimentary layers or lenses. The figure below shows a cross-section through an outcrop of dipping geologic strata. EVS can easily model the layers truncating on the top ground surface.



Geologic File Formats

EVS conducts most of its geologic modeling and analysis using input data contained in one (or more) of three different ASCII files. The content and format of these files has been designed to facilitate export of the data from GIS systems, boring log programs, common database and spreadsheet data storage formats.

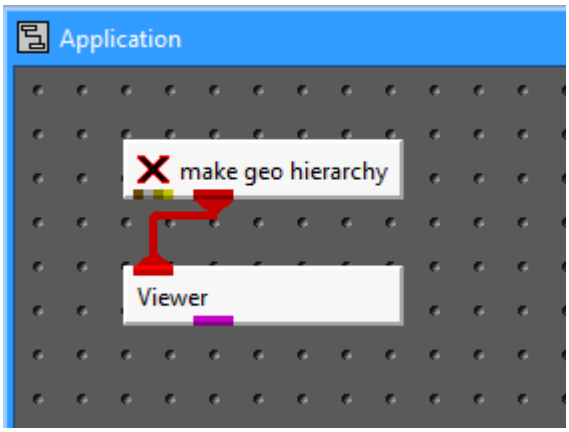
The simplest format used to represent geologic information is the **Pre-Geology Format** (PGF). This format represents raw 3D boring logs where materials have been assigned sequential integer values. This workbook covers the process of taking the Pre-Geology file information and converting it to a Hierarchical geologic file (.geo). More advanced use of the PGF file format for Geologic Indicator Kriging is covered in the **Advanced Geologic Modeling Concepts workbook**.

Probably the most commonly used geology file format throughout the workbooks is the **Borehole (.geo) Geology Format**. This file represents interpreted hierarchical geologic layer data collected from vertical borings. More complex examples of .geo files are given in **Geology File for Layers and Lenses** and **Geology File for Dipping Strata**

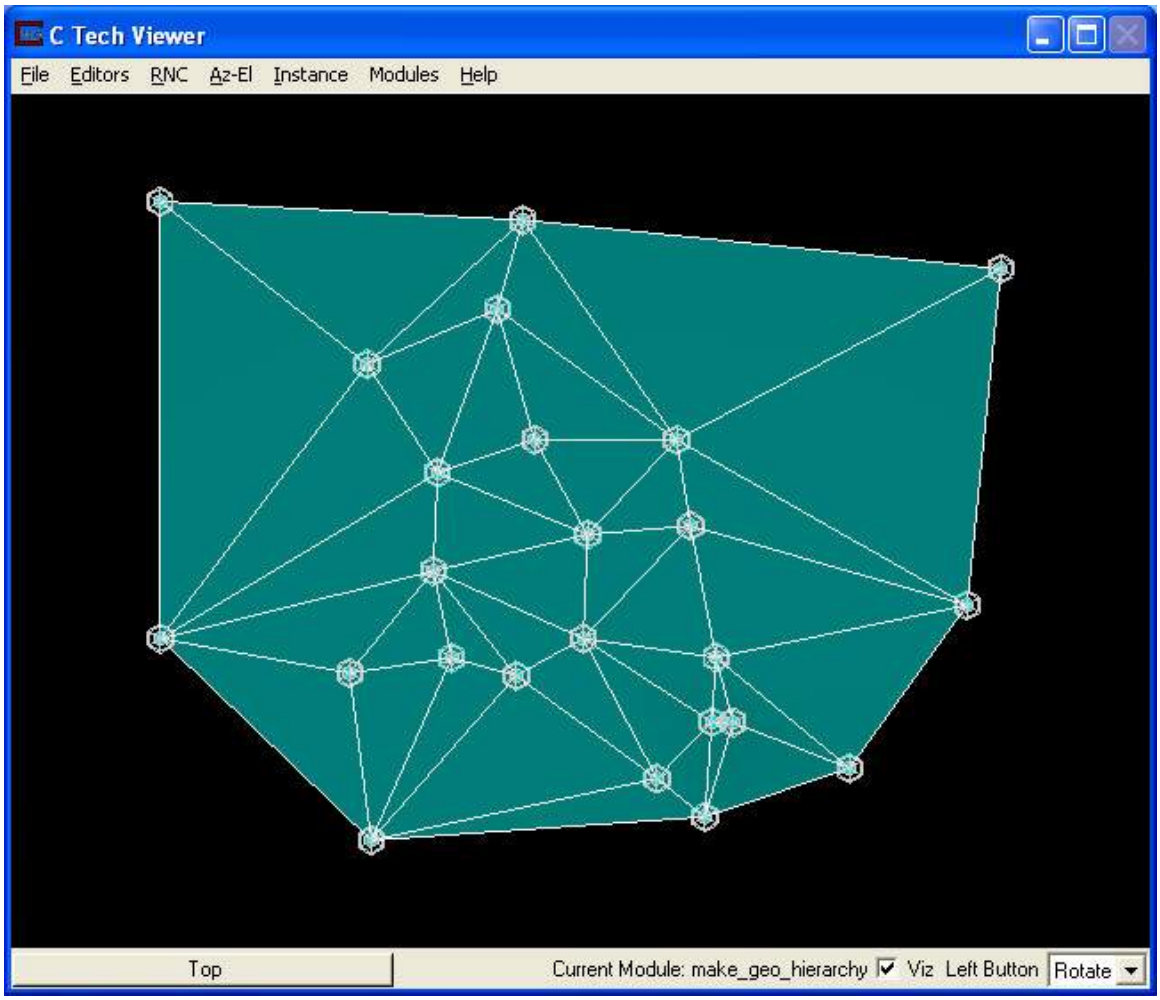
When geologic data (again representing hierarchical geologic layers) is collected from multiple sources such as ground surveys, limited borings, and seismic data, the **Geology Multi-File (.gmf) Format** is generally used. All modules which use .GEO files can use .gmf files instead.

Create A Simple Geologic Hierarchy

Instance both the make_geo_hierarchy module and the Viewer module and connect them to form the simple application below.



Double click on make_geo_hierarchy and on the main window and Load the file "initial_soil_investigation_subsite.pgf". The image in your Viewer should be:



The surface that is shown is a TIN (Triangular Integrated Network = connected grid of triangles) created by connecting each of your borings as the corner nodes of triangles. Initially, it is the top surface across all of the borings. Once the top surface is written to the file the surface (or just portions of it) can be moved downward to represent the bottom of each successive hierarchical layer. Each boring in the pgf file results in a corner node (intersection) in this TIN.

Each intersection is demarked by either a white mesh sphere or a grey mesh sphere. This change in color is related to the current state of the boring which can be either Normal (white) or Locked (grey). This state limits the boring movement options and will be discussed in more detail below.

The first step when creating a geologic hierarchy is to select the material of the top surface. The surface at this point is colored according the selected material, if a boring existed that had a different material at its top elevation it could be set as the top material by changing the selected material in the "Material #: Material Name" list box. Since there is only one material present at the top surface for this PGF file, the correct material is selected by default. At this point, press the Write Surface button. This will create a GMF file consisting of the single surface just created. The file name by default is the PGF file name with the tag "_pgf" appended to the base file name.

After writing the top surface other options become available. The Method option should be selected first.

Choosing The Appropriate Method

There are two different methods for generating a hierarchy, the Layer Thickness method and the Strike/Dip method. We will demonstrate both as we go through the workbook.

The Layer Thickness method uses the Distance slider to move the surface up or down the borings within the selected material. The bottom elevation of each layer at every boring is specified by adjusting the distance slider within the maximum value which corresponds to the maximum thickness for the slected material at any boring.

The strike and dip values are used to define creates two *visualization planes,* a Top surface and a lower second parallel surface whose position is defined by the *Tolerance* slider. The upper surface is automatically positioned at a the highest possible (locked or automatically selected) sphere. No other point defining the surface we are working on can be above the Top surface. The lower plane is twice the tolerance below the top plane.

By increasing your tolerance value, you allow the bottom of your layer to vary more, which will create a "thicker" layer overall. Shrinking your tolerance value will force the layer to more closely follow your strike and dip, but may result in creating extra layers.

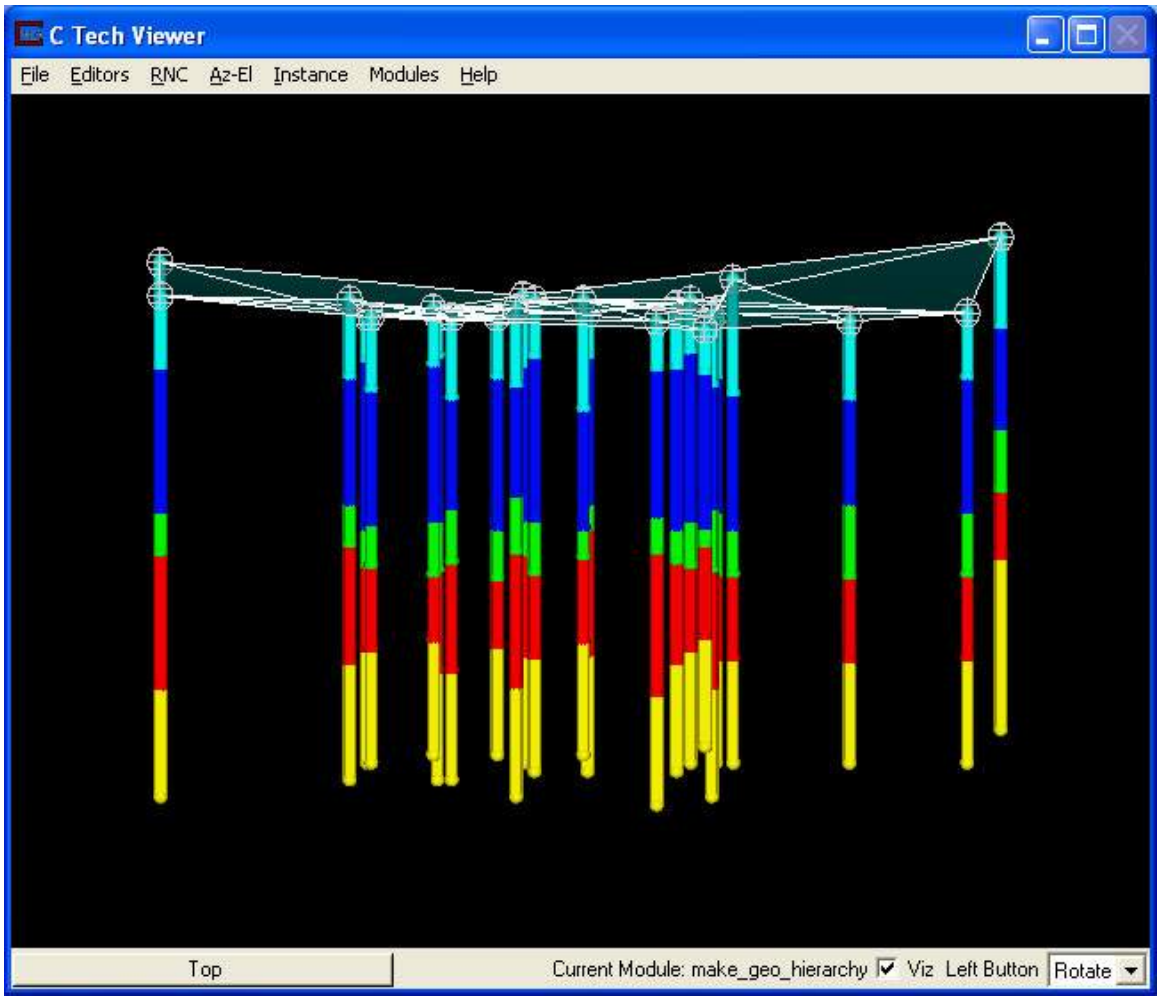
The difference between the two is highlighted when using the cycle option. The cycle option will try to automatically create your hierarchy for you. If the Layer Thickness method has been chosen, the cycle option will set the distance to the max distance and write out the bottom of the layer, it will do

this repeatedly until it all borings have been dropped from the surface. This works very well if you have very simple, "pancake-style" lithology. If there was a strike or dip to the layers and if any layers are pinching out at the ground surface, the geology would not be properly created with this method. The Strike/Dip Method can be used to automatically generate models to avoid these problems, but requires more setup and user guidance. Since this method is based on a strike and dip these will be honored even when cycling. The difficult aspect of this automated method is choosing the correct tolerance.

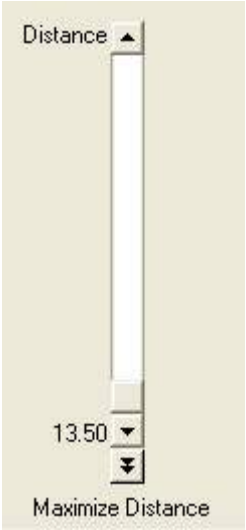
We will begin by choosing the Layer Thickness method.

Building A Hierarchy with the Layer Thickness Method

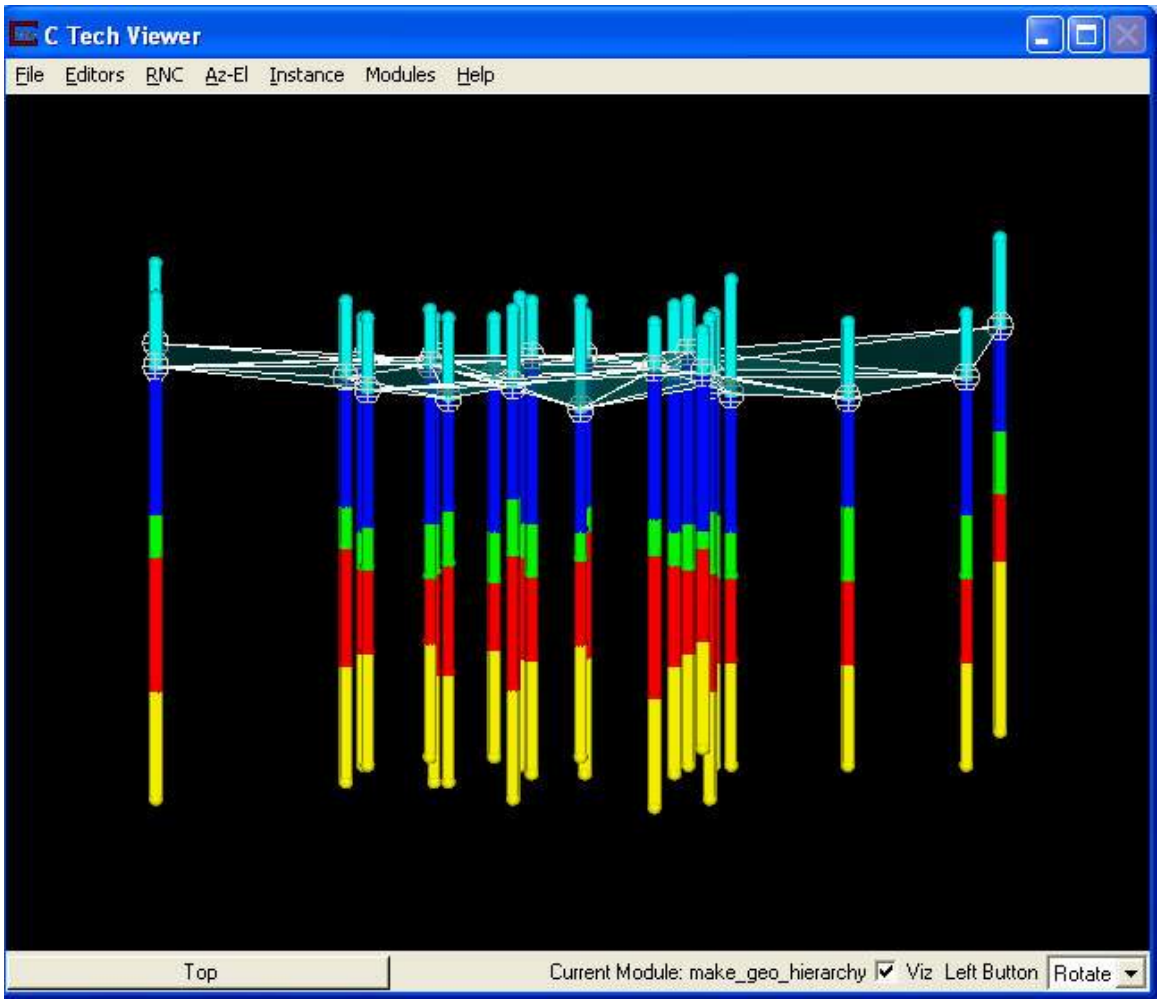
Open the Az-El panel for the viewer and set the elevation to zero and you should see the picture below.



Notice that this is a relatively flat geology without any lenses or pinch-outs. Because of this, the Layer Thickness method is a simple, effective way to specify our hierarchy. Slowly drag the Distance Slider to its maximum extent.

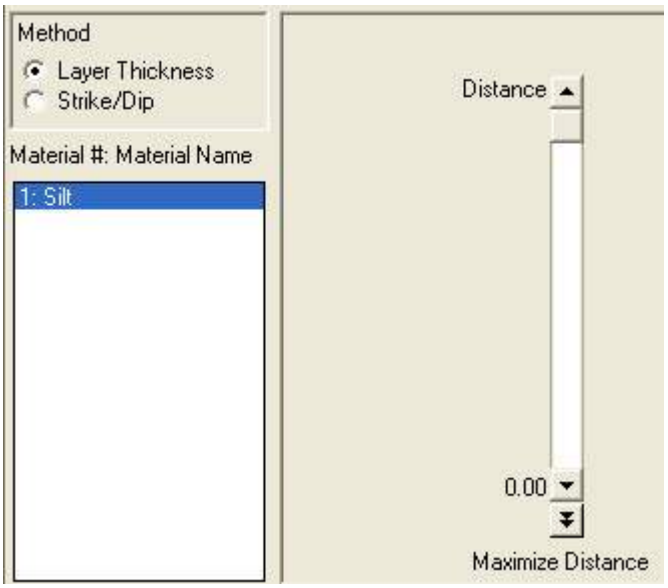


Your view should now look like the following image:

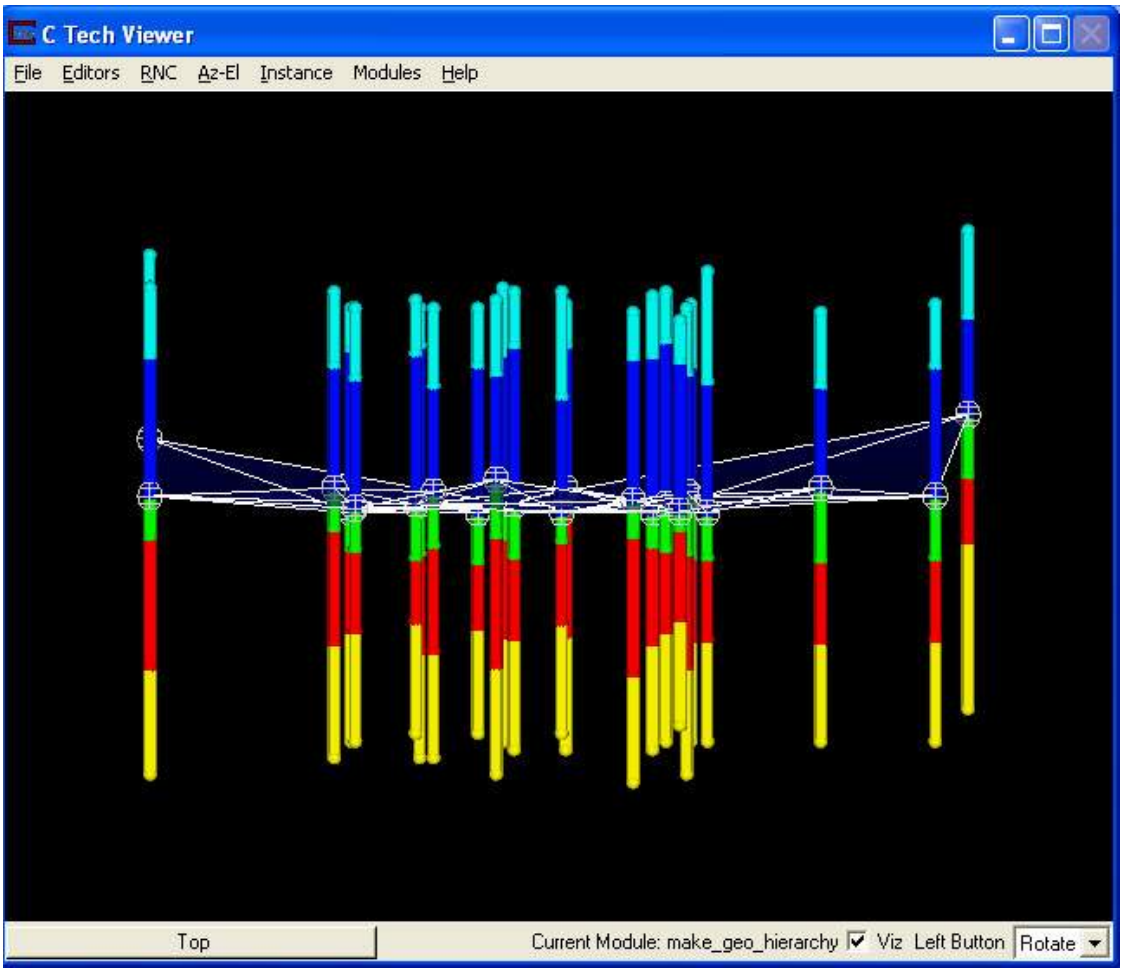


Notice that when the TIN intersection point reaches a different material it stops moving, this does not mean that the bottom of the layer has been fully defined, it has just been defined at that point. The maximum distance on the slider should generally be used unless trying to define a lens or pinch-out. Select the Write Surface button again.

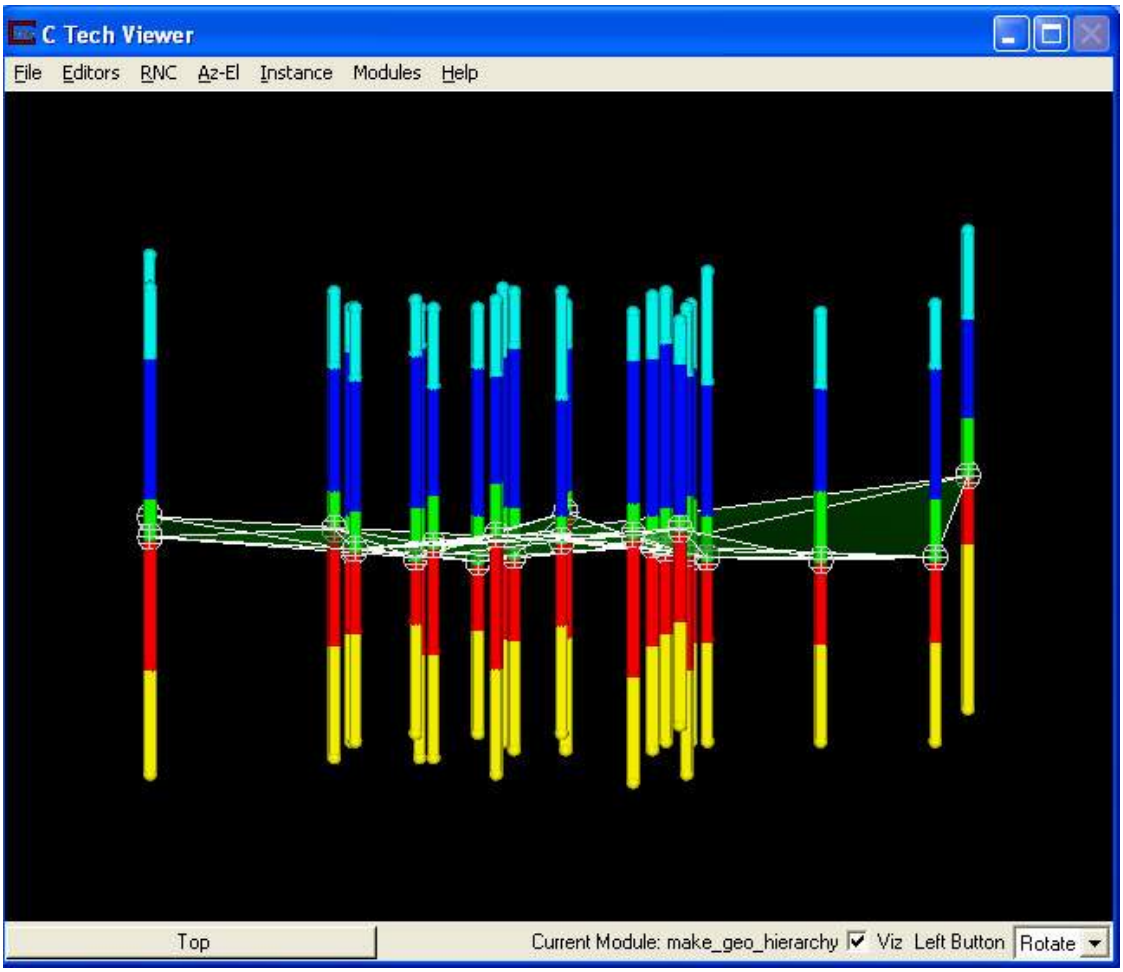
1. Notice that the distance has been reset to zero and the selected material type is now "1: Silt".



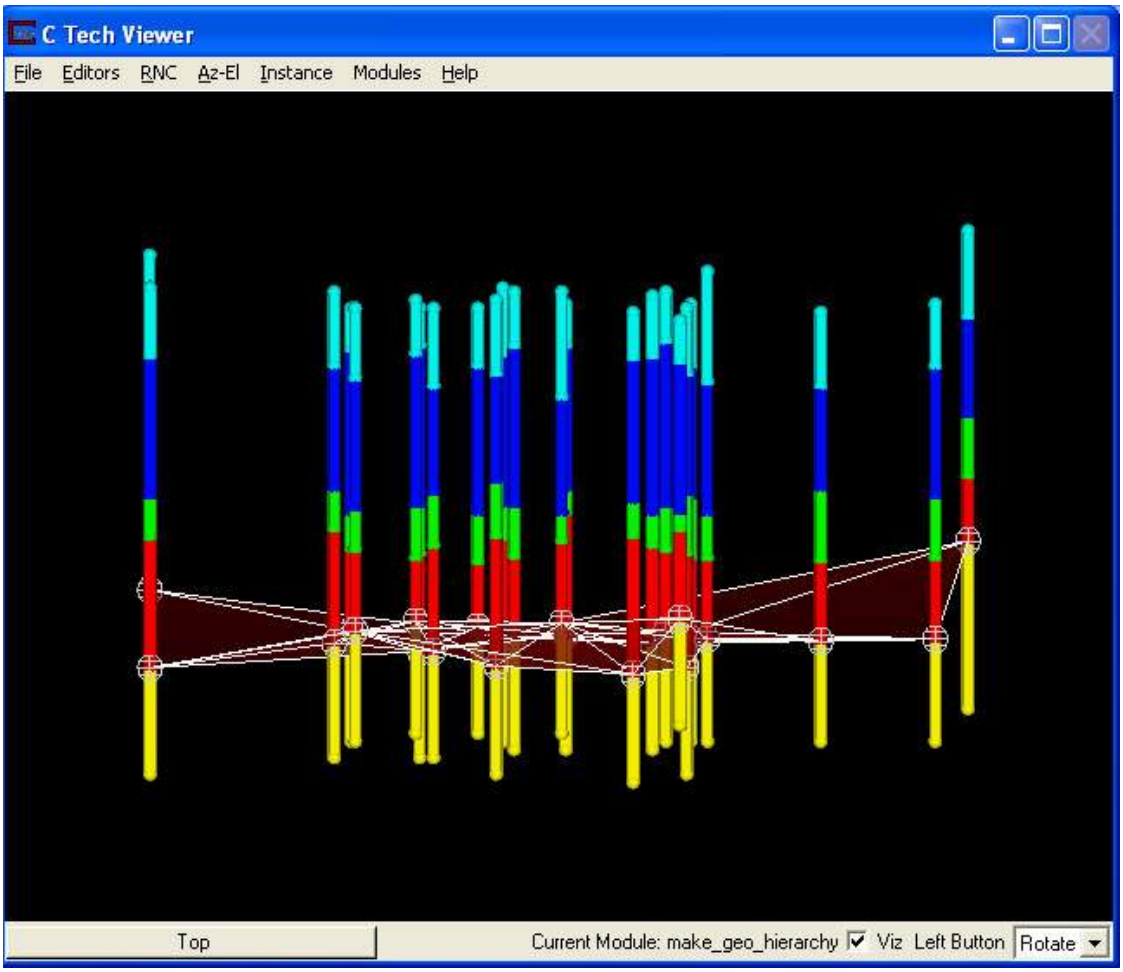
2. This is the only material option for this layer. Click the double down arrow button below the distance slider to completely define the bottom of the next layer. The maximum distance in this case is 20 which corresponds to the maximum thickness of the Silt layer being created.



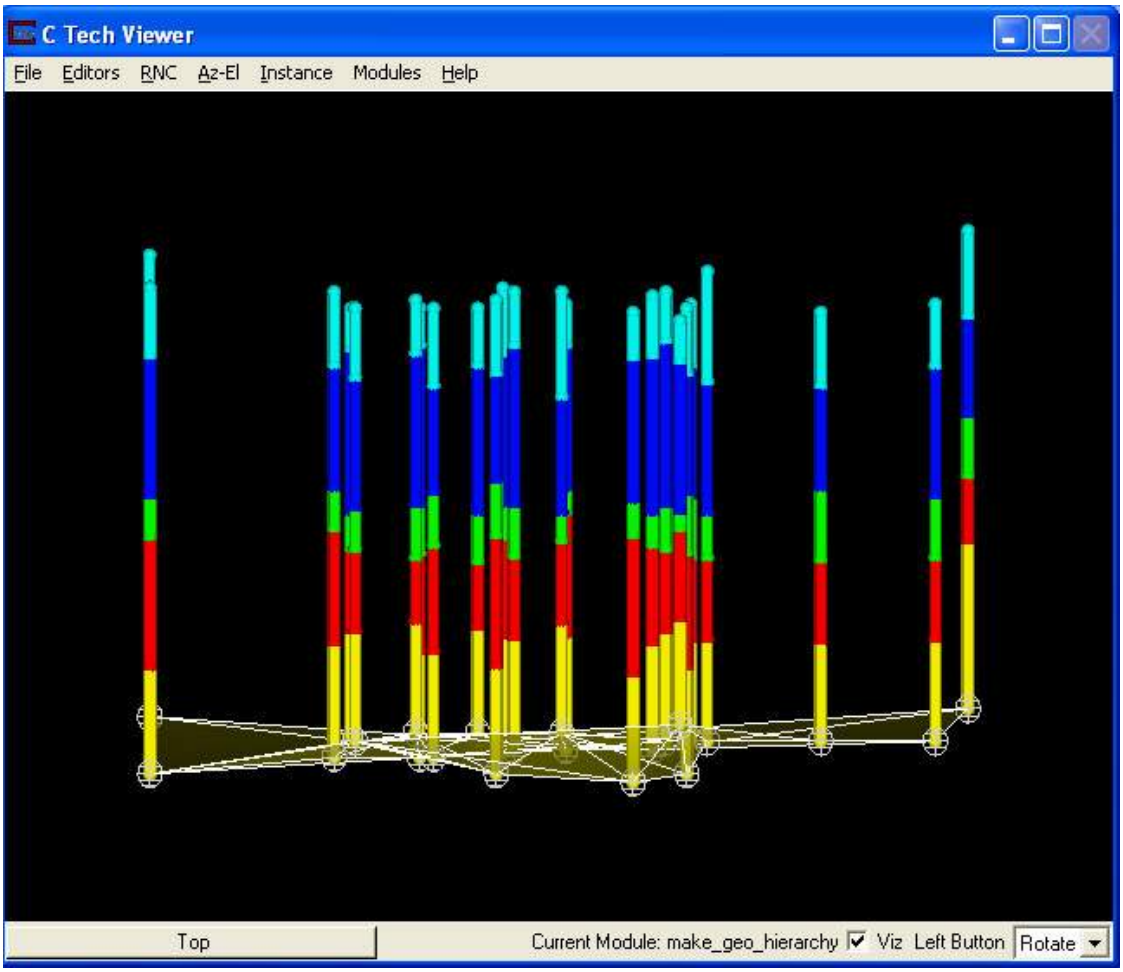
3. Select the Write Surface button to write this surface to the GMF file.
4. The next surface is "Clay", click the double down arrow button below the distance slider to completely define the bottom of the layer. The layer should look like the one below.



5. Select the Write Surface button to write this surface to the GMF file.
6. The next surface is "Gravel", click the double down arrow button below the distance slider to completely define the bottom of the layer. The layer should look like the one below.



7. Select the Write Surface button to write this surface to the GMF file.
8. The next surface is "Sand", slowly drag the distance slider to its maximum extent. Since your PGF file represents raw boring logs, it is sometimes the case that the bottom of each boring corresponds to the maximum logging depth and not the true bottom of the last material. To define the bottom of model at the bottom of each boring would be artificially thinning the thickness of the last material by some unknown amount. In most cases you may know that the bottom of each boring is in fact the bottom of that last material OR if it is not you may still want to define the last surface in your model corresponding to the deepest depth for each boring where geologic data was collected. The default behaviour for the module is to use (preserve) the bottom of each boring that defines the bottom surface. This behaviour may be changed by unselecting the "*Preserve Bottom*" toggle.



9. For our example we will leave the bottom surface undefined, so unselect the Preserve Bottom toggle, make sure the distance slider is at its maximum extent and then select Write Surface. It is important to write out the bottom surface even when all borings have been dropped from it. The last surface in the GMF file should look like the following :

Surface 4 "Sand"
ND_FOR_GEO 11566.299805 12850.599609 short "B-30"
ND_FOR_GEO 11586.299805 13050.599609 short "B-31"
ND_FOR_GEO 11086.299805 13090.599609 short "B-32"
ND_FOR_GEO 11286.799805 13026.700195 short "B-59"
ND_FOR_GEO 11393.500000 12948.900391 short "B-60"
ND_FOR_GEO 11251.299805 12929.299805 short "B-75"

Note the "#" symbol at the beginning of each line, this symbol is universal in all of EVS file formats and will cause these lines to be treated as comments, so in the case of the GMF file it is as if these lines do not exist. When the last surface is written or the Finish Geology button is pressed the GMF file will be converted into a GEO file. In this conversion process the "# ND_FOR_GEO"

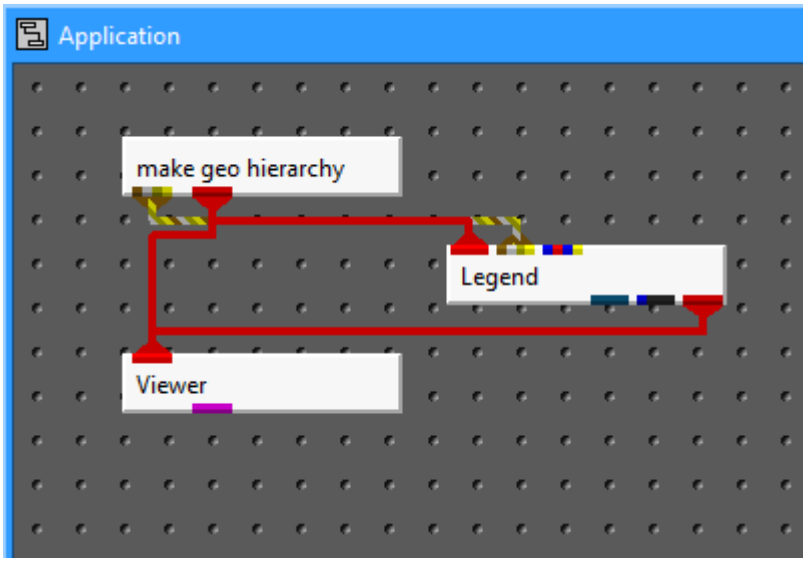
flag will given a "short" value in the GEO file. The purpose of this value is discussed in the file format help for GEO files.

At this point, we have created both a .GEO and .GMF file representing the hierarchical geologic layers associated with our PGF file. In the next topic we will look at a much more complex PGF file and ultimately view the surfaces that we've created.

Hierarchy With Strike/Dip Method

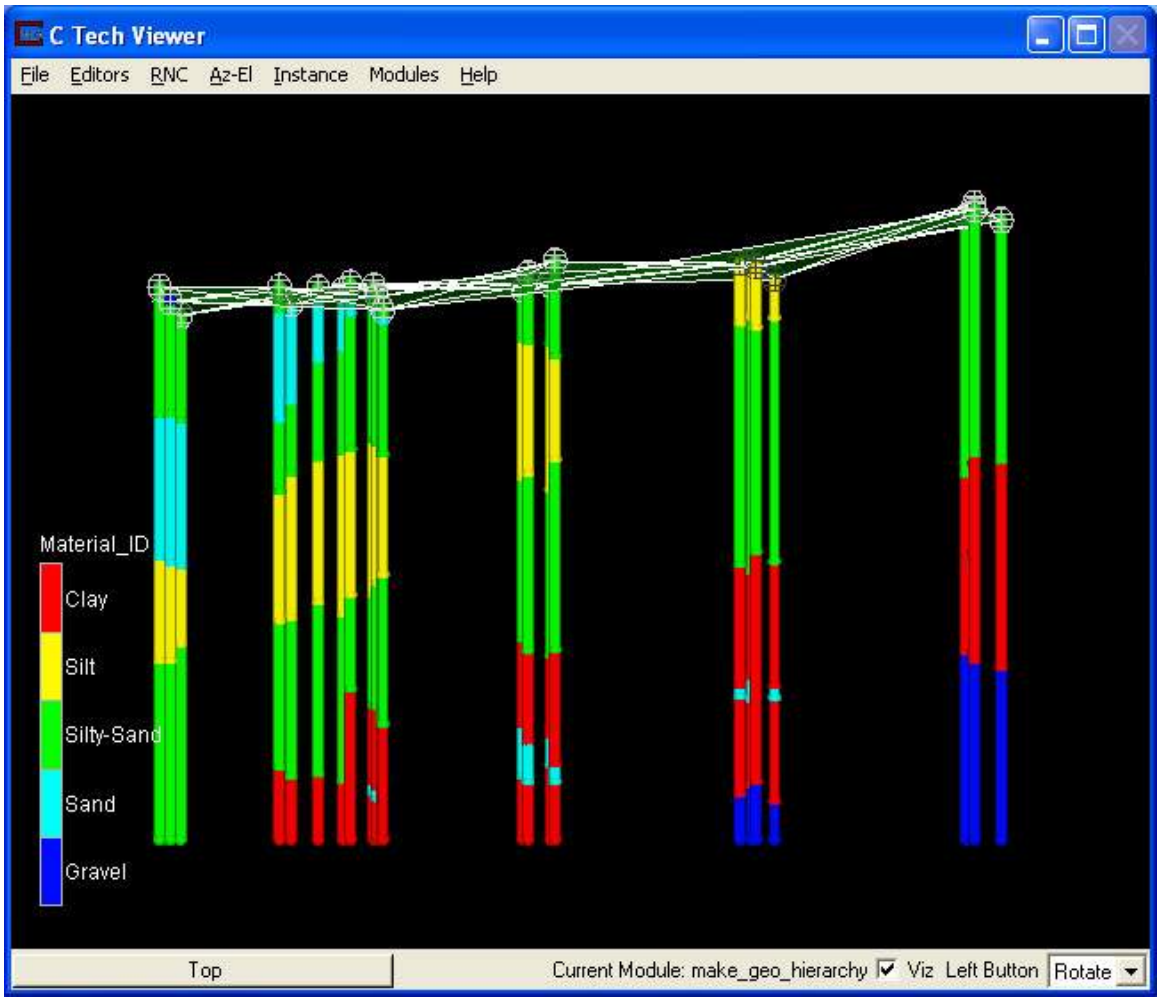
In this topic we will discuss the Strike/Dip method for automatically generating hierarchy. Geologic hierarchy can be generated using either method (even automatically), however for geology that has a pronounced strike or dip, or contains lenses or pinch-outs the Strike/Dip method will produce the most accurate results. This method should be used carefully and all results should be checked against the original PGF file.

First instance a make_geo_hierarchy module, a Legend module and a Viewer and connect them as shown in the image below.



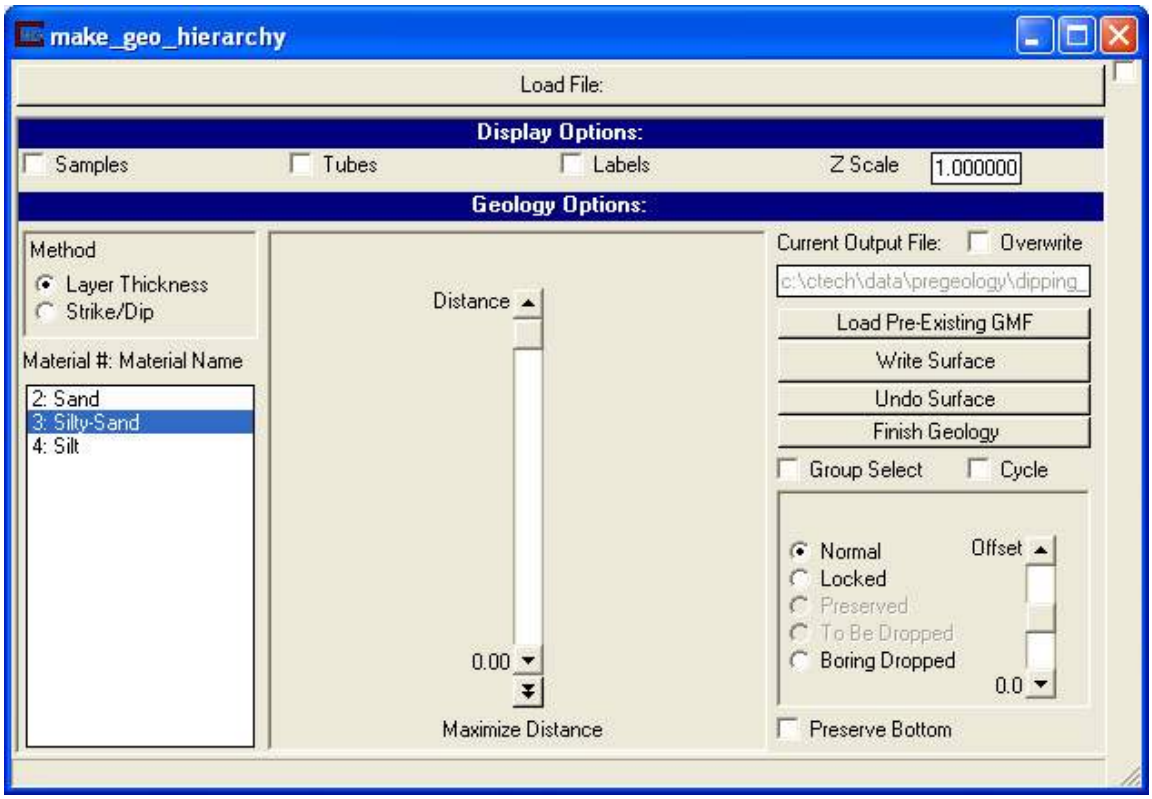
We have added a Legend to this network to help identify the material name of each layer.

1. Next load the pgf file "data\pregeology\dipping_strata_lens.pgf" in the make_geo_hierarchy module.
2. Set the "Z Scale" to 1.0 (this data set has a relatively small horizontal extent relative to its depth).
3. Then open the Az-El panel on the viewer and set the Azimuth to 225 and the elevation to zero. You should get the following image.

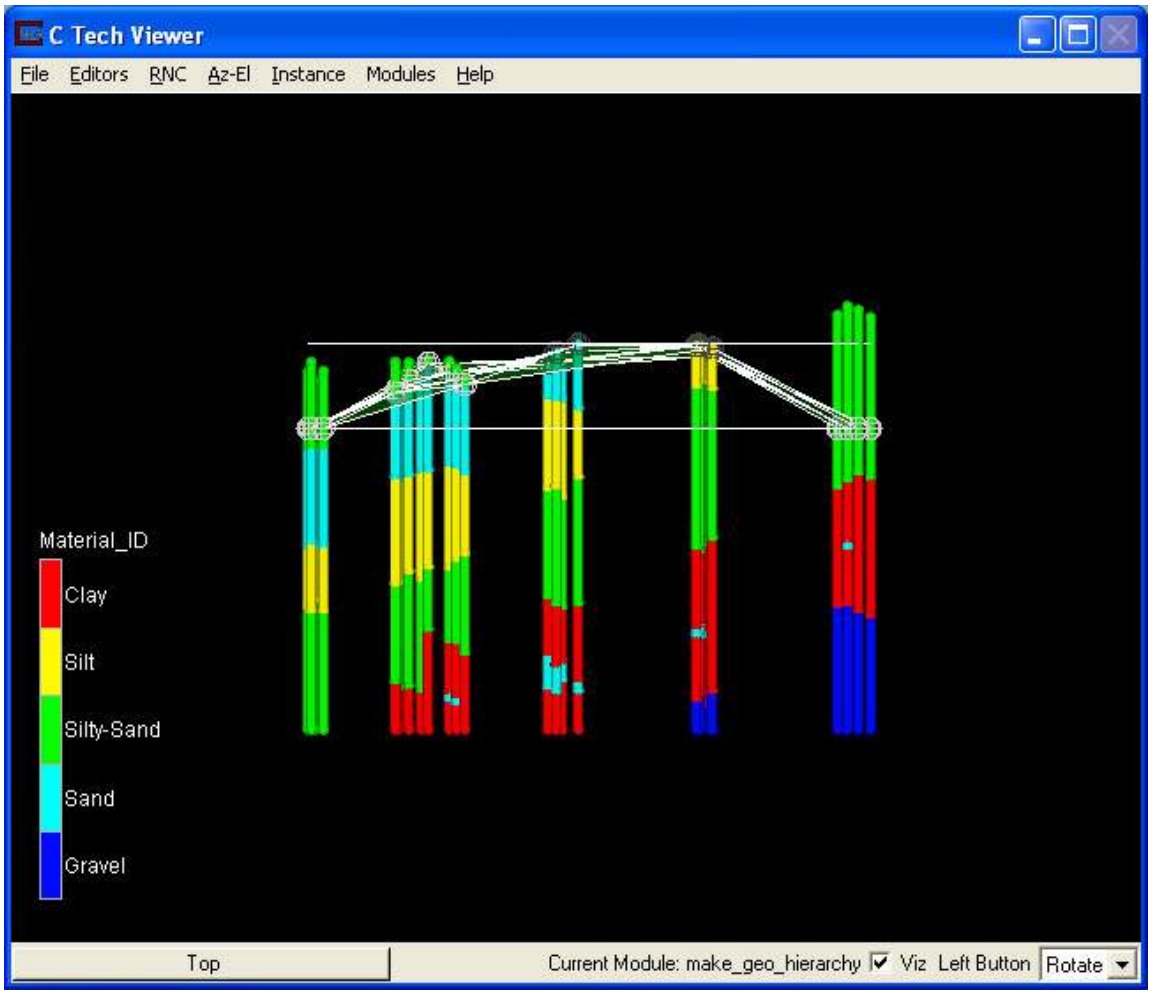


This view of the model shows the dip that is present in the geology, as well as a sand lens.

1. Our first step is to select "*3.Silty-Sand"* in the material selector.. Please note that all borings with *Silty-Sand* at the top are white while all others that do not exhibit this material at ground surface are grey (locked).



1. Click the *WriteSurface* button to write the top surface
2. *Silty-Sand* will automatically be selected in the material selector because the material is always the same for first two surfaces since they represent the top and bottom of the first layer (material).
3. Now, select the Strike/Dip method (vs. Layer Thickness). At this point you should see two horizontal transparent planes in your viewer.

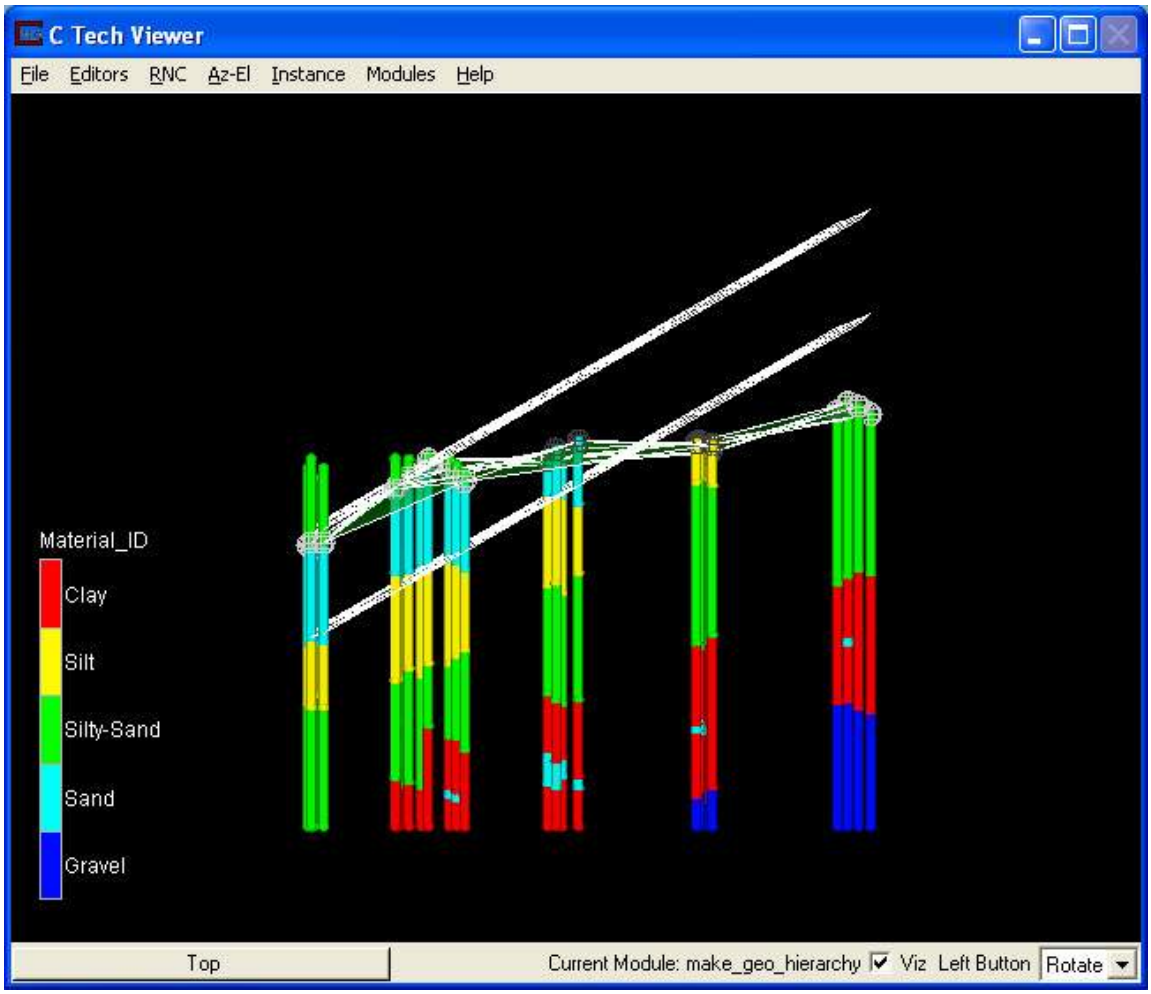


The strike and dip values are used to define creates two *visualization planes,* a Top surface and a lower second parallel surface whose position is defined by the *Tolerance* slider. The upper surface is automatically positioned at a the highest possible (locked or automatically selected) sphere. No other point defining the surface we are working on can be above the Top surface. The lower plane is twice the tolerance below the top plane.

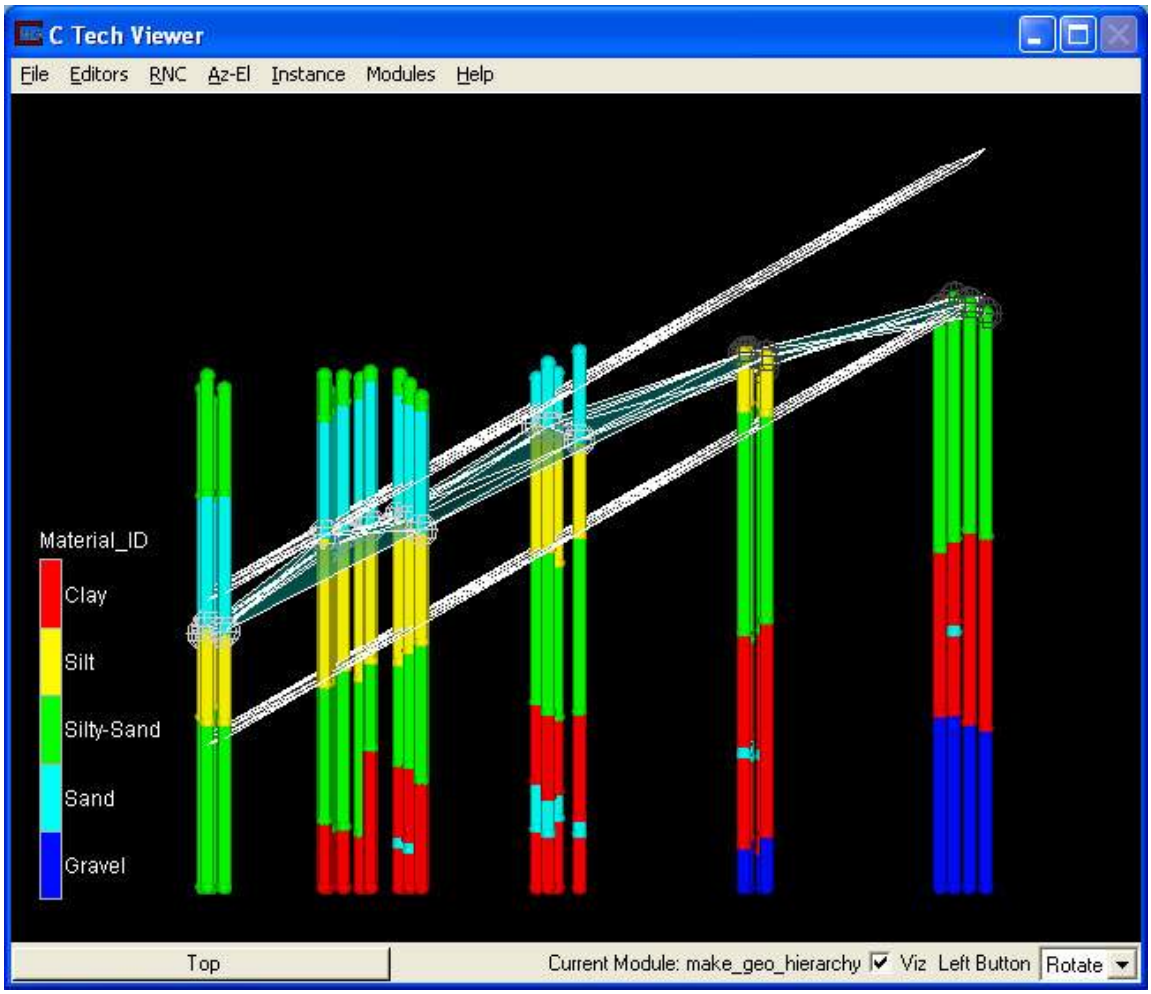
By increasing your tolerance value you allow the bottom of your layer to vary more, which will allow the module to create a "thicker" layer. However the maximum thickness should be defined by the bottom of all boring sections that contain the selected material. Shrinking your tolerance value will force the layer to parallel your strike and dip, but may result in creating extra layers as you progress through your model.

1. We would recommend that you experiment with different strike/dip values to get a feel for the process of selecting appropriate values. However, for this dataset you should set the strike to 42, and the dip to 30 S/W. You may feel that slightly different values are optimal, but for the workbook we recommend

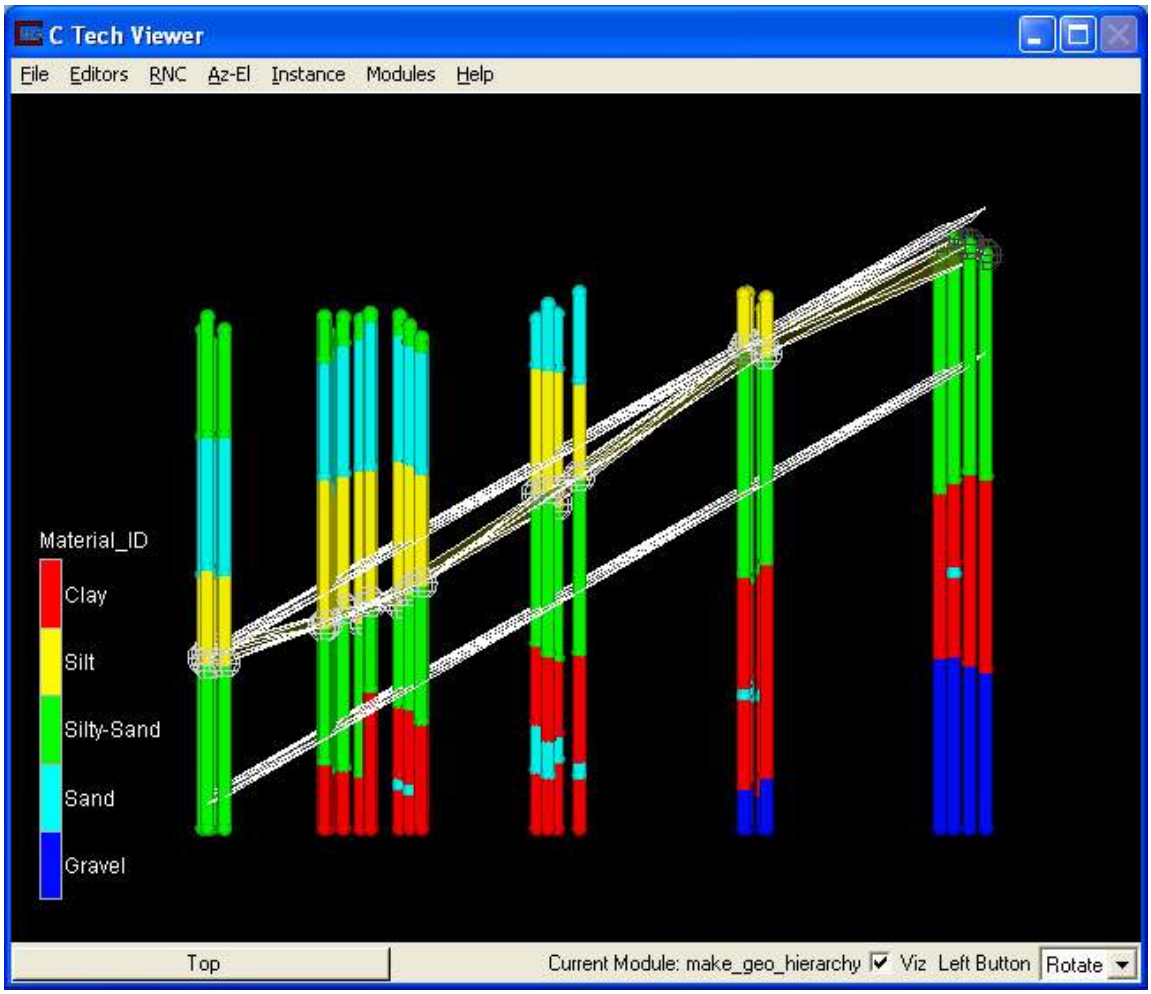
you use our values. Upon setting these values your viewer should show:



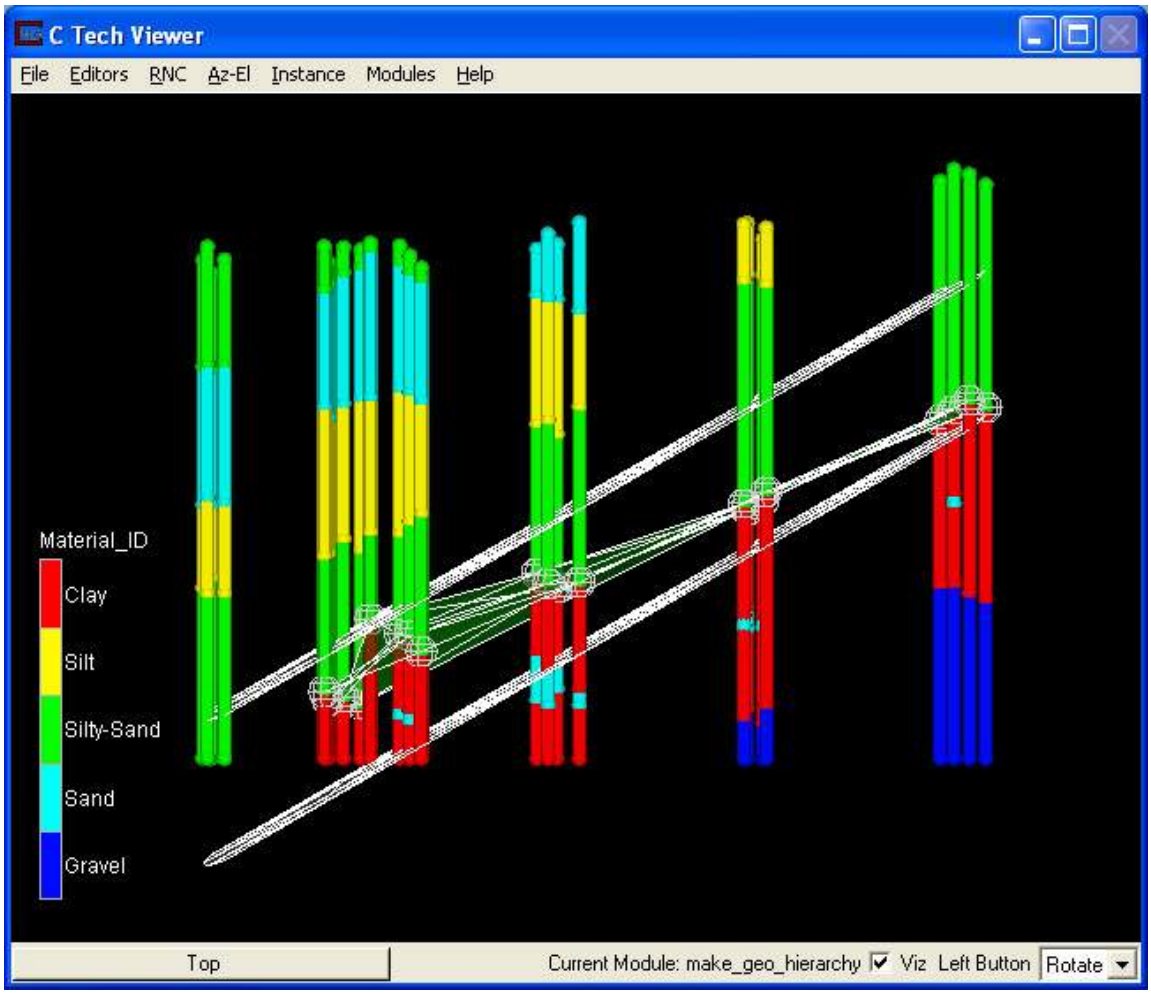
1. First examine the Tolerance at the default value of 11. Note that for this second surface, the bottoms of all *Silty-Sand* borings for the layer we're working on fall within the two *visualization planes.* This is the criteria for an acceptable tolerance. In this case the tolerance could be made much smaller, however in other models this could cause some of the borings to be split. If it is made too large, other borings (where this layer should be pinching out at ground surface) are affected. Rotate the model around with your mouse to see how well things fit.
2. Click the *Write Surface* button to write the 2nd*Silty-Sand* surface. At this point the planes will drop down to the next surface.



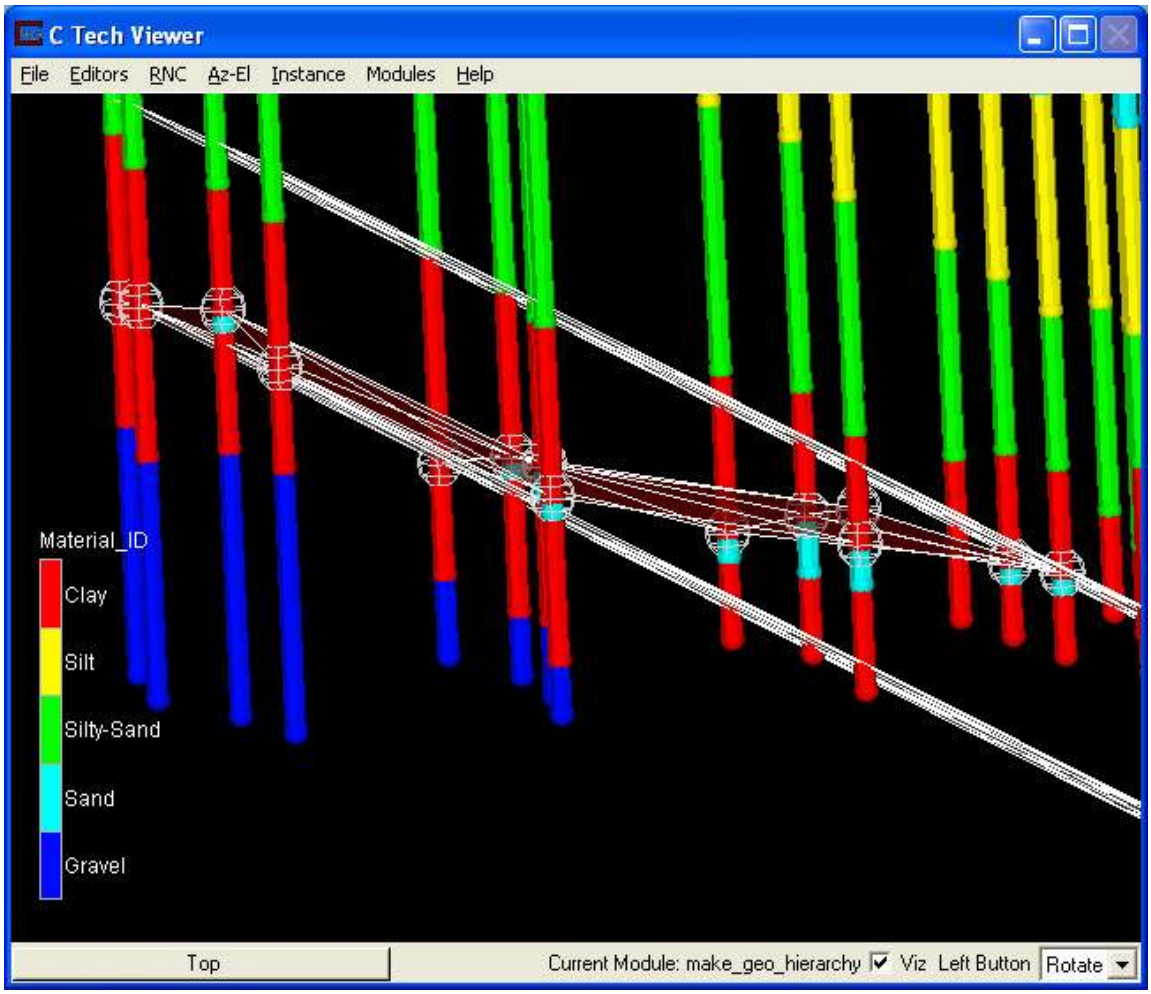
1. Now select *2.Sand* in the material selector to begine the definition of the second layer (third surface).
2. Again rotate the model around with your mouse to see how well things fit.
3. Click the *Write Surface* button to write the 3rd surface which is defined as the bottom of *"Sand"*
4. Select *Silt* in the material selector
5. Rotate the model around with your mouse to see how well things fit.



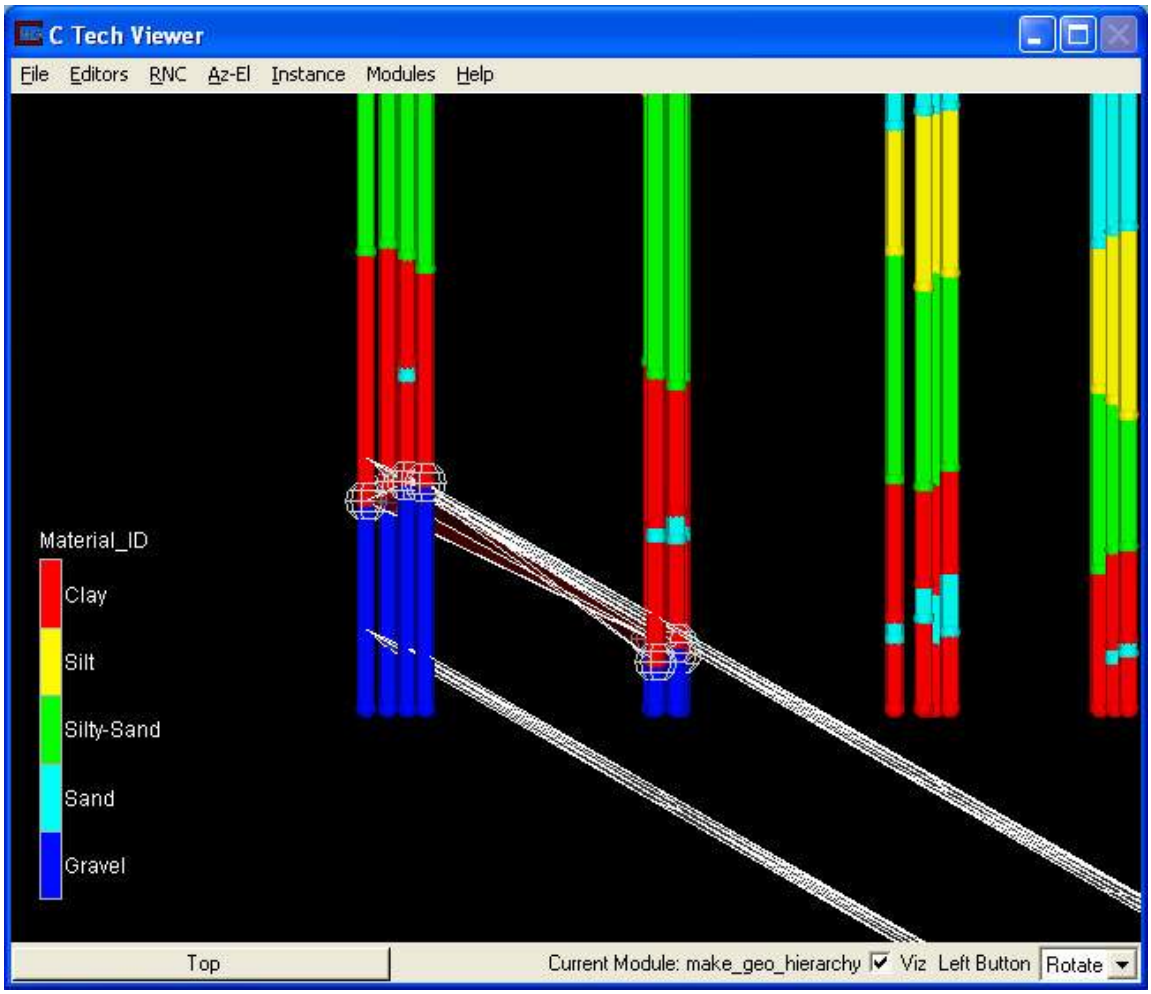
1. Click the *Write Surface* button to write the 4th surface which is defined as the bottom of *"Silt"*
2. *Silty-Sand* becomes the selected material since there are no others present in any borings.
3. With our default tolerance of 11, this z-offset just barely encompasses the bottoms of *Silty-Sand*. If you experiment with smaller tolerances such as 10.0 you will see that the TIN surface is cutting the *Silty-Sand* in the eastern most borings. Our Tolerance of 11 is adequate but just barely.



1. Click the *Write Surface* button to write the 5th surface which is the bottom of another *Silty-Sand* layer.
2. Now things get interesting. We want to define the bottom of *upper* clay that represents the **top** of a sand lens inside the clay. In this case, the Tolerance dramatically affects the shape of the lens.
3. Set the Tolerance to 10.0. At this setting, all clay borings that include the sand lens will have reached the bottom of upper clay material and the borings without the lens will be split near the center of the lens. Note that a value of 9 would have inappropriately split some of the borings. Our previous value of 11 could be used, but would have pushed the piched out edges of the sand lens further down some of the borings.

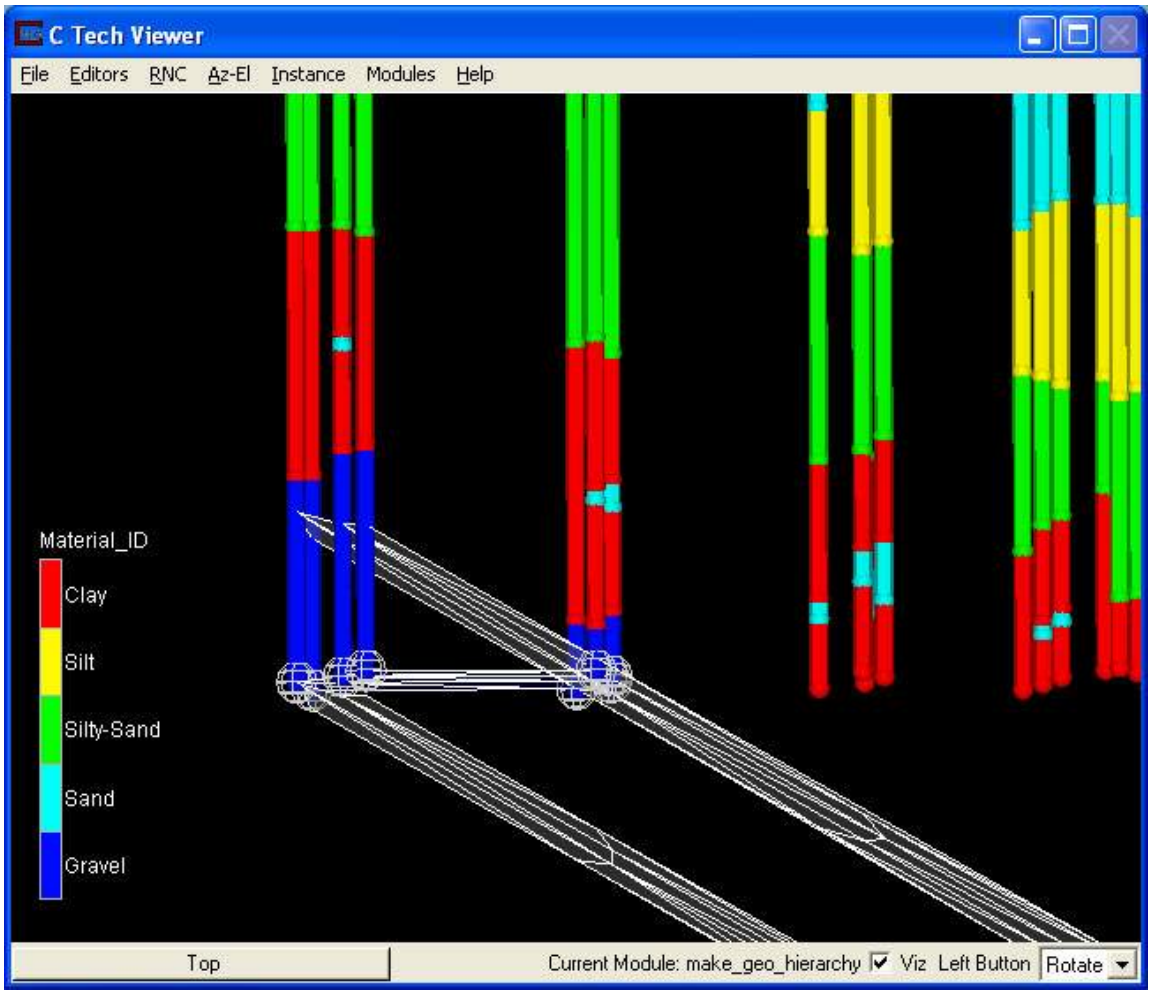


1. Click the *Write Surface* button to write the 6th surface which is the bottom of the upper *Clay* layer.
2. Click the *Write Surface* button to write the 7th surface which is the bottom of the *Sand* lens.



1. Click the *Write Surface* button to write the 8th surface which is the bottom of the lower *Clay* layer.

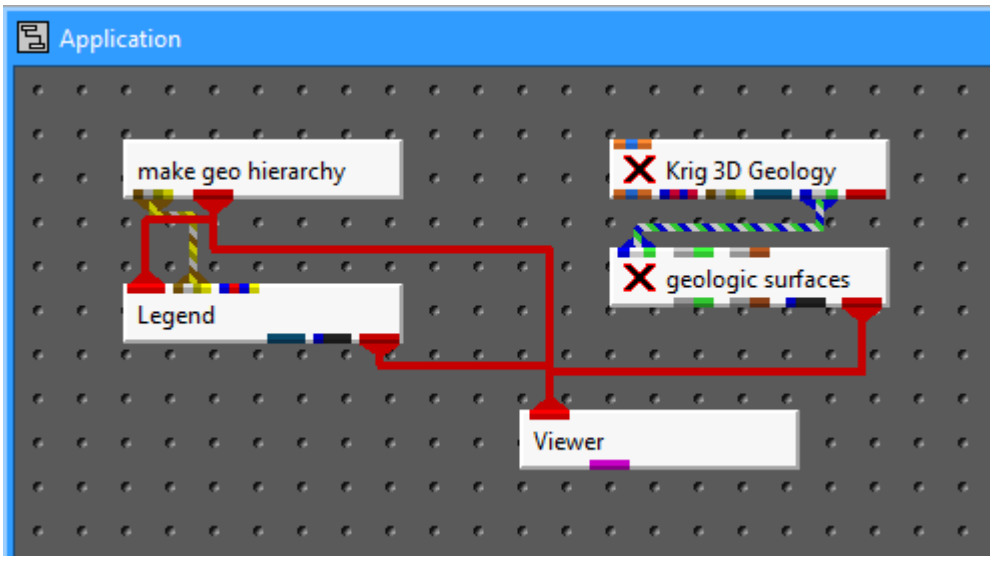
Notice now how the generated surface no longer matches the strike and dip. This is because we have reached the bottom surface and the bottom of our borings which are defining this surface are not really the bottom of this gravel layer. These points are the bottoms of our borings representing the deepest point where geologic information was collected.They are not affeced by the Tolerance value. Since this is our final surface, the "Preserve Bottom" toggle is automatically selected which allows you to use these points to define the bottom of your model (not necessarily the bottom of the last layer".



1. Click the Write Surface button to write the 9th and final *Gravel* surface. The module will then automatically convert the gmf into a geo file.

Visualizing Geologic Hierarchy

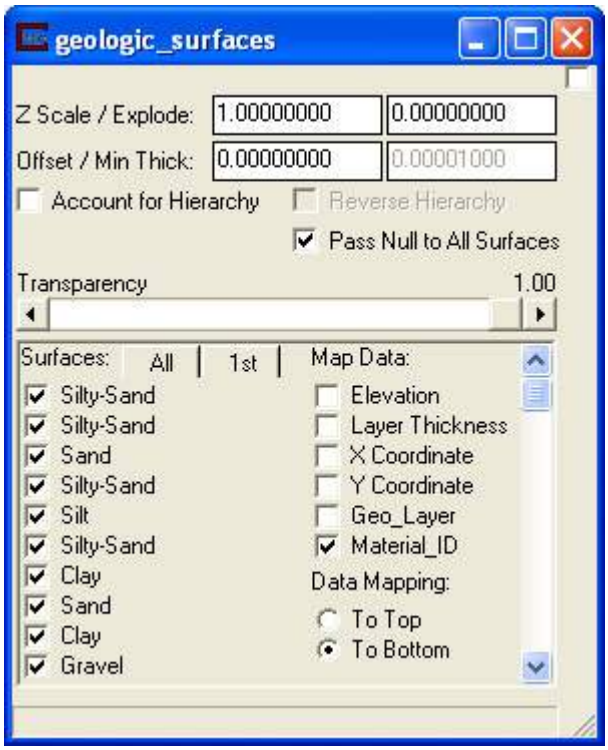
Lets visualize our final product, first instance both the Krig_3D_Geology module and the geologic surfaces module and connect them to the network as shown below.



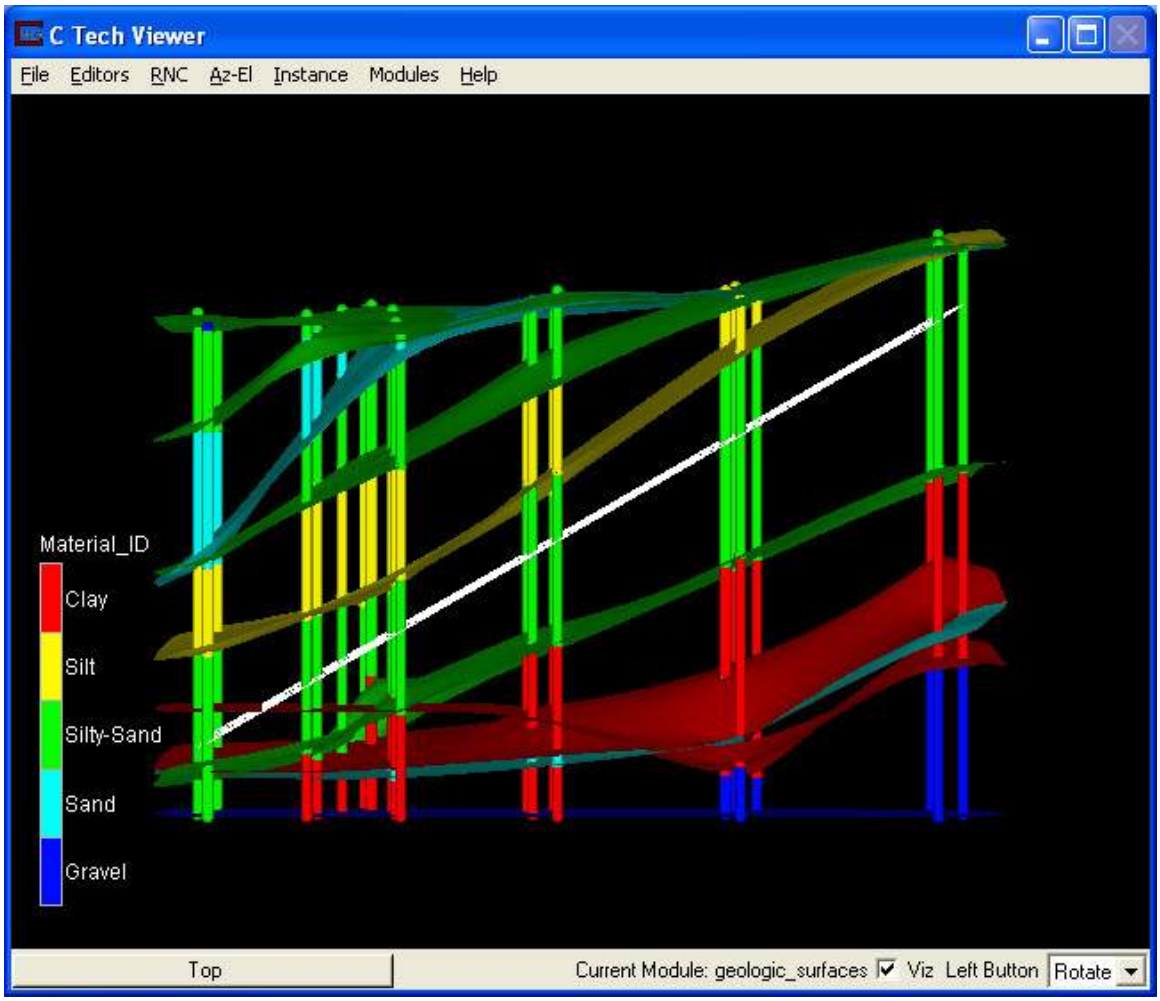
In Krig_3D_Geology select the file "data\pregeology\dipping_strata_lens_pgf_created.gmf" and hit accept. Notice this is not the file you created in the previous workbook sections but a precreated file which will work in the DEMO mode of EVS and MVS. If you are running a licensed copy of EVS or MVS, you can use the file we created in the previous section.

In the geologic_surfaces module hit the All button to display all of the layers. Set the Z Scale to 1.0, turn off the Account for Hierarchy, and turn off all of the Map Data components except for Material_ID.

Your control panel should look like the one below:



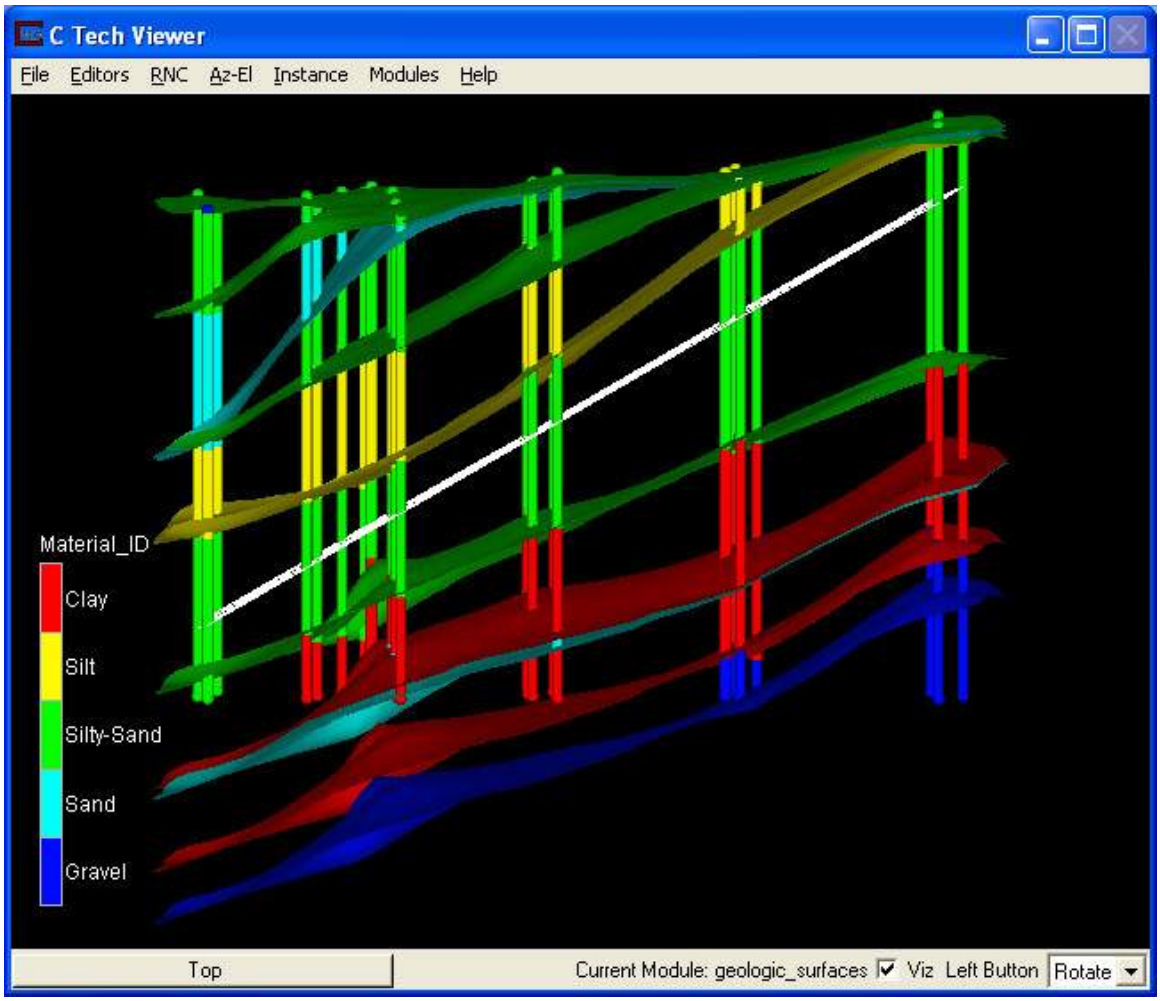
Open up the Az-El panel on the Viewer and set the Elevation to 1 (this allows us to see the bottom surface). You should see the image below



Notice how the second clay surface as it approaches its minimum x extent tends to rise through several of the surfaces above it. This is because the surface only has data defined in the borings on the left of the screen, so kriging it will cause it to approach the mean of elevation as it moves into lower x coordinates.

Next in Krig_3D_Geology select the file "data\pregeology\dipping_strata_lens_pgf_created.geo" and hit accept. Notice this is not the file you created in the previous workbook sections but a precreated file which will work in the DEMO mode of EVS and MVS. If you are running a licensed copy of EVS or MVS, you can use the file we created in the previous section.

You should see the image below:



Notice how the surfaces that contained borings which were dropped from the gmf file follow the topology of the surface above them in the undefined regions. This difference occurs because the default option in Krig_3D_Geology krigs the data in a thickness space when loading a GEO file. This causes the lithology to be honored across areas with less data if you have clean layers. This is the benefit of creating and using a GEO file over a GMF file.

Workbook 4: Three-Dimensional Geologic Modeling

- **Introduction**
- **Modeling Geologic Structures**
- **Krig 3D Geology**
- **Geologic Surface**
- **3D Geology Map**
- **Using plume_shell to Eliminate Pinch-Outs**
- **Conclusion**

Introduction

In this Workbook, we will explore the application of EVS to the modeling of complex geologic structures. Once modeling of geologic structures is understood, the following workbooks extend kriging of analyte (e.g. chemistry) data to three-dimensions and will address mapping analyte (e.g. chemistry) into geologic structures.

This workbook assumes that you understand the subjects covered previous workbooks including:

1) Network Editor Basics such as instancing, deleting, connecting and disconnecting modules.

2) Viewer basics such as setting views, rotations, scaling, translation, object selection and object property modification.

3) Two dimensional kriging of analyte (e.g. chemistry) data.

If any of these topics are unfamiliar, we recommend you review the previous workbooks before beginning this lesson.

Modeling Geologic Structures

Three dimensional modeling of geologic structures using EVS is straightforward and provides powerful capabilities to assess complex underground formations. EVS provides a number of tools for the investigation and modeling of geologic structures. The relevant modules specific to geologic modeling are:

1) Krig_3D Geology - Uses kriging to interpolate between borings and create continuous geologic surfaces.

2) Spline_Geology – Uses the Thin Plate Spline technique to interpolate between borings and create continuous geologic surfaces.

3) Indicator_Geology – Uses indicator kriging to assign geologic materials to cells in a model grid. The use of this module is covered in the Advanced Geologic Modeling Concepts workbook.

4) 3D_Geology_Map - Uses the output of Krig_3D Geology or Spline_Geology to create 3D geologic models from geologic surfaces.

5) Geologic_Surface - Used for visualizing the surfaces created by Krig_3D Geology or Spline_Geology.

6) Fence_Geology – Used with Krig_Fence to build 2.5D fence diagrams from boring data.

Spline_Geology, Indicator_Geology and Fence_Geology will not be covered in this workbook. Spline_Geology and Indicator_Geology are covered in Advanced Geologic Modeling Concepts workbook and Fence_Geology (and its alternatives) are covered in Fence Diagrams workbook.

The first and second modules above also provide inputs to analyte (e.g. chemistry) kriging modules to allow mapping of chemical concentrations to geologic structures. This topic will be covered in subsequent workbooks. Additionally, a very large number of other EVS modules provide useful and important capabilities related to geologic modeling, but their functionality is not restricted to geologic structures.

Throughout this workbook, we will use a few terms which may have slightly modified meaning in this context. They are:

Domain: The extent of the model. This is usually defined by the convex hull or rectilinear bounding box which encloses the borings from which geologic observations were made.

Layer: An individual geologic unit or layer.

Krig_3D_Geology

Krig_3D_Geology is the primary module for creating three-dimensional models of geologic structures. In the following topics we will create a model of complex geologic structures.

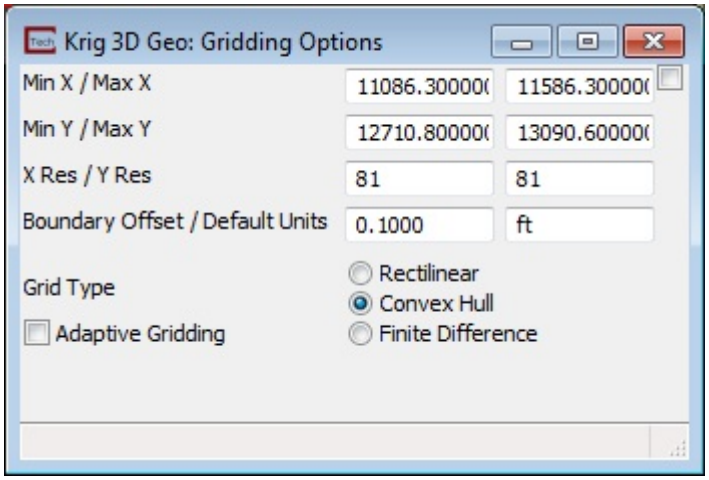
First, let's start our copy of EVS or MVS. We will then instance Krig_3D_Geology, which is the first module in the File sublibrary.

Double click on the Krig_3D_Geology module in the Application area and the following window should appear on your screen.

The Krig_3D_Geology module uses kriging to interpolate and extrapolate between borings to create a continuous geologic surface. Let's go through

some of the parameter options for this module and then we will select a file and run the module. For this workbook, we will look at the Gridding Options and some of the Kriging Parameters.

Select Gridding Options and the following window should appear. This window has the most commonly changed parameters and many of these parameters affect and override settings in downstream modules. The parameters with values of 0.0 are automatically filled in by the expert system during the kriging process if they have not been changed. Leaving these at their defaults will typically provide good results.



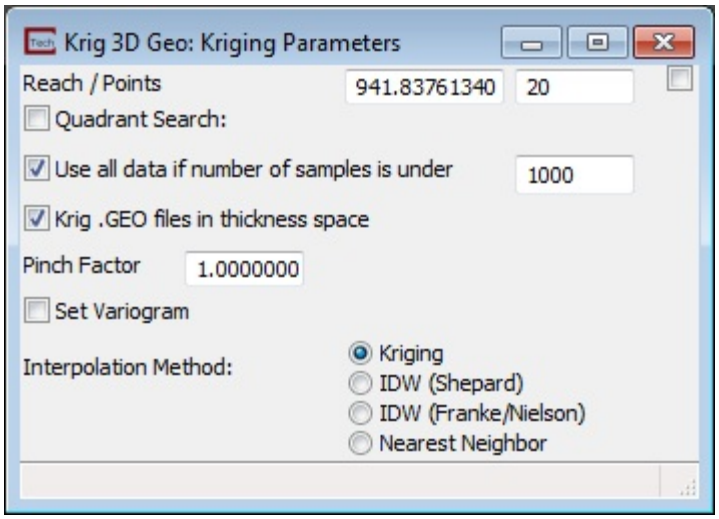
The first four parameters define a rectangular region which is the "Rectilinear" domain (in the x-y plane). These parameters will be automatically defined by your geology data file if the defaults are used. These values are ignored if Convex Hull (default) is used.

X Res and Y Res are the most commonly changed parameters. These determine the resolution of the grid in the x-y plane. The default values of 41x41 give a reasonably fine model, but higher resolutions are generally used for final analyses. Remember that the compute time and system resource requirements are dependent on the product of the x, y, (and z) resolutions.

Boundary Offset is the convex hull boundary offset. This is similar to the offset concept in AutoCAD (tm). It extends the domain beyond the convex hull of the sample locations by a percentage of the x-y extent. A value of .1 puts an offset of 10% of the extent on all sides.

Grid Type: This is another commonly changed parameter. Rectilinear provides a domain with a grid which is composed of uniform sized rectangles and has straight sides. Rectilinear also allows for subsetting or extending the domain with the Min/Max X/Y parameters. Convex hull confines the kriging (domain) to a region defined by the sample locations. Finite Difference provides a domain with a grid which can be composed of variable sized rectangles and has straight sides. This type of grid can also be rotated to any x/y orientation.

Now open the Kriging Parameters window as shown below.



Reach defines the distance from a kriged point to all data samples which will be used in the matrix solution. Larger reaches will produce smoother distributions.

Points is the maximum number of samples points which will be considered in the matrix solution when predicting the value at a node location. Larger numbers will produce smoother distributions, but dramatically slow down the kriging process.

Now that we have discussed these parameters, close the Kriging Parameters window. We will leave all of the default settings for this workbook.

Read adatafile:

a. Go back to the main menu window (labeled Krig_3D_Geology) and close all open subwindows (those with the toggle checked) by clicking on any that have an check mark.

b. Select the **Read DataFile** button.

c. Select *initial_soil_investigation_subsite.geo* from the Files list box.

d. Select the **Open** button.

e. Select the Accept All Current Values button.

Krig_3D_Geology reads the geology data and begins the kriging process. In less than one minute, it calculates the surfaces for the grid we selected. While it runs, Krig_3D_Geology prints messages to the console such as percentage completion. When it is done, the console messages look like this:

```
OM(Root) -> File check passed
File check passed


Reading .geo data from
c:\ctech\data\geology\initial_soil_investigation_subsite.geo
*** Surface 0 ***
Computing Pairs: Done
276 Pairs in Semivariogram model: Range = 895.126556 Sill =
14.029811
```

```
1681 points Kriged for surface 0 using 85 matrices in 0.00
minutes
*** Surface 1 ***
Computing Pairs: Done
276 Pairs in Semivariogram model: Range = 166.578261 Sill =
4.763116
1681 points Kriged for surface 1 using 85 matrices in 0.00
minutes
*** Surface 2 ***
Computing Pairs: Done
276 Pairs in Semivariogram model: Range = 173.335077 Sill =
7.211553
1681 points Kriged for surface 2 using 85 matrices in 0.00
minutes
*** Surface 3 ***
Computing Pairs: Done
276 Pairs in Semivariogram model: Range = 156.972936 Sill =
3.352522
1681 points Kriged for surface 3 using 85 matrices in 0.00
minutes
*** Surface 4 ***
Computing Pairs: Done
276 Pairs in Semivariogram model: Range = 156.972936 Sill =
7.062694
1681 points Kriged for surface 4 using 85 matrices in 0.00
minutes
*** Surface 5 ***
Computing Pairs: Done
276 Pairs in Semivariogram model: Range = 1236.354406 Sill =
8.695820
1681 points Kriged for surface 5 using 85 matrices in 0.00
minutes
Layer 1 is Fill with material color 2
Layer 2 is Silt with material color 1
Layer 3 is Clay with material color 3
Layer 4 is Gravel with material color 5
Layer 5 is Sand with material color 4
```

Krig_3D_Geology produces a grid (in the x-y plane) and maps the geologic surfaces defined in the input file to that grid. Its output is a set of surfaces, one for the top ground surface and one for the bottom of each geologic layer in the domain.

This one module network is not yet particularly interesting. Note that one of the output ports for this module is tri-colored and different than anything we

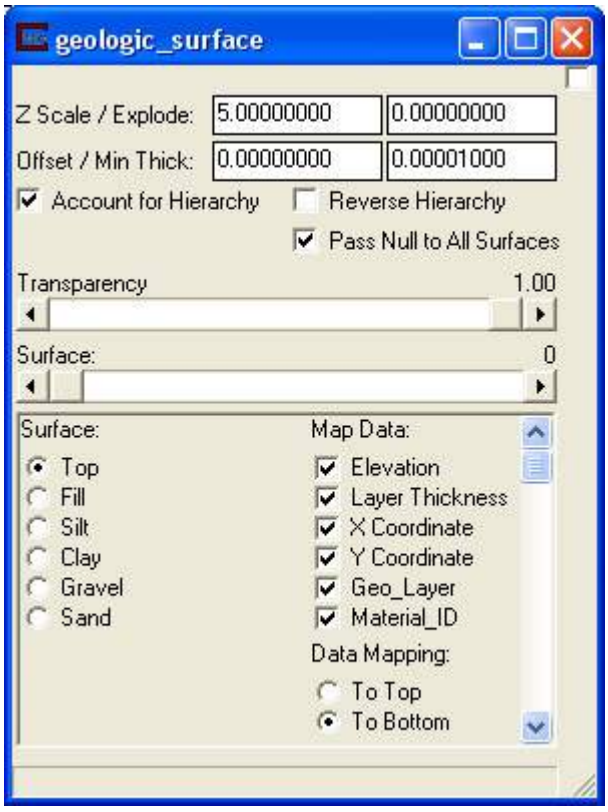
used in Workbook 1. We need to find modules which can receive this output in order to process and display the results of this module.

Geologic Surface

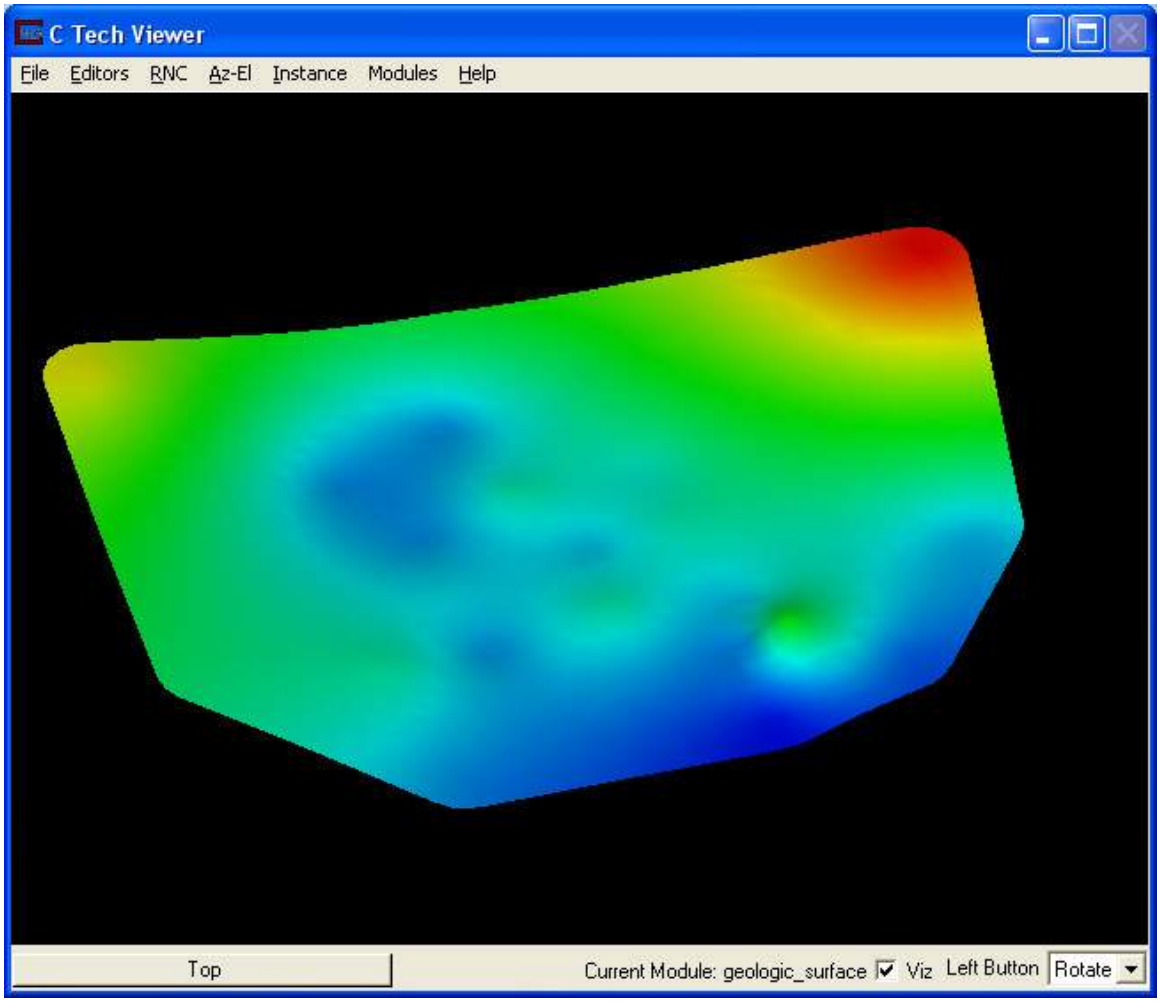
Instance geologic_surface from Processing sublibrary, and connect the tri-colored output port on Krig_3D_Geology to its matching input port.

Instance a Viewer and connect the red output of geologic_surface to create the following network.

The Modules pull-down menu now has a pull-down command to access the user interface for Krig_3D_Geology and geologic_surface. Select geologic_surface and the following window should appear on your screen.



Click on Az-El on the Viewer menu and choose scale 1.10, Elevation 40 and 195 Azimuth to get a view of the Top Geologic Surface. This network will allow you to visualize any of the geologic surfaces and is useful for debugging geologic layer definitions and visualizing individual layer surfaces. It also provides a means to display other types of geologic surfaces such as water tables (which may vary over time). Use the slider, in the Geologic Surface module, to select from the six surfaces (0-5) for this model. Z Exaggeration allows you to magnify the z-coordinate and the Surface Z Offset provides a means to move surfaces. When looking at the first surface (Surface 0/Top), your Viewer should look like this.

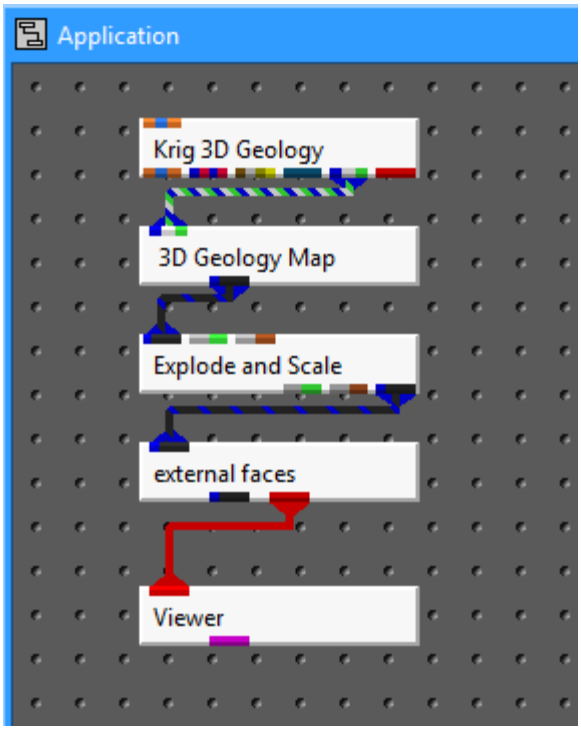


By default surfaces will be colored (from blue to red) based on the variation in the z-coordinate (elevation) of the individual surface. By unchecking Elevation in the Map Data section, we can change how the surface is colored. The surface will always be colored by the first data component mapping through. When visualizing water tables or when no color (or a specific solid color) is wanted, uncheck all of the Map Data toggles. The color (and other properties) can be edited using the Object Selector and Object->Advanced Settings editor in the Viewer.

3D Geology Map

Delete the Geologic Surface module and instance 3D_Geology_Map (in Processing Sublibrary) and connect the tri-colored output port on Krig_3D_Geology to its matching input port.

Instance Explode_and_Scale (top of Display) and external_faces (top of Subsetting). Connect the blue/black output of 3D_Geology_Map to Explode_and_Scale and connect its blue/black output to external_faces. Connect the red output of external_faces to the Viewer to create the following network.



Connecting 3D_Geology_Map should result in the following message being printed to the console:

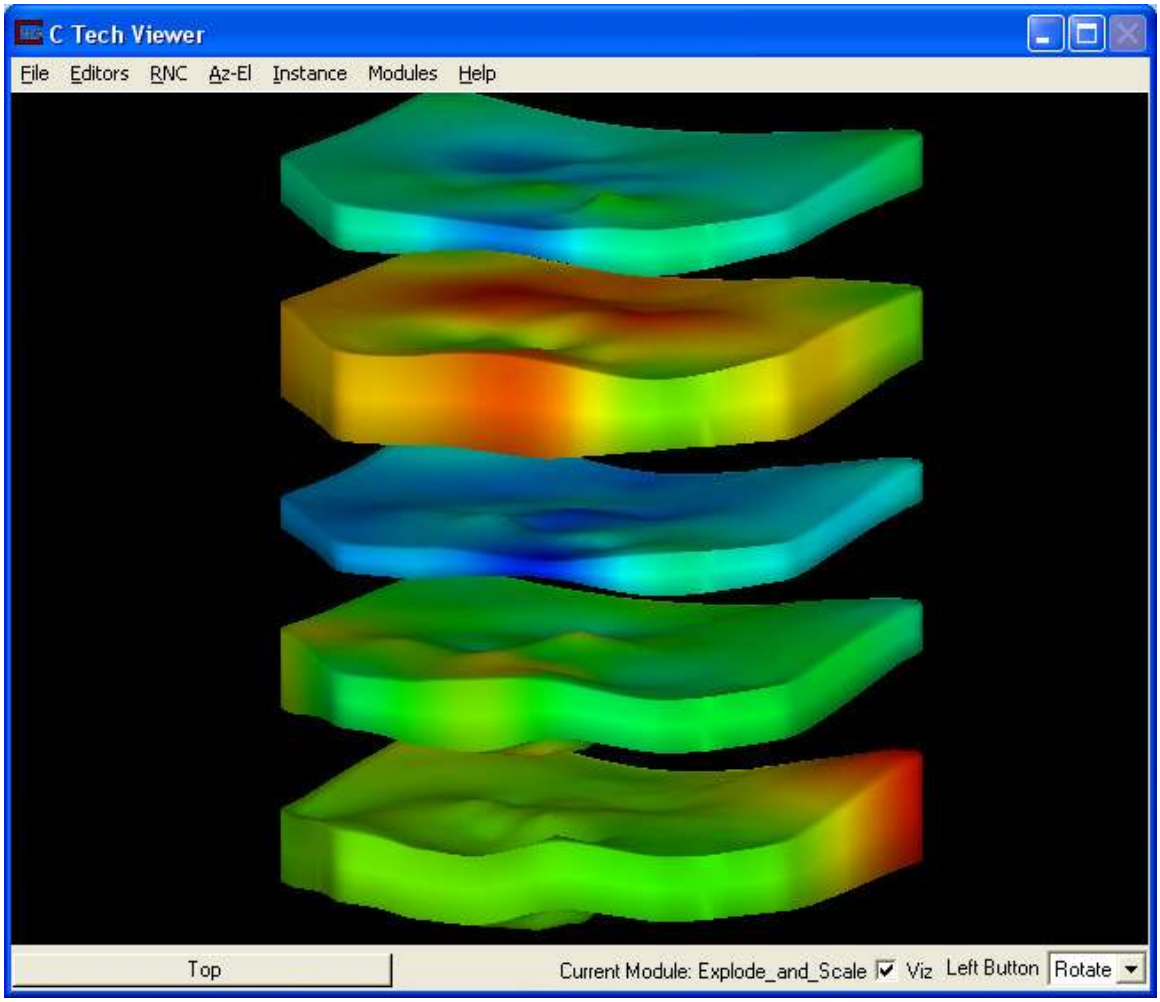
```
_3D_Geology_Map:
Layer Thickness: Min 2.260 Max 20.420
X Coordinates: Min 11054.905 Max 11617.664
Y Coordinates: Min 12679.596 Max 13121.776
Geo_Layer: Min 0.000 Max 4.000
Elevation: Min -55.945 Max 11.407
Material_ID: Min 1.000 Max 5.000
Depth: Min 0.000 Max 59.395
```

This data provides the ranges for all of the nodal parameters associated with the solid model. Since Layer Thickness is the first parameter, the model is colored according to layer thickness by default. Note that this model has no layers with zero thickness (i.e. no pinched out areas).

The Modules list now has a pull-down command to access the user interface for Krig_3D_Geology, _3D_Geology_Map, external_faces, and

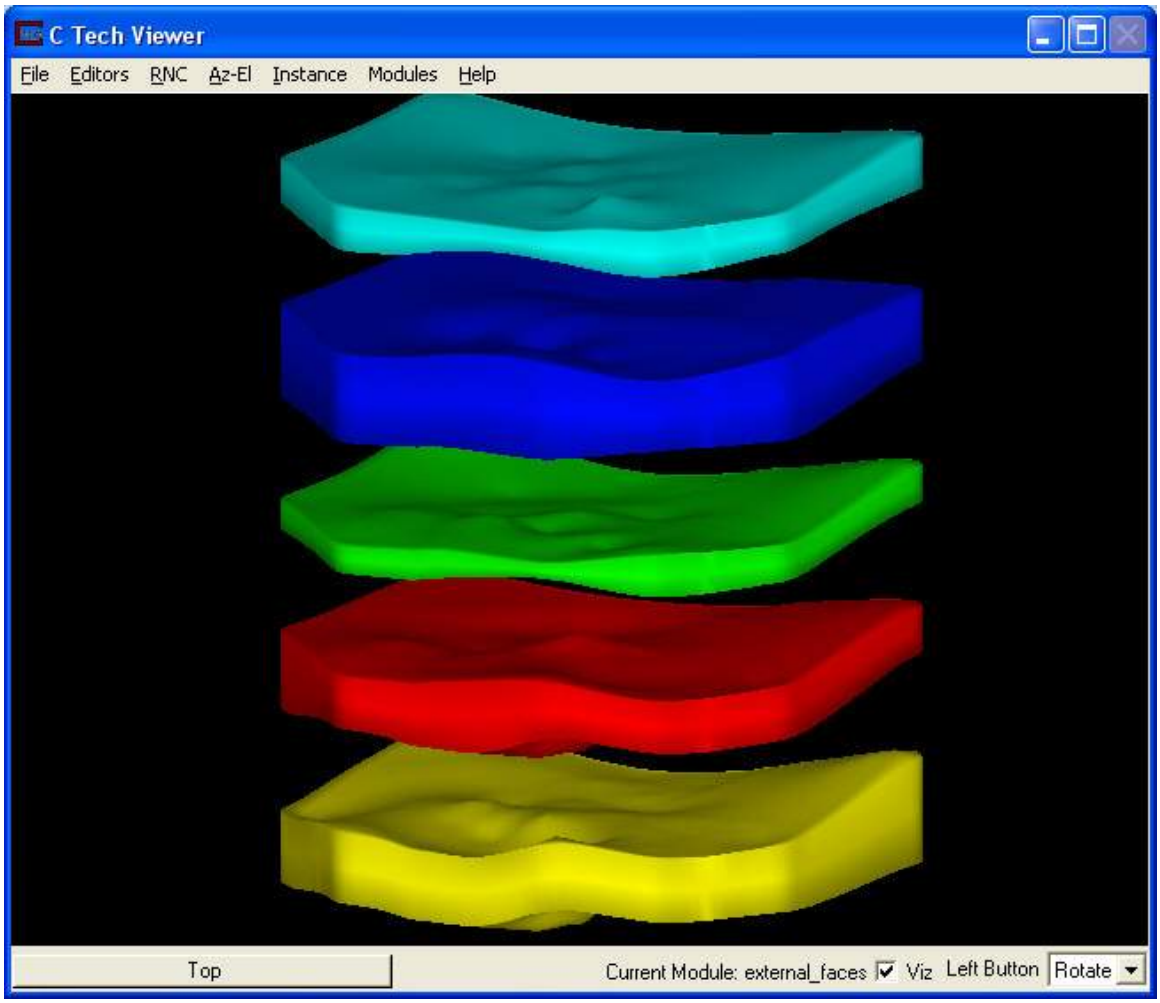
Explode_and_Scale. 3D_Geology_Map produces a solid model using the geologic surface data created by Krig_3D_Geology. Its interface has options for proportional gridding and grid resolution in the Z direction. You can use Explode_and_Scale to adjust the exaggeration and explode distance of the layers. For now, leave the default values.

Click on Az-El on the Viewer menu and choose .70 Scale, 10 degree Elevation and 150 Azimuth to get a view of the solid model. Once you've done this, hit the multi-colored button (Reset/Normalize/Center) in the Az-El panel. You should have the following view in your Viewer.



Open the user interface for external_faces in the Modules pull-down menu. You can choose which of the nodal data components is used to color the model.

If you choose Material_ID as the Data component you will be coloring according to the geologic material number on line 3 of the .geo file. The order of those numbers were 2, 1, 3, 5 and 4 which gives layers colored as:

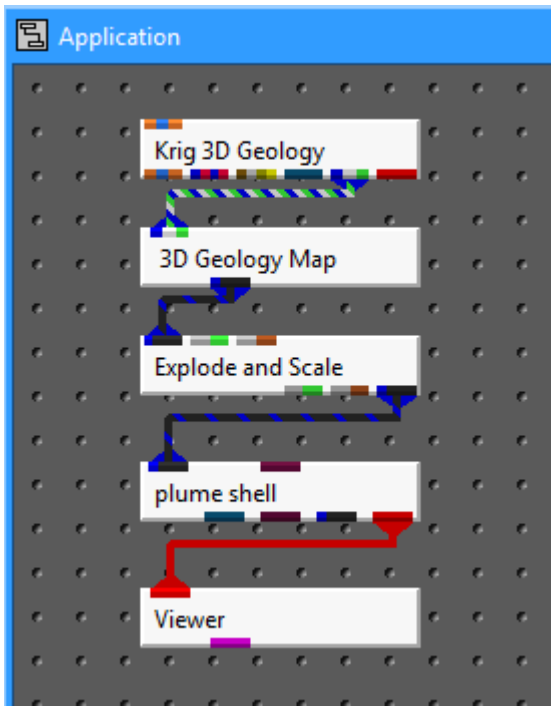


To obtain more appropriate geologic colors we must edit the datamap of external_faces. For instructions on how to do this see the Datamap_Editor.

Using plume_shell to Eliminate Pinch-Outs

When dealing with more complex geometric structures with pinched-out regions, we need a method to eliminate the portions of the solid model where the thickness of the geologic layers is effectively zero. To do this, we use the plume_shell module (Display). plume_shell creates a subset of the solid model based on one of the nodal data components. Since the first nodal parameter is thickness, we can easily remove thin portions of layers which we want to be pinched-out..

To demonstrate this capability, we will modify our network and read a different geology file. The easiest way to make sure that we reset all of the default parameters is to delete everything but the Viewer and recreate the following network.



First, we want to modify a few of the parameters in Krig_3D_Geology:

1. Under Gridding Options: change Boundary Offset to 0.01. This parameter adjusts the distance of the Convex Hull boundary. The distance is a percentage of the diagonal extent in the X-Y plane. The setting of 0.01 is 1%. For example: if data extent is 100 in x and 100 in y, diagonal distance is 144. It will make the convex hull offset 1.44 total by offsetting 0.72 on all sides.
2. Under Kriging Options: turn off Krig .GEOfiles in thickness space. Although this is on by default, this particular dataset gives us a better result with this option off. Having the ***Krig .GEO files in thickness space*** toggle off causes GEO files to be kriged like GMF files. Each surface get kriged independently of the other surface instead of being kriged in thickness space. This only applies to GEO files without the $W/$G flags.

Next, we will read a data file and krige the geology:

a. Go back to the main menu window (labeled Krig_3D_Geology) and close all open subwindows (those with the toggle checked) by clicking on any that have an "x".

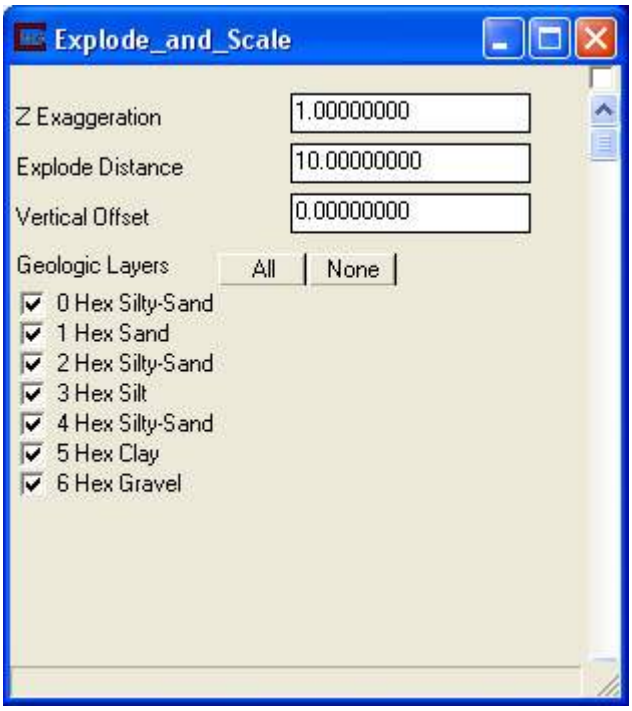
b. Select the **Read GEO File** button.

c. Select *7_layers_dipping.geo* from the Files list box.

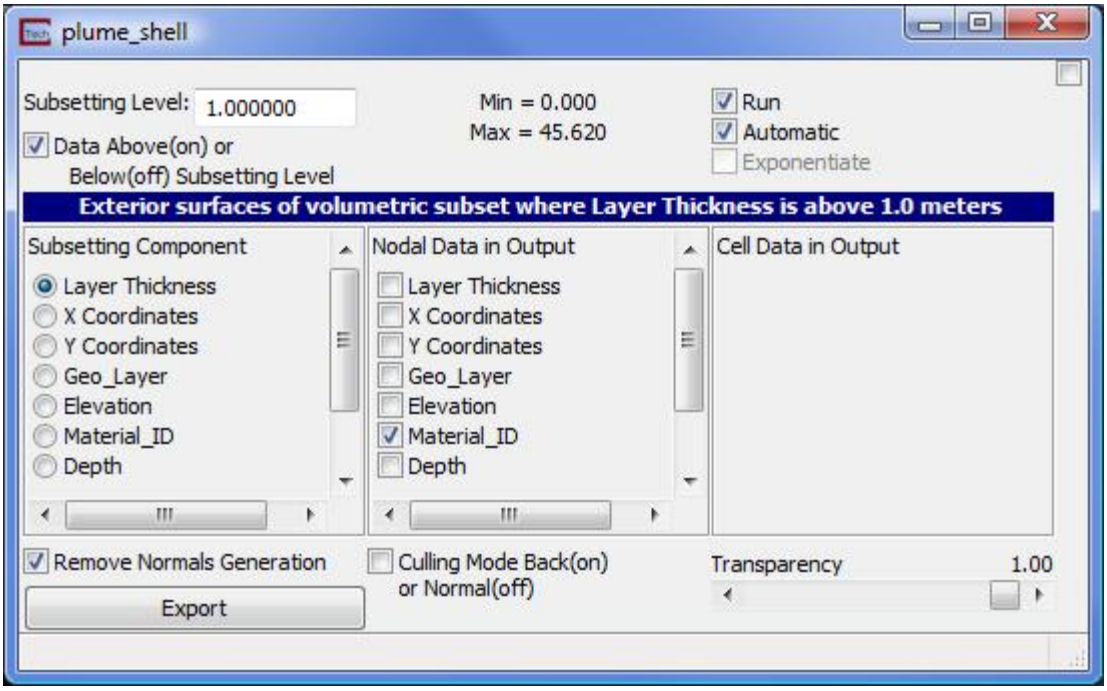
d. Select the **Open** button.

e. Select the Accept Current Values button.

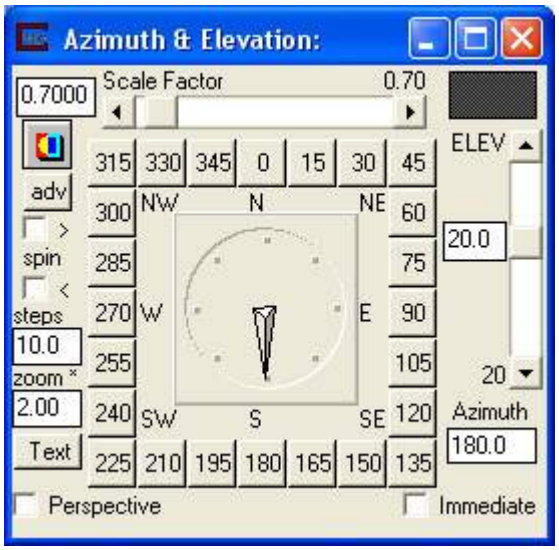
Select the user interface for Explode_and_Scale and set the parameters to match:



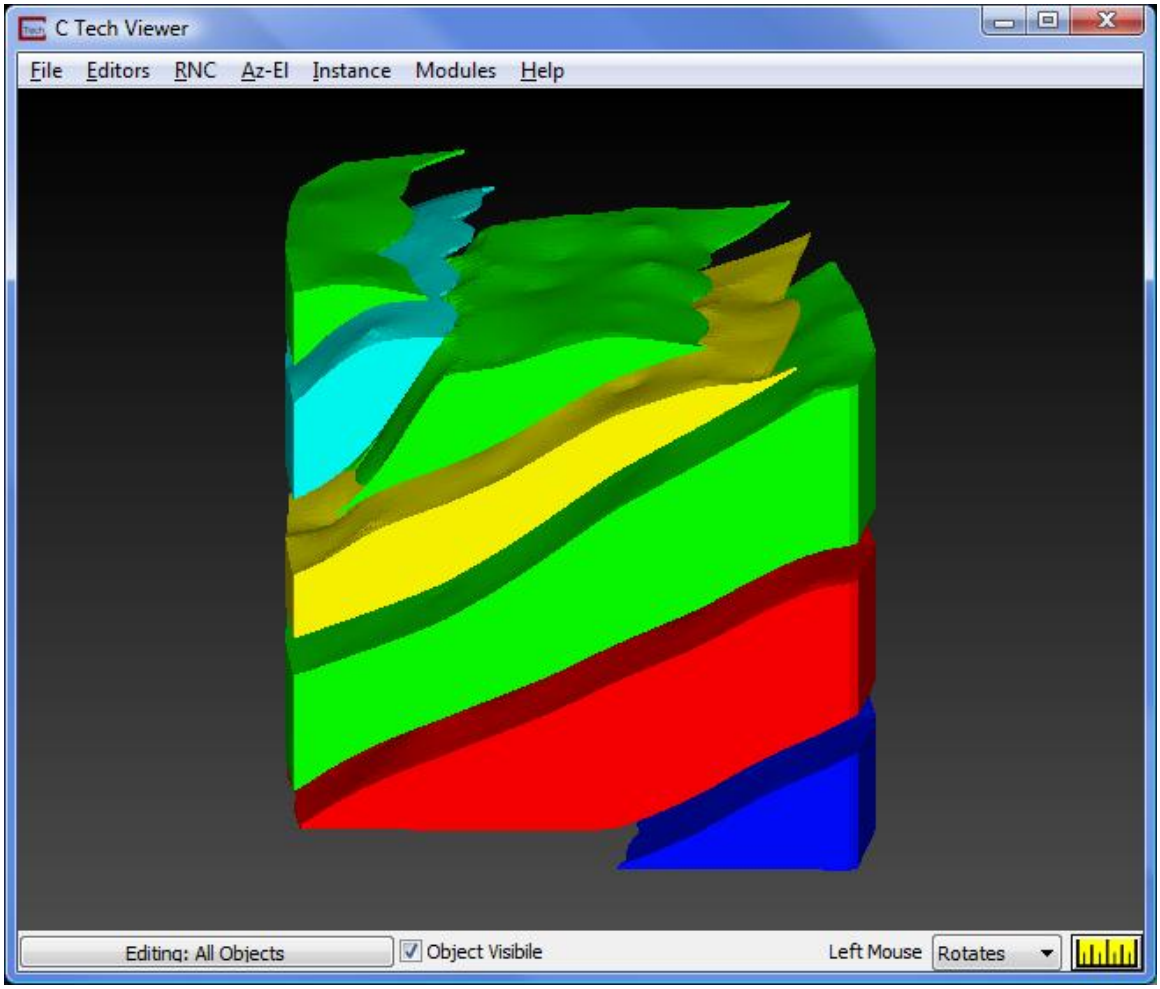
Select the user interface for plume_shell and set the paramters to match:



Click on Az-El on the Viewer menu. Press the multi-colored button (Reset/Normalize/Center), then set the parameters to match:



to get a view of the solid model. You should have the following view in your Viewer.



Geology Workbook Conclusion

In this Workbook, we explored the application of EVS to the modeling of complex geologic structures. You should now be able to take simple ASCII input files determined from boring logs and produce three-dimensional models of geologic structures. The following workbooks will extend this to include chemical (or property) mapping to geologic structures.

Workbook 5: Three-Dimensional Kriging

Introduction

In this Workbook, we will explore the application of EVS to parameter estimation using kriging and mapping of parameters to complex geologic structures.

This workbook assumes that you understand all of the subjects covered in the previous workbooks including:

1) Network Editor Basics such as instancing, deleting, connecting and disconnecting modules.

2) Viewer basics such as setting views, rotations, scaling, translation, object selection and object property modification.

3) Two dimensional kriging of analyte (e.g. chemistry) data.

4) Geologic Hierarchy and preparing geology input files

5) The function of the Krig_3D_Geology Module.

6) The function of the plume_shell Module.

You must also be comfortable using the Modules pull-down menu to access the user interface for modules which have been instanced.

If any of these topics is unfamiliar, we recommend you review the preceding workbooks before beginning this lesson.

Throughout this workbook, we will use a few terms which may have slightly modified meaning in this context. They are:

Domain: The extent of the model. This is usually defined by the convex hull or rectilinear bounding box which encloses the borings from which chemical measurements or geologic observations were made.

Layer: An individual geologic layer, or a portion of an actual geologic layer.

Parameter Estimation: The process of assigning values to nodes on a grid (rectilinear, convex hull, or finite difference) based on the values of nearby sparse measured data. EVS performs parameter estimation using expert system driven kriging modules.

Three-Dimensional Kriging

Three dimensional parameter estimation using EVS's kriging modules is straightforward and provides powerful capabilities to assess complex underground distributions of chemicals, ore bodies or other geologic properties (head, porosity, conductivity, etc.). The relevant modules specific to three dimensional kriging are:

1) Krig_3D

2) Krig_3D_Geology

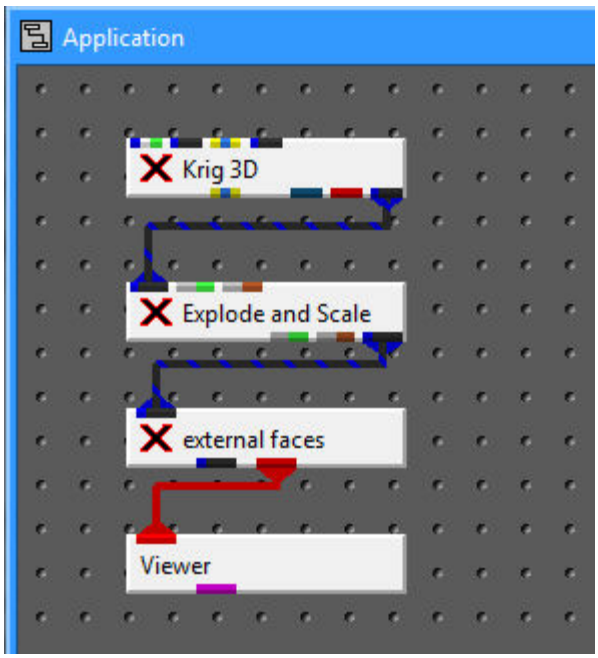
3) Spline_Geology

4) Indicator_Geology

The second and third modules provide inputs to Krig_3D to allow mapping of chemical concentrations to geologic structures. A very large number of other EVS modules provide useful and important capabilities related to visualization and analysis of the output of Krig_3D, but their functionality is not restricted to its output. The Indicator_Geology module performs 3D kriging of geologic data.

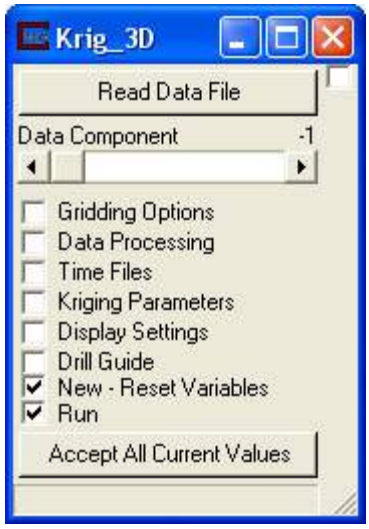
A Simple Network for 3D Kriging

Using the methods covered in Workbook 1 and 2, instance the modules and connect them to form the network shown in the figure below.



In this network, Krig_3D produces the model, Explode_and_Scale allows us to exaggerate the "Z" coordinate (and later to explode layers) and external_faces allows us to visualize the external surfaces of the model.

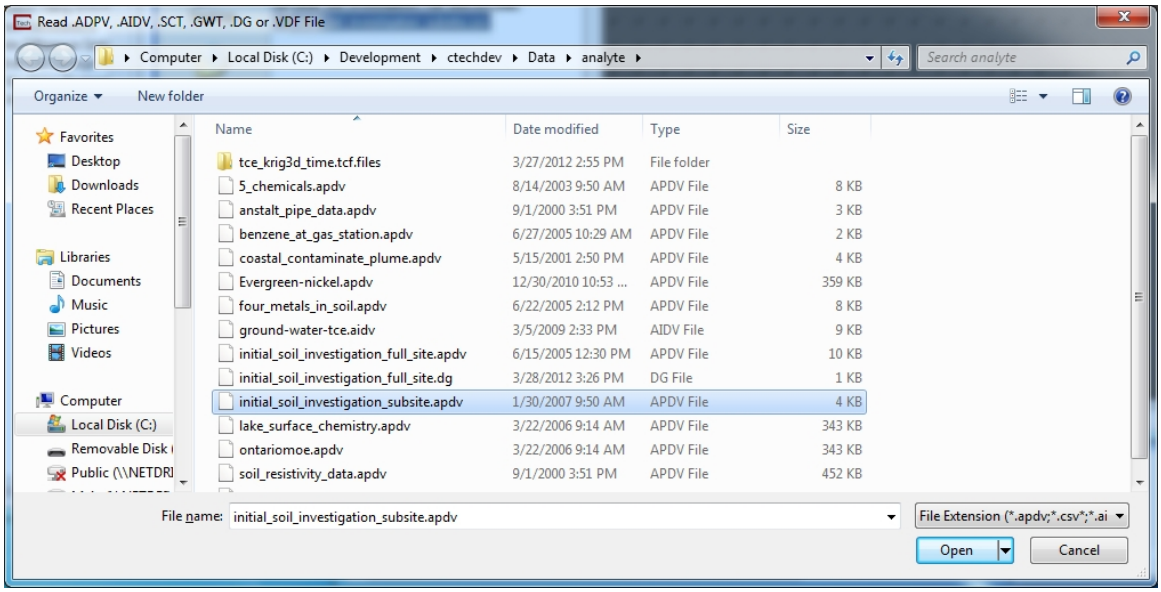
Before we run this network, let's modify a few of the parameters. Begin by selecting Krig_3D from the Modules pull-down menu to reveal the Krig_3D Main menu shown below.



Select*Data Processing* so that we may change a few values.

Now modify *Post-ClipMax:* in the Data Processing window to be 1.000000e+003 (1000).

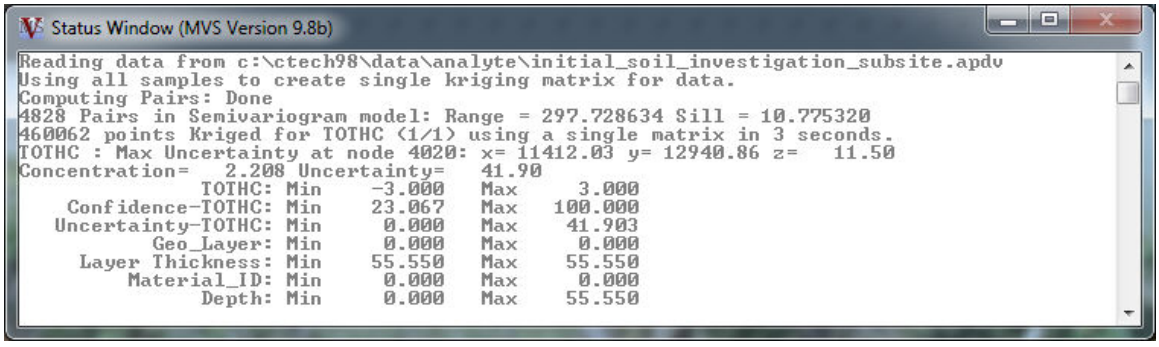
Close only the *Data Postprocessing* subwindow (we'll want to look at the Gridding Options after the module runs) and click on the Read Data File button to reveal the file browser shown below.



Select the file shown (initial_soil_investigation_subsite.apdv) by double clicking, or by clicking once and then selecting Open.

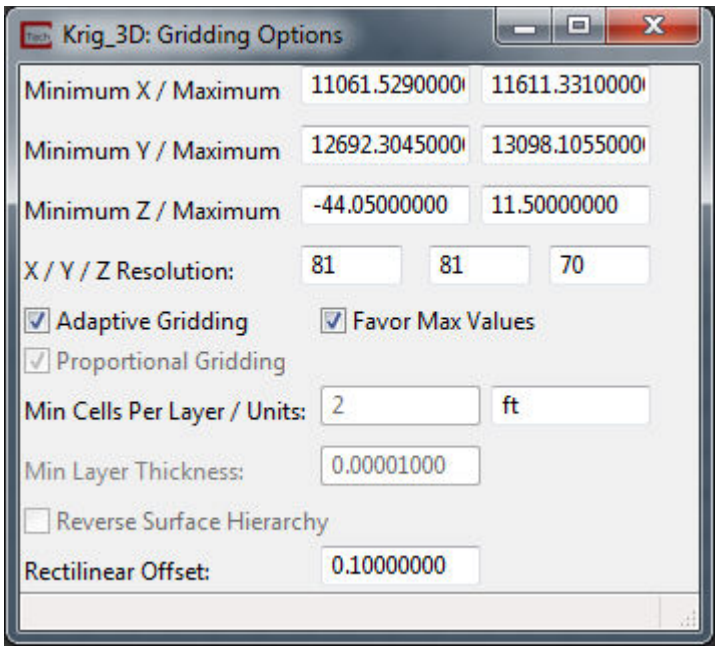
Click on the "Accept All Current Values" button and the network will run.

As Krig_3D runs, it will print status messages to the console. These include time to completion estimates. On most modern computers, this network will run almost instantaneously. When it is complete, the console will show the following messages.



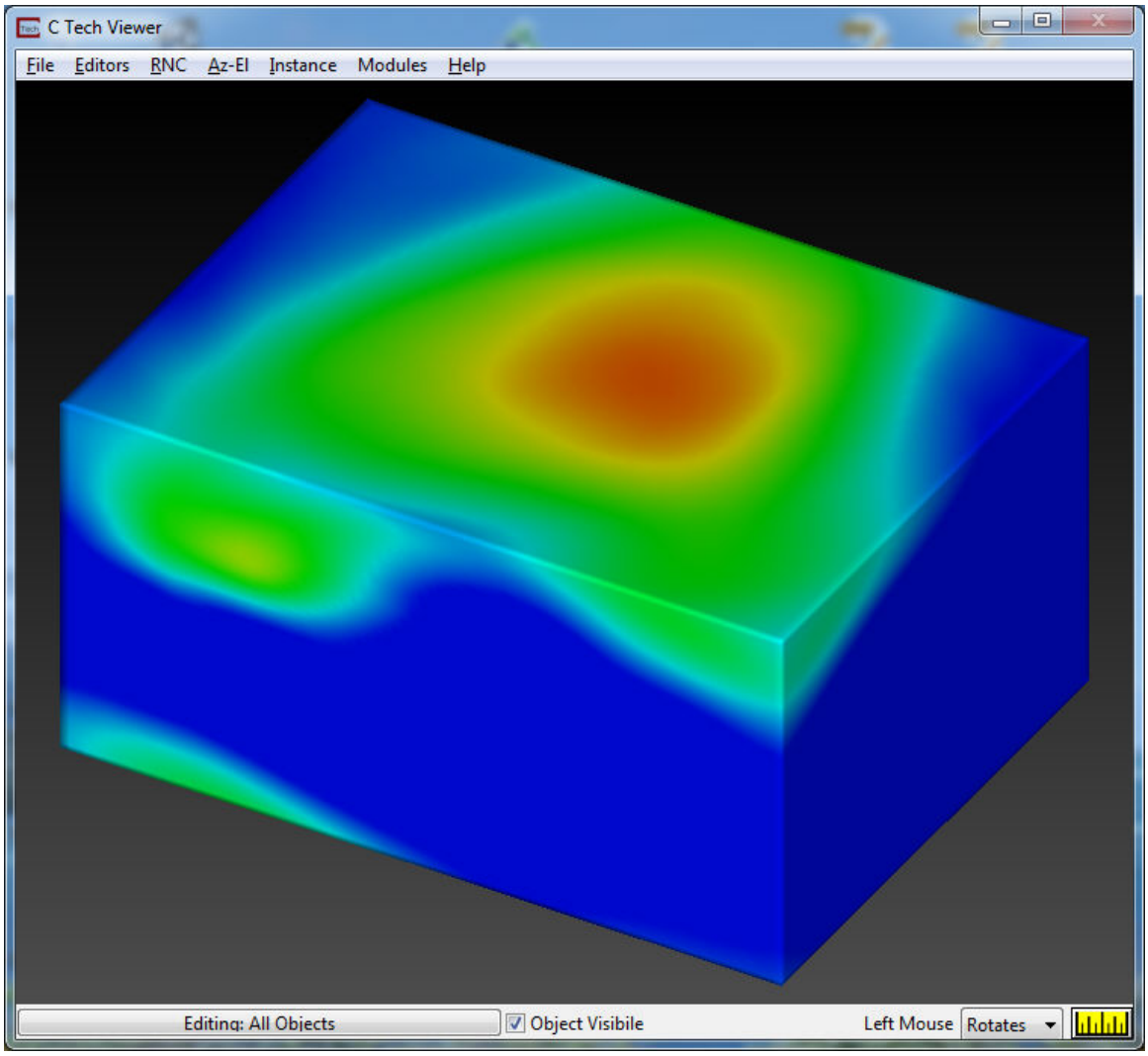
If we now open *Gridding Options,* those parameters which were initially 0.00 (zero default values) have been updated. The Min/Max X-Y-Z values have been set to the extents of the analyte (e.g. chemistry) file. Note that the maximum Z value is now determined using the Top of boring elevation from the highest boring, rather than the elevation of the highest sample. It is nearly always best to use geologic input, however when there is none, this will give you a grid that goes to the highest ground surface elevation.

The updated gridding values are shown in the figure below.



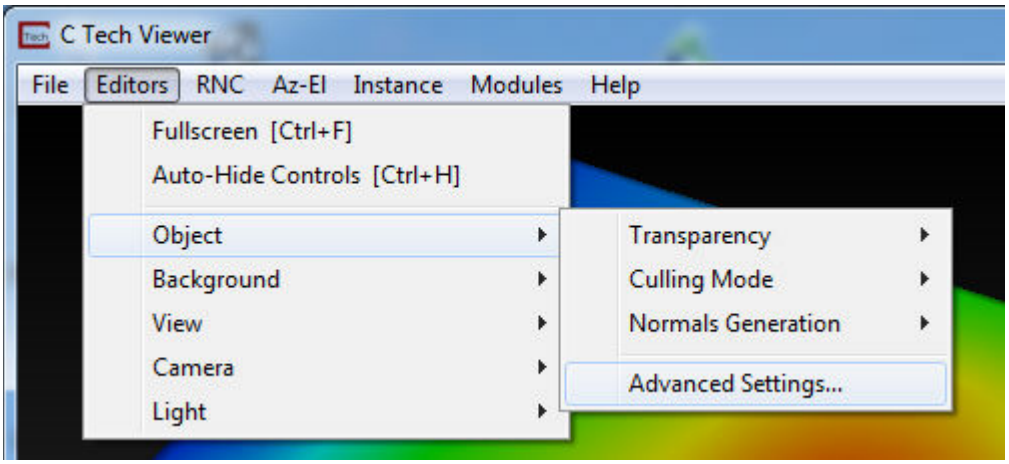
Other values which were initially 0.00 (zero default values) such as Reach have also been set in Kriging Parameters. For detailed descriptions of all of the parameters in Krig_3D (or any other EVS module) please refer to the Module Libraries references in the manual or on-line help....OR...select any module, click the right mouse button and choose help.

At this point the network has run to completion and a visualization of the external faces of the domain has been produced. We now see a top view of the model in the Viewer. Choose Az-EL and set the Elevation to 35 degrees, Scale to 0.90 and click on the 150 degree Azimuth button. At this point your Viewer should look like this:

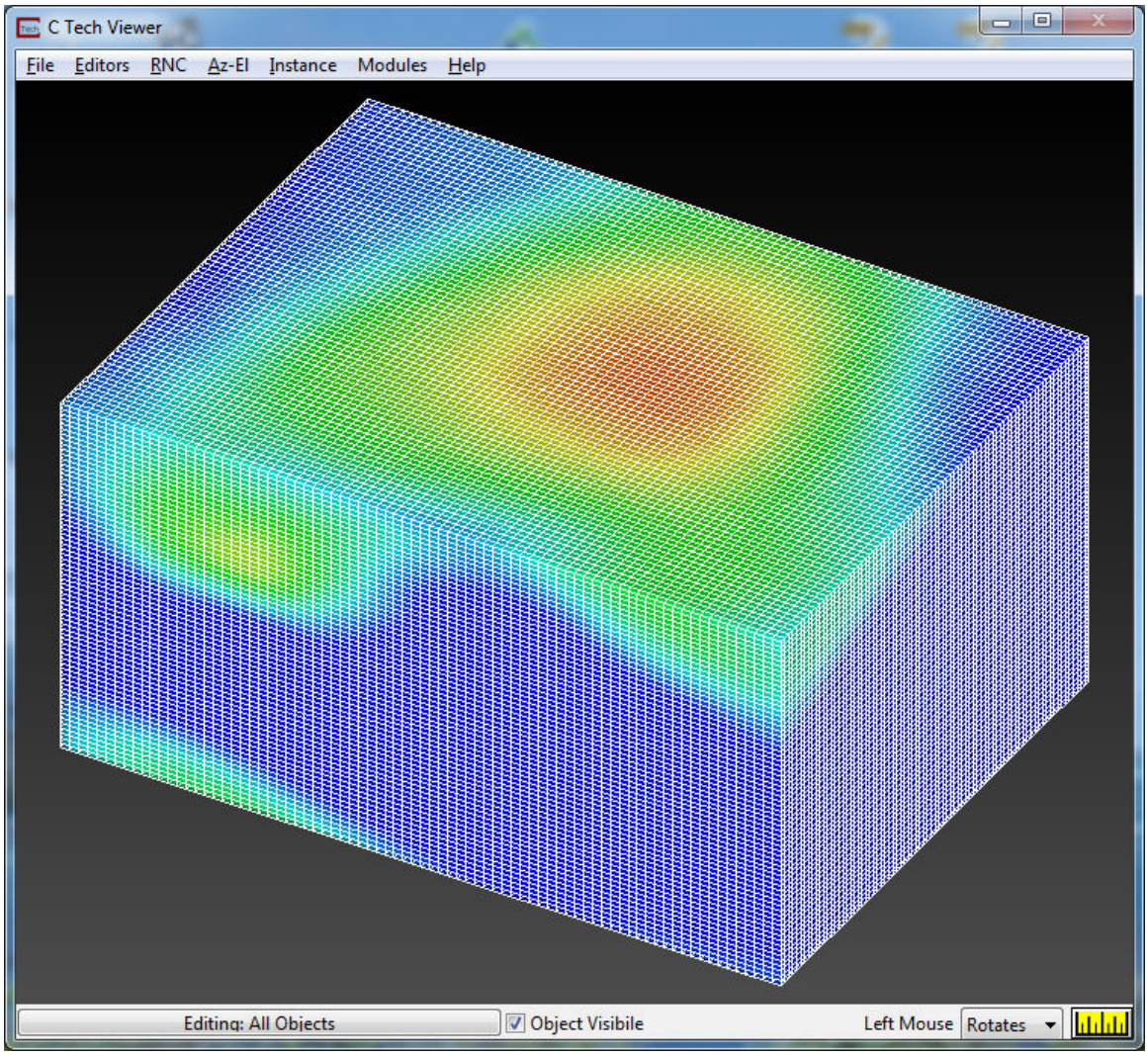


Notice that the domain is a flat-surfaced "brick". Since no geology data was used to produce the output, the extents of the domain were determined by the extents of the input file.

If we want to see the grid used to create this model, go to the Viewer pull down menus and select Editors->Object->Advanced Settings



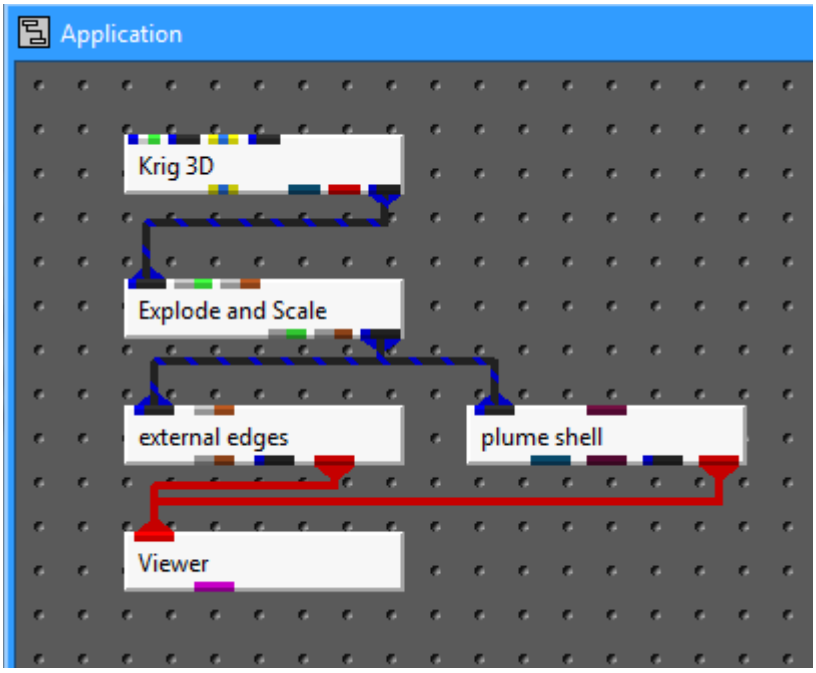
and in the *ObjectSetting to Edit:* select **Rendering Modes**. Change Line_Rendering to Regular. This should display the picture below.



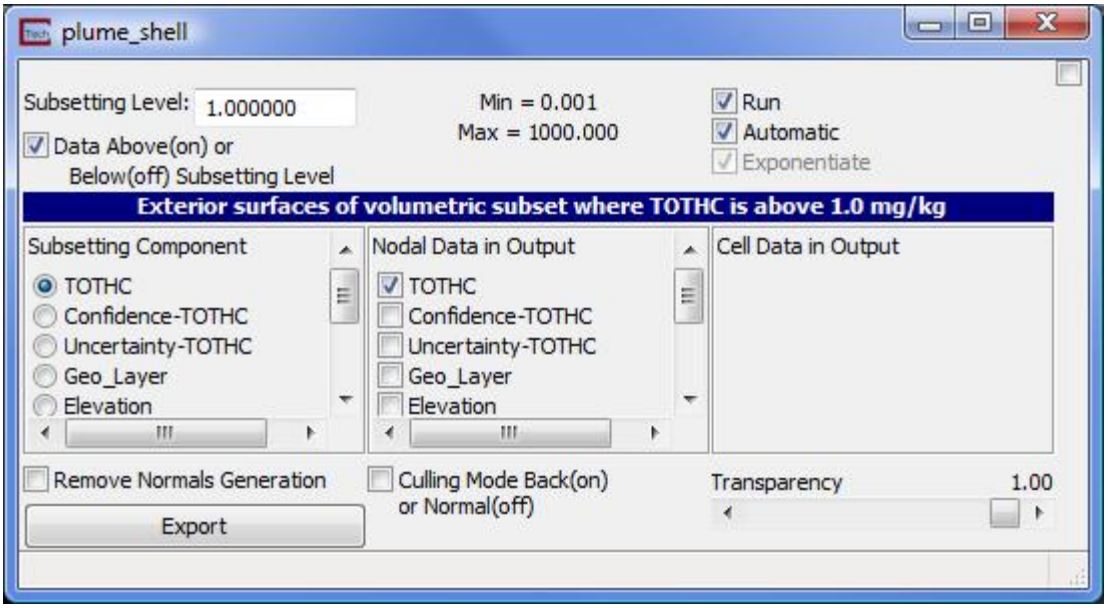
Before proceeding to the next topic, reset Line_Rendering (Object Editor) to inherit and close the *Viewer:Object Editor* window.

Plume Visualization Using plume_shell

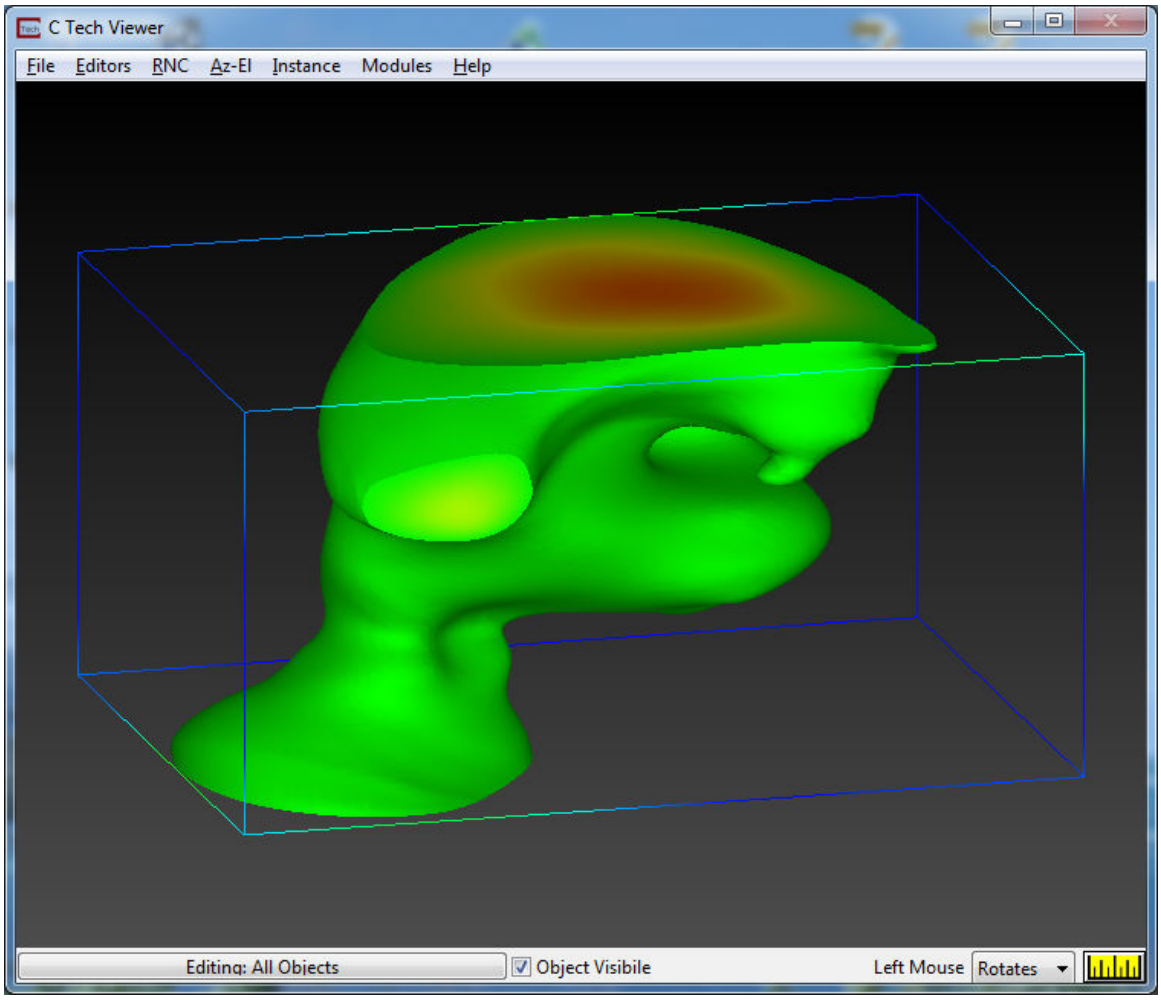
Krig_3D has produced a finite element model with the chemical concentrations of the input file mapped to a regular grid. From this model, we can quickly create a visualization of the contaminant plume. All we need to do is replace the external_faces module with plume_shell and add external_edges. Your network should look like the one below.



Since our data was clipped at 0.001 ppm minimum (log10 value -3) and 1000.0 ppm maximum (log10 value 3) the initial subsetting level will be set to 1 ppm (log10 value of 0.00 zero) which is the midpoint.

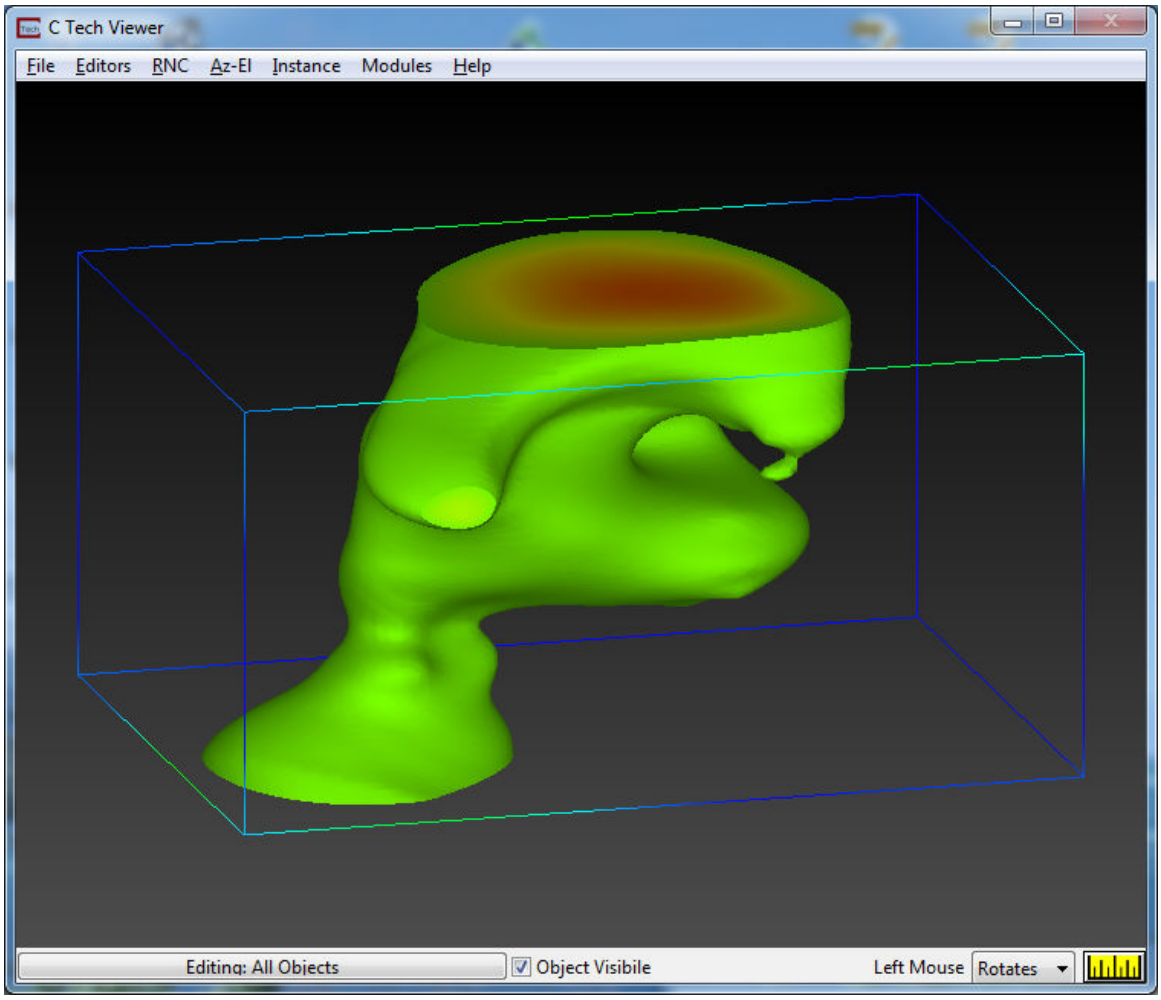


If you set your Elevation to 15 degrees, Scale to 1.0 and choose 195 degree Azimuth, your view should look like this:



You can adjust the *SubsettingLevel* to any value between the minimum and maximum values in your data. Open plume_shell's window and set the *SubsettingLevel* to 5.0.

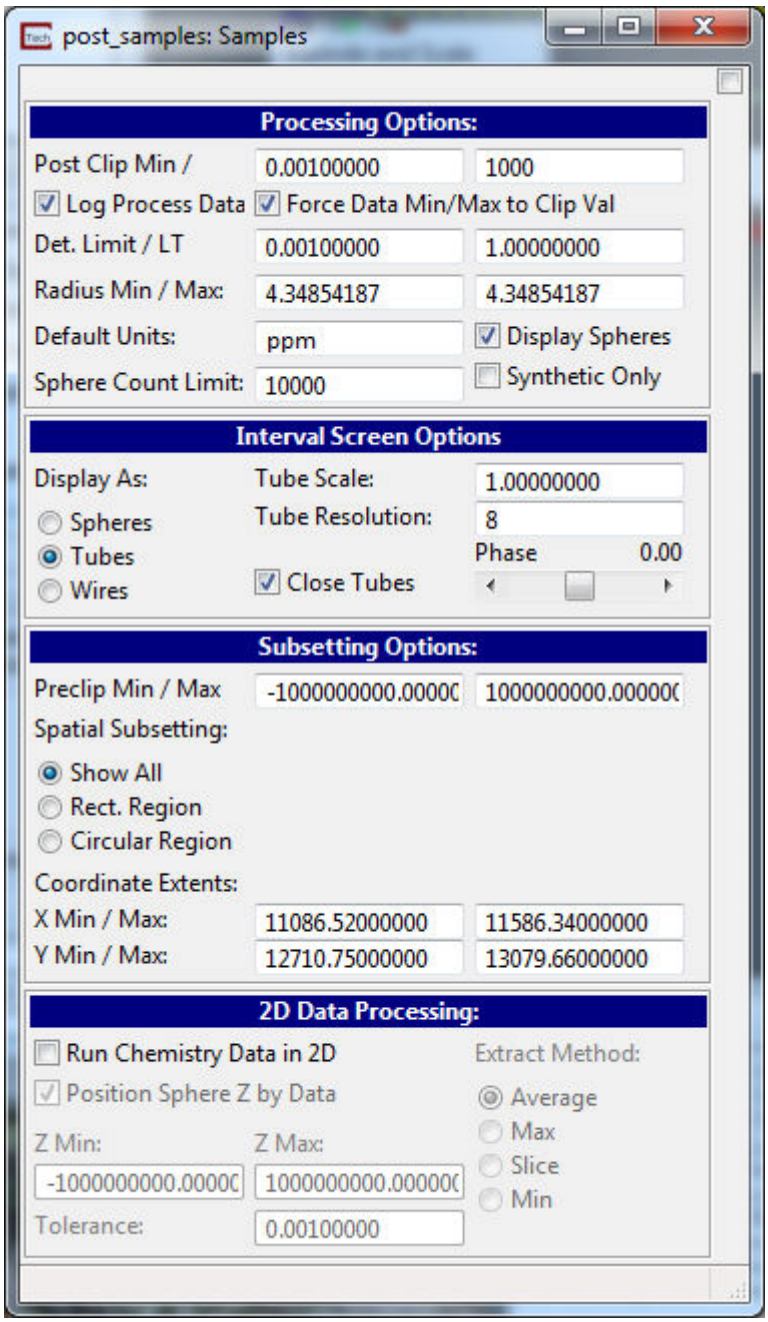
Your Viewer should now show:



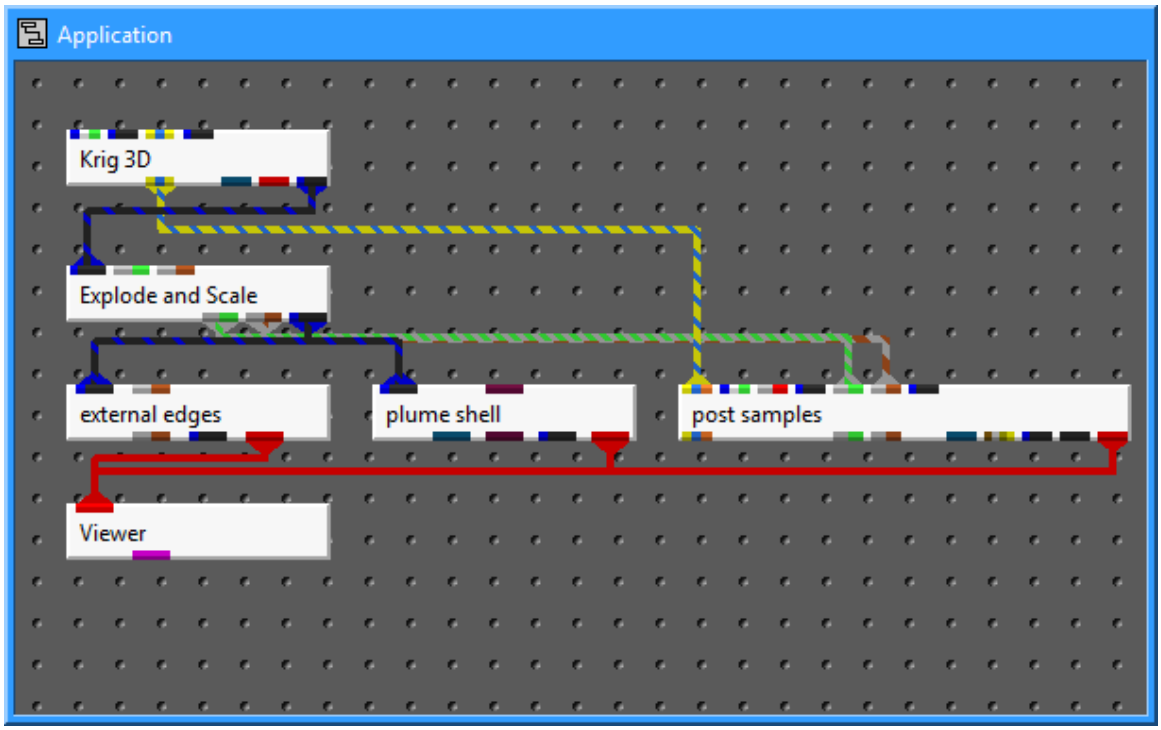
Using post_samples to Post Data

An important part of assessing the accuracy and effectiveness of your analysis is to post the actual measured data used to create the model. To do this, add the post_samples module to your network.

Before connecting (or running) the module, open the Samples window and set the Post Clip Max to 1000.0 to match the values set in Krig_3D.

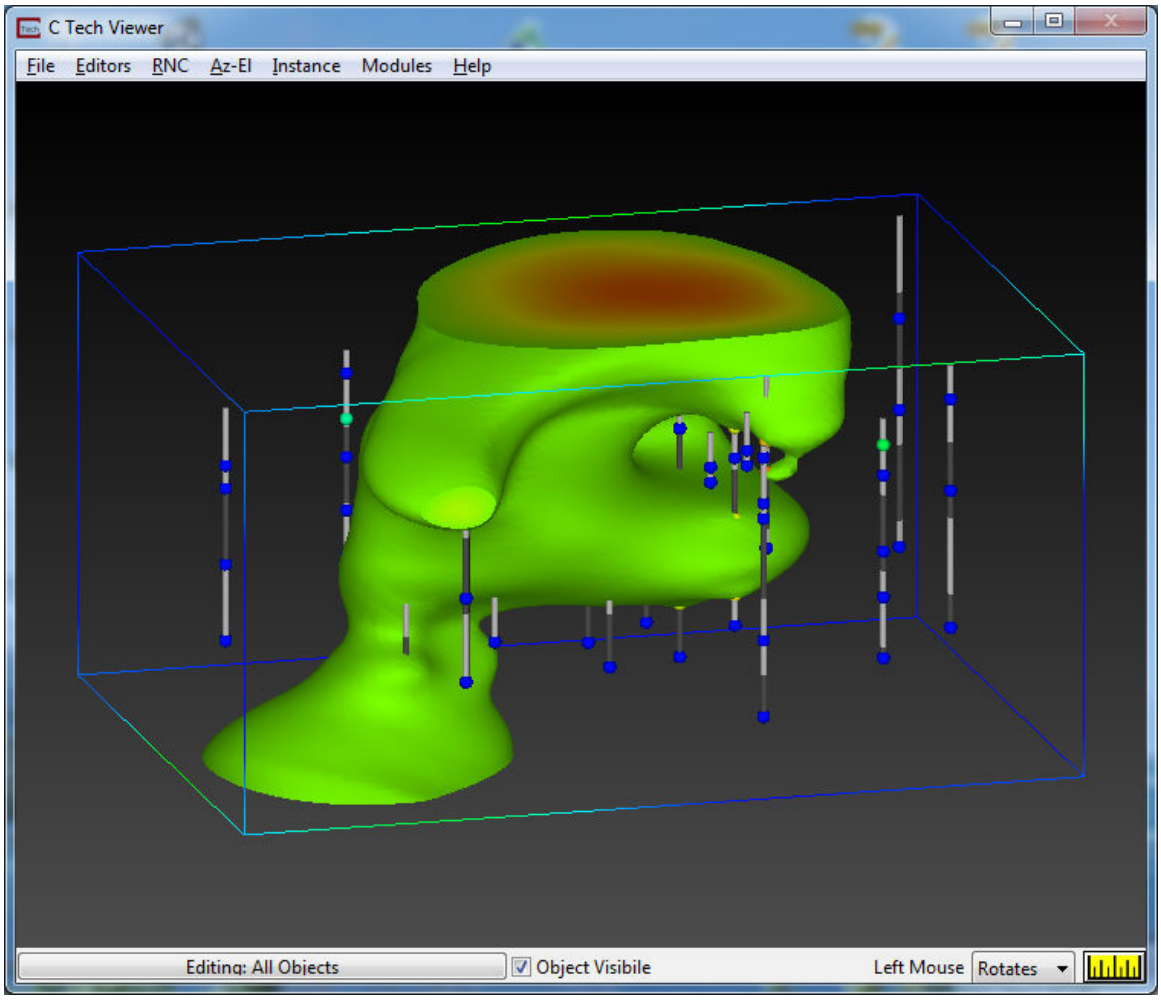


Then make the connections as shown in the figure below.



Because we've connected the yellow-blue-yellow port Krig_3D, we don't need to push Load File on post_samples and choose the same file we selected for Krig_3D (initial_soil_investigation_subsite.apdv). This port passes the file name for us and causes the module to run. The gray-brown port from Explode_and_Scale passes our current z scale to post_samples as well. This makes sure our Z scale always matches in these two modules.

Update the Az-El panel by clicking on the colored (Reset-Normalize-Center) button and then on Azimuth 195, and your view should look like this:



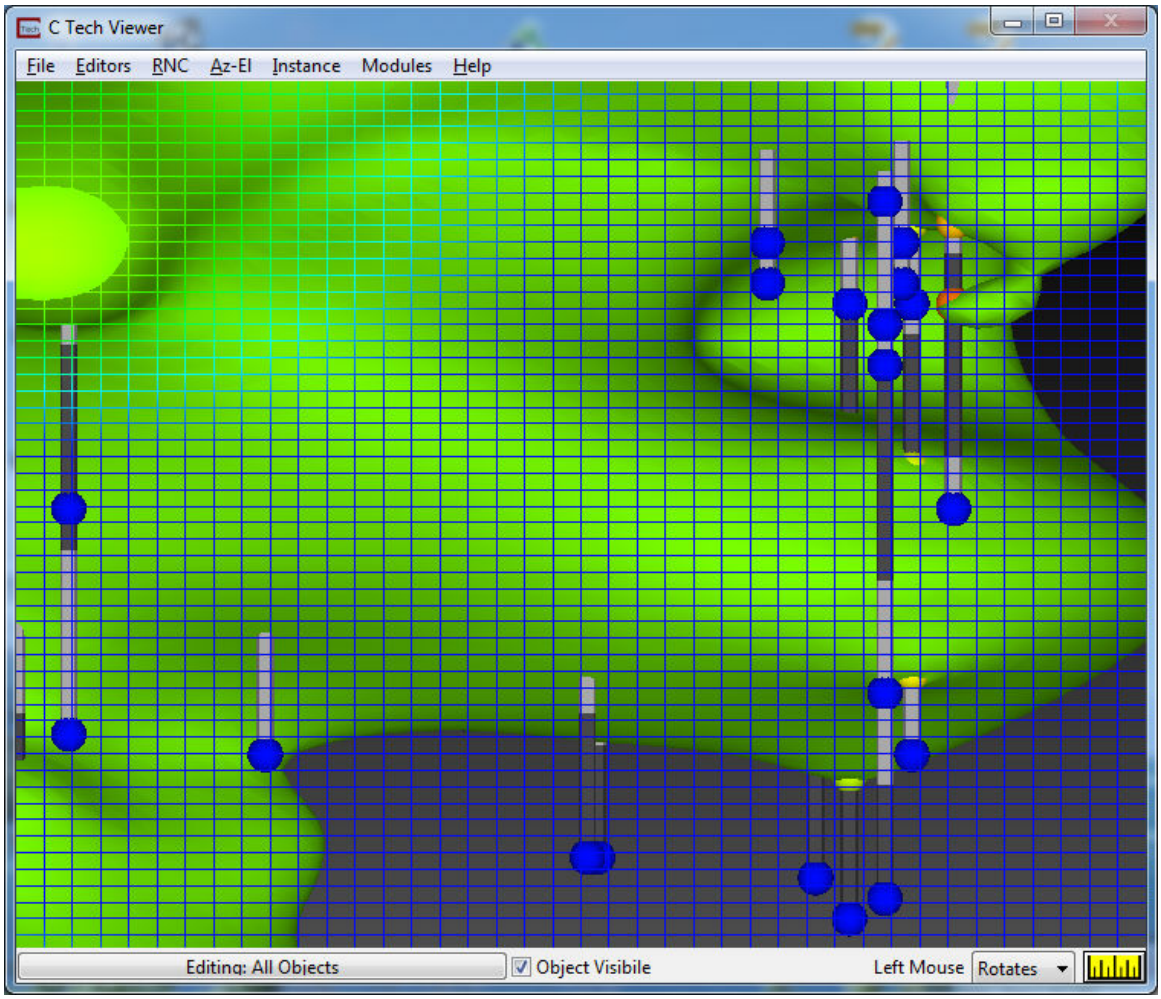
EVS Honors Measured Data Samples

Now let's address an important topic, honoring of measured data. By honoring we mean that the predicted concentrations should precisely match the actual data. This is only possible if grid nodes actually match the coordinates of the input (measured) data. Otherwise, we want our predicted distribution to approach our actual data as nodes approach measured sample locations.

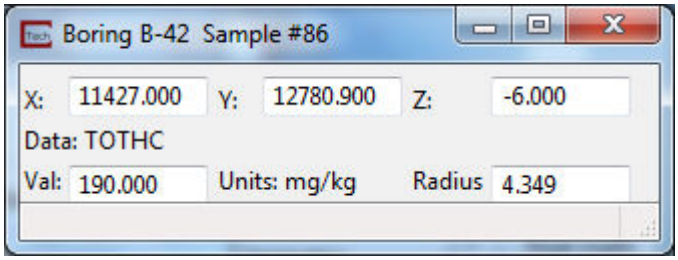
This concept of honoring the data seems obvious and hardly worth mentioning. However historically, kriging used a variogram parameter called a nugget. This concept originated with using kriging for predicting ore bodies (which contain randomly dispersed nuggets). If a nugget is used in the variogram (it often seems to make it easier to fit the variogram to the data), the predicted concentrations will not honor the measured data.

First, set the *MaxEdgeAngle* on external_edges to 0, and adjust your view in Azimuth & Elevation by setting the Scale to 2.6, Elevation to 0.0 degrees and choose 180 degree Azimuth. It may seem strange that the picture below which may suggest that EVS does not honor the data. Note that there is an

orange sphere with a concentration of 190.0 mg/kg which is partially outside of the 5.0 mg/kg plume.



To see the properties of this sample, hold down the ***Alt*** key and click on the orange sphere with your left mouse button. This should pop-up the window below in the upper right corner of your screen.



Because this is an important topic, we wanted to show an example which illustrates the effects of inadequate resolution. We set the Max Edge Angle to zero so that the grid would be obvious.

Though the entire spheres are not surrounded by the plume, the centers of the spheres with concentrations higher than 5 are definitely inside. Adapative Gridding insures this will be true.

For this dataset, the gradients in concentration are often high.

EVS's kriging routines will always honor the measured data value. It will calculate a geostatistically defensible value at all other locations. However, if a plume does not seem to adequately honor the measured data increasing resolution may resolve the problem. Remember, when dealing with 3-D kriging, the resolution must be considered in all three dimensions X, Y, and Z. Also, be aware that "honoring" your data means that the kriged estimate at a location will match the data at that same location. If you make a grid that is much larger than your dataset, you should expect far beyond the data, you may not be satisfied with the estimate. Kriging will project trends and far from the data will tend towards the average.

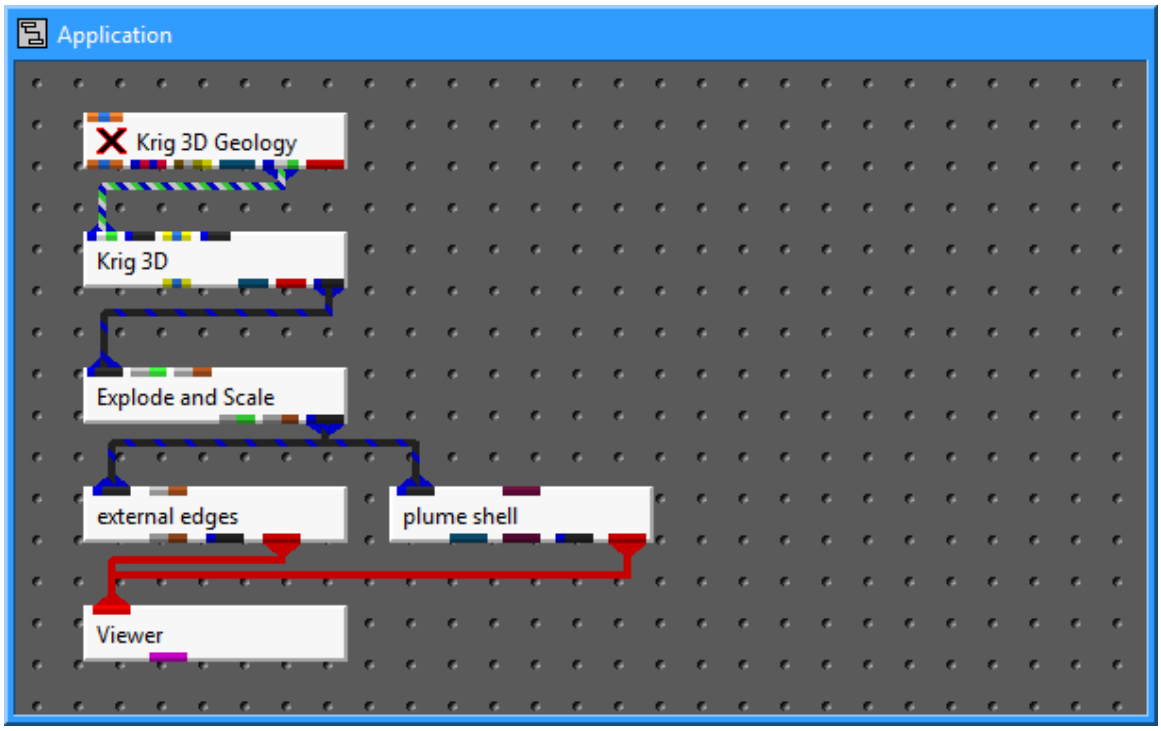
Krig_2D and Krig_3D have adaptive gridding on as the default.

Adaptive gridding assures that we will have grid nodes at all data points and reduces (almost eliminates) the need for excessively high resolutions.

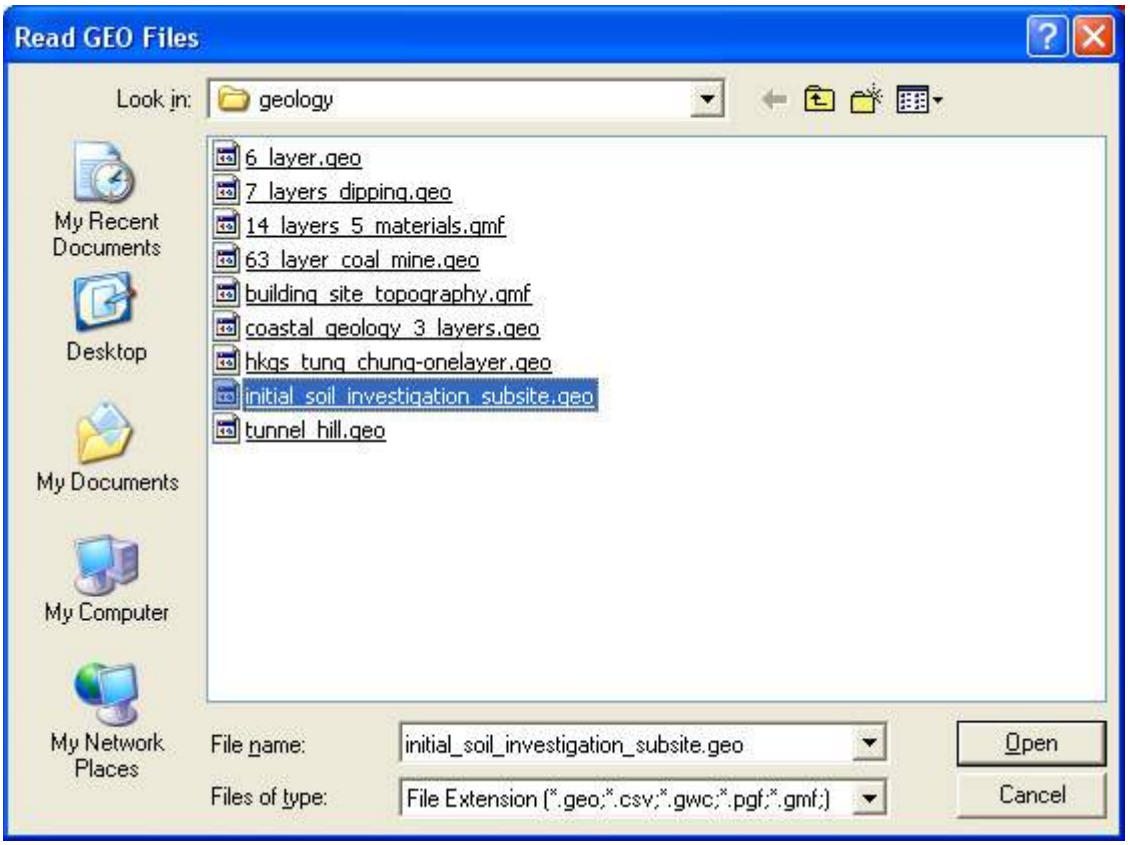
Before proceeding to the next topic adjust the Max Edge Angle back to 60 degrees and set the view to Elevation 12 degrees, Scale 0.7 and Azimuth 210.

Adding Geologic Data with Krig 3D Geology

It is not always adequate to model chemistries within a "brick". To address the most general case, we must map analyte (e.g. chemistry) into geologic structures. Krig_3D_Geology is the primary module for creating three-dimensional models of geologic structures. We will now add it to the network in order to remap the analyte (e.g. chemistry) data to a grid which is separated into geologic layers. Modify your network to match the one below by adding Krig_3D_Geology and deleting post_samples:



Select the Krig_3D_Geology main menu. Click on the **Read DataFile** button and choose the file as shown below.

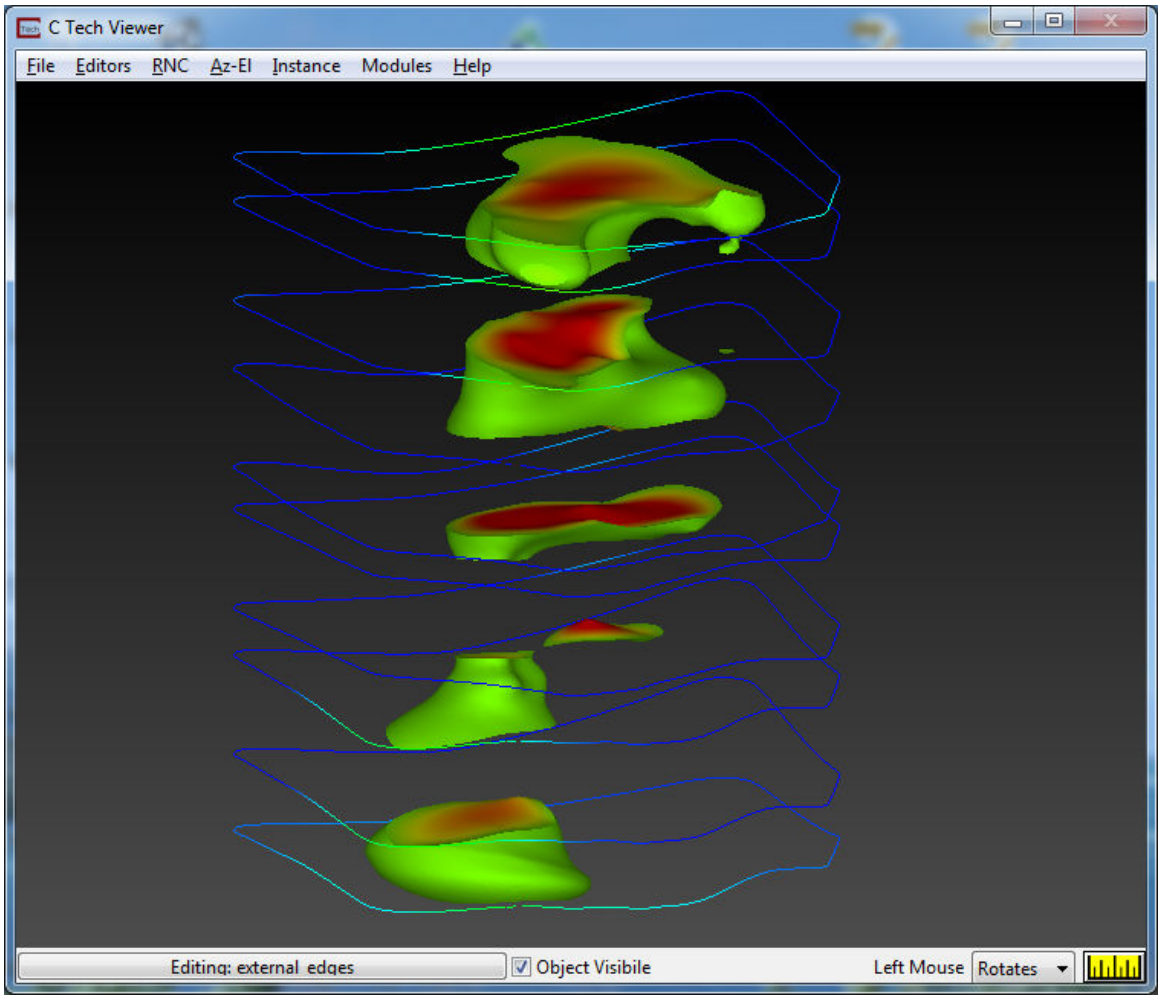


Select the **Accept Current Values** button and Krig_3D_Geology reads the geology data and begins the kriging process. In less than one minute, it calculates the surfaces for the grid we selected. These are passed to Krig_3D, which automatically re-runs having detected geologic input and since it already has a data file.

Krig_3D will rerun, but this time, the grid to which the parameters are estimated will be defined by the convex hull of the geology file (in the x-y plane) and by the geologic layers in the z direction. Remember that the convex hull is only one of three options for the extents in the x-y plane. After a few seconds (or more with slower computers), the kriging calculations should be complete.

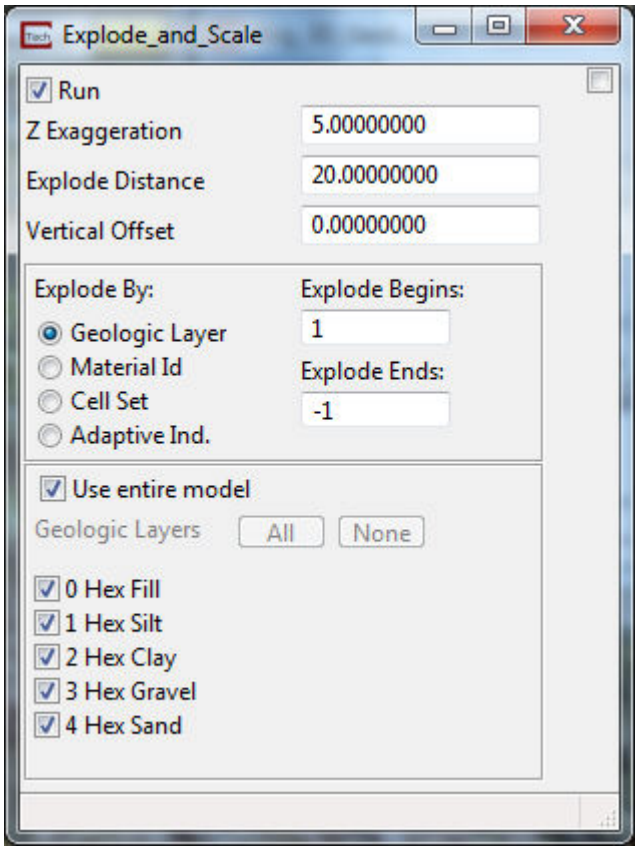
Verify your Az-El settings and set the view to Elevation 12 degrees, Scale 0.7 and Azimuth 210.

Your view should look like this:



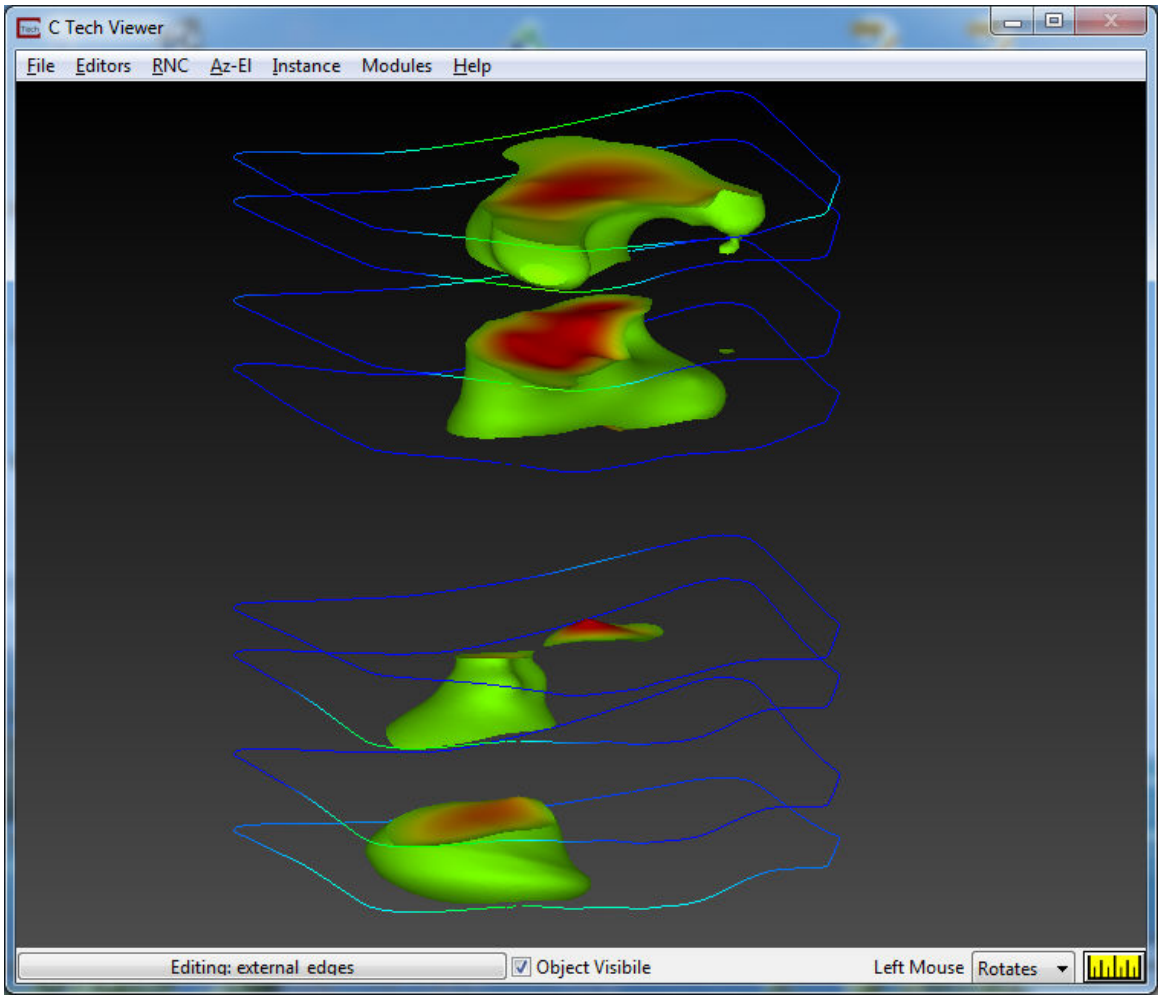
Explode_and_Scale Layer Controls

Select Explode_and_Scale's menu



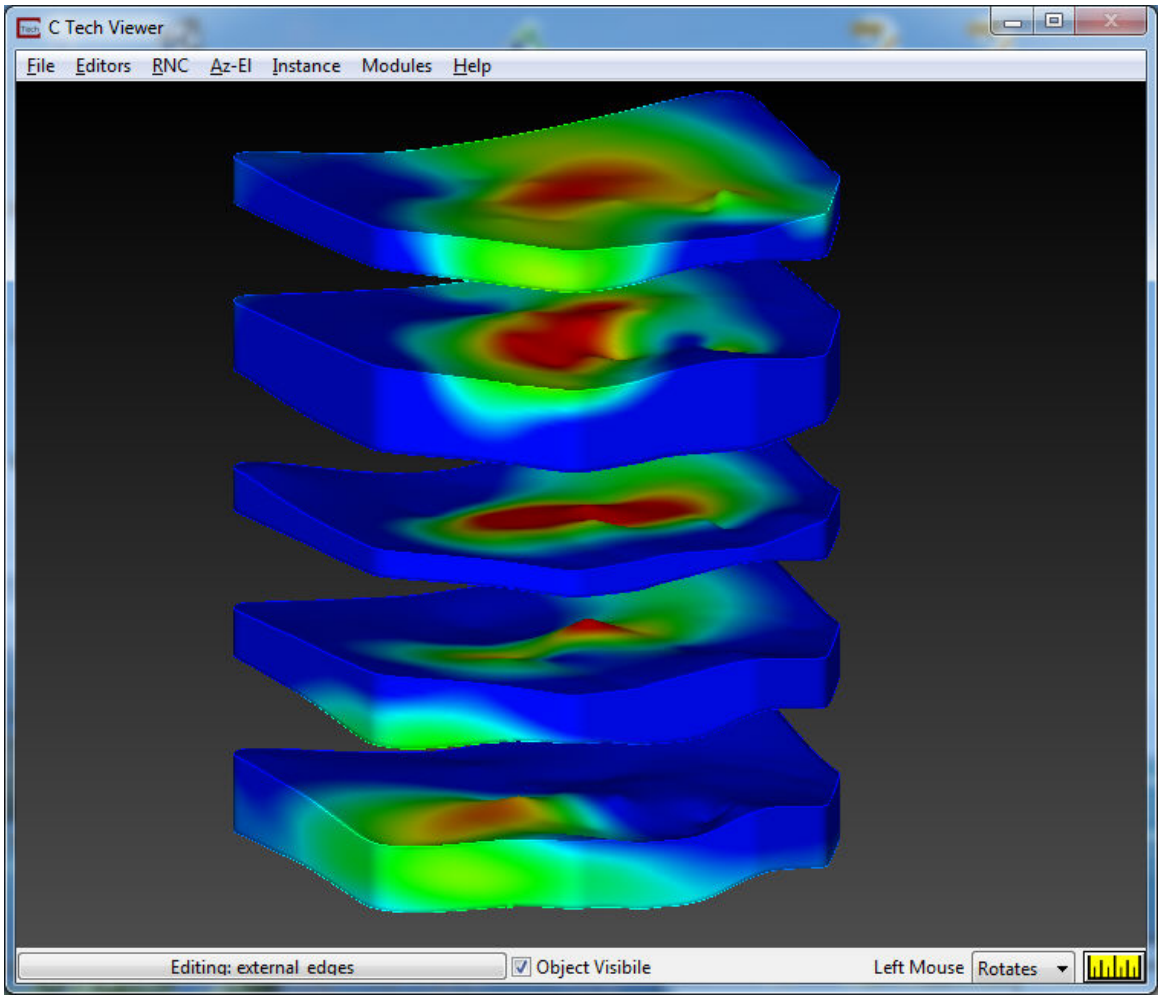
Note that you have toggles to control the visibility of each layer (they are named based on the material names in your .geo or .gmf file). Let's turn off *Use Entire Model*as well as the middle Clay layer.

Your view should look like this:



If it does not, reset your view to 12 degree Elevation, Scale of 0.70 and 210 degree Azimuth.

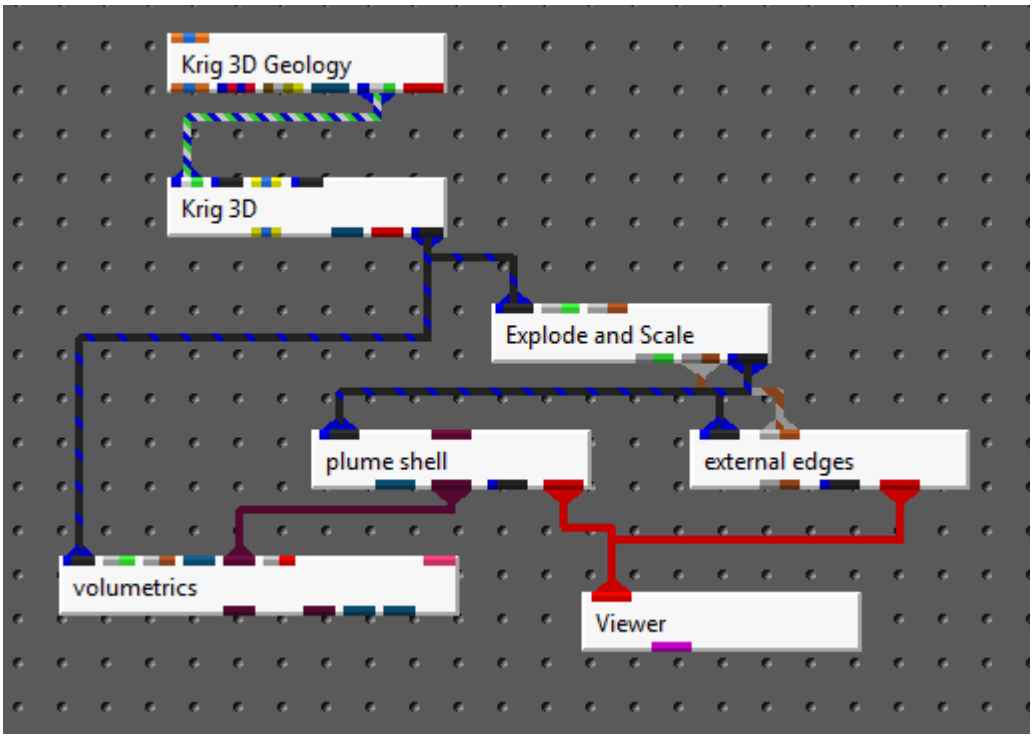
To see the entire model's external faces, turn the clay layer back on and set the *SubsettingLevel* in plume_shell to 0.001. You should now see:



Now that we have a model of the analyte (e.g. chemistry) mapped onto geologic layers, let's calculate volumes and masses within plume levels.

Calculating Volumes and Masses

Add the volumetrics module to your network so that it looks like this:



The volumetrics module should be connected directly to Krig_3D. If you connected it to Explode_and_Scale, the volumes and masses would be exaggerated by the scale factor in Explode_and_Scale. You can use the explode and scale ports in volumetrics to compensate for this, but it is simpler if it is connected to the original, unscaled, unexploded field.

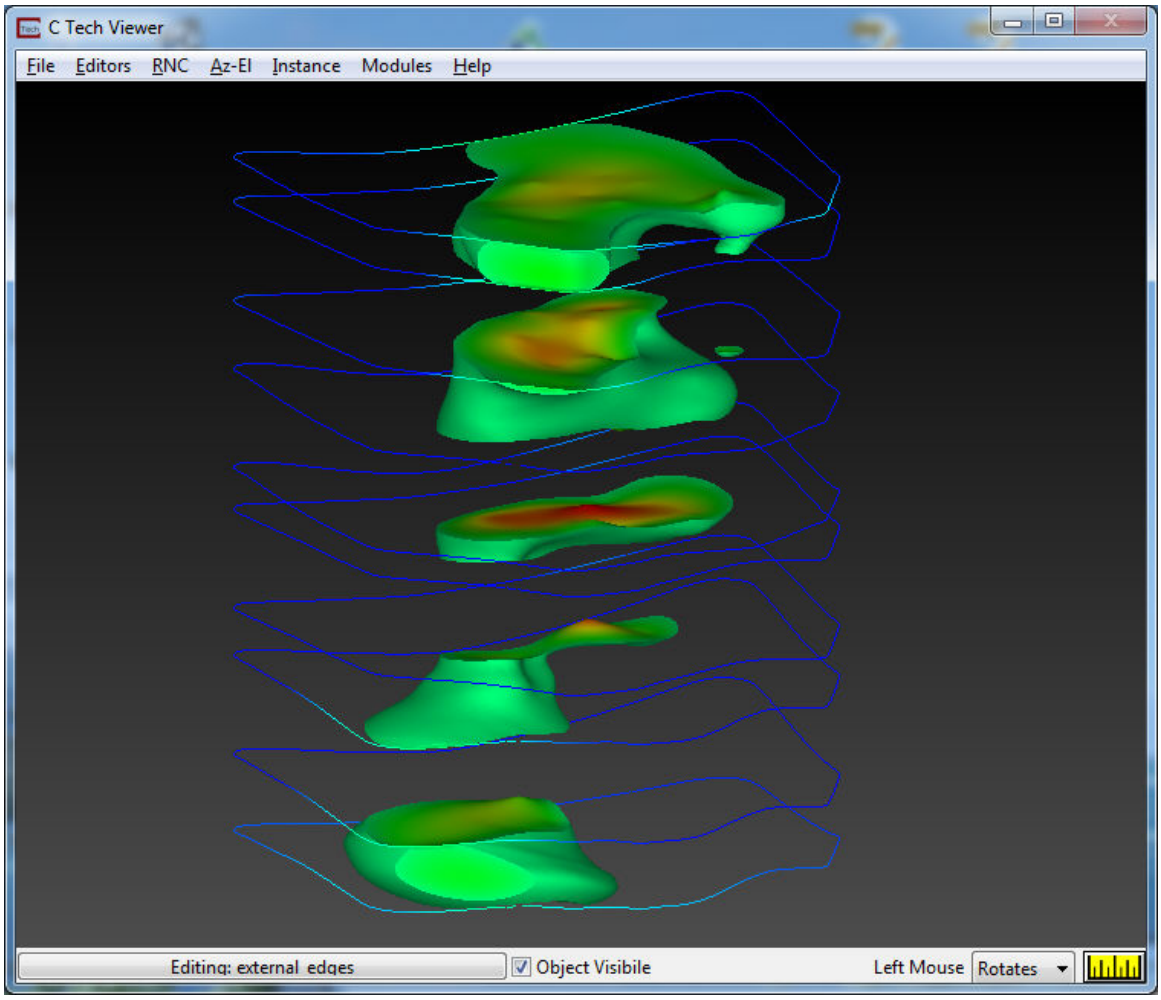
There are times when you might want to insert other Subsetting or Processing modules in between Krig_3D and volumetrics. Examples are plume_volume or field_math. As an example, the plume_volume (lower case) module could be used to subset the domain to include only those regions above or below a particular elevation or depth.

After connecting volumetrics, we must change one of Krig_3D's parameters to be sure we don't calculate an inappropriate estimation of chemical mass. Go to Krig_3D's *Data Processing* window and change the Post-Clip max parameter to 1.000000e+009 (1 billion), thus insuring we aren't clamping the extreme maximum values in our kriged output. Now re-run Krig_3D by clicking *AcceptAll Current Values* in the Krig_3D window.

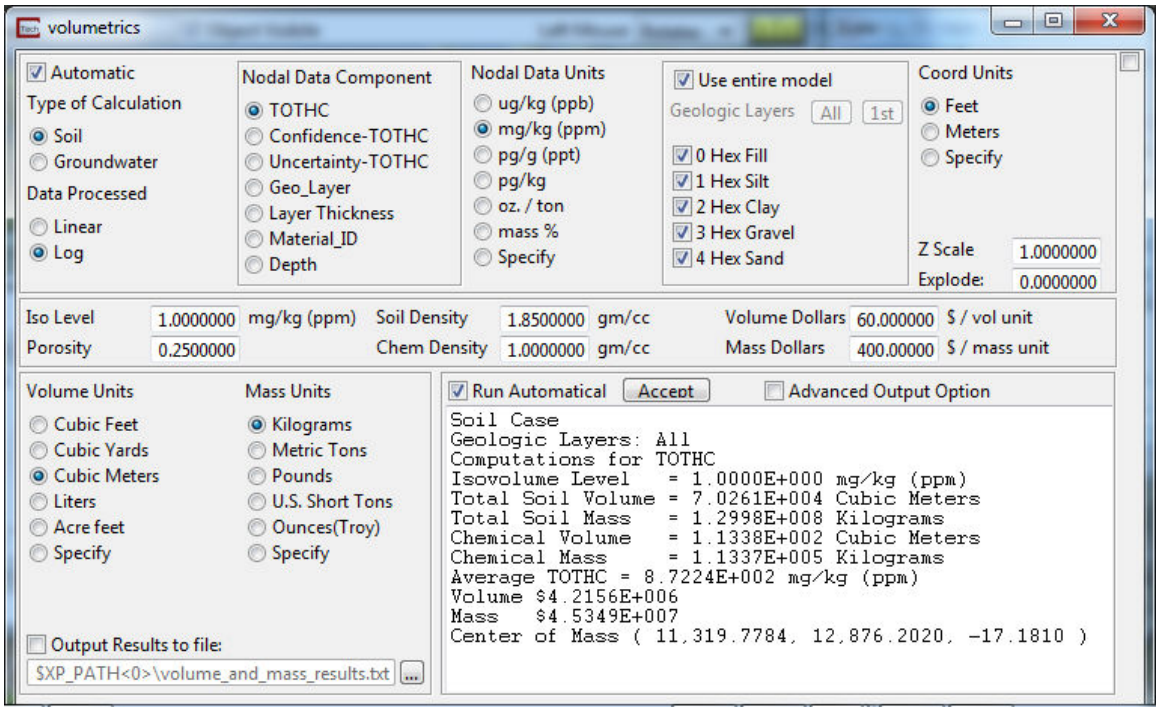
Now, open the volumetrics window. Change the *Iso Level* to 1 mg/kg and the *Volume Units* to Cubic Meters. This causes volumetrics to calculate the volume and chemical mass for the regions of the model above 1 mg/kg.

This will also update the Subsetting Level in plume_shell

The Viewer should show:



The volumetrics window should now look like this:



Since our data files have their units specified correctly, volumetrics automatically determined the appropriate values for *Type of Calculation* (Soil), *Data Processed* (Log), *CoordinateUnits* (feet), and *units* (mg/kg). Setting up the input data with this information can help prevent mistakes from occuring in the post-processing.

You can adjust the subsetting level (which should be in original units: not log10 values), the dollars per volume or mass unit, the input units or output units. You can also select specific geologic layers (versus calculating the entire domain). This is especially useful if the porosity of individual layers is highly variable and an accurate estimate of ground water chemical mass is required. For detailed information on volumetrics parameters refer to the Module Library help section.

3D Kriging Conclusion

In this Workbook, we explored the application of EVS to the modeling of three-dimensional chemical plumes or ore bodies and mapping those to complex geologic structures. You should now be able to take simple ASCII input files determined from boring logs and produce three-dimensional models which include chemical (or property) mapping to geologic structures.

The final application in this workbook has been saved for you in the Workbooks folder under Applications as workbook-5-final.v

Workbook 6: Fence Diagrams.

Introduction

In this workbook, three different methods for creating fence diagrams will be investigated. The first two approaches begin with a 3D model. The last

approach is a more traditional fence diagram that must be defined between individual borings. This approach is available with the EVS standard product or higher, however, EVS-Pro and MVS have more options when using this approach.

This workbook assumes that you understand all of the subjects covered in previous workbooks including:

1) Network Editor Basics such as instancing, deleting, connecting and disconnecting modules.

2) Viewer basics such as setting views, rotations, scaling, translation, object selection and object property modification.

3) Two dimensional kriging of analyte (e.g. chemistry) data.

4) Geologic hierarchy and preparing geology input files

5) The function of the Krig_3D_Geology module.

6) The function of the plume_shell module.

7) Combining geologic and chemical data

8) The function of the post_samples module.

9) Layer control using Explode_and_Scale

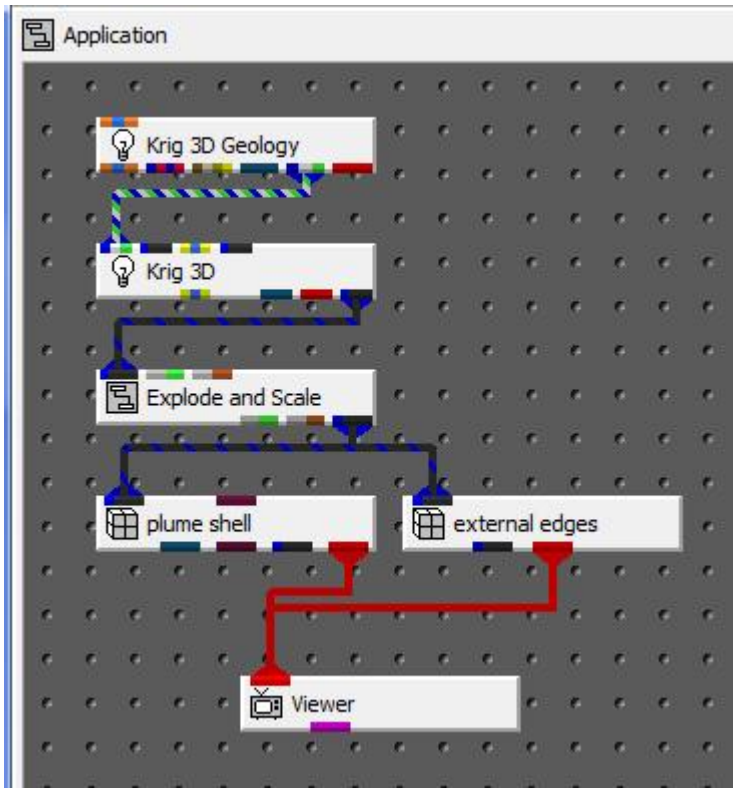
You must also be comfortable using the Modules pull-down menu to access the user interface for modules which have been instanced.

If any of these topics is unfamiliar, we recommend you review the previous workbooks before beginning this lesson.

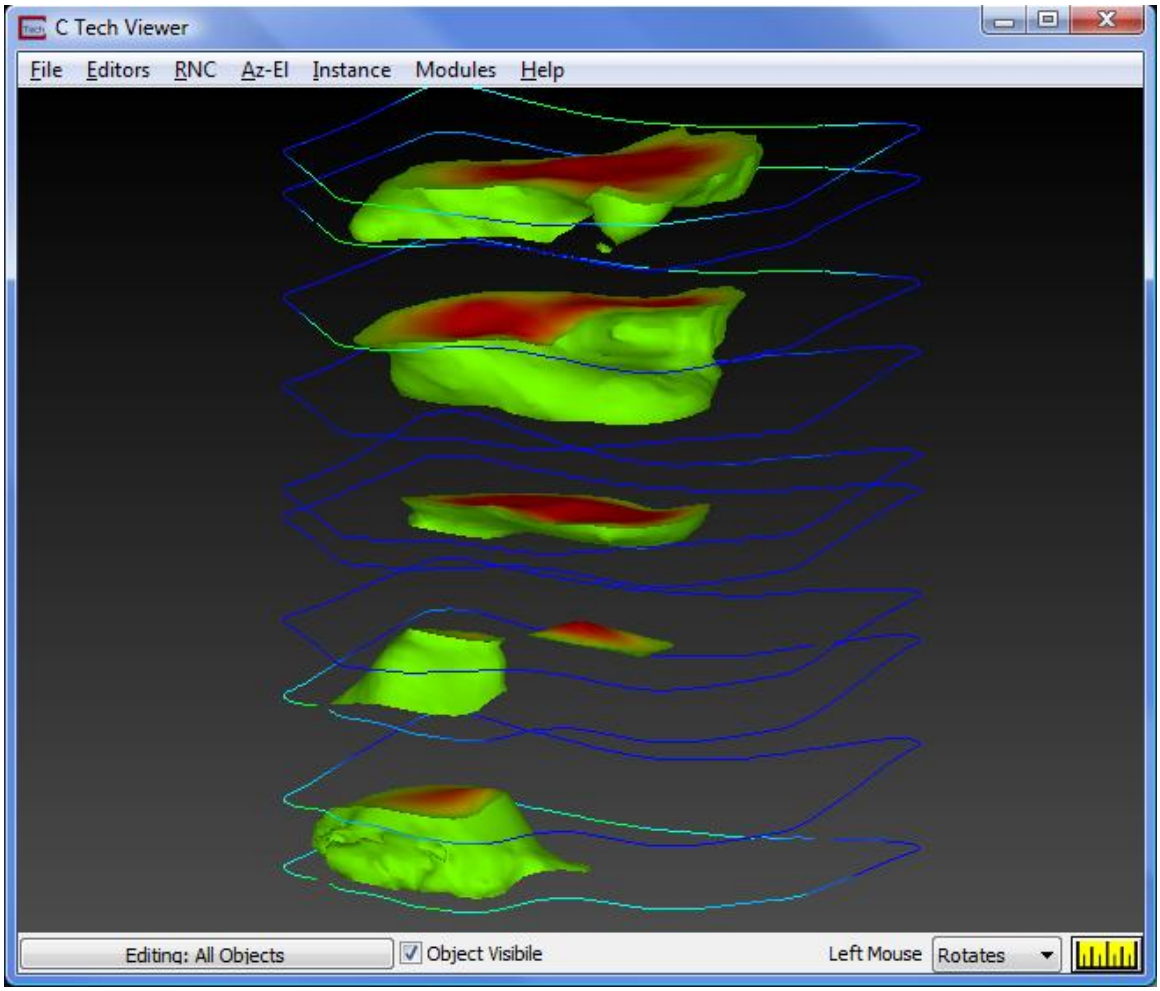
Fence Cut Module

The fence_cut module provides a means to subset three-dimensional models and produce fence diagrams with thick cross-sections. Fence cut may be used with applications that use chemical data, geologic data or both. The first step to use fence cut is to build a network that produces a three-dimensional model. For our example we will use an application with both geology and analyte (e.g. chemistry) data.

Build the application shown below using the data files initial_soil_investigation_subsite.geo and initial_soil_investigation_subsite.apdv for Krig_3D_Geology and Krig_3D respectively. Use all of the default values.



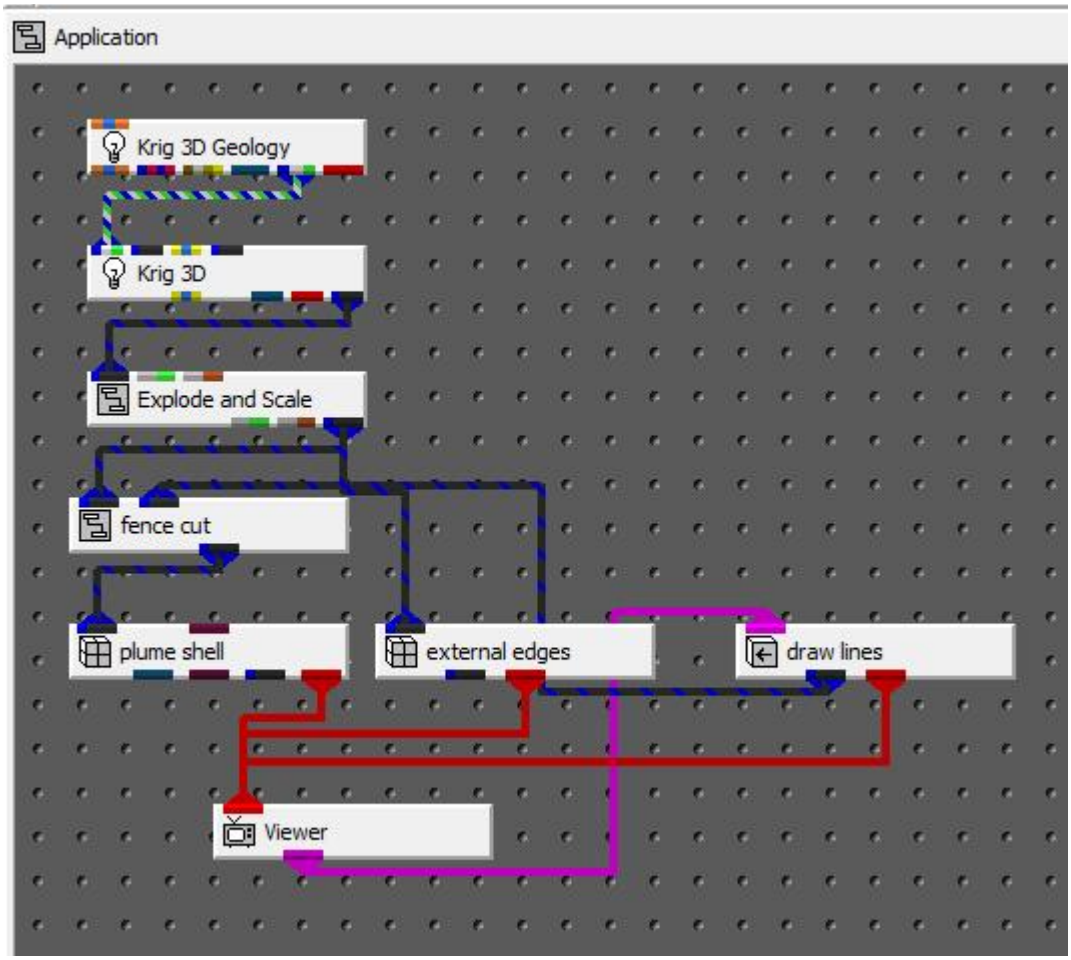
This basic network should create a three-dimensional model with five geologic layers. By setting your Azimuth and Elevation settings to an Azimuth of 150, Elevation or 10, and Scale of 0.7, your viewer should look similar to that below.



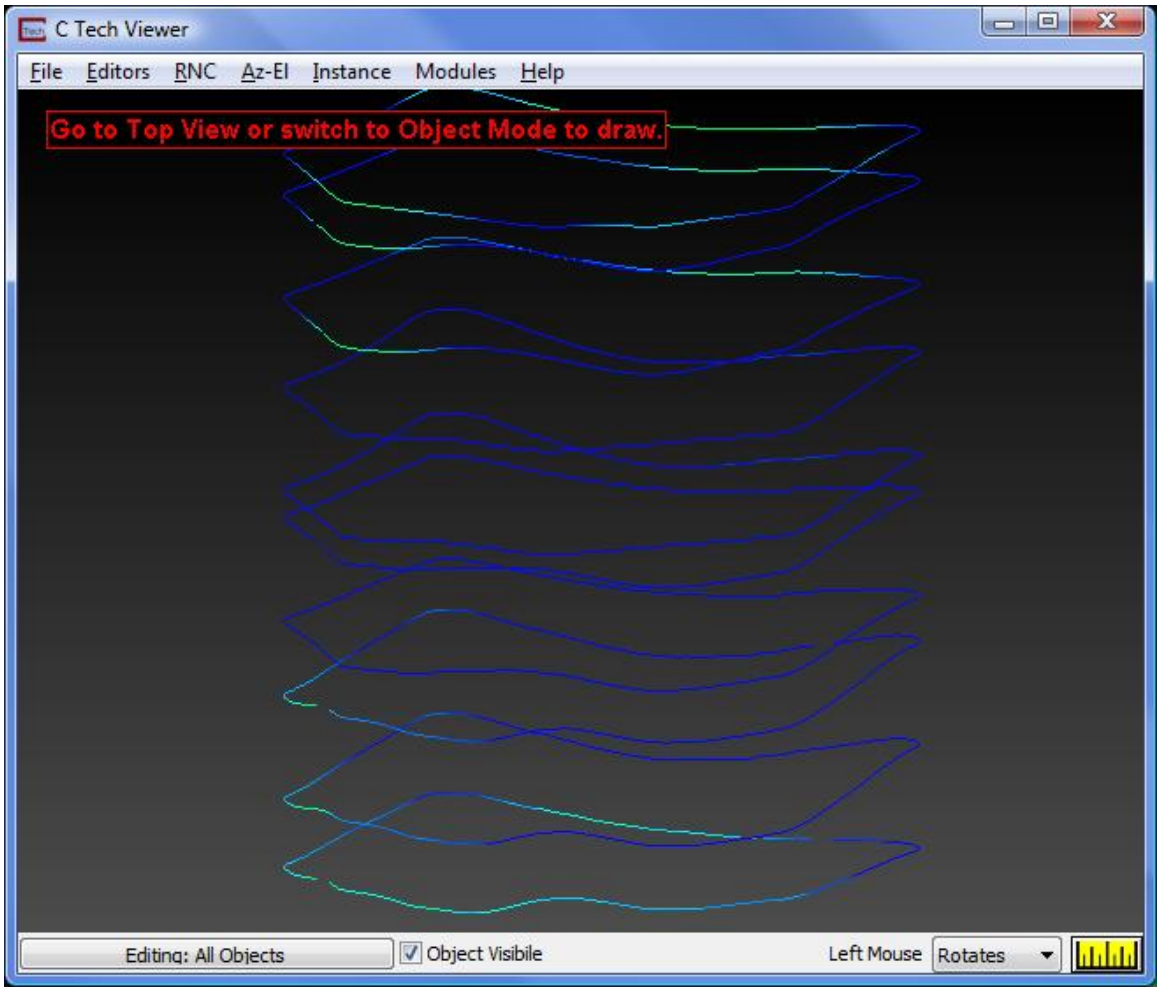
To this network we will add two new modules. The first module is called draw_lines. It provides a means to draw in the viewer. draw_lines has two drawing modes. It can either draw in the x-y plane while in a TOP view (180 azimuth 90 elevation) or it can draw ON objects' surfaces in the Viewer. For this topic we will draw in a top view.

We will also add the fence_cut module to this network. Fence_cut takes the three-dimensional model into its left input port and the output lines from draw_lines into its right port. When fence_cut runs it adds an additional data component to those already present in the input field. This data component is used to subset the model producing the fence cut which represents a thick cross-section similar to a fence diagram. Connect these two modules so your application looks similar to the one below.

Note that draw_lines is an unusual module in that it connects to the viewer's red input port and is connected to the viewer's purple output port



Once you have made the connections above your Viewer should show:



Because we have selected to draw in a TOP view, the warning message in the Viewer is telling us we need to be in a TOP view.

The easiest way to obtain a top view is to choose the *Top View* option in the RNC pull down menu of the Viewer.

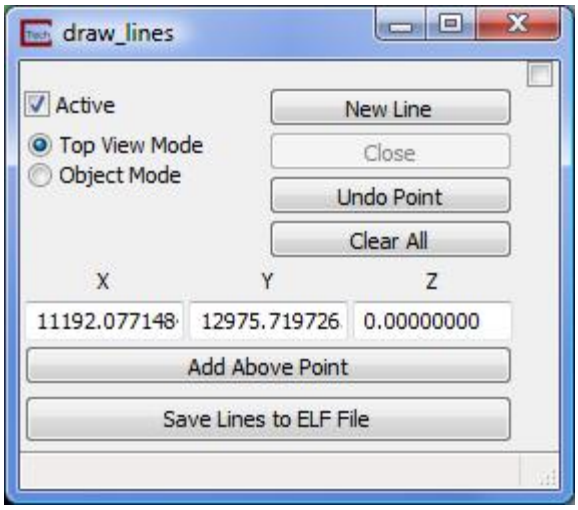
Drawing with draw_lines

To begin we will use the RNC pull down menu to set the view to a Top View. At this point we can draw in the viewer the path on which we wish to have the fence cross-section.

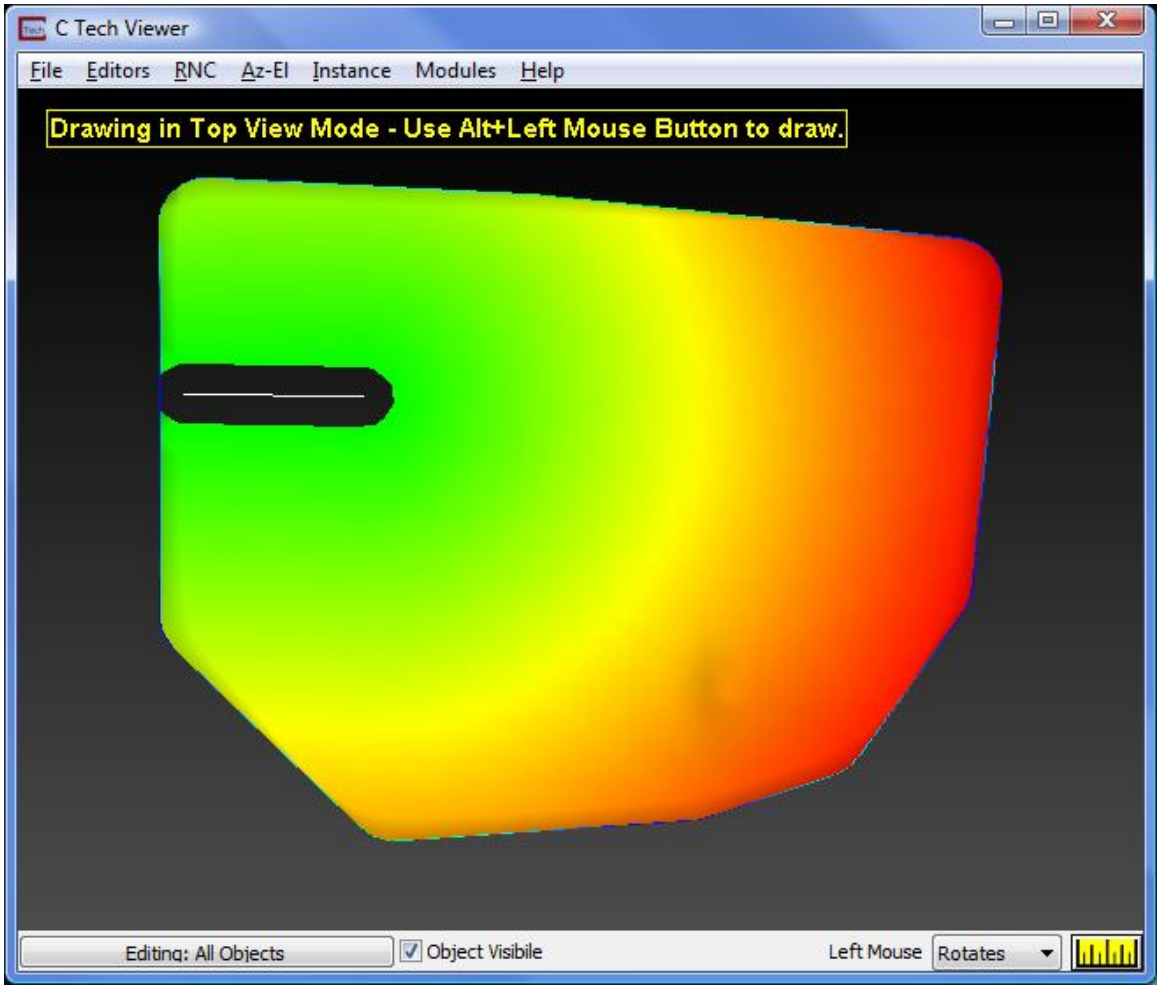
Open draw_lines' user interface. If you have setup everything correctly, a yellow message which says

"Drawing in Top View Mode - Use Alt+Left Mouse Button to draw."

should appear at the top of the window. If not, choose RNC->Top View in the Viewer's window as directed in the end of the last topic.

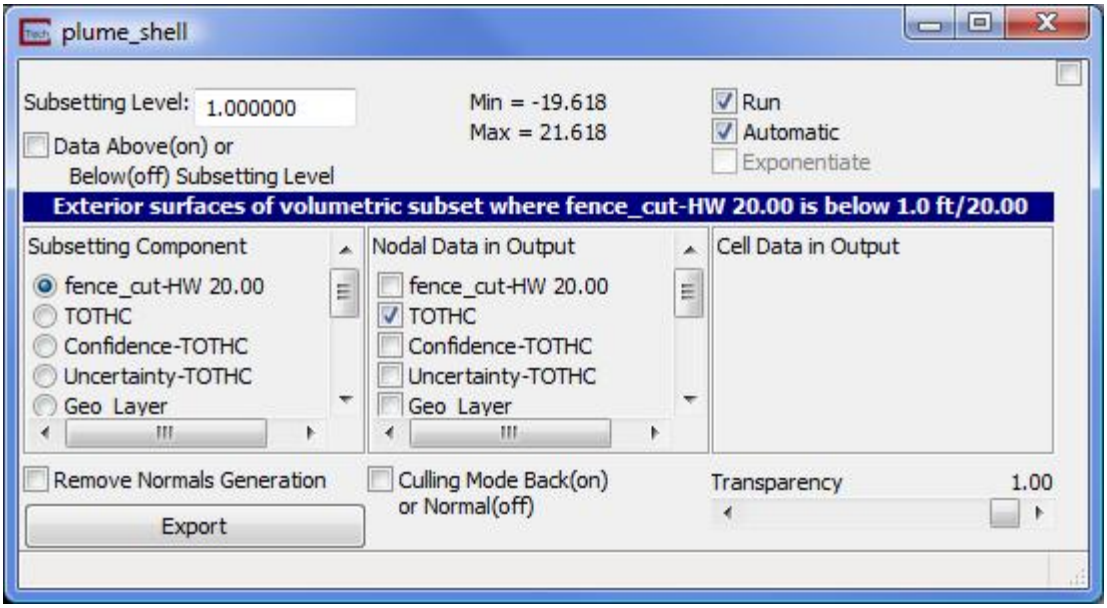


To draw in the viewer we use the ALT Left mouse button. Begin by pressing the "New Line" button. Then with your cursor near the left edge of the outline of the model created by the external edges module. while holding down the ALT key draw using your left mouse button from the left side of the model towards the right side of the model.

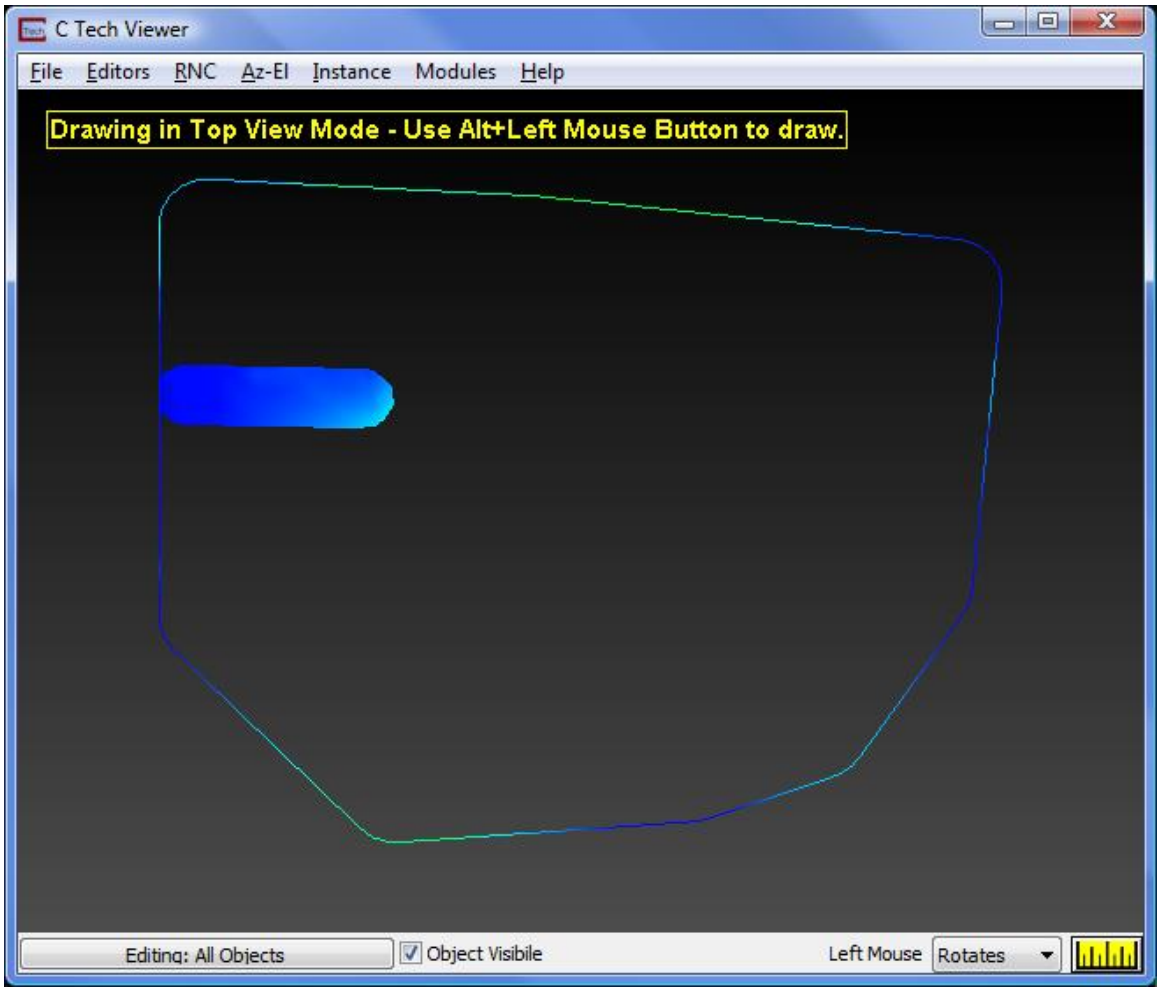


Cutting with plume_shell

From the module list choose the plume_shell module and change the settings to match those shown in the figure below.

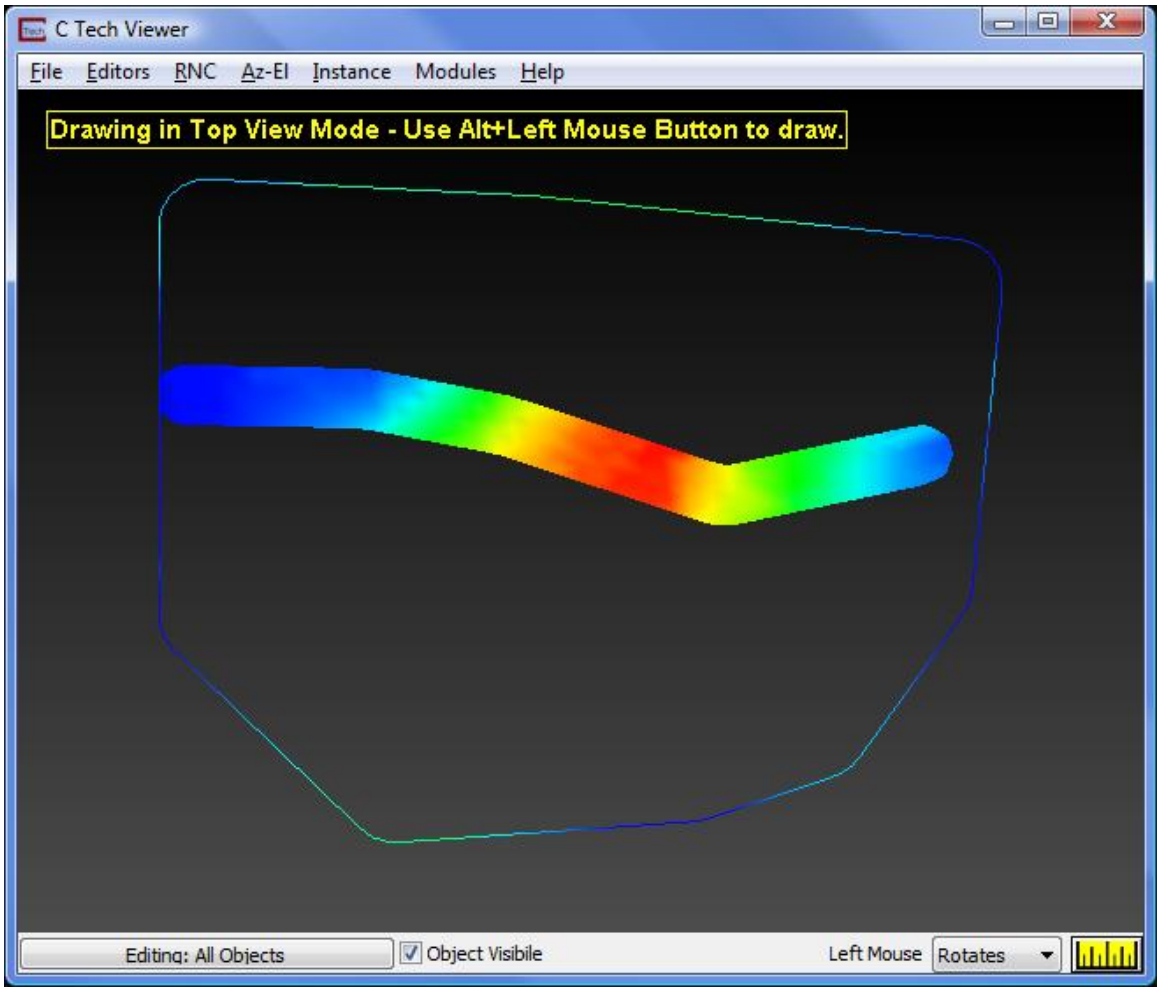


At this point your viewer should be similar to the figure below.

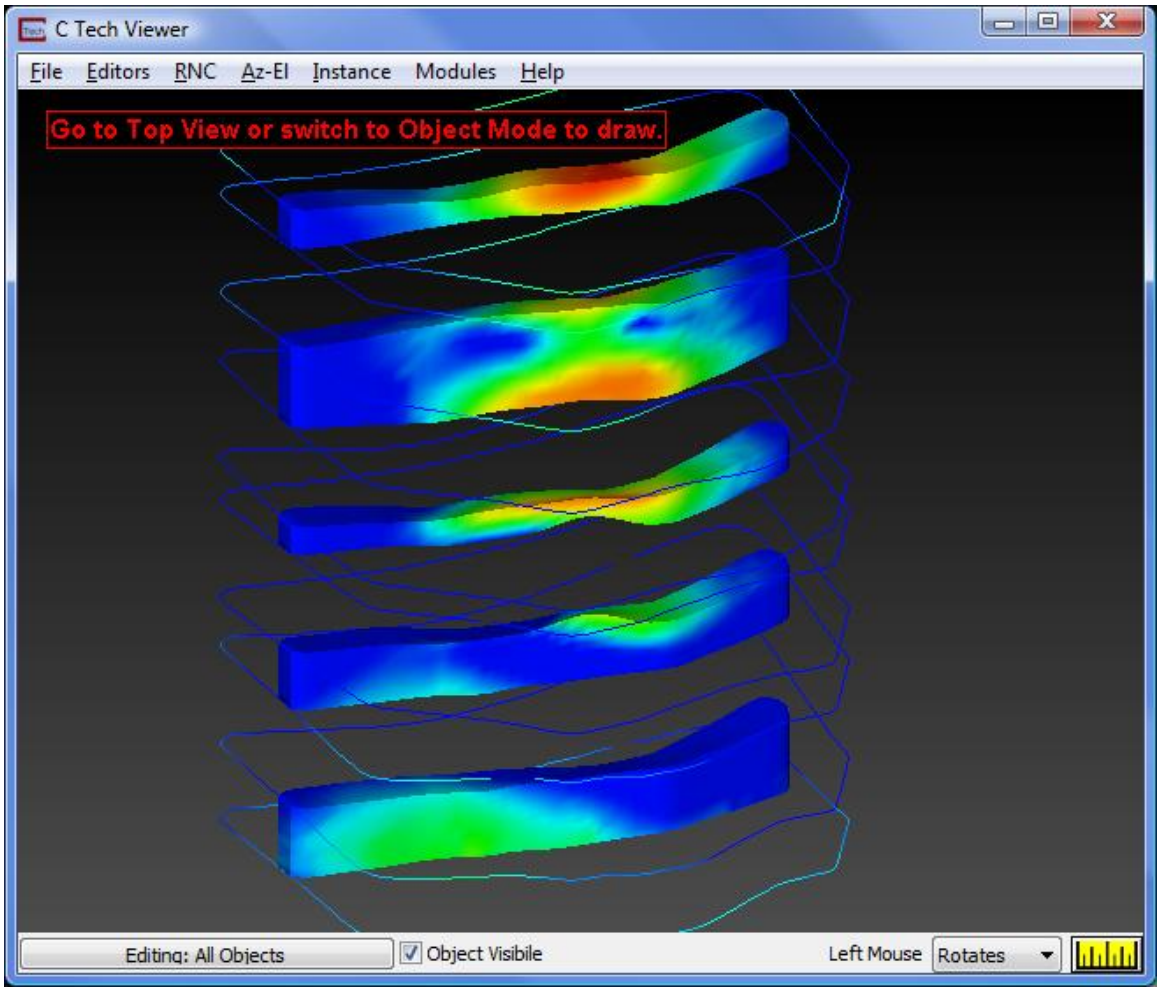


From a top view, this doesn't look like much. But before we change our view, let's draw a few more line segments. Continue drawing until your view is similar to the one below.

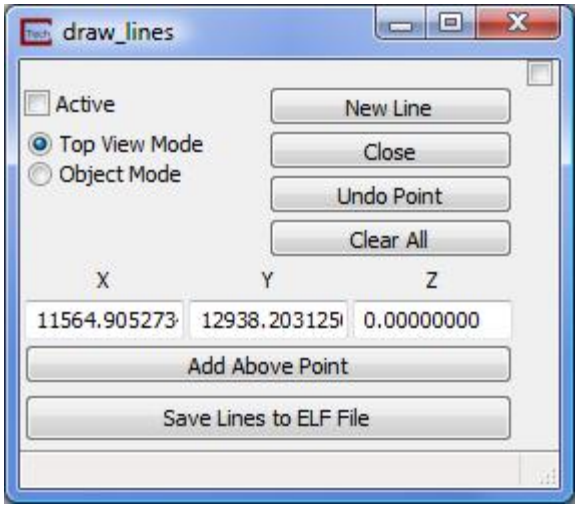
To do this, re-open draw_lines' control panel and use your left mouse button to add 3-4 more points.



Using the Azimuth & Elevation window, choose a scale factor of 0.7, an elevation of 20, and an azimuth of 210. At this point your viewer should look similar to the figure below.



To turn off the red warning message from the draw_lines module, just turn off its "Active" toggle.



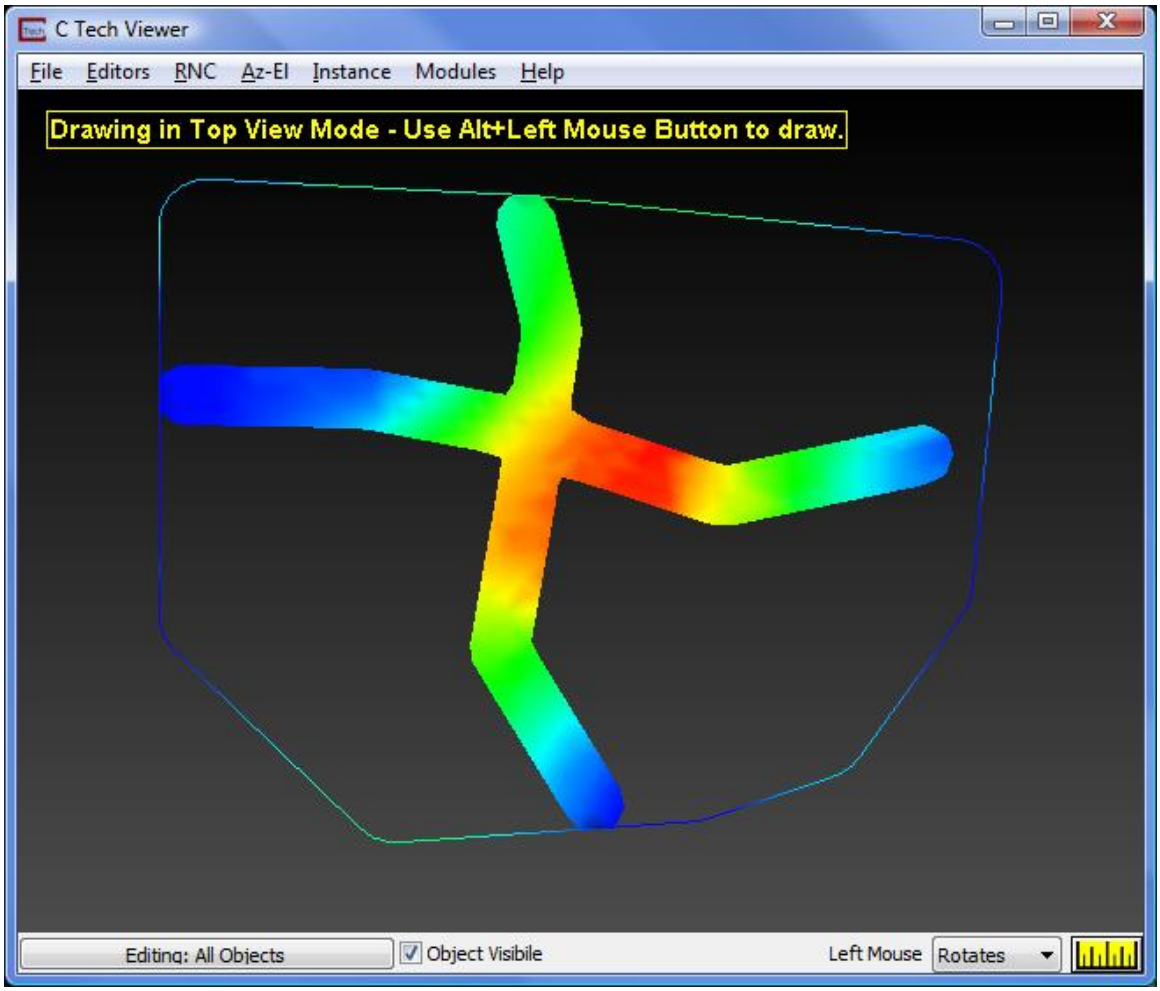
Adding Additional Cross-sections

To add additional cross-sections to the fence diagram, we must:

1. Return to a Top VIew
2. Turn Active on in draw_lines
3. Press the "New Line" button. If you do not do this, your new line will connect to the last point of your previous line.

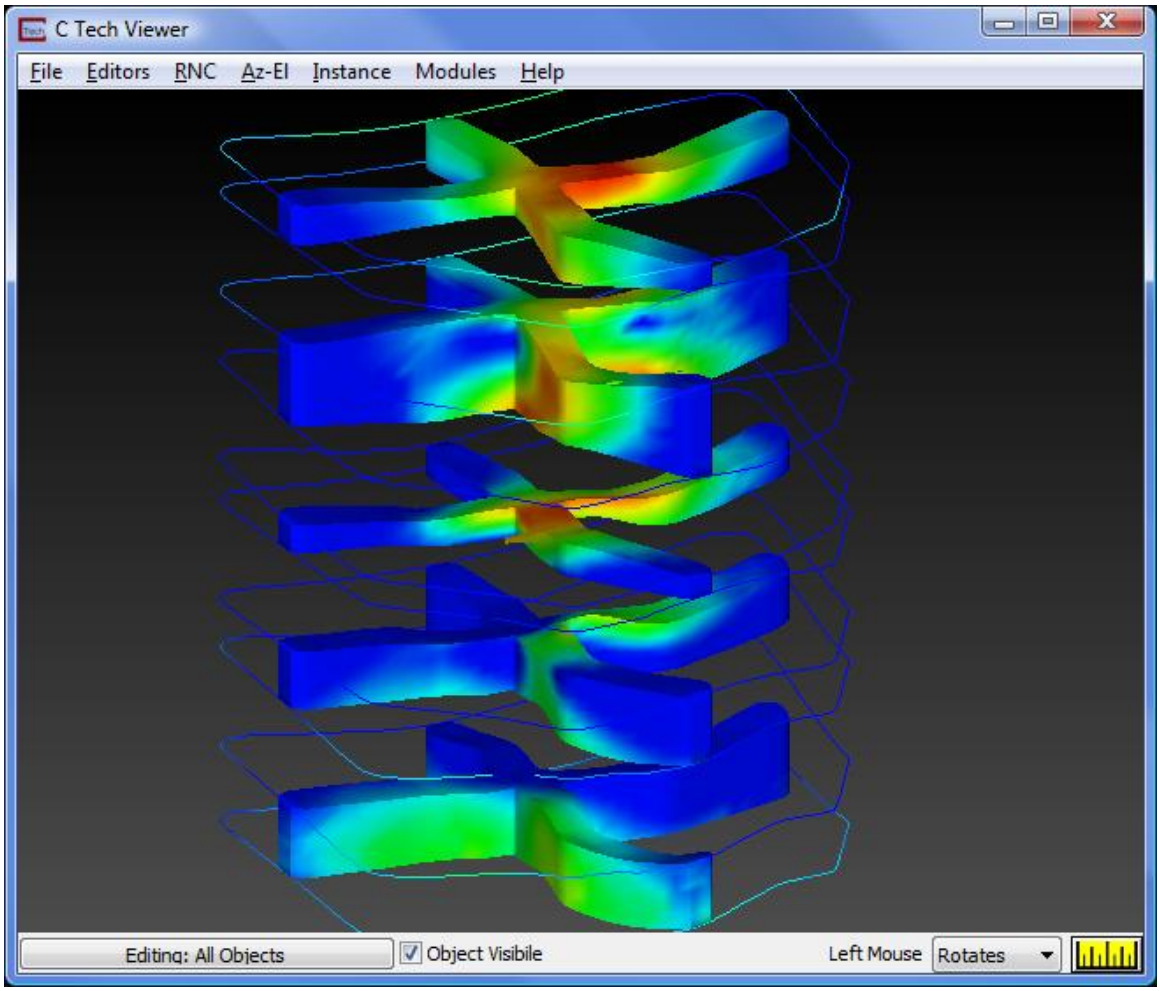
With these settings, each time we draw a new cross-section it will be appended to the previous one we've created.

We will now return to a top view and once again draw a cross-section. Draw a slightly jagged line from the top of the model to the bottom of the model. After drawing, your viewer should be similar to figure below.



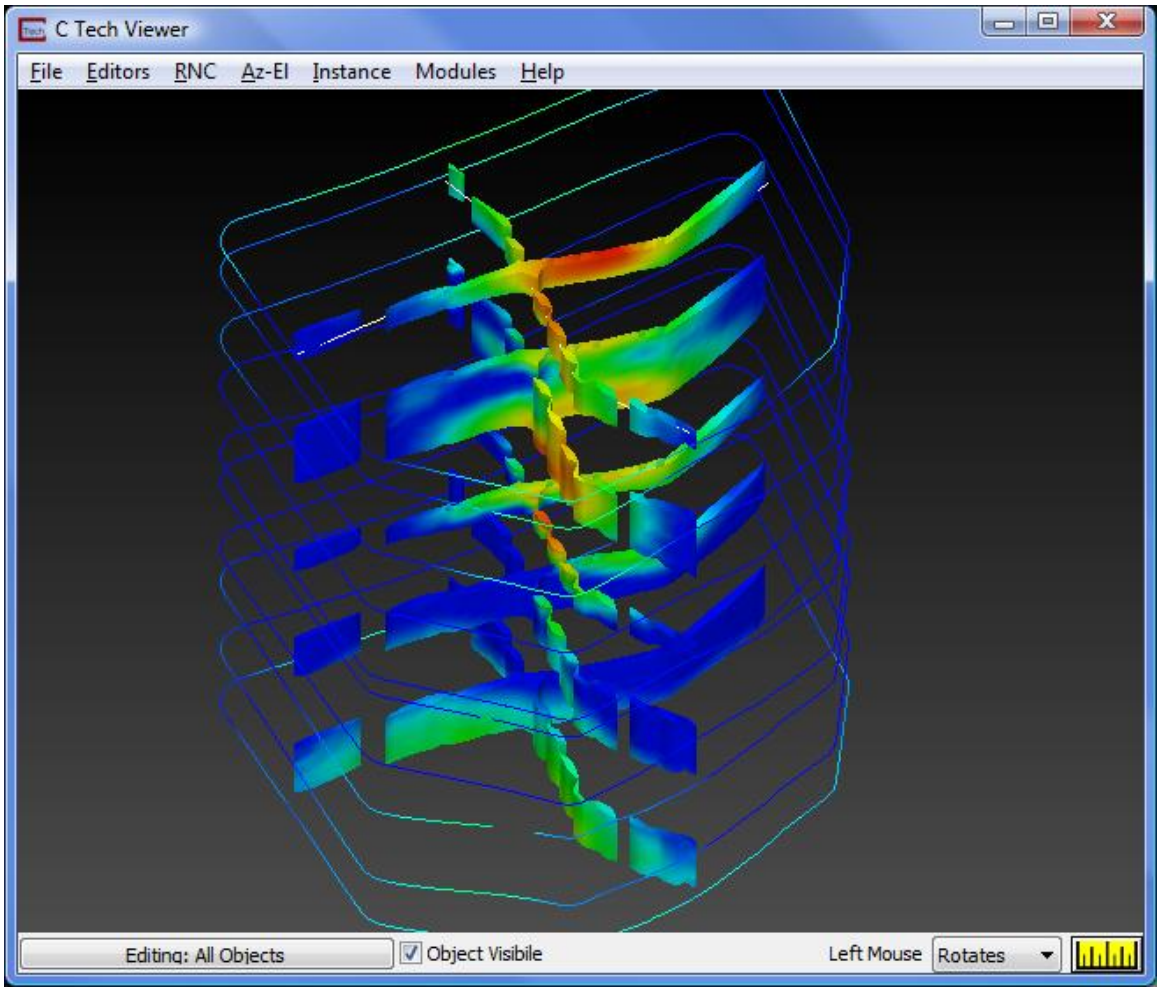
Using the Azimuth & Elevation window, choose a scale factor of 0.7, an elevation of 20, and an azimuth of 210. Also, turn off the Active toggle in draw_lines.

At this point your viewer should look similar to the figure below.

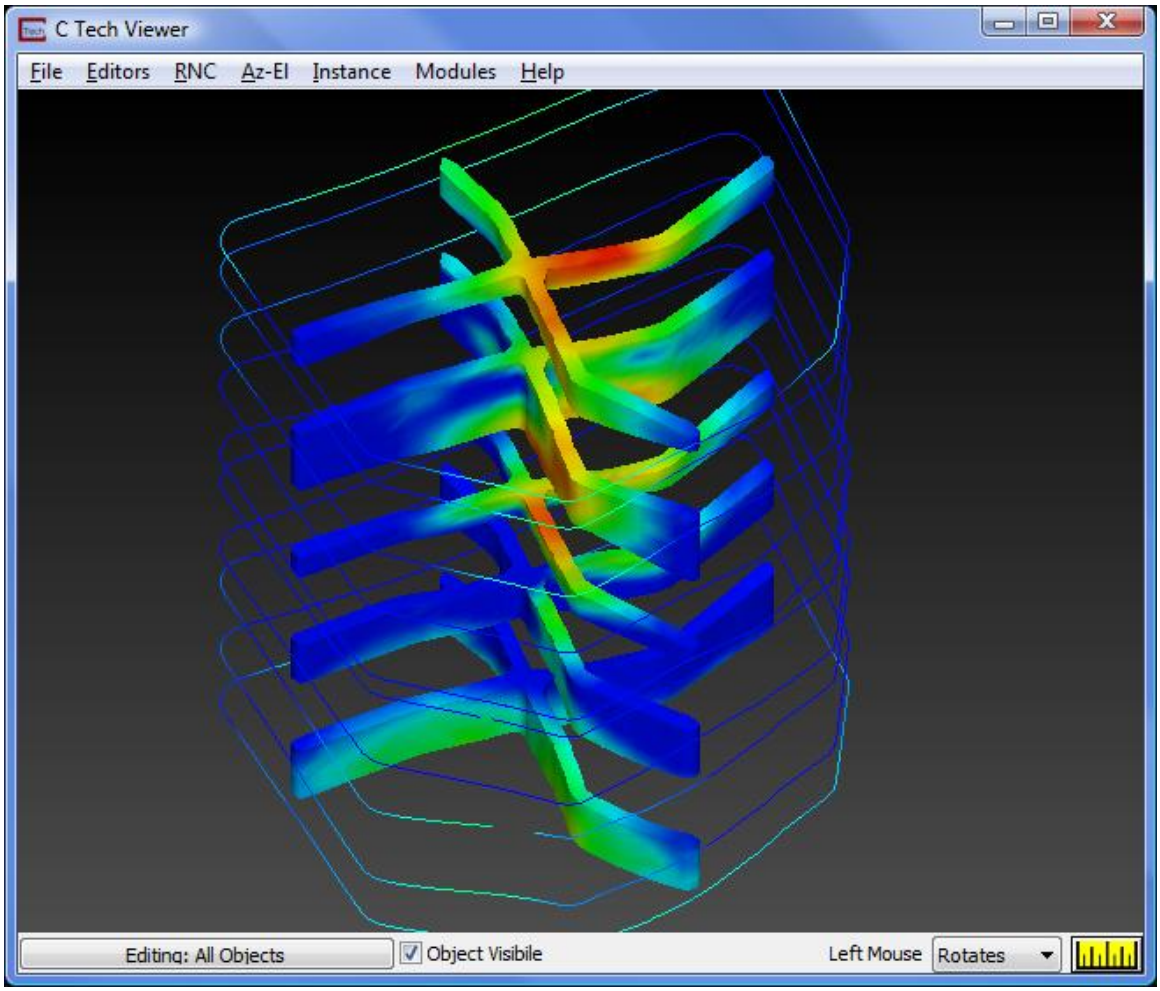


Fence_Cut Width Limitations

One of the limitations of the fence cut module is that the cross-sections it creates must have a thickness approximately equal to one cell width. If we change the Fence Half Width to a value of 5 (and click on Accept) and change the viewer settings to an elevation of 50° it becomes obvious that our fence cross-section is noticeably broken. You must remember that the plume_shell module performs the actual cutting. When the fence half width is set too low, the cross-sections we created will break-up. This is shown in the figure below.



However, a Fence Half Width of 8 works well for this grid resolution, as shown below

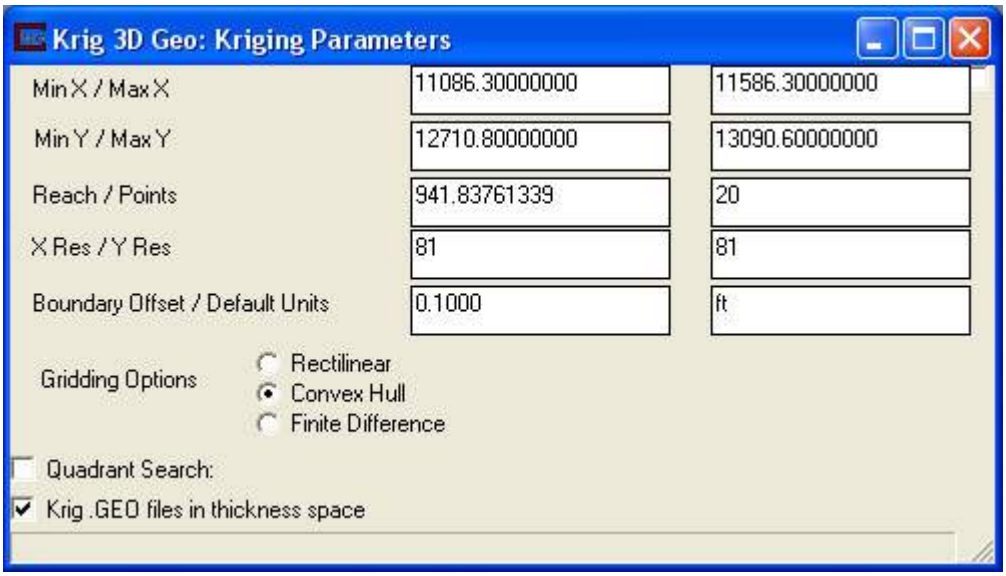


If a thinner fence is desired you have two options:

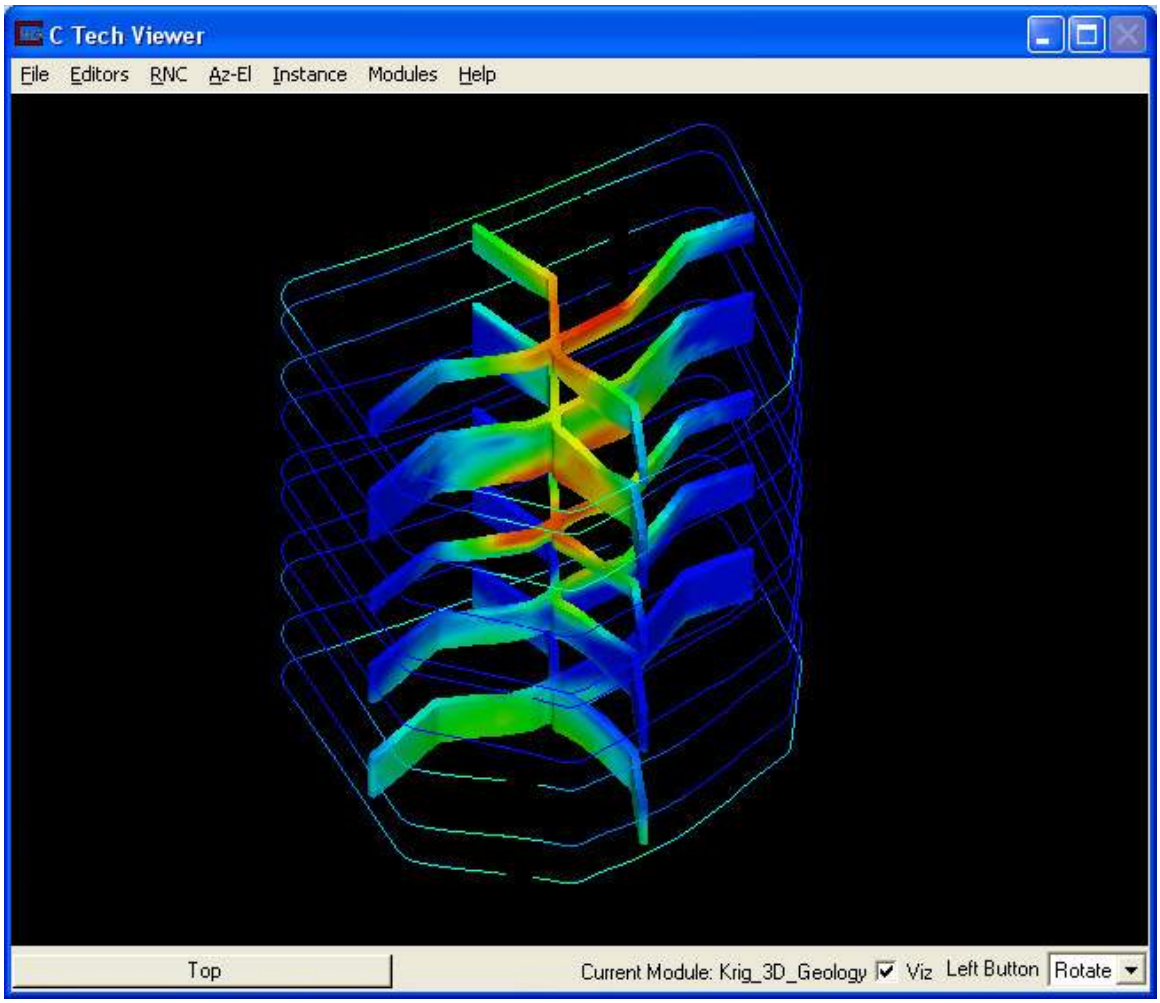
1. Increase the grid resolution in Krig_3D_Geology (this is where the resolution in the x-y plane is defined for our application) or
2. Use the module thin_fence which is discussed after the next topic

Thin Cross-Sections with Fence_Cut

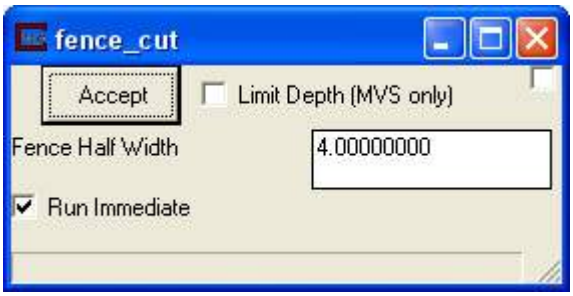
In order to create a fence diagram with a thinner cross-section we must increase the resolution of the grid in the x-y plane. For our network the x-y resolution is defined in the Krig_3D_Geology module's Gridding Options. Change the x-y resolution in Krig_3D_Geology both to 81. Your settings should match those in the figure below.



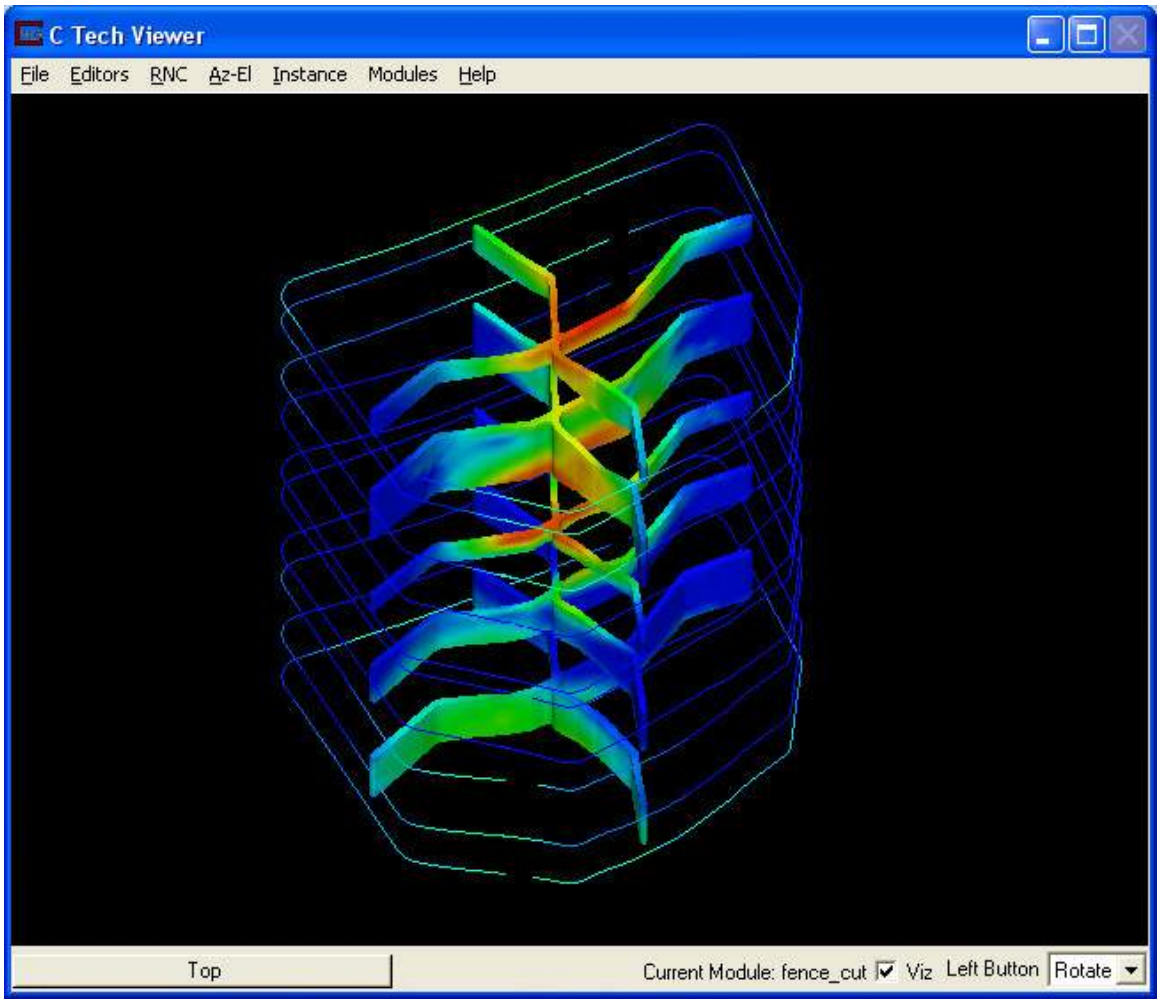
Press *Accept all Current Values* on Krig_3D_Geology. When all of the modules have completed running we can change the Fence Half Width back to 5 and click Accept on the fence cut module. At this point your viewer should be similar to the figure below.



With the resolution set to 81 we can actually set the fence that half width as low as 4.0 as shown in the figure below.



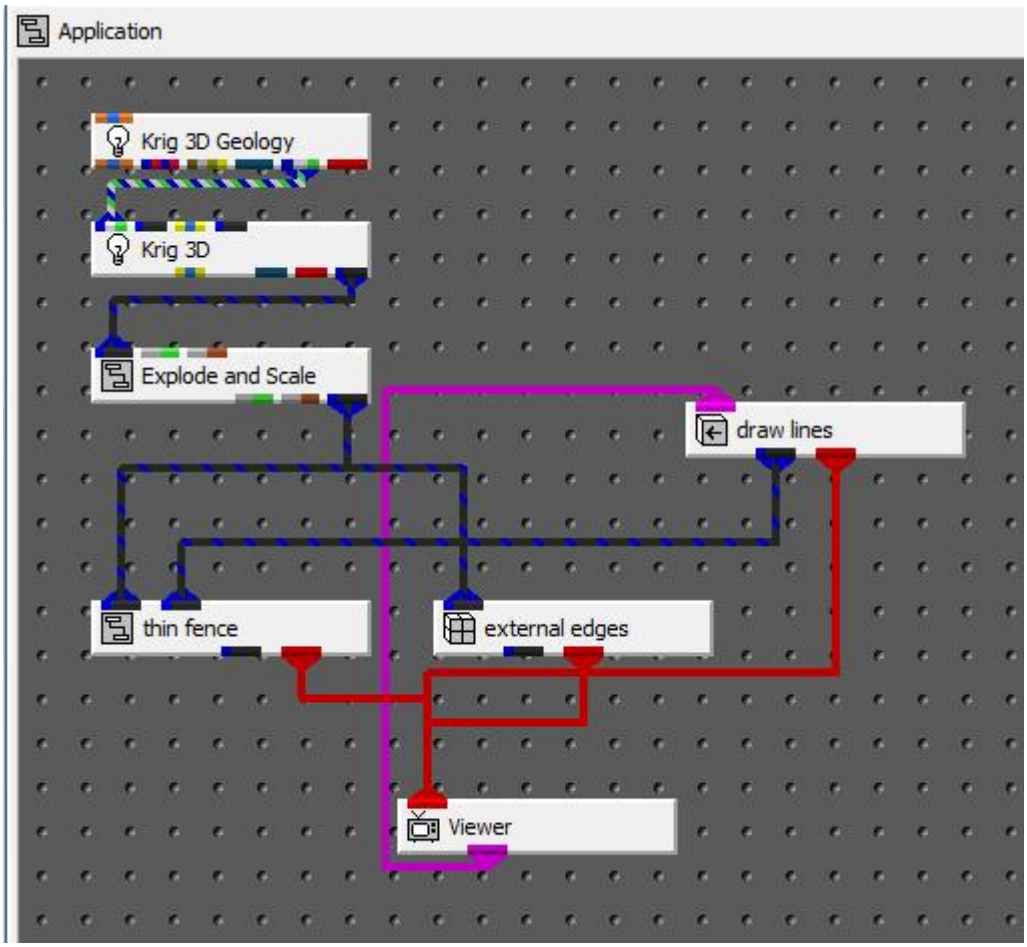
After hitting Accept, fence cut will run and the cross-section in your viewer should be similar to the figure below.



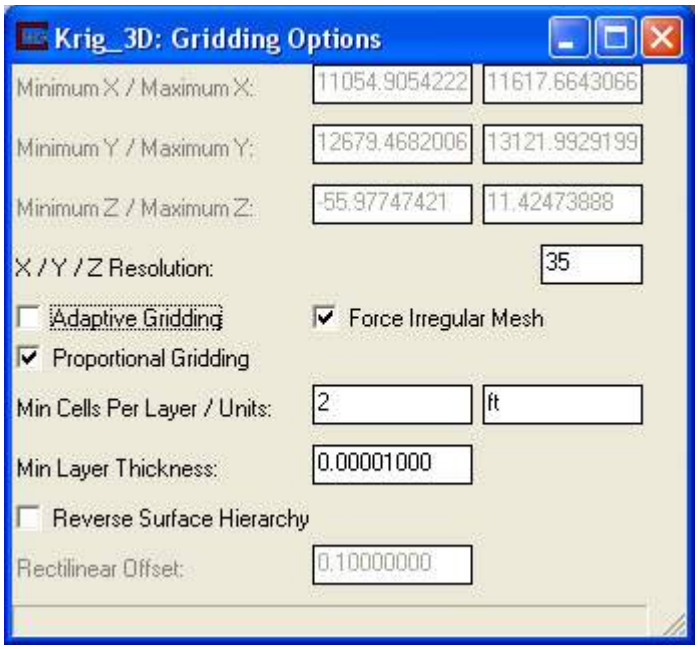
Fence Diagrams with the Thin_Fence Module

Thin_fence provides an alternative method of creating true 3D fence diagrams. Unlike fence_cut, it actually creates the fence (does the subsetting).

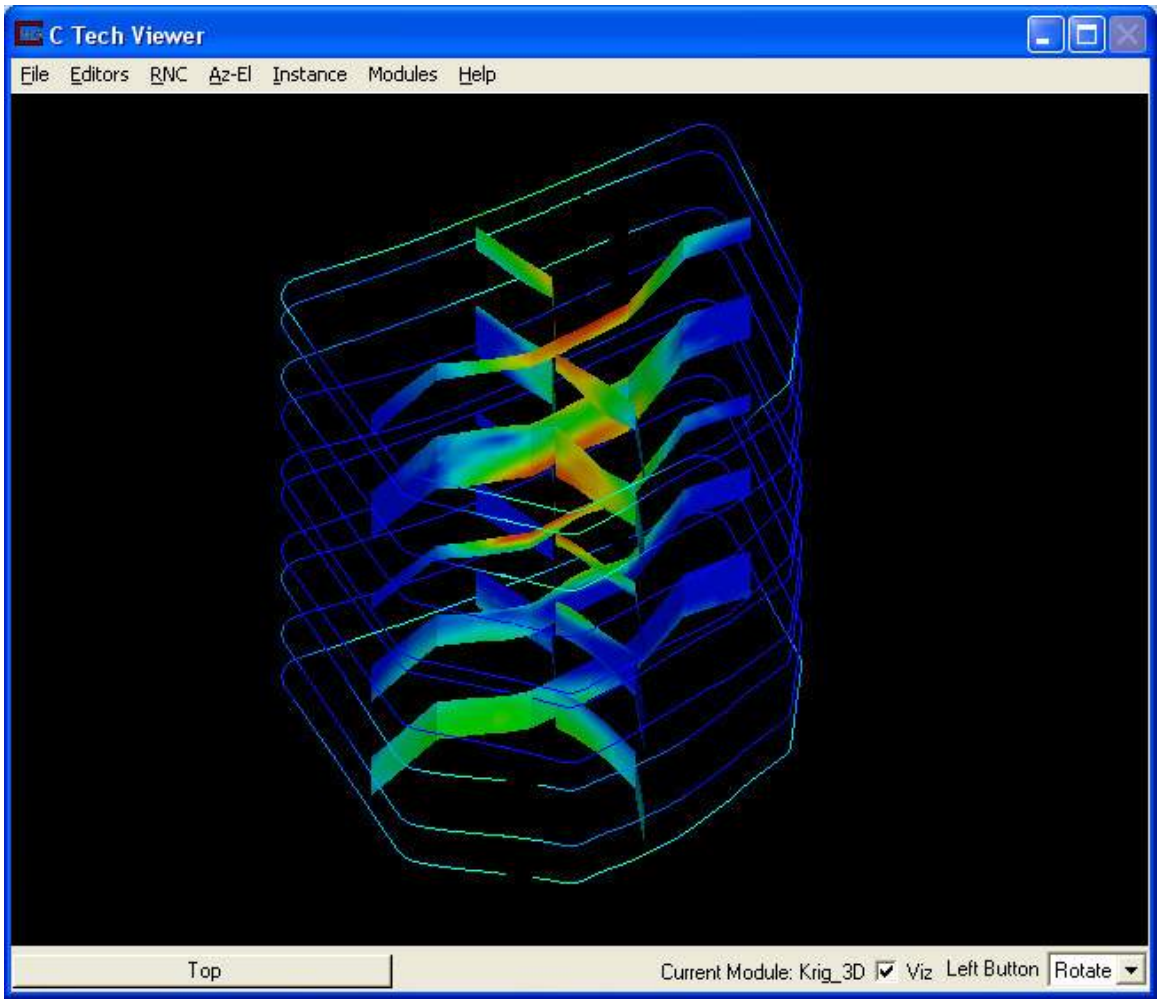
We will begin by modifying the application from the previous topic, deleting fence_cut and plume_shell and adding thin_fence to match that shown below.



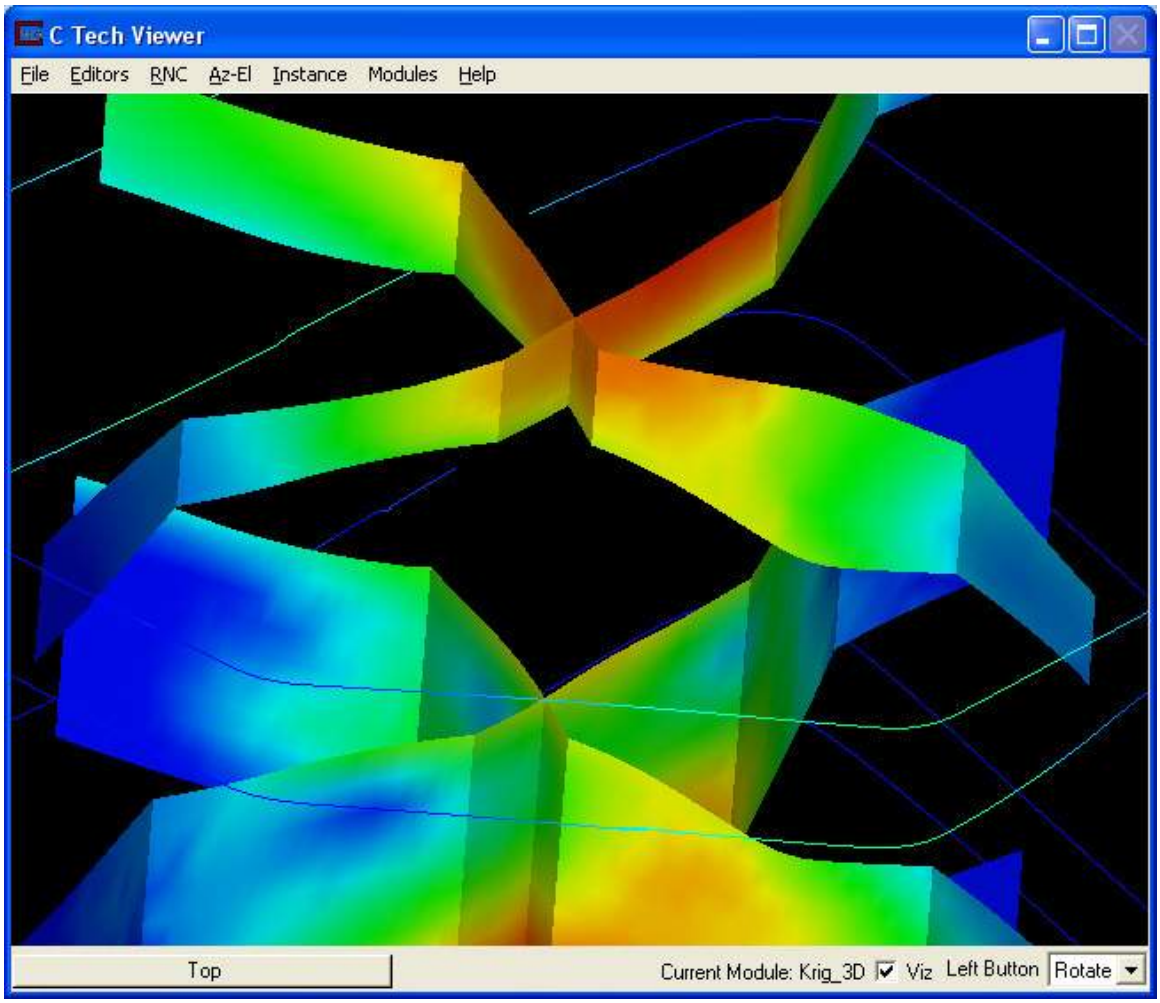
Open the Gridding Options window in Krig_3D and turn off Adaptive Gridding as shown below. Adaptive Gridding can cause problems with thin_fence.



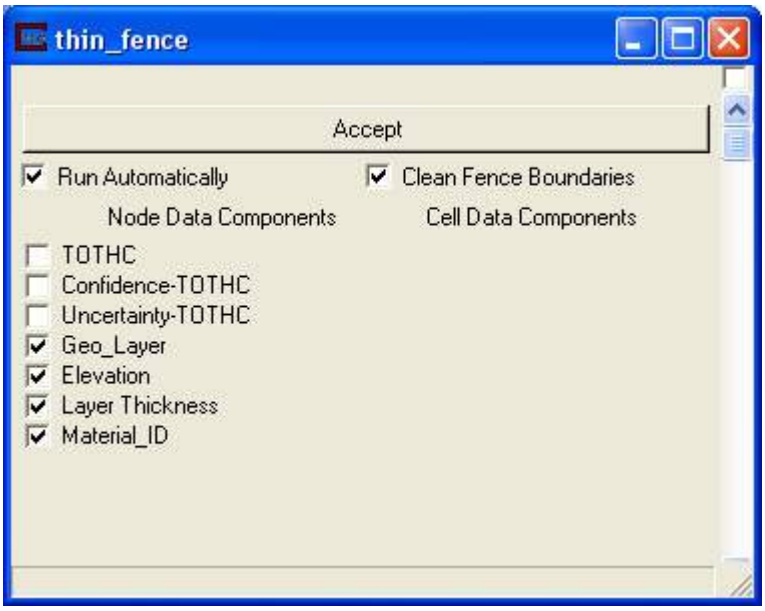
Hit Accept All Current Values on Krig_3D so it re-runs without Adaptive Gridding. At this point, your viewer should be similar to the figure below.



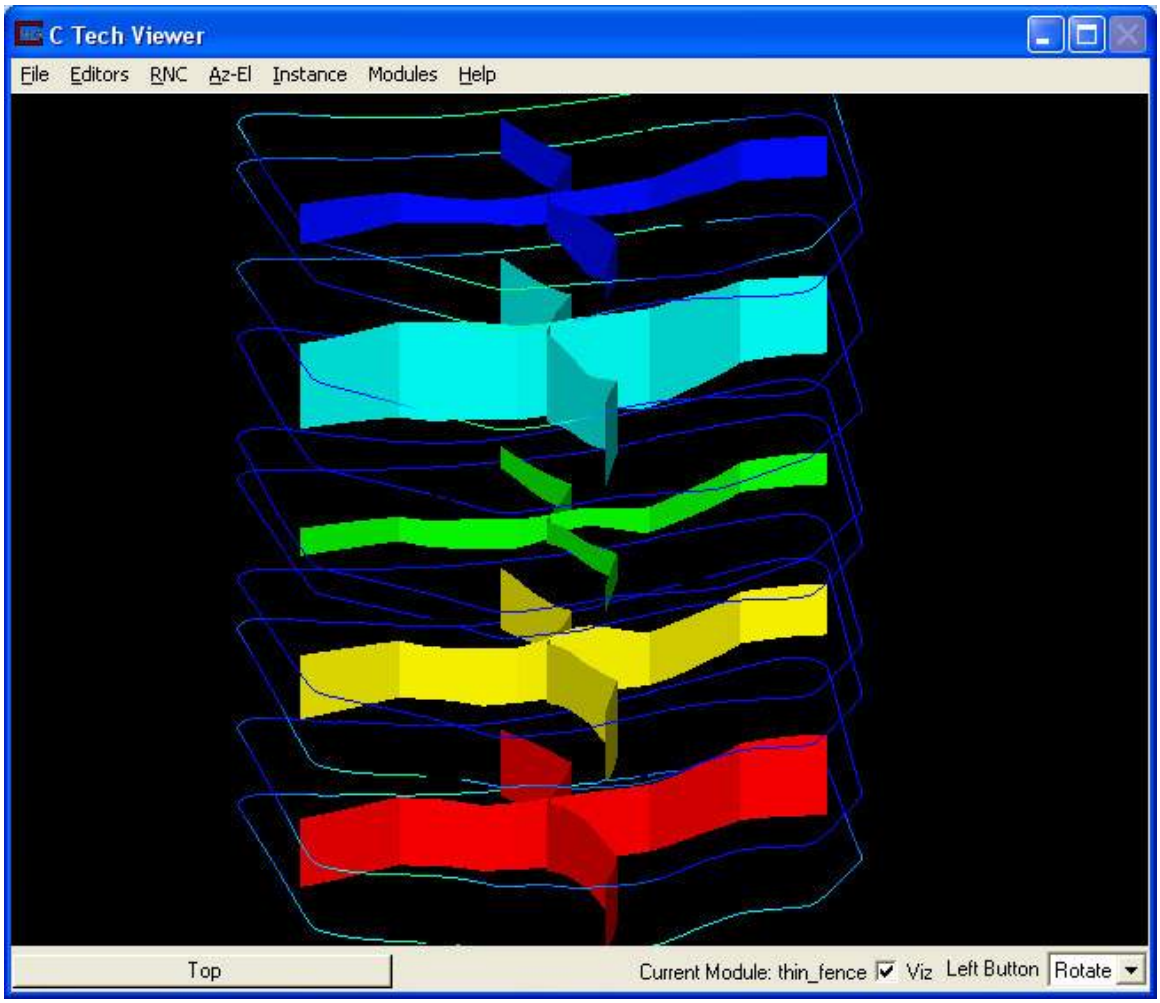
If we zoom in for a close up view, you can see that the fence-diagram produced with the thin_fence module has NO thickness whatsoever. It is a three-dimensional surface passing through our kriged volume upon which we map one or more of the data components in the original volumetric grid.



If we uncheck the first three Node Data Components in the thin fence module so that Geo_Layer is the first checked component, we can see our fence diagram colored by geologic layer. The correct settings are shown in the figure below.



Setting your Azimuth & Elevation to an azimuth of 195, a scale of 0.7, and an elevation of 20 should result in your Viewer looking very similar to the picture below.

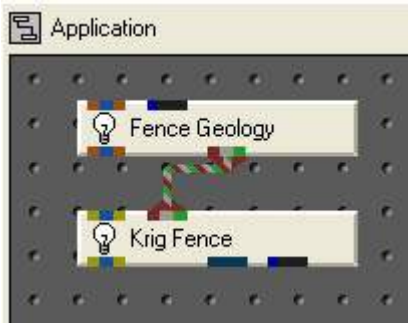


Using Fence_Geology and Krig_Fence

There is one final method for creating fence diagrams in EVS. This method uses two new modules, Fence_Geology and Krig_Fence, to create fence diagrams which pass directly through borings. There are limitations to this approach; however, it works with EVS-Standard.

The network fragment below shows the two primary modules used to produce a fence diagram with geologic layer information. Fence_Geology uses a .geo file to produce a 3D fence section which passes through ordered borings. Krig_Fence reads a .apdv file containing chemical data which is then mapped to the geologic layering within a fence cross-section produced by Fence_Geology. Fence diagrams can be produced without geologic information by eliminating the Fence_Geology module and having the .apdv file contain ordered borings.

Begin by creating a new application (File->New Application) then instancing and connecting these two modules. We will use a new technique to produce the final network.

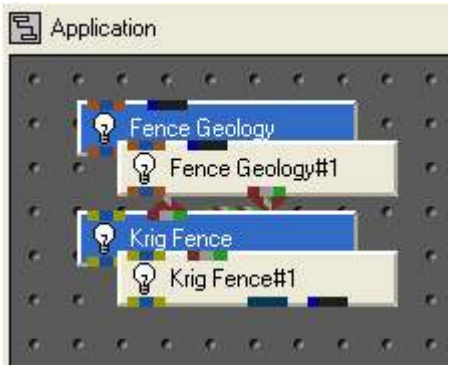


Edit.Duplicate Command

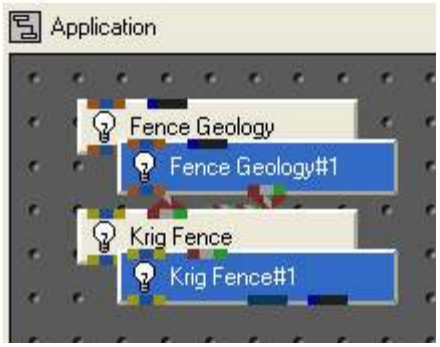
Begin by selecting both of these modules. Remember that once selected, the modules become blue. In the Network Editor main window, use the Edit pull down menu and select Duplicate. Duplicate reproduces the selected modules and attempts to retain all connections.

(Connections cannot always be maintained if one of the modules to be duplicated is connected to a module which will not be duplicated and supports only one connection.)

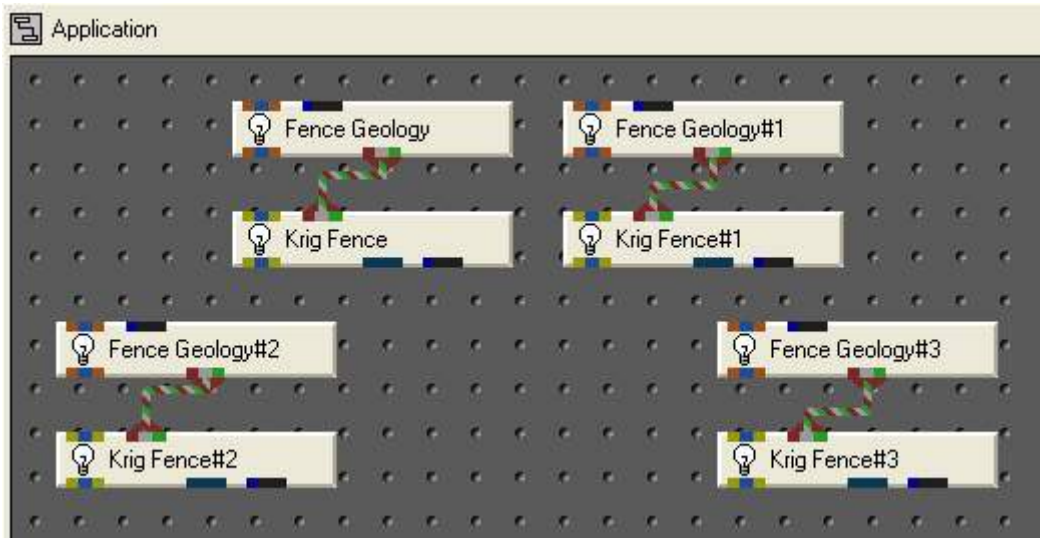
After the duplication your network should look like this:



The next step is to lasso the new pair of modules (with the #1 after their base names). Lassoing is done by clicking with the left mouse button and holding it down while dragging a box which intersects the modules to be selected. This is shown in the figure below.



Once these two modules are selected they can be moved as a group and duplicated again. Repeat the process until your network matches the figure below.



We will now add a few other modules which are required for our initial network.

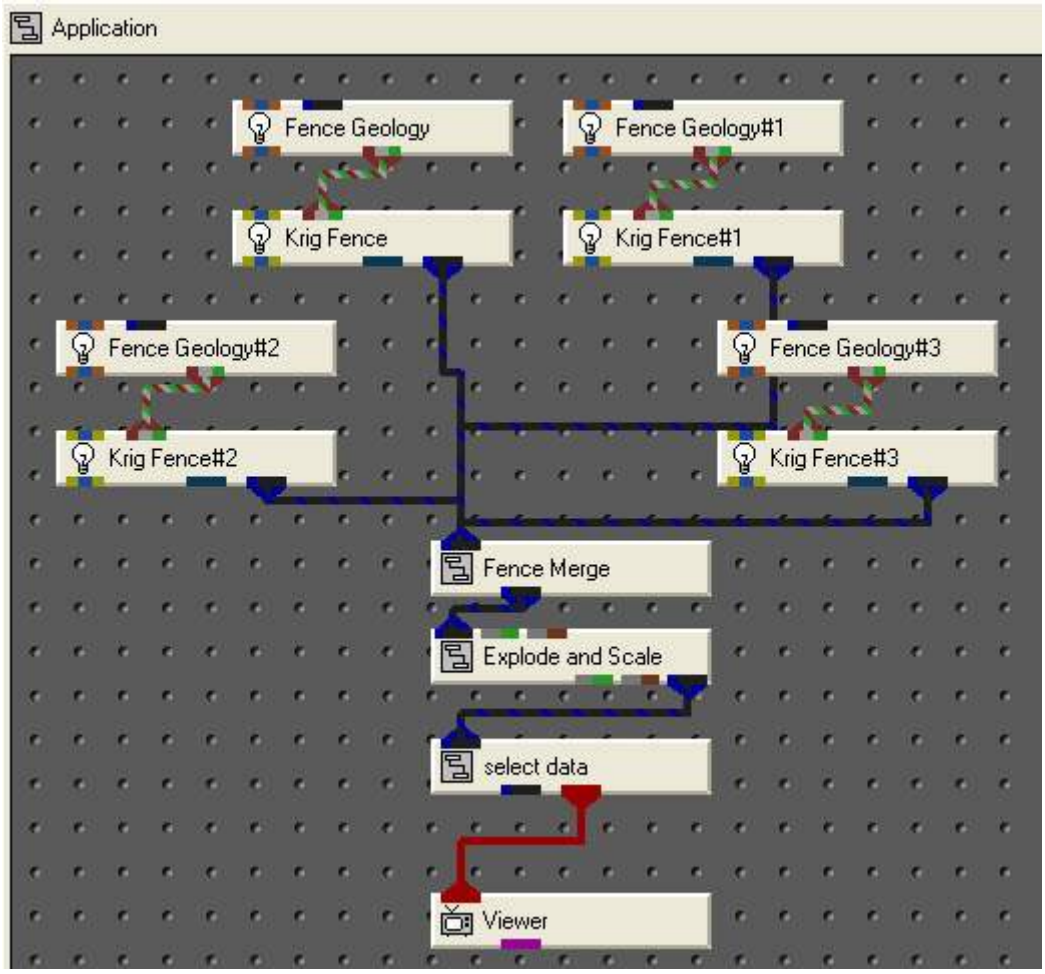
1) Fence_Merge combines the fence sections from up to 8 Krig_Fence modules into a single field. It allows us to perform other operations on all of the sections of several separate fence sections.

2) Explode_and_Scale allows us to exaggerate the z-coordinate and separate geologic layers.

3) select_data allows us to specify the nodal data component for color mapping.

4) The Viewer renders the output.

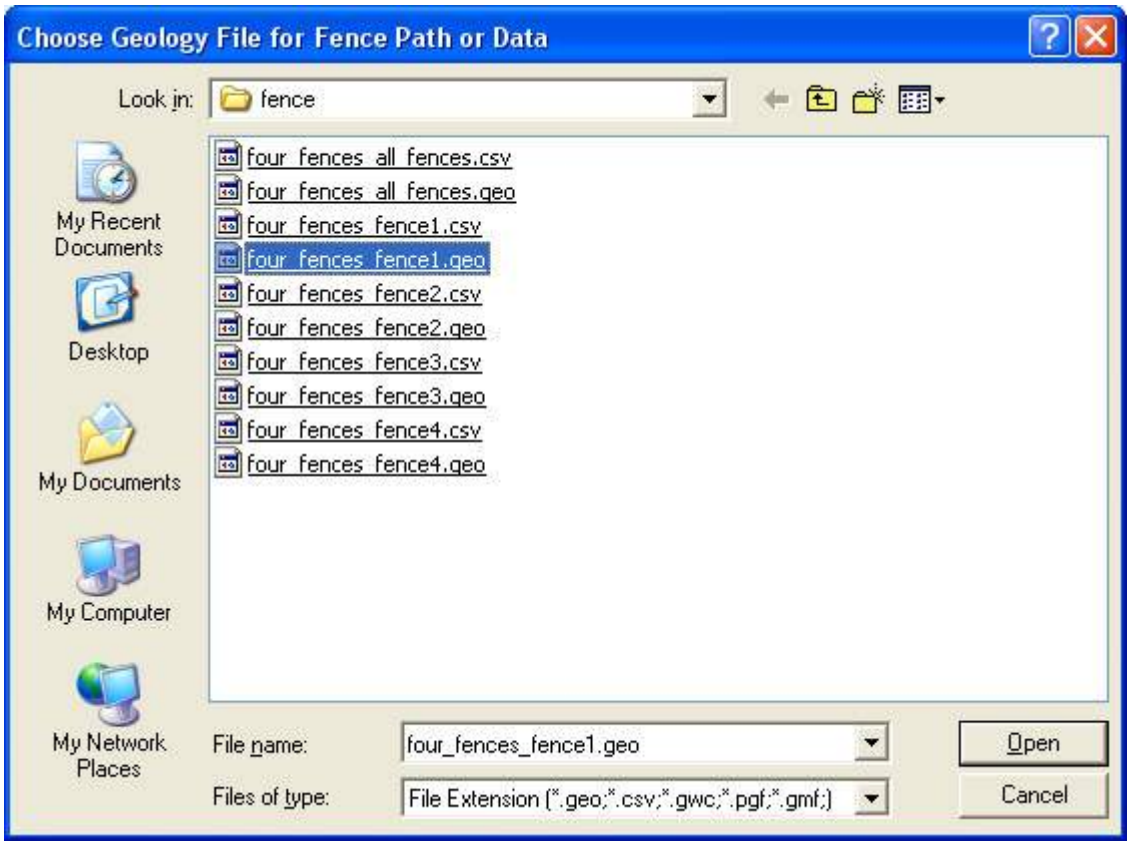
Your network should now look like:



Fence_Geology

From the Modules pull down list select Fence_Geology to reveal its main window:

Select Fence .GEO File so we can specify the geology file for input. Choose the file shown in the figure below:



(For this workbook, we strongly recommend that when a .apdv or .geo file is specified, you should open the file in an editor and make sure that you are familiar with the file format and understand its content.)

Detailed information regarding the options and parameters for this and other modules used in this workbook are given in the Main Help file under Module Libraries. You can also select any module and use the right mouse button to jump directly to help on that module.

We will use the default values for all other parameters. Choose "Accept All Current Values" to execute Fence_Geology. When completed (in just a few seconds), your console should have the following messages.

```
Reading .geo data from
c:\ctechreleases\8.0b6\data\fence\four_fences_fence1.geo
File check passed


Reading fence path data from
c:\ctechreleases\8.0b6\data\fence\four_fences_fence1.geo
*** Surface 0 ***
Computing Pairs: Done
21 Pairs in Semivariogram model: Range = 125.366311 Sill =
0.050000
51 points Kriged for surface 0 using 1 matrices in 0.00 minutes
*** Surface 1 ***
```

```
Computing Pairs: Done
21 Pairs in Semivariogram model: Range = 119.403781 Sill =
6.844386
51 points Kriged for surface 1 using 1 matrices in 0.00 minutes
*** Surface 2 ***
Computing Pairs: Done
21 Pairs in Semivariogram model: Range = 104.251426 Sill =
27.399565
51 points Kriged for surface 2 using 1 matrices in 0.00 minutes
*** Surface 3 ***
Computing Pairs: Done
21 Pairs in Semivariogram model: Range = 204.927701 Sill =
17.091279
51 points Kriged for surface 3 using 1 matrices in 0.00 minutes
*** Surface 4 ***
Computing Pairs: Done
21 Pairs in Semivariogram model: Range = 104.251426 Sill =
38.777106
51 points Kriged for surface 4 using 1 matrices in 0.00 minutes
Layer 1 is Layer_0 with material color 1
Layer 2 is Layer_1 with material color 2
Layer 3 is Layer_2 with material color 3
Layer 4 is Layer_3 with material color 4
```

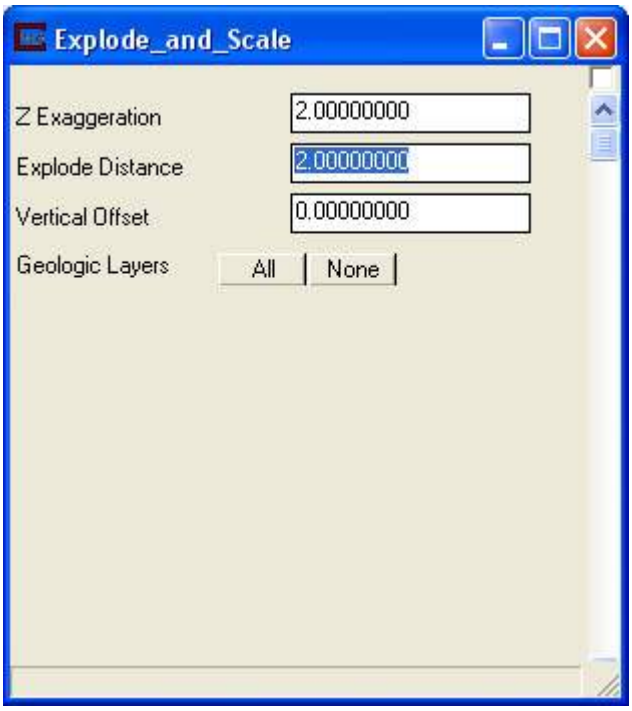
Close Fence_Geology by double clicking in the upper right had corner of that window.

Select the user interfaces for Fence_Geology#1 through Fence_Geology#3 selecting the files four_fences_fence2.geo through four_fences_fence4.geo respectively. As you select files for each, also click on the "Accept All Current Values" button so they will each run. As each is run, similar messages to the set above will be printed to the console. The reason that the Total time for kriging each layer is 0.00 minutes is that with only 51 points along the length of the fence, each layer takes less than 1/100th of a minute.

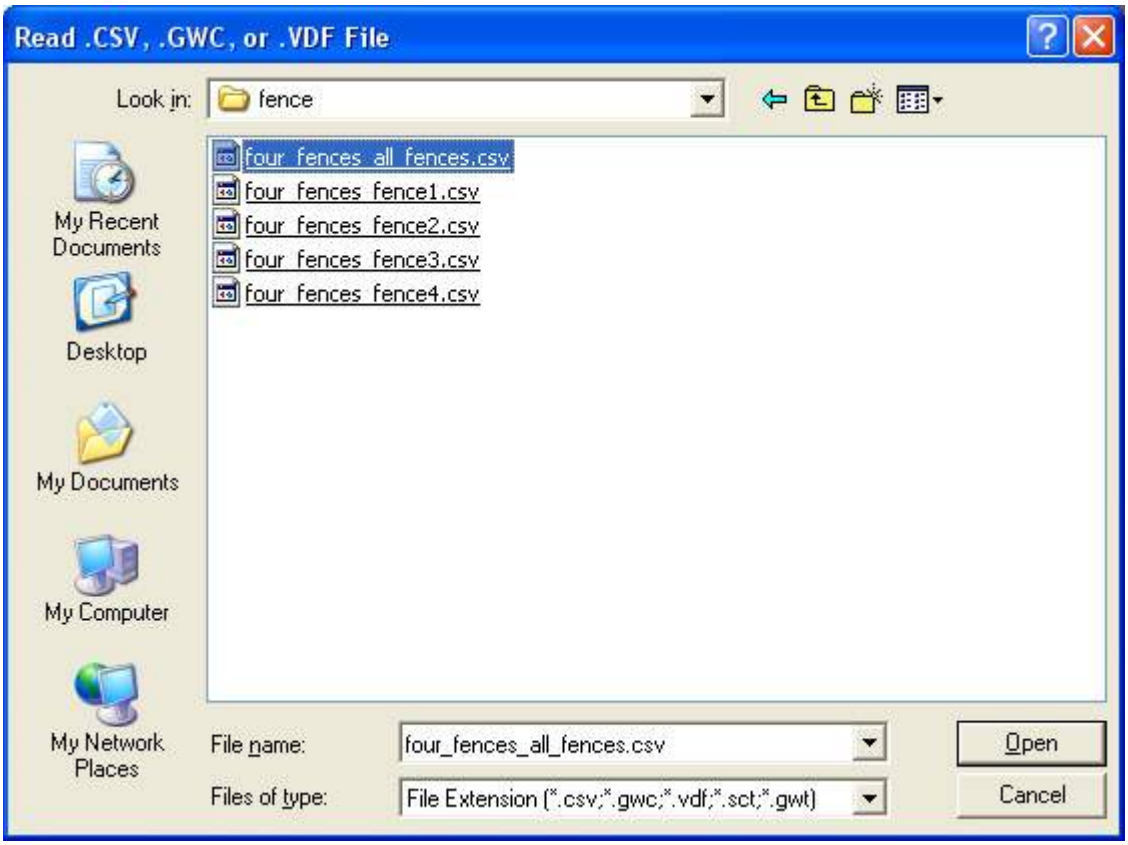
Krig_Fence

Now we are going to do things a little backwards (for good reason). Select the user interface for Krig_Fence#3. Why? We want to execute the Krig_Fence#x modules so that Krig_Fence is executed last. The reason for this is due to the nature of the Fence_Merge module to which these are connected. Fence_Merge will not run until there is a valid output from the FIRST module connected to it. By running Krig_Fence last, we avoid having to recompute and render the network each time we execute a Krig_Fence module.

Before we read a analyte (e.g. chemistry) file, we will first change the parameters of the Explode_and_Scale module to match the figure below.



Now select Read .Chem File so we can specify the analyte (e.g. chemistry) file for input. Choose the data\fence\four_fences_all_fences.apdv file shown in the figure below:

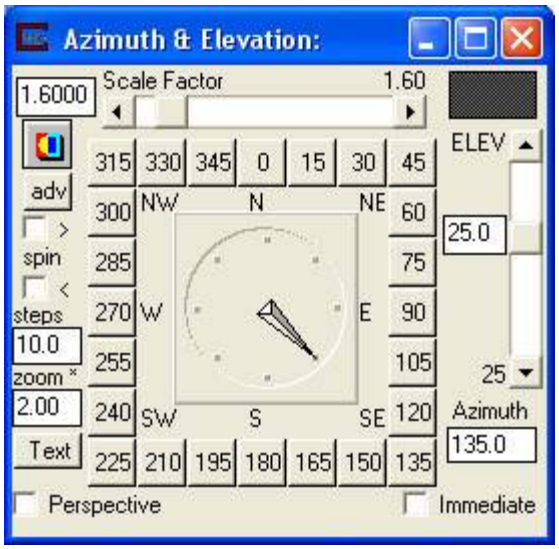


Click on "Accept All Current Values" so that Krig_Fence#3 executes. When it does, the following message will be printed to the console.

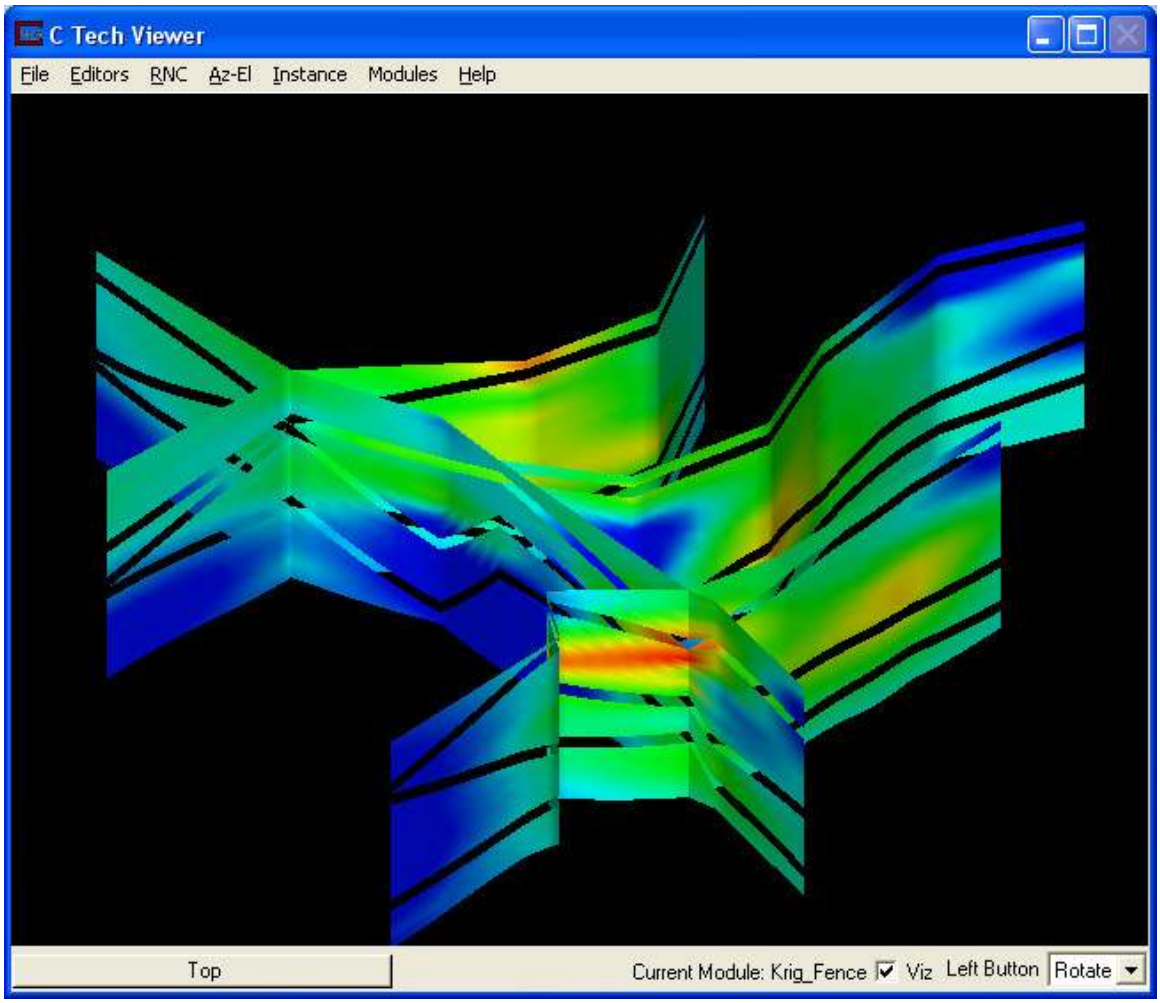
```
Reading .apdv data from
c:\ctechreleases\8.0b6\data\fence\four_fences_all_fences.apdv
Computing Pairs: Done
5442 Pairs in Semivariogram model: Range = 270.580922 Sill =
3.426106
1900 points Kriged for analyte 0 using 1122 matrices in 0.00
minutes
Concentration0 : Max Uncertainty at node 1883: x= 11398.85 y=
12733.81 z= -31.89
Concentration= -1.378 Uncertainty= 1.19
Concentration0: Min -3.000 Max 3.487
Confidence-Concentra: Min 46.800 Max 100.000
Uncertainty-Concentr: Min 0.000 Max 1.191
Geo_Layer: Min 0.000 Max 3.000
Elevation: Min -32.504 Max 0.000
Layer Thickness: Min 0.000 Max 21.250
Material_ID: Min 1.000 Max 4.000
```

Next select Krig_Fence#2 from the module pull down list, and read the same analyte (e.g. chemistry) file 'four_fences_all_fences.apdv', then Krig_Fence#1, and finally Krig_Fence. Be sure to click on "Accept All Current Values" in each module. When you execute the Krig_Fence module (the last one) the remainder of the network will execute.

Nothing will appear in the Viewer until we select the view orientation. Match the settings in the Az-El panel shown below:



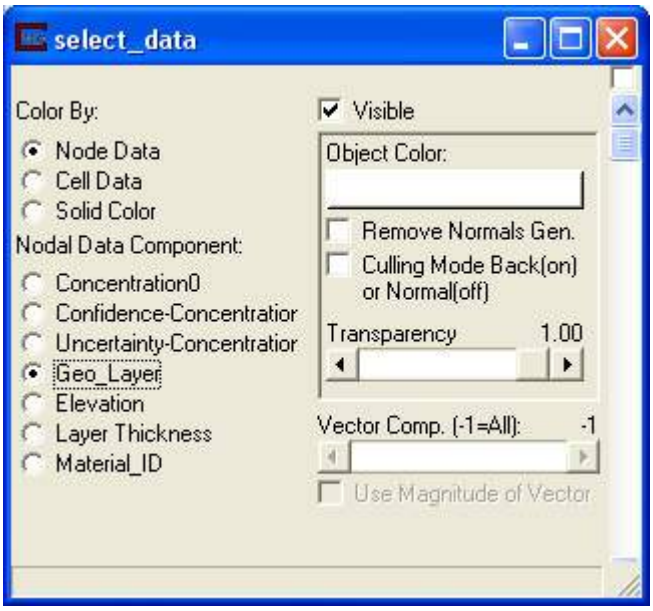
Your viewer should now have 4 fences as shown below.



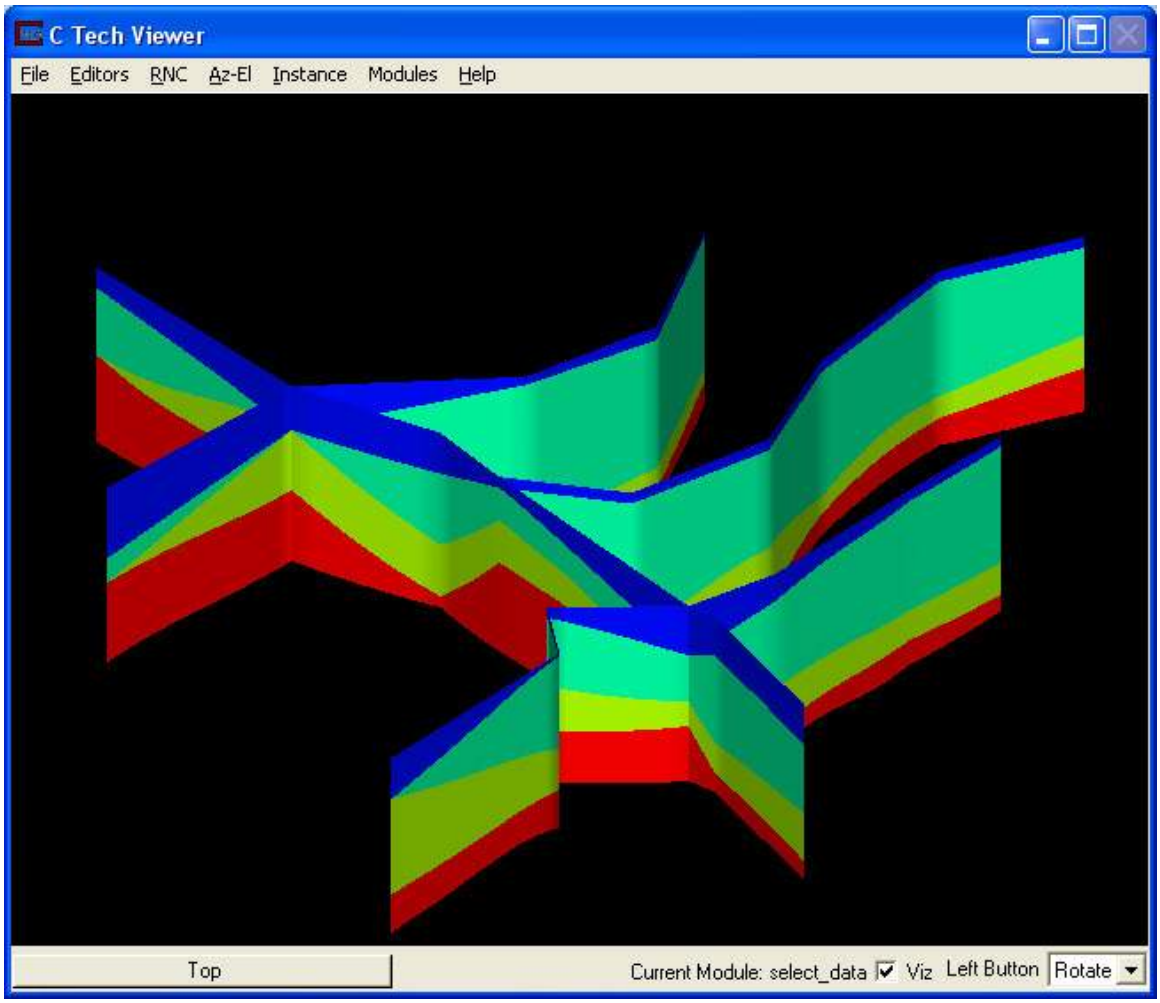
select_data

Select the Explode_and_Scale user interface again and set the Explode Distance to 0 ft. (feet are our user units).

Now we will select the select_data module interface to examine the geology. Choose "Geo Layer" instead of the default Concentration0 data component.



Press the multi-colored (Reset/Normalize/Center) button in Azimuth & Elevation. Your Viewer should look like this:

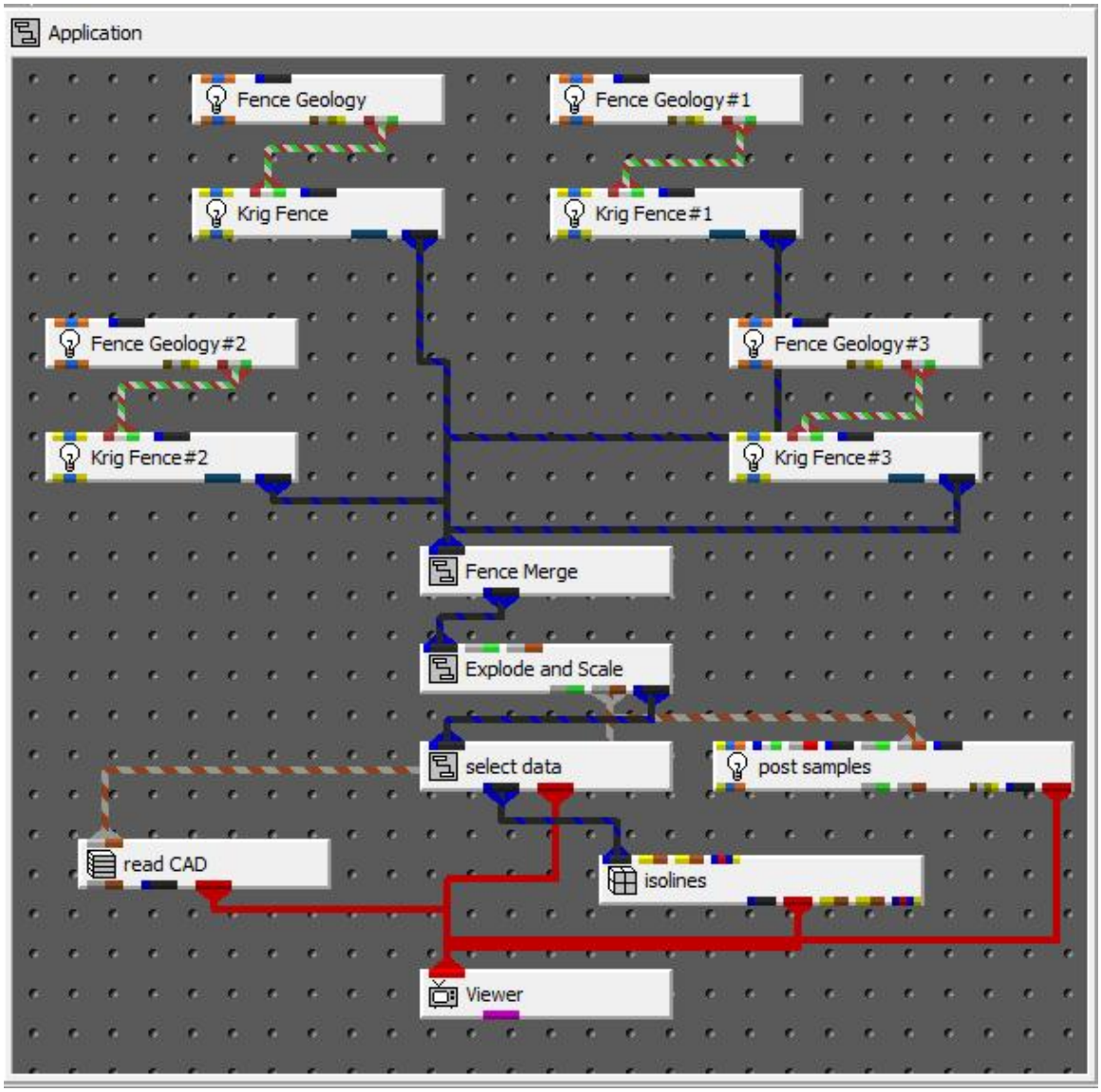


The default colors are not the most appropriate for geology. Although it is possible to use the Datamap_Editor to set these colors in a variety of ways, we will not do that now.

Go back to the select_data module and reselect the default Concentration0 data component.

Adding Measured Data Posting

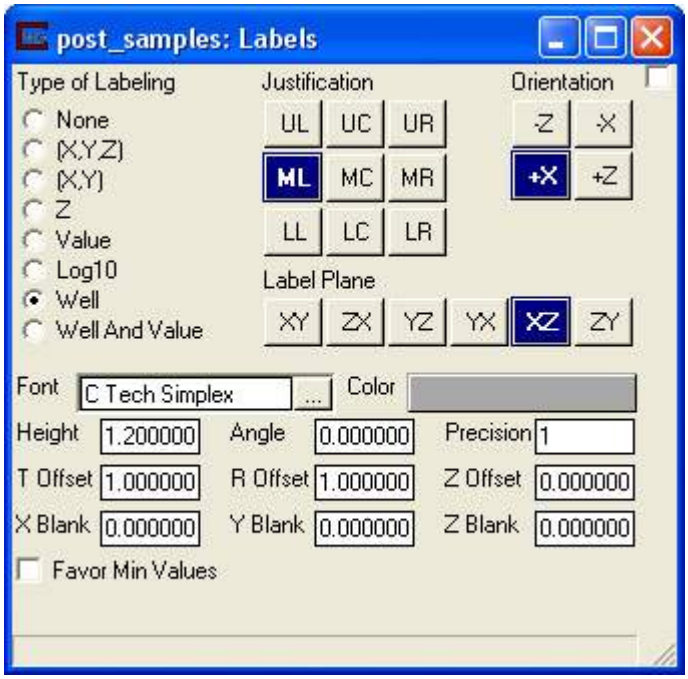
Now we will post our measured data as spheres, overlay a drawing showing building outlines and add isolines to the fences. Add three new modules to your network to match:



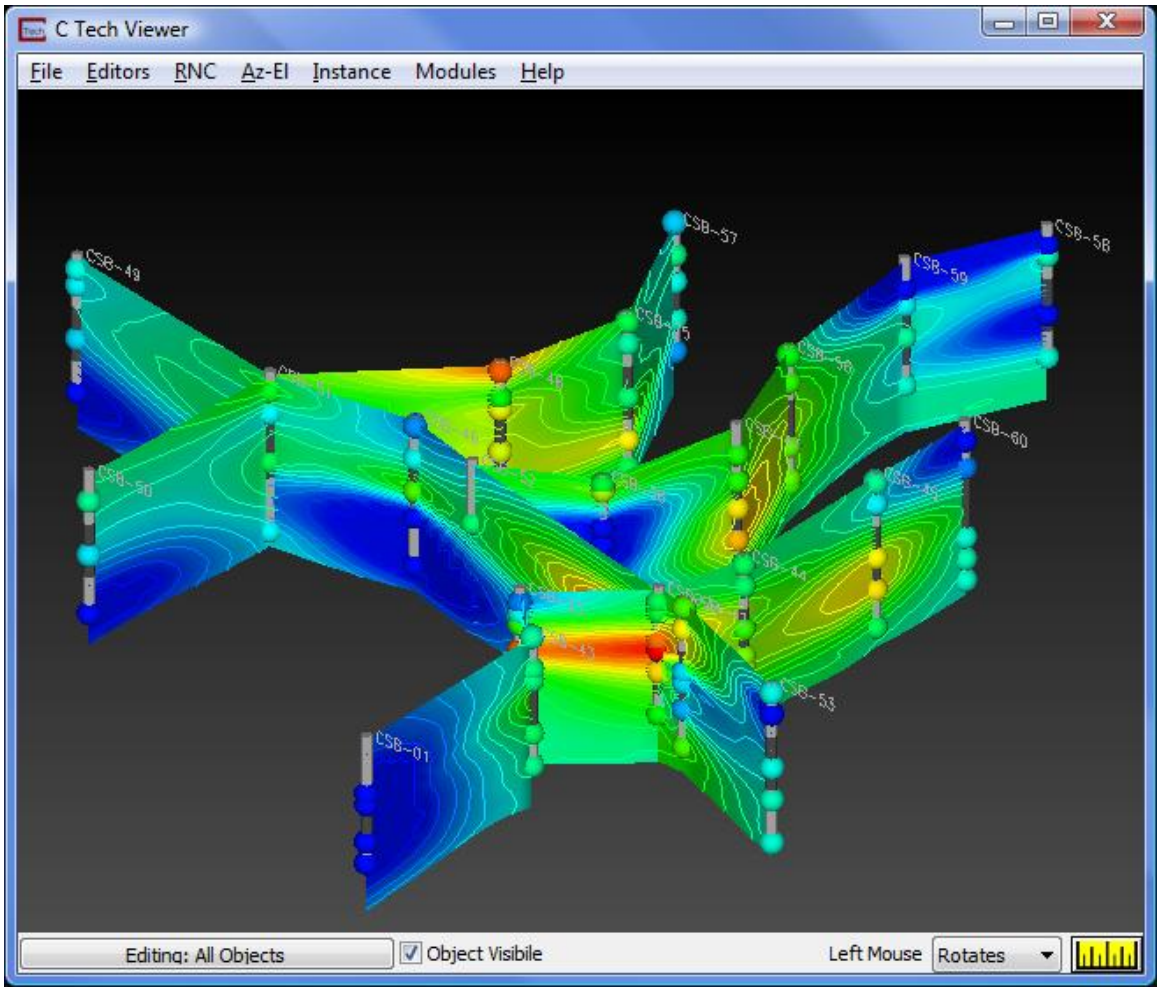
We will start with adding spheres and boreholes corresponding to our measured data. We have already used the post_samples module in previous workbooks, so we won't cover its options in detail.

The first thing we will do is read the .apdv file which contains all four fence section data samples. Choose Load File in post_samples and select four_fences_all_fences.apdv in the fence folder.

Now activate the Labels window of post_samples and choose the 'Well' option for Type of Labeling, as shown below. Also, set the label height to 1.2.

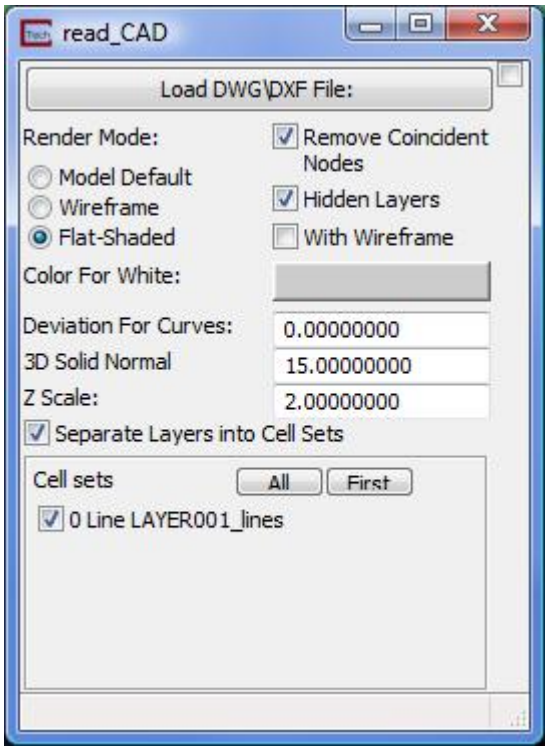


Click on the multi-colored (Reset/Normalize/Center) button in Azimuth & Elevation to achieve a view like this:

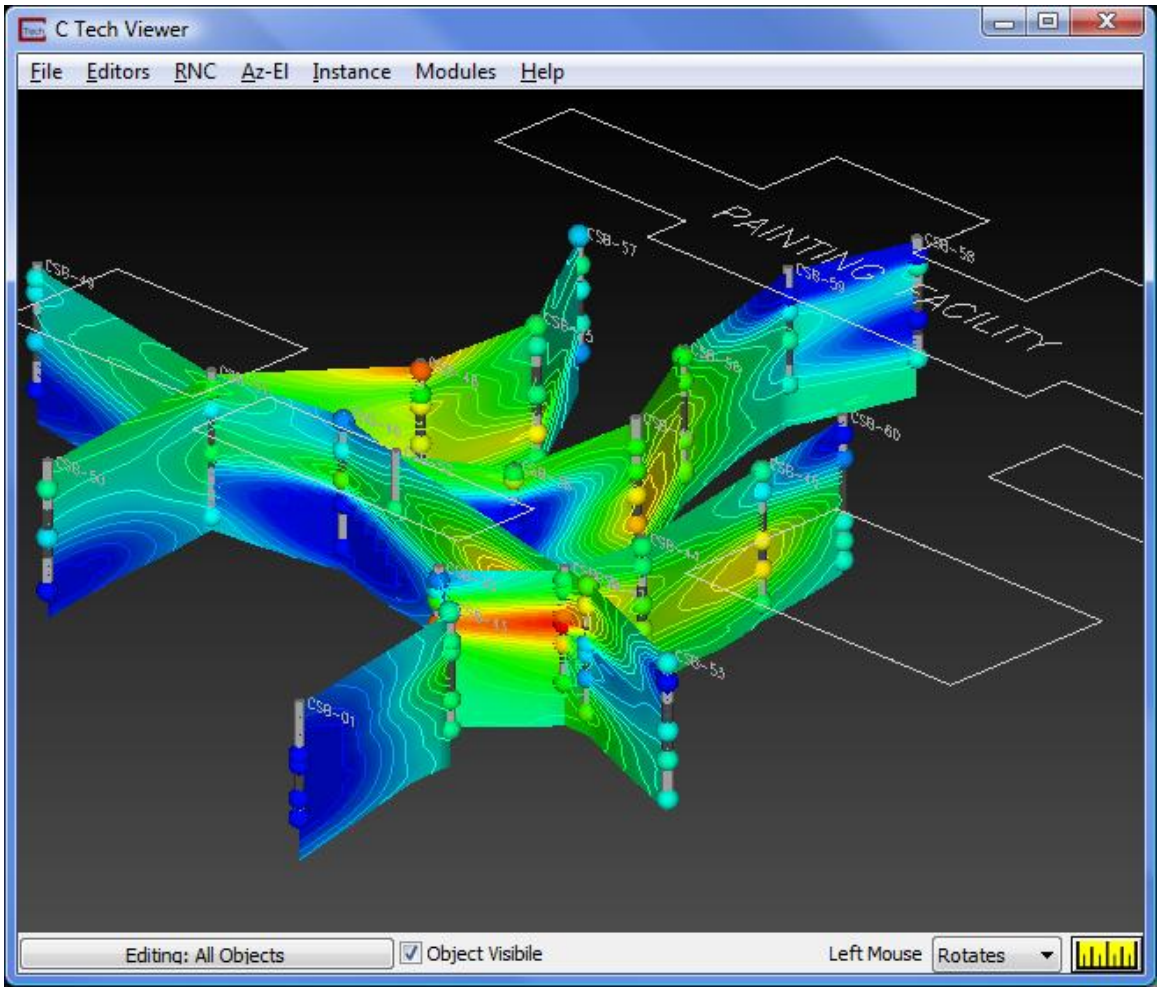


Overlaying CAD Files

Now we will add the overlay a CAD file (in .dwg or .dxf format). Select read_CAD and the following window will appear.



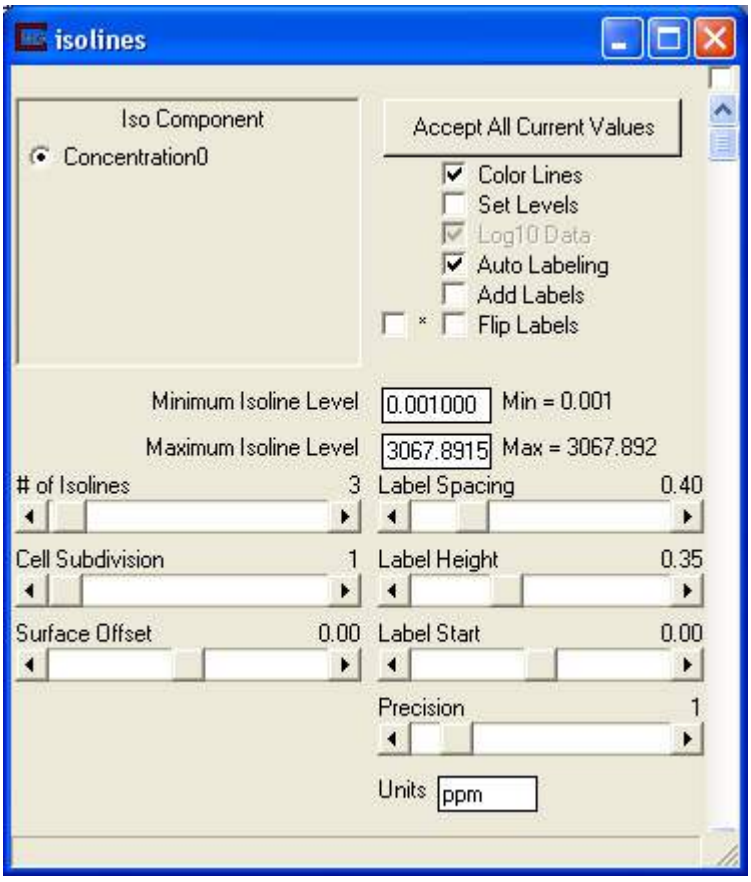
Click on the "Load DWG/DXF File" button and select initial_soil_investigation_subsite_bldg_layout.dxf in the file browser. Another Reset/Normalize/Center will cause your viewer to look like this:



The read_CAD module provides parameters to vertically scale CAD file, however if it is not in the right coordinate system you may need to use other modules such as transform_field or project_field to properly place the file. These modules are discussed in the "Module Libraries" section of the main help.

Looking at isolines

Select isolines from the Modules pull down menu. We use a Surface Offset of 0.0 by default for a reasons explained below.



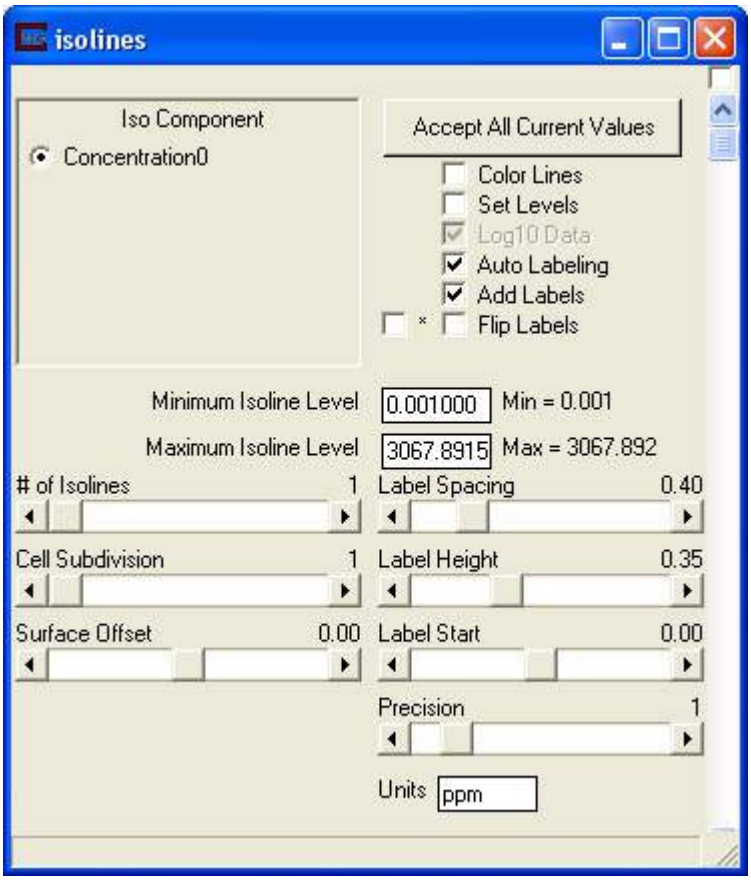
One frequently adjusted parameter for isolines is the # of Isolines slider. This option allows us to change the density of lines depending on the distribution of the attribute. The number of isolines slider is interpreted two different ways depending on whether or not the log10 check box is selected. If the log10 check box is selected, the number of isolines refers to the number of isolines per decade. The logarithmic option allows for non-integer data limits. The default number of lines is three and the maximum (for log option) is three. Isolines are placed on specific intervals within each decade. It will create isolines at suitable intervals dependent on number of isolines selected.

```
1 will give even decades.
2 will give lines at 1, 3, 10, 30, etc.
3 will give lines at 1, 2, 5, 10, 20, 50, etc.
```

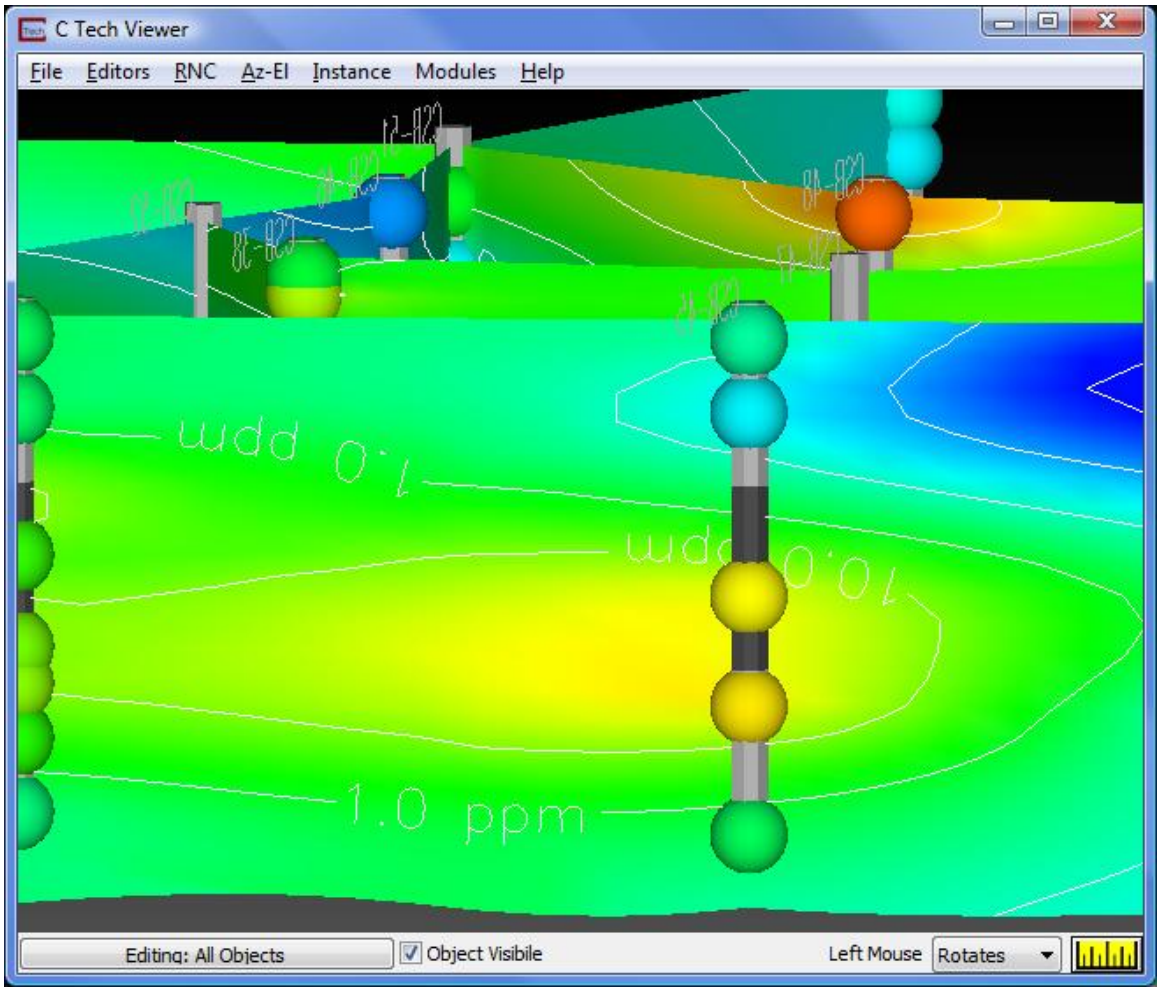
If the log10 check box is not selected, then the number of isolines refers to the total number of isolines to be drawn on the surface. The default is three and the maximum allowable is 100. isolines are placed at equal intervals based on (max-min)/number of isolines.

Labeling Isolines

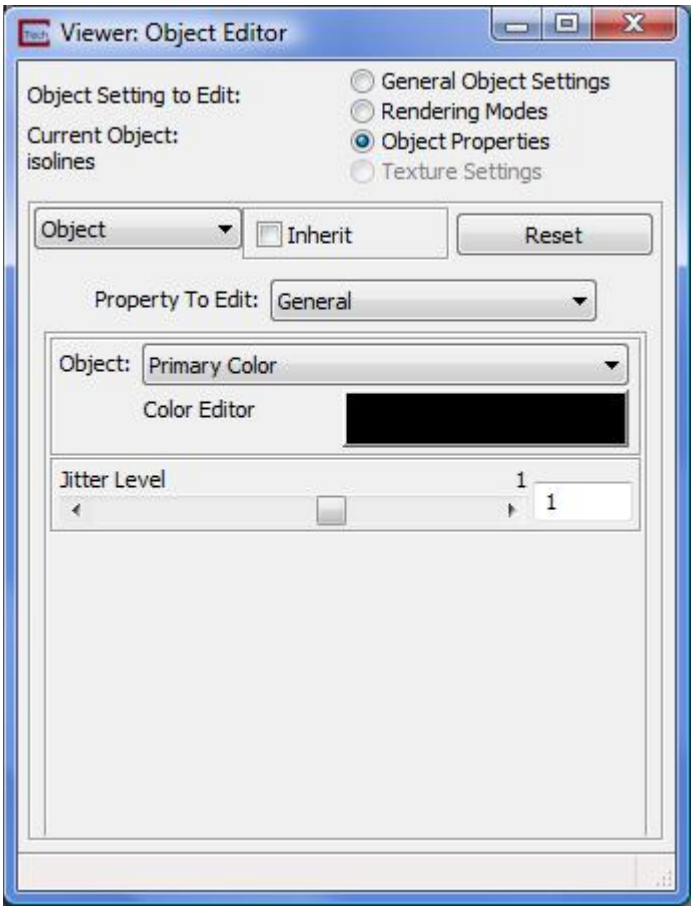
We will now add labels indicating concentration to the isolines we just created. This requires adjustment of the Isolines module interface to match the following:



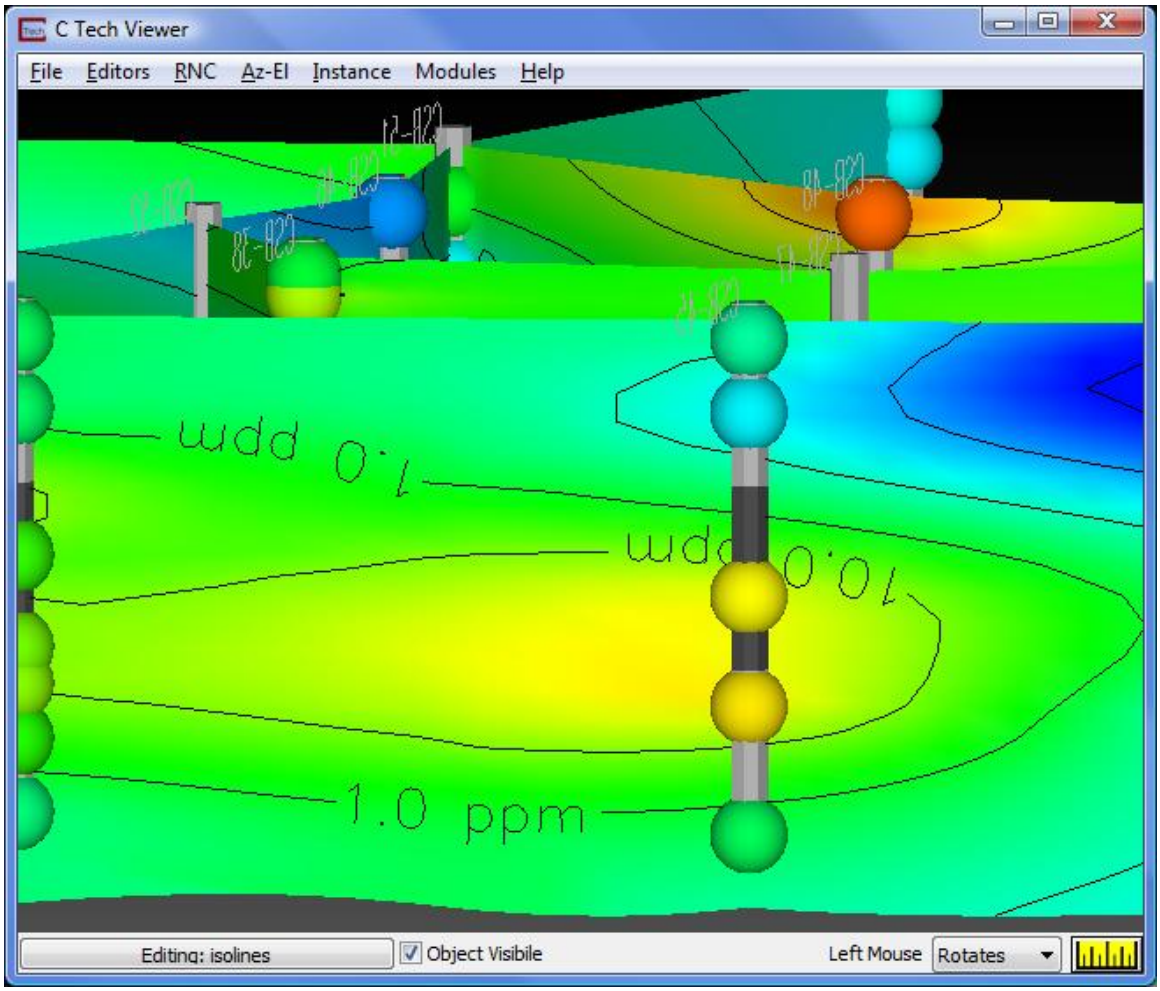
Now adjust the Az-El panel to Scale Factor 5.00, Elevation 5 and click on Azimuth of 75. This should produce the following view:



The white isolines on the bright yellow and green don't show up as well as we'd like. To fix this, we select the isolines object in the Object_Selector (the *Editing: All Objects* button at the bottom left corner of the Viewer) and choose Editors..Object..Advanced Settings to open:



Change the Color Editor from White to Black to see:



Our isolines now have labels and we have adjusted the line/label color (black) and orientation to optimize this view. Let's explore a commonly encountered problem with isolines and then discuss solutions. You'll notice that when viewing from this direction that the some of the labels are backwards and upside down. One "quick fix" to this problem is to adjust the toggles in Flip Labels until you achieve the desired result. In the case of this view, we should turn on the second toggle switch to see the 1 and 10 ppm labels clearly. Unfortunately, rotating to the opposite side of this fence will cause those labels to be backwards.

Before we discuss the more robust approach to using Labeled Isolines, a description of the flip toggles is useful. The Flip_labels toggle will flip all labels upside down. Depending on the viewpoint and the data, this may be useful for making the labels more readable. The toggle before it, '*', flips the isolines (and labels) to the other side of the surface. This is useful for putting isolines on a slice module or for putting labeled lines on both sides (with two isolines modules) of a surface such as a fence diagram.

The more robust solution for the labeling application is to take an alternate approach whereby we instance another isolines module, connect it in parallel and set one module with a positive Surface Offset and the other with a negative offset. In this case we need only set the options once for each of

the two Isolines modules. This approach therefore prevents the need for toggling Flip Labels every time the view is adjusted.

Fence Diagram Conclusion

In this Workbook, we explored the application of EVS to the modeling of three-dimensional chemical plumes or ore bodies and mapping those to complex geologic structures using fence diagrams. Fence diagrams can be combined with any of the other techniques that you have learned in the other workbooks. You are now ready to explore the many saved networks, which are included with EVS. These networks will introduce you to new analysis and visualization concepts. If a module's function is unfamiliar, refer to its' on-line help. Using a rigorous approach you will quickly become familiar with EVS's modules and will doubtless discover new ways to employ them.

Workbook 7: Visualizing Groundwater Modeling Results

Modeling Visualization Introduction

EVS has powerful features for visualizing flow velocities, head gradients, time-series output and mesh geometries from MODFLOW, MT3D and most other commercially available flow and transport programs. The MODFLOW program is used for simulating three-dimensional ground water flow with

specified boundary conditions including a free surface (water table). MT3D uses ground water fluxes calculated by MODFLOW to simulate the three-dimensional transport of a single solute undergoing advection, dispersion and chemical reaction processes.

The output of MODFLOW and MT3D is not immediately suitable for input into EVS and therefore must be post-processed into a format usable by Load_EVS_Field in order to be used in EVS. The EFF format is most suitable in that it can preserve layer information and thus allow coloring by layer, exploding layers and specific layer selection. The file structure is more complex than others because it requires describing the nodal connectivity for each cell in the model. The old field file (.fld) is simpler to produce in that the nodal connectivity (definitions of the cells) does NOT have to be explicitly described, however it will not allow the layer manipulations described above nor will it allow the specification of cell data.

This workbook details approach for visualizing the UCD (.inp) format output from MODFLOW and MT3D, as output by Groundwater Vistas. This workbook also assumes that the user has a thorough understanding of the use of both MODFLOW and MT3D. Also, the user should be familiar with all aspects of EVS covered in previous workbooks. Basic EVS functions the user should be comfortable with include:

* Basic network editor functions such as instancing, connecting, and deleting modules.

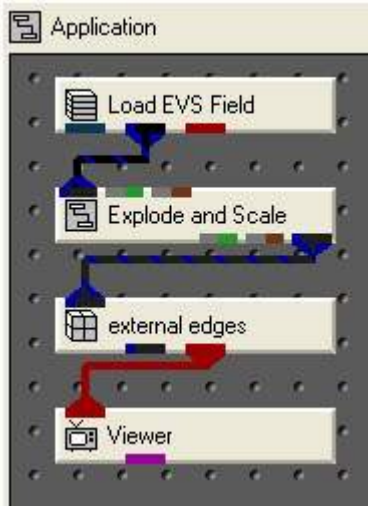
* Use of **Module Control Panels** including how to make them visible using the pull down Modules menu.

* Basic viewer operation such as rotating, translating and scaling the transformable objects in the viewer window.

If the user is not familiar with these functions of EVS, they should refer to the previous workbooks to gain a basic understanding of EVS.

Visualize the Grid

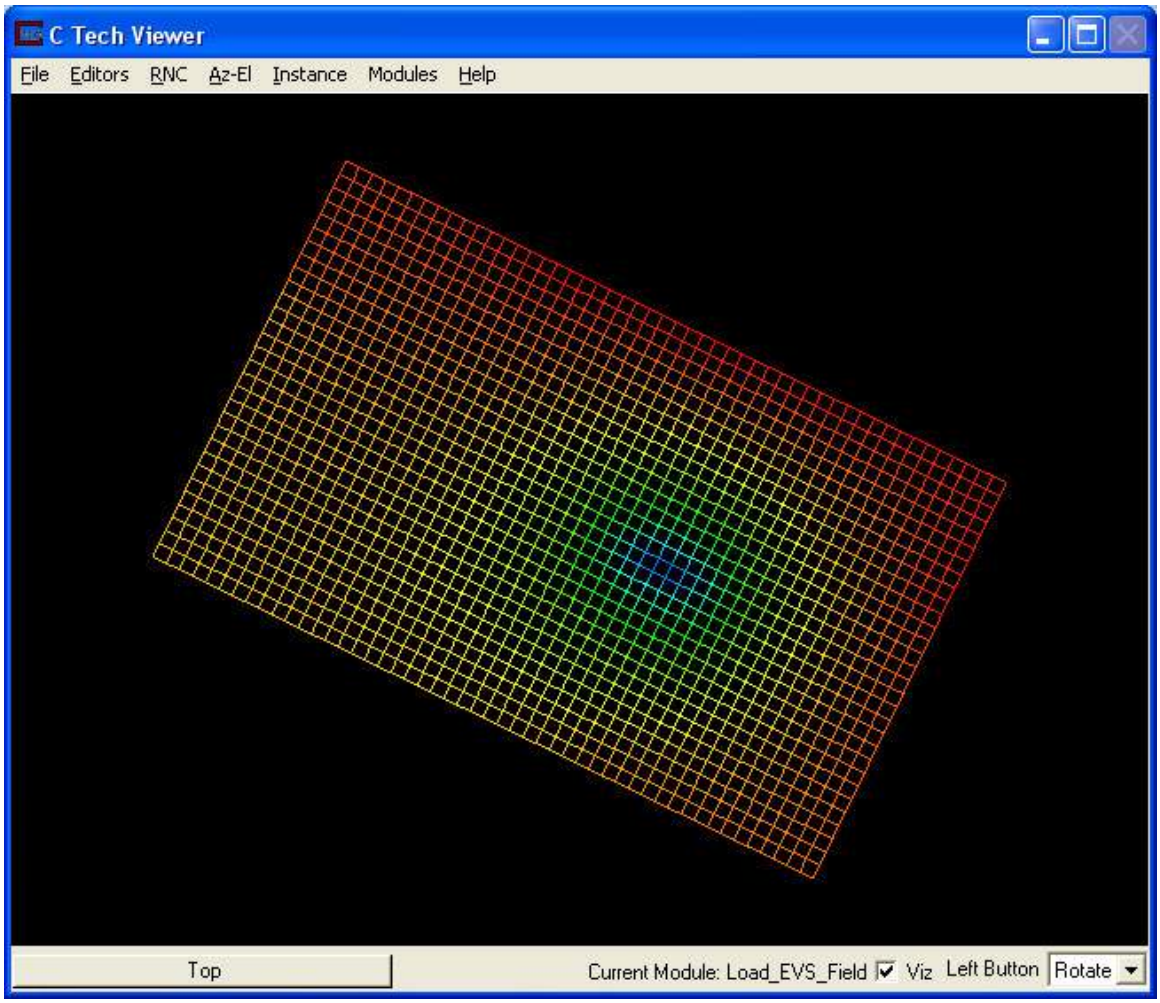
The finite-difference grid used in MODFLOW can be visualized using EVS. First, the user should create a simple application like the one shown below.



In Load_EVS_Field, choose the file *multi_layer_modflow.inp* in the \ctech\data\modeling folder.

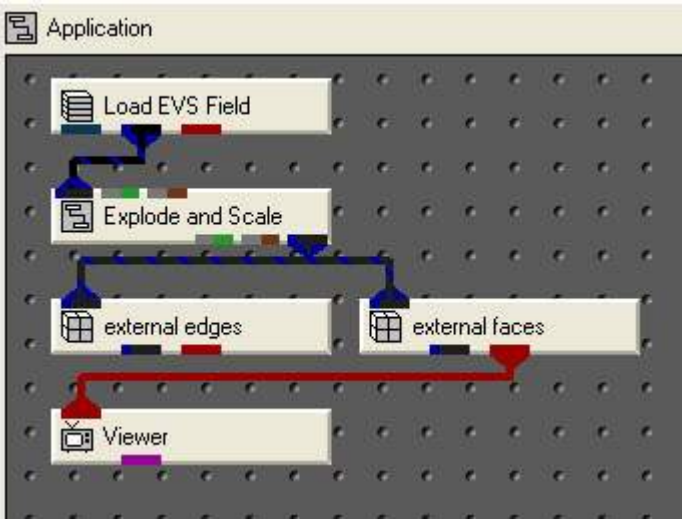
Sets external_edges to have a Max Edge Angle of 0, so that all grid lines are visible.

The viewer should now appear like this:

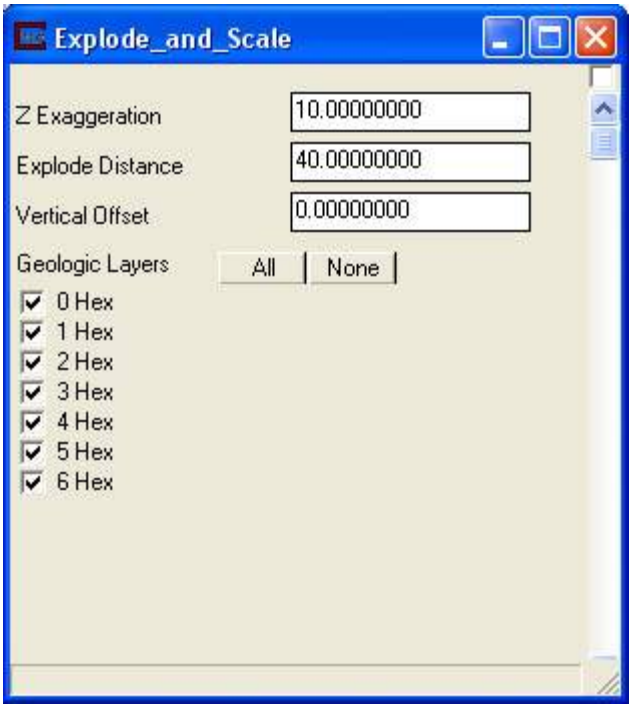


This view shows the outline of every cell in the model. The cells are currently being colored by the value of head. However, this can be changed by selecting a different Nodal Data Component in the external_edges module.

Now, let's look at the grid for this model in a different way. Modify your application to match the one below.



Open the main panel for Explode_and_Scale and adjust its parameters to match:

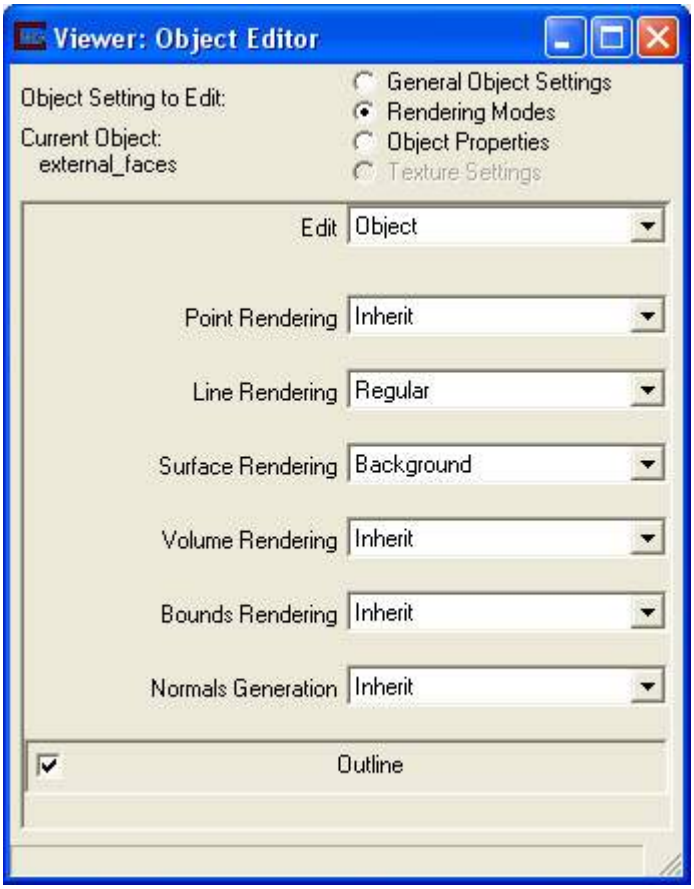


Now we are going to change the modes of object display to create a hidden line view of the grid. Our layers are exaggerated and exploded and now we will color the faces the background color, and display the cell boundaries with colored lines. Since we only have one object in the Viewer (external_edges is disconnected) we could modify the properties of All Objects. We therefore don't necessarily need to use the object selector. However, if we add new modules later, they would also inherit the properties of All Objects, which is normally undesirable behavior. For this reason, we will explicitly modify the external_faces module.

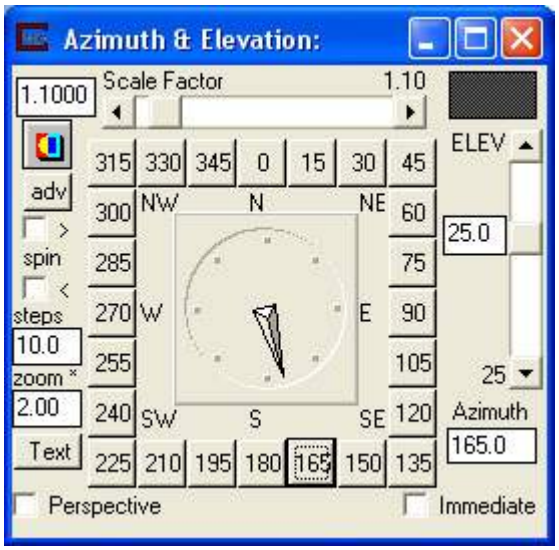
In the Viewer, push the button marked "Top" in the lower left corner. This will open the Object Selector window, which should list "Top" and

"external_faces". Highlight external_faces and press *OK*. The button in the lower left corner of the Viewer should now say "external_faces".

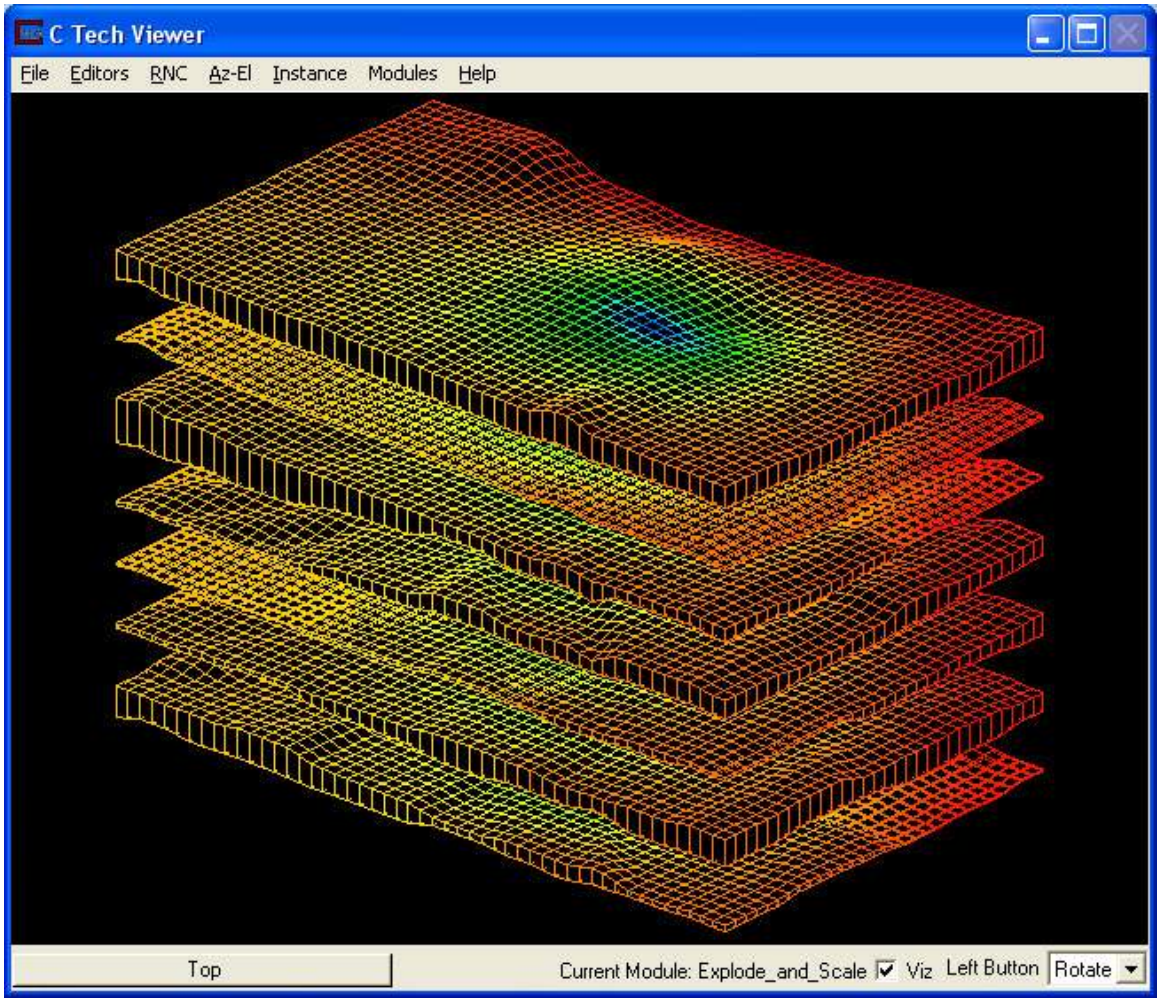
Go to Viewer, select the Editors->Object->Advanced Settings pull-down menu. Select the Rendering Modes radio button in the Viewer: Object Editor window and set the Line Rendering to Regular and Surface Rendering to Background. Your panel should look like this



Finally let's choose to view this object differently. Open Az-El in the Viewer, and set the view to: Scale = 1.10, Elevation = 25 and Azimuth = 165, then press the multi-colored (Reset/Normalize/Center) button in the Azimuth & Elevation window in the Viewer.



Your view should now be:



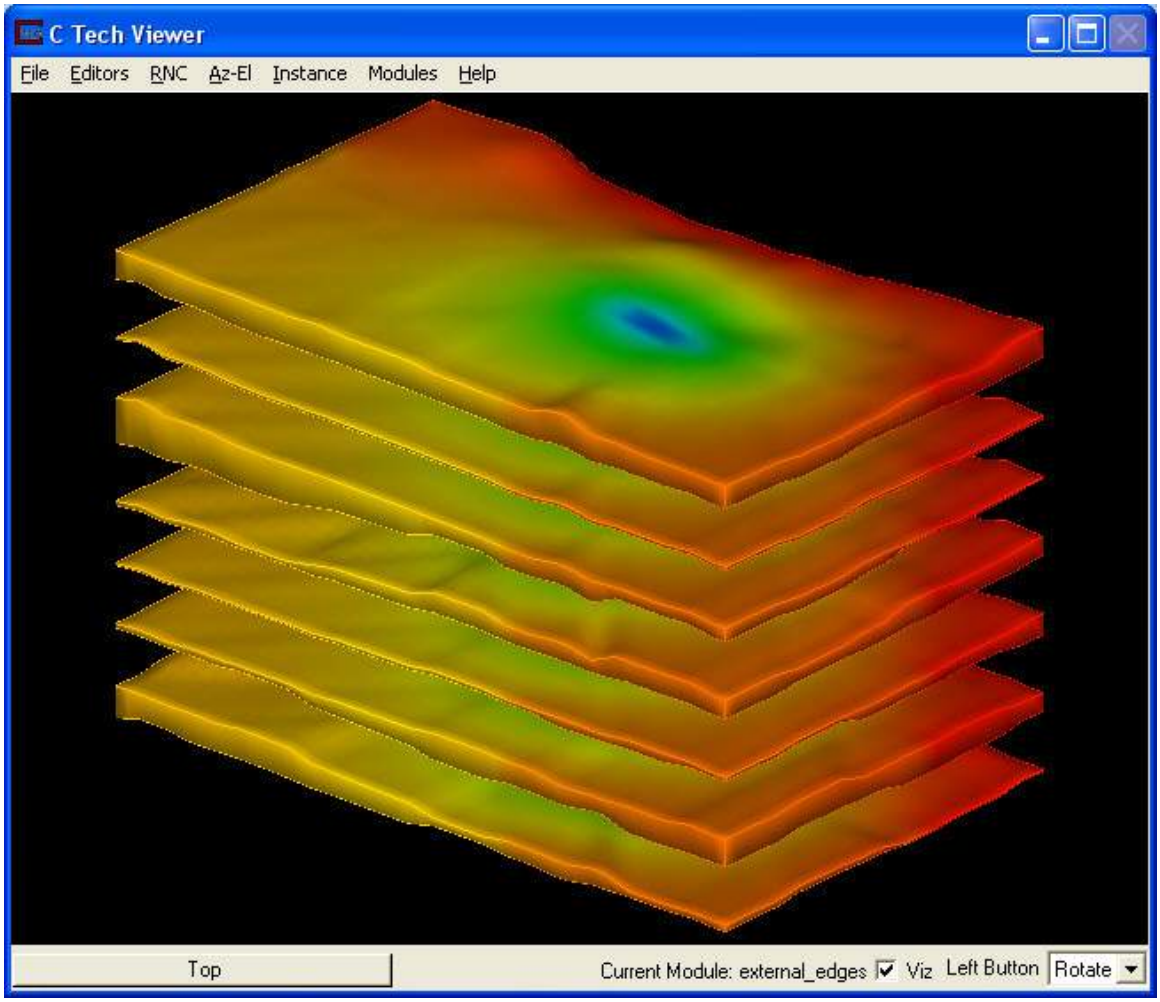
Note that we have 7 geologic layers. Each layer has only one layer of elements. This is the way the layers were modeled in MODFLOW and MT3D. Each layer has a top and bottom, but there aren't any additional (nodes)

data in between. Subdividing these layers isn't necessary because EVS will interpolate for us. The outlines of the elements are colored according to Head, because this is the first data component. We could color by any (scalar) component by selecting it in external_faces.

Let's proceed to the next section and examine a few different ways to visualize head and drawdown.

Visualizing Head and Drawdown

First, return the Modes settings for external_faces to their original values, reconnect external_edges to the Viewer and set external_edges to have a Max Edge Angle of 60. If you hit the Reset/Normalize/Center (multi-colored) button in Az-El, the display of the surfaces colored by head should now be:



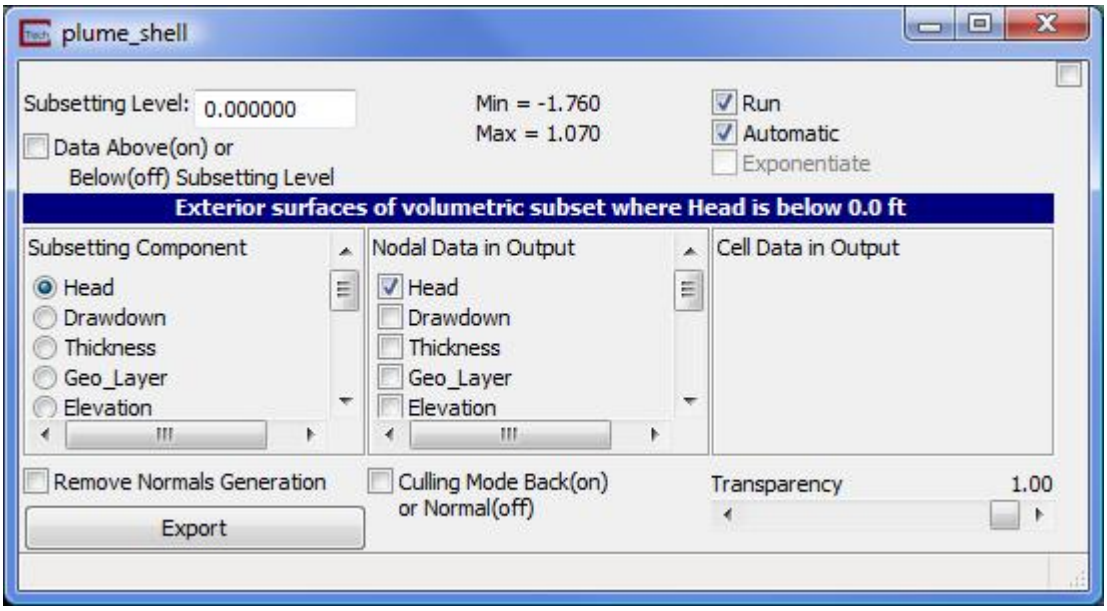
Note the center region with low heads. The default color map results in low head values colored blue and high head values colored red.

Visualize Heads

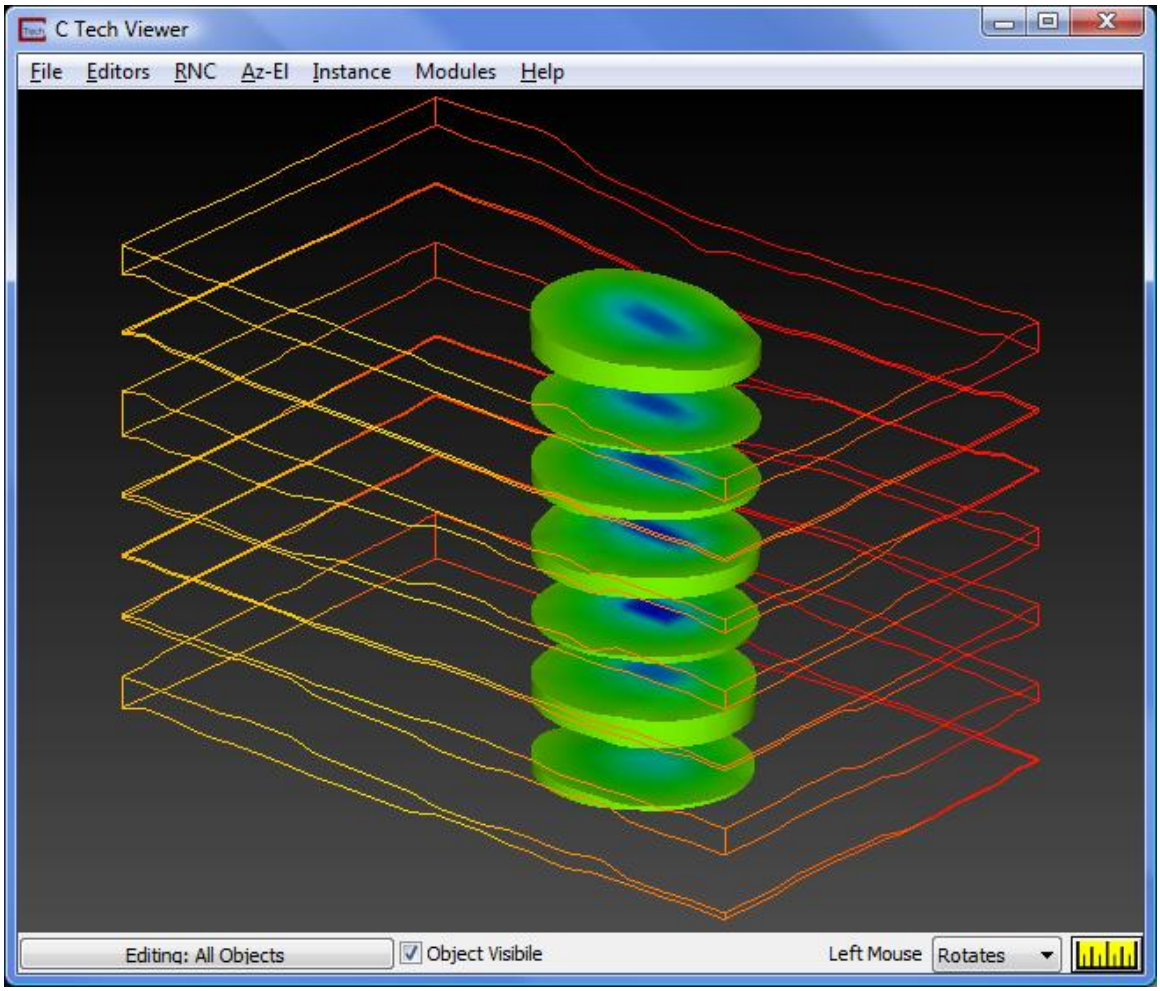
To visualize a volumetric subset of Model calculated heads:

1. Modify your application by replacing external_faces with plume_shell.

2. Open plume_shell's control panel and set the subsetting level to 0.0 and turn off the Above toggle.



The viewer display should now look similar to this:



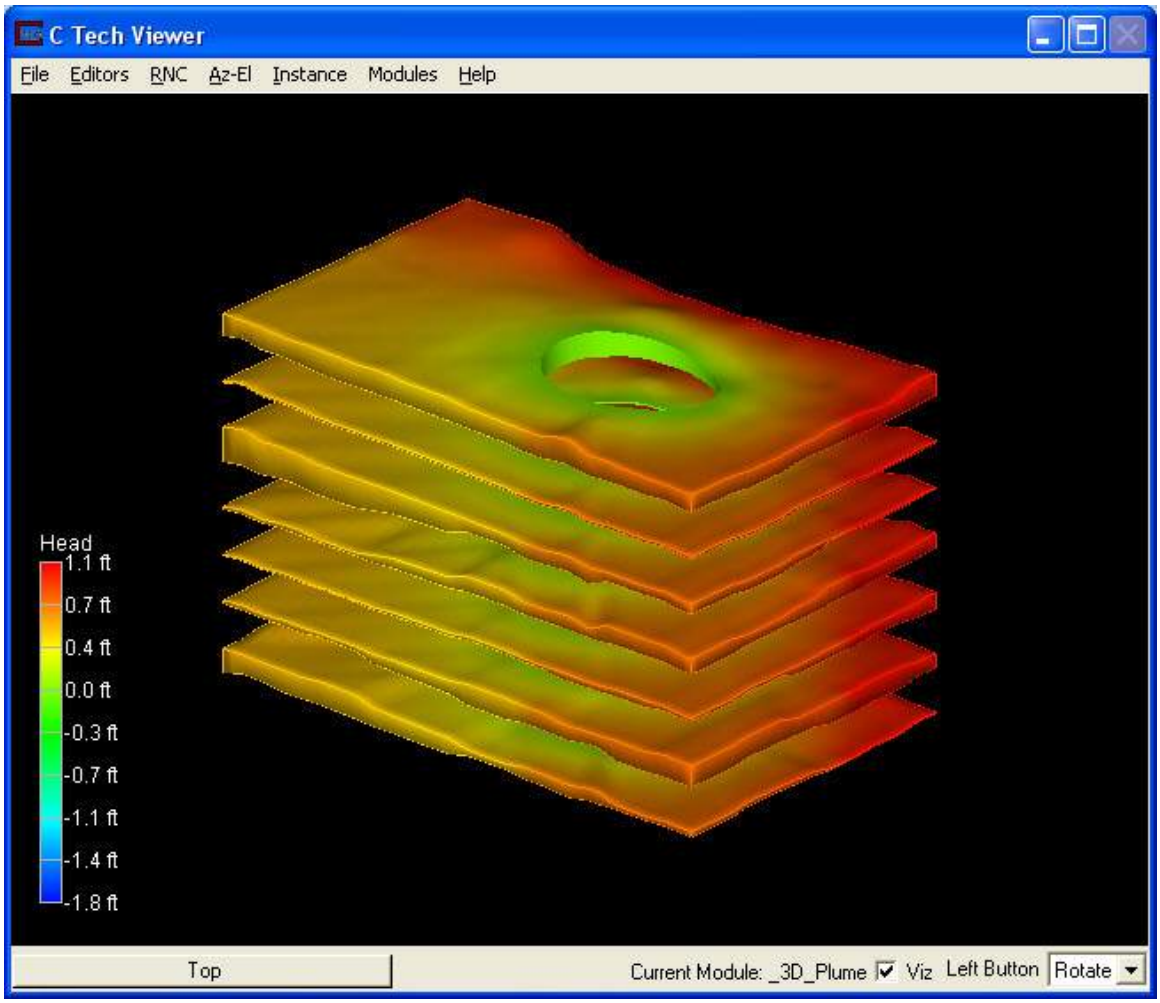
This display shows a volumetric subset of all regions in the model where head is below 0 feet. These areas are clearly affected by one or more wells.

Turn on the Above toggle so the display shows all regions in the model where head is above 0 feet. and let's also add a color legend. To do this:

1. Instance the Legend module

2. Connect the output port (red) on plume_shell to the (red) input port on Legend and connect the output port (red) on Legend to the Viewer's (red) input.

You'll notice the legend appears in the lower left corner of your screen. The units (ft) and data component (Head) are automatically pulled from the output of plume_shell and populated in the Viewer.



Visualize Drawdowns

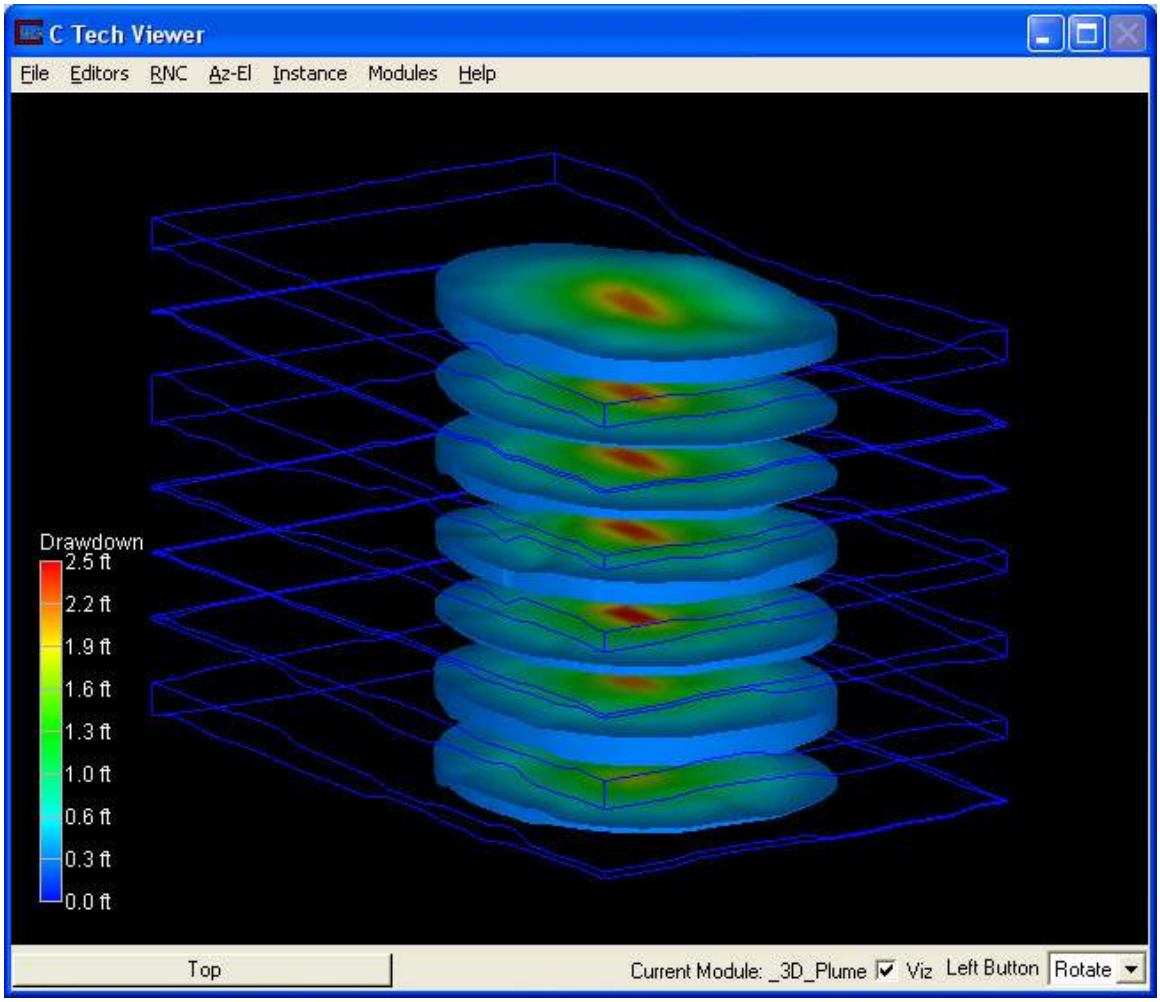
We will now take a look at the calculated drawdowns. Drawdown in MODFLOW actually means change in head since the start of simulation. Models incorporating area or point sources can actually have negative values of drawdown (Drawdown is positive for declining water levels).

To visualize drawdown:

1. Open the external_edges control panel and set the data component to drawdown.

2. Open the plume_shell control panel and set the *subsetting level* to 0.3, the *SubsettingComponent* to Drawdown, and make sure that "Drawdown" is the only selected data under *Nodal Data in Output*. (You will need to uncheck "Head" and check on "Drawdown" in *Map Components*.)

3. Set your view (Az-El) parameters to Scale = 1.10, Elevation = 15, and Azimuth = 150. The display in the viewer now shows all drawdowns greater than 0.3 feet. For this model the drawdown distributions are similar in all layers, but note the brightest red (high drawdown) location is in the center layers. This center region is the location of wells in this model.



Add a CAD File

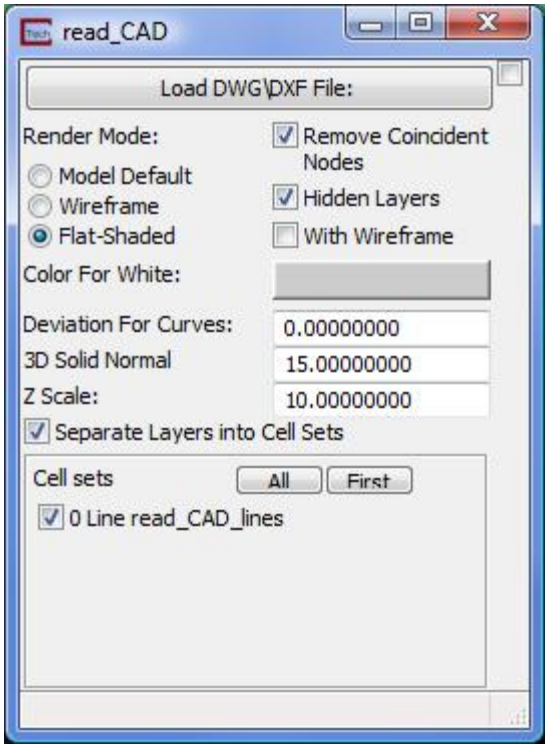
We will now add a .DXF file overlay on the model. Adding a .DXF file provides familiar spatial references to the model results.

To add a CAD file to the model:

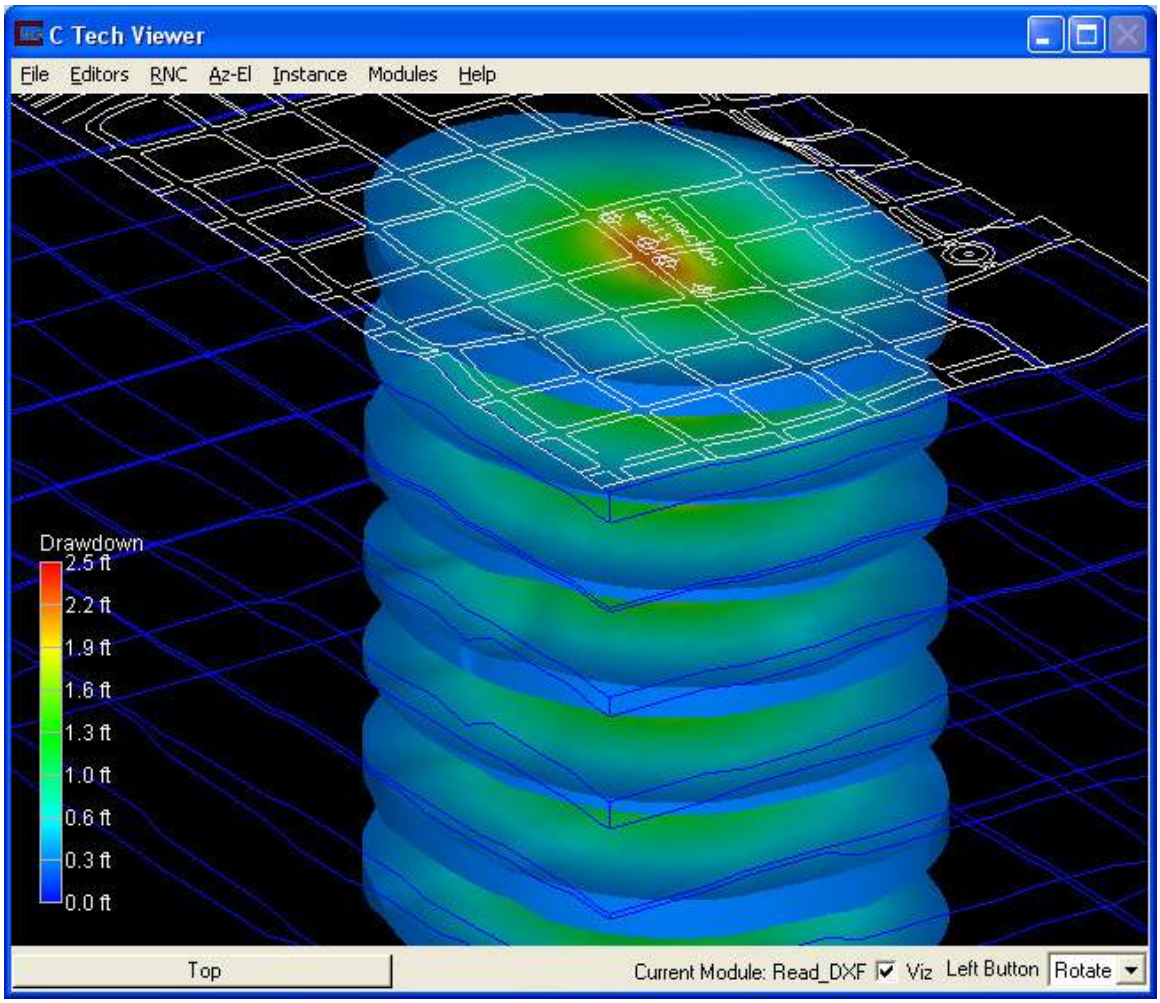
1. Instance the read_CAD module. This module is found in the File module library.

2. Connect up its output port to the input port on the viewer and connect the grey-brown (z-scale) port from Explode_and_Scale..

3. Open the read_CAD control panel and select the Load DWG/DXF File button. Change to the \ctech\data\modeling\ directory and choose the file multi_layer_modflow_site_map.dxf. This file was created using the surf_map module to drape the constant elevation lines onto the top geologic surface. When it was created, the surface was exaggerated by a scale factor of 1. The grey-brown port will cause the z coordinates to be properly scale to lie on the topmost surface. The draped DXF file is also much larger than the original file because each simple straight line is broken into a complex polyline which must follow the contours of the underlying surface.



Set your view (Az-El) parameters to Scale = 1.60, Elevation = 25, and Azimuth = 150. The image in the viewer now shows an volumetric subset of the drawdown greater than .3 foot and an overlay of the site basemap.

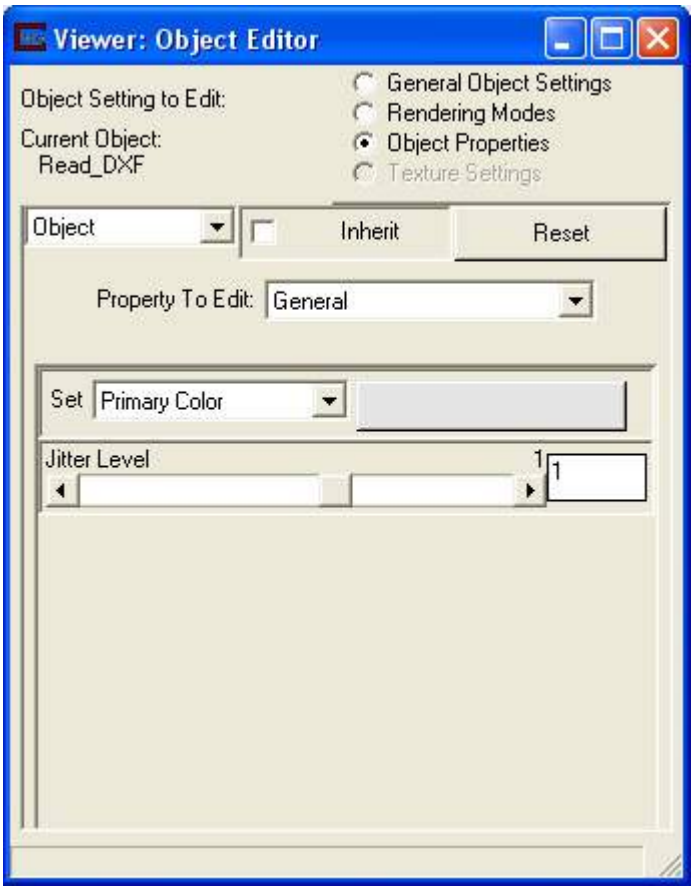


There is just one thing a bit wrong here. Note that the lines of the DXF may seem a bit broken or erratic in the area of the drawdown (plume). This results from the fact that our DXF lines and the plume surface have the same z-coordinates. These objects are coincident in space. When this happens (or nearly happens) the rendering of objects can become ambiguous. The Viewer doesn't know which object is closer to our eye and should be drawn and which should be masked by the other.

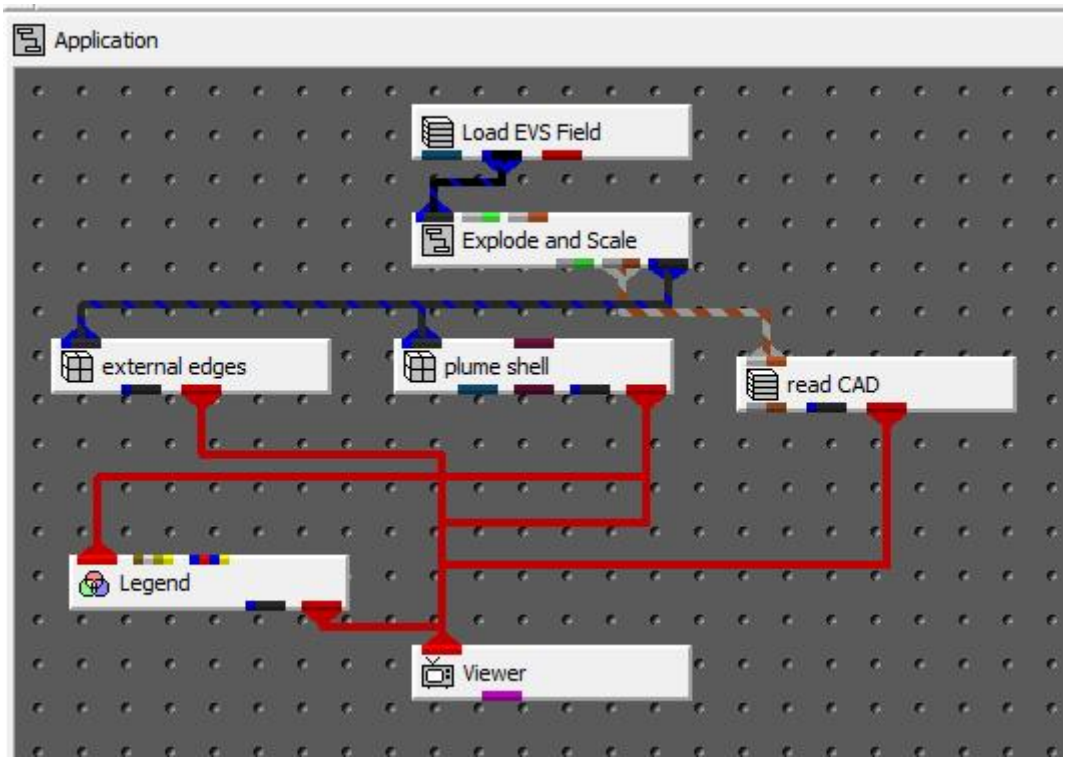
C Tech's viewer includes an object setting that can address this issue. It is called jitter. If two objects are coincident but one has a higher jitter, it is preferentially drawn. We will set the jitter of the DXF lines higher.

To do this, select the read_CAD object. The quickest way is to use the Alt+Left mouse button and click directly on some of the lines. If you do it right, the button in the lower left corner of the Viewer should change to say "read_CAD." You can also use the Object_Selector, just as we did earlier.

Now, from the Viewer, choose *Editors->Object->Advanced Settings* Select *Object Properties* at the top of the Object Editor window, and then select **General** in *PropertytoEdit*. At the bottom of this window you will see a Jitter Level slider. Set it to 1 (it should have been zero). As soon as you do this, the lines should become clearer. If you experiment with jitter you'll see that setting it to -1 will make the lines completely disappear where they overlap the plume. We'll leave it at 1.



The current application should match:



Visualize Contaminant Plume

Visualization of the MT3D simulated contaminant plume can be done using the application built in the previous section of this workbook. The MT3D simulation has already been run and the results have been appended to the UCD (multi_layer_modflow.inp) currently being used. The first seven (all scalar) components were written by a MODFLOW simulation. Concentration, Change in Conc(entration) and Velocity (a 3 component vector) were created by MT3D and were appended to the same UCD. To visualize the MT3D results we merely select any of those data components appended by MT3D.

To visualize the concentration:

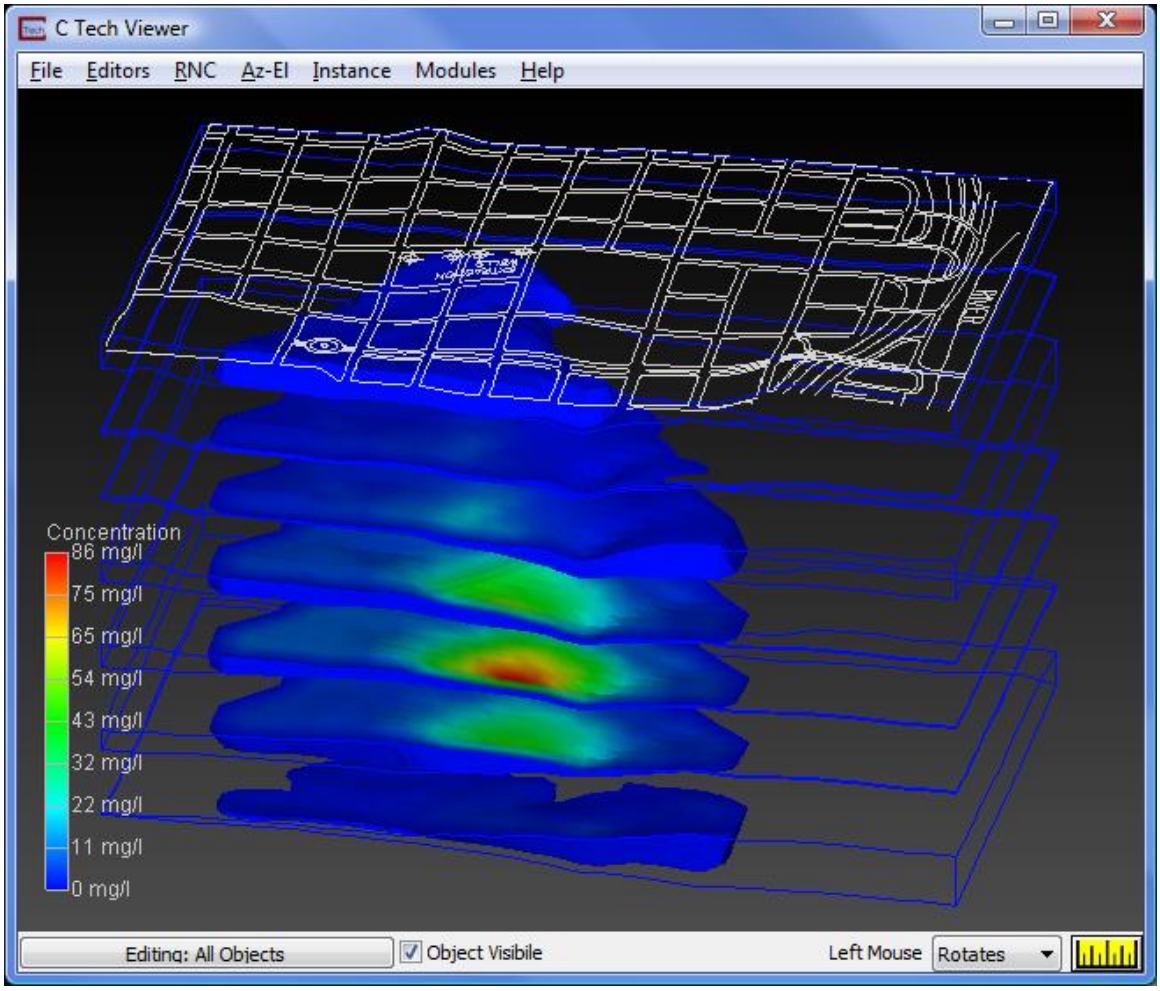
1. Open external_edge's panel and select Concentration as the *Nodal DataComponent*.

2. Open the plume_shell control panel. Change the *SubsettingComponent* to Concentration. Then uncheck "Drawdown" and check on "Concentration" in the *Nodal Data to Output*. This will cause us to color our output by Concentration (the first mapped component). Finally, set the *subsetting level* to 0.1 ppm. Please note that MT3D does not output concentrations on a log10 scale. All values are actual linear values. They could be converted to log10 values using the data_math module if desired.

When you hit enter or move the cursor (and click it) it notifies this window that the value has changed. If you know that you don't want to change it immediately it is always faster to click in another window, because if you hit enter, plume_shell recalculates... and it will recalculate again when you move your cursor, click the mouse or close this window.

The view in the viewer now displays a volumetric subset of concentrations greater than 0.1 mg/l. Notice that the legend automatically updated to show the correct data component and units.

Set your view (Az-El) parameters to Scale = 1.2, Elevation = 25, and Azimuth = 15. The image in the viewer now shows a volumetric subset of the Concentration greater than .1 M/L^3 and an overlay of the site basemap.



This plume_shell of concentration displays all concentrations greater than 0.1 mg/l. Because the MT3D example problem incorporates dispersion, the higher concentrations will occur at the center of the isovolume. In order to see inside the contaminant plume we will next add a cut plane vertically through the plume.

Add A Cut Plane

Adding a cut plane to this application allows us to visualize the concentrations inside of the simulated plume. Before we add a cut plane though, let's first slice. You might ask what is the difference between a cut and a slice. In our terminology, cutting through a volume yields a subset of the volume (but still a volume). Slicing a volume yields a planar object. In other words, cutting preserves dimensionality, slicing reduces it by one.

1. Start by deleting plume_shell.

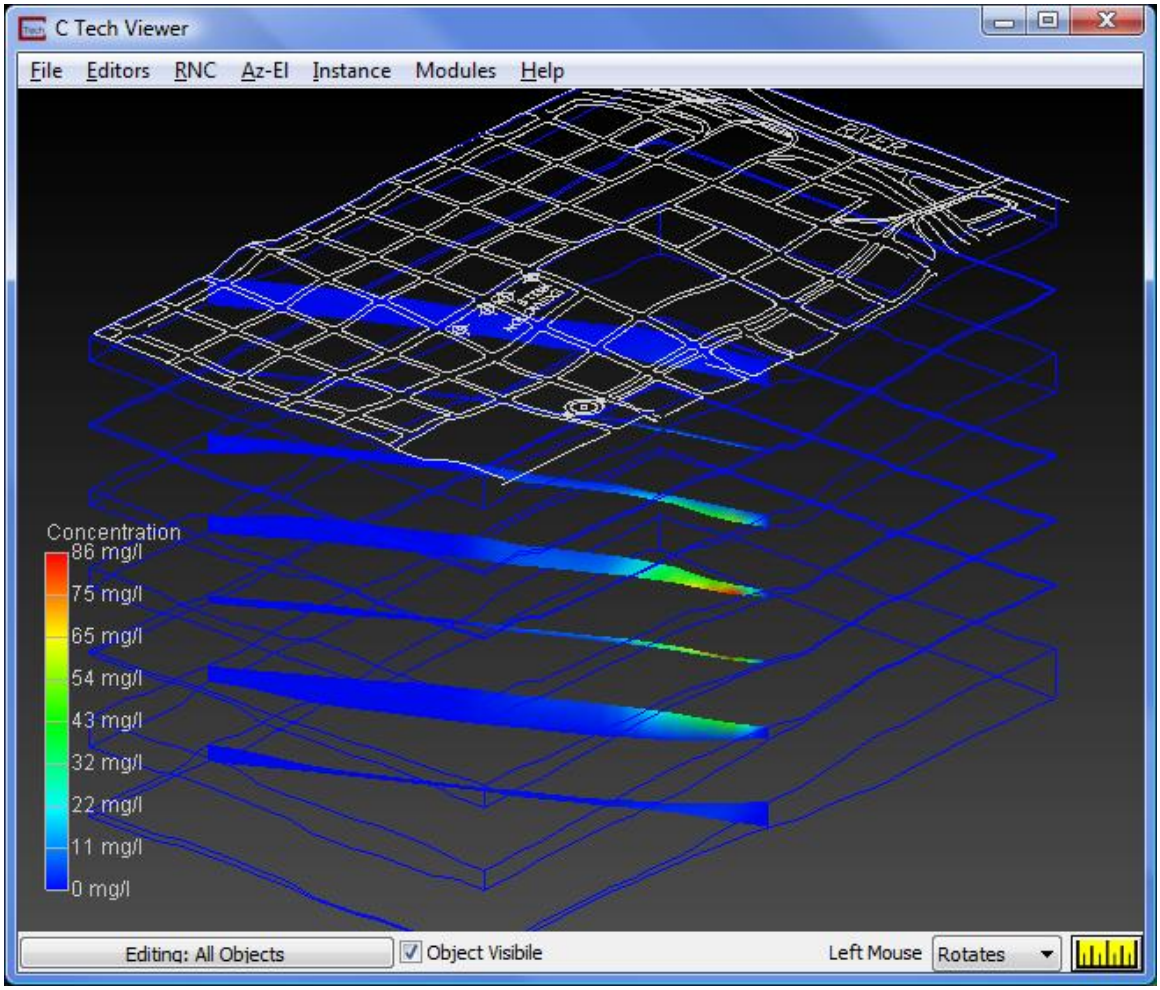
2. Instance the slice_easting module from the Subsetting library and position it between Explode_and_Scale and the Viewer.

3. Connect the blue output port on Explode_and_Scale to the input port on slice_easting.

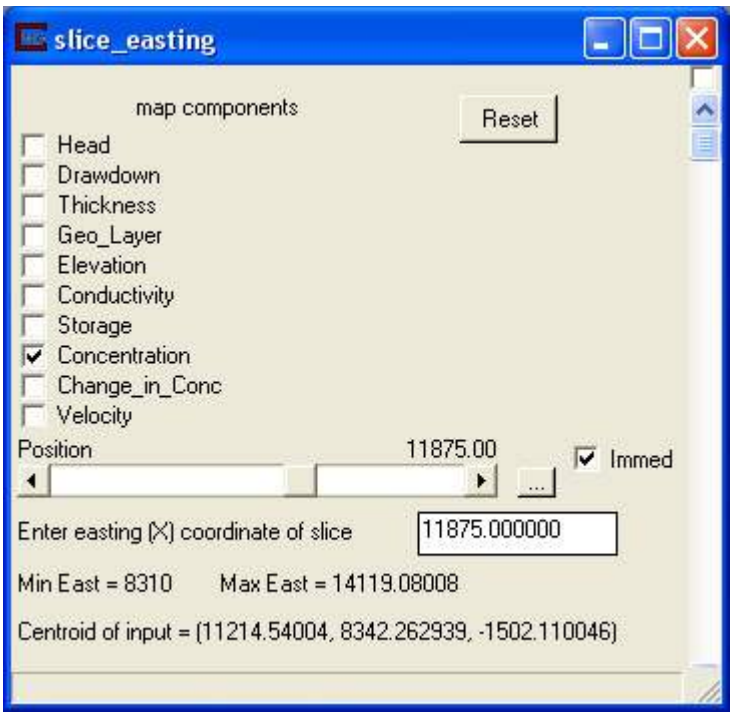
4. Connect the rightmost (red) output port on slice_easting to the Viewer and to the Legend module.

5. Open slice_easting's control panel and uncheck "Head" and check "Concentration" in the *Map Components* section.

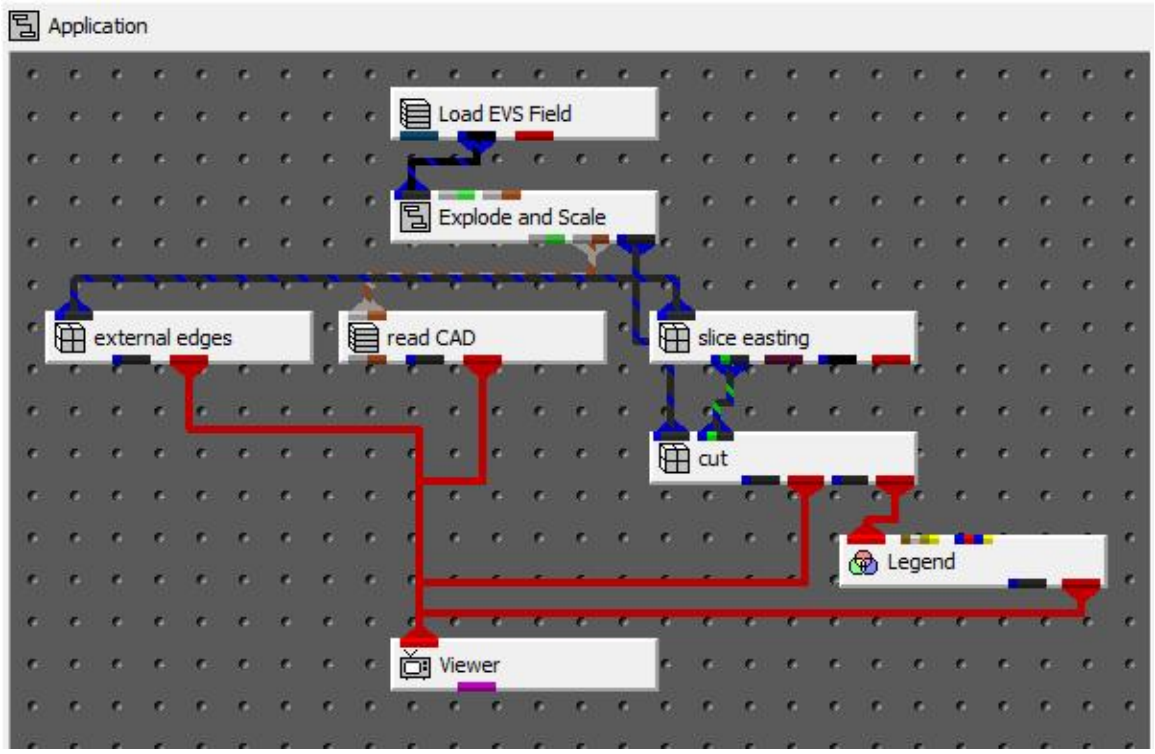
Set your view (Az-El) parameters to Scale = 1.2, Elevation = 25, and Azimuth = 75. The image in the viewer now shows a vertical slice through our model:



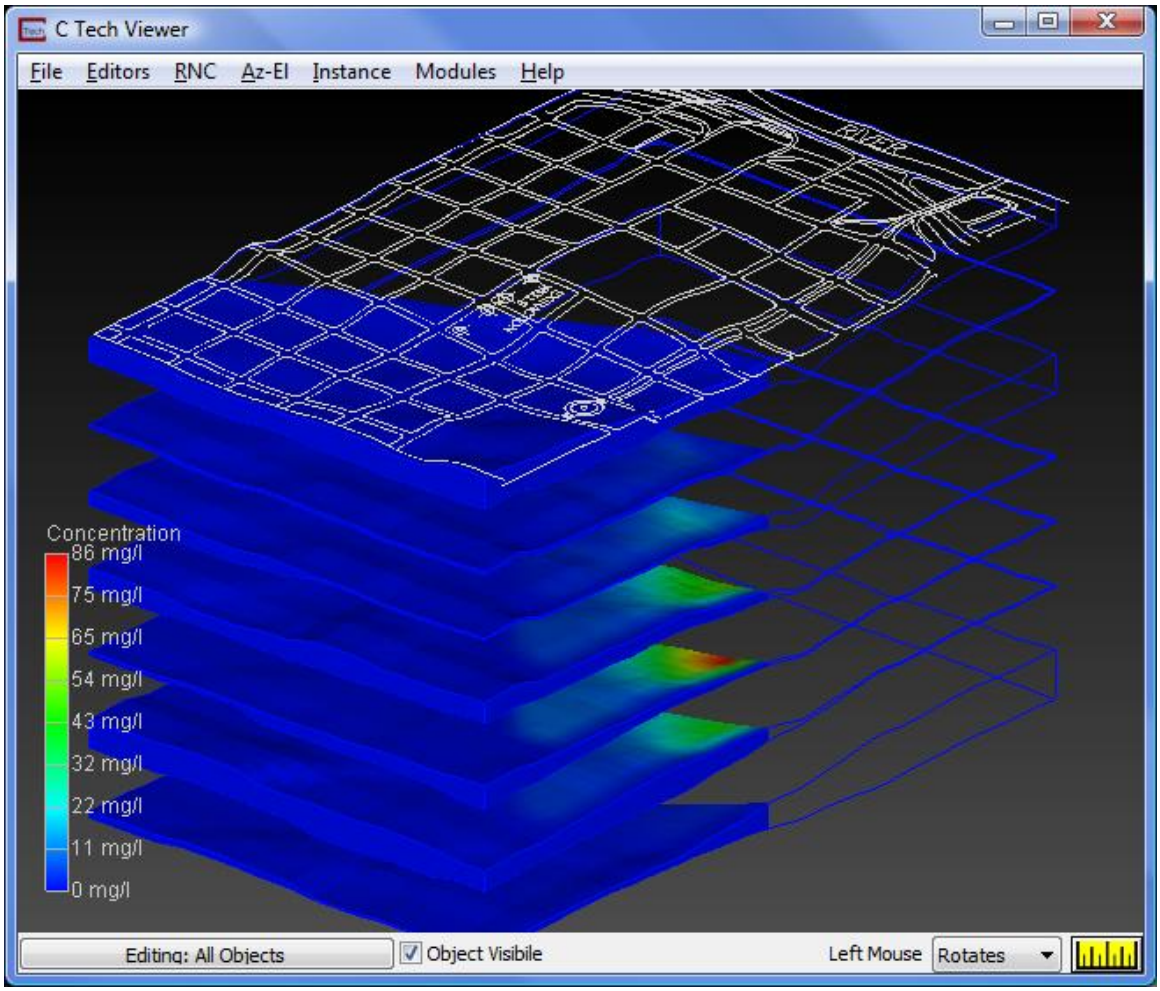
Now open the slice_easting's control panel and turn on the Immed(iate) toggle and start moving the slice plane by changing the *Position* slider. With a little playing you'll find the center of our plume is at about 11875. Set your slice to be at an x coordinate (easting) of 11875 as shown below.



EVS (MVS) has a simple way to "cut" through a model where we have a slice plane. Instance a cut module and connect it as shown below. Open the cut module's control panel and select the External Cut Plane toggle. Also, uncheck "Head" and check "Concentration" under the *Map Components* in cut.

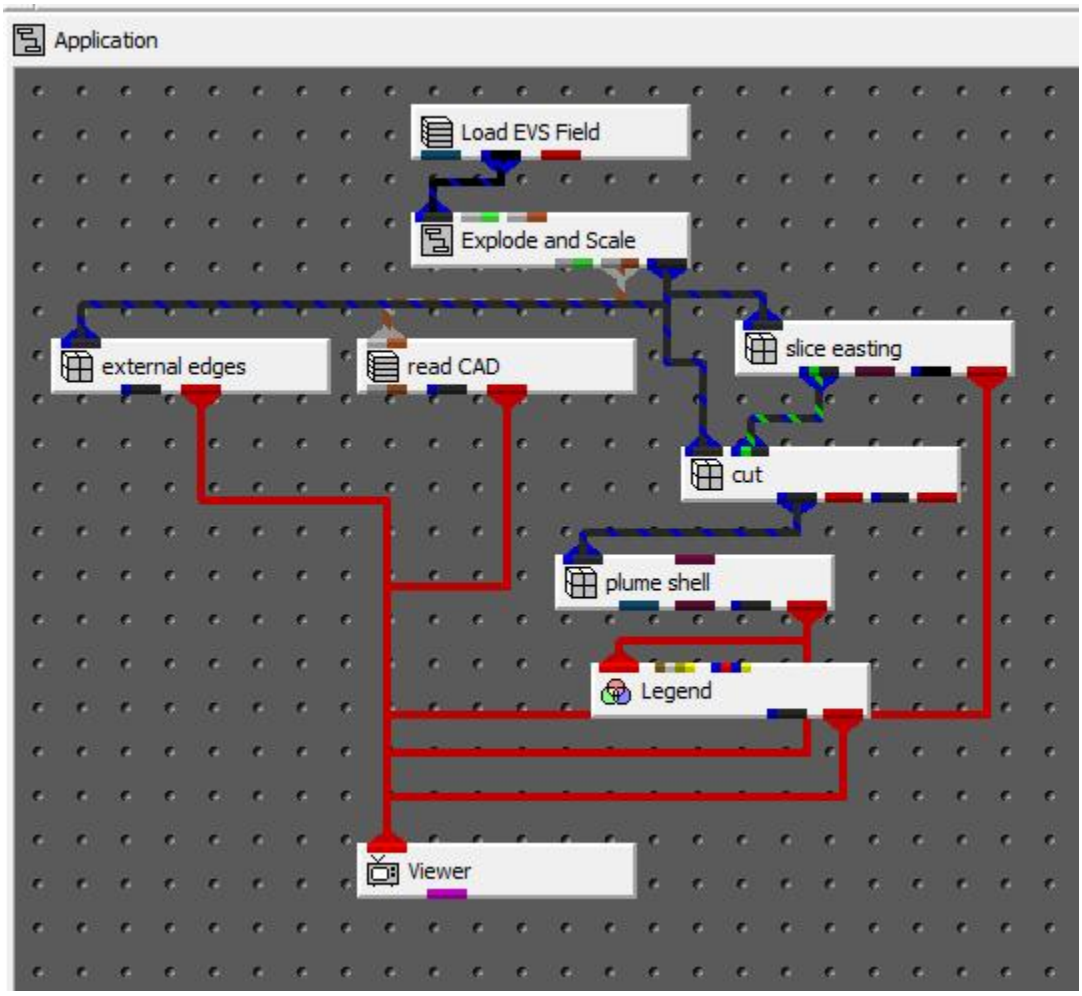


The image in the viewer now shows a vertical cut through our model (not just the plume.yet):



If we turn off the "Above" toggle, we'll see the other "half" of the model. Do that now.

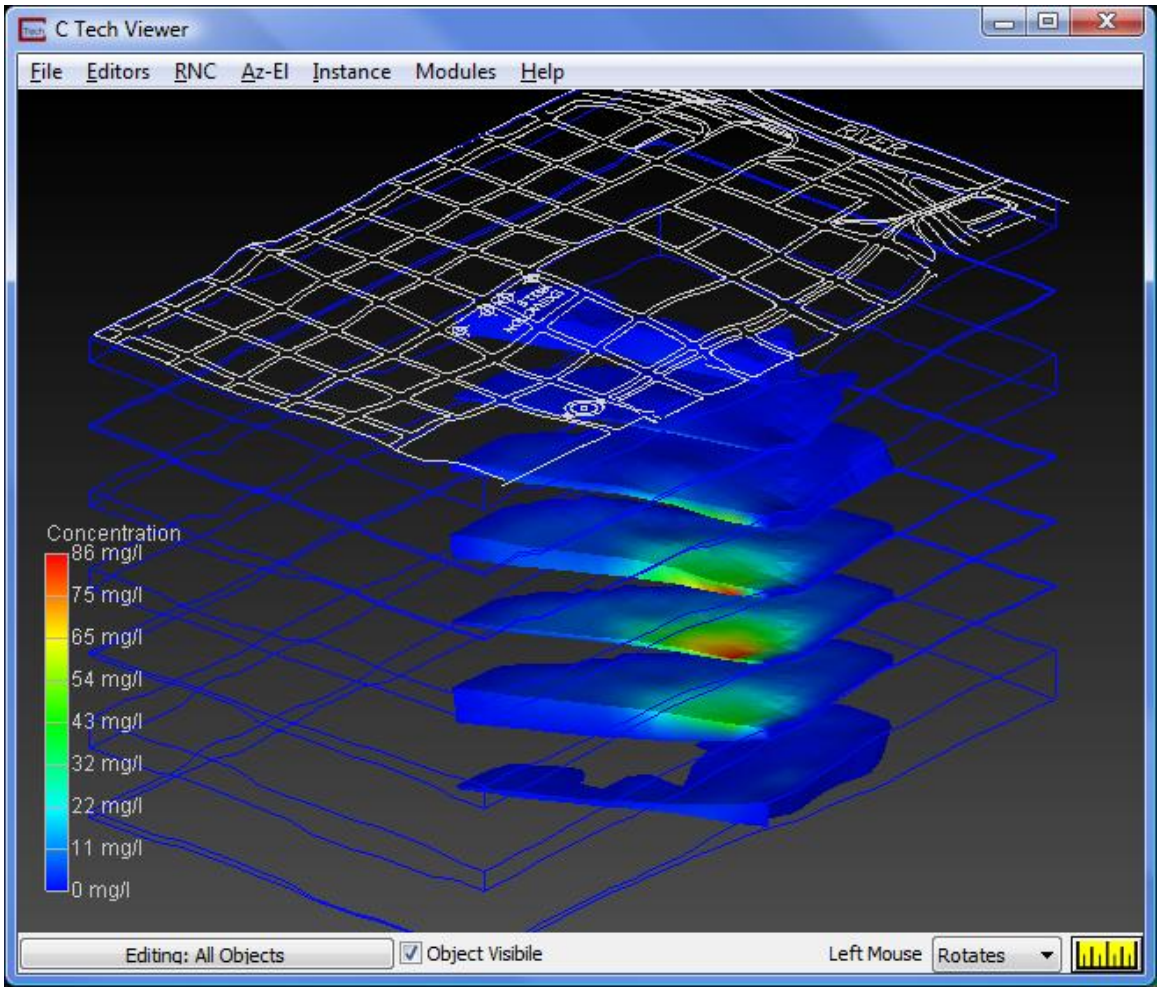
We now want to see the plume (all regions above 0.1 mg/l) after cutting. To do this disconnect cut from the viewer and instance a plume_shell module and connect it as shown below.



Set your view (Az-El) parameters to Scale = 1.2, Elevation = 25, and Azimuth = 60.

Also change the Subsetting Level in plume_shell to .1 and turn on the Remove Normals Generation toggle.

The image in the viewer now shows a cut-away of the plume:



The application has been saved as ctech\applications\Workbooks**modflow_mt3d.v**.

Importing and Exporting Data from Groundwater Vistas

Groundwater Vistas (GV) works with EVS/MVS in two ways. First, GV can import EVS/MVS geology files (*.gvg) to initiate the model grid and layer elevations. Second, GV can export both field and UCD format files for visualization or animation in EVS/MVS. The following sections describe how to accomplish these tasks in Groundwater Vistas.

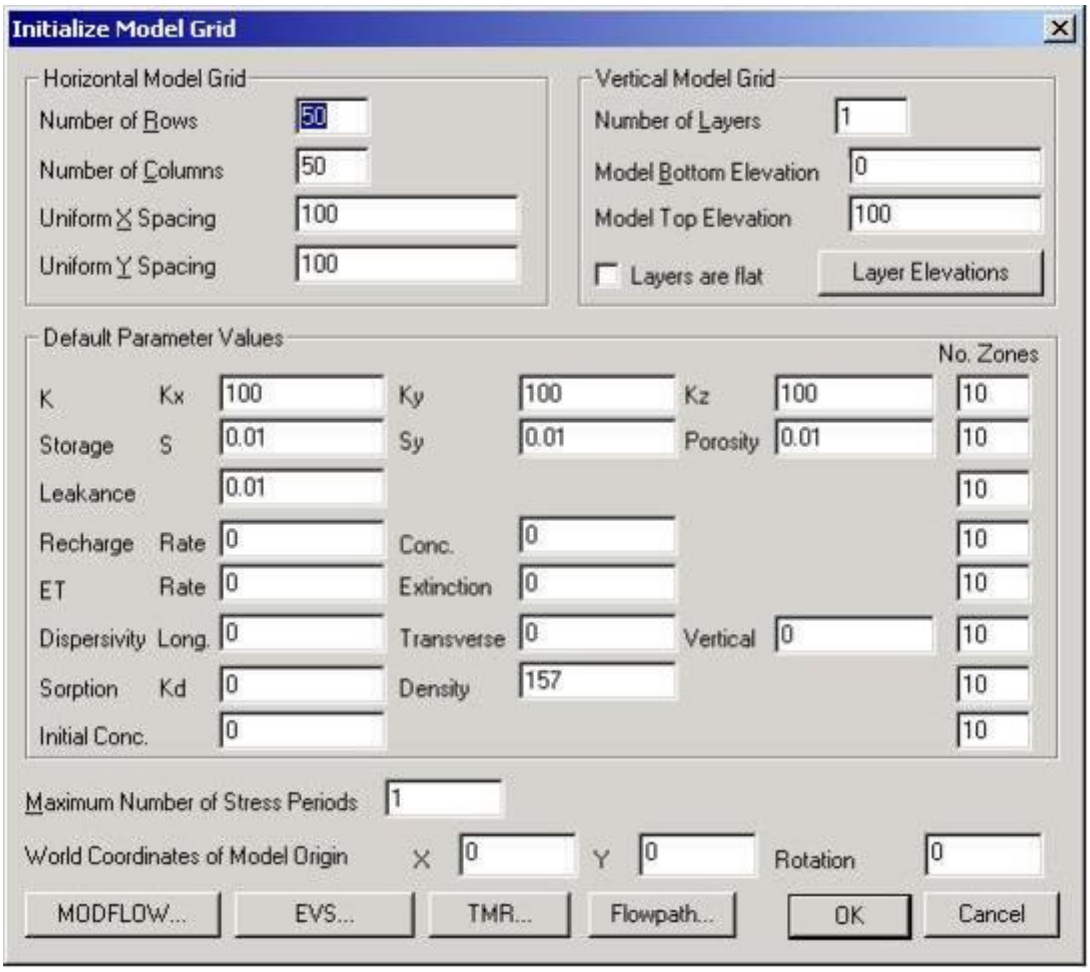
When exporting data to EVS/MVS as either a UCD or Field file, Groundwater Vistas interpolates/extrapolates the block-centered model results to the nodes representing the corners of the model grid cells. This happens in all three dimensions so you end up with an extra layer of data, representing the top of layer one (1). This is done in order to convert data in GV that represents cells to the default **nodal data paradigm** in EVS/MVS. The advantage of this technique is that the grid will look the same as your grid in GV. The disadvantage is that there will be some loss of information, especially for point sources in transport models.

However, for UCD files only, the GV cell data is **also** exported as cell data. In this case, there is no interpolation or extrapolation performed on the model.

The advantage is that your model output results exactly match the model in GV. The disadvantage in EVS/MVS is that cell data is somewhat more limited than nodal data and modules that subset cell data (e.g. for display of a contaminant plume) remove whole cells resulting in a blocky "Lego" like plume.

Importing GVG Files

The EVS/MVS gvg file contains information about the finite-difference grid (rows, columns, layers, and spacings) and the elevation of each cell in each layer. A feature has been added in Groundwater Vistas Version 3 to read the gvg file at the start of a new model. The gvg file then determines the size of the model and location of the model in real-world coordinates. To create a groundwater model from a gvg file, select **File / New** in Groundwater Vistas. The following dialog is displayed:



You will see an **EVS / MVS** button at the bottom of the dialog. Simply click that button and go browse to find the *.gvg file created by EVS/MVS. Click OK when you are done. GV will get the number of rows, columns, and layers from the gvg file. In addition, row and column spacings and layer elevations will be obtained from the gvg file.

Exporting UCD & Field Files from GV

UCD files can be created by GV after the results of a model run are imported. This can happen directly after the model is finished or by selecting **Plot / Import Results** from the Groundwater Vistas menu. Whatever variables you import from MODFLOW/MT3D for display in GV are then exported to the UCD or Field file. When importing cell-by-cell flow terms from the MODFLOW runs, GV exports velocity components to the UCD file.

To export the UCD file, select **File / Export** in GV. A standard Windows file save dialog is displayed. Pull down the list of file types and scroll to the bottom. Choose "EVS UCD (*.inp)" as the file type and click the **Save** button.

EVS/MVS Field files are exported in the same way. Just change the file type to "EVS Field (*.fld)".

Modeling Visualization Conclusion

In this workbook we've discovered a few of the many ways that EVS can be used to visualize results from MODFLOW, MT3D or other finite element or finite difference simulations. We've investigated two fundamentally different methods of visualizing vector data such as flow velocity and have learned how to use several new modules.

Virtually all of EVS's modules can be applied to analysis or visualization of simulation results. The applications used in this lesson should only be viewed as examples to guide your efforts. EVS's flexibility is the key to being able to produce virtually any display that your job requires.

Also, although this workbook focused used data exported by Groundwater Vistas, the results from virtually any modeling package can be converted into a file format usable by Load_EVS_Field.

Users of GMS and Visual Modflow can use the conversion tools provided by C Tech to automatically convert their modeling projects into usable files. See the help for the GMS Converter and Visual Modflow Converter utilities for more information on using these programs.

Workbook 8: Animation

In this workbook, we use the Animator Module to create animations.

Animation Creation

Animator Module

C Tech's software has the ability to create animations (e.g. AVI, MPEG or HAV files), 4DIM vector animations and/or sequences of bitmap images to produce animations. The Animator module makes this process very simple, and provides powerful capabilities to simultaneously modify the parameters of many modules in an application for each frame of the animation. The Animator directly controls all parameters in your application.

Creating Animations

Computer Bitmap Animations Sequences of image are converted to one of several standard computer animation file formats. These formats include .AVI and .MPEG. Our experience is that .MPEG usually plays back slower than .AVI without special hardware. Without special hardware, high quality .AVI animation sequences can be played back on your Windows computer system using the Media Player included with Windows.

We also recommend a freeware program called Imagen for playing back animations. Gromada's Imagen is a superior multimedia player. It has

several advantages over Microsoft's players. Most notable are the ability to *Display all Frames* and to reset *Framerate*.

Imagen is a multimedia player mainly developed to support our HAV format including some special features very useful to most of video producers. Here is the quick list of what it offers:

* support for a total of 14 different formats (i.e. AVI, HAV, FLIC, MPEG-1, JPEG, PNG, TIFF, BMP, SGI etc.)

* capability to play image sequences (sequentially numbered images) which are frequently generated by many video rendering programs or video capture devices

* optionally it can be called with command line options (e.g. play and close)

* Imagen, the HAV player, is a freeware, and can be used for your private or commercial purposes with no restrictions. It is installed with EVS and MVS by default, and can be downloaded from:

http://www.gromada.com

DVD or Videotape: A very high quality animation can be produced as a DVD by having some reasonably priced hardware and software. A DVD recorder is required. Most of these include some software for transcoding normal animations into DVD format. This allows for converting an AVI file to DVD format.

This is the simplest and cheapest way to create high quality DVD or videotape.

C Tech 4D Interactive Models: Using the Record_4DIM module you can create 4DIM files where each frame of the animation is a true 3D model. You can interact with each frame by zooming, panning, or rotating. Frames can include moving planes, changing plume levels or completely new models!

Create An Application

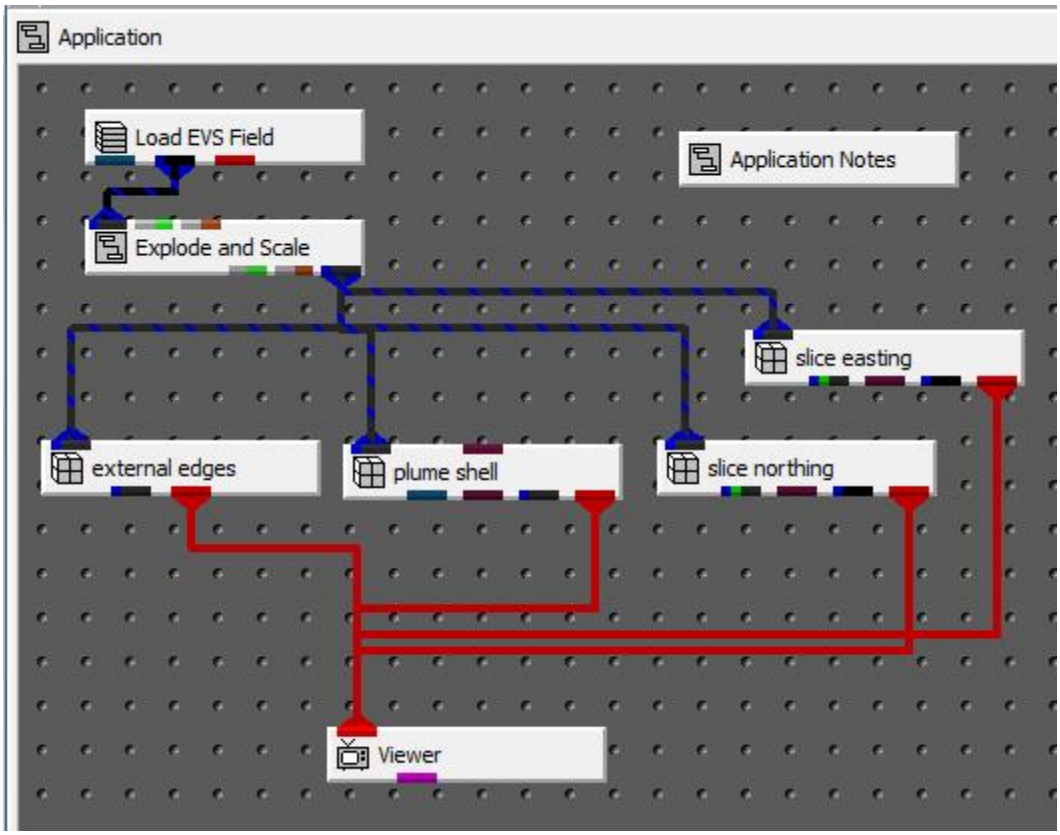
The first step in creating an animation is to decide what you are going to animate. In EVS, that involves:

1) Selecting (or building) an application

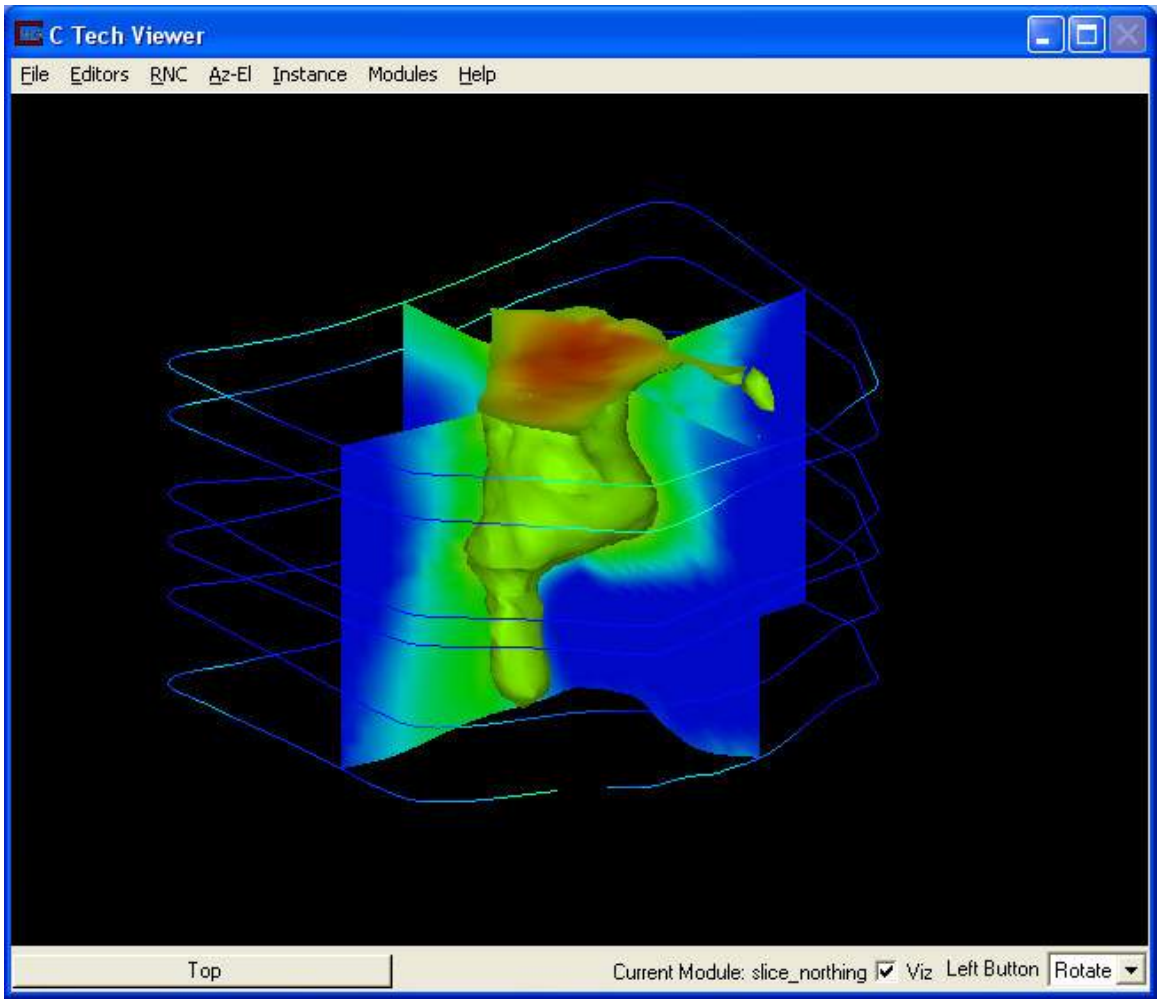
2) Choosing data files and parameters for the application....and

3) Running the application.

For this workbook, we will use the saved application animation_workbook.v. At this time start EVS and load the application \ctech\Applications\Workbooks\animation_workbook.v. Your network editor should look like this.



You should see the following in your Viewer.



Develop the Script

If you were going to produce a movie or TV show, an essential ingredient is the script. Actually, when you get comfortable with EVS and the Animator you can occasionally skip creating a script, but the more complex your animation, the more important it is to plan ahead. In our consulting, negotiation of the script often takes as much time as the actual creation of the animation. However, the script is the content of the animation and it is critically important that you convey the proper message by having the proper content.

For us, the script is a set of key frames. Each of these frames will serve as a starting and ending point for each sequence (or scene) of our animation. The Animator determines any module parameter that is to change during a sequence and interpolates the parameter during the frames for that scene.

The script does not have to be elaborate. All you need are notes to define the key frames and identify active module parameters (those that change) between key frames. Since this is our first animation, we won't get too fancy.

The key ingredients of a script can be presented as four questions which are:

1. What are the starting conditions of this scene?

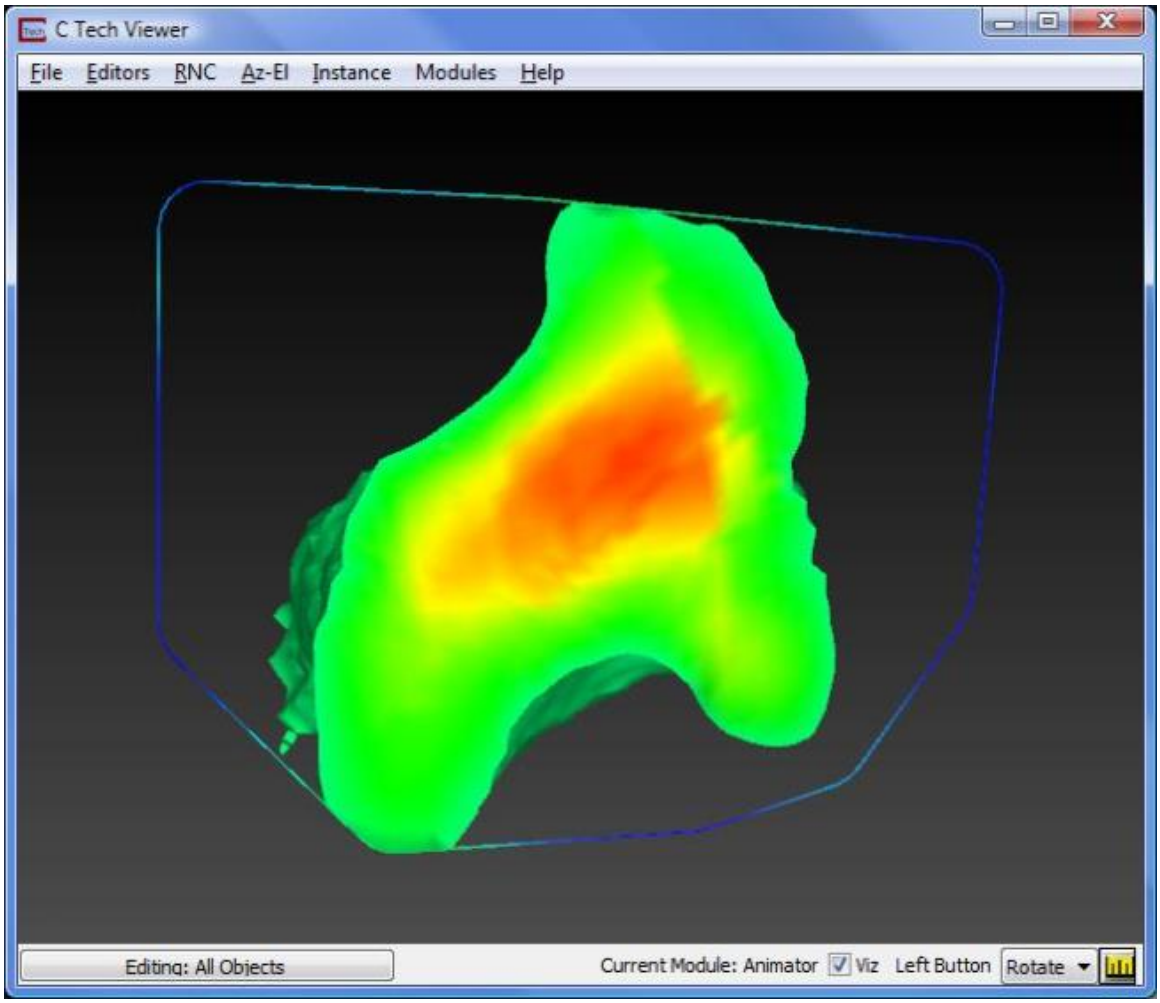
2. What is the ending conditions of this scene?
3. How long should the scene last?
4. How long should we pause at the end of the scene?

For this workbook, we'll develop the script as we progress, but, even here, we have a plan for the animation.

Initial Parameter Settings: Scene 1 Start

1. On plume_shell set the subsetting level to 1.0
2. On slice_northing set the Position to 13122.0.
3. On slice_easting set the Position to 11618.0.
4. On Az-El set the view to: Scale = 1.00; Elevation = 90; Azimuth = 180.

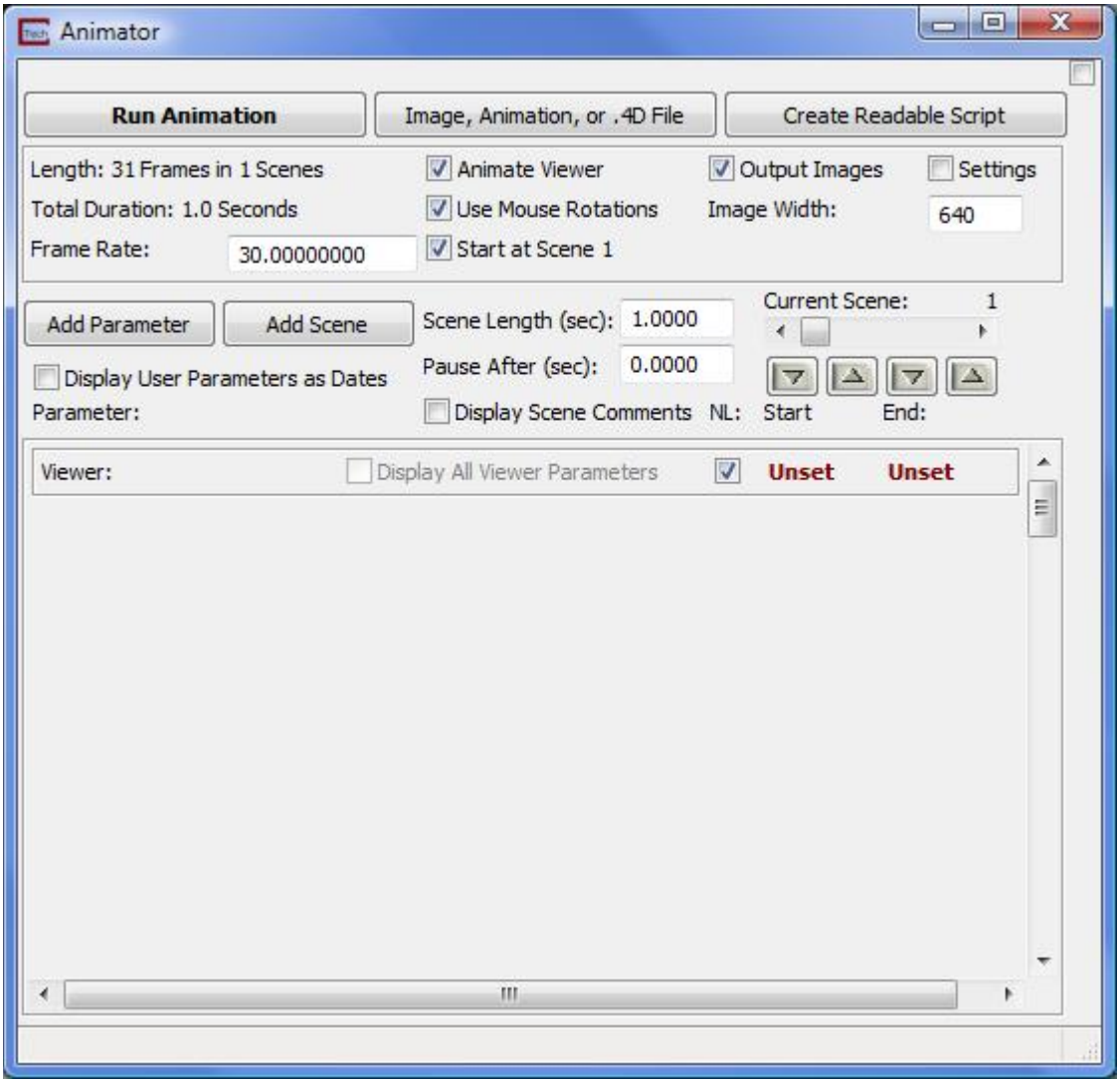
When this is done you should get the following view which will be our first key frame which is the Start of Scene 1.



Using the Animator

Now let's instance the Animator module and add all of the parameters we plan to animate. We are fortunate that should we decide later in the

animation creation process that we want to animate an additional parameter, we can easily go back, add the new parameter and adjust its setting at each scene.



There are two fundamentally different ways that we can control the rotations, translations and zooming in the Viewer using the Animator. The default method is "Use Mouse Rotations". Basically, this method works in a very intuitive manner allowing you to set the view using any combination of Az-El operations and mouse actions (rotations, translations and zooming). You set the view at the beginning of Scene 1 and the end of every other scene.

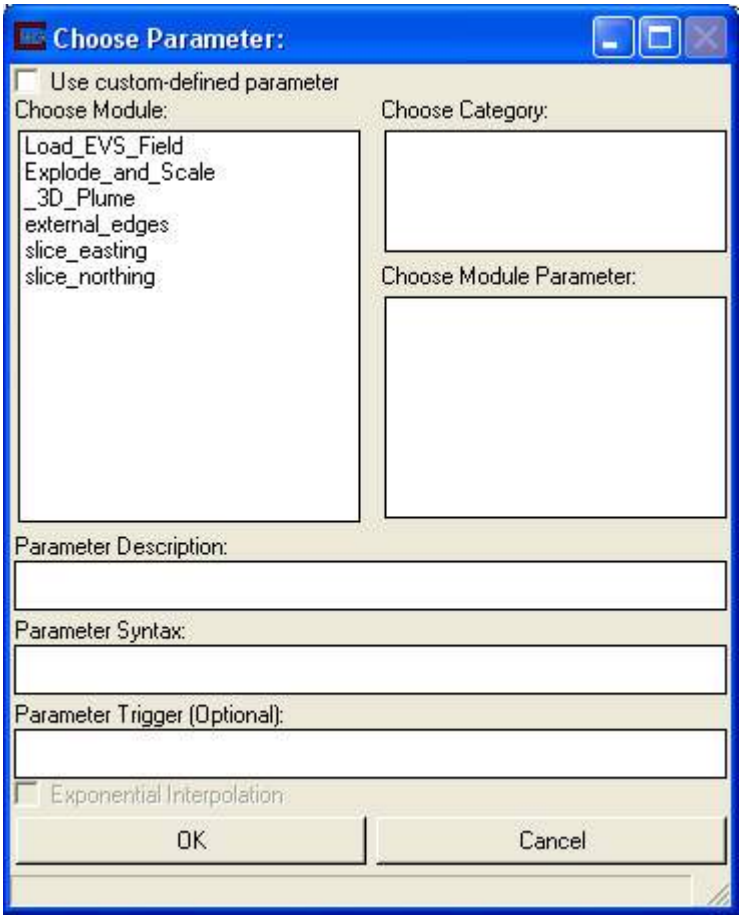
When the "Use Mouse Rotations" toggle is off, the view for each scene must be set with the Az-El panel exclusively. No mouse actions can be used since these will not be recorded in the Animator nor used to create your animation. Though this method is more restrictive, it offers one advantage which is that Azimuth values can be less than 0 or more than 360 degrees, allowing you to rotate clockwise or counterclockwise more than 180 (or even 360) degrees.

When specifying the view with mouse rotations, a single scene cannot cause rotations of more than 180 degrees since the path taken will always be the shortest route on the surface of a sphere.

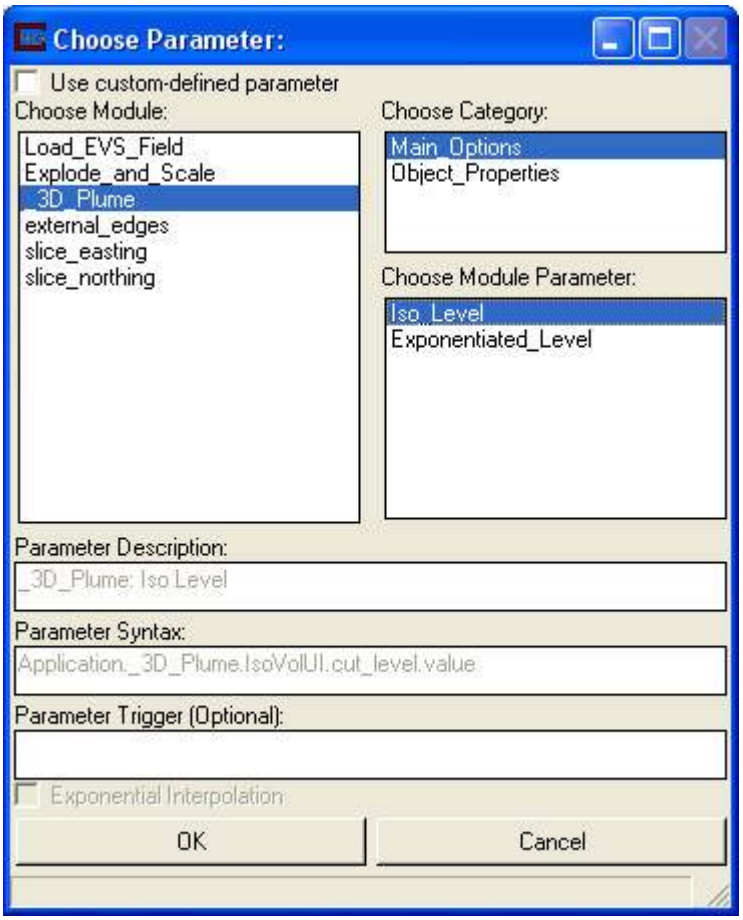
For this workbook we're going to use the Viewer Rotation specification option which can be **exactly** duplicated in this workbook which is to have the "Use Mouse Rotations" toggle off.

Once you've instanced the Animator (it doesn't matter where you put it in the application area) begin by turning the "Use Mouse Rotations" toggle off.

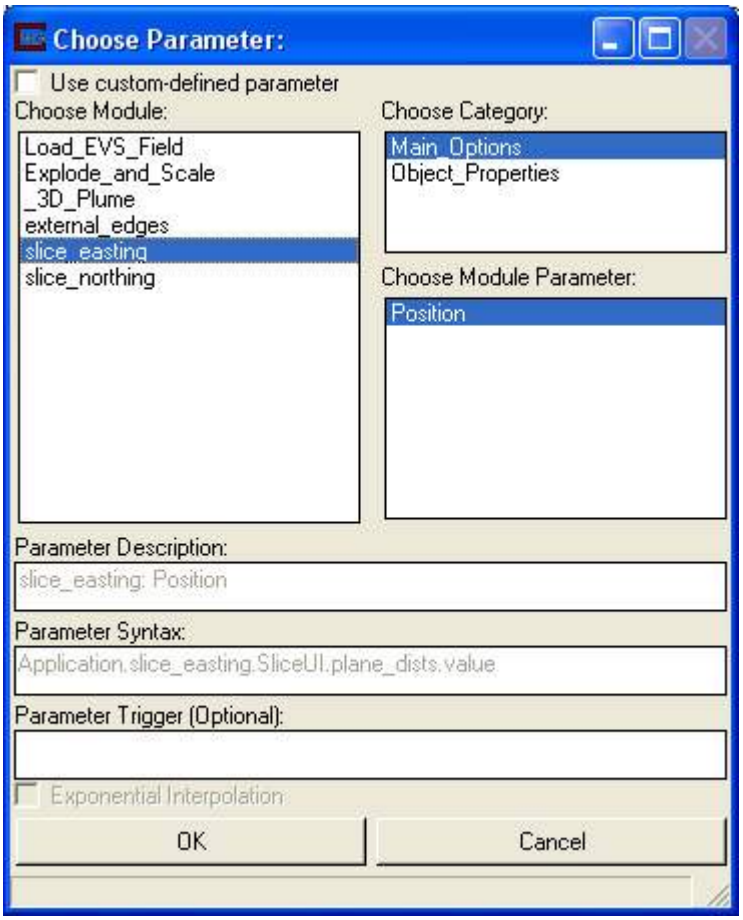
Next, press the *Add Parameter* button. You should see:



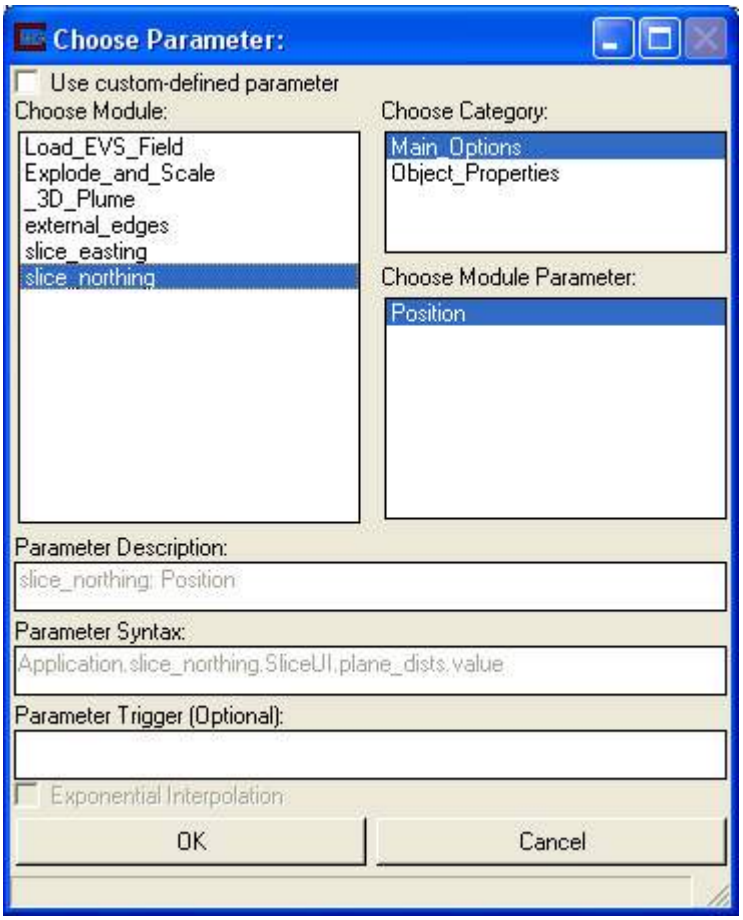
Select **_plume_shell**, **Main_Options**, and **subsetting level** and click on **OK** to add this parameter.



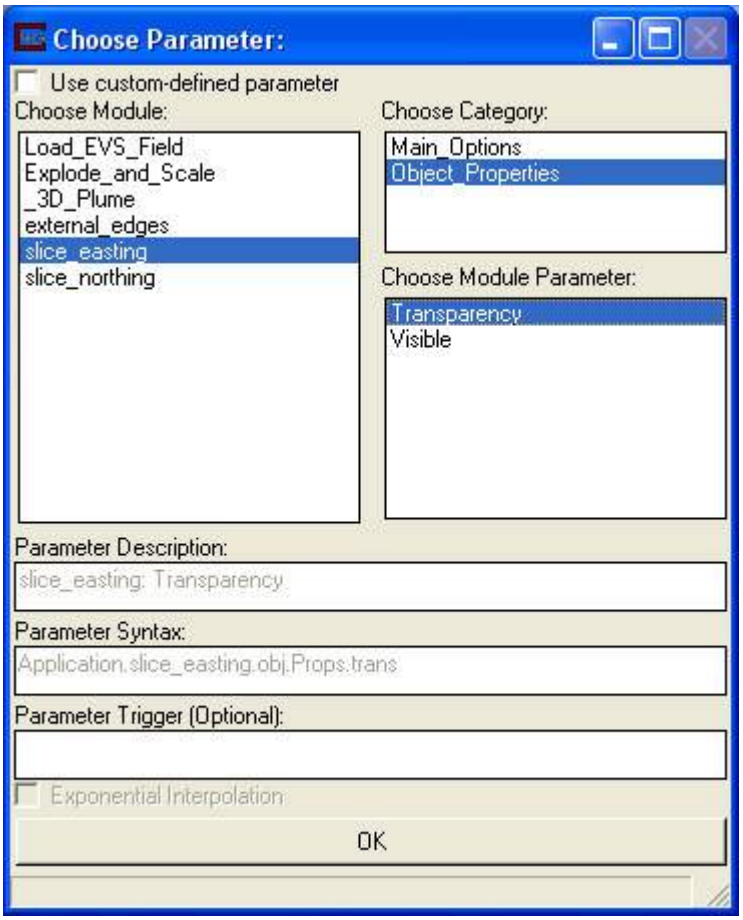
Press *Add Parameter* again and select **slice_easting**, **Main_Options**, and **Position** and click on **OK** to add this parameter.



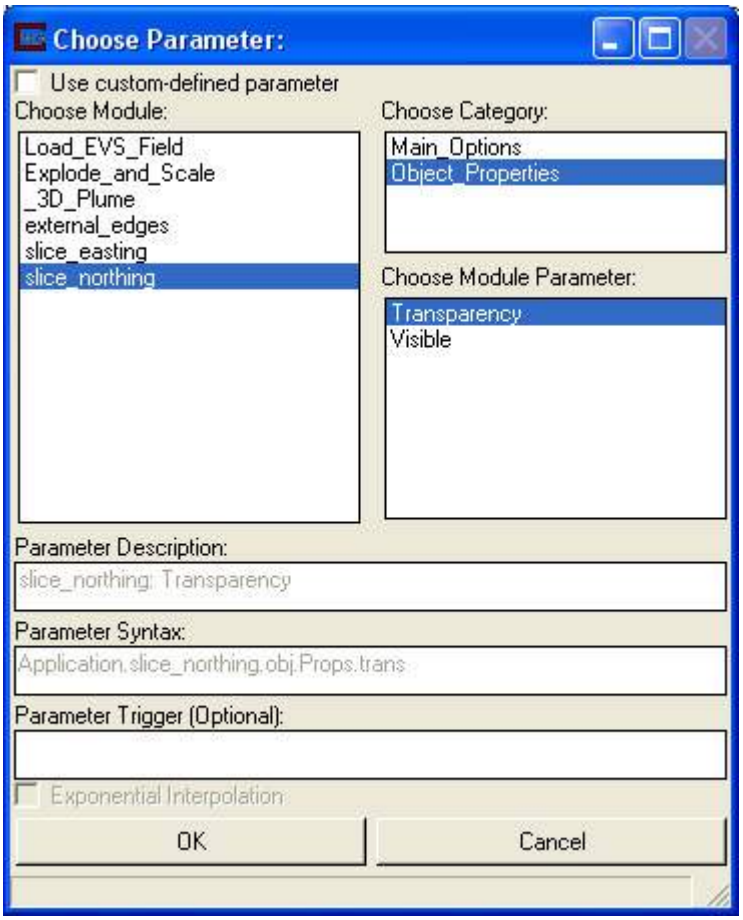
Press *Add Parameter* again and select **slice_northing**, **Main_Options**, and **Position** and click on **OK** to add this parameter.



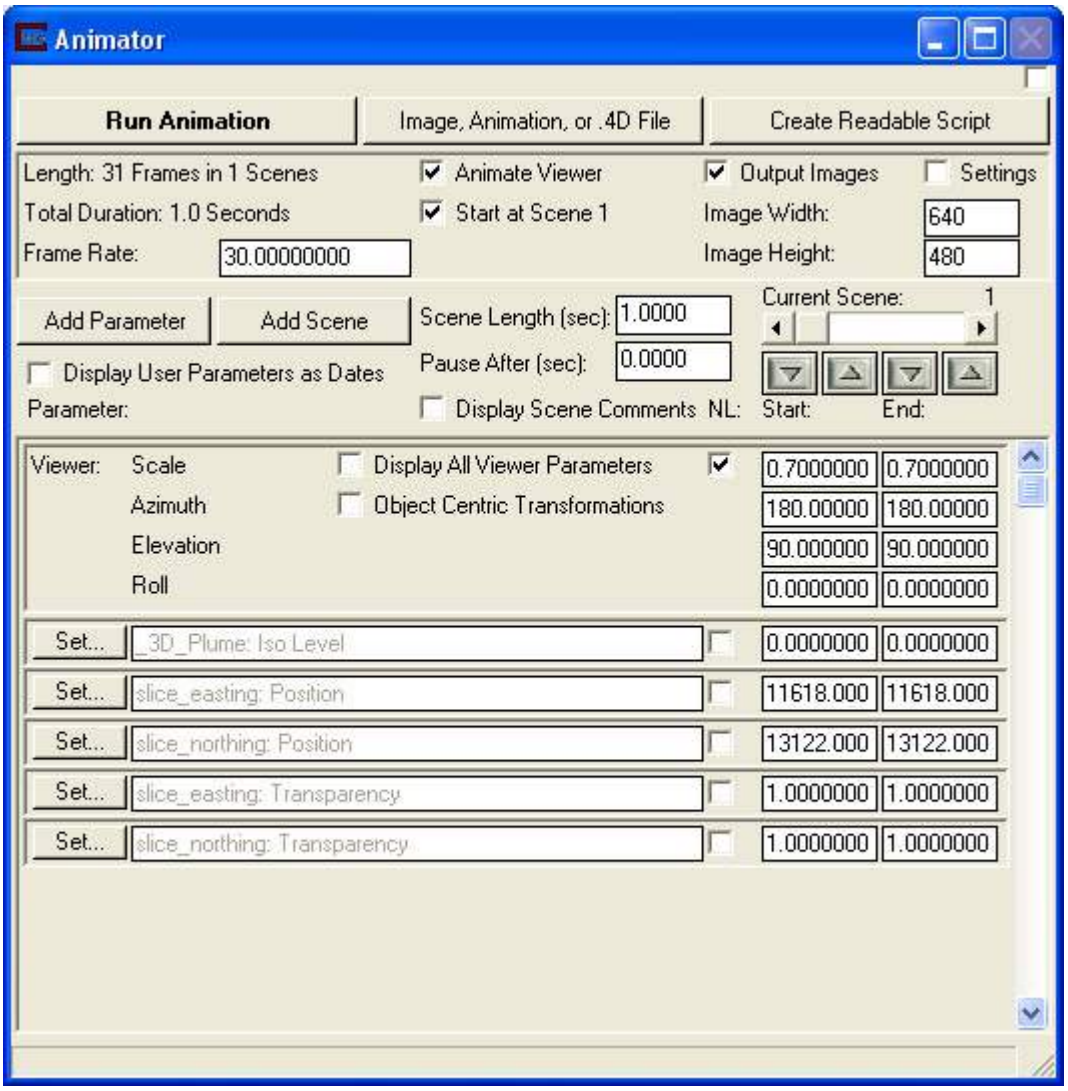
Press *Add Parameter* again and select **slice_easting**, **Object_Properties**, and **Transparency** and click on **OK** to add this parameter.



Press *Add Parameter* again and select **slice_northing**, **Object_Properties**, and **Transparency** and click on **OK** to add this parameter.



At this point, the Animator window should look like:

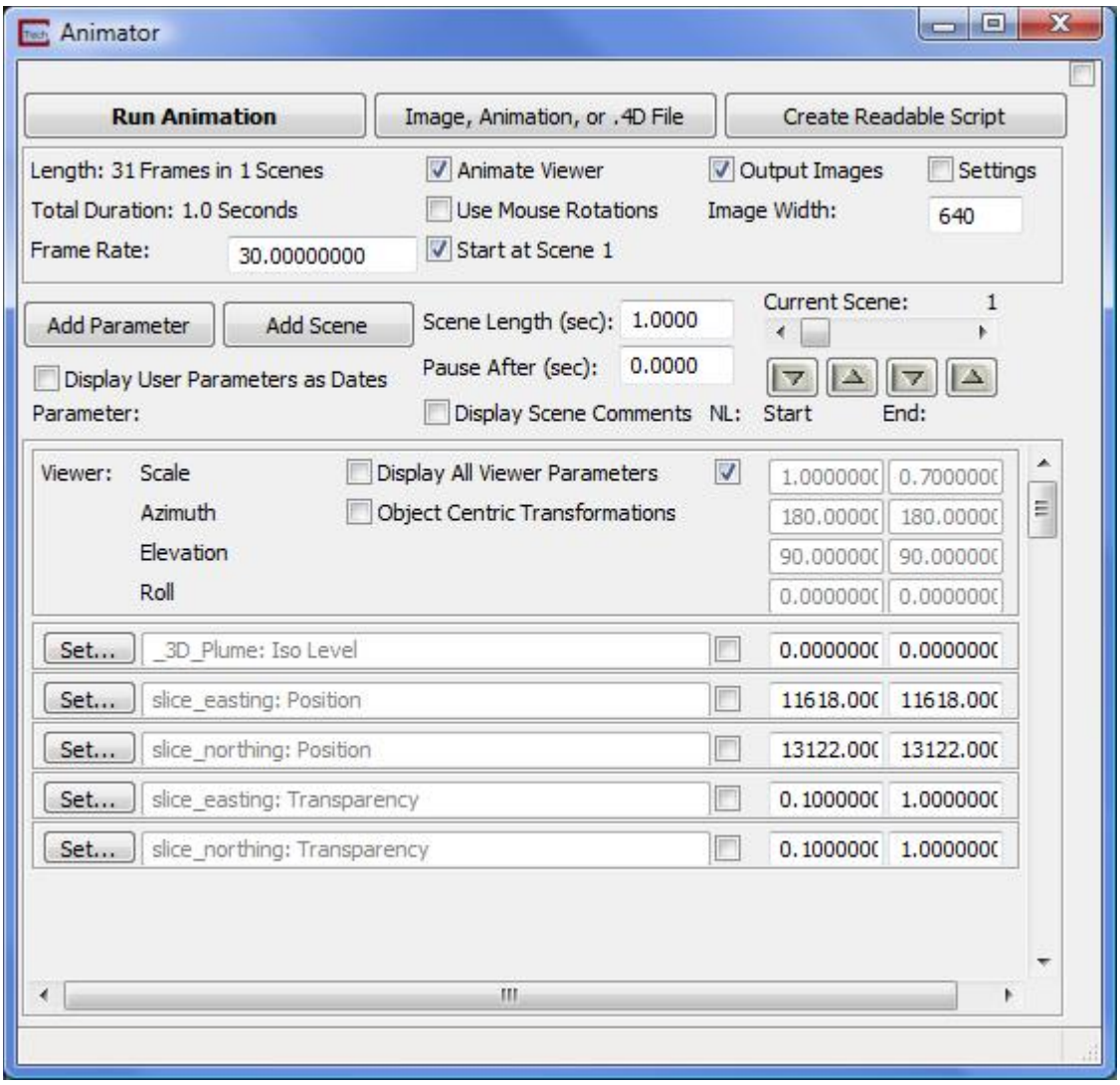


Note that when we add a parameter, its current value is set for all scenes of our animation. We only have one scene, so the Start and End values for all of the parameters we've added are set to their current levels. However, our Viewer settings (which were not added by us) are set to their default values, not the ones in our application.

To set these press the **Down Button** above *Start:* one time.

The only other change we want to make to our starting conditions is that we want the initial transparency of the slice planes to be very low whereas they are now 1.0. There are two ways we can accomplish this. We could set their values with the Object Editor and then use Down (arrow) Button again or we change their values in the Animator and use the Up Button.

Change slice_easting: Transparency and slice_northing: Transparency so that their Start values are both 0.1 and press the **UpButton above** Start one time. At this point, the Animator window should look like:



Our Viewer has actually changed, but, due to the Top View we now have, and the fact that the slices are out of the center of the model, we can't see it. Both of the slice planes are now transparent. This will be obvious later, but you can rotate your view if you want to check it out!

Now let's finish our first scene. Back in our application, let's decide what the ending conditions for the first scene should be. To recap, we have:

Scene 1 Start

1. Az-El view to: Scale = 1.00; Elevation = 90; Azimuth = 180.
2. plume_shell subsetting level to 0.0
3. slice_easting Position set to 11618.0.
4. slice_northing Position set to 13122.0.
5. slice_easting transparency set to 0.1.
6. slice_northing transparency set to 0.1.

Let's set our Scene 1 End to have the following values. Note that only the Viewer settings will change for this scene:

1. Az-El view to: Scale = 0.95; Elevation = 25.; Azimuth = 210.
2. plume_shell subsetting level to 0.0
3. slice_easting Position set to 11618.0.
4. slice_northing Position set to 13122.0.
5. slice_easting transparency set to 0.1.
6. slice_northing transparency set to 0.1.

To do this, all you need to do is change the view settings with Az-El and hit the  **DownButton** above *End:* one time.

We need to select a Scene Length (duration in seconds) for the first scene. As a guide, rotation rates should normally be in the range of 10 to 40 degrees per second. Given that we are rotating ~70 degrees during this scene, we will use a length of **5 seconds**.

The duration of each sequence and the frame rate selected are very important. Too low of a frame rate makes for a jerky animation but higher values make larger animation files that may not play well on slower computers. For DVD output, in the U.S. you will always want to use the 29.97002997 standard (yes, we know this number seems odd, but we didn't create the U.S. television standard). The duration of each scene determines how quickly things change. Faster can be dramatic, but may go by so quickly the message is lost. This is part of the art of animation. We will touch on some guidelines as we go through this workbook.

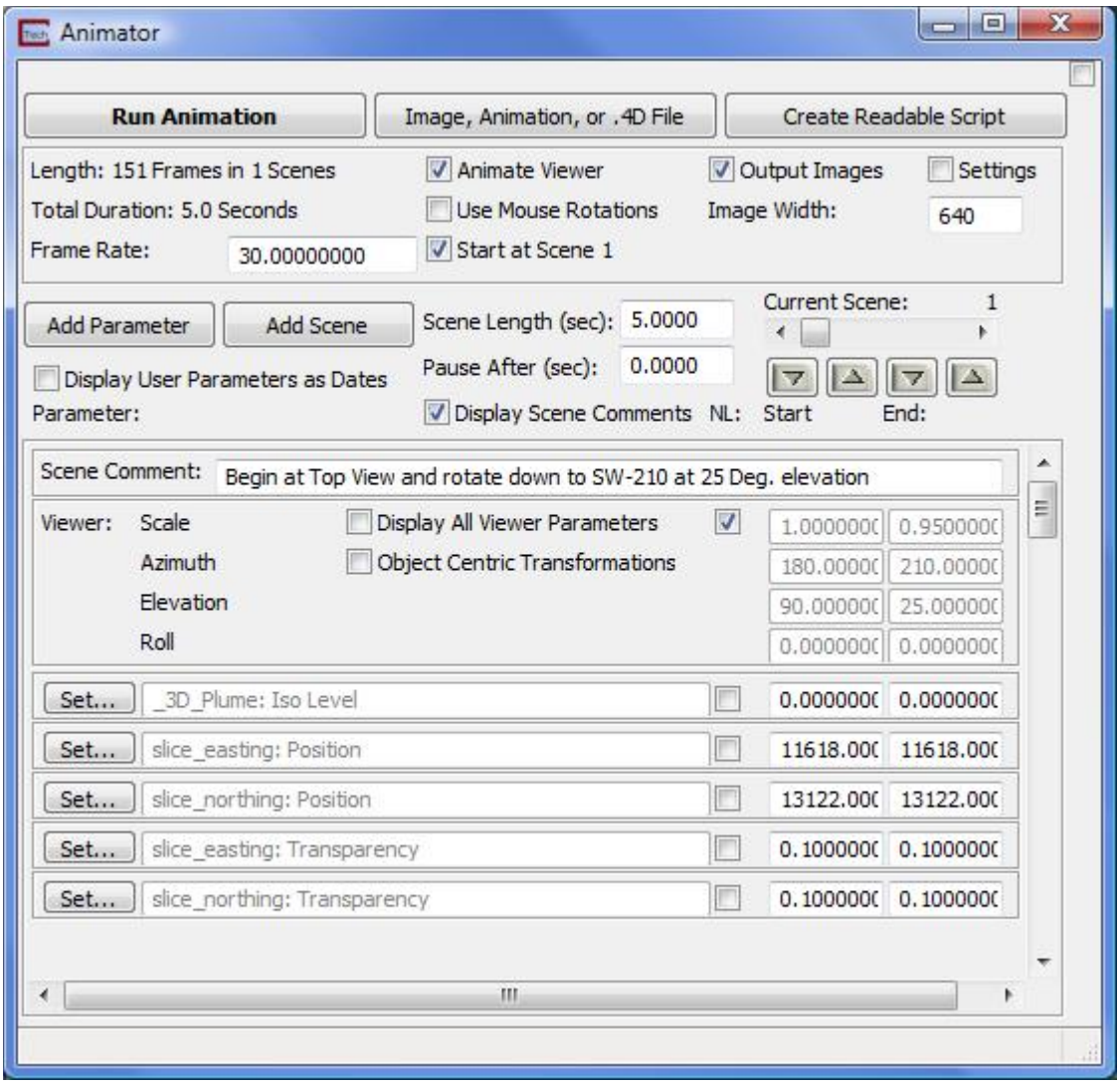
Finally, we have the option of including comments to describe the purpose of each scene. This is a great idea if you will be negotiating the script details with someone not familiar with C Tech software. For example, if you are preparing an animation for a lawyer or expert witness to use in court, your animation must communicate the message that they are trying to convey. Furthermore, it is important that each scene's message be of proper duration to allow them to narrate and make their points.

To do this, turn on the **Display Scene Comments** toggle which opens a large type-in that lets you describe the purpose or theme of the scene. These comments will be included in the Readable Script that we'll create later.

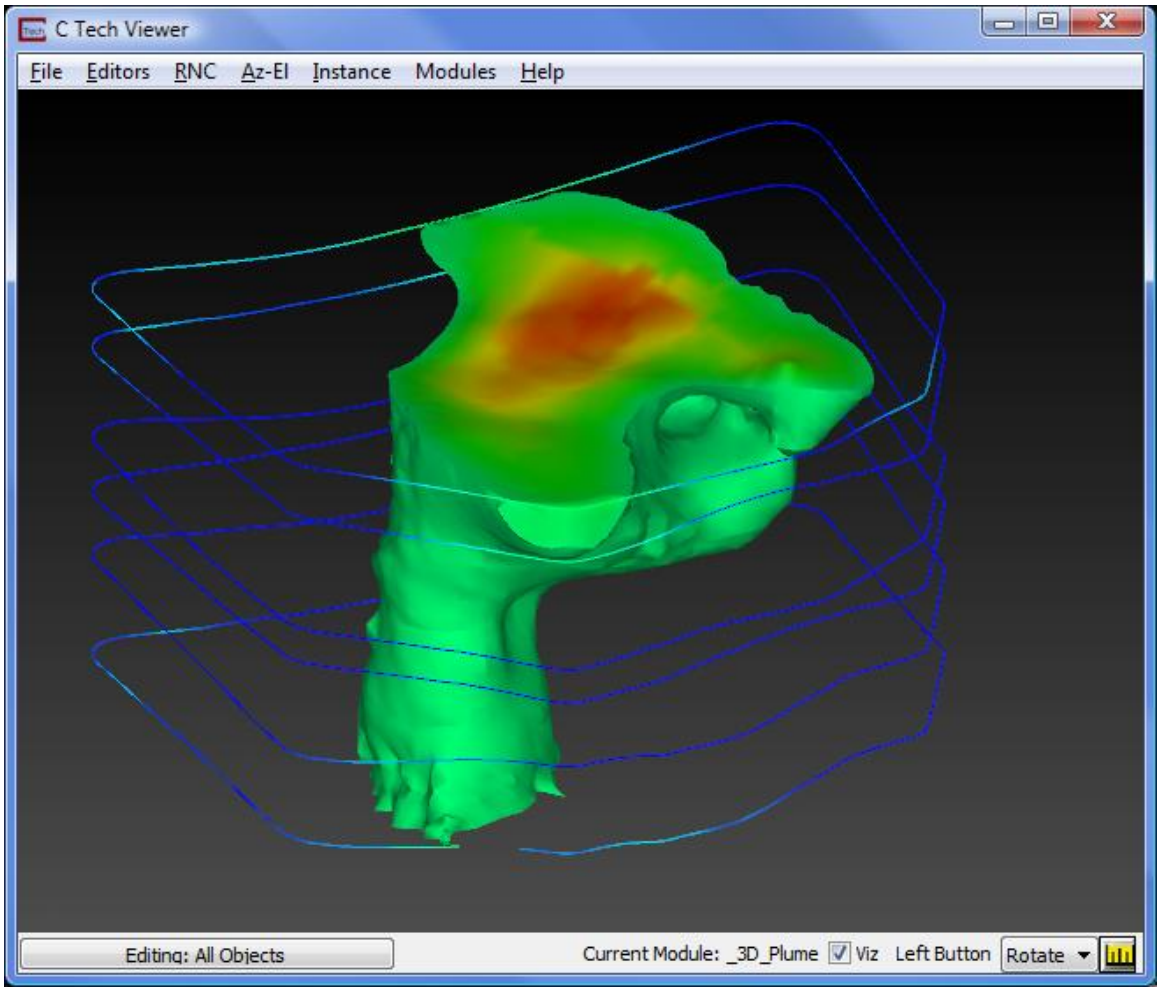
We'll enter the following description/comment for the first scene:

Begin at Top View and rotate down to SW-210 at 25 Deg. elevation

At this point, the Animator window should look like:



and the Viewer should show:



Now we are ready to add more scenes to our animation.

Setting Scenes

The procedure we'll use to set each of our additional scenes will be:

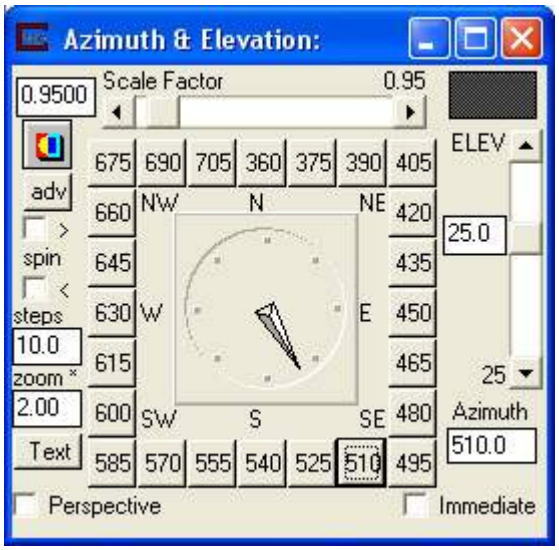
1. Set the Viewer (with Az-El) to the desired view for the end of the scene
2. Set all parameters that should change for the scene
3. Press the Add Scene button
4. Select a Scene Length (duration) in seconds
5. Enter a description/comment for the scene (optional)

SCENE TWO:

So.what do we want to do for our second scene? First, let's rotate the model through North to the Southeast and while we're doing this we'll increase the plume level (to make the plume smaller).

In order to make the rotation go through North we need the Azimuth to go from 210 to 360+150 or 510. We can get this by typing 510 into the Azimuth type-in . Alternatively, you can grab the Azimuth dial with your left mouse

and rotate it through North around to the SE. Notice that if you do either, the numbers in the Azimuth buttons all increase by 360 as shown below.



Let's also change plume_shell's subsetting level to 3.0. One note on this: We could have animated either the subsetting level (which is for this dataset in Log values) or the Exponentiated level. However, if we change levels with exponentiated values it will cause very rapid changes to the apparent plume size in the beginning of each scene, but towards the end, there will be very little apparent change. By using the log values (subsetting level vs. Exponentiated level) the apparent changes will be more uniform.

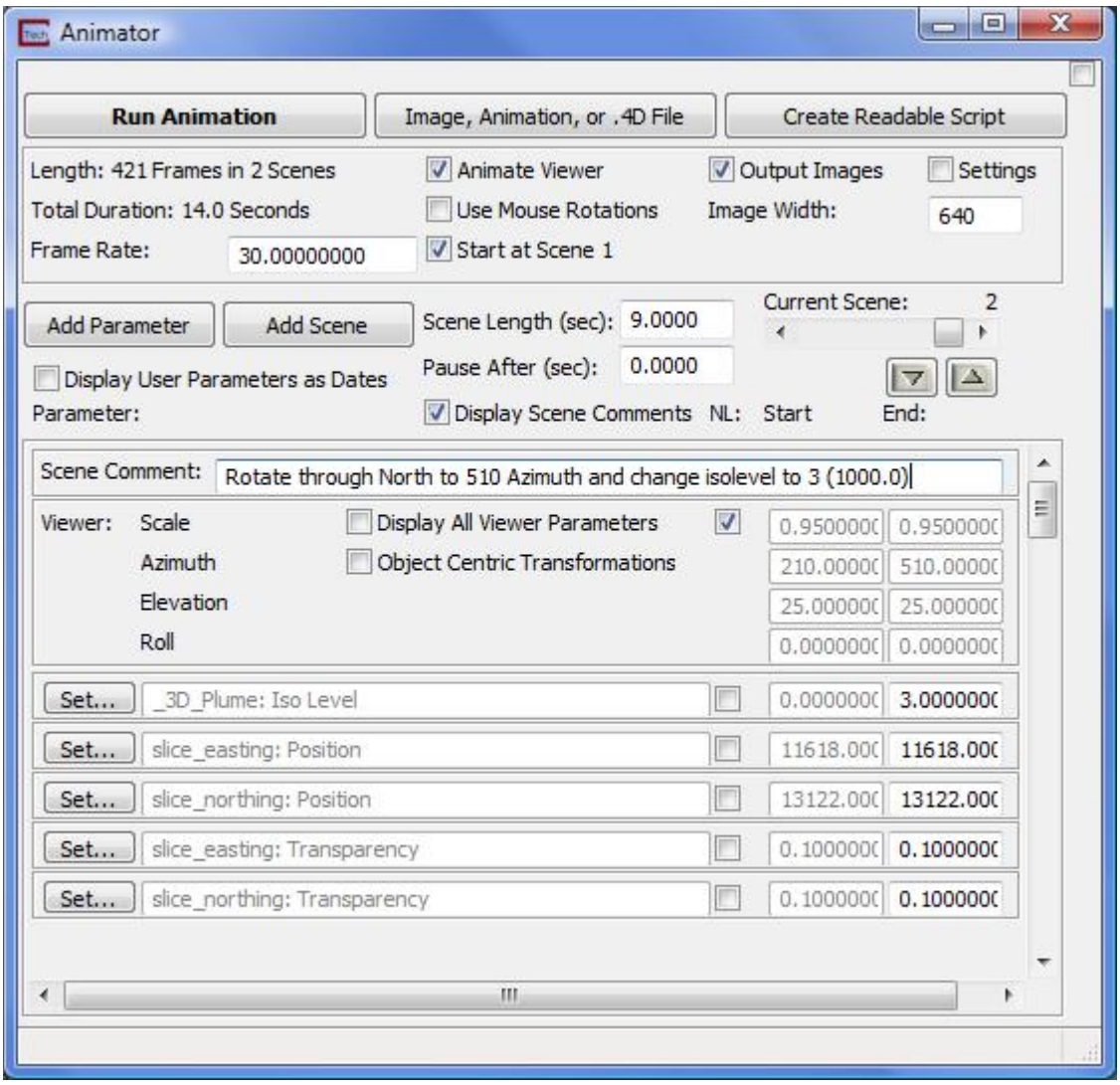
We could change other things, but let's stick with these two changes for this scene (rotation and plume level).

To set these values into our Animator for Scene 2 all we do now is press the **Add Scene** button once.

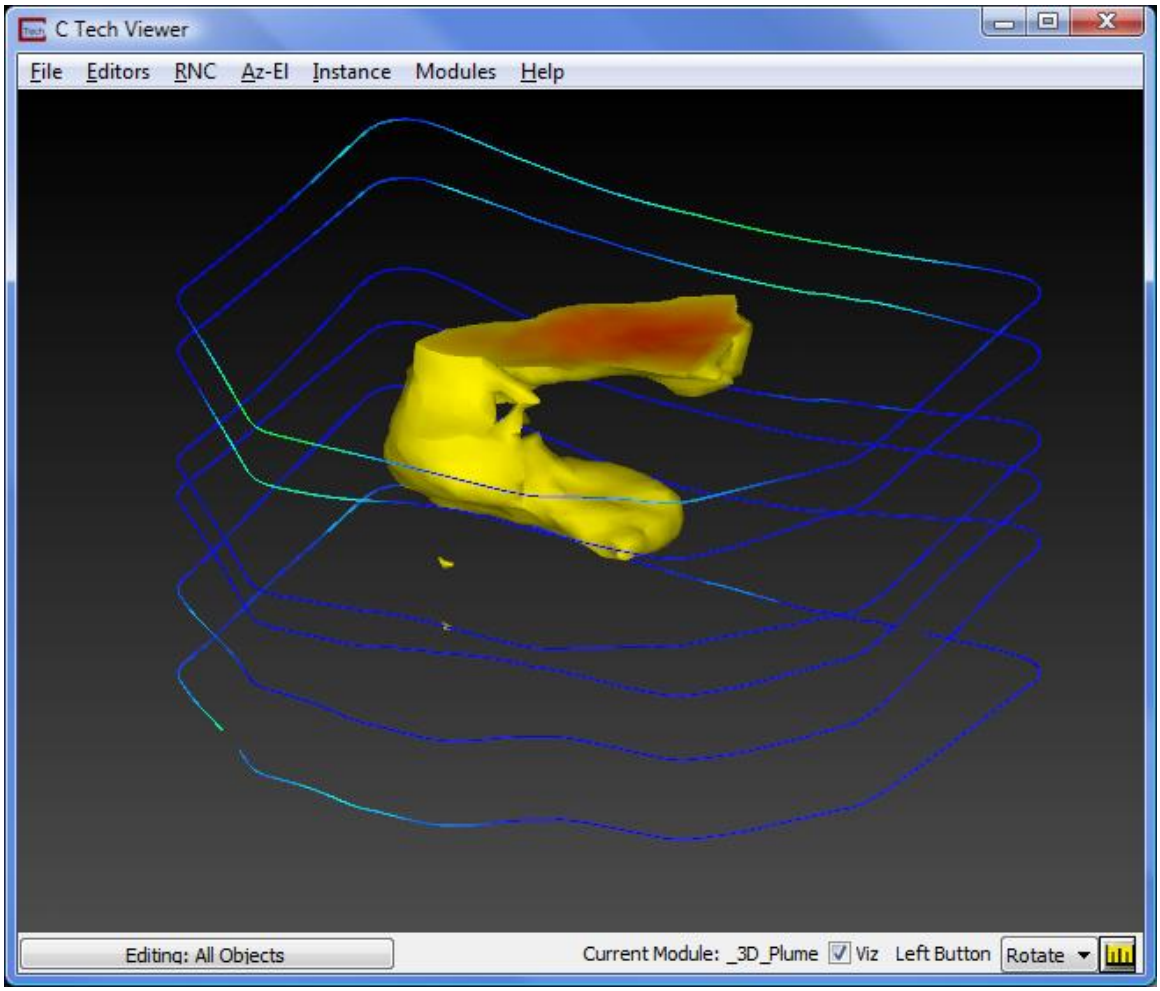
We'll set the scene length to 9 seconds and the Scene Comment to be:

Rotate through North to 510 Azimuth and change subsetting level to 3 (1000.0)

At this point our Animator should look like:



and the Viewer should show:

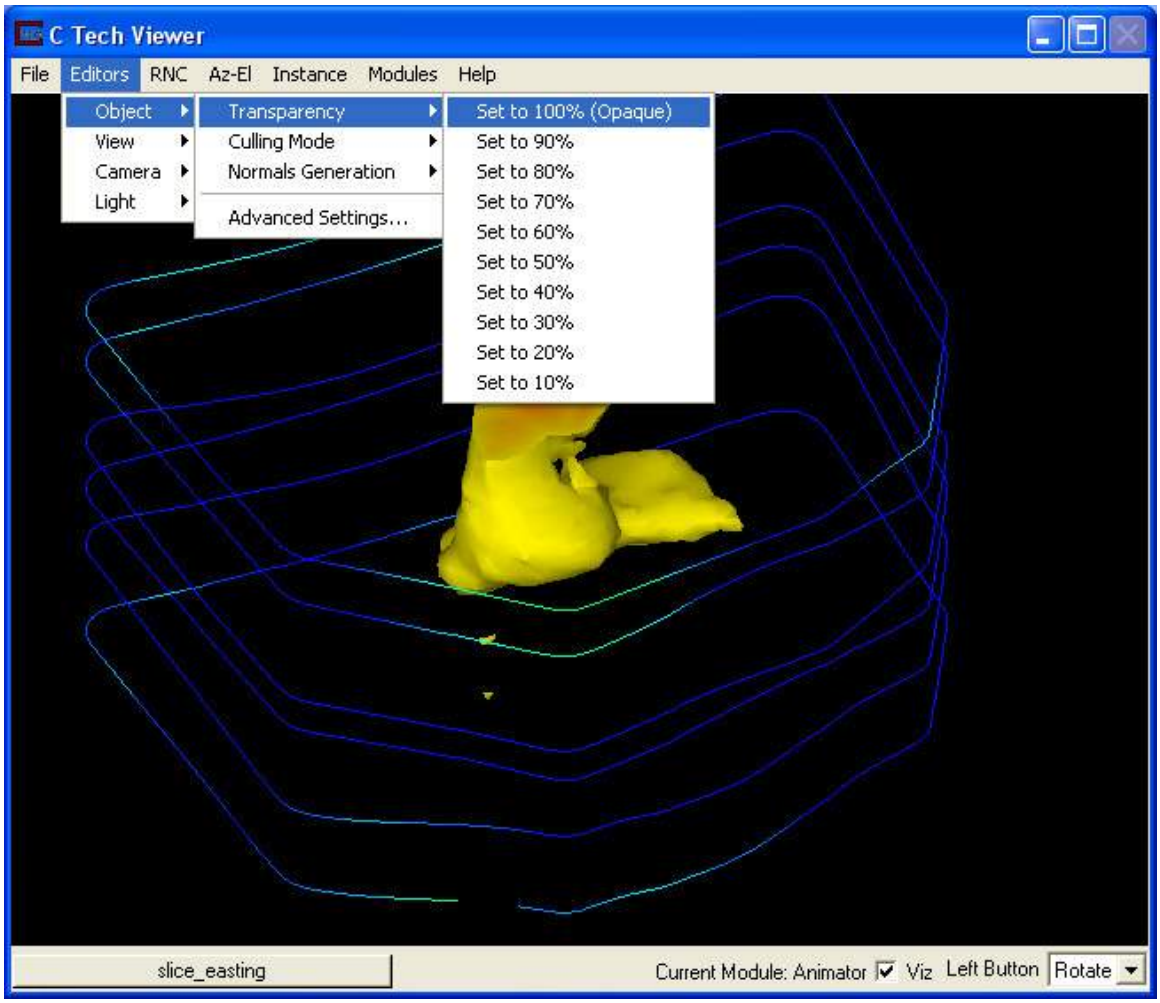


SCENE THREE:

For our third scene let's rotate through South to the Southwest while we increase the opacity of both slice planes and move them to opposite sides of the model.

Click on the Azimuth 570 button and change the elevation to 38.

Open the Object Selector (All Objects button at the bottom of the Viewer) and select slice_easting. Change the Transparency of this object as shown below:



Repeat this process for slice_northing.

Open the Object Selector and choose Top and hit OK.

Change slice_northing's position to 12679.0 and slice_easting's position to 11054.0.

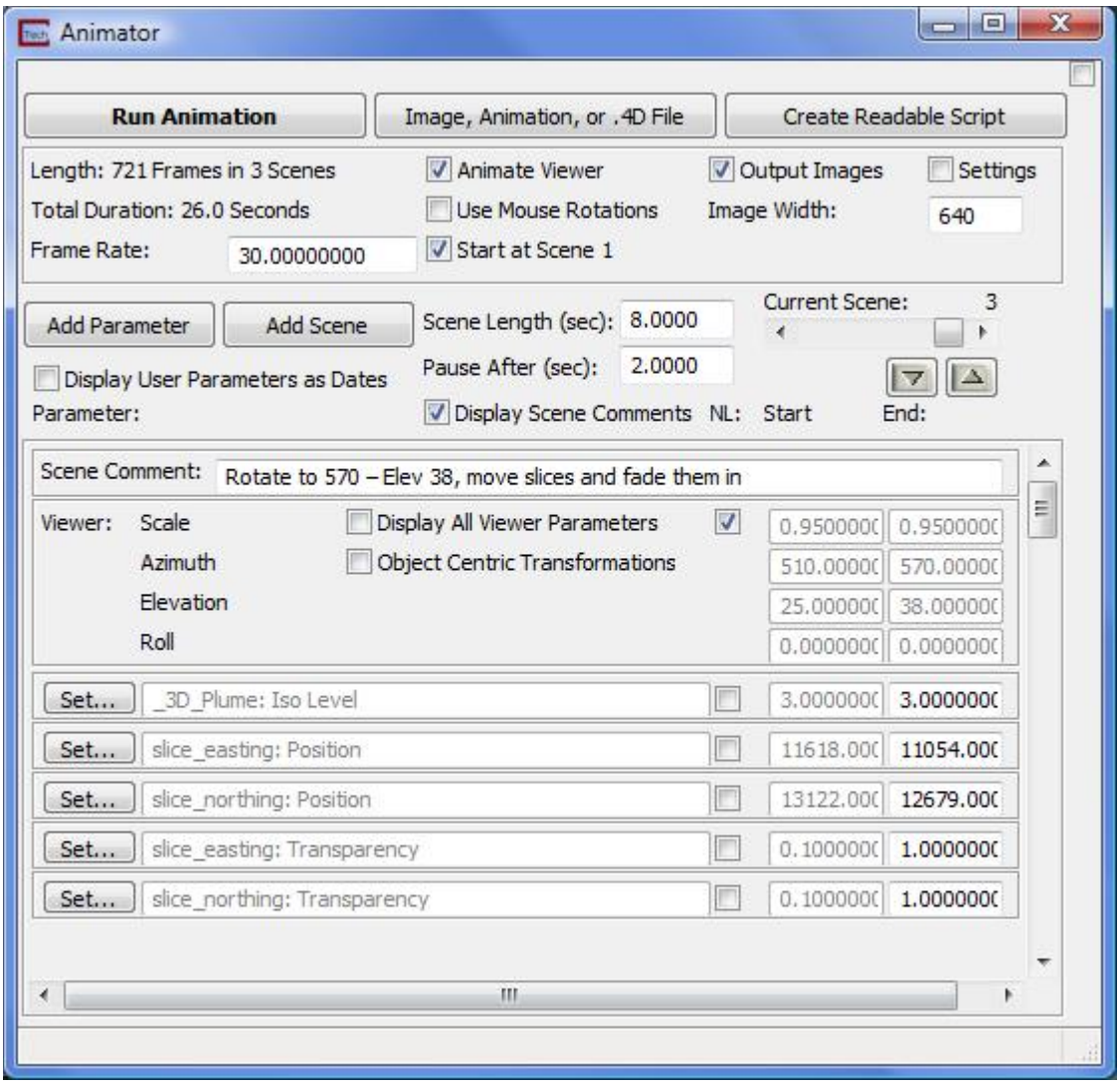
To set these values into our Animator for Scene 3 press the **Add Scene** button again.

We'll set the scene length to 8 seconds and the Scene Comment to be:

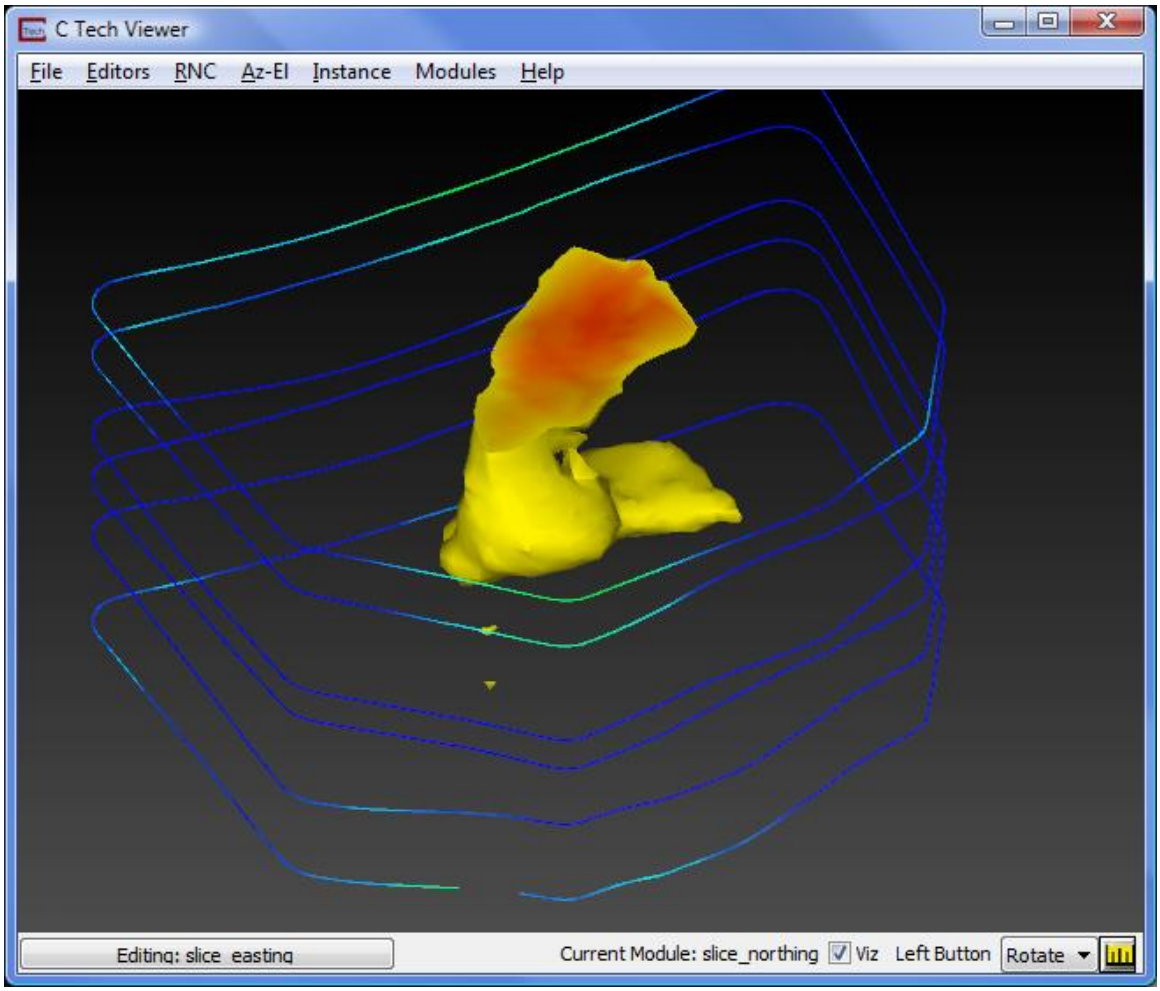
Rotate to 570– Elev 38, move slices and fade them in

Let's also set a 2 second pause at the end of this scene.

At this point our Animator should look like:



and the Viewer should show:



SCENEFOUR:

For our fourth and final scene let's rotate back to the original Top view while we move both slice planes to the centroid of the model and return the plume to its orginal level.

If you plan to have an animation run in a loop it is often nice if the starting and ending of the animation match. We won't be having the slice planes in the same position or transparency, however since the start and end views are Top views where the slices can't be seen (they're on edge), it should work out fine.

Also, since our current Azimuth is 570, if we return to 180 where we started it will require a (near) full rotation. Instead, we can finish at 540 deg. Azimuth which is visually equivalent.

Click on the Azimuth 540 button and change the elevation to 90 and the scale to 1.0.

Change slice_northing's position to 12900.0 and slice_easting's position to 11336.0. Let's also change plume_shell's subsetting level back to 0.0.

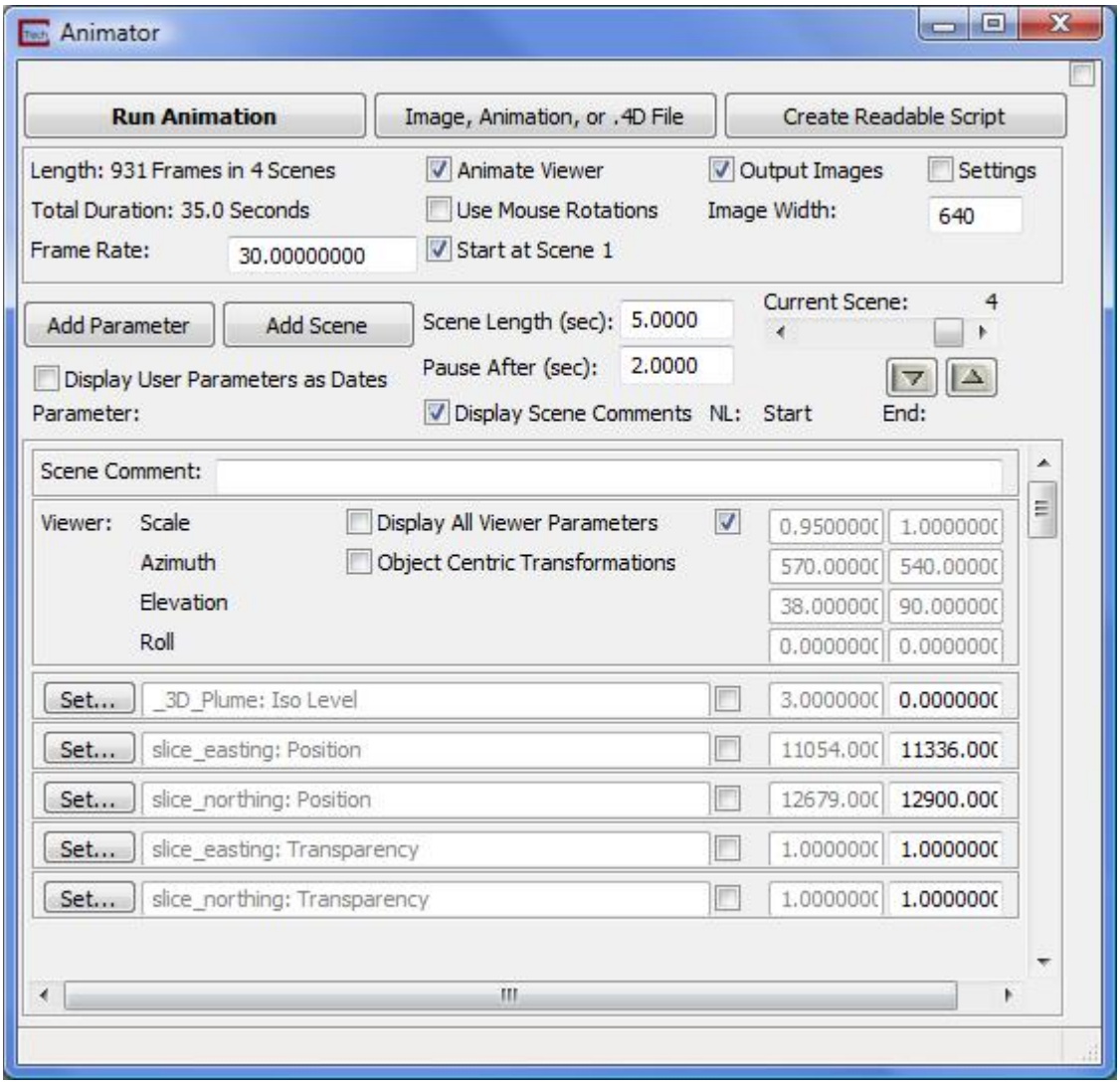
To set these values into our Animator for Scene 3 press the **Add Scene** button again.

We'll set the scene length to 5 seconds and the Scene Comment to be:

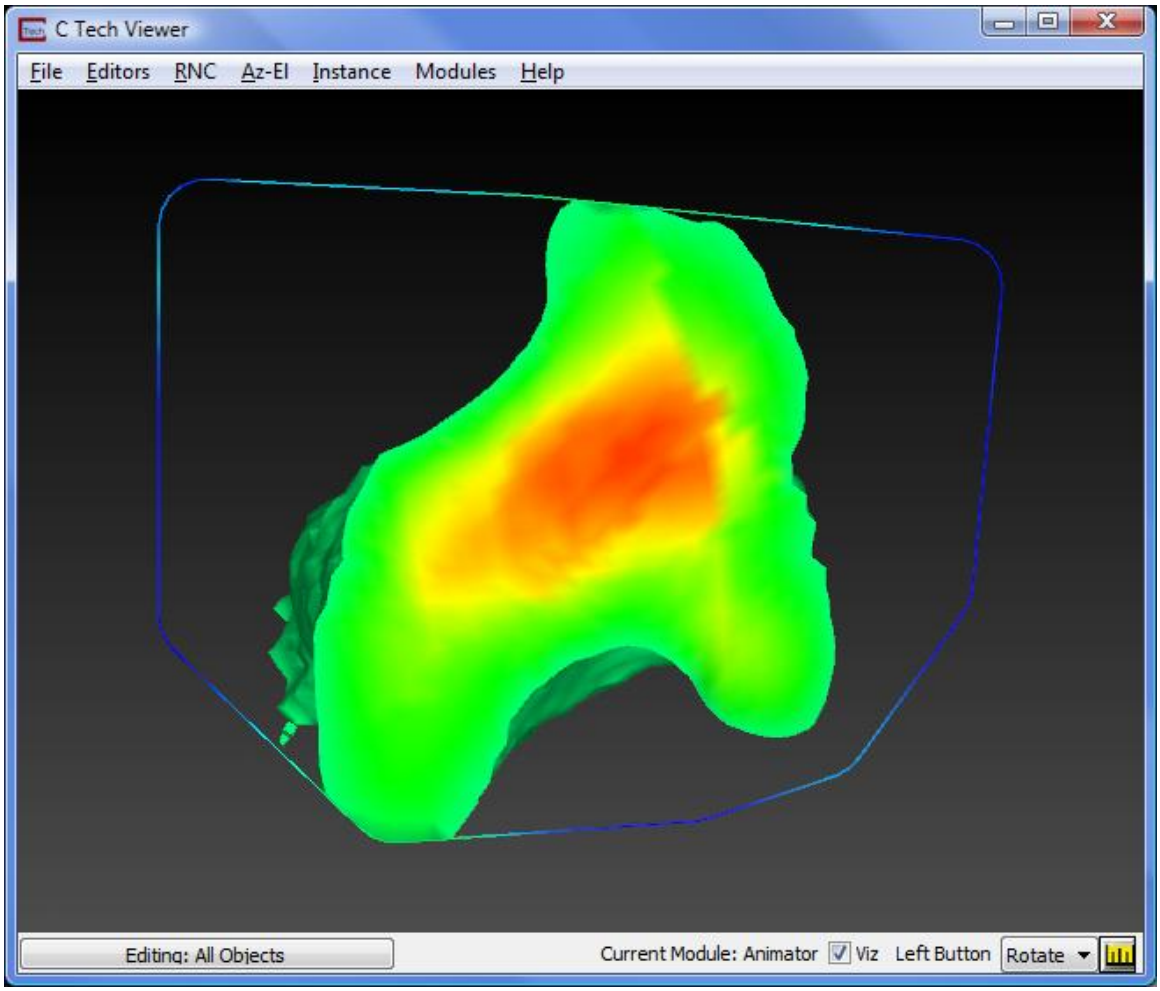
Rotate to 540– Elev 90, move slices to center and reset plume level to 0.0

Let's also set a 2 second pause at the end of this scene.

At this point our Animator should look like:



and the Viewer should show:



Next we'll review our Animation and if we like it we'll make a movie!

Testing the Animation

At this point we have a four scene animation with a duration of 35 seconds.

The Animator provides us with an excellent way to record and document our script. In addition to saving all aspects of our animation when we save the application (or the Animator object), we can also create a Readable Script. If you do this, it will create a file with the following content. Please note that the indentation shown below is as created to help segregate the scenes.

```
Initial Conditions:
 View:
  Azimuth = 180.00
  Elevation = 90.00
  Scale = 1.00
  Roll = 0.00
  True Scale = 0.017769601196
  XCenter = 11336.29
  YCenter = 12900.69
  ZCenter = -111.35
```

```
  XTrans = -201.44
  YTrans = -229.24
  ZTrans = 1.98
 _plume_shell: subsetting level = 0.000000
 slice_easting: Position = 11618.000000
 slice_northing: Position = 13122.000000
 slice_easting: Transparency = 0.100000
 slice_northing: Transparency = 0.100000
Start scene 1:
 Comment: Begin at Top View and rotate down to SW-210 at 25 Deg.
elevation
 Duration: 5.00 Seconds
 Change view to:
  Azimuth = 210.00
  Elevation = 25.00
  Scale = 0.95
  Roll = 0.00
  True Scale = 0.017769601196
  XCenter = 11336.29
  YCenter = 12900.69
  ZCenter = -111.35
  XTrans = -201.44
  YTrans = -229.24
  ZTrans = 1.98
Start scene 2:
 Comment: Rotate through North to 510 Azimuth and change
subsetting level to 3 (1000.0)
 Duration: 9.00 Seconds
 Change view to:
  Azimuth = 510.00
  Elevation = 25.00
  Scale = 0.95
  Roll = 0.00
  True Scale = 0.017769601196
  XCenter = 11336.29
  YCenter = 12900.69
  ZCenter = -111.35
  XTrans = -201.44
  YTrans = -229.24
  ZTrans = 1.98
 Change _plume_shell: subsetting level to 3.000000
Start scene 3:
```

```
 Comment: Rotate to 570 - Elev 38, move slices and fade them in
 Duration: 8.00 Seconds
 Change view to:
  Azimuth = 570.00
  Elevation = 38.00
  Scale = 0.95
  Roll = 0.00
  True Scale = 0.017769601196
  XCenter = 11336.29
  YCenter = 12900.69
  ZCenter = -111.35
  XTrans = -201.44
  YTrans = -229.24
  ZTrans = 1.98
 Change slice_easting: Position to 11054.000000
 Change slice_northing: Position to 12679.000000
 Change slice_easting: Transparency to 1.000000
 Change slice_northing: Transparency to 1.000000
 Pause for 2.00 seconds.
Start scene 4:
 Comment: Rotate to 540 - Elev 90, move slices to center and
reset plume level to 0.0
 Duration: 5.00 Seconds
 Change view to:
  Azimuth = 540.00
  Elevation = 90.00
  Scale = 1.00
  Roll = 0.00
  True Scale = 0.017769601196
  XCenter = 11336.29
  YCenter = 12900.69
  ZCenter = -111.35
  XTrans = -201.44
  YTrans = -229.24
  ZTrans = 1.98
 Change _plume_shell: subsetting level to 0.000000
 Change slice_easting: Position to 11336.000000
 Change slice_northing: Position to 12900.000000
 Pause for 2.00 seconds.
```

Combined with a few pictures, a 4DIM file, and/or a coarse version of your animation, this script should provide you the ability to communicate the

intentions of your script to a layperson. This should allow you to discuss and revise the script as necessary

Since we have yet to set an Image, Animation or .4D file name, if we RUN the animation it won't create anything, but it will control our application and we'll see each frame in the Viewer. For testing purposes, we usually change the frame rate to a lower level so we can get a quick review of the content and duration of the animation. Change the Frame Rate to 5.0 (frames per second) and click on the **Run Animation** button.

If we don't like some aspect of our animation we can change it. The things we can change are:

1. Content
2. Scene Duration
3. After Scene Pauses
4. Adding additional parameters
5. Adding additional scenes

In other words, we can change anything!

After my test run, there was only one thing that I would change and it was minor. For Scene 3, I didn't like the 38 degree elevation. I felt it should be lower.

To change this you do the following:

1. Set the Current Scene slider to 3
2. Hit the  **Up Button** above End one time to use the end of scene values from the Animator.
3. Change the view with Az-El to 30 degree elevation.
4. Hit the  **Down Button** above End one time to set the values in the Animator.

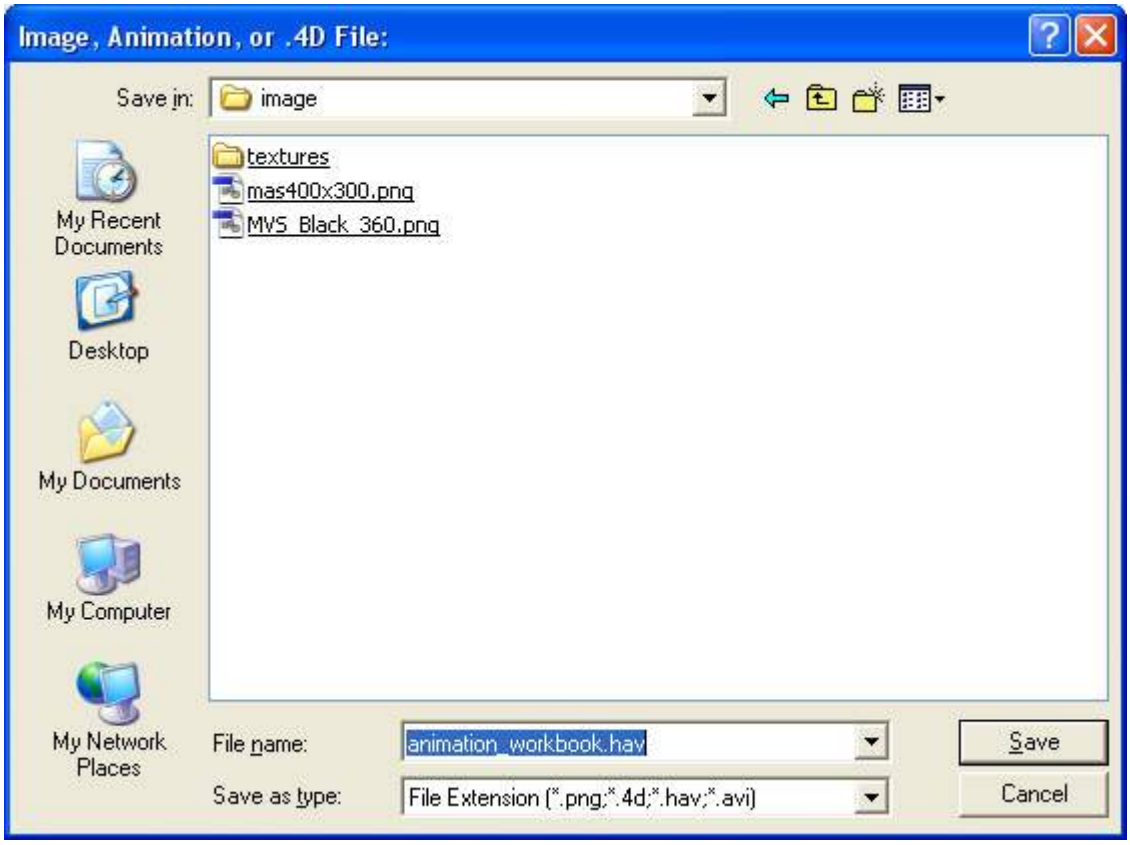
Note: You can see what the end of scene key frame is for every scene by setting the current scene and using the Up Button.

Set File Name

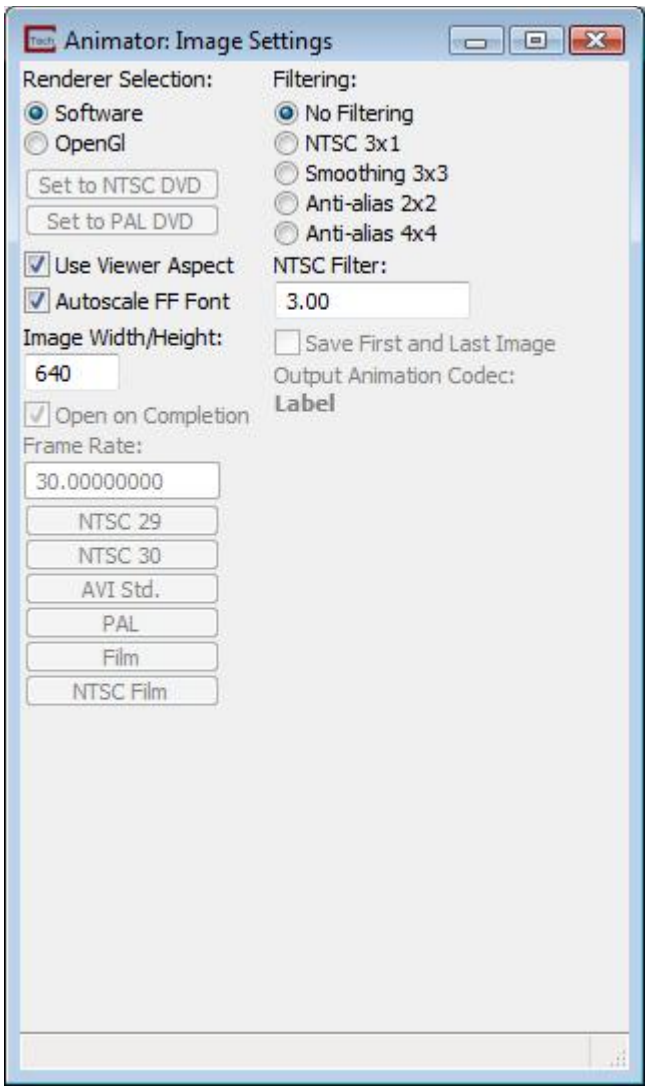
Now let's specify the name and type of animation file to create. You can specify any image file format type (e.g. .bmp, .png, .tif, .jpg, etc.), but if you want a sequence of images, we suggest using PNG. However, if your goal is a movie then you can specify that directly. The three principal bitmap animation formats are AVI, MPG and HAV. An important consideration with any of these formats is whether they are lossy or lossless. Lossy CODECs cause degradation in the picture quality but often produce smaller animations. All MPEG (MPG) CODECs are lossy and very few AVI codecs are lossless. However, the HAV format offers a lossless CODEC that is included with C Tech software.

For this workbook, we'll choose the HAV format by selecting a file name with a .HAV suffix. If using an AVI codec, it is a good idea to use a lossless AVI codec. C Tech recommends the Lagarith AVI codec, which can be downloaded from the C Tech website. http://www.ctech.com

Click on the Image, Animation or .4D File button and select the file name animation_workbook.hav



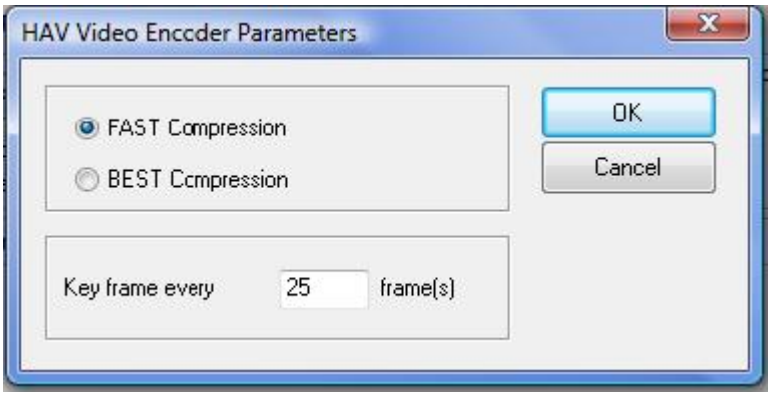
The Animator: Image Setting window should open. Choose the settings as shown below:



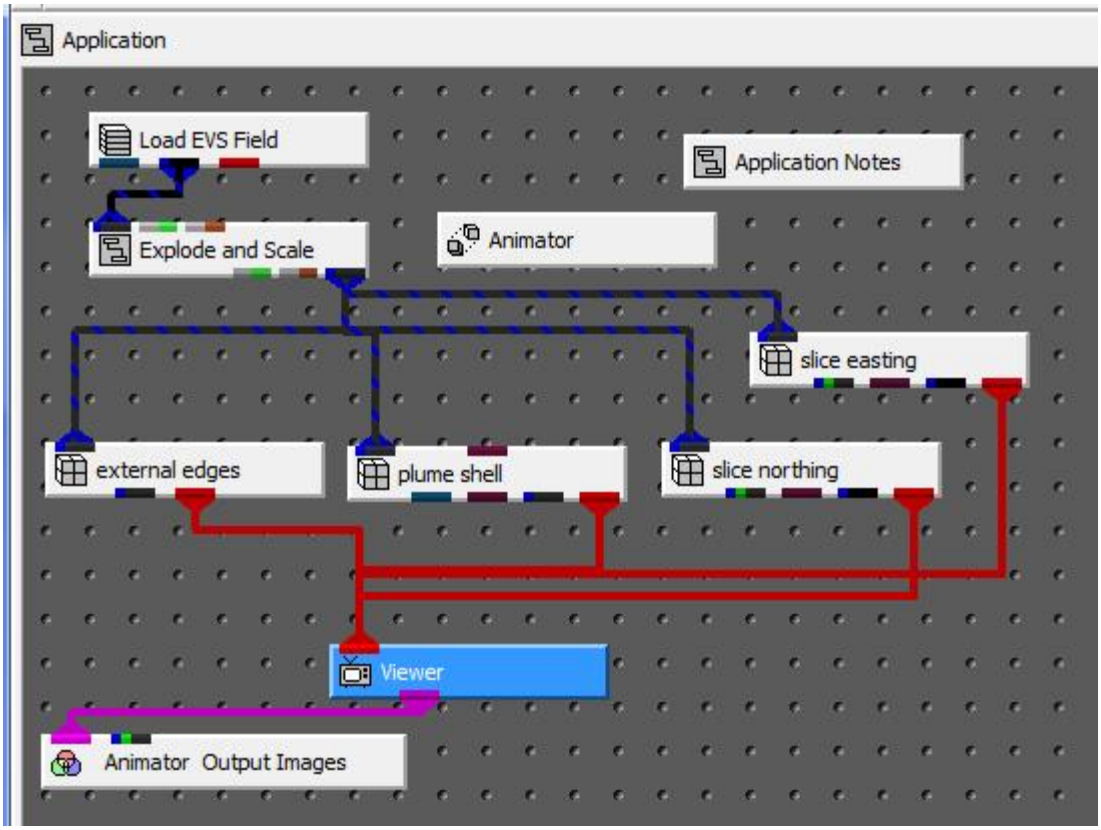
Create Animation File

At this point you can close the settings window and press the **Run Animation** button.

The first thing that will happen is that the CODEC selection and/or configuration window. Since we've chosen HAV format, this window has only two options:

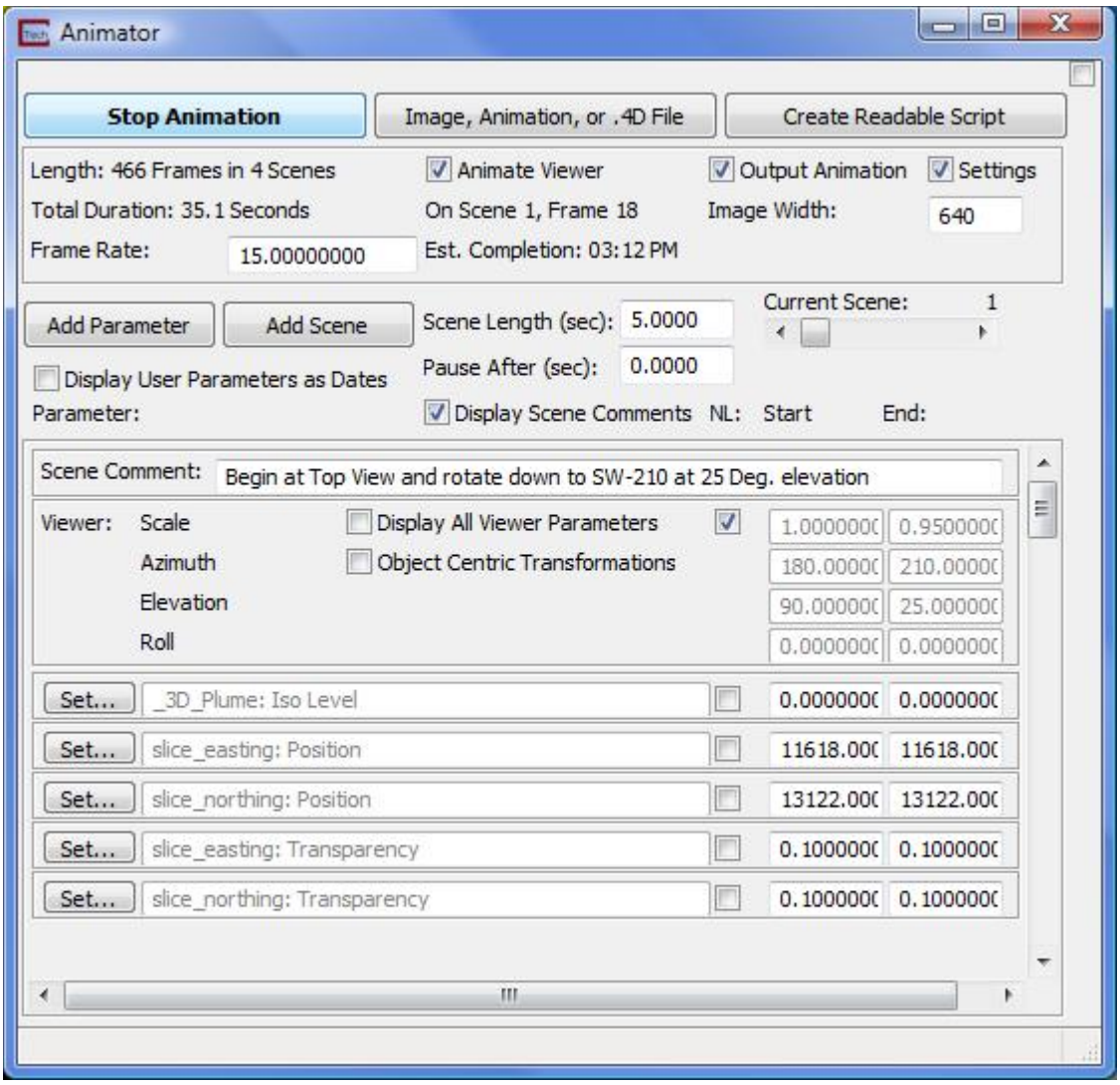


While the animator is running it will add another module to your application automatically. In the Applications window you will now see:



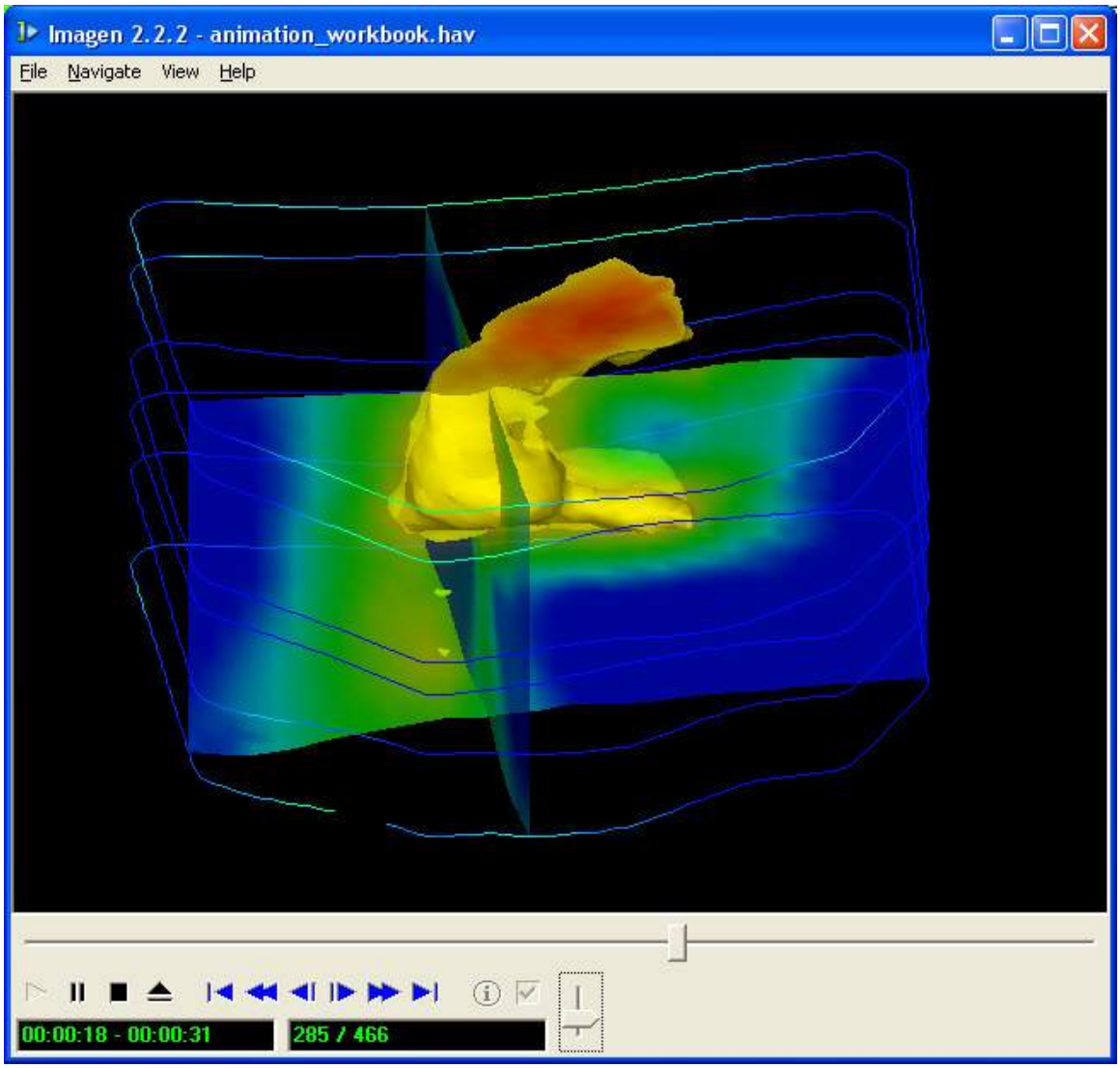
This special module is automatically saving images and building them into an animation in HAV format. Because it does not need to save all the image frames in advance it is both quicker and more efficient (disk space) than other approaches.

While the Animator is running it will tell you its current status (Scene and Frame) and an estimated time of completion as shown below:



Since we chose the HAV format we need to have the HAV player, **Imagen**, installed to view our animation. The installer is included with C Tech software. If you haven't installed it, it is on the EVS/MVS CD or can be downloaded from http://www.gromada.com.

If you have Imagen installed, when the animation is complete it should open up in Imagen and look like:



If you create an AVI file or MPG file, you will have more options for compression (codecs) and animation players (though Imagen is a good player for all types).

This application, with the final settings for the animator, has been saved as ctech\Applications\Workbooks\animation_workbook_final.v.

Animation Suggestions

There are many other animations which we could have produced with this same application or a slight modification of it. Below are a few suggestions:

1. Create a 4DIM model by specifying a .4d file instead of an .png file.
2. We never modified the parameters of Explode_and_Scale. This data had multiple geologic layers which we chose not to explode.
3. Try labeled isolines on one or more of the slice planes.

4. Add the string_format and Titles modules and report the plume level or slice positions.

Workbook 9: C Tech Geostatistics

- **Geostatistics Workbook Overview**
- **Kriging**
- **Geostatistical Confidence**
- **Uncertainty**
- **Min-Max Plume**
- **Three-Dimensional Plume Variations**
- **Min-Max Algorithm**
- **volumetrics**
- **Geostatistics Conclusion**
- **Kriging References**

- **Workbook 1 Fundamentals and Two-Dimensional Kriging:**
- **Workbook 2 DrillGuide© Analytically Guided Site Assessment:**
- **Workbook 3 Creating A Geologic hierarchy:**
- **Workbook 4 Three-Dimensional Geologic Modeling:**
- **Workbook 5 Three-Dimensional Kriging:**
- **Workbook 6 Three-Dimensional Fence Diagrams:**
- **Workbook 7 Visualizing Groundwater Modeling Results:**
- **Workbook 8 Animation Using EVS-PRO & MVS:**
- **Workbook 9 Geostatistics in EVS:**
- **Workbook 10 Finite Difference Gridding:**
- **Workbook 11 Advanced Geologic Modeling Concepts:**
- Workbook 12 Controlling Geologic Hierarchy:
- **Visualization Fundamentals**
- **C Tech Main Help**

Geostatistics Workbook Overview

In this workbook, we provide an understanding of the philosophy and implementation of parameter estimation and geostatistics in EVS. EVS's capabilities to grid and interpolate sparse measured data in three dimensions are unparalleled. EVS performs most of its interpolation using an accurate and geostatistically defensible process called Kriging. Kriging is a mathematical process recognized by the EPA as the best and standard means for interpolation and extrapolation of measured data. EVS provides a user-friendly expert system to drive its Kriging modules, lifting the burden of determining optimal semivariogram parameters from the user. With EVS, the user can rely on expert system calculated default values to provide quality answers in minimal time.

Kriging is the only data estimation method that also provides statistical measures of goodness. When data is analyzed in EVS, the user is also provided statistical confidence and an additional parameter that we call uncertainty. These additional statistical measures are extremely useful in guiding future site investigation efforts. Our experience is that by using maximum uncertainty to guide site investigations, a 30% reduction (or more) in sampling locations can be realized for an equivalent quality of assessment.

As a user defined alternative to confidence and uncertainty, EVS will provide statistical bounds on the parameter estimate (This feature is only available in EVS PRO). In other words, EVS will determine the nominal, minimum, and maximum estimated distribution based on a user specified confidence level. With EVS, you can now directly answer the question: *With my limited measured data, to an 80% confidence level, what is the largest and smallest plume I can expect?*

Kriging

Kriging is a weighted moving average interpolation (extrapolation) method that minimizes the estimated variance of a predicted point (node) with the weighted average of its neighbors. The weighting factors and the variance are calculated using a semivariogram model that describes the differences versus distance for pairs of samples in the input dataset. Determining an optimal semivariogram model is the important first step in producing a defensible kriged estimate. With other software, the burden of fitting the semivariogram is left to the user. In EVS, this task is automated with an expert system.

EVS's kriging is based on sound mathematical and statistical concepts. The U.S. Environmental Protection Agency (EPA) developed a two-dimensional kriging software package that is called Geo-EAS. Geo-EAS version 1.1 was released in September 1988.

However, EVS is not limited to two-dimensional estimation. In fact every object in EVS is truly three-dimensional and all analysis and visualization techniques can be combined in a single display.

Two Dimensional Kriging

EVS gridding options for two-dimensional estimation include: rectilinear grids with uniform spacing in x & y directions; convex hull bounded gridding; and adaptive gridding that automatically refines gridding in the cell (s) surrounding measured samples to ensure that the interpolated results and contours accurately honor measured sample data.

Three Dimensional Kriging

EVS provides a full spectrum of three-dimensional gridding options. These include: rectilinear grids with uniform spacing in x, y, & z directions; rectilinear grids with uniform spacing in x & y directions with z spacing determined by geologic layers; finite difference type grids with variable spacing in x & y directions and z spacing determined by geologic layers; convex hull bounded gridding with z spacing determined by geologic layers; and adaptive gridding that automatically refines gridding in the cell (s)

surrounding measured samples to ensure that the interpolated results and isosurfaces accurately honor measured sample data.

Geostatistical Confidence

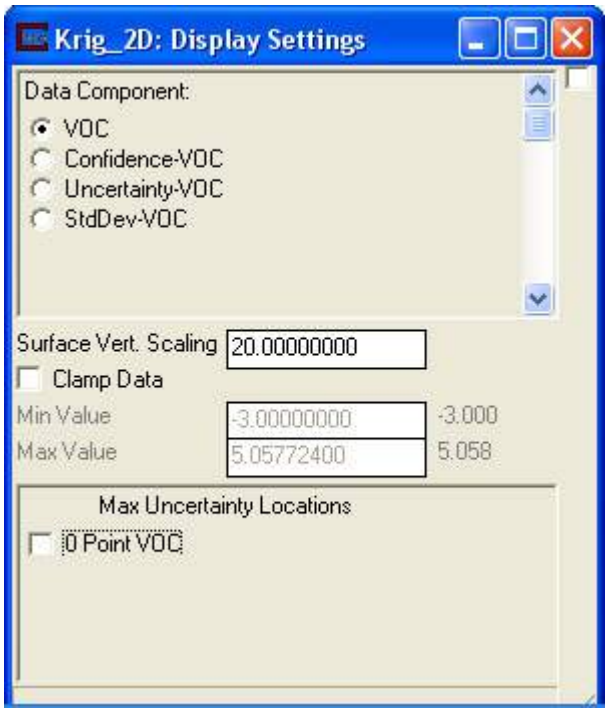
EVS computes the standard deviation for each estimated point. The standard deviation is used to compute the confidence and uncertainty. The standard deviation is proportional to the square root of the sill in the variogram, since the standard deviation is computed as the square root of the variance (the second data component in MVS's KT3D module) and the variance is directly proportional to the sill.

Delete the existing application and create a simple application with Krig_2D (red port) connected to the Viewer.

Select **initial_soil_investigation_full_site.apdv**. Select **Accept All Current Values** on Krig_2D and when it runs your Kriging Parameters should match:

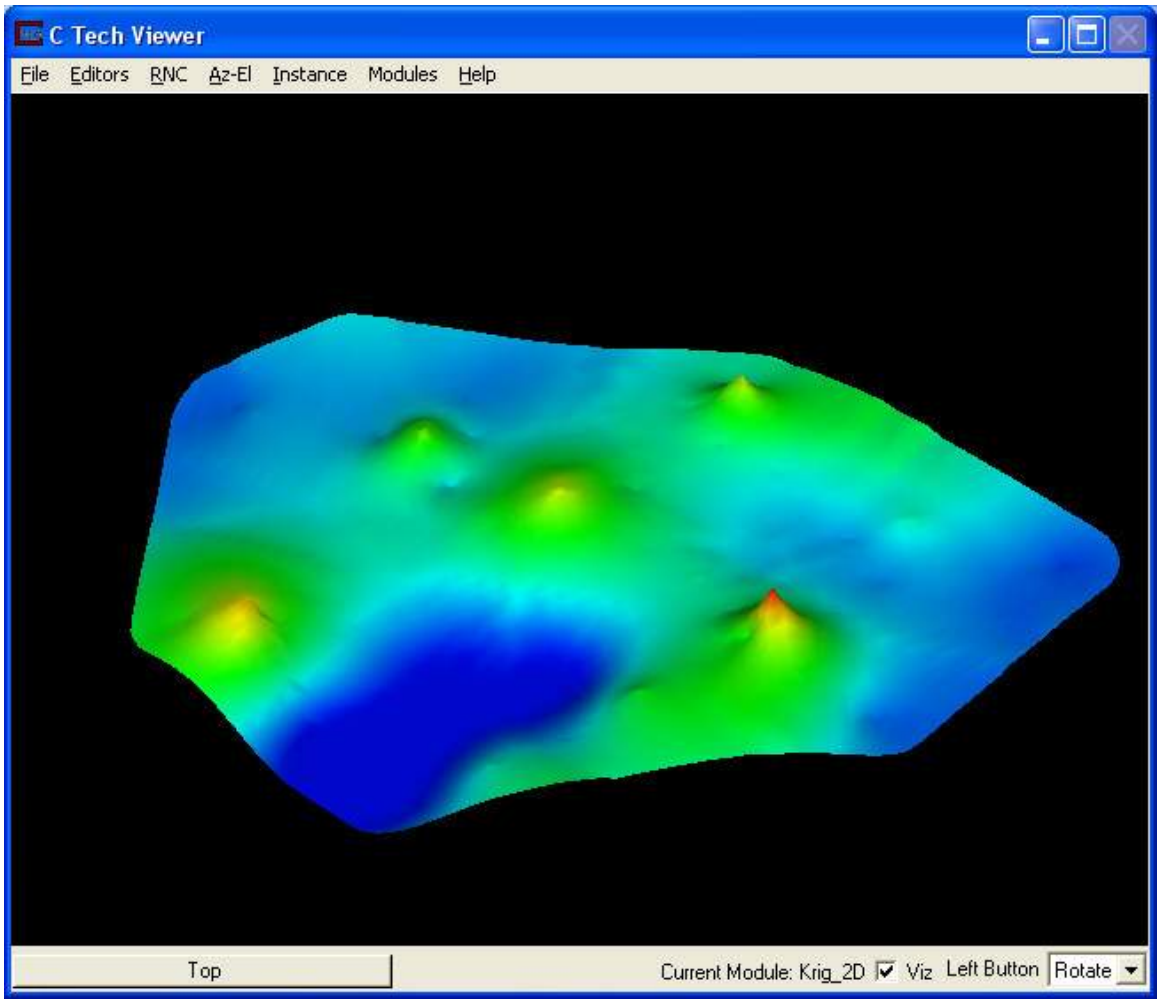
Note that the *Confidence BoundFactor for Statistics* value is set to 10. That value will be used to determine our confidence calculations (which we will discuss very soon).

Now close the Kriging Parameters and bring up the Display Settings panel. Let's set the *Surface Vert.Scaling* to 20.0 and turn off the Max Uncertainty Location (since we don't want to deal with that for this topic).

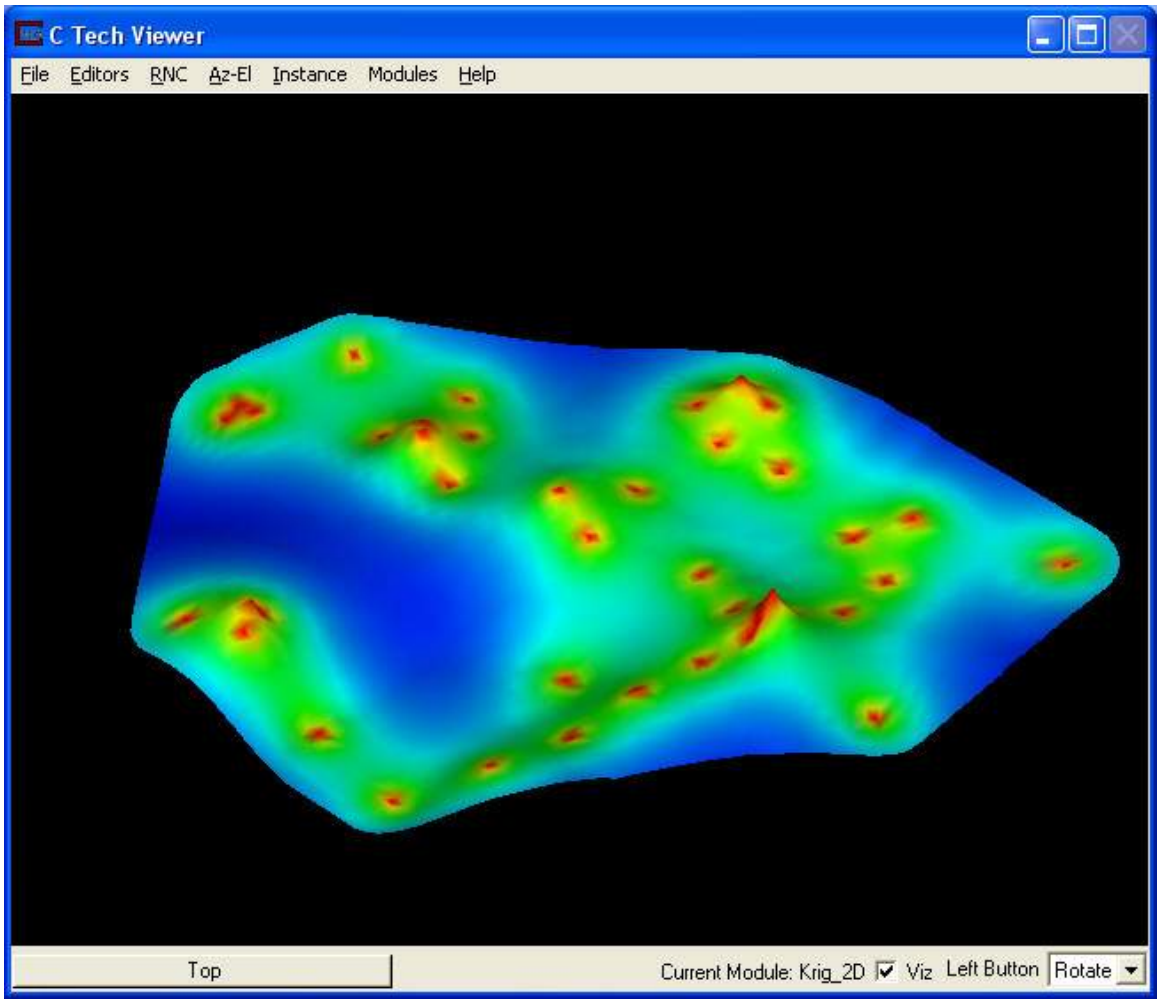


Let's also bring up the Az-El panel and set the view so that Scale = 1.10; Elevation = 45; and Azimuth = 210. Your view should now match the picture below:

Remember from Workbook 1, that this surface is colored according to concentration, and also has surface elevations (topology) which is the log of concentration times the data scaling value of 20.0.

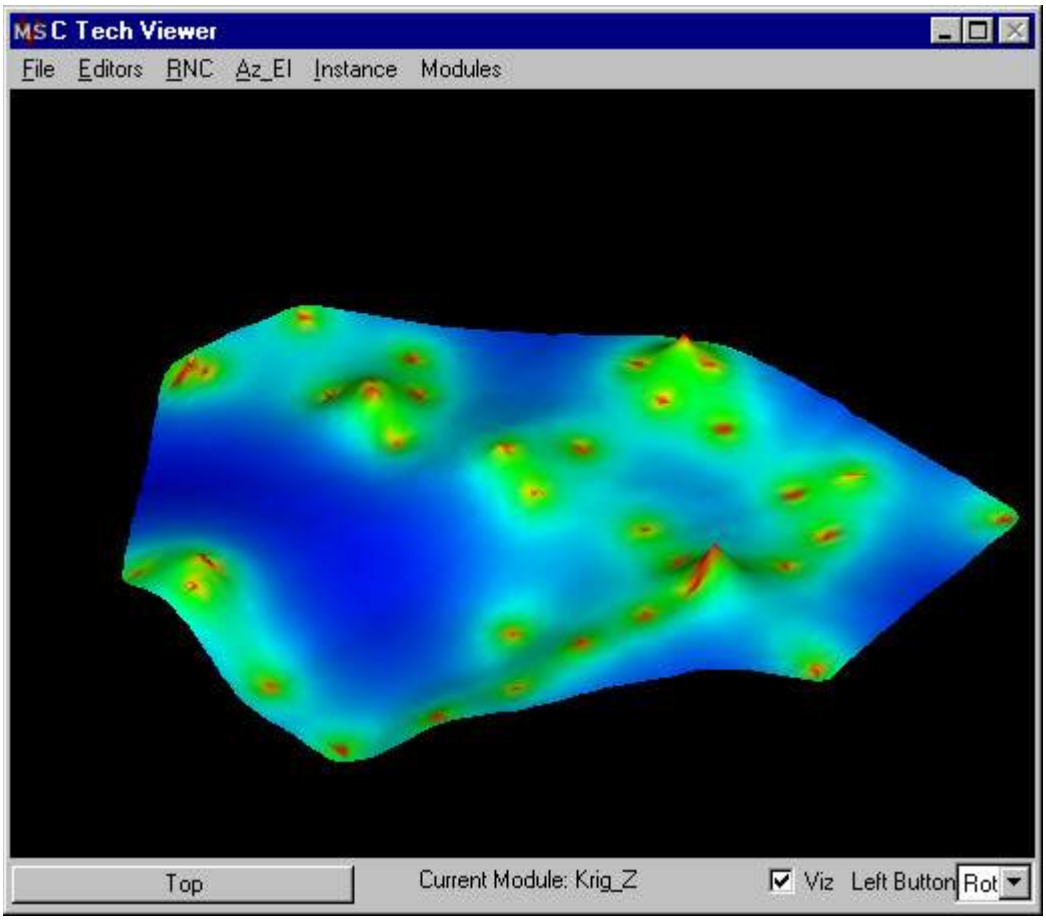


We will now select Confidence as our Data Component. This will recolor the surface, but leave the elevations unchanged. From a single picture we will be able to see confidence levels (by color) and concentration levels (by elevation).

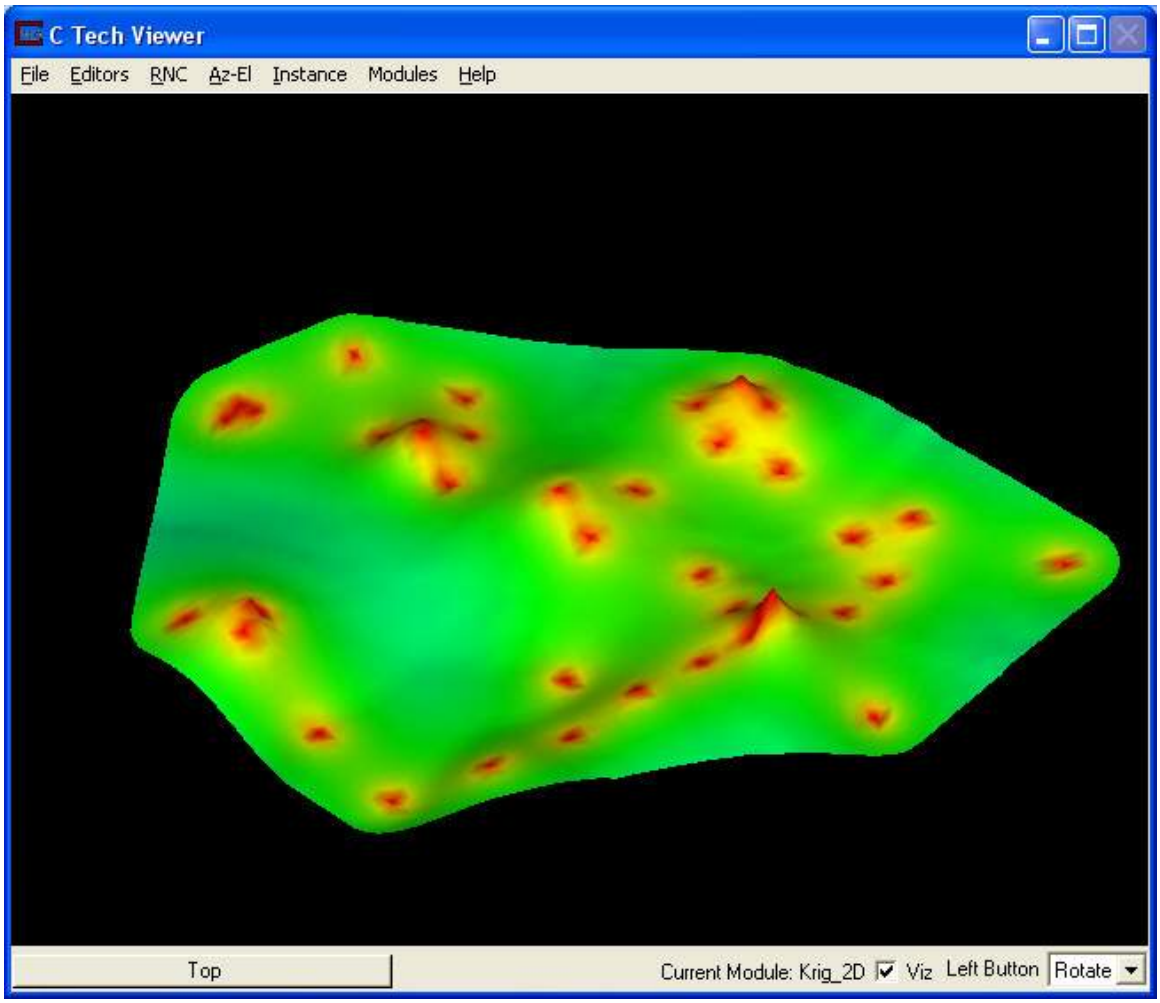


The actual Min(imum) confidence is 35.9.

Notice that the picture below has a lot of red spots. Each of these spots corresponds to a measured sample location. At every measured sample, the confidence in the estimate is 100%.



With Confidence still as our Data Component, check the toggle called *ClampData*. Then, set the Min Value to 0 and the Max Value to 100. This will map the blue (minimum) color to 0.0 instead of our actual data minimum, and 100 instead of our actual max (although our actual max is approximately 100). The picture will have no blue areas because our actual minimum is 35.916. This will give the following picture.

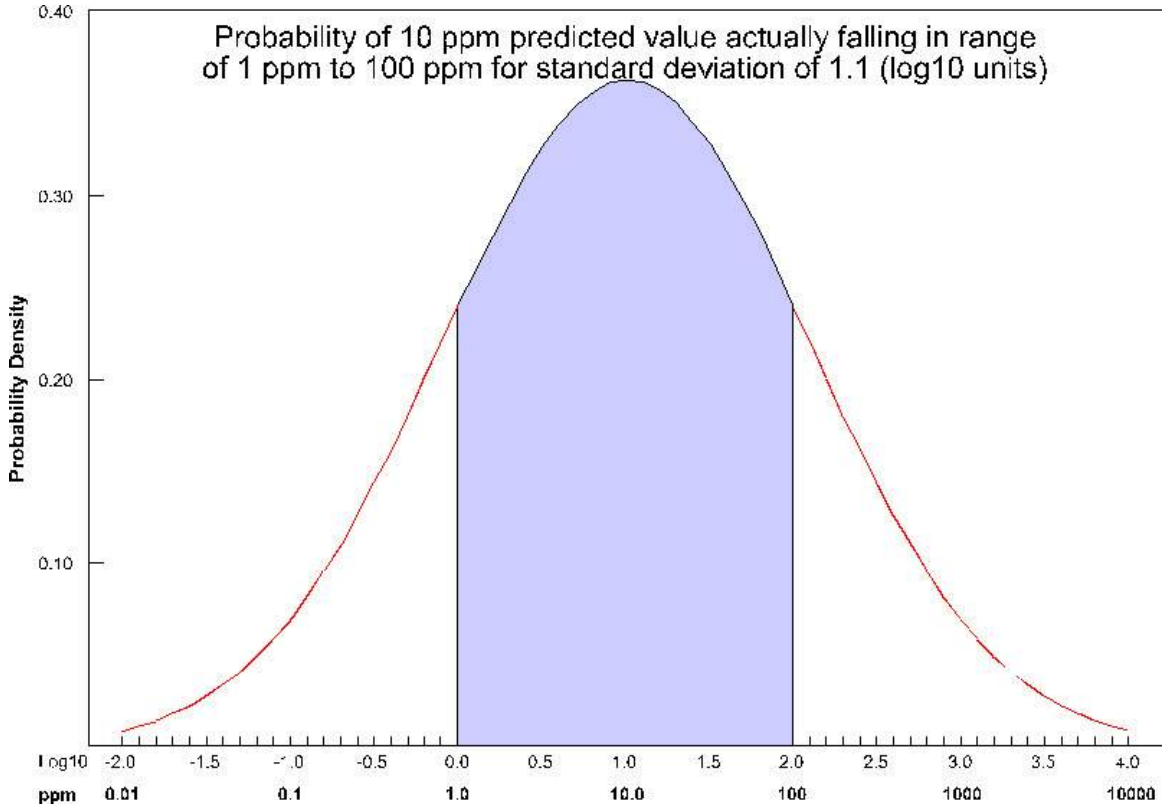


But what do we mean by Confidence? In EVS, the confidence is calculated in response to a question. For this model, the question is: What is the confidence that the predicted concentration will be within a factor of 10 of the actual concentration?

Why a "factor of ten"? Because we are working on a log10 scale, EVS took the log10 of the Conf Bound value (the value was 10, so the log is 1.0). It then compares the log concentration values and a corresponding standard deviation that was calculated for every node in our domain. For log concentrations, one unit is a factor of ten, therefore we are asking what is the probability that we will be within one unit. If you had changed the Conf Bound to 2.0, the questions would have been: What is the confidence that the predicted concentration will be within a factor of 2 of the actual concentration?

The actual calculation to determine confidence requires the standard deviation of the estimate at a node, and the Conf Bound value. The figure below shows the confidence (as the shaded area under the "bell" curve) for a Conf Bound of 10 at a node where the predicted concentration was 10 ppm (1.0 log concentration) and the standard deviation for this point was 1.1 (in

log10 units). For this example, the confidence would be ~64%, which means that 64% of the time, the value would lie in the shaded region.

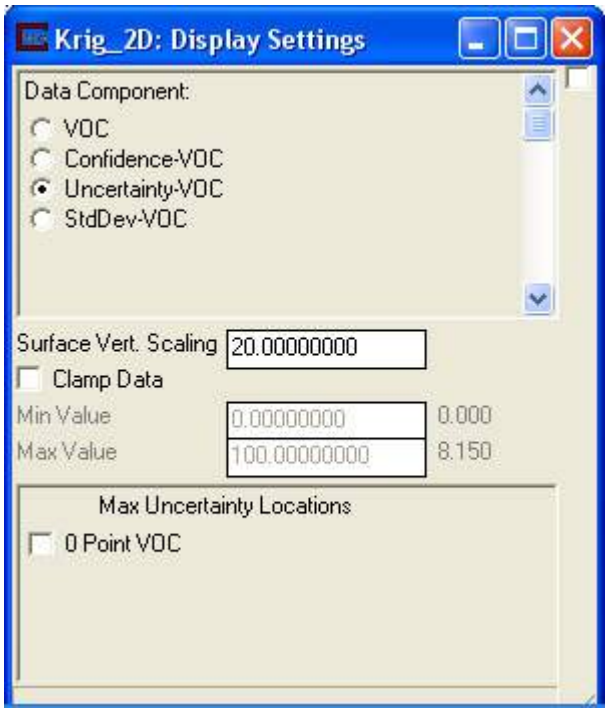


Uncertainty

At first glance, confidence seems to be a reasonable measure of site assessment quality. We know that if the confidence is high, we can be assured of the reasonableness of the predicted values. You might be tempted to collect samples everywhere that the confidence was low, and if you did, your site would be well characterized.

But, there is a better, more cost-effective way. Instead of focusing on every place where confidence was low, we could focus on only those locations where there was low confidence and where the predicted concentration was reasonably high. We could try to see these locations by looking at the confidence colored plot for high elevation areas...or, we could use the EVS measure called uncertainty. In EVS, uncertainty is high where concentrations are predicted to be relatively high (above the Clip Min), but the confidence in that prediction is low. If the goal is to find the contamination, using uncertainty will allow for more rapid, cost effective site assessment.

Let's look at the Uncertainty plot. Choose Uncertainty as the Data Component and turn off the *Clamp Data* toggle. Note that it ranges from 0.00 to 8.15.

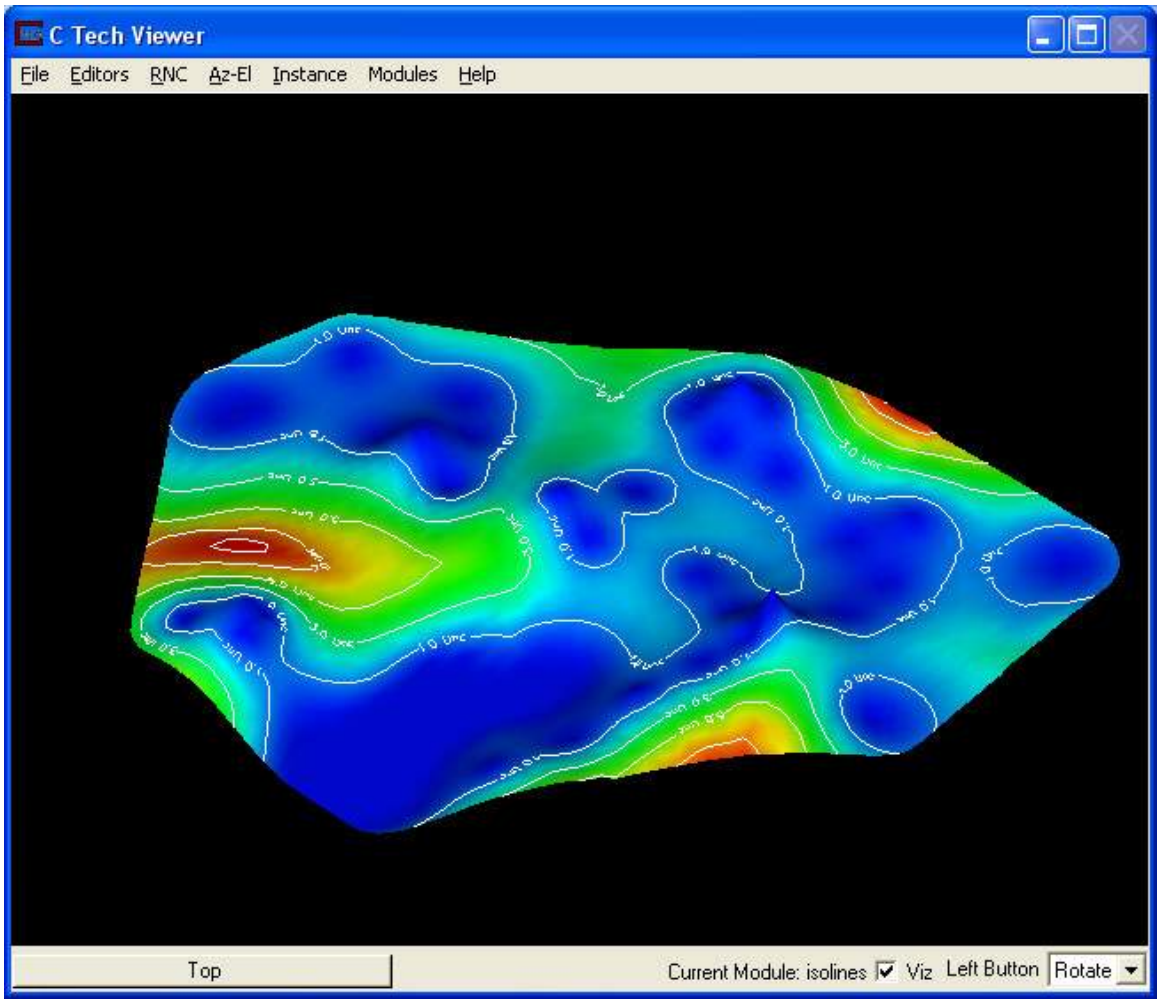


When we ran this calculation, the EVS Status Window reported the precise coordinate of the maximum uncertainty. That location is a very appropriate next sampling site.

VOC : Max Uncertainty at node 388: x= 11512.49 y= 12921.77

Concentration= 0.223 Uncertainty= 8.15

Note the bright red spot (dark spot in black & white) on the left side of the domain. This area is far from any sample locations (regions of high confidence) and it is on the side of our topological hill (meaning that the concentration is well above our minimum. To make the region more obvious in the printed book, we have added labeled isolines (This feature is only available in EVS PRO) at uncertainty value of 1, 3, 5, 7, and 8.

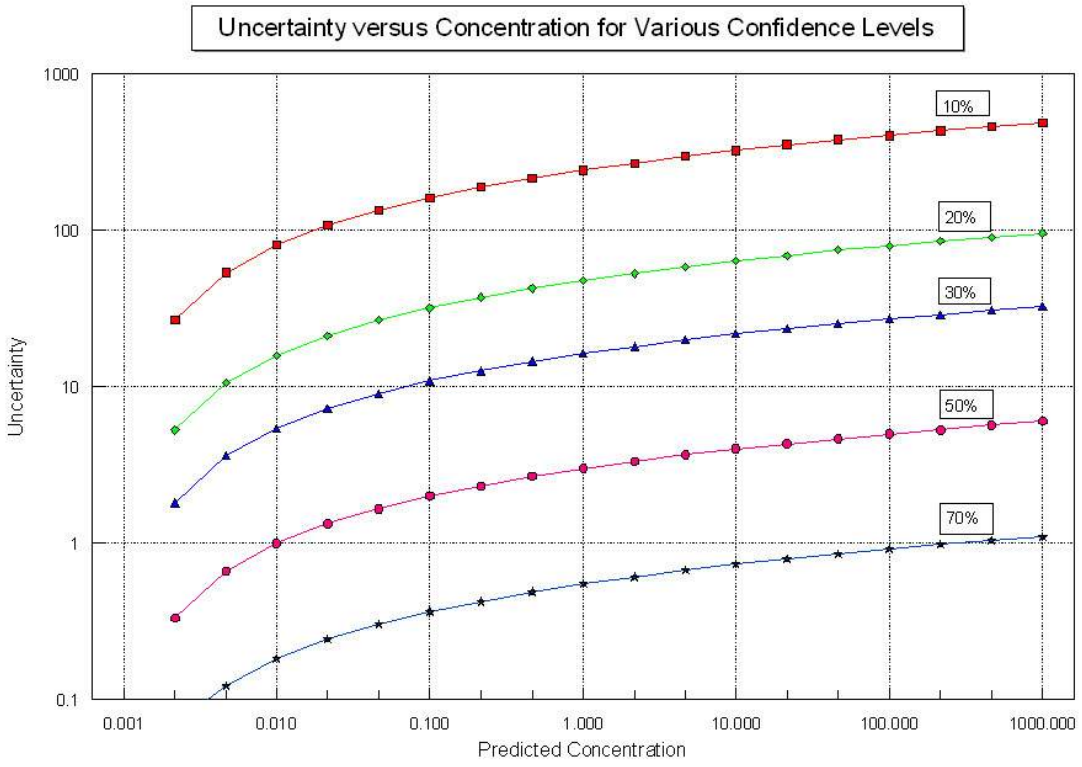


Uncertainty does not have physical units, rather it is a dimensionless parameter for which higher values are bad and lower values are good. High uncertainty means a high probability of poorly characterized contamination. Low uncertainty means a low probability of uncharacterized contamination. The function used to compute uncertainty has been optimized to minimize the number of new sampling locations required to lower the maximum uncertainty anywhere in the model domain to a specific low value. The confidence levels and corresponding uncertainties are data dependent. If the measured data has regions of high concentration gradient, nearby areas away from sample locations will have lower confidence and higher uncertainty.

The absolute magnitude of uncertainty values is also dependent on three key user specified values; the confidence interval, and the display clipping minimum and maximum. The "clip min" establishes the *floor* for concentration values. Values below this level are considered unimportant (or more specifically, they are set to that level). The "clip max" establishes the *ceiling* for concentration values. Values above this level are clipped to this value. Setting the clip max at the concentration value of most interest, such as the toxic level for a plume boundary, will cause the selection of new sampling locations to converge most rapidly on defining the plume boundary.

This will however sacrifice the accurate characterization of the most contaminated regions in the domain. This limitation is generally acceptable if the primary goal is determination of the extent (rather than specific distribution) of contamination. A superior approach is to use the DrillGuide© technology documented in the previous workbooks to target the characterization of a specific concentration of interest.

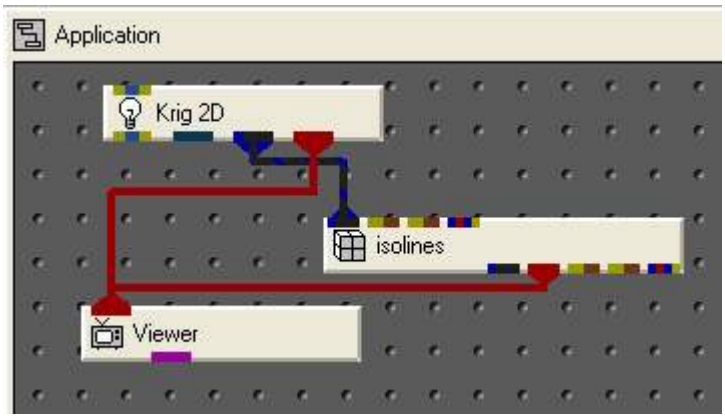
The next figure shows the calculated uncertainty as a function of predicted concentration for 5 different confidence levels. This plot assumed a *clip min* of .001, and *clip max* of 1000, and a *confidence interval* of 10. During the early stages of site characterization uncertainty levels can exceed 100.



Min-Max Plume

Sometimes confidence and uncertainty just won't answer your (or your customer's) question. Just how big (or how small) might the contaminant plume be? EVS displays concentration, but what are the limits? The Min-Max Plume option directly answers those questions.

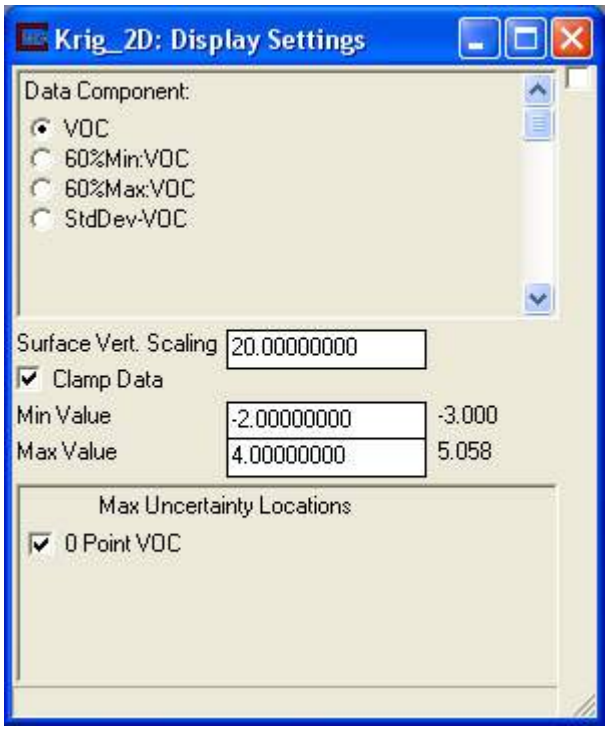
Create the network below (start with a new copy of Krig_2D).



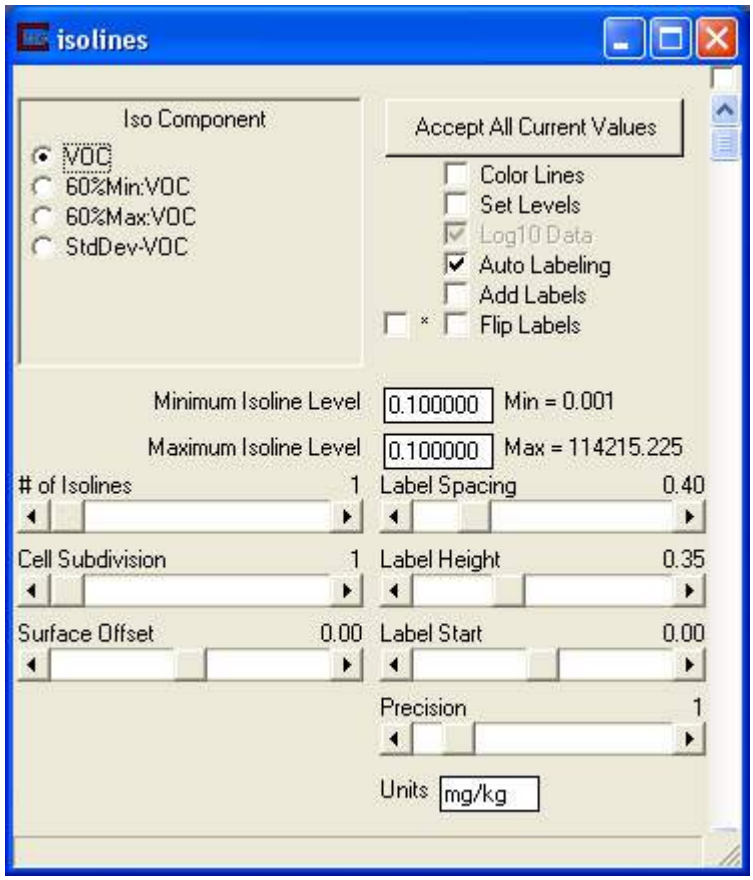
Open up the Kriging Parameters and choose Min-Max Plume for the Statistics Options (This feature is only available in EVS PRO). Note that the *Confidencefor Min/Max Plume (percent)* parameter is set to 60%.

Choose **initial_soil_investigation_full_site.apdv** again as our data file and after it has run, close Kriging Parameters and open up the Display Settings.

We want to set a clamped range for outputting the min, nominal and max plumes. The limits of all three will be slightly different so we should choose values that will be compatible with all. We will set the Clamp toggle **on** and the Min val to -2 (equiv. to 0.01 ppm) and the Max val to 4.0 (equiv. to 10,000 ppm). Also set the Data Scaling to 20. Your parameters should match:

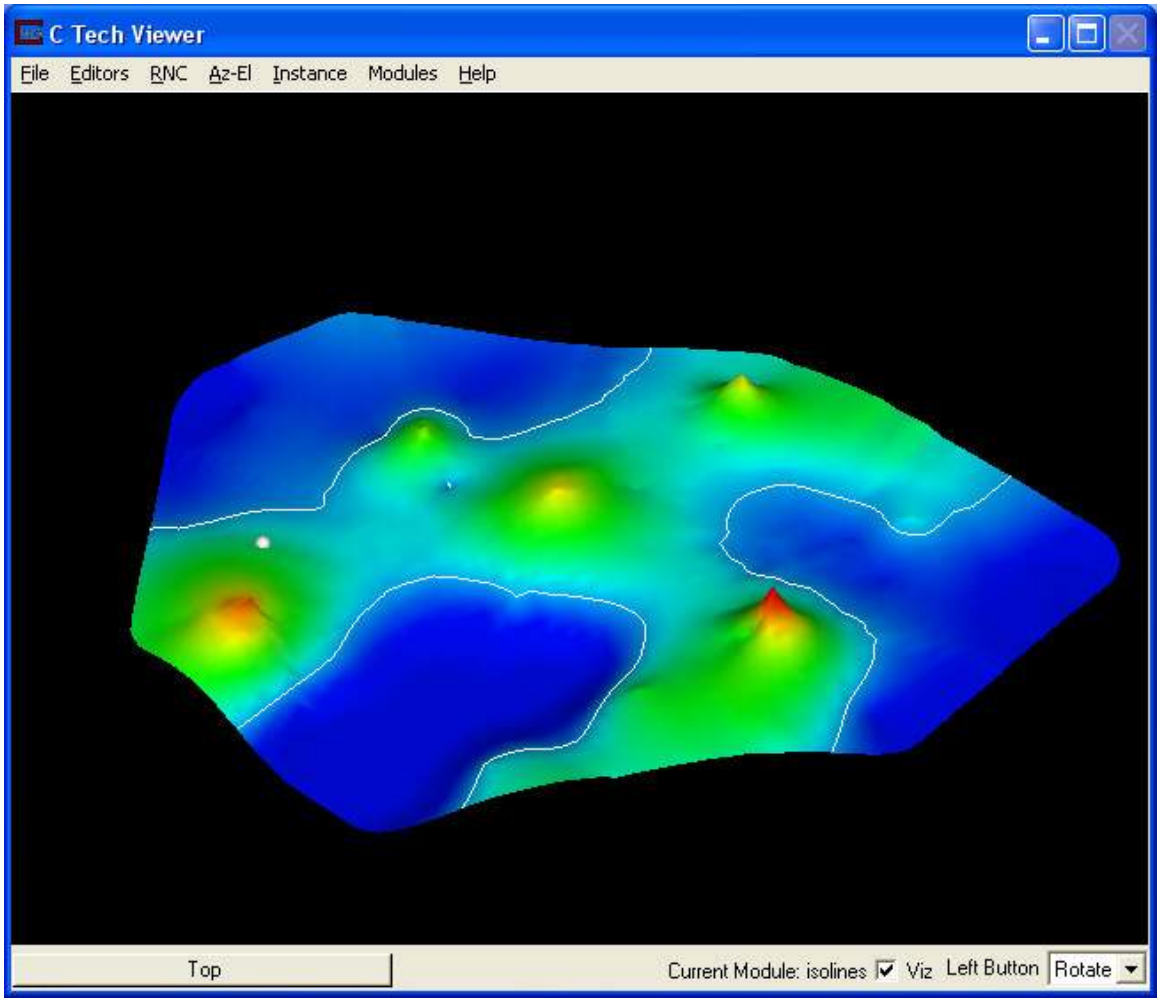


Set the parameters of isolines to match the figure below and choose accept.



Bring up the Az-El panel and set the view so that Scale = 1.10; Elevation = 45; and Azimuth = 210. Your view should now be:

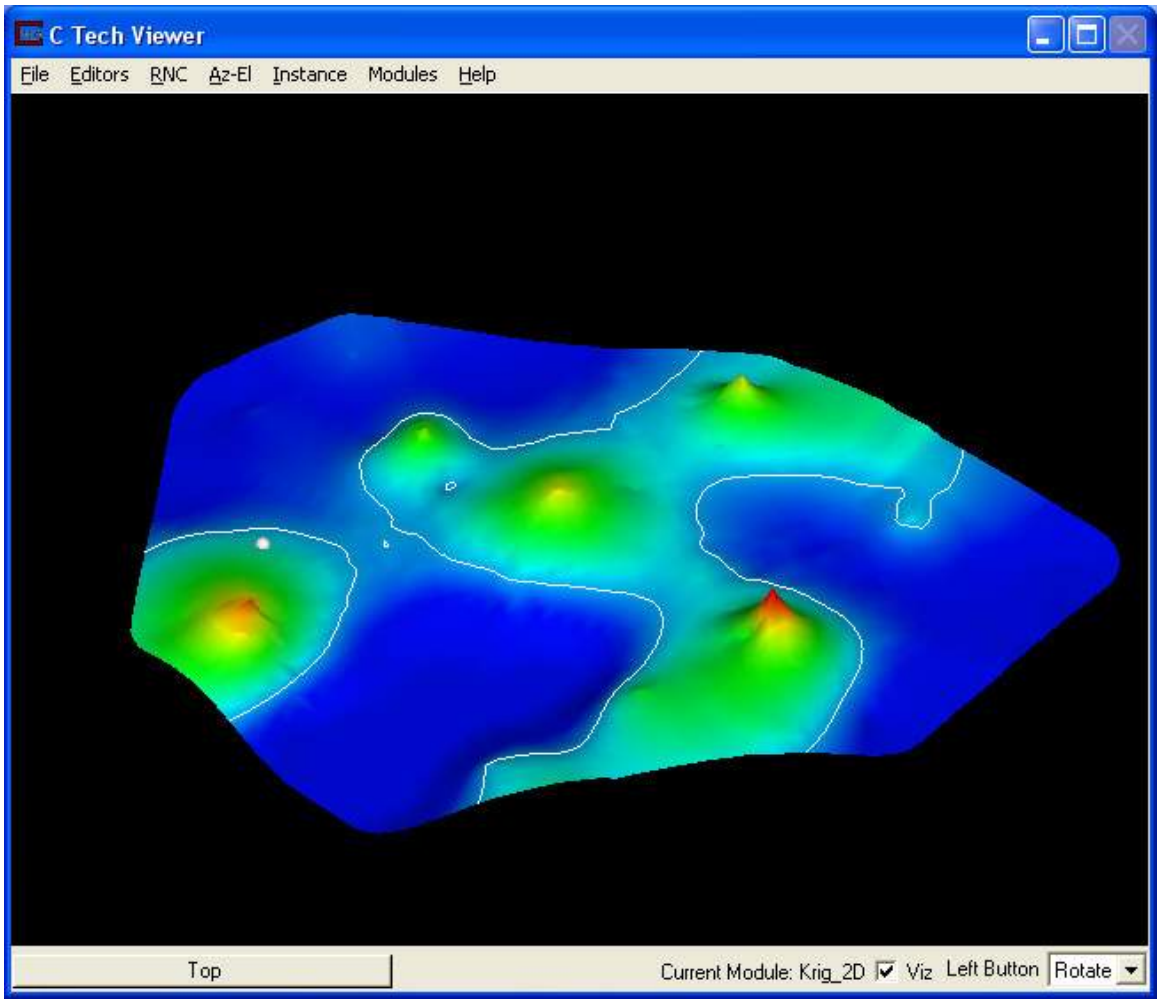
Nominal Plume:



The white outline is the nominal plume at 0.1 ppm (log value of -1.0). What does nominal mean? Nominal is the distribution where 50% of the time the predicted values are higher and 50% of the time they are lower. That means that our confidence in this plume size is never lower than 50%. Be careful, this confidence has a different meaning than the confidence discussed in the early section.

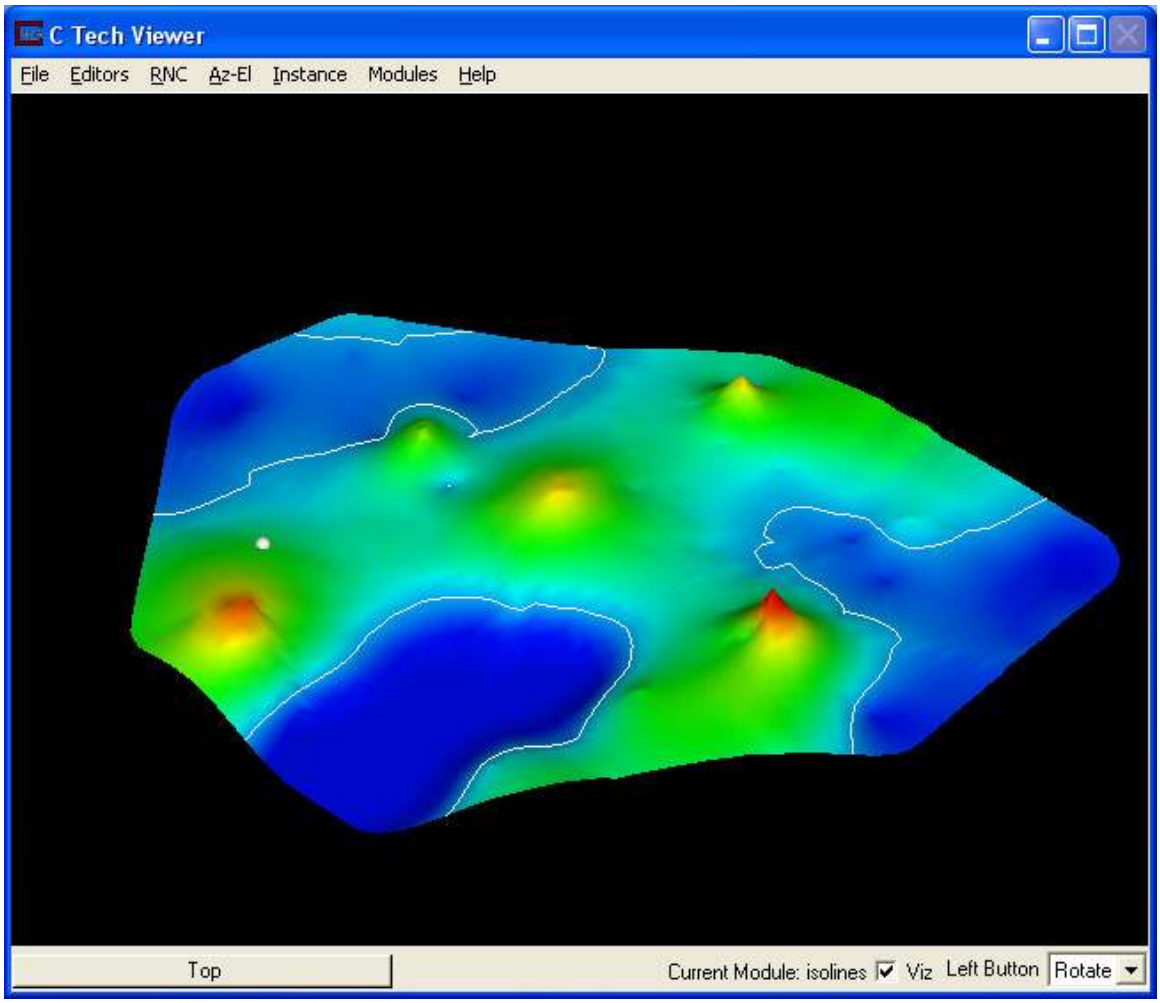
Now let's see what the minimum size plume is for our 60% confidence level. To do this we need to change both the Data Component in *Krig_2D:Display Settings* and the *Iso Component* in isolines to 60%Min:VOC (remember to hit accept in isolines also) Note a marked reduction in the 0.1 ppm plume versus the nominal plume.

Minimum Plume:



Similarly, the maximum plume (for a 60% confidence) is:

Maximum Plume:



What would have happened if we had set the Confidence level higher (or lower)? The Confidence parameter cannot be set lower that 50%, because by default, the nominal distribution has a 50% confidence. If it is set at 50%, the Min Plume will be the same as the Max Plume. Values below 50% don't make sense.

Similarly, a value of 100% is impossible. We can never be 100% sure of estimates. The maximum allowable value is 99.99% and we don't recommend it. Unless your site is extremely well characterized, there will be significant variation between min and max plumes at Confidence levels of 80 or 90%.

Everything we have covered on geostatistics has used two-dimensional kriging. Is that all EVS can do?.....NO!

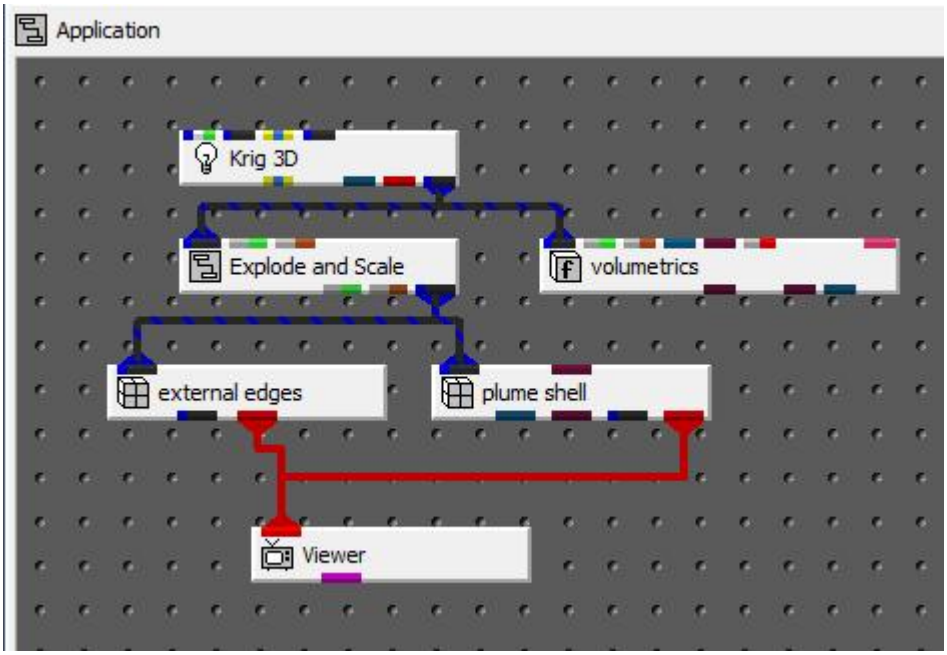
Three-Dimensional Plume Variations

When is a site sufficiently characterized? This is a difficult question. Sometimes the answer is simple...when the money runs out. The real answer obviously depends on risk assessments, remediation plans and other social and political factors. This Min-Max Plume technology in EVS (This feature is only available in EVS-Pro and MVS) offers a quantitative and justifiable method of deciding when enough is enough.

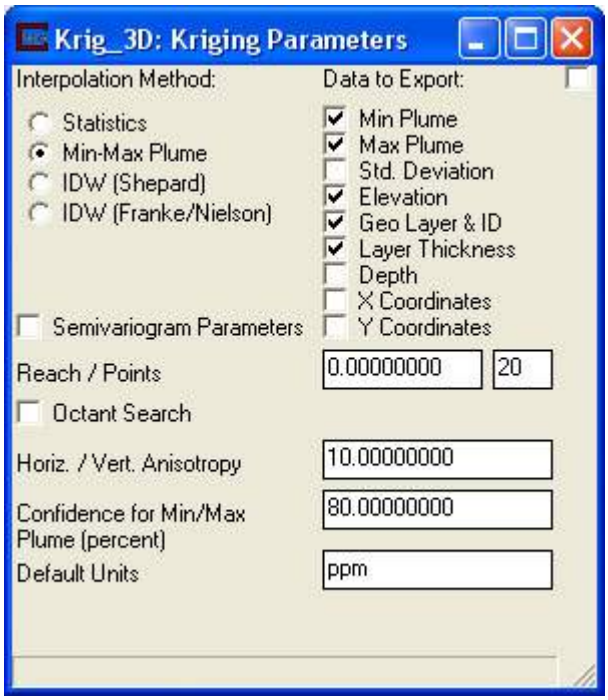
The first step should always be a risk assessment. Depending on the toxicity of the contaminants, the location of the plume relative to the water table, wells etc., and many other factors, a confidence level should be selected. Reasonable values might range from 60 to 95%.

We can use that confidence level to compute the volumetric variation between the minimum and maximum three-dimensional plume that is predicted with the available site data. Based on the variations and/or the predicted minimum and maximum values, site assessment quality can be directly assessed.

Let's look at an example of an under-characterized site. Construct the following network and choose the file initial_soil_investigation_subsite.apdv for Krig_3D.



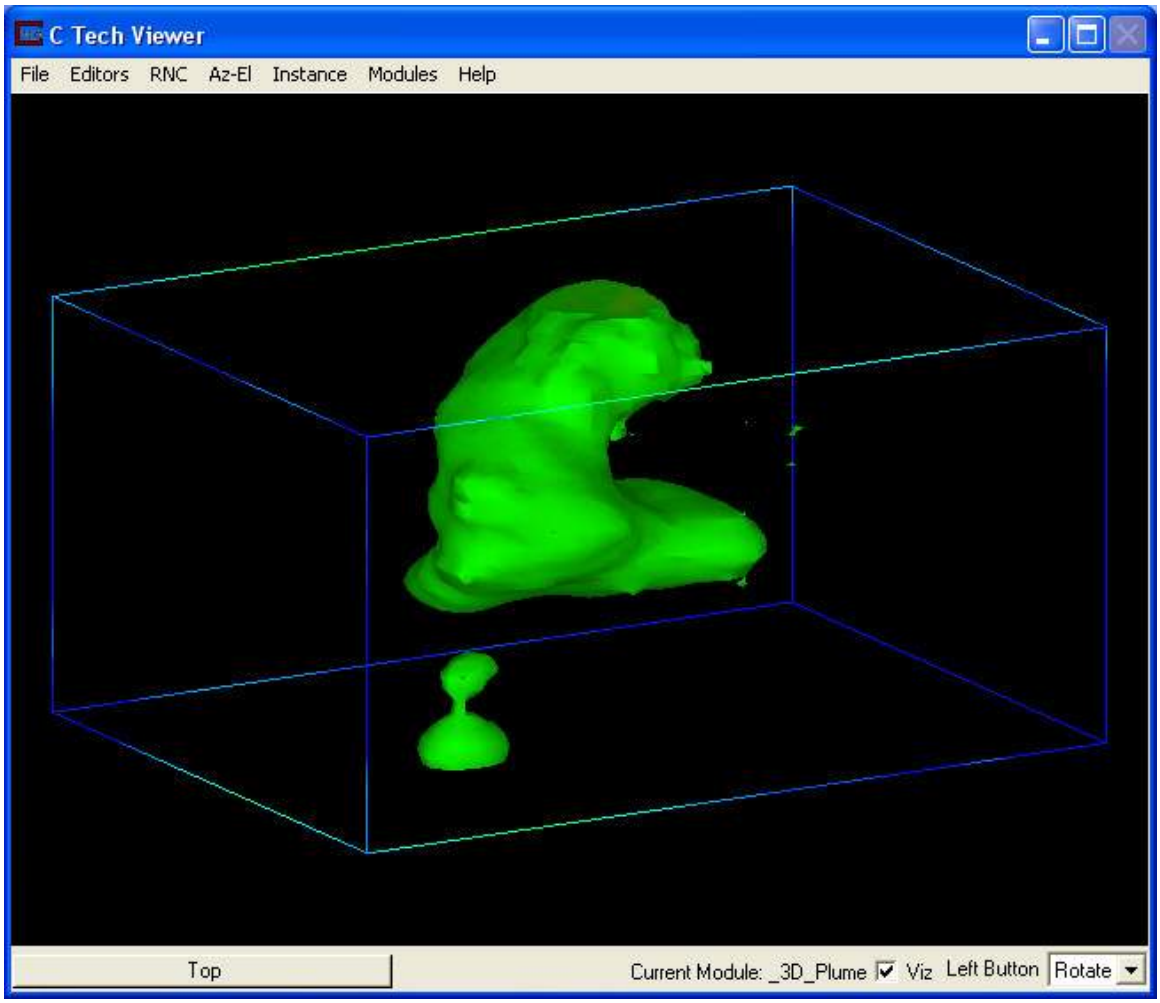
Bring up the Kriging Parameters and adjust it to match the figure below. Make sure to select Min-Max Plume and set the Confidence to 80%.



Close Kriging Parameters and click on **Accept All Current Values** to run Krig_3D.

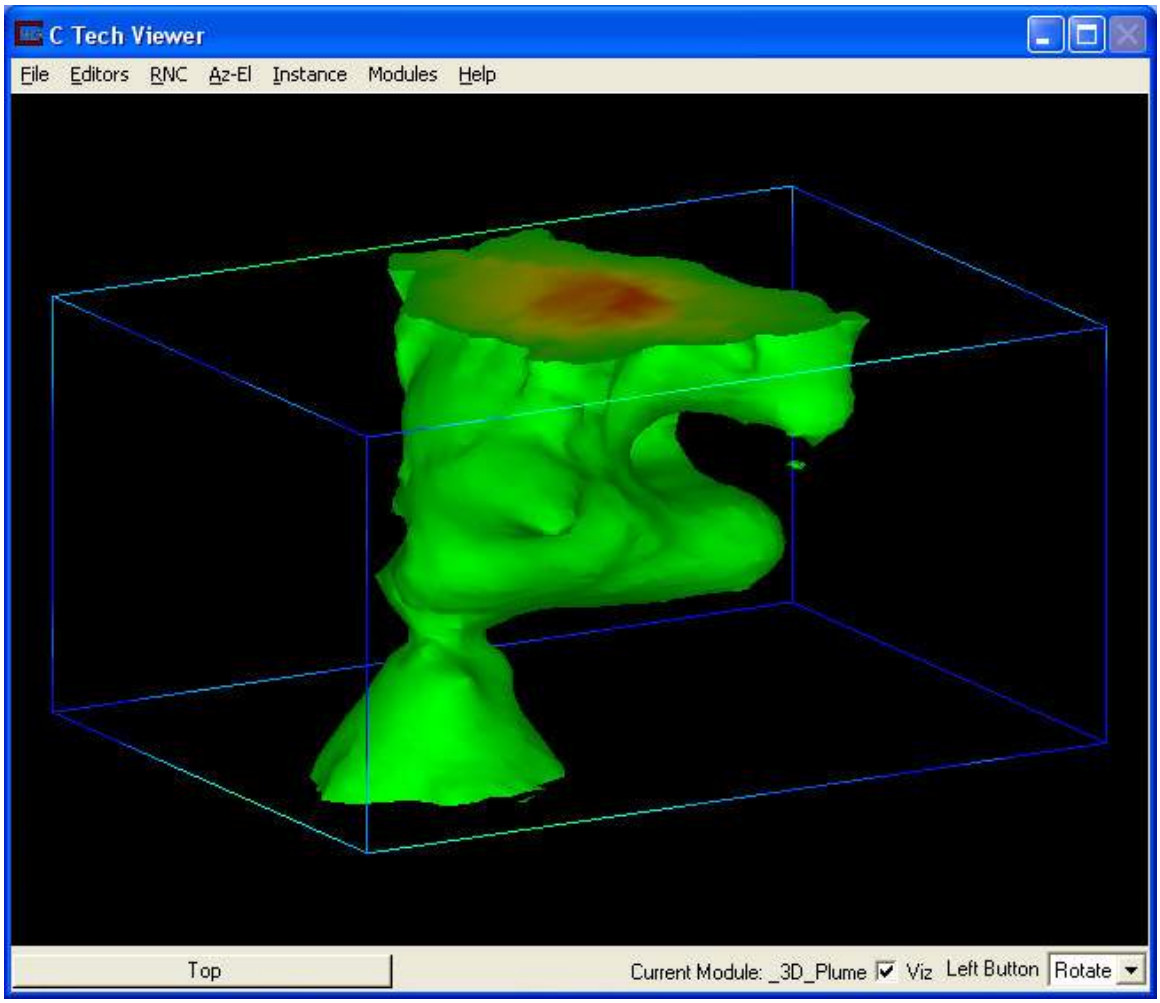
Bring up the Az-El panel and set Scale = 1.00, Elevation = 15, Azimuth = 210. Open the plume_shell panel and set the subsetting level to 1.0 (or Exponentiated Level to 10 ppm). Also set both the Iso Component and Map Component to 80%Min:TOTHC. For the Map Component, multiple data components can be selected for output. Only the first component checked will be used for coloring the output. Make sure that only 80%Min:TOTHC is selected. When you're done, your minimum plume should look like the figure below.

Minimum Plume:



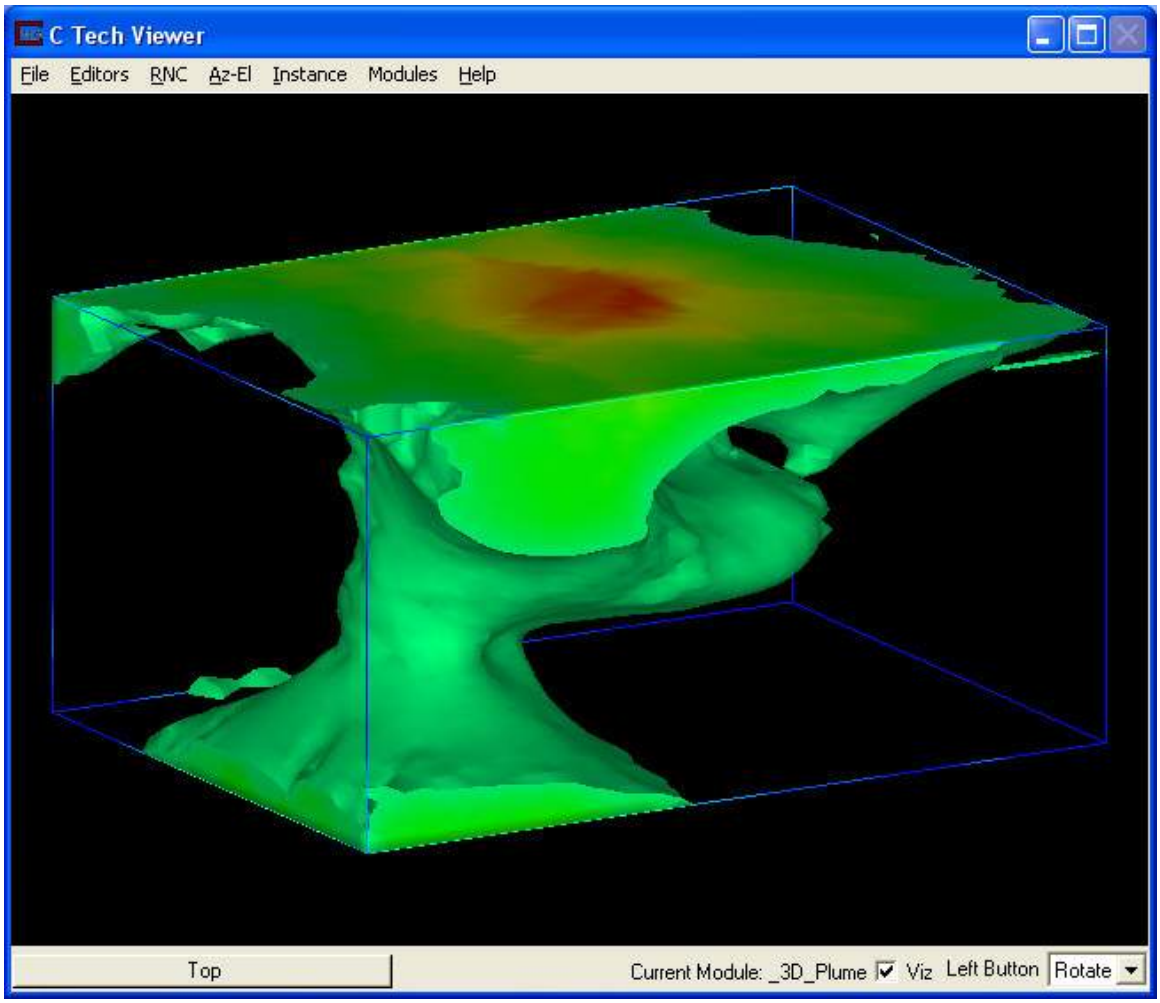
If you set the plume_shell Iso Component and Map Component both to TOTHC, your nominal plume should look like this. Note that it is both dramatically bigger and has much higher concentrations in the center at the top surface. One surprising fact is that both of these plumes honor the measured data samples. The variations are occurring in the regions between samples. Obviously if we had more samples, we would have less variation. The quality of our site assessment is becoming more obvious.

Nominal Plume:



Set both the Iso Component and **the only** Map Component to 80%Max:TOTHC to see the maximum plume. Note again how much higher the concentrations are at the boundaries of the domain, and how much bigger this plume is. Again, this plume will honor our measured data. Sometimes this can result in a plume with interior cavities (surrounding low concentration measured samples. We could check for these using either transparency or by cutting through this plume with a **cut** module. We will leave that exercise for the student.

Maximum Plume:



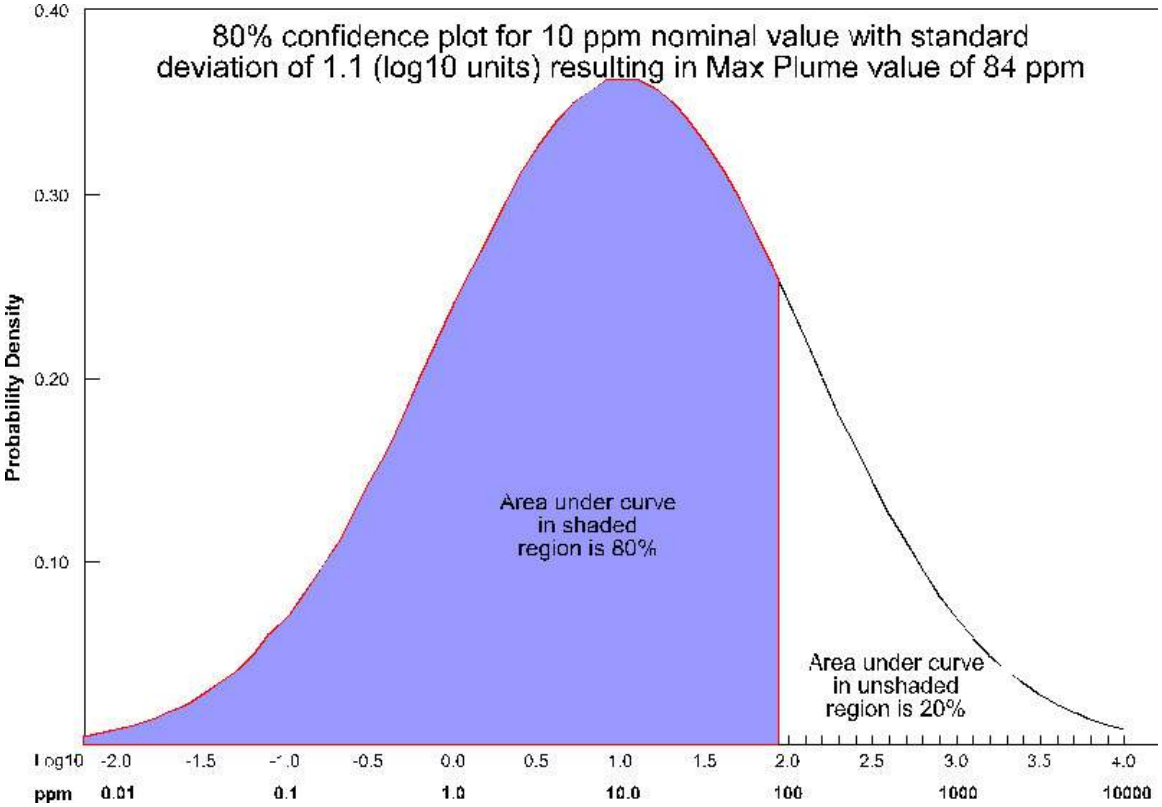
The pictures are useful and can obviously be enhanced with posted data, color legends, axes, overlay drawings, etc. Now let's look at how we determine the Min and Max values at each node.

Note: for the three images above, different Max values result from the kriging, therefore the colors above do not exactly correspond to the same concentrations in all pictures. This could be fixed by clamping the max values (Post-Clip_Max) or using a set_minmax module before plume_shell.

Min-Max Algorithm

EVS determines the Min and Max plumes (This feature is only available in EVS-Pro and MVS) by first calculating the nominal value and associated standard deviation at every node in the model. For the case of Max Plume and 80% confidence, at each node, a maximum value is determined such that 80% of the time, the actual values will fall below the maximum value (for that nominal concentration and standard deviation). This process is shown below pictorially for the case of a nominal value of 10 ppm with a standard deviation of 1.1 (log units). For this case, the maximum value at that node would be ~84 ppm. This process is repeated for every node (tens or hundreds of thousands) in the model.

Note that for this plot, the entire left portion of the bell curve is shaded. If we were assessing the minimum value, it would be the right side. Statistically, we are asking a different type of question than when we calculate confidence for our nominal concentrations.



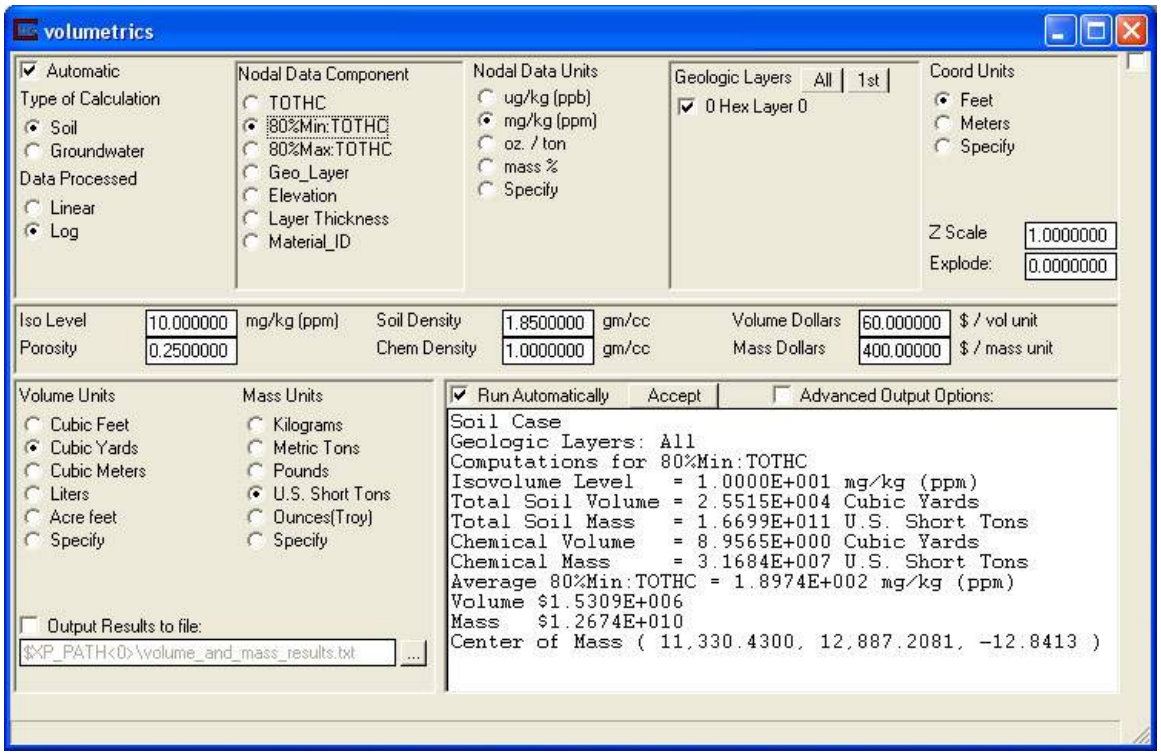
Now let's quantify the variations between our minimum, nominal and maximum plumes.

volumetrics

Remember that our network has a volumetrics module connected to Krig_3D.

volumetrics calculates plume volume and associated masses on any data component in its input. If you open up volumetrics user interface we can quantify the variation in plume volume in seconds.

We will start with the 80%Min:TOTHC (minimum plume). Select it in volumetrics so your window matches the figure below.



Now repeat this process for TOTHC and the 80%Max:TOTHC by selecting the *Nodal DataComponent* in volumetrics.

Each time the volumes are calculated, they are printed on the volumetrics panel.

The Mass $ figures aren't usually applicable for environmental, but could be useful if you had a remediation process whose costs were proportional to extracted contaminant mass. This feature is primarily there to calculate the value of ore bodies (e.g. 42,356 ounces of gold at $200 per ounce). The Volume Dollars are useful for estimating the costs for extracting and disposing of contaminated soil.

When we ran all three, the results were:

Minimum Plume:

```
Soil Case
Geologic Layers: All
Computations for 80%Min:TOTHC
plume_volume Level = 1.0000E+001 mg/kg (ppm)
Total Soil Volume = 2.5515E+004 Cubic Yards
Total Soil Mass = 1.6699E+011 U.S. Short Tons
Chemical Volume = 8.9565E+000 Cubic Yards
Chemical Mass = 3.1684E+007 U.S. Short Tons
Average 80%Min:TOTHC = 1.8974E+002 mg/kg (ppm)
Volume $1.5309E+006
Mass $1.2674E+010
Center of Mass ( 11,330.4300, 12,887.2081, -12.8413 )
```

Nominal Plume:

```
Soil Case
Geologic Layers: All
Computations for TOTHC
plume_volume Level = 1.0000E+001 mg/kg (ppm)
Total Soil Volume = 6.7199E+004 Cubic Yards
Total Soil Mass = 4.3979E+011 U.S. Short Tons
Chemical Volume = 2.6300E+002 Cubic Yards
Chemical Mass = 9.3040E+008 U.S. Short Tons
Average TOTHC = 2.1156E+003 mg/kg (ppm)
Volume $4.0319E+006
Mass $3.7216E+011
Center of Mass ( 11,349.3369, 12,905.2901, -2.4122 )
```

Maximum Plume:

```
Soil Case
Geologic Layers: All
Computations for 80%Max:TOTHC
plume_volume Level = 1.0000E+001 mg/kg (ppm)
Total Soil Volume = 1.5187E+005 Cubic Yards
Total Soil Mass = 9.9391E+011 U.S. Short Tons
Chemical Volume = 2.7851E+004 Cubic Yards
Chemical Mass = 9.8527E+010 U.S. Short Tons
Average 80%Max:TOTHC = 9.9131E+004 mg/kg (ppm)
Volume $9.1121E+006
Mass $3.9411E+013
Center of Mass ( 11,364.5953, 12,918.0579, 6.1397 )
```

Note that there is nearly a six fold difference between the Min and Max plume volumes. We wouldn't normally consider this well characterized, especially since it would result in a nearly $8M difference in the cost to extract the max soil plume (vs. the min plume) if disposal and extraction costs were $60 per cu. yd. This kind of information makes it easier to see the cost advantages associated with further (better characterized) site assessment, and can help identify when you've done enough.

Geostatistics Conclusion

EVS combines ease of use with the tools and speed to get your jobs done on-time and on-cost. Its strong geostatistical foundation guarantees that your work will be defensible, and that work can confidently be used to guide a cost-effective remediation program.

Kriging References

Many texts have been written on kriging since the mathematical method is so widely used and has been accepted for decades. The following two references will provide good background material related to kriging and the numerical methods employed in the calculations.

GSLIB (Geostatistical Software Library and User's Guide) Second Edition

Clayton V. Deutsch and Andre G. Journel
Oxford University Press 1998.

2) Practical Geostatistics
Isobel Clark Ph.D.
Published by Applied Science Publishers, Ltd. London
Ripple Road, Barking, Essex, England
ISBN 0-85334-843-X

3) Geo-EAS (Geostatistical Environmental Assessment Software) User's Guide
Evan J. Englund, U.S. EPA, and
Allen R. Sparks, Computer Sciences Corporation
EPA Contract 68-01-7325
Write to: Evan J. Englund (Geo-EAS)
USEPA EMSL-LV, EAD
P.O. Box 93478
Las Vegas, NV 89193-3478

Workbook 10: Finite Difference Gridding

Workbook 10: Finite Difference Gridding

Finite Difference Gridding

Finite Difference Gridding (FDG) is a gridding tool provided by EVS/MVS for situations in which standard gridding methods have proven less efficient and/or less practical.

Examples of where FDG can be used include:

Sparse or irregular data.

Varied clusters of data, specially, those taken around DNAPLs.

When trying to match a building's boundary to standard grid, FDG is the only tool that rotates a grid.

As a preprocessor for MODFLOW and MT3D, FDG is used for initial geological input for Groundwater Vistas.

Introduction

In this Workbook, we use the Network Editor to build a network to perform finite difference gridding (FDG). The application:

Reads an EVS chemistry data file.

Performs data estimation (Kriging) of the sparse measured chemistry data.

Usespost_samplestodisplaythemeasureddata.

Develops a grid to lay out the measured data on the surface using Krig 3D Geology.

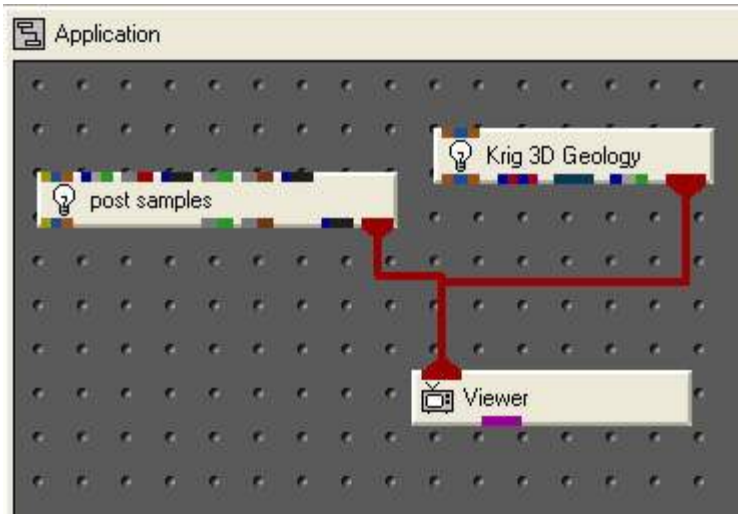
Usesgeologic_surface to visualize the surface elevations at the given data points.

Visualizes a plume of the data using plume_shell.

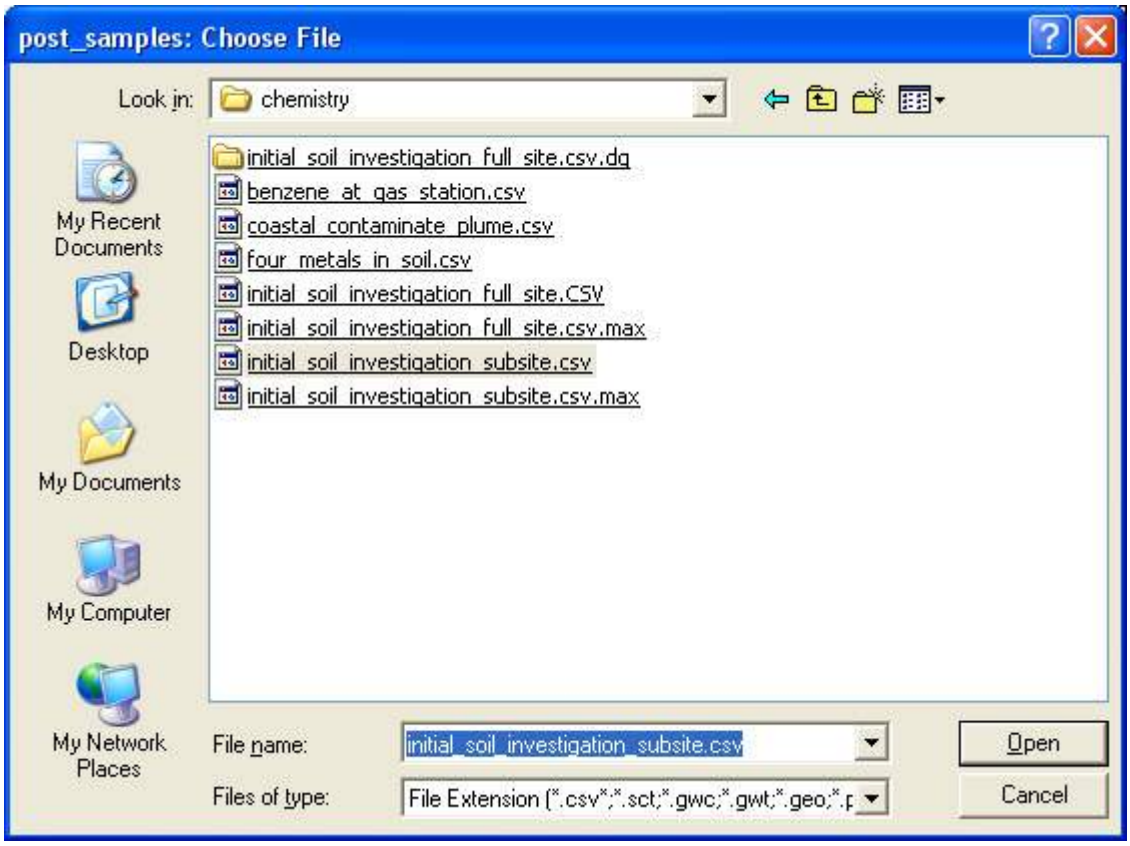
We are then able to use Explode and Scale to separate the different layers of the plume, and then explode the post_samplesto match the separated layers.

Add post_samples

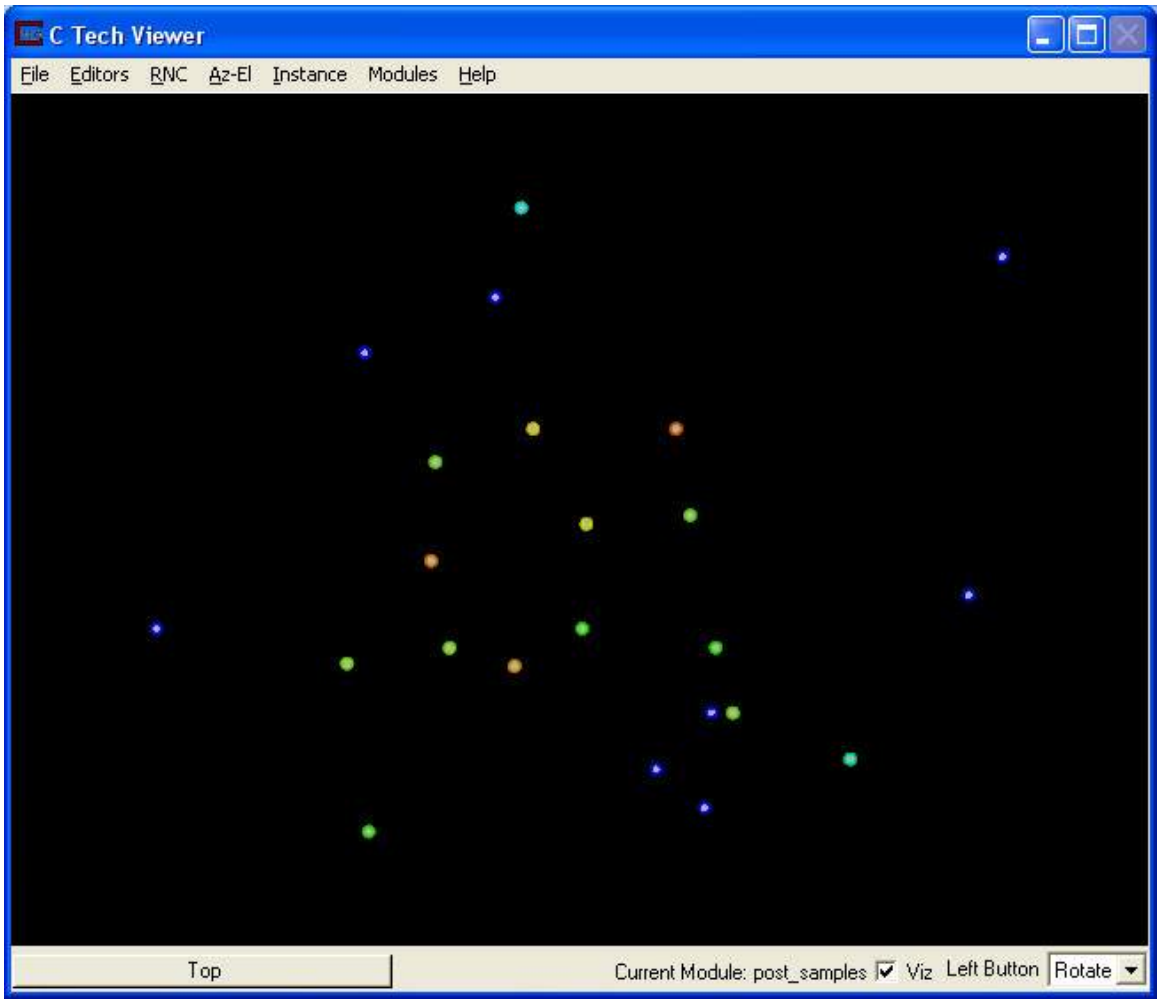
Instance the modules as shown in the figure below:



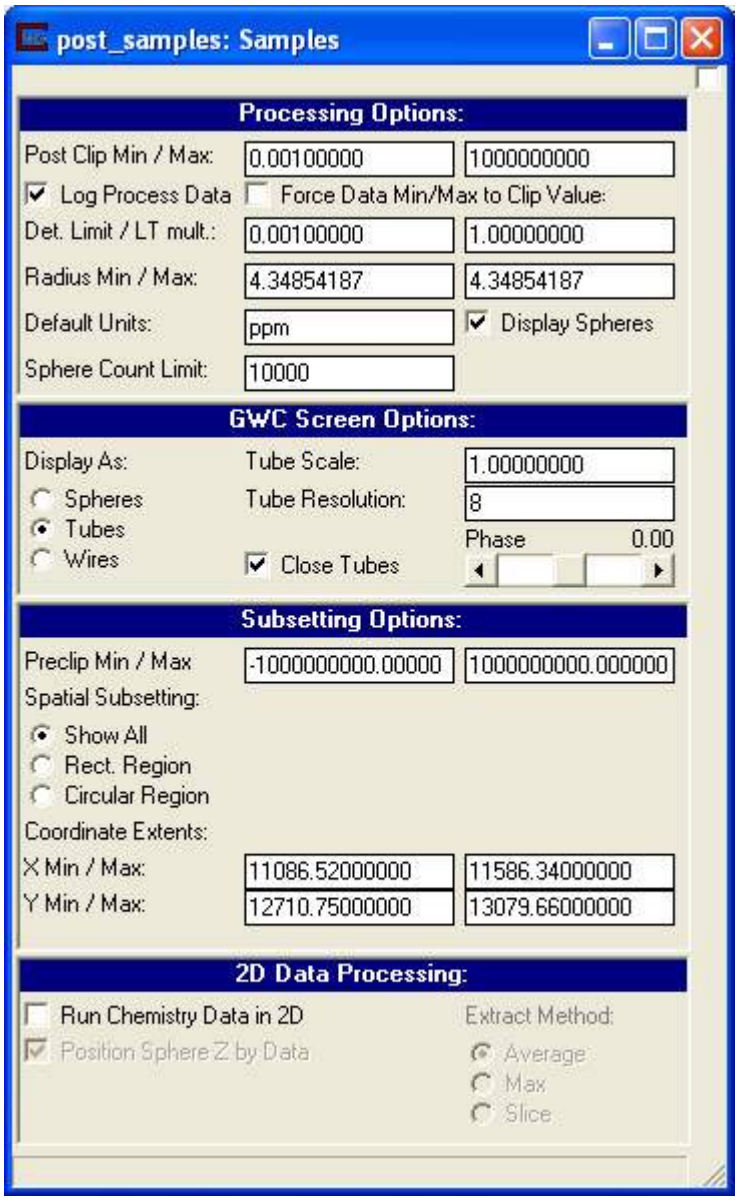
In post_samples click on the *Load File:* button. Choose the file initial_soil_investigation_subsite.apdv and click OK.



You should see the following image in your Viewer.



On the post_samples interface, open the *Samples* window.



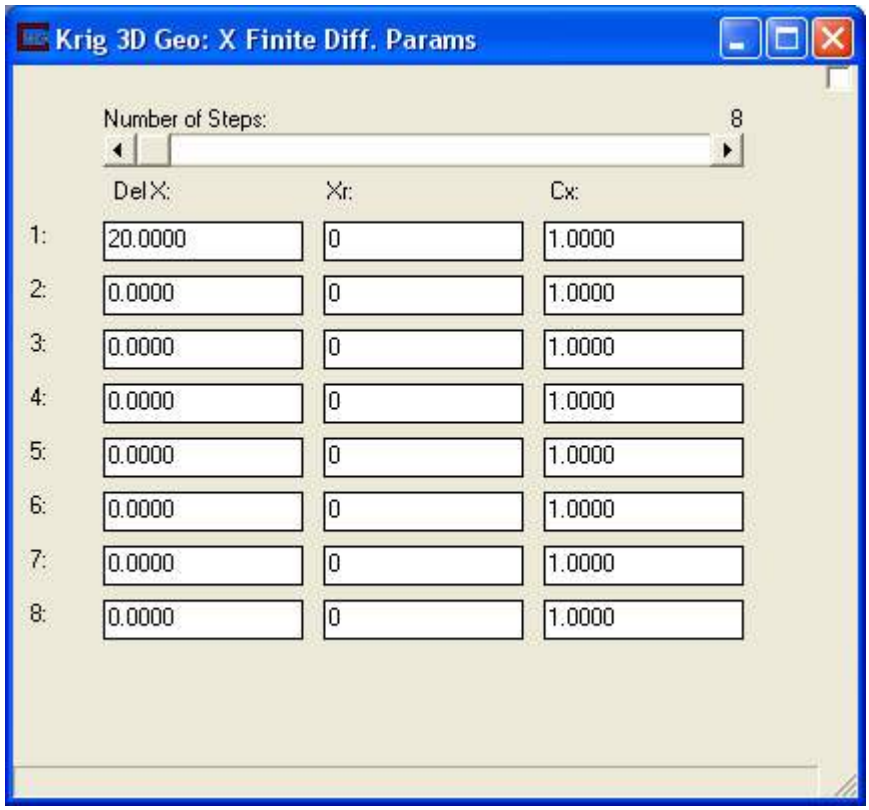
Note the data extents of Xmin/ Ymin. We will need this data later, so make a note of the values. If we turn on Options->Show Multiple Modules, we can keep this window open when we open other modules.

Krig_3D geology

Instance the Krig_3D Geology Module. In the user interface, select Gridding Options:

Enter the data extents (enter only MinX, MinY) noted from *post_samples: Samples* parameters.

In the same window (Gridding Options), for Grid Type, select **Finite Difference Gridding** then open the X window.



This user interface allows us to customize our grid to our needs. Since we have chosen to customize our grid in the X direction:

DelX: is the length of the first unit square in the X direction.

Xr: is the number of squares in the X direction

Cr: is the percentage of change for each progressive square. The change allows us to vary our grid based on the density of our data.

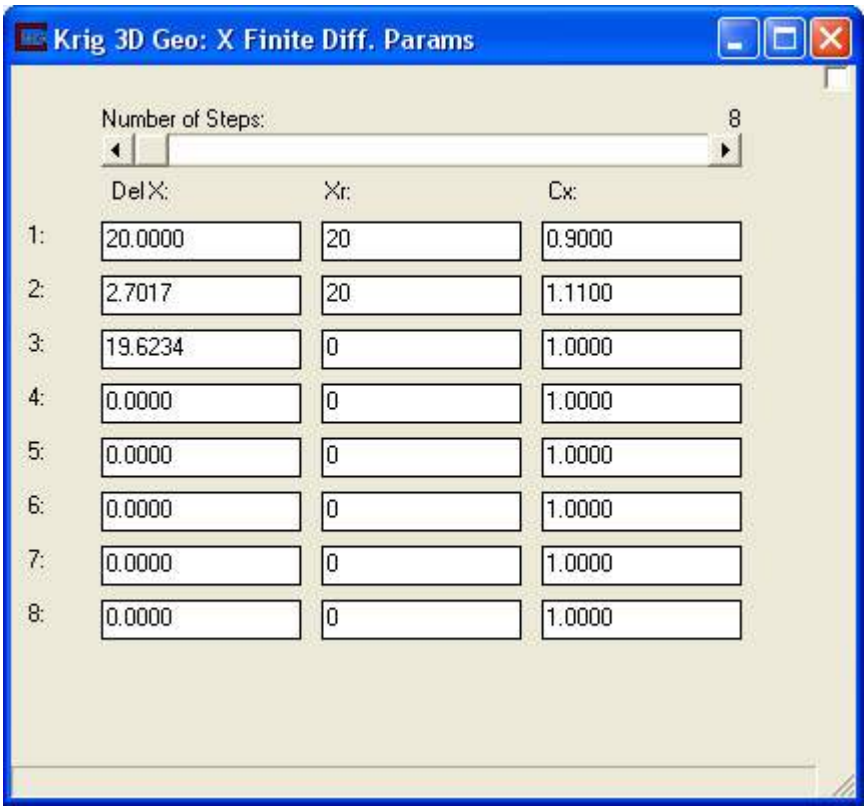
So, our grid can have an initial cell length of DelX, the next cell would be Cr percent smaller or larger, in magnitude, than our initial DelX, and there is only Xr cells in our initial grid. We can have up to 8 different variations in the X direction. The same is true in the Y direction.

To expand our grid to fit the area of our data points, we add to our initial grid by changing our parameters in X.

Start by entering a value of 20 as the first Xr parameter, and 0.9 in the first Cx parameter. Then hit *Accept All Current Values* in the main Krig_3D_Geology window. Once you've done that, you'll notice that the Del X parameter for the second line in X was set to 2.7017. This was automatically calculated based off the values we set before.

Now set Xr to 20 and Cx to 1.11 in the second row, and hit *Accept* again.

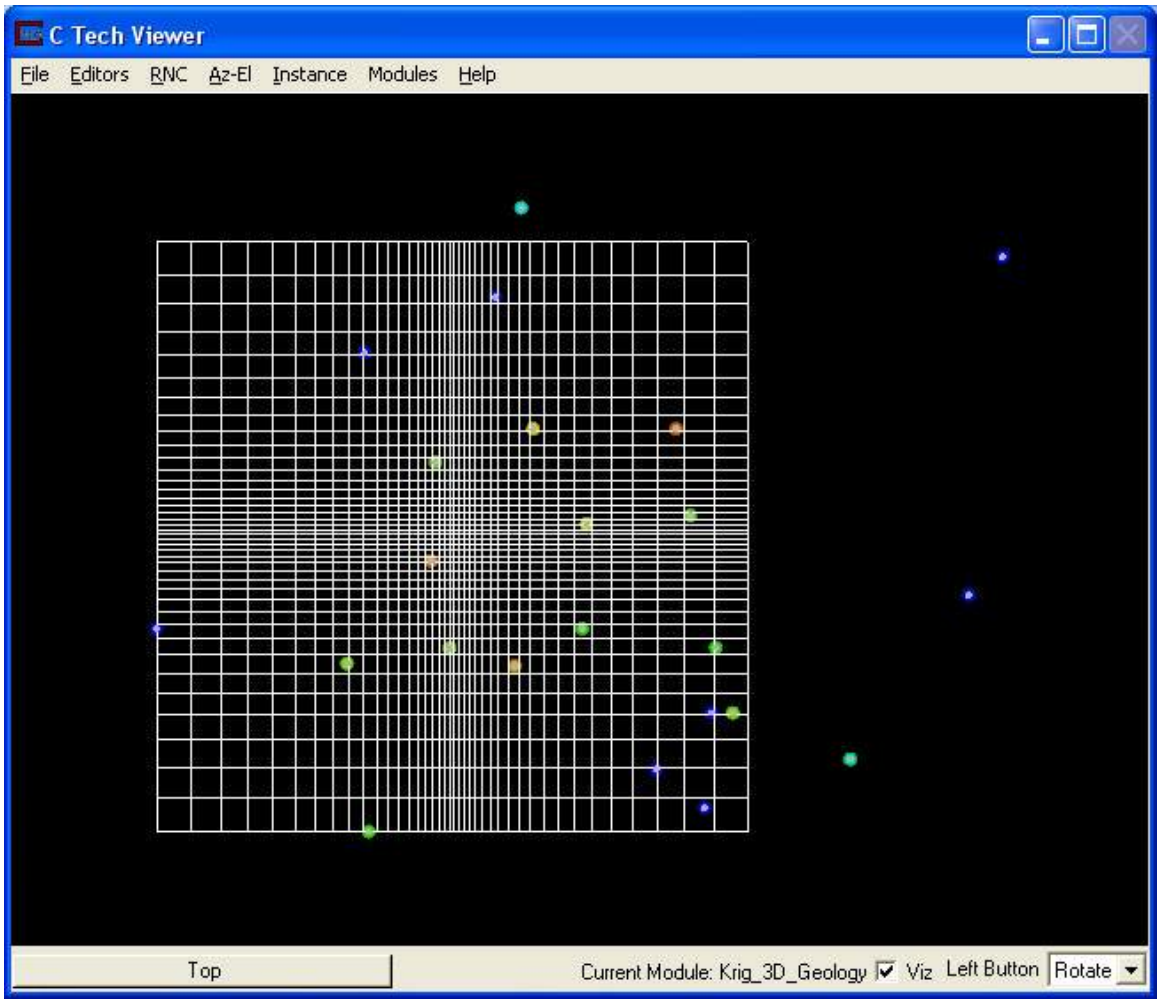
You're window should now show:



Repeat the above process exactly for Y.

Press "Accept all current values" in the Krig_3D_Geology user interface

In the viewer, you should see the following image.

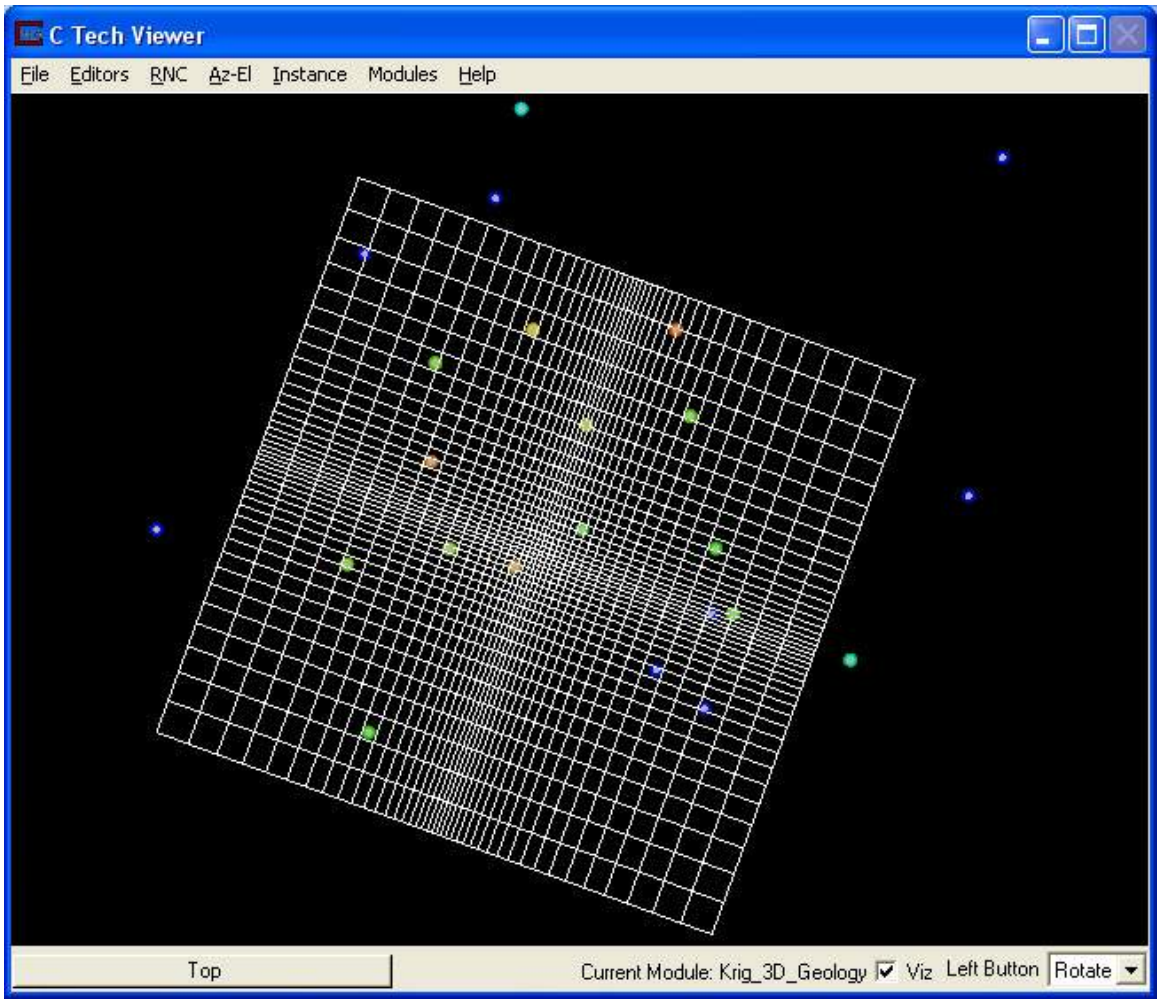


Rotating the Grid

Note that the grid we have created does not include a few of the data points. To remedy this situation we can redraw the grid or we could rotate it to include more data points. In this exercise we will rotate the grid.

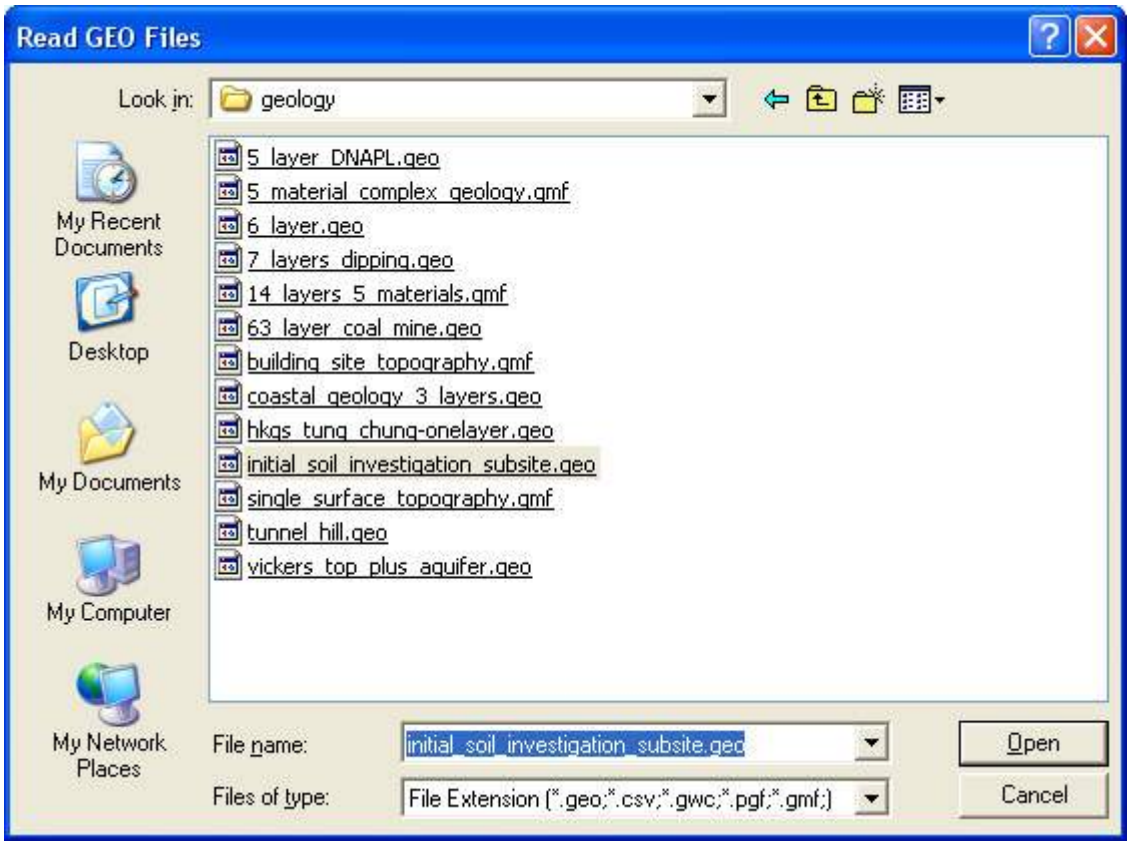
Note the box next to Angle. We can input any angle we feel will cover a larger number of data points. In this case we have chosen -20 degrees.

Hit "Accept All Current Values" and the Viewer will update to show:



Instancing Geologic Surface Module

In the user interface, press the "Read Geo File" button and select the initial_soil_investigation_subsite.geo file as shown below.



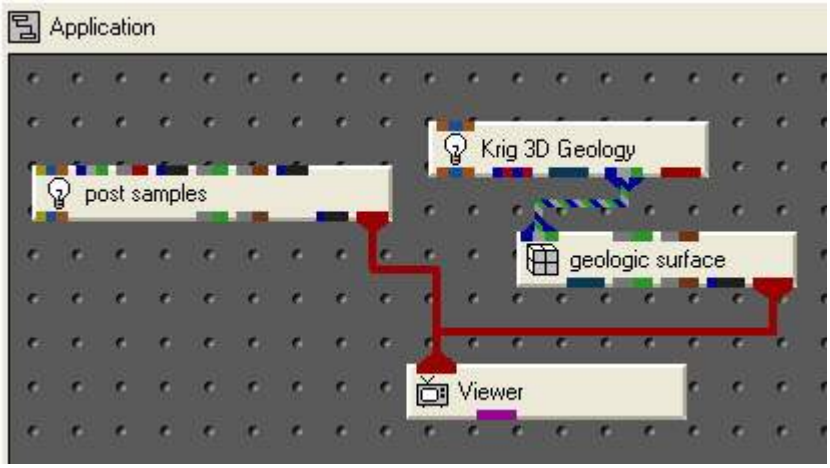
Choose *Accept AllCurrent Values* in Krig_3D_Geology.

Delete the connection between Krig 3D Geology and the viewer module.

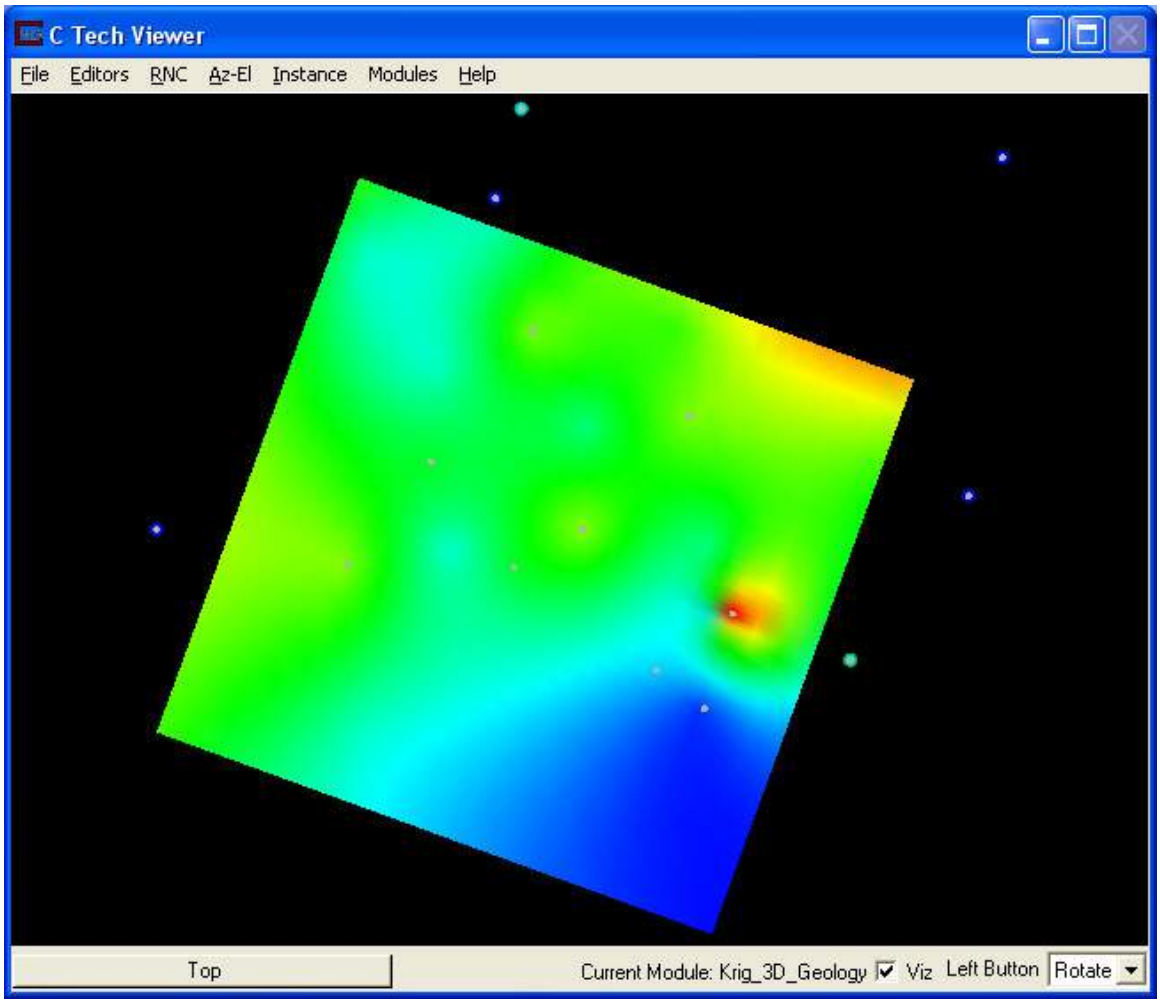
Instance geologic_surface under Krig_3D_Geology.

Connect the output data port (blue gray green) on the Krig 3D Geology to the same colored input port on geologic_surface.

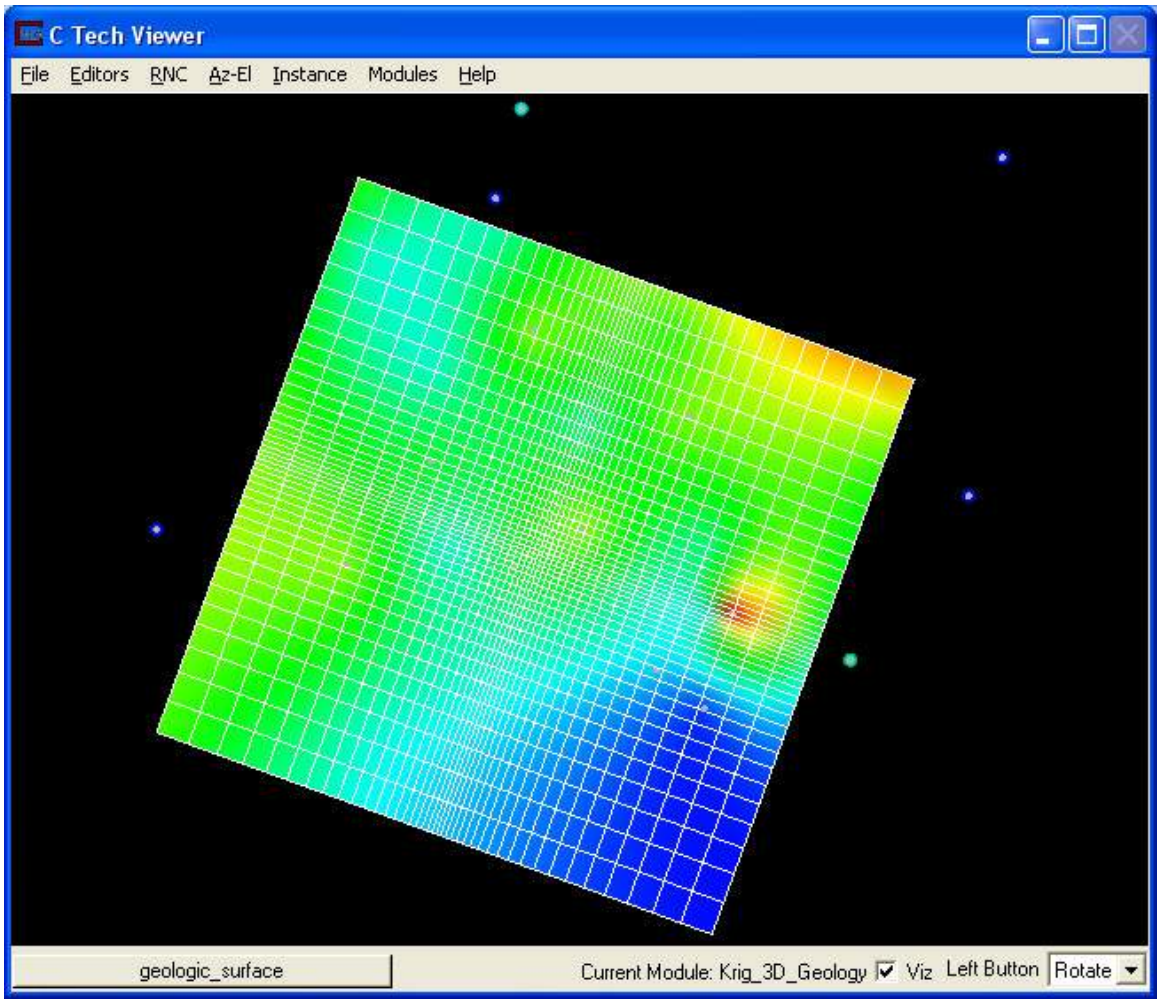
Connect the red output port of geologic_surface to the viewer module.



The image we see below is the image of our surface on our shaped to our grid. The colors indicate the elevations of our data points at those locations.


C Tech Viewer
File Editors RNC Az-El Instance Modules Help
Top
Current Module: Krig_3D_Geology
Viz
Left Button
Rotate

Line Rendering



To view the grid along with our surface, follow these steps.

Choose geologic_surface as the object to edit. You can do this by pressing the *Top* button on the bottom of Viewer and selecting *geologic_surface*. (Clicking on the surface itself with Alt+Left mouse button would give you the same result.) If you correctly select the geologic_surface object, the button in the lower left corner of the Viewer will be labeled "geologic_surface."

Go to *Editors->Object->Advanced Settings* in the Viewer

At the top of the *Viewer: Object Editor* window, choose *Rendering Modes*.

Under Line Rendering, choose *Regular*.

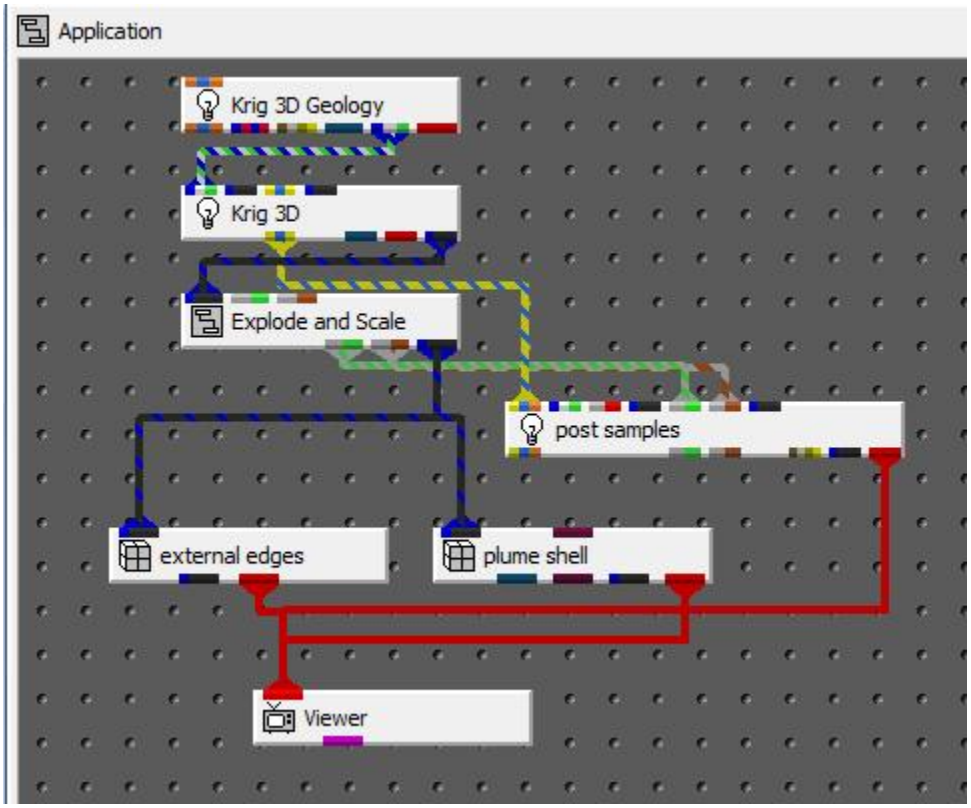
You should get the image shown above. Be sure to return the Object editor to it original state for the next steps.

Visualizing the Plume

To visualize the plume of our data, follow these steps:

Delete the module geologic_surface.

Add the modules and connect them as shown in image below.

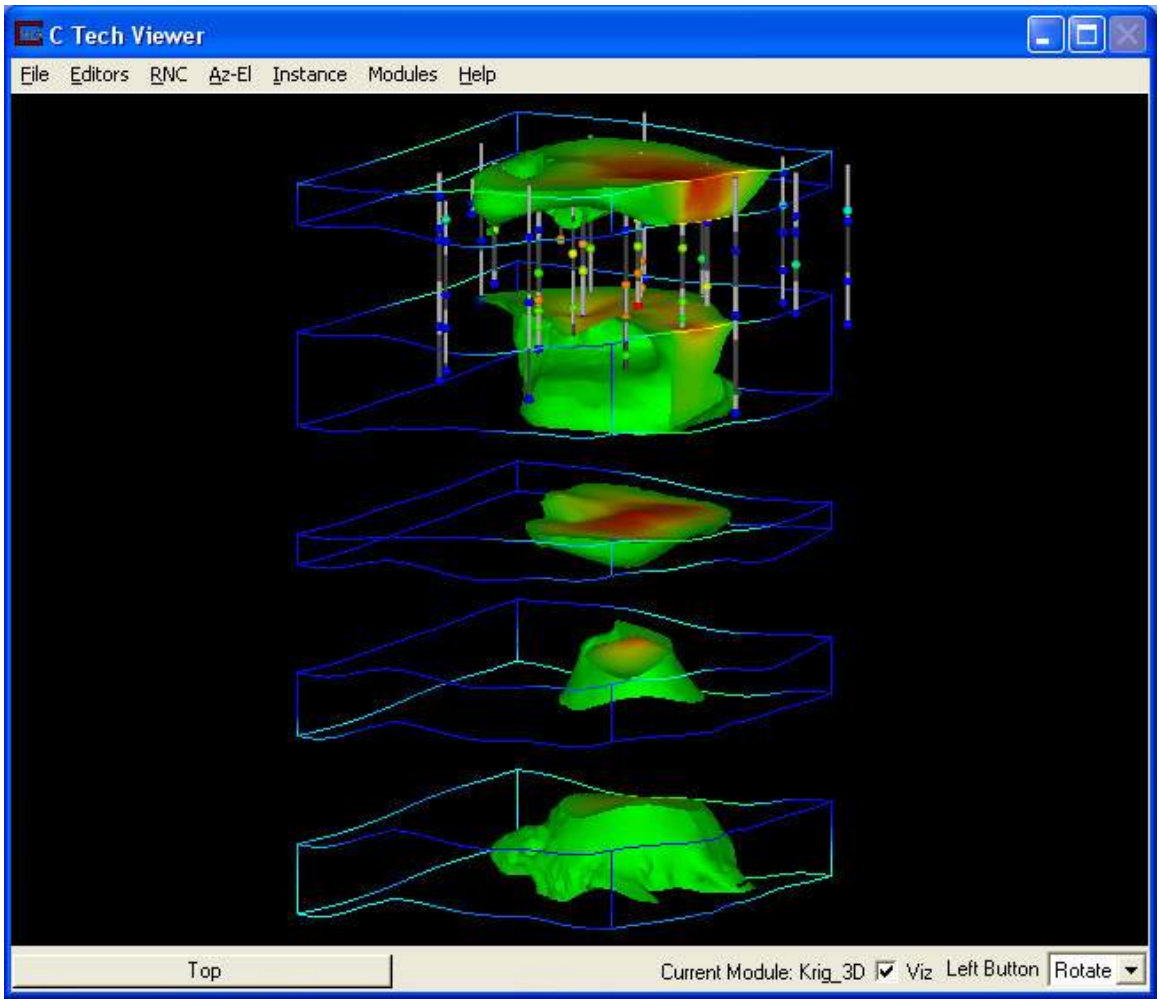


From the Modules menu, open the Krig 3D user interface. Press on the "Accept All Current Values" button.

Once the process is complete, instance the Az_El, change the following:

scale factor = 0.65, Elev = 10, and rotation = 75.

The image in the viewer should look like the image below.

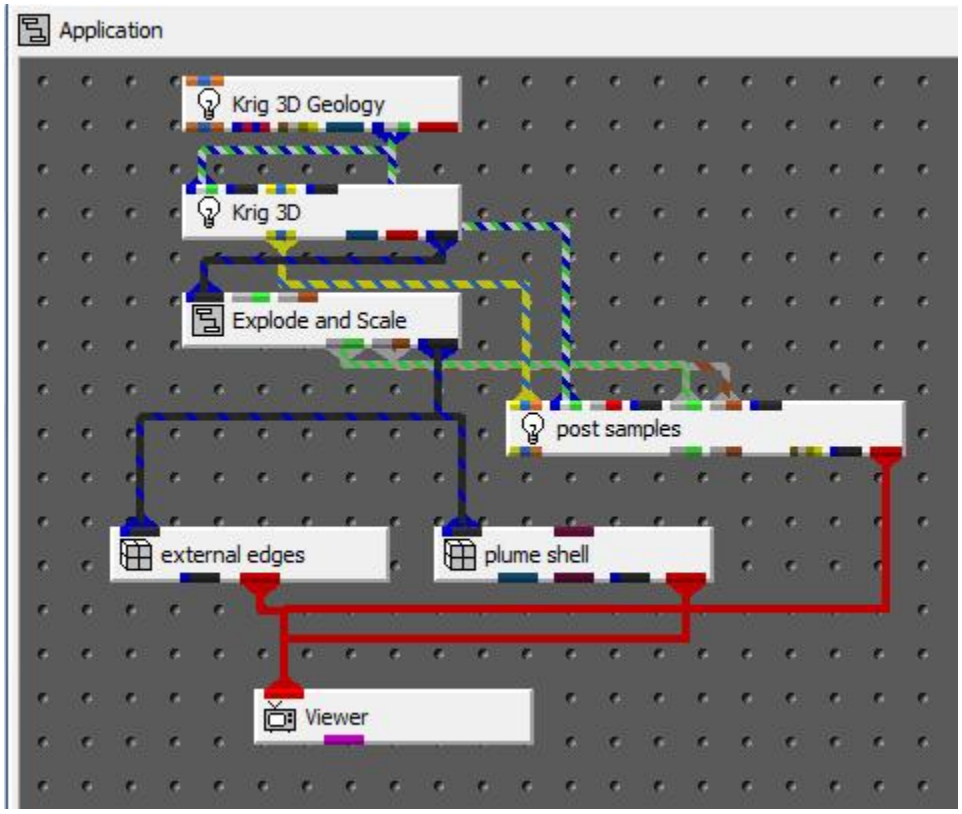


Exploding the Spheres

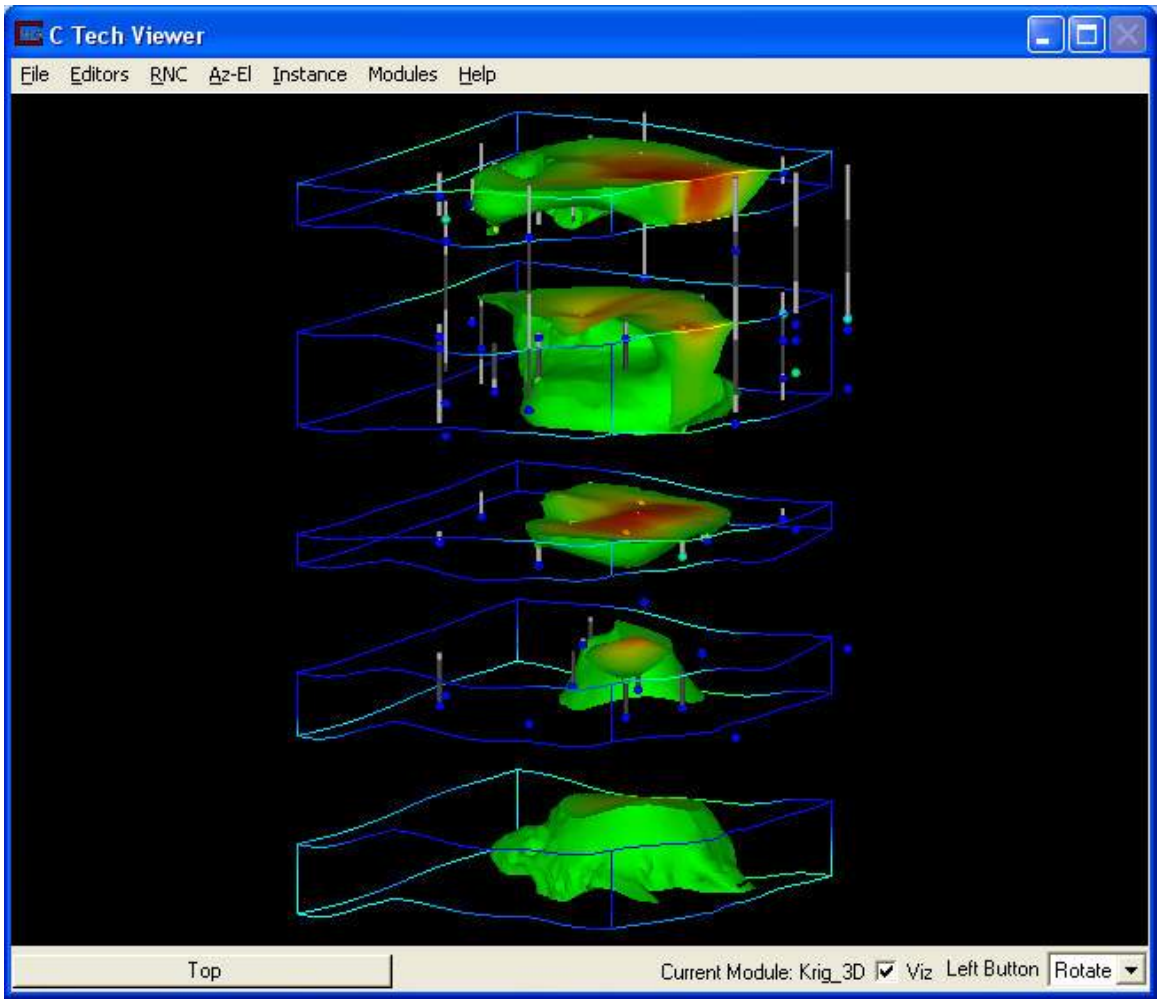
In the above image of the exploded plume, we can see that the plume has been separated/ exploded along the edges of the geologic layers. Notice that the post_samples data points have not been exploded. To explode the post_samples to match the geologic layers we need to do the following:

Connect the x,y,z, geologic data output port (blue/gray/green) on Krig 3D Geology module to the corresponding input port of post_samples module.

Since you have already connected the gray/green & gray/brown output ports of Explode and Scale to the corresponding input ports of the post_samples module. it will know the explode distance and z-scale.



We should have the following image in the viewer:



Borings falling outside of the extents of the geology are not exploded; however, all borings within the range of the geology are exploded based on the geologic information pass in from Krig_3D_Geology. By extending the range of our finite difference grid to include all of the sample locations, we could explode all of the borings.

Workbook 11: Advanced Geologic Modeling Concepts

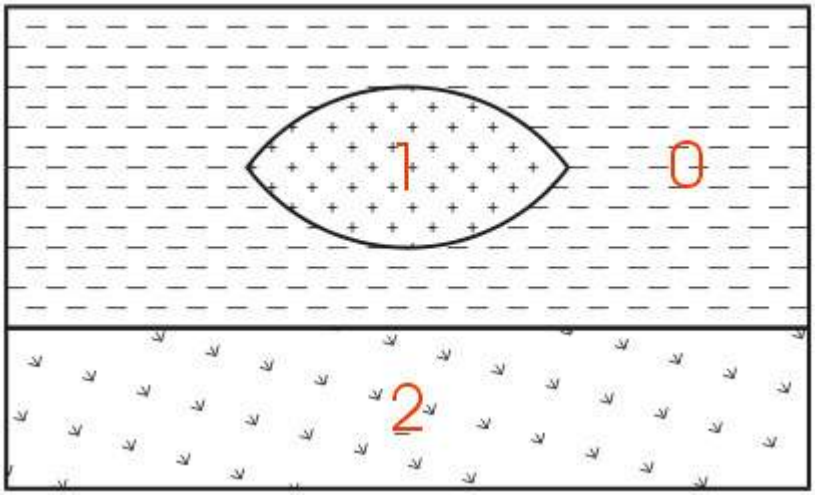
- Geologic Modeling Concepts
- Pregeology File Format
- Visualizing Raw 3D Boring Logs
- Advanced Hierarchical Layer-Based Geologic Modeling
- Indicator_Geology - Indicator Kriging
- Indicator_Geology - Probabilistic Kriging
- Indicator_Geology - Material Probabilities
- Anisotropy Effects on Indicator Kriging
- Advanced Geologic Modeling Conclusion

- Workbook 1 Fundamentals and Two-Dimensional Kriging:

Geologic Modeling Concepts

Geologic Hierarchy

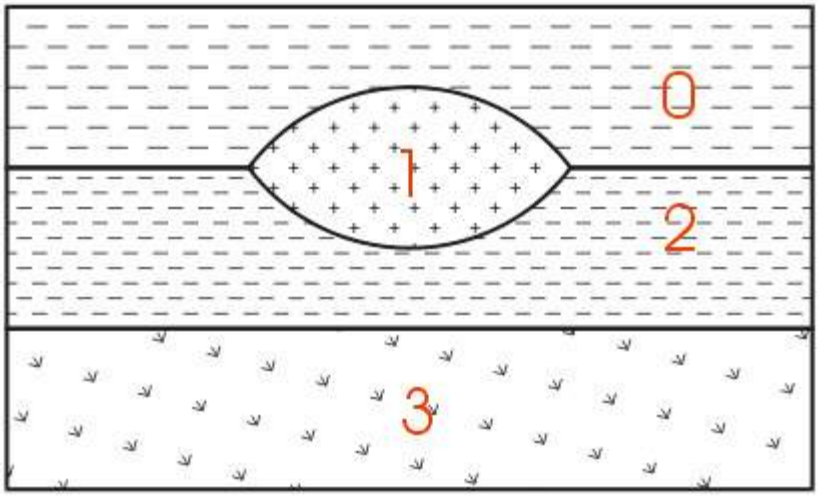
C Tech's original method for 3D geologic modeling utilizes one of two different ASCII file formats which contain "interpreted" geologic information. These two file formats both describe points on each geologic surface (ground surface and bottom of each geologic layer), based on the assumption of a geologic hierarchy. Simply stated, layer hierarchy requires that all geologic layers throughout the domain be ordered from top to bottom and that a consistent hierarchy be used for all borings. At first, it may not seem possible for a uniform layer hierarchy to be applicable for all borings. Layers often pinch out or exist as localized lenses. Also layers may be continuous in one portion of the domain, but are split by another layer in other portions of the domain. However, all of these scenarios and many others **can** be easily modeled using a hierarchical approach.



The easiest way to describe geologic hierarchy is with an example. Consider the example above of a clay lens in sand with gravel below.

Imagine borings on the left and right sides of the domain and one in the center. Those outside the center would not detect the clay lens. On the sides, it appears that there are only two layers in the hierarchy, but in the middle there are three materials and four layers.

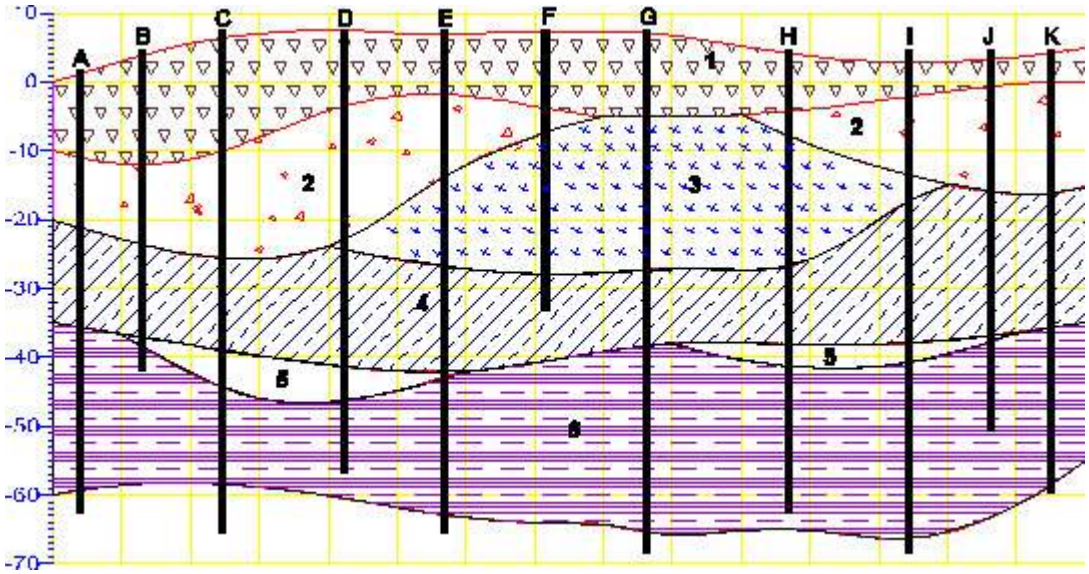
EVS's & MVS's hierarchical geologic modeling approach accommodates the clay lens by treating every layer as a sedimentary layer. Because we can accommodate "pinching out" layers (making the thickness of layers ZERO) we are able to produce most geologic structures with this approach. Geologic layer hierarchy requires that we treat this domain as 4 geologic layers. These layers would be Upper Sand (0), Clay (1), Lower Sand (2) and Gravel (3).



If desired, both Upper and Lower Sand can have identical colors or hatching patterns in the final output.

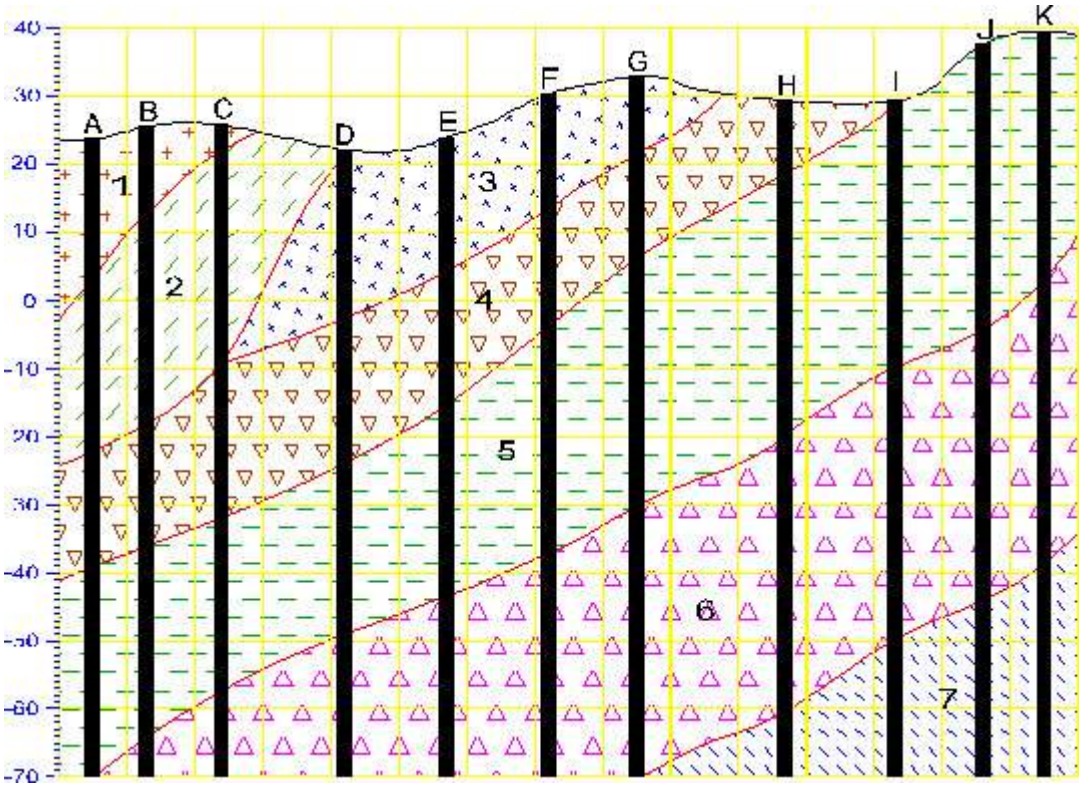
For those sites that can be described using the above method, it remains the best approach for building a 3D geologic model. Each layer has smooth boundaries and the layers (by nature of hierarchy) can be exploded apart to reveal the individual layer surface features. In the above example, the numbers represent the layer numbers for this site (even though layers 1 and 3 are both sand). Two examples of much more complex sites that are best described by this original approach are shown below.

Geologic Example: Sedimentary Layers and Lenses



Geology Example & Figure: Outcrop of Dipping Strata

EVS is not limited to sedimentary layers or lenses. The figure below shows a cross-section through an outcrop of dipping geologic strata. EVS can easily model the layers truncating on the top ground surface.



Pregeology File Format

This pregeology file is used to represent raw 3D boring logs. We also refer to this geologic data format as "uninterpreted". This is not meant to imply that no form of geologic evaluation or interpretation has occurred. On the contrary, it is required that someone categorizes the materials on the site

and in each boring. Furthermore, if the .pgf file is to be used for GIK, each boring must specify these materials as "CONSECUTIVE" integers. By consecutive we mean that if there are 7 materials they must be represented as integers from 0 to 6 (for many reasons we do not recommend that you start at 1 versus 0). There cannot be any gaps in the material number assignments. This requirement for consecutive assignments does not apply if the pgf file is to be used exclusively by the Pre_Geology module (the only other module that reads .pgf files).

The table below shows the file 5_material_complex_geology.pgf located in the ctech\data\pregeology directory. This file will be used for the bulk of the examples used in this workbook.

```
EAST NORTH TOP-BOT MATERIAL-ID WELL_ID
Elevation Clay Silt Silty-sand Sand Gravel meters
169 1
2085487.12 322869.95 31 4 AW-3
2085487.12 322869.95 -1 4 AW-3
2085487.12 322869.95 -3 2 AW-3
2085487.12 322869.95 -19 1 AW-3
2085487.12 322869.95 -24 4 AW-3
2085108.47 323363.89 32 4 MW-10A
2085108.47 323363.89 20 4 MW-10A
2085108.47 323363.89 12 3 MW-10A
2085108.47 323363.89 -6 1 MW-10A
2085108.47 323363.89 -16 4 MW-10A
2085108.47 323363.89 -18 1 MW-10A
2085079.22 323361  32 4 MW-10B
2085079.22 323361  20 4 MW-10B
2085079.22 323361  12 3 MW-10B
2085079.22 323361  -6 1 MW-10B
2085079.22 323361  -16 4 MW-10B
2085079.22 323361  -22 1 MW-10B
2085079.22 323361  -46 0 MW-10B
2085079.22 323361  -58 1 MW-10B
2085266.93 323410.05 32 4 MW-11A
2085266.93 323410.05 14 4 MW-11A
2085266.93 323410.05 7 3 MW-11A
2085266.93 323410.05 2 2 MW-11A
2085266.93 323410.05 -11 1 MW-11A
2085266.93 323410.05 -13 0 MW-11A
2085274.82 323407.7 32 4 MW-11B
2085274.82 323407.7 14 4 MW-11B
2085274.82 323407.7 7 3 MW-11B
2085274.82 323407.7 2 2 MW-11B
2085274.82 323407.7 -11 1 MW-11B
2085274.82 323407.7 -14 0 MW-11B
2085274.82 323407.7 -31 4 MW-11B
2085274.82 323407.7 -40 1 MW-11B
```

```
2085274.82 323407.7 -58 0 MW-11B
2085538.95 323379.77 31 4 MW-12A
2085538.95 323379.77 -1 4 MW-12A
2085538.95 323379.77 -15 2 MW-12A
2085553.92 323379.03 31 4 MW-12B
2085553.92 323379.03 -1 4 MW-12B
2085553.92 323379.03 -15 2 MW-12B
2085553.92 323379.03 -29 1 MW-12B
2085553.92 323379.03 -54 0 MW-12B
2085553.92 323379.03 -59 1 MW-12B
2084941.15 323298.84 33 4 OW-1A
2084941.15 323298.84 18 4 OW-1A
2084941.15 323298.84 8 3 OW-1A
2084941.15 323298.84 0 2 OW-1A
2084941.15 323298.84 -6 1 OW-1A
2084941.15 323298.84 -15 0 OW-1A
2084941.15 323298.84 -21 4 OW-1A
2084941.15 323298.84 -26 1 OW-1A
2084941.29 323310.53 32 4 OW-1B
2084941.29 323310.53 17 4 OW-1B
2084941.29 323310.53 7 3 OW-1B
2084941.29 323310.53 -1 2 OW-1B
2084941.29 323310.53 -7 1 OW-1B
2084941.29 323310.53 -16 0 OW-1B
2084941.29 323310.53 -22 4 OW-1B
2084941.29 323310.53 -35 1 OW-1B
2084941.29 323310.53 -40 4 OW-1B
2084941.29 323310.53 -57 1 OW-1B
2084941.29 323310.53 -58 4 OW-1B
2084948.07 323309.4 32 4 OW-1C
2084948.07 323309.4 17 4 OW-1C
2084948.07 323309.4 7 3 OW-1C
2084948.07 323309.4 -1 2 OW-1C
2084948.07 323309.4 -7 1 OW-1C
2084948.07 323309.4 -16 0 OW-1C
2084948.07 323309.4 -22 4 OW-1C
2084948.07 323309.4 -35 1 OW-1C
2084948.07 323309.4 -40 4 OW-1C
2084948.07 323309.4 -57 1 OW-1C
2084948.07 323309.4 -68 4 OW-1C
2084948.07 323309.4 -78 1 OW-1C
2084948.07 323309.4 -106 0 OW-1C
2085536.45 323527.4 31 4 OW-3A
2085536.45 323527.4 -4 4 OW-3A
2085536.45 323527.4 -7 2 OW-3A
2085536.45 323527.4 -11 1 OW-3A
2085536.45 323527.4 -19 0 OW-3A
2085536.45 323527.4 -20 4 OW-3A
2085555.89 323523.87 31 4 OW-3B
```

```
2085555.89 323523.87 -4 4 OW-3B
2085555.89 323523.87 -7 2 OW-3B
2085555.89 323523.87 -11 1 OW-3B
2085555.89 323523.87 -19 0 OW-3B
2085555.89 323523.87 -31 4 OW-3B
2085555.89 323523.87 -39 1 OW-3B
2085555.89 323523.87 -52 0 OW-3B
2085555.89 323523.87 -55 4 OW-3B
2085547.71 323517.63 31 4 OW-3C
2085547.71 323517.63 -4 4 OW-3C
2085547.71 323517.63 -7 2 OW-3C
2085547.71 323517.63 -11 1 OW-3C
2085547.71 323517.63 -19 0 OW-3C
2085547.71 323517.63 -31 4 OW-3C
2085547.71 323517.63 -39 1 OW-3C
2085547.71 323517.63 -52 0 OW-3C
2085547.71 323517.63 -65 4 OW-3C
2085547.71 323517.63 -73 1 OW-3C
2085547.71 323517.63 -108 0 OW-3C
2085640.82 323381.97 31 4 OW-4A
2085640.82 323381.97 0 4 OW-4A
2085640.82 323381.97 -16 2 OW-4A
2085640.82 323381.97 -18 1 OW-4A
2085634.32 323366.94 31 4 OW-4B
2085634.32 323366.94 0 4 OW-4B
2085634.32 323366.94 -16 2 OW-4B
2085634.32 323366.94 -45 1 OW-4B
2085634.32 323366.94 -49 0 OW-4B
2085634.32 323366.94 -56 1 OW-4B
2085629.75 323379.66 31 4 OW-4C
2085629.75 323379.66 0 4 OW-4C
2085629.75 323379.66 -16 2 OW-4C
2085629.75 323379.66 -45 1 OW-4C
2085629.75 323379.66 -49 0 OW-4C
2085629.75 323379.66 -56 1 OW-4C
2085629.75 323379.66 -76 4 OW-4C
2085629.75 323379.66 -116 0 OW-4C
2085194.26 323052.76 31 4 X-1A
2085194.26 323052.76 -1 4 X-1A
2085194.26 323052.76 -7 2 X-1A
2085194.26 323052.76 -11 1 X-1A
2085194.26 323052.76 -19 0 X-1A
2085194.26 323052.76 -23 4 X-1A
2085176.87 323061.14 31 4 X-1B
2085176.87 323061.14 15 4 X-1B
2085176.87 323061.14 8 3 X-1B
2085176.87 323061.14 -1 4 X-1B
2085176.87 323061.14 -7 2 X-1B
2085176.87 323061.14 -17 1 X-1B
```

```
2085176.87 323061.14 -19 0 X-1B
2085176.87 323061.14 -30 4 X-1B
2085176.87 323061.14 -41 1 X-1B
2085176.87 323061.14 -51 0 X-1B
2085176.87 323061.14 -63 1 X-1B
2085192.22 323066.87 31 4 X-1C
2085192.22 323066.87 -1 4 X-1C
2085192.22 323066.87 -7 2 X-1C
2085192.22 323066.87 -11 1 X-1C
2085192.22 323066.87 -19 0 X-1C
2085192.22 323066.87 -30 4 X-1C
2085192.22 323066.87 -38 1 X-1C
2085192.22 323066.87 -51 0 X-1C
2085192.22 323066.87 -61 1 X-1C
2085192.22 323066.87 -74 4 X-1C
2085192.22 323066.87 -83 1 X-1C
2085192.22 323066.87 -103 0 X-1C
2085153.84 323253.4 31 4 X-2A
2085153.84 323253.4 15 4 X-2A
2085153.84 323253.4 8 2 X-2A
2085153.84 323253.4 -17 1 X-2A
2085153.84 323253.4 -21 4 X-2A
2085184.59 323397.19 32 4 X4-A
2085184.59 323397.19 16 4 X4-A
2085184.59 323397.19 6 3 X4-A
2085184.59 323397.19 3 2 X4-A
2085184.59 323397.19 -9 1 X4-A
2085184.59 323397.19 -15 0 X4-A
2085184.59 323397.19 -22 4 X4-A
2085171.22 323387.25 31 4 X4-B
2085171.22 323387.25 15 4 X4-B
2085171.22 323387.25 5 3 X4-B
2085171.22 323387.25 2 2 X4-B
2085171.22 323387.25 -10 1 X4-B
2085171.22 323387.25 -16 0 X4-B
2085171.22 323387.25 -32 4 X4-B
2085171.22 323387.25 -39 1 X4-B
2085171.22 323387.25 -60 0 X4-B


037D6727D273B9216098B11AECC87192
```

note: the last line is the password that allows this file to be read in demo mode.

Visualizing Raw 3D Boring Logs

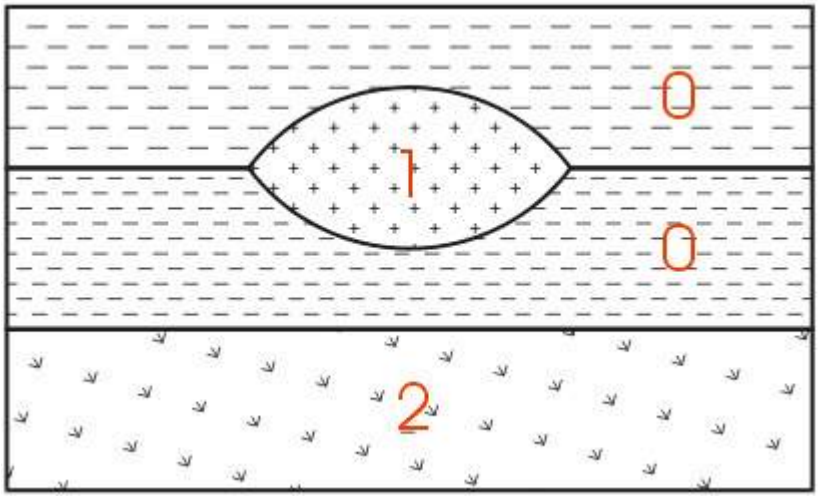
Many sites have geologic structures (plutons, karst geology, sand channels, etc.) that do not lend themselves to description within the context of hierarchical layers. For these sites, GIK offers the ability to build extremely complex models with a minimum of effort (and virtually no interpretation) on

the part of the geologist. GIK can also be a useful check of geologic hierarchies developed for sites that do lend themselves to a model based upon hierarchical layers.

GIK uses the raw, uninterpreted 3D borings logs in the form of a .pgf (pre-geology file) as the input. Consecutive integer values (e.g. 0 through n-1, for n total observed materials in the site) are used to describe each material observed in the entire site.

NOTE: It isrecommendedto start your material ID numbering at zero (0) instead of 1.

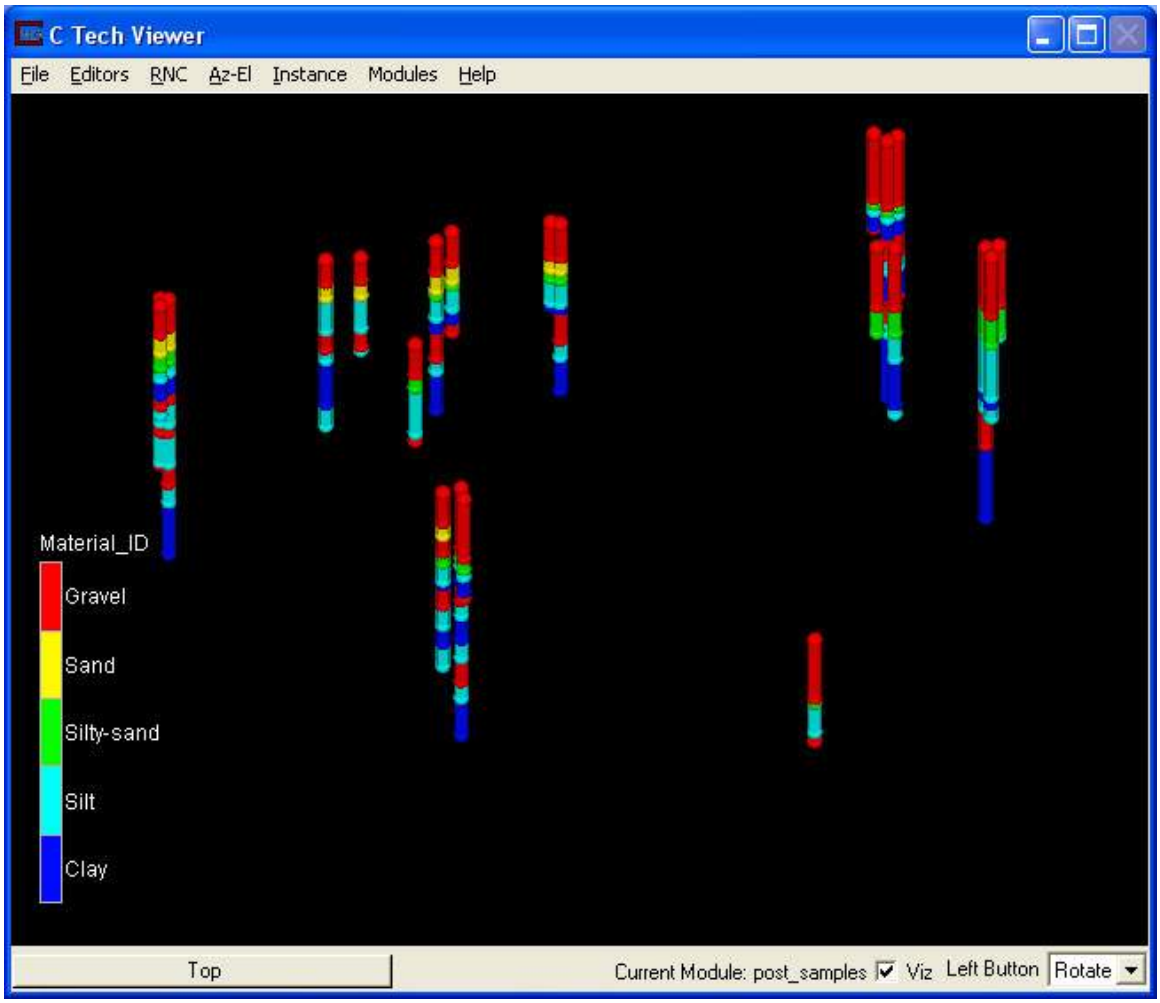
Usually, materials are numbered based upon a logical classification (such as porosity or particle size), however the numbering can be arbitrary as long as the numbers are consecutive (don't leave numbers out of the sequence). For the example given above, we could number the materials as shown in the figure below (even though it is not a numbering sequence based on porosity or particle size).



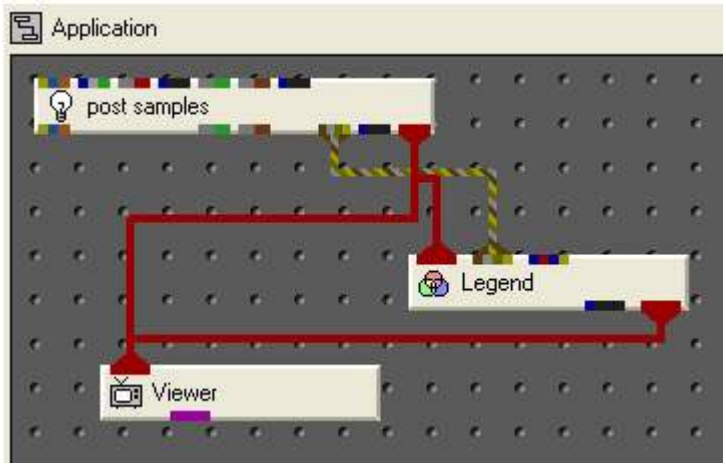
For a .pgf file, borings that do not see the clay (material 1 in the figure) would not need to consider the sand as being divided into upper and lower. Rather, every boring is merely a simple ASCII representation of the raw borings logs. The only interpretation involves classification of the observed soil types in each boring and assigning an associated numbering scheme.

As an example, we have taken the five material geologic data and created a model using both methods. This is a moderately complex site with geology that can be represented reasonably well with hierarchical layering. However the complexity at this site does require a significant amount of interpretation (by a geologist or analyst) that lends a subjective element to the resulting model.

The interpretation was performed interactively using the make_geo_hierarchy module in EVS/MVS. A display of 5_material_complex_geology.pgf using post_samples is shown in the figure below.



The application that was used to display the borings is shown in the figure below.



Note that there is a significant difference in the depths of some of the borings. Most notably, AW-3 (right foreground) is much shallower than the rest and is located quite a distance from any neighboring borings. With both techniques, extrapolation of the geology to the region below AW-3 will be low

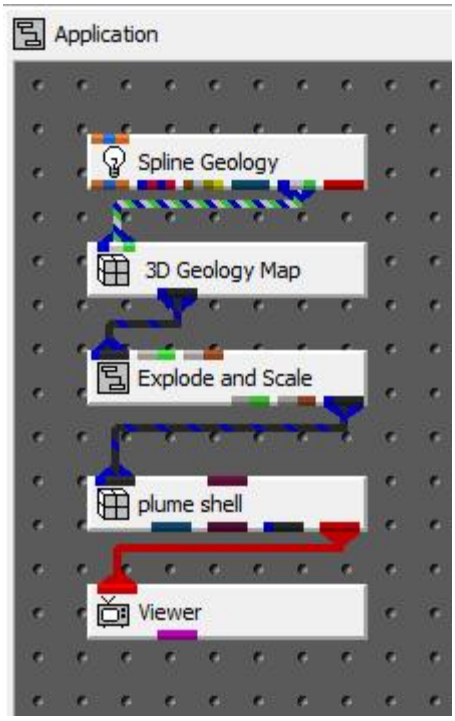
confidence and we expect that the two methods (Geologic Hierarchy and Geologic Indicator Kriging) will yield different results.

However both techniques should (and will) result in the same layering in the vicinity of the borings. This is a common theme in EVS for both geology and chemistry data. EVS will honor your measured data.

Advanced Hierarchical Layer-Based Geologic Modeling

Once the geologic hierarchy is established, a .geo or .gmf file is created (by hand or interactively using make_geo_hierarchy). The figure below is the geologic hierarchy model for the five material complex geology site. Note that the layers have relatively smooth boundaries and transitions in thickness. Also note that the extrapolated geology in the region below AW-3 is (arguably) a logical extension from the neighboring borings. Please remember that a significant effort was required to interpret the .pgf file and create an interpreted geologic data file (.gmf file) that was used to create this model.

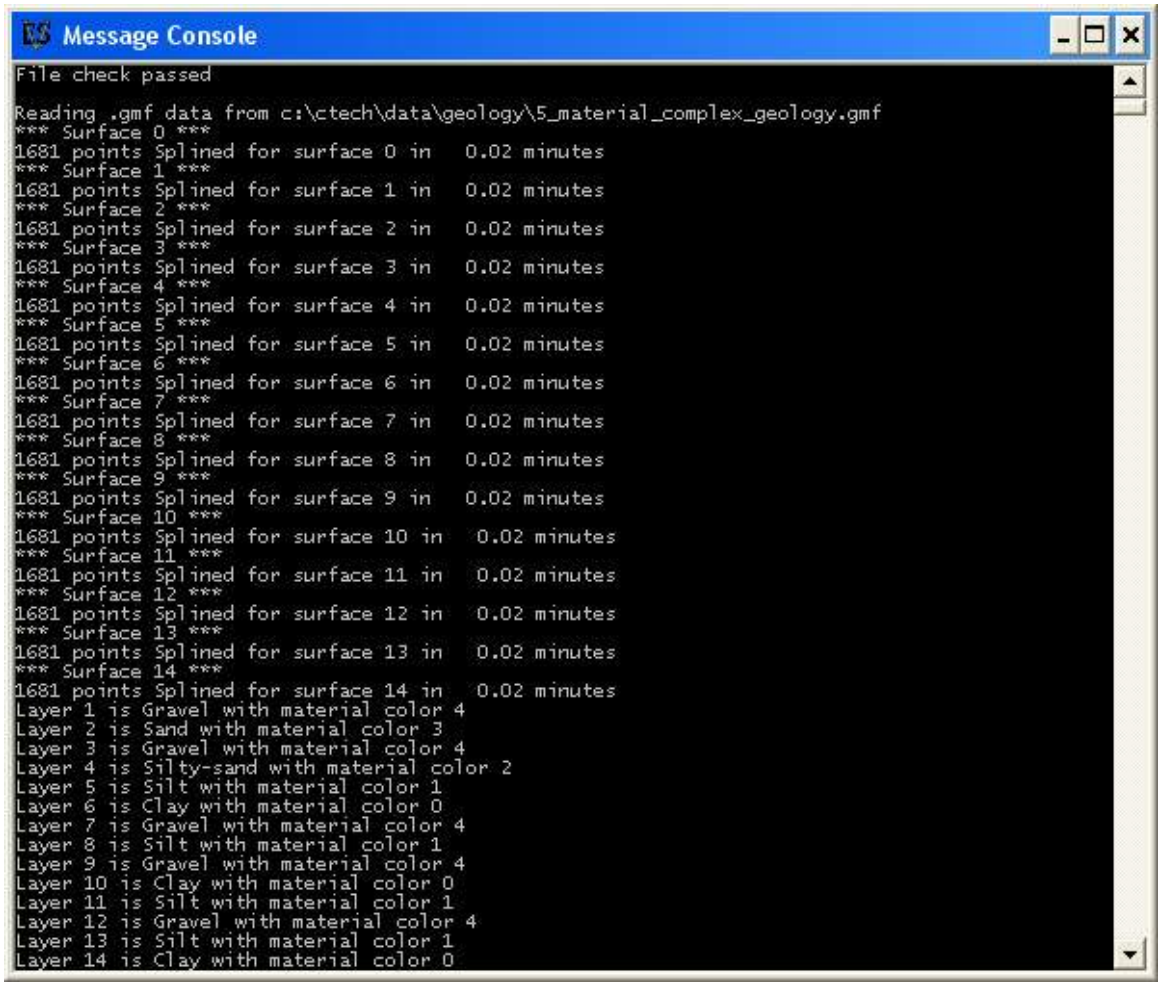
The application that we will begin with is shown below:



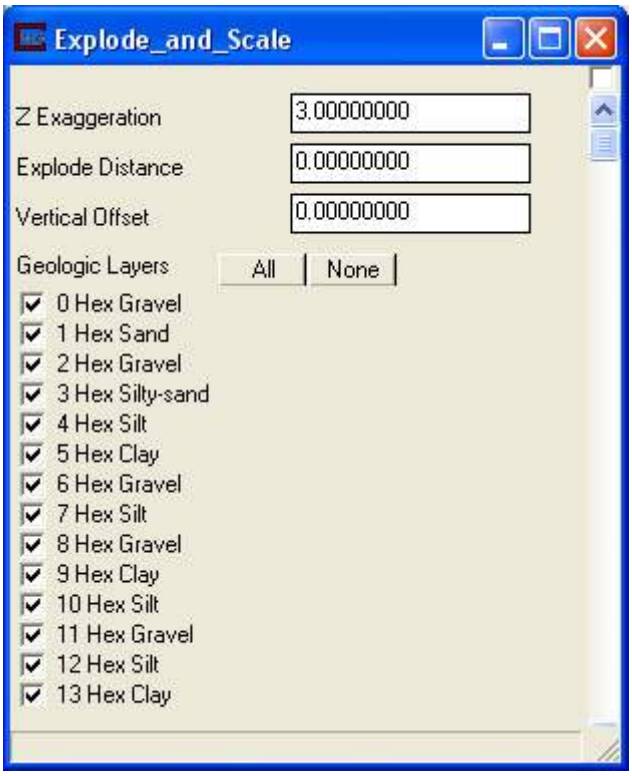
For this topic we have chosen to use Spline_Geology instead of Krig_3D_Geology. There are several reasons for this including:

1. 5_material_complex_geology.gmf has relatively few borings and splining tends to create smoother surfaces in these cases.
2. Splining's nature tends to overshoot and undershoot the data resulting in more dramatic surface variations.
3. Dramatic surface extrapolations often result in more realistic pinching of lenses.

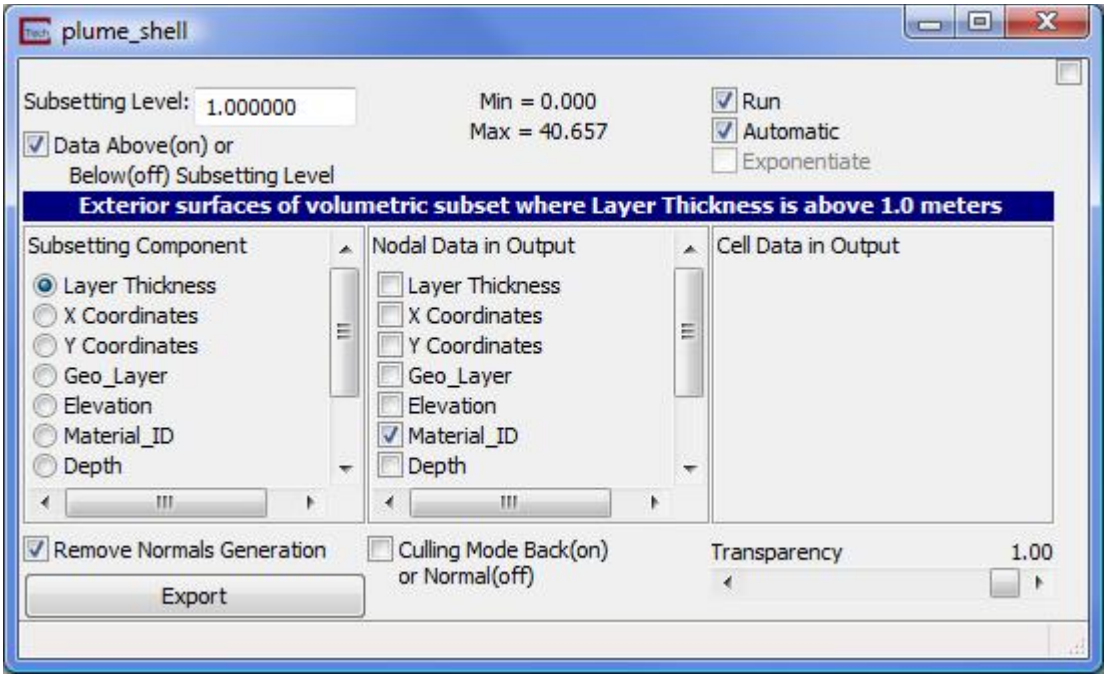
If we choose the file 5_material_complex_geology.gmf in Spline_Geology and run Spline_Geology using all of the default values the following messages will be printed to the Status Window.



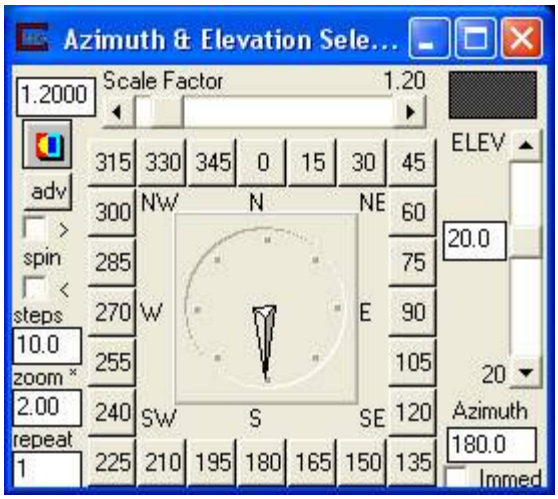
First, let's look at this model with the layers unexploded. Set the parameters in Explode_and_Scale to match:



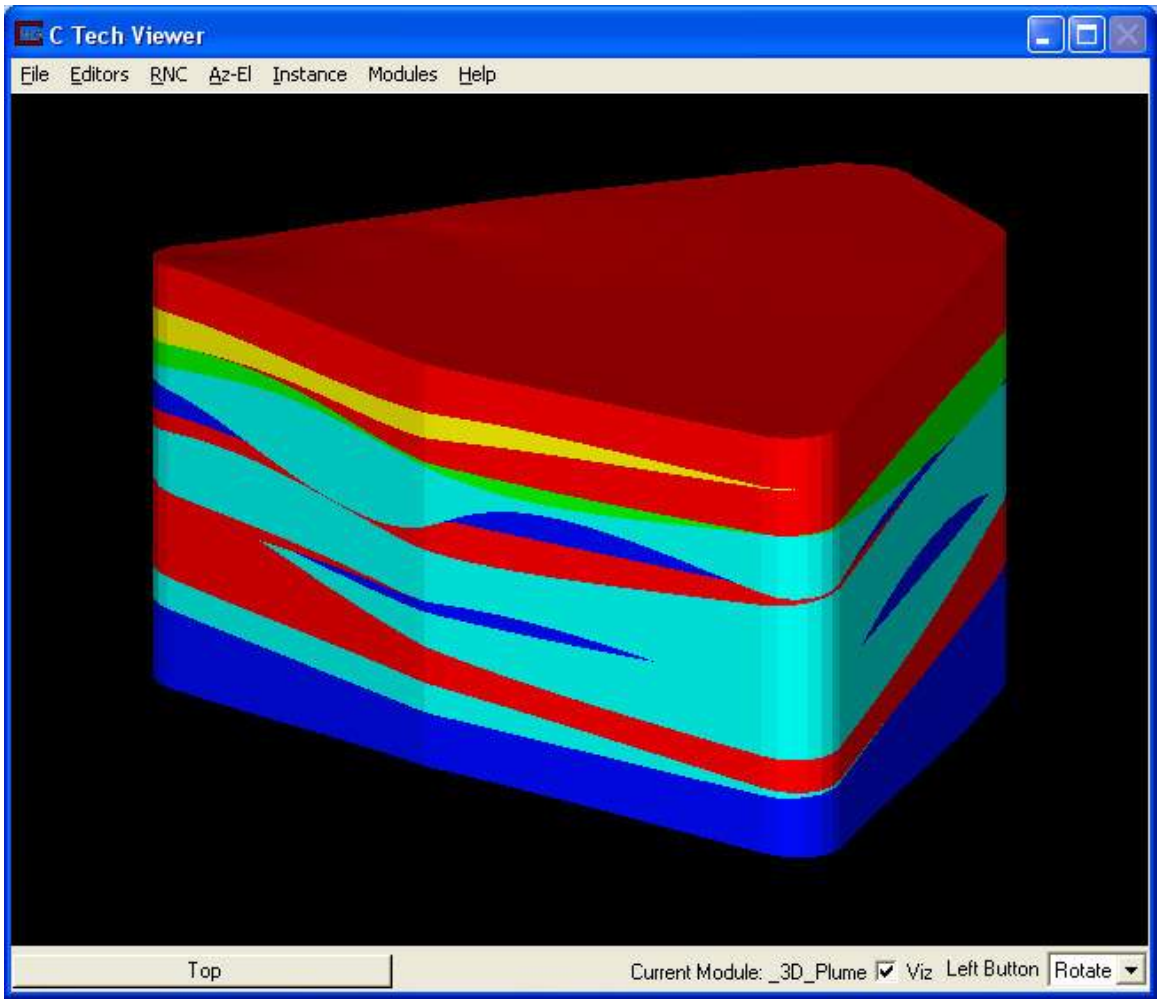
Set plume_shell to match:



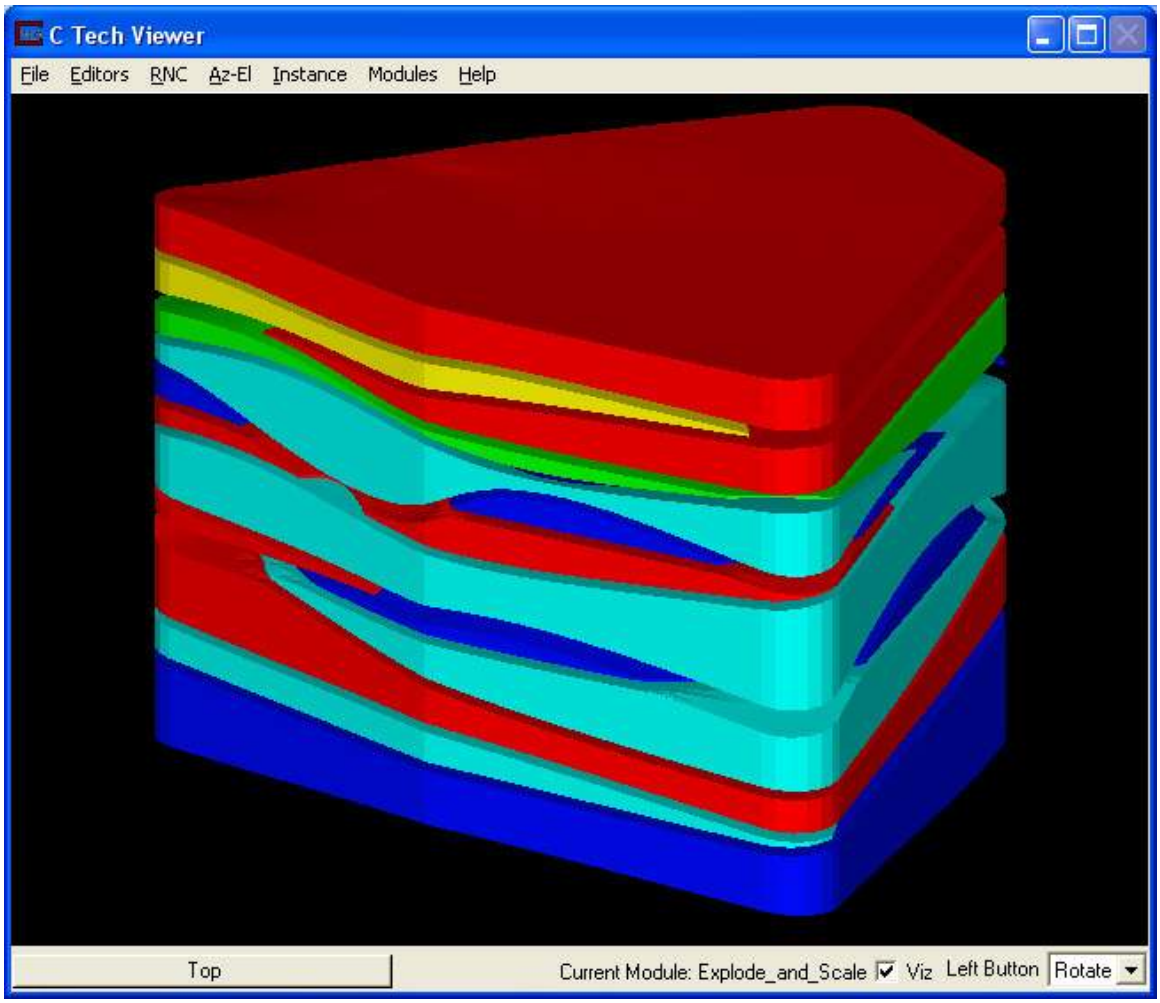
Set the parameters in Azimuth & Elevation to match:



and your viewer should look like:



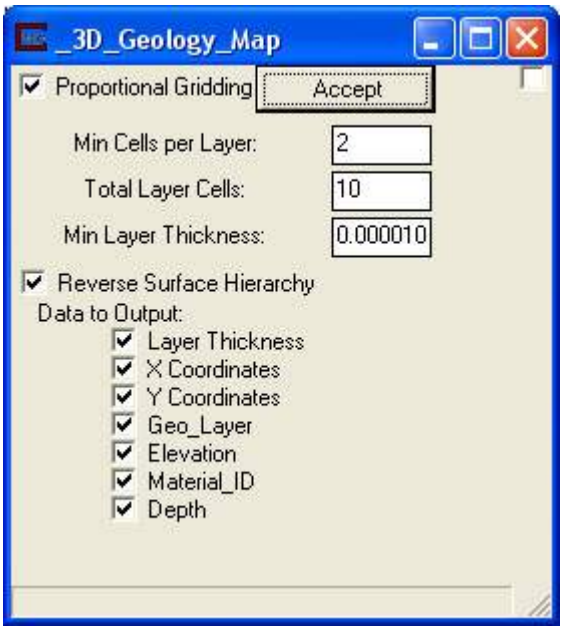
This same model shown with the geologic layers separated (exploded) by only 3 feet clearly reveals the pinched out layers.



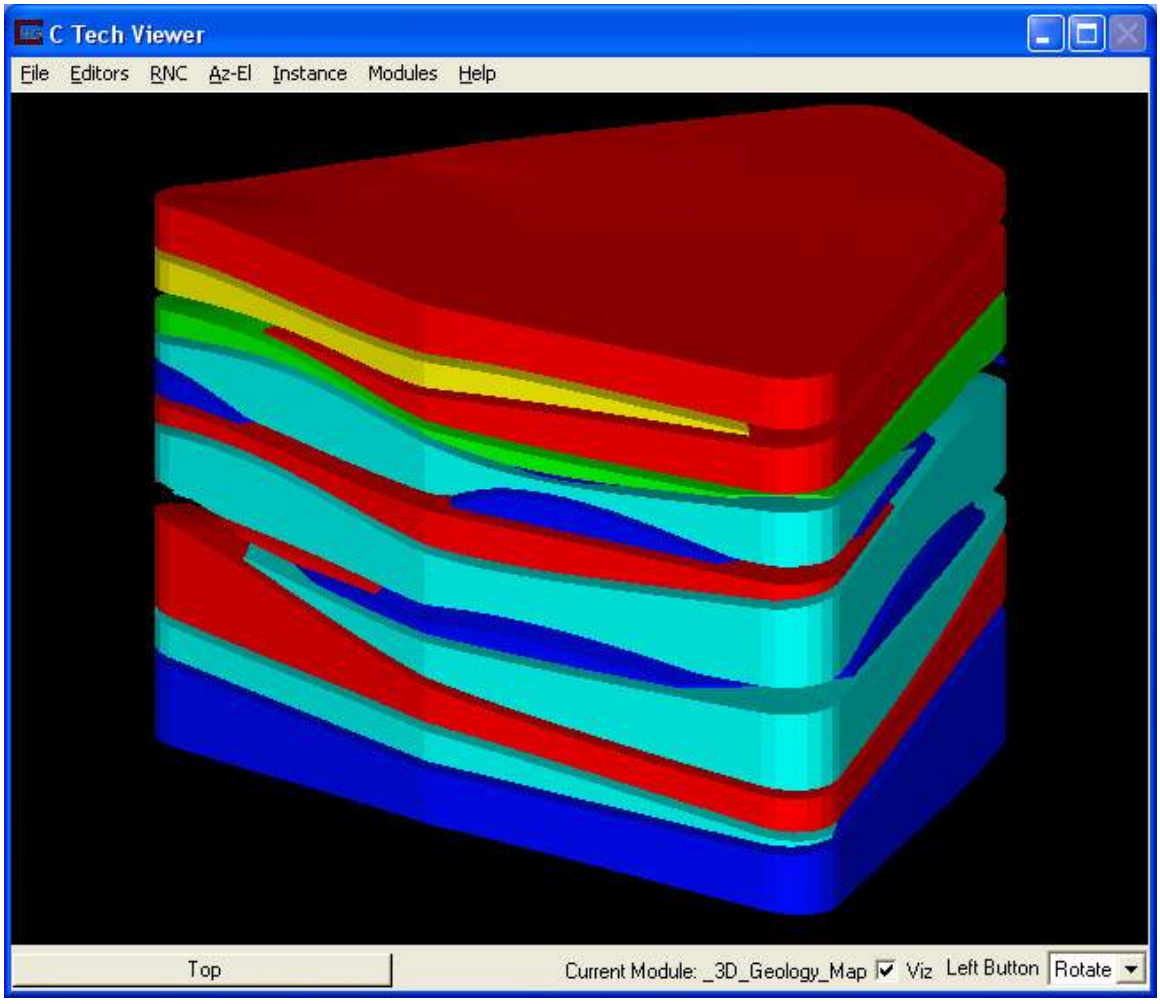
Reversed Surface Hierarchy

Because we are using Spline_Geology with this GMF file some of our splined surfaces actually violate the concepts of geologic hierarchy. However this is not a problem, because modules line 3D_Geology_Map and Krig_3D check the surface intersections and re-establish hierarchy based on one of two rules: **Normal and Reversed Hierarchy.**

With no other setting changed, open the module panel for 3D_Geology_Map and select the ***Reverse Surface Hierarchy*** toggle and hit ***Accept***.



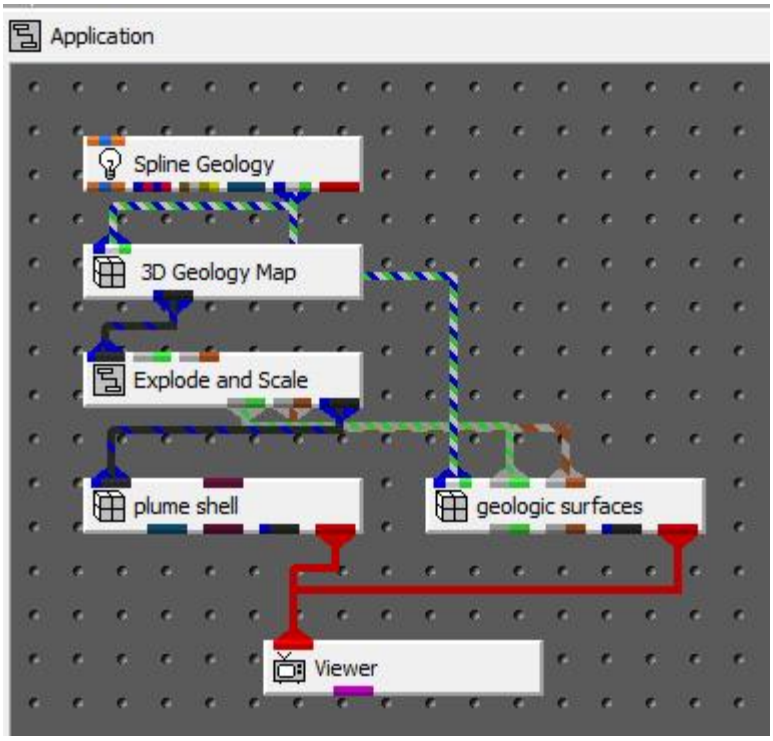
After a few seconds the model is regenerated and your viewer should show:



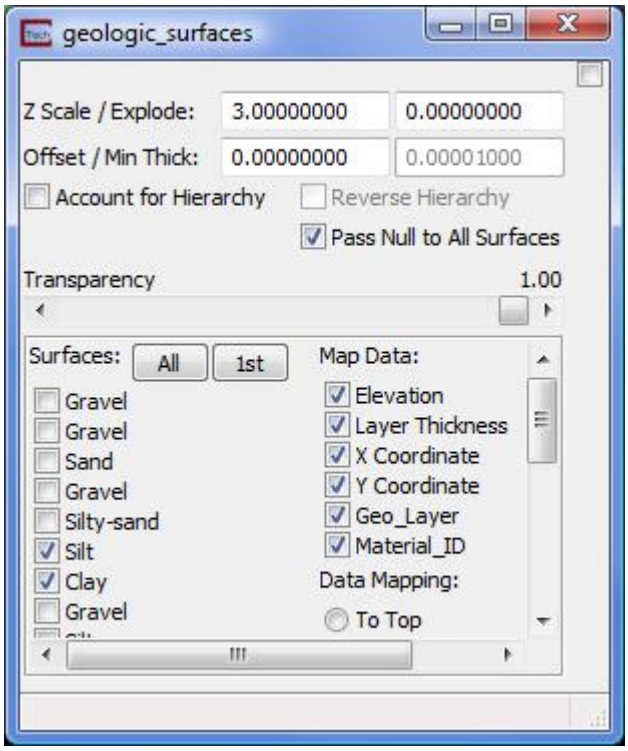
Take some time to compare this model to the Viewer shown above it with normal hierarchy. In most areas the two models are identical, but there are some significant differences.

If none of the surfaces created by Spline_Geology intersected one-another, the two hierarchy approaches would be identical. In a perfectly defined hierarchical model, surfaces should not intersect, however as modeler we seldom have enough data to build **good** models let alone perfect ones.

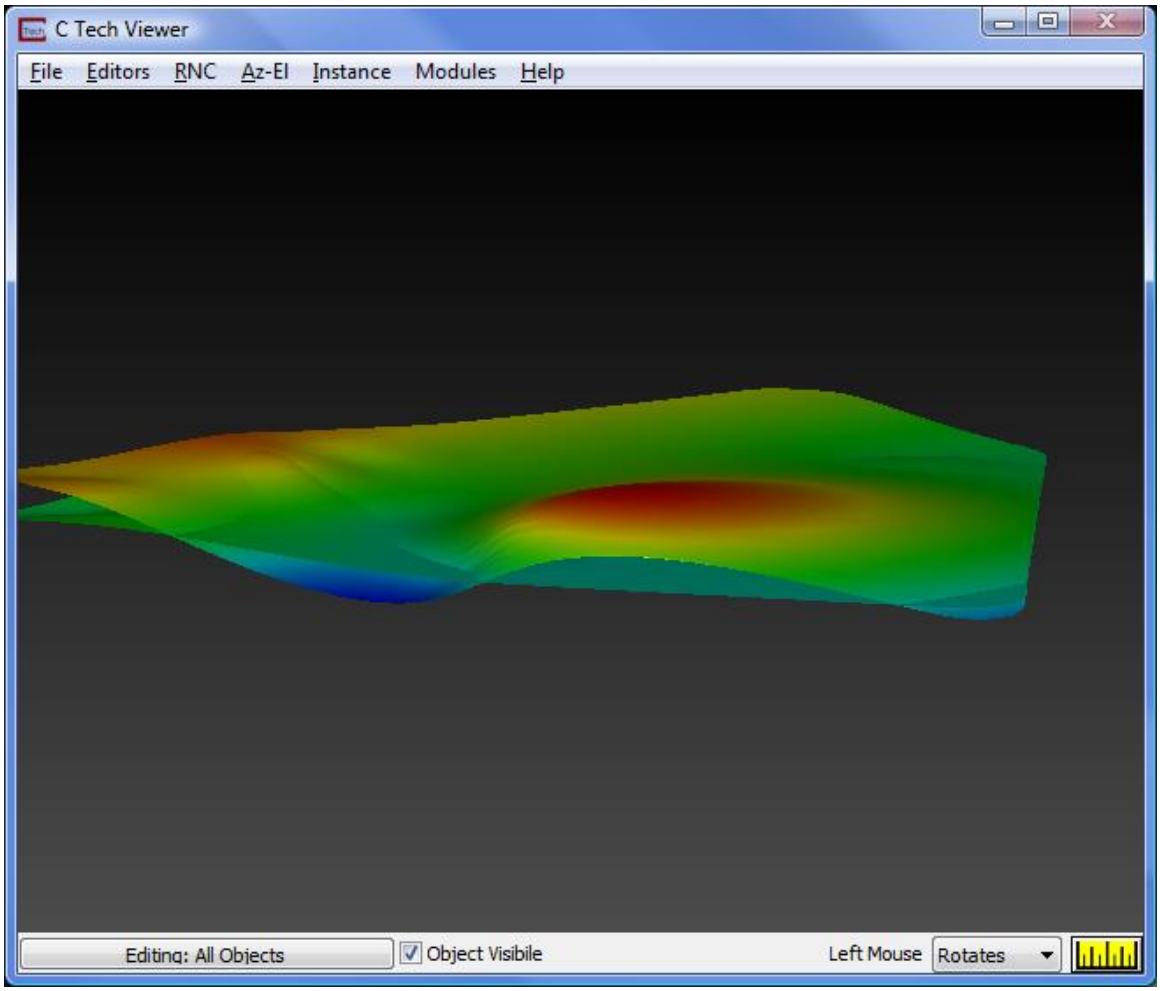
In order to see that we do have intersecting surfaces we'll modify our network (application) by adding a geologic_surfaces module.



Set its parameters to match:



Your viewer should now show two surfaces interesting. One was displayed uncolored to make the intersections more obvious.



Actually the surfaces splined here have many intersections. We leave it to the student to investigate them further.

However, returningto the subject of normal and reversed hierarchy.

With **normal hierarchy**, the grey surface would be forced coincident with the colored surface (i.e. the **lowergrey surface moves downwhere it penetrates above**). This normal hierarchy process of **forcing** surface elevations starts at the top and continues downward.

With **reversed hierarchy**, the colored surface would be forced coincident with the grey surface (i.e. the **uppercolored surface moves upwhere it penetrates below**). This reversed hierarchy process of **forcing** surface elevations starts at the bottom and continues upward.

By creating surfaces with intentional intersections (penetrations) the size of lenses and where they pinch-out can be more accurately controlled.

Indicator_Geology-Indicator Kriging

The most notable difference using Indicator_Geology to perform Geologic Indicator Kriging (GIK) as compared with hierarchical layered models is that the boundaries between materials are blocky. This is the result of using a totally different approach. With GIK, a default grid is created (with a user-

specified fineness). This grid can be (and is in this case) defined between two geologic surfaces (e.g. ground surface and the bottom of gravel). There are significant advantages to constraining the GIK to a top and bottom surface. First (and foremost) is that the ground surface of the resulting geologic model WILL follow the actual terrain. The single layer (two surfaces) geology file that was used to constrain the model domain in this workbook topic is 5_material_complex_geology_single_layer.gmf. The content of this file is shown below. Note that only four borings were deep enough to define the bottom of the model.

```
# Single layer 5_material_complex_geology GMF file
# Surface #0
units meters
surface 4 Gravel
2084941.150002 323298.839996 33.000000 OW-1A
2084941.290001 323310.529999 32.000000 OW-1B
2084948.070000 323309.399994 32.000000 OW-1C
2085079.220001 323361.000000 32.000000 MW-10B
2085108.470001 323363.890015 32.000000 MW-10A
2085171.220001 323387.250000 31.000000 X4-B
2085184.589996 323397.190002 32.000000 X4-A
2085266.929993 323410.049988 32.000000 MW-11A
2085274.820007 323407.700012 32.000000 MW-11B
2085153.839996 323253.399994 31.000000 X-2A
2085176.869995 323061.139999 31.000000 X-1B
2085192.220001 323066.870003 31.000000 X-1C
2085194.259995 323052.759998 31.000000 X-1A
2085487.119995 322869.949997 31.000000 AW-3
2085634.320007 323366.940002 31.000000 OW-4B
2085640.820007 323381.970001 31.000000 OW-4A
2085629.750000 323379.660004 31.000000 OW-4C
2085553.919983 323379.029999 31.000000 MW-12B
2085538.950012 323379.769989 31.000000 MW-12A
2085555.890015 323523.869995 31.000000 OW-3B
2085547.710022 323517.630005 31.000000 OW-3C
2085536.450012 323527.400024 31.000000 OW-3A
# Surface #1
surface 0 Clay
2084948.070000 323309.399994 -105.950002 OW-1C
2085192.220001 323066.870003 -102.950002 X-1C
2085547.710022 323517.630005 -107.950002 OW-3C
2085629.750000 323379.660004 -115.950002 OW-4C
end

AEC9615FC43E7EB25ECDD17298166091
```

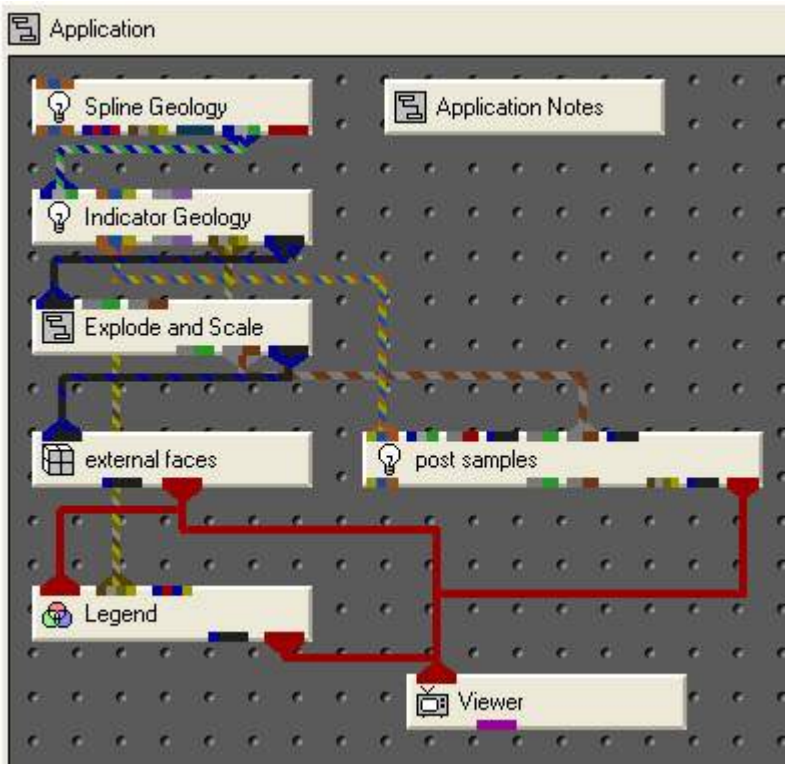
The Indicator Kriging process uses a .pgf file as DATA for the material assignment. It does not use it to constrain the extent of the model creation. We will be using 5_material_complex_geology.pgf.

Indicator_Geology includestwodifferent methods for assigning geologic data to cells in your model grids.

1)The*Quick Method*is similar (but improved) to the Geologic Indicator Kriging that was in Krig_3Dprior to version 6.0. It assigns the geologic material cell data based on the nearest geologic material (in anisotropic space) to your PGF borings. This is done based on thecoordinateof thecell'scenter and an enhanced refinement scheme for the PGF borings.

2)Indicator_Geology also includes a rigorous probabilistic approach to geologic indicator krigingthat will be discussed in thenext topic.

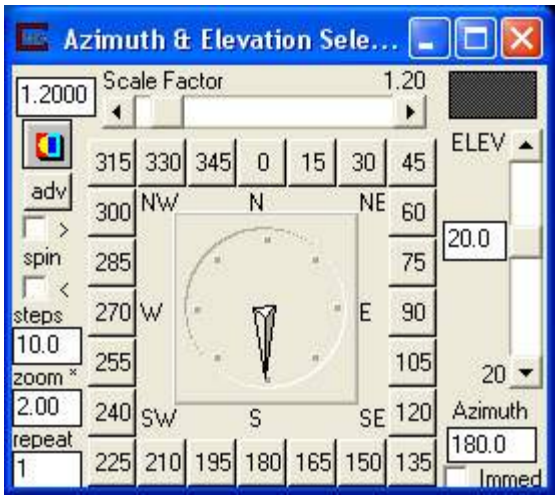
The application we will be running in this topic uses the ***QuickMethod*** which runs quite fast. It can be loaded from ctech\applications\pro\5_material_complex_indicator_geology_quick.v. The network is shown in the figure below:



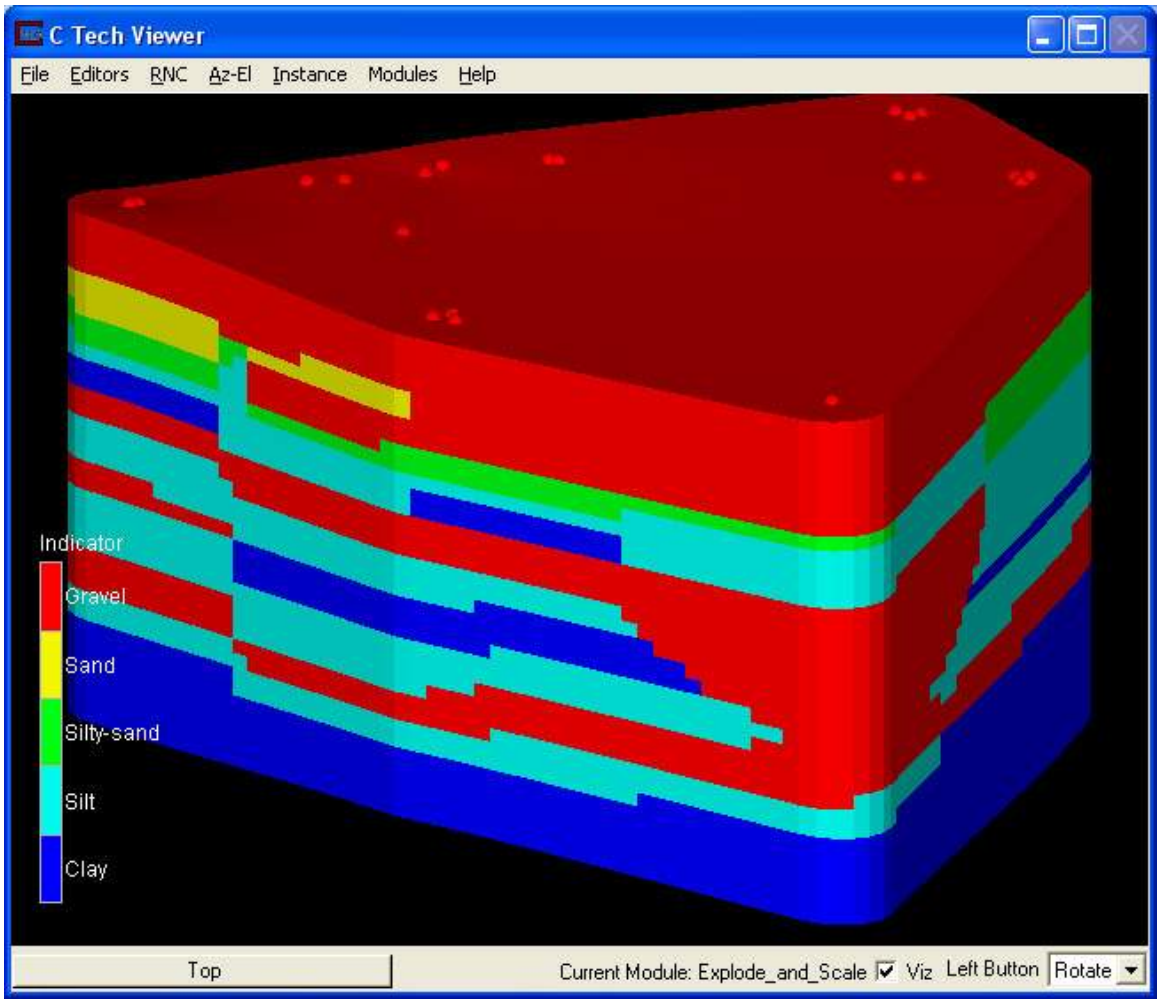
The finer the grid (especially in the vertical direction) the more accurate the representation. GIK assigns the most probable material to each CELL in the model. The assignment to a cell is a significant distinction as compared with all other interpolation performed in EVS/MVS. If geologic data were assigned to nodes (vs. cells), in areas where there is a transition between two non-sequential material id's, colors would be interpolated between them that could correspond to intermediate materials. Using the cell data paradigm, we

can have regions that transition from Clay to Gravel without seeing any other materials interpolated in between.

Set the parameters in Azimuth & Elevation to match:

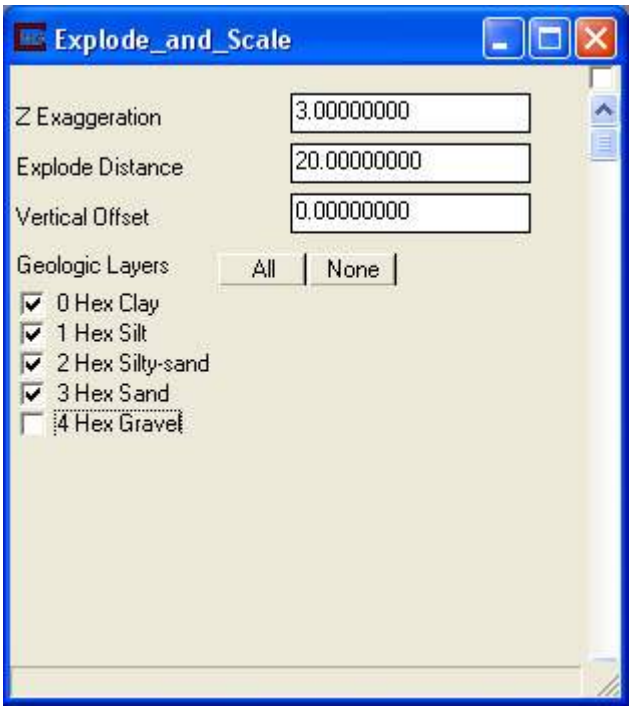


Then turn on all of the materials in the *Geologic Layers* section in Explode_and_Scale. Your Viewer should show the GIK model produced with the ***QuickMethod***.

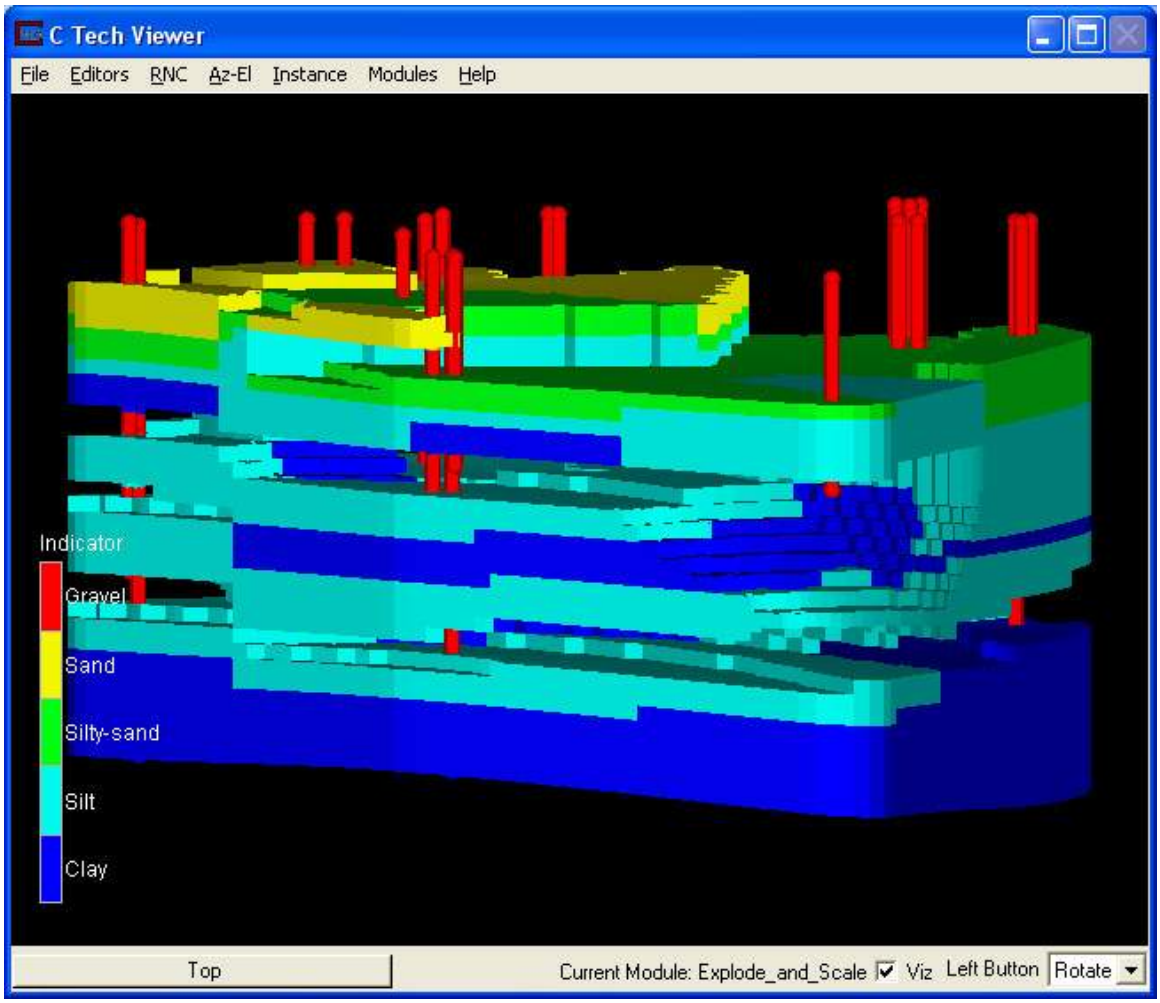


If you compare this output to the unexploded model in the previous topic you will see many similarities **and** differences. The most notable area of differences in the near right corner of the model. Let's investigate why this area is so different.

If we turn off the Gravel material:



and change the elevation of the view to 5.0 degrees the reason for the differences are obvious! With the Gravel removed, the nearest boring is revealed to be very short. It was stopped just after it entered the gravel. With the hierarchical approach the subsequent surfaces were extrapolated into this region without any data to support their elevations. Similarly with the GIK approach the material assignments below the boring are based on other borings (far away!) and consequently the gravel extrapolates downward a significant distance in the absence of better data!



So which is right?

The answer is that we don't have any data to determine which method is more correct. Both approaches give us some insight into the geology of this site and the GIK approach makes our data shortcomings quite obvious.

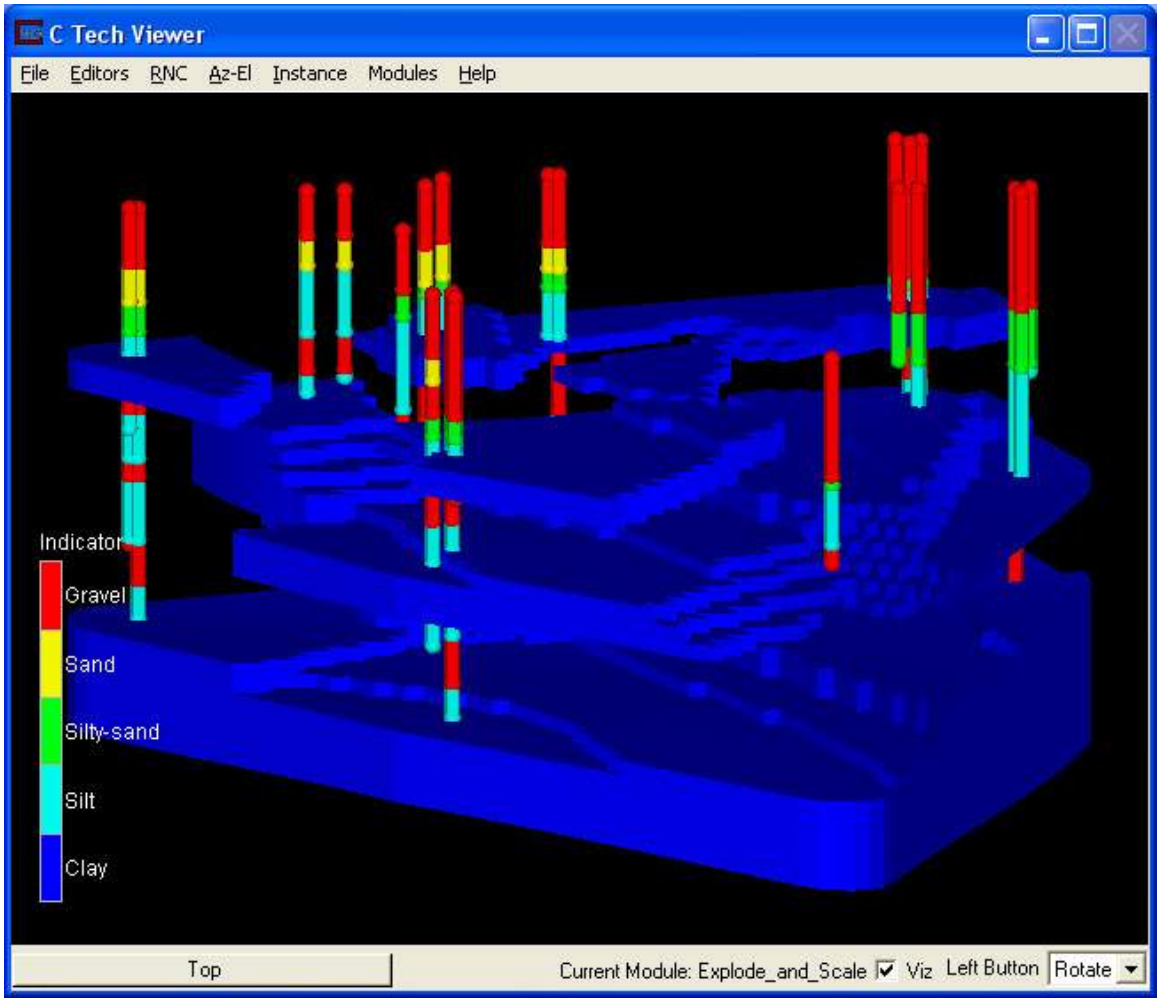
Also note that regions produced with GIK can have vertical transitions and folds. It is not encumbered by the layered approach of geologic hierarchy. Similarly it does not provide any means to *explode layers*. Moreover the entire concept of layers is meaningless with GIK. There are not layers represented by a GIK model, only materials.

If the resulting model shows clear evidence of layering it is a good indication that we should probably be using the hierarchical modeling approach instead!

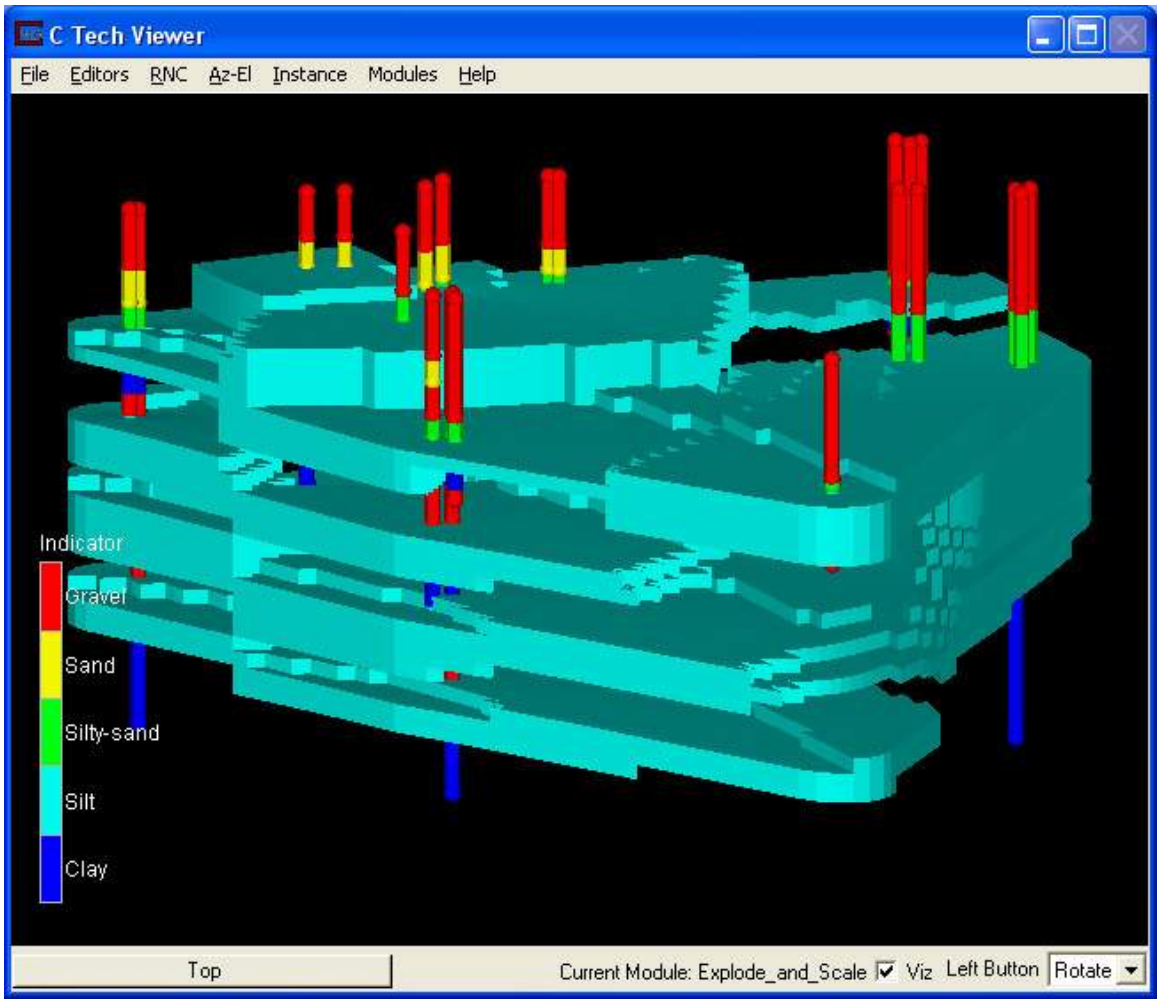
Rigorous investigation of the geology of a site should generally include GIK in addition to hierarchical modeling (if appropriate).

The following five figures show the individual cell sets corresponding to each material. The visibility of each cell set is controlled using the toggles in the Explode_and_Scale module.

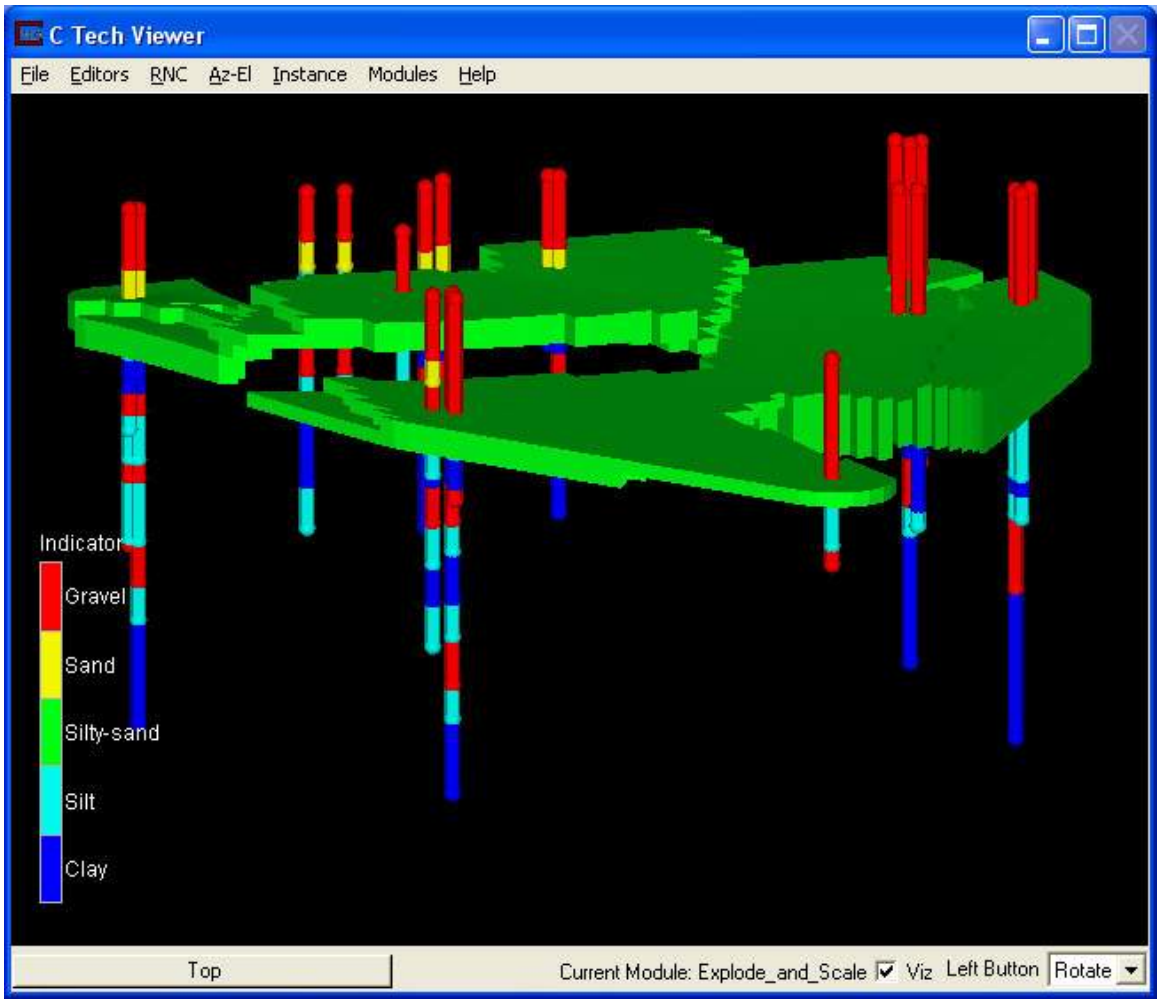
For the first material, the clay zones are shown below.



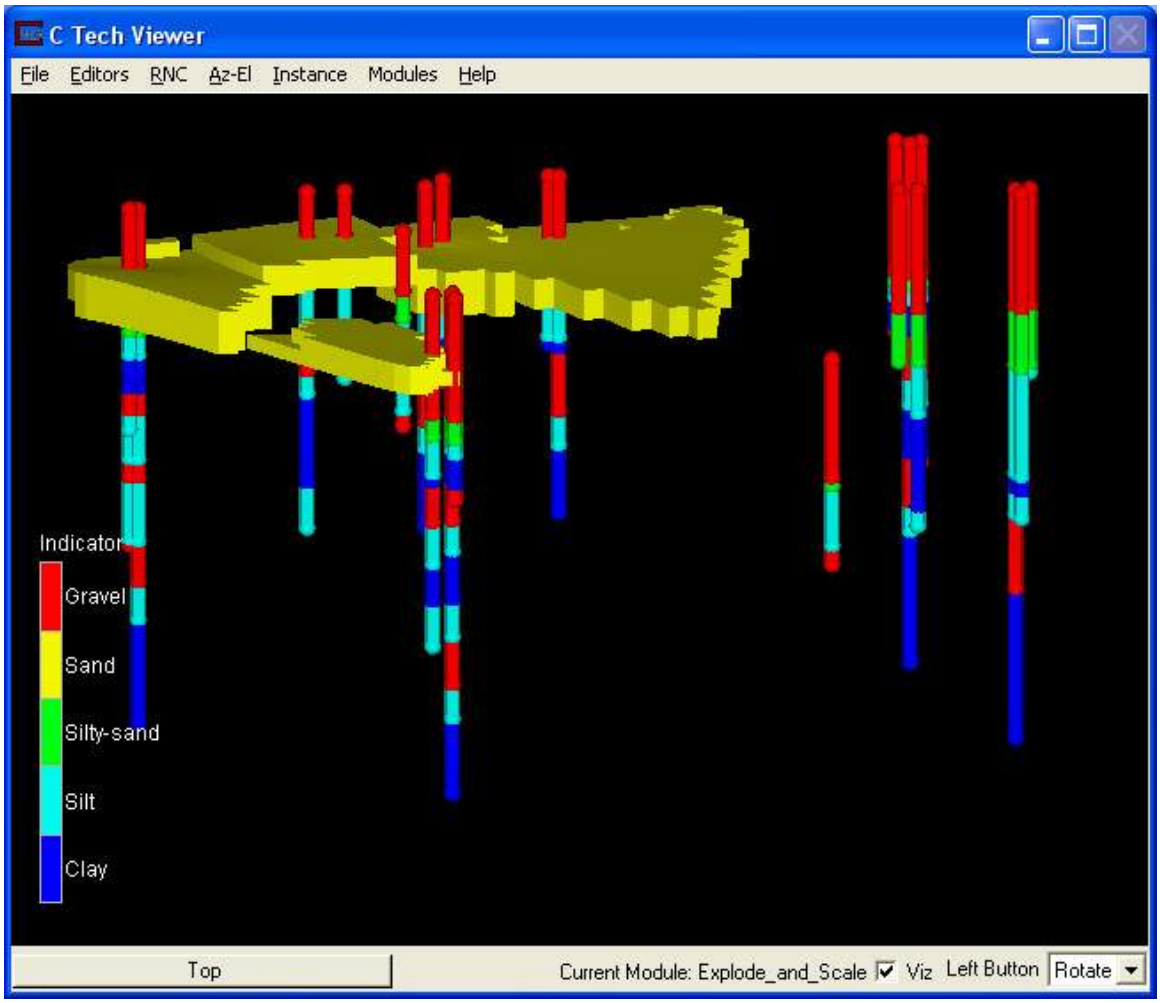
For the second material, the silt zones are shown below.



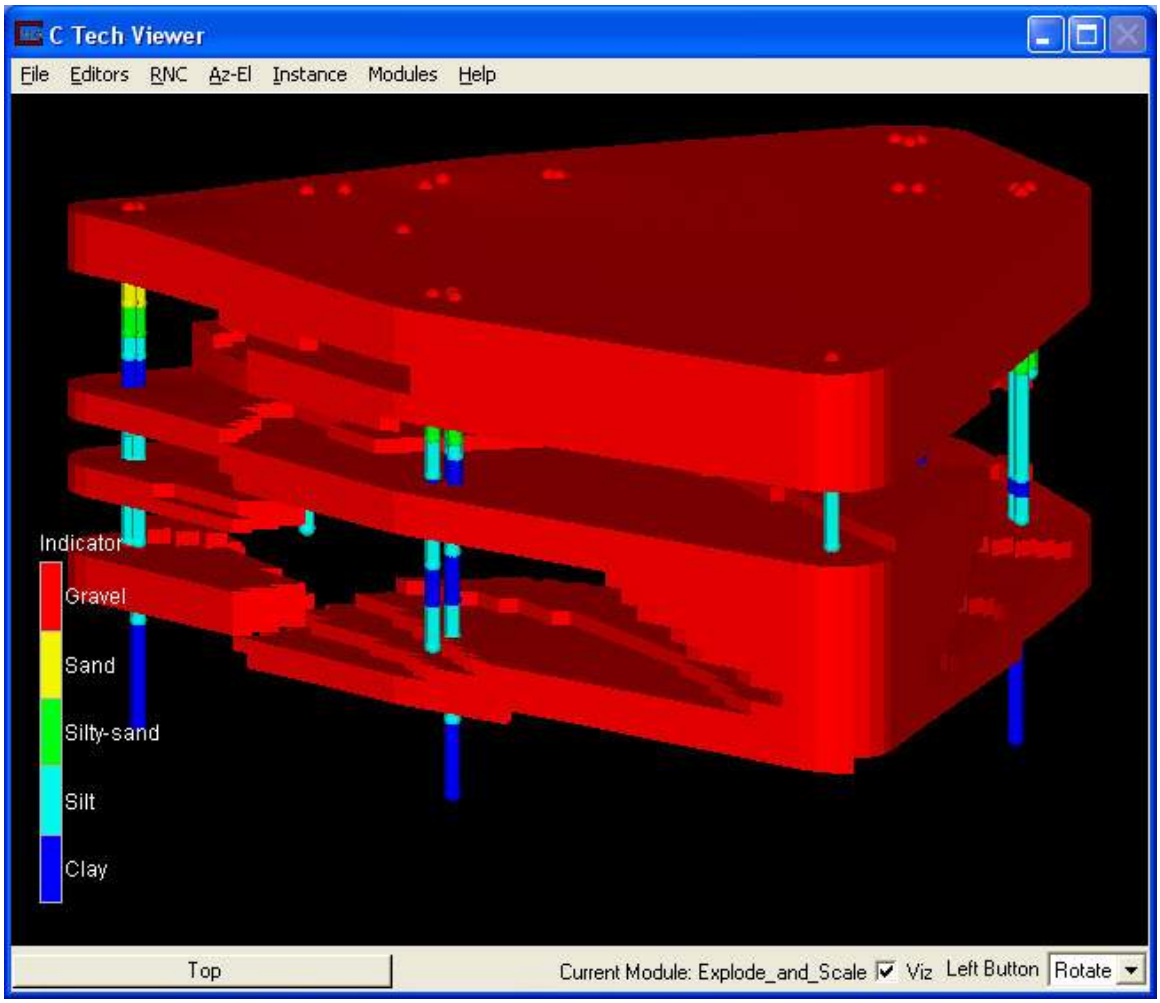
For the third material, the silt/sand zones are shown below.



For the fourth material, the sand zones are shown below.



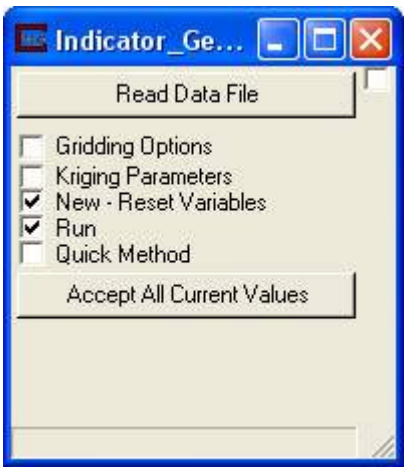
For the fifth material, the gravel zones are shown below.



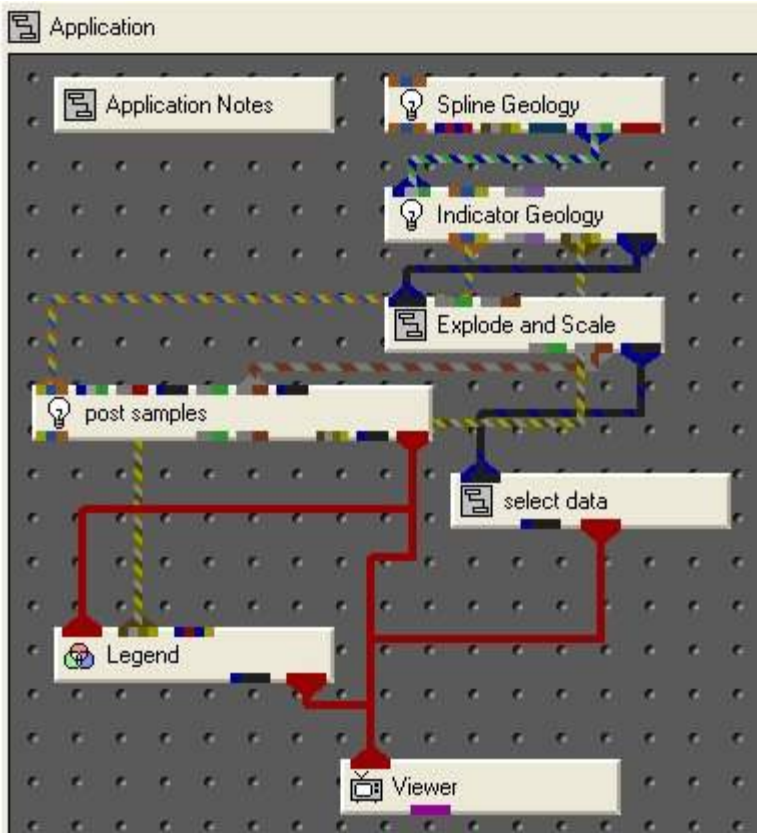
Indicator_Geology - Probabilistic Kriging

Indicator_Geology also includes a rigorous probabilistic approach to geologic indicator kriging. For this approach, the probability for **each** material is computed for every cell in your grid. The material having the highest probability (for an individual cell) is assigned to the cell. All of the individual material probabilities are also provided as additional cell data components. This will allow you to identify regions where the material assignment is somewhat ambiguous. Needless to say, this approach is much slower (especially with many materials), but often yields superior results and interesting insights.

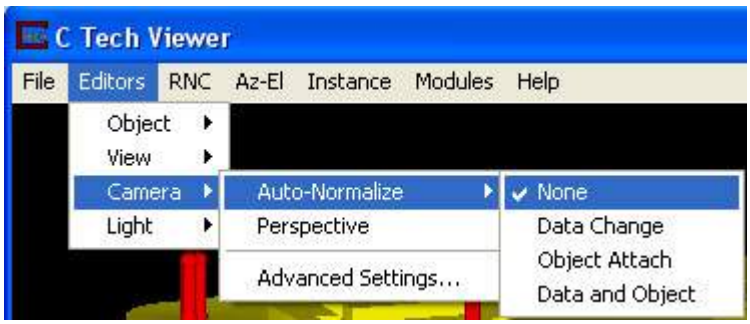
The application we will be running in this topic uses the Rigorous (not the Quick) Method which takes a little longer to run. It can be loaded from ctech\applications\pro\ 5_material_complex_indicator_geology_prob.v. It looks identical to the previous topic's application, except that the Quick Method toggle is **off.**



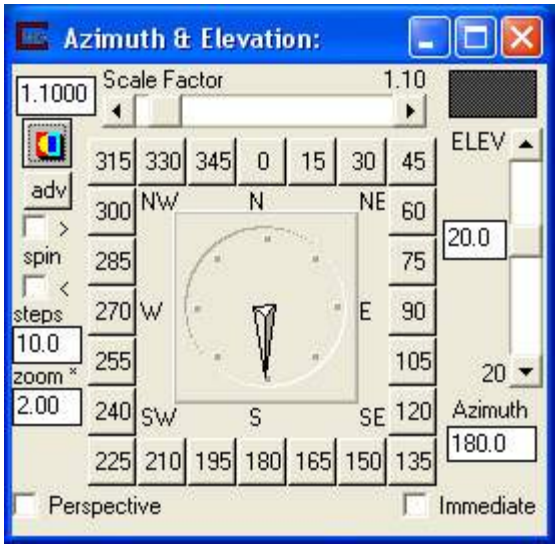
The network is shown in the figure below:



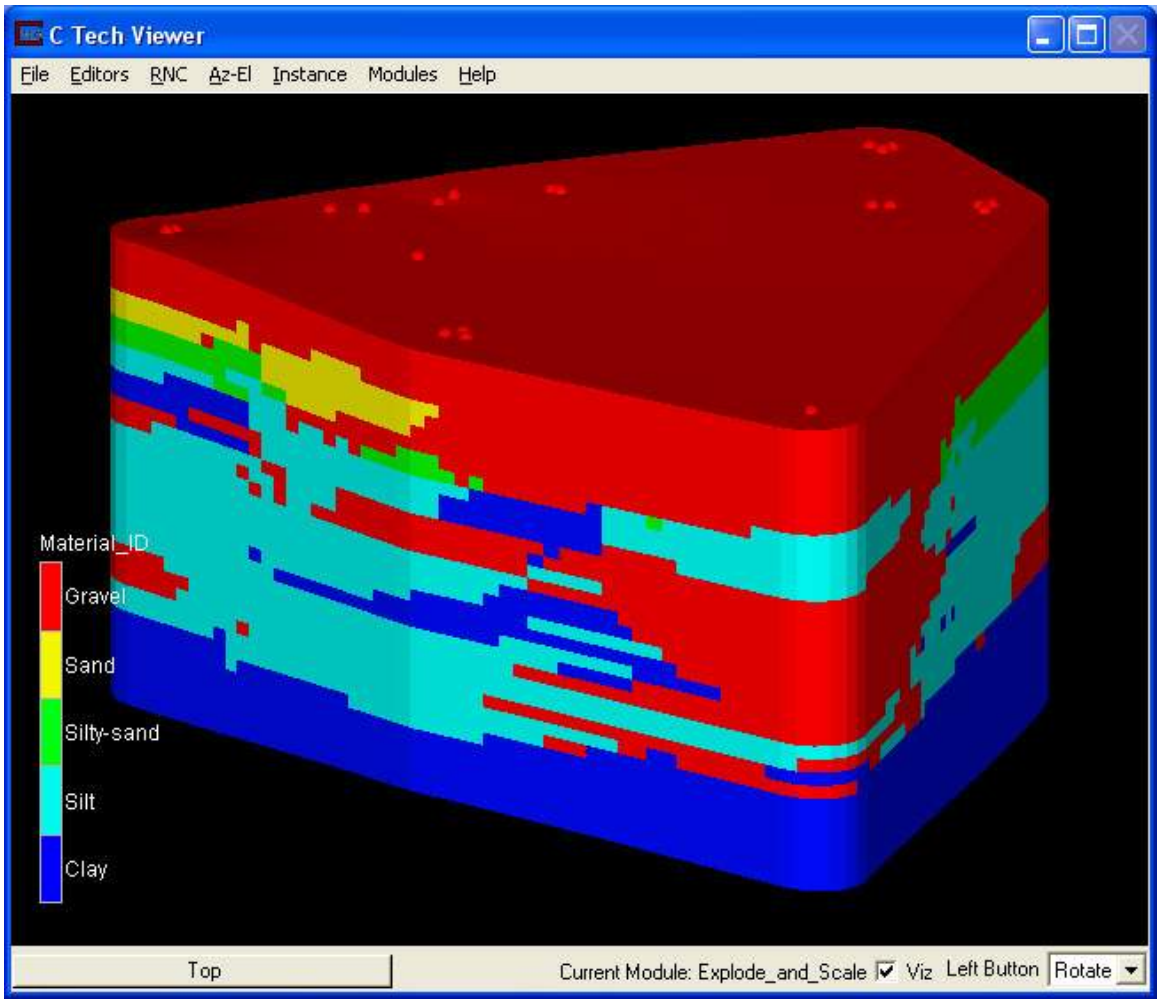
If you build this network instead of loading it, turn off Autonormalize in the Viewer to keep our view from changing when the data changes:



When this application runs, set the view to match:



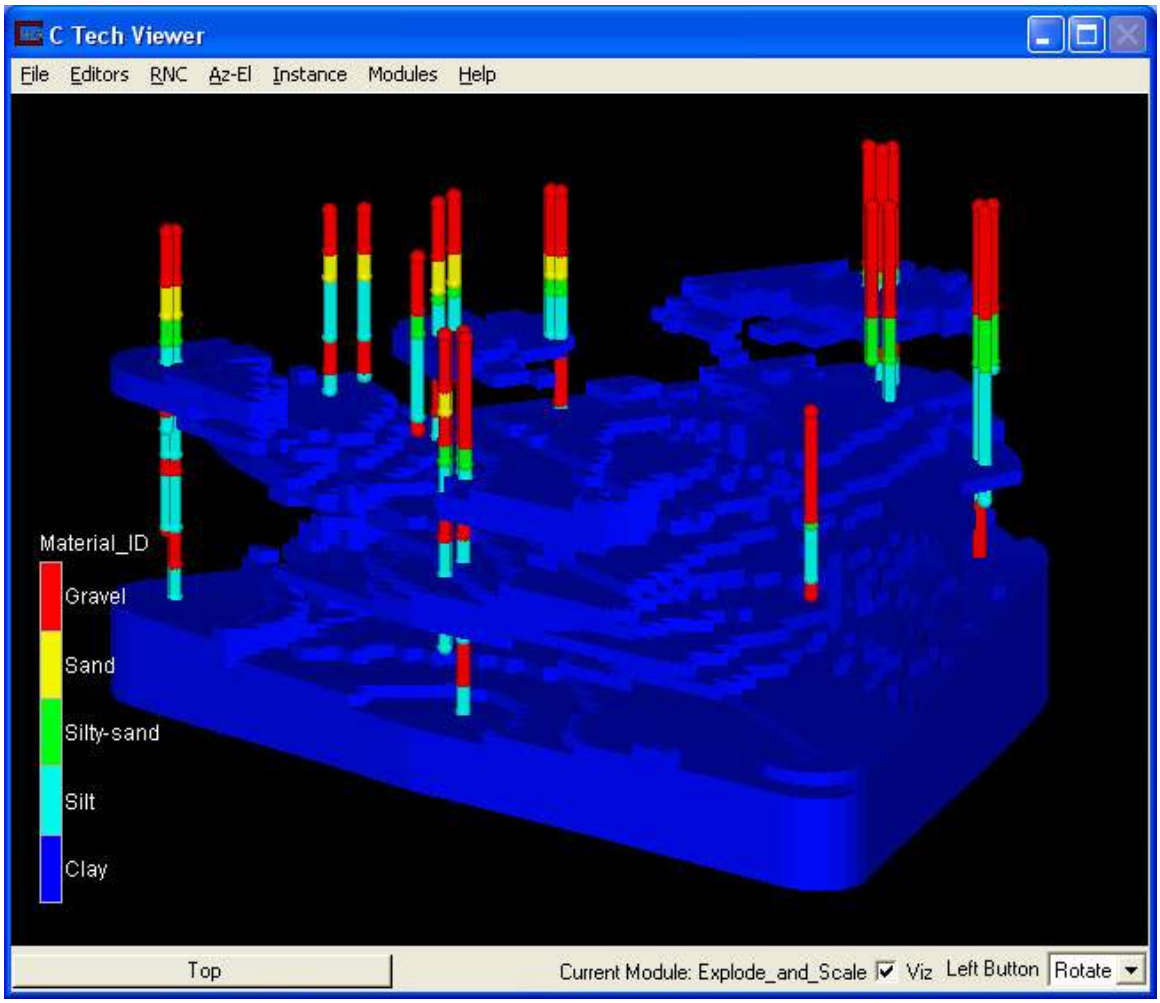
Turn on the remaining materials in Explode_and_Scale, and the resulting model shows:



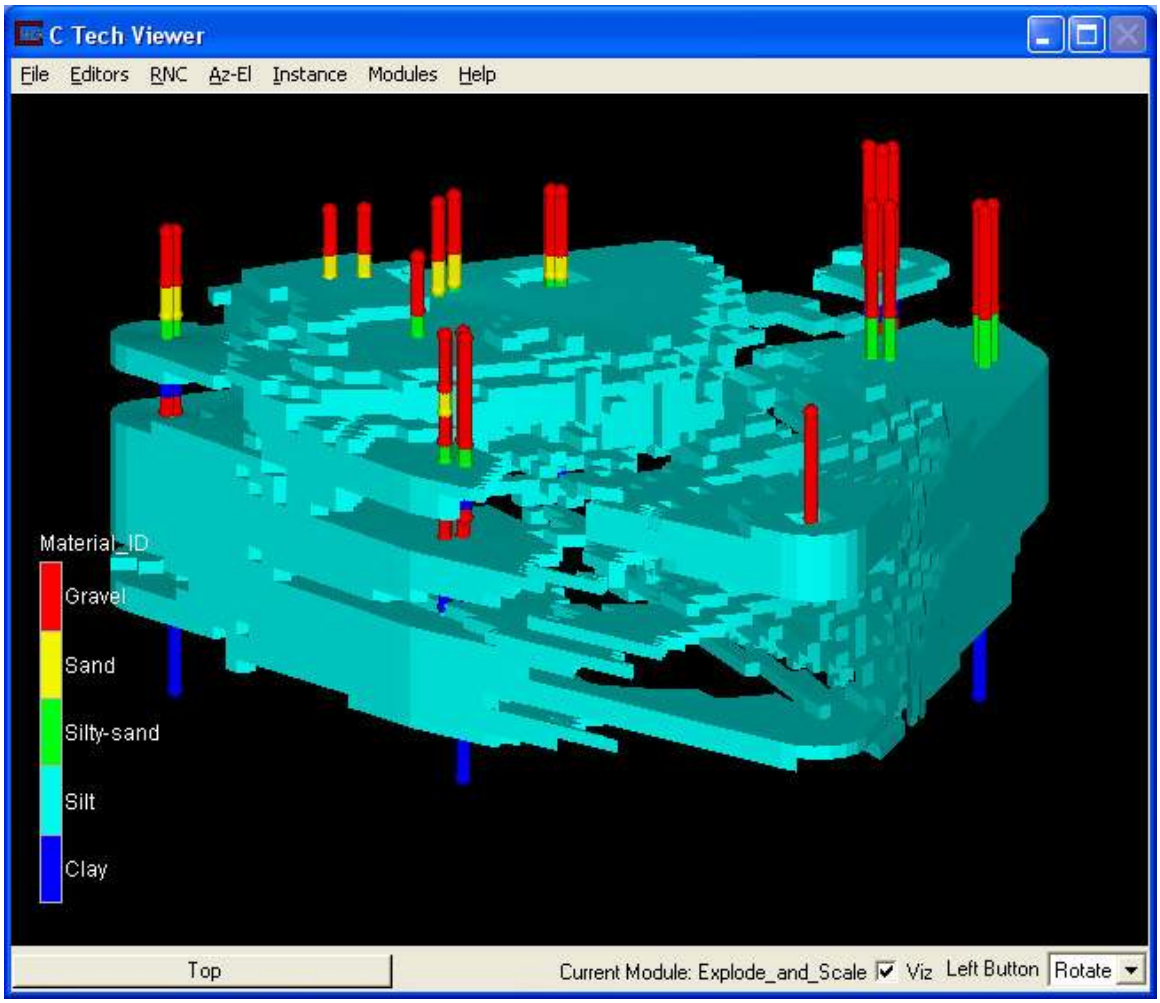
Note the similarities and differences as compared with the Quick Method. With the probabilistic method there seems to be more fine structure and small detail variation. This tends to occur only in areas where the confidence in the material assignment is low. This kriging approach gives us several ways to investigate this.

The following five figures show the individual cell sets corresponding to each material. The visibility of each cell set is controlled using the Explode_and_Scale module's *Geologic Layer* toggles (even though these are materialsnot layers).

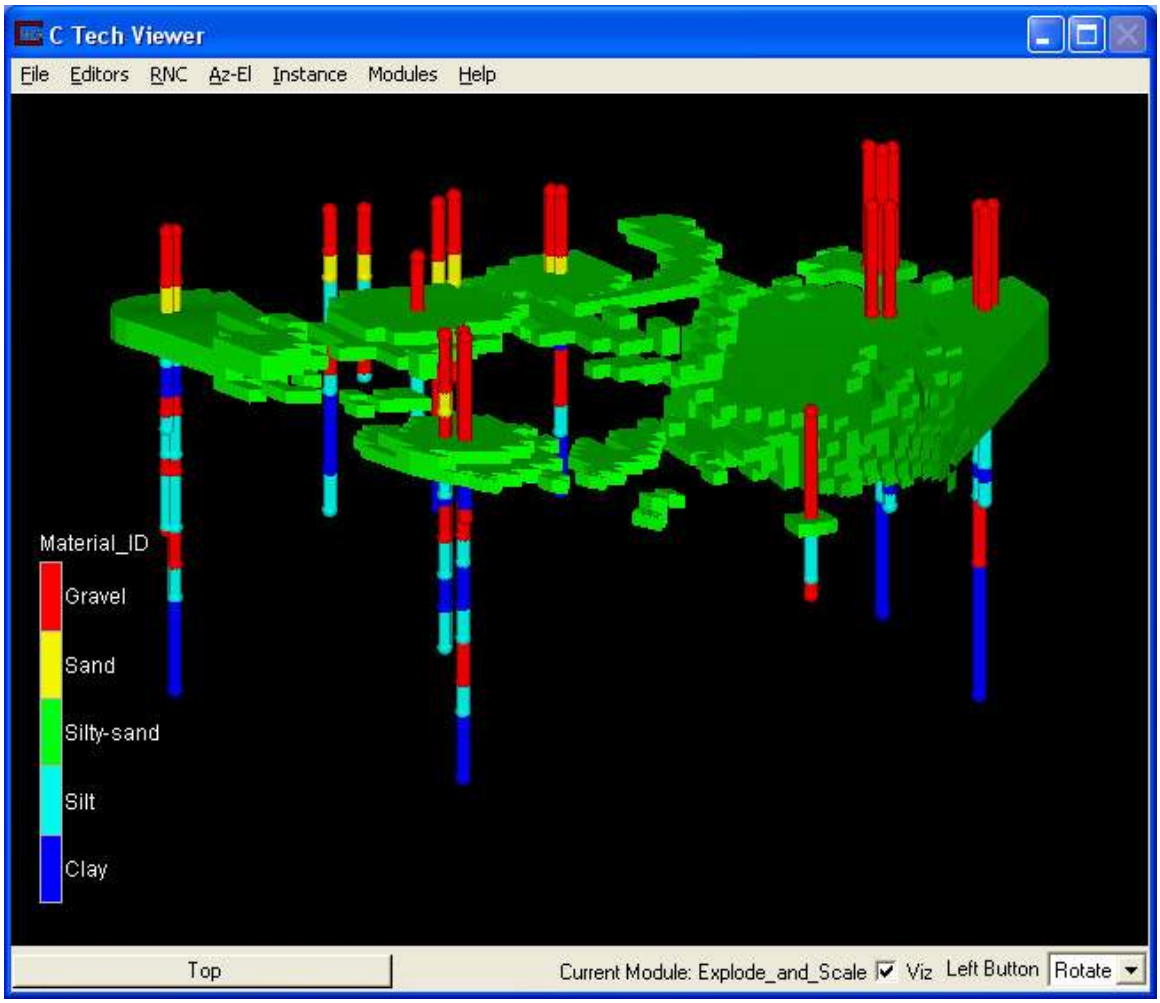
By selecting only Clay (the first material) in Explode_and_Scale, the clay zones are shown below.



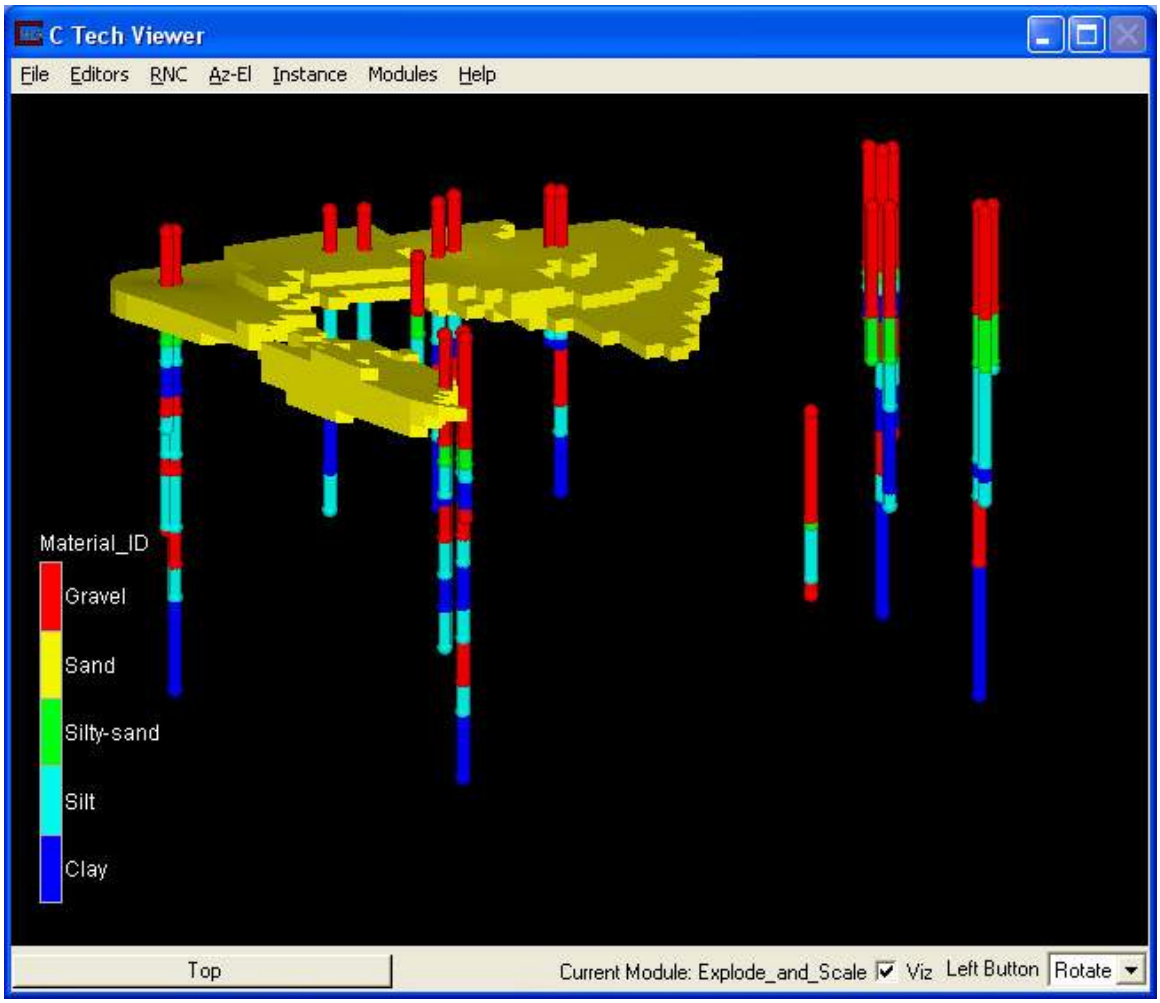
For the second material, the silt zones are shown below.



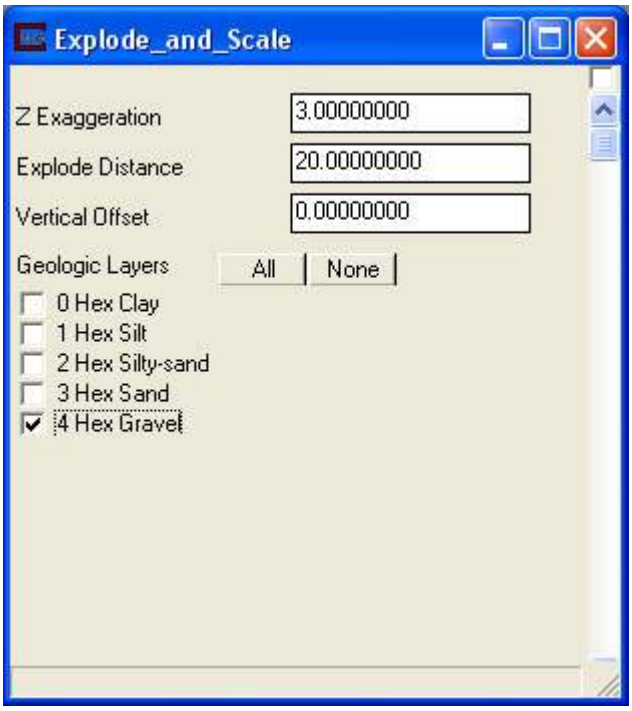
For the third material, the silt/sand zones are shown below.



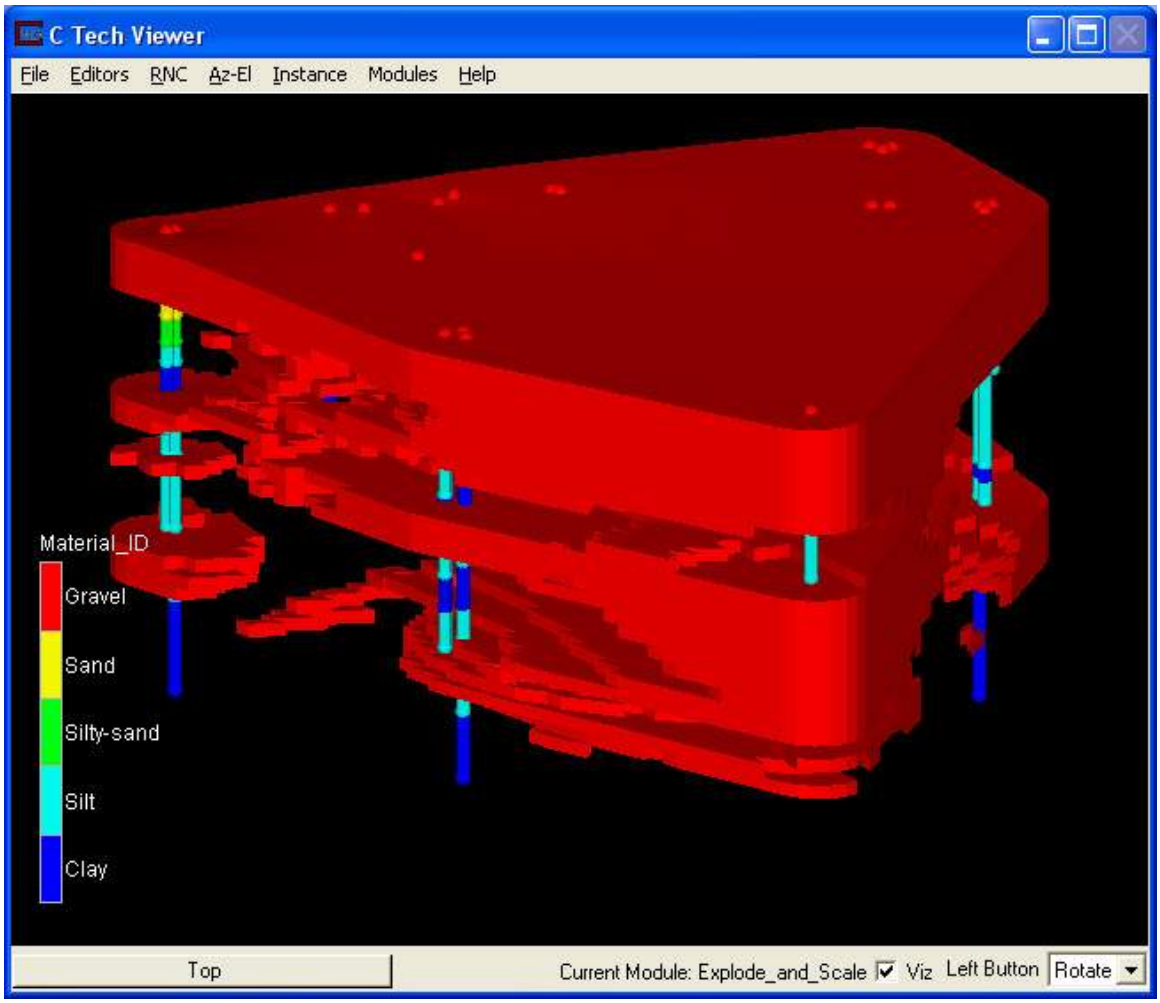
For the fourth material, the sand zones are shown below.



For the fifth material, Explode_and_Scale should look like:

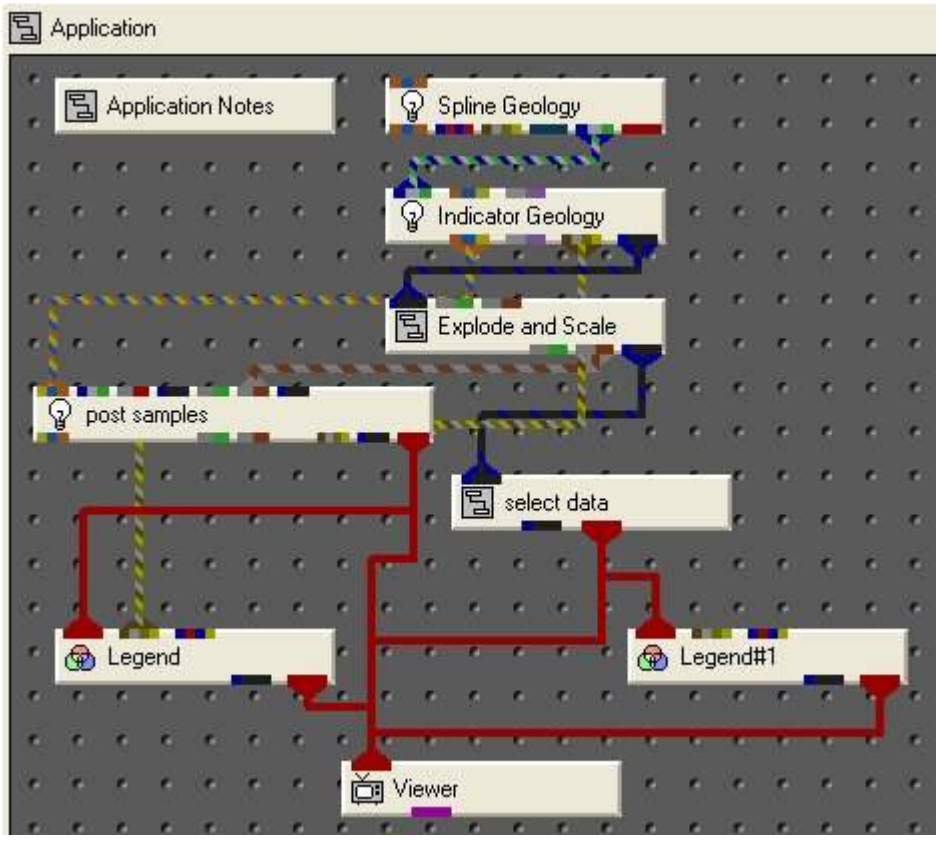


and the gravel zones are shown below.



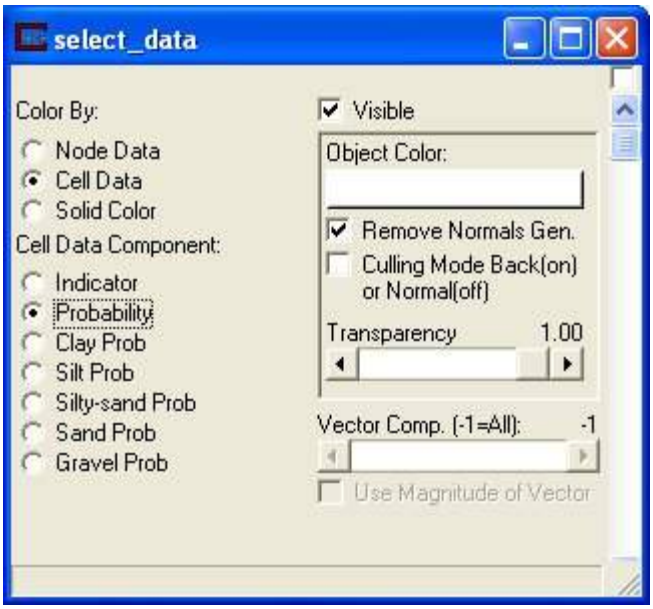
Indicator_Geology–MaterialProbabilities

Now, let's make a small addition to our network (application). We'll add a second Legend module because we're going to need a scale to show us what probabilities our colors represent. Disconnect the first Legend and add Legend#1 as shown below:

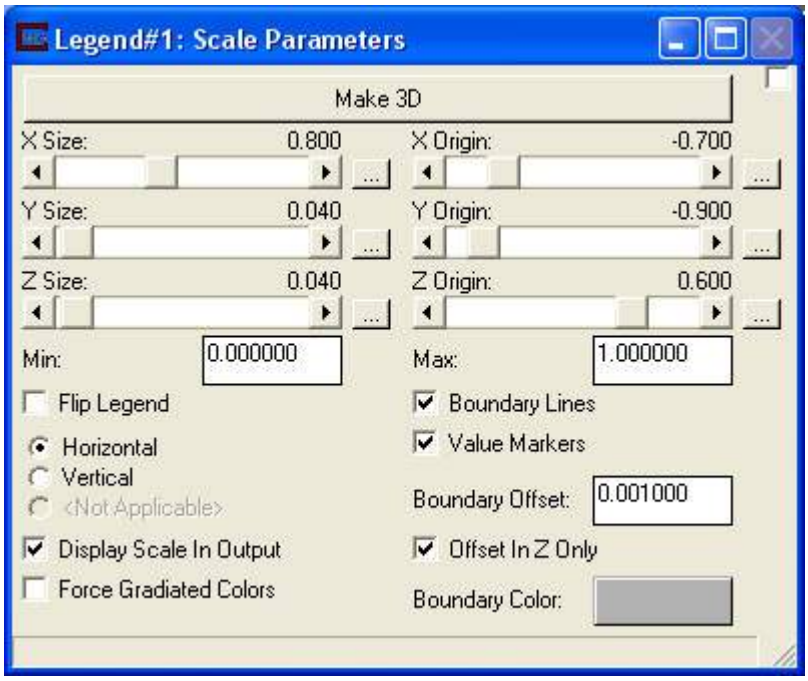


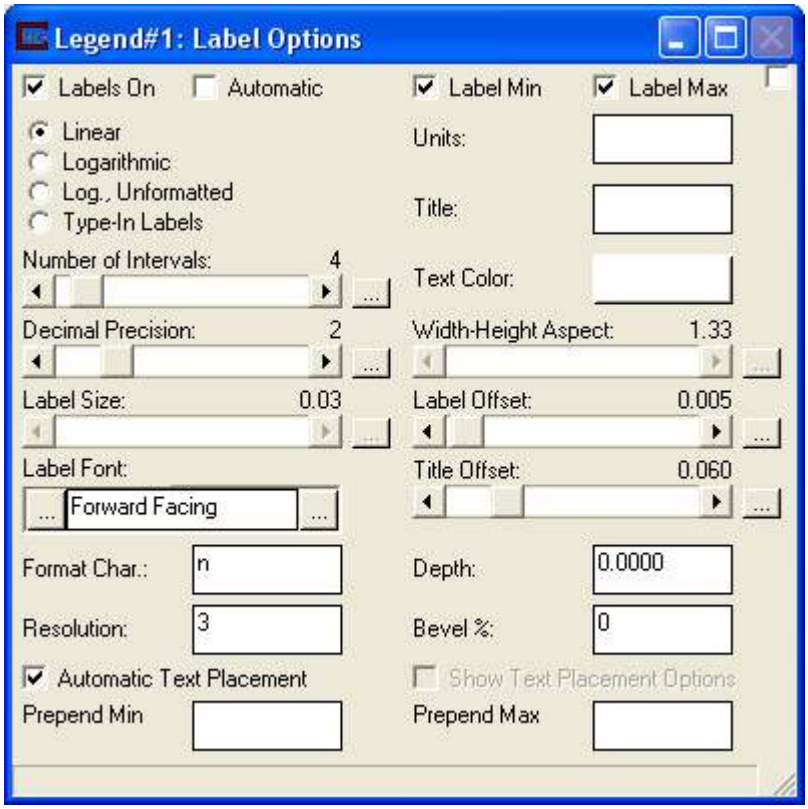
Turn on all materials in Explode_and_Scale.

Now open the select_data module's window and choose Probability:



Open Legend#1's two windows and match the settings below:

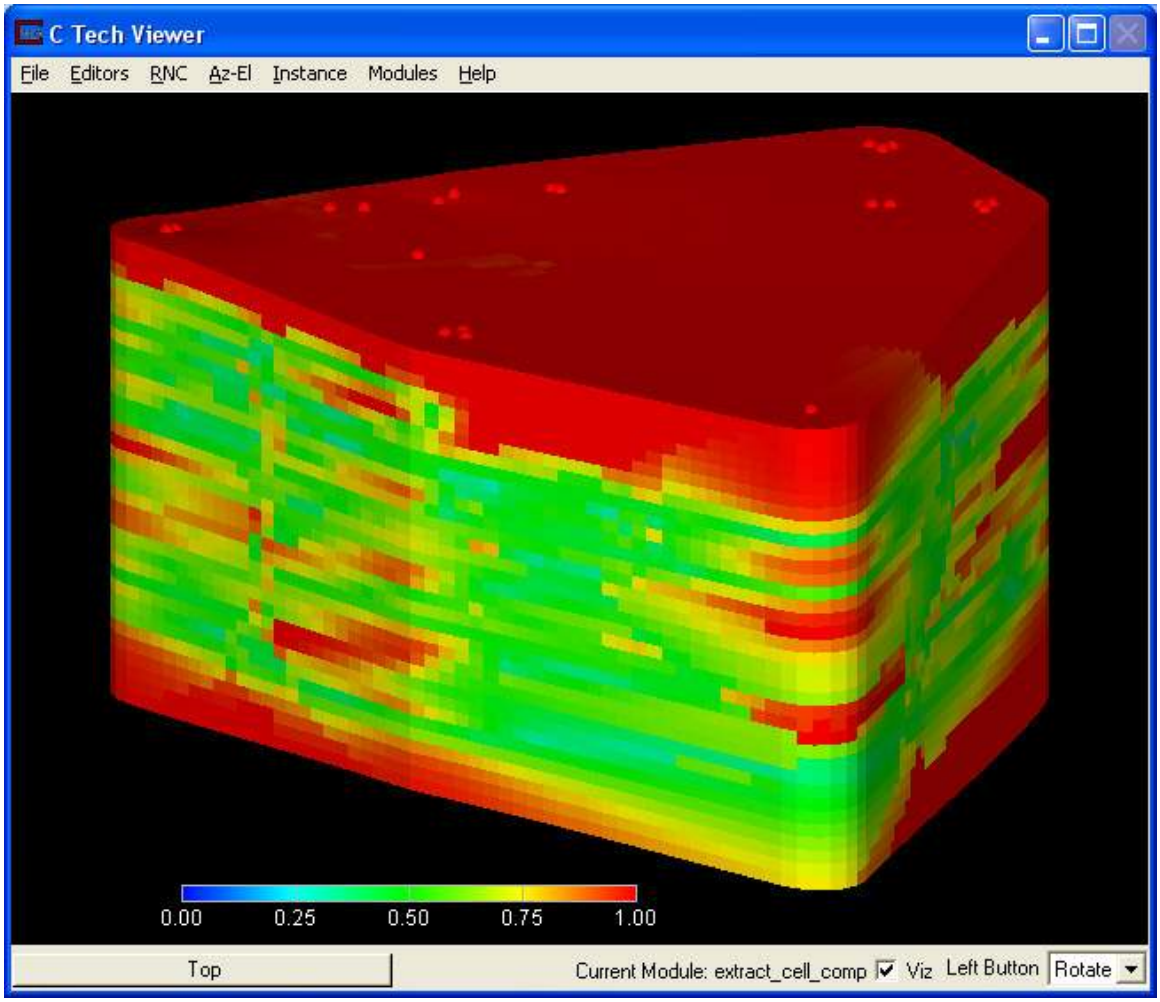




Click on *Accept All Current Values* and the second legend will be created.

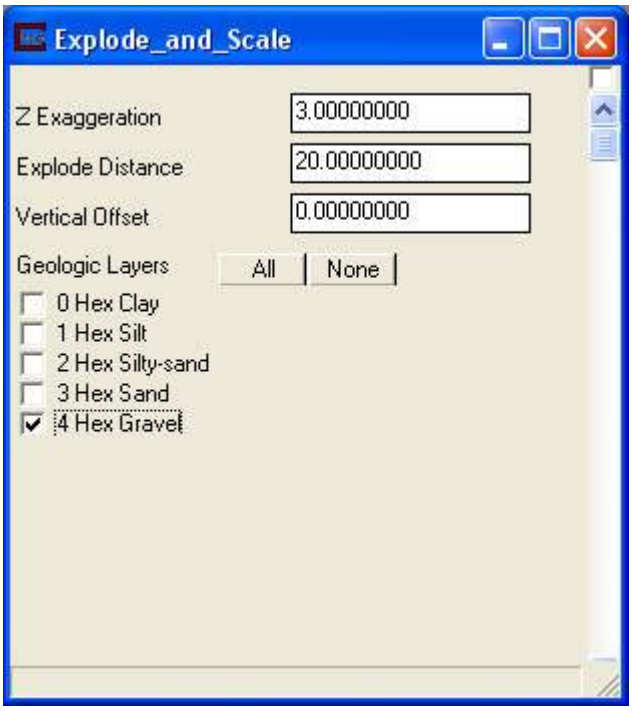
Your Viewer will show the probability that was computed for each cell's material assignment. Note that we are seeing probabilities on the near surface that range from near 1.0 down to about 0.25. Even though the

probability may be low, the selected material had the **highest** probabilities of all the available materials.

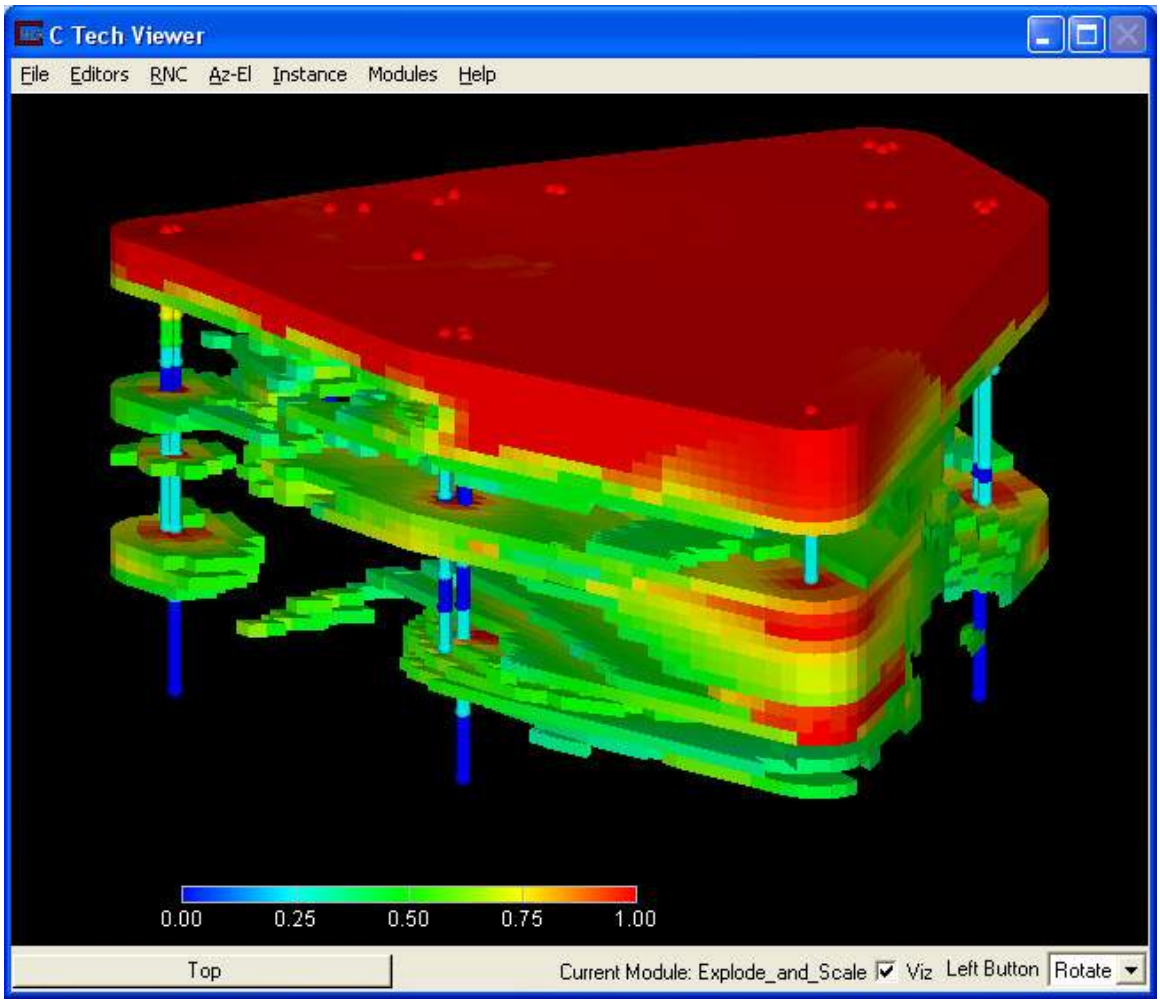


Note that the coloring is rather blocky. That is because we are dealing with cell data versus nodal data. Cells are now colored a single color based on the value assigned to the entire cell. With nodal data, the data was assigned to the corners of cells and the colors were interpolated across each face of the cell. Each hexahedron cell could have markedly different face colors with nodal data, but here the colors are constant for each cell (actually the color is constant, but the shading can still affect the brightness of a cell's face depending on the angle to the light source).

Go back to Explode_and_Scale and once again let's turn on only Gravel.



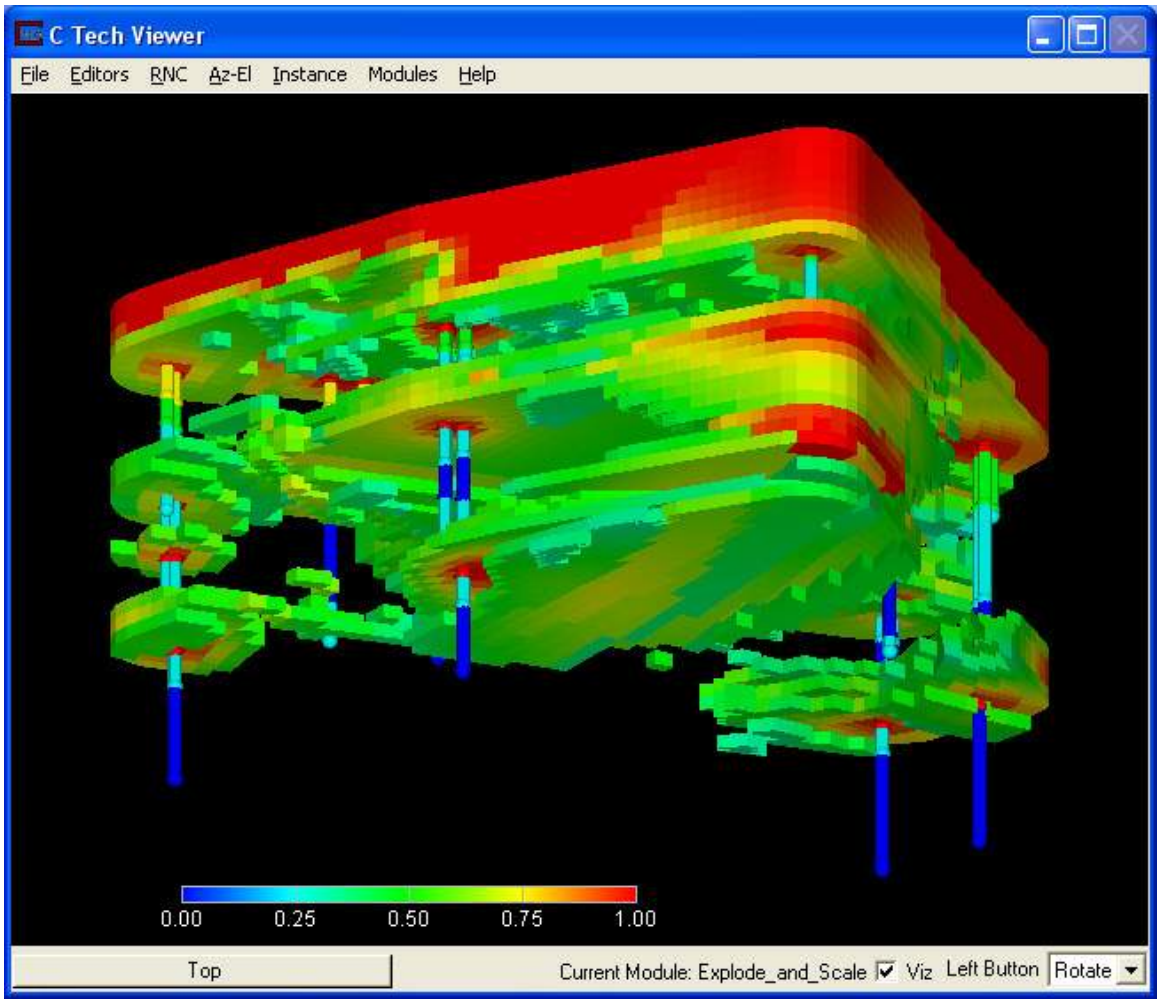
The probability of Gravel within the Gravel material is now displayed in the Viewer.



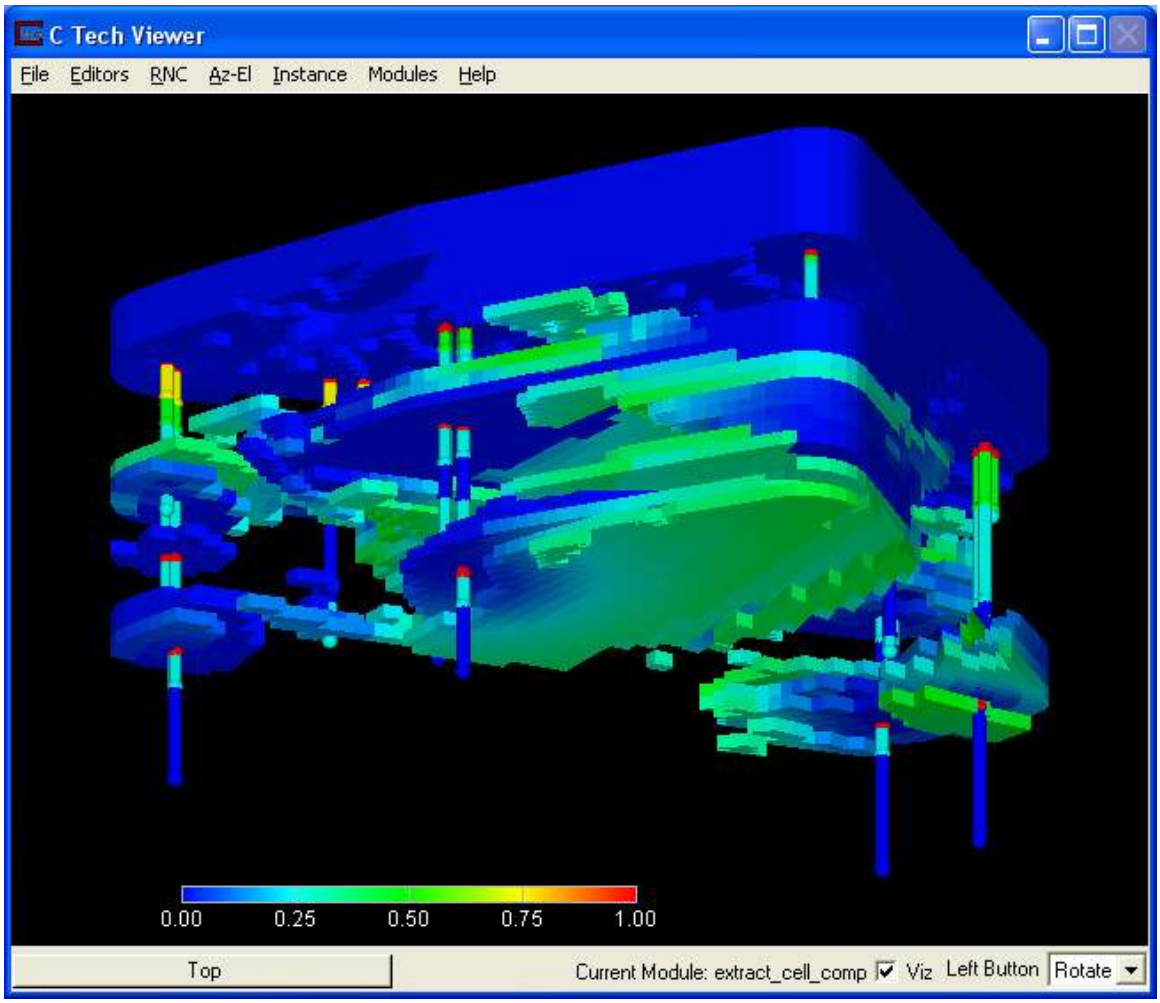
Note: If you change select_data to Gravel Prob., the picture doesn't change! This is important and fundamental to what we're doing. The **Indicated Material** is that material which had the highest computed probability for each individual cell. The **Probability** cell data component is the highest computed probability for every cell regardless of material, however the material that gave the highest probability is the one we select!

Therefore, the Probability within the Gravel will always be the same as the Gravel Prob. within the gravel.

However, in the gravel zone, the probabilities that any of these cells could be any other material will be lower. Let's look at the Gravel Prob(ability) from a different view. With a -20 degree elevation it looks like:

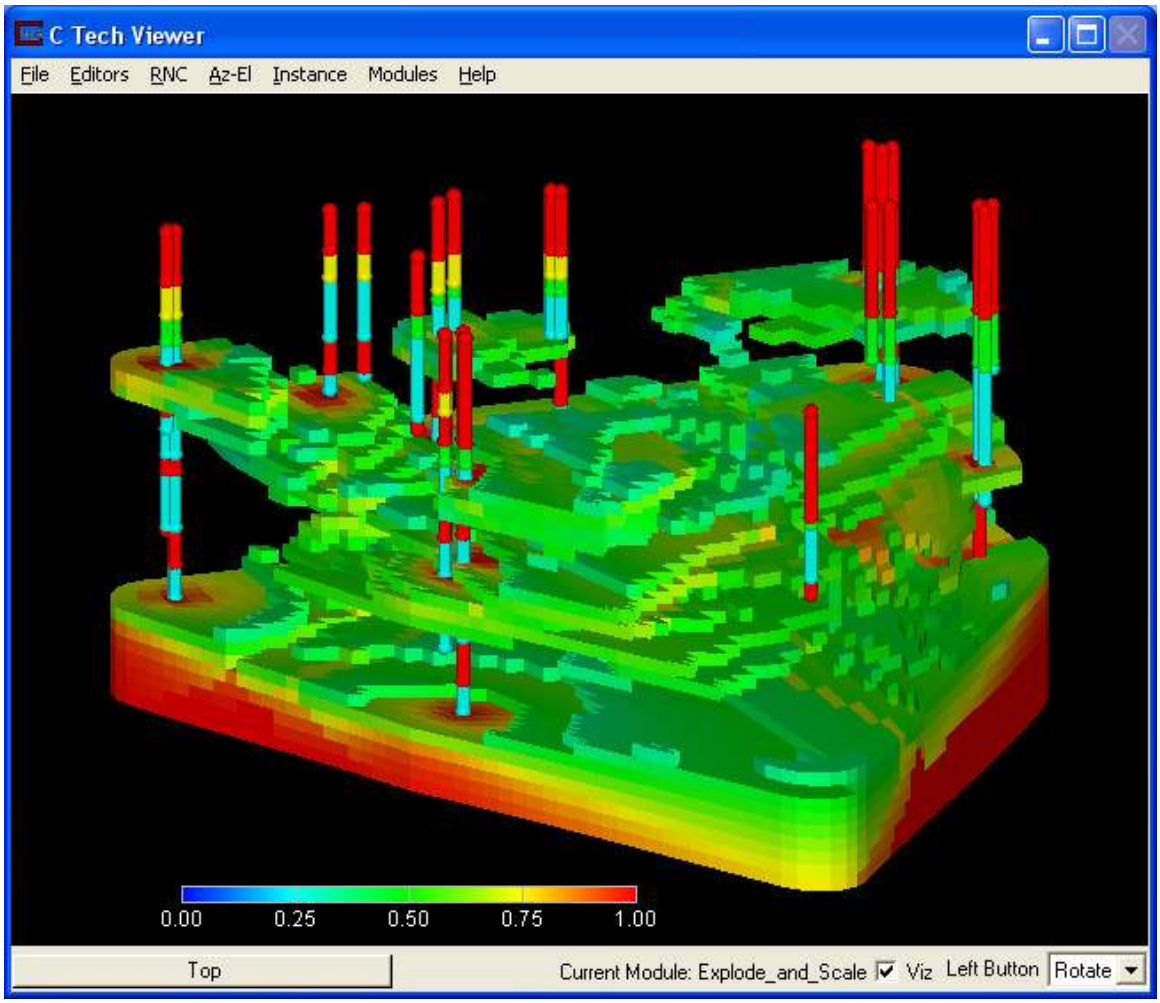


Now if we change to color by Clay Prob(ability) our picture changes to:

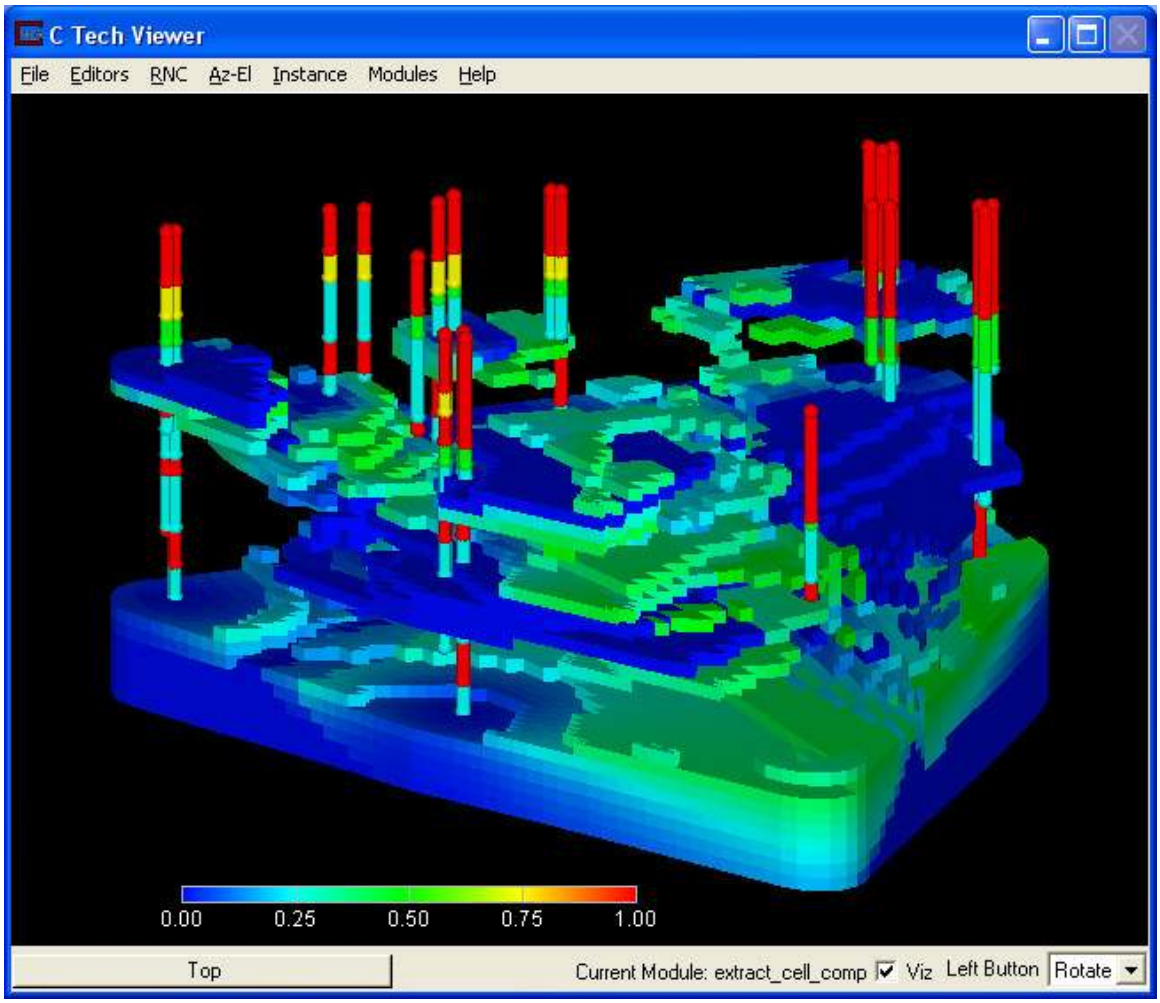


Note that over most of the gravel, the clay probability is much lower, but for the near region at the bottom (where we bound clay!), the probability is nearly identical! At the boundaries between materials, the probabilities will be nearly equal and this tells us that the material assignment here is rather ambiguous.

We'll look at another example. If we choose Explode_and_Scale and select Clay and in select_data's window choose Clay Prob(ability):



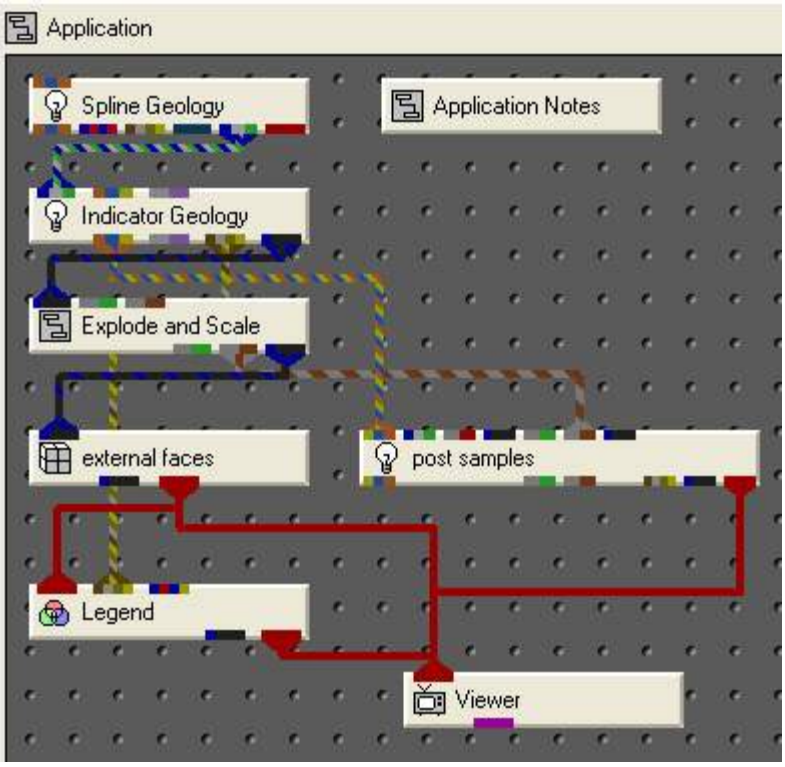
Now if you change select_data to Gravel Prob most of the picture changes, but the near top surface is nearly the same (since it borders gravel).



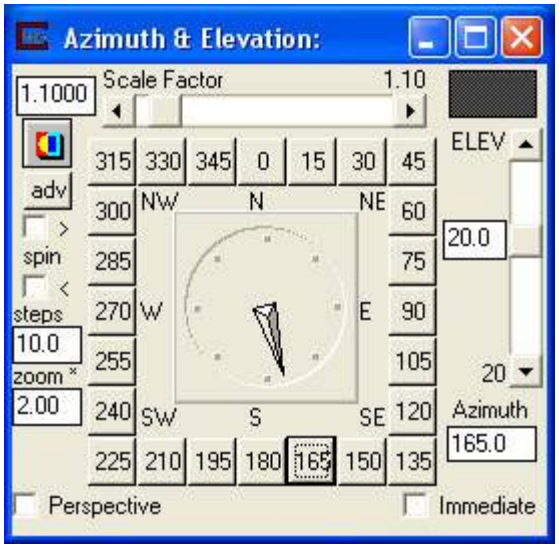
Anisotropy Effects on Indicator Kriging

Since the effect of anisotropy on Indicator Kriging is virtually identical for the Quick Method and probabilistic kriging, let's return to the basic network used two topics ago.

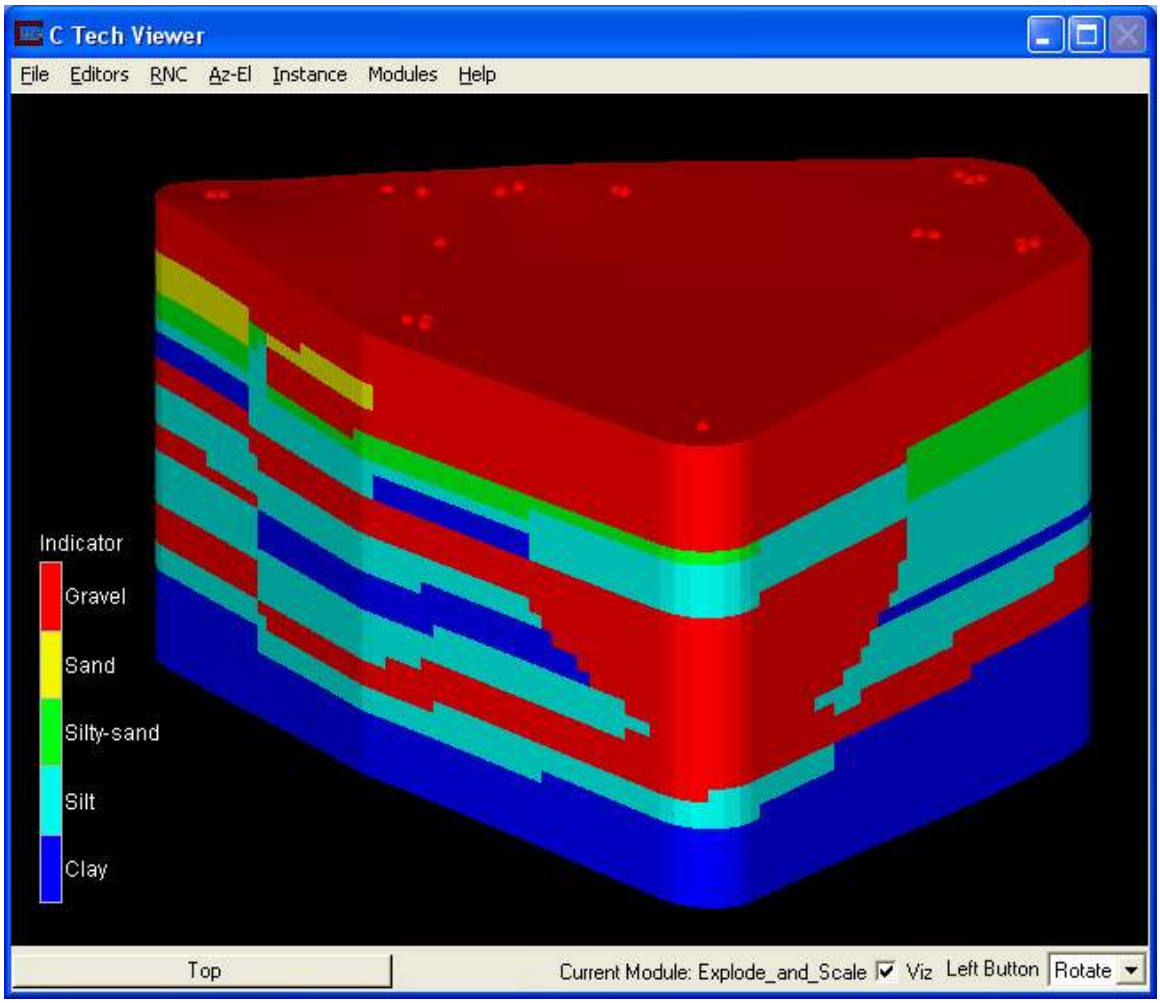
It can be loaded from ctech\applications\pro\5_material_complex_indicator_geology_quick.v. The network is shown in the figure below:



When this application runs, turn on all of the materials in Explode_and_Scale, and set the view to match:

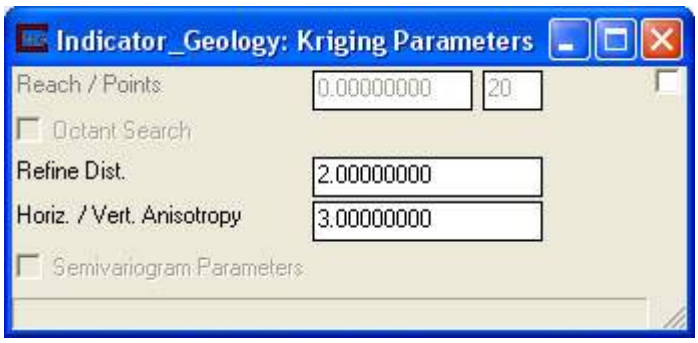


The resulting model shows:

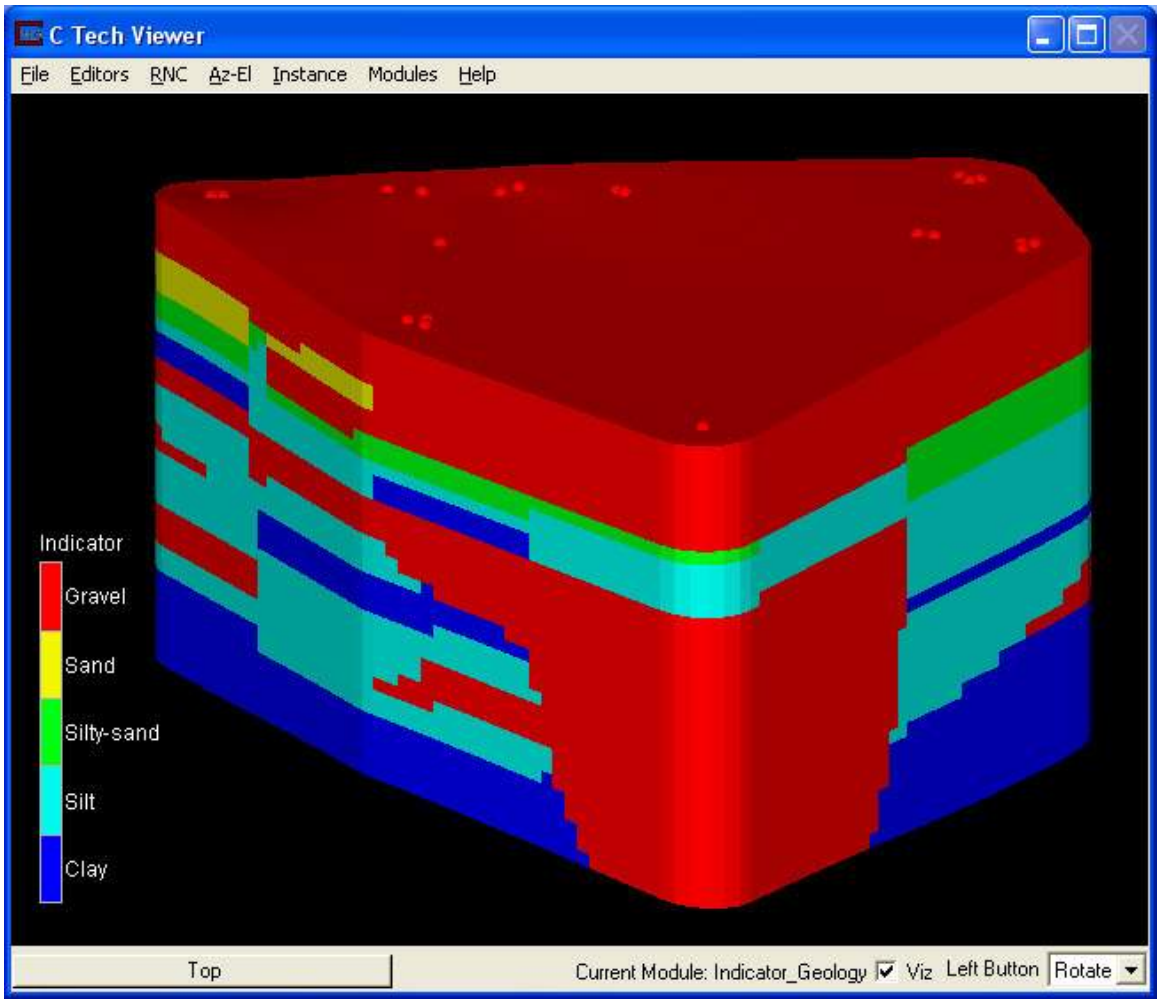


We'll make a single change and see how dramatic its affects can be.

Select Indicator_Geology from the module list and open the Kriging parameters window. Change anisotropy from 10.0 to 3.0 and Click on *Accept All Current Values* to re-run.

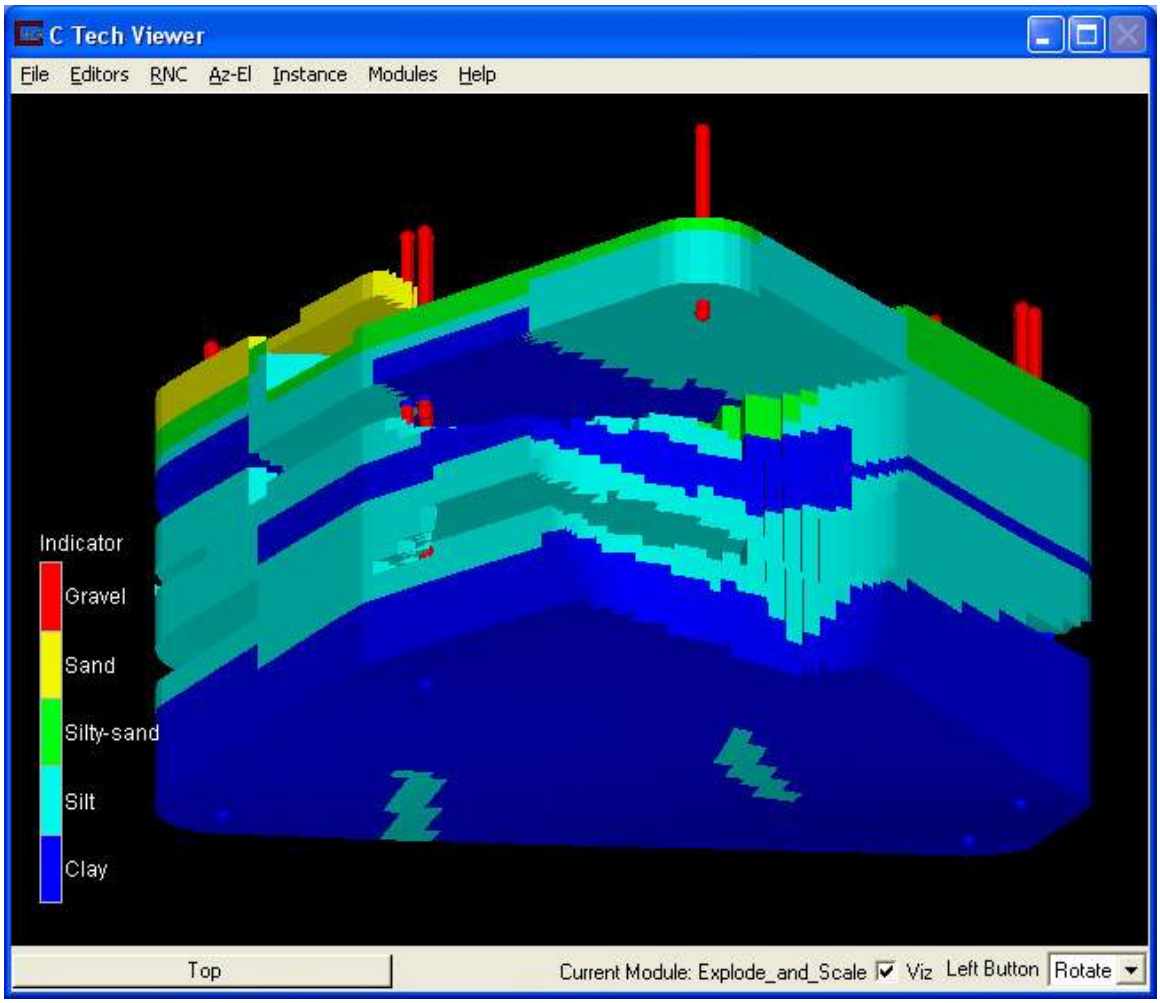


When complete your view should now show



Note that this model changes most dramatically in areas with little or no data.

The near corner (for this view) is where we have a single *short* boring. Let's turn off Gravel (in Explode_and_Scale) so we can better see what is happening. With a -20 degree elevation it looks like:

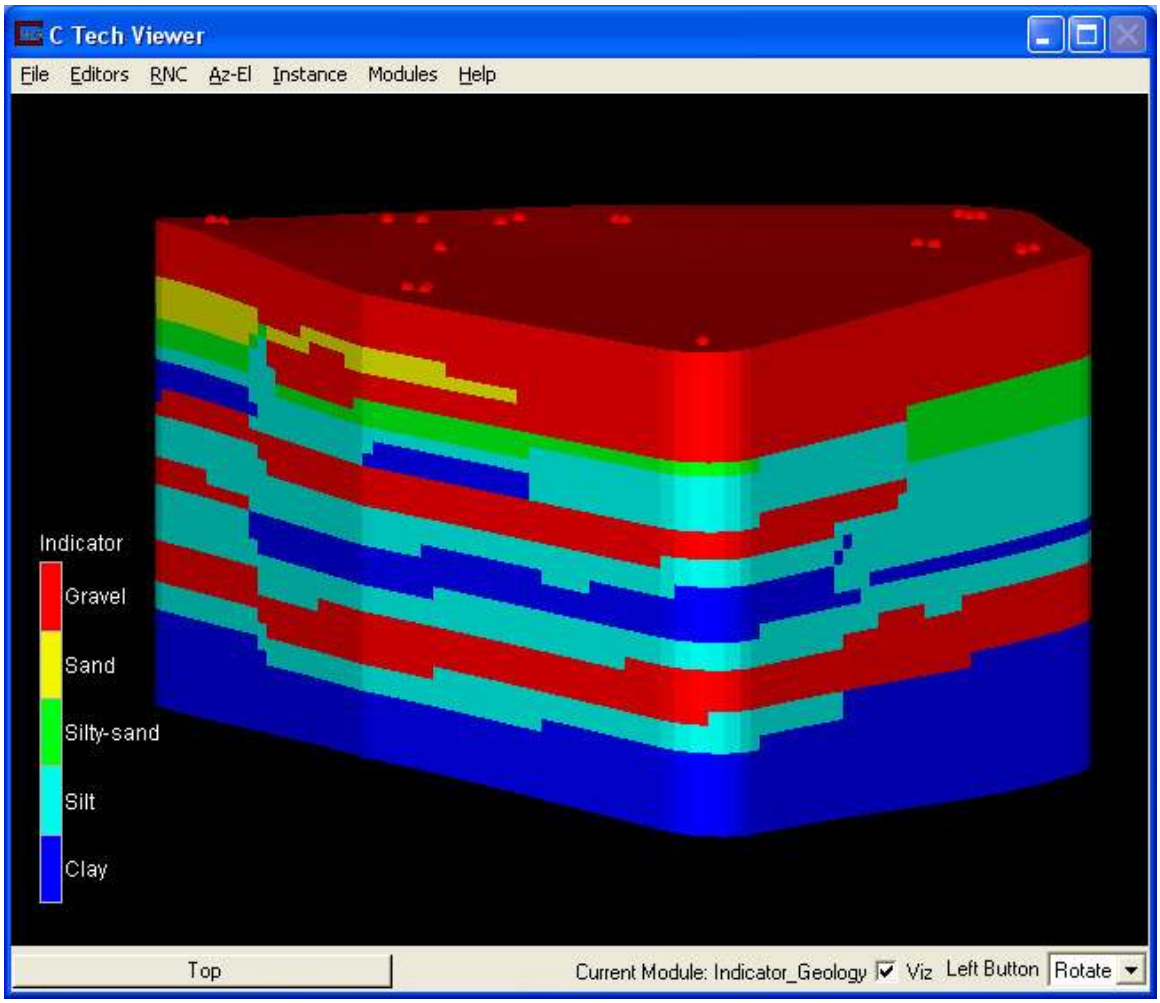


What are we seeing?

When the anisotropy was set to 10, we are saying that distances in the x-y plane are 10 times further than distances in the z direction.

With the anisotropy set to 3, x-y distances are 3 times further than distances in the z direction. This causes the effect of this short boring that ends in gravel to propagate further into our model. In fact since our z-exaggeration (in Explode_and_Scale) is set to 3, distances ***appear*** to be computed isotropically. Effectively, the short boring creates a nearly spherical region of gravel for this vertical exaggeration.

Similarly if we change anisotropy to 100 we should expect the layers to propagate horizontally more like flat sheets. With a 10 degree elevation and the Gravel turned back on, this would change our view to:



Advanced Geologic Modeling Conclusion

In this workbook, we've covered a number of advanced geologic module topics. Let's quickly review the important points:

1. When working with Geologic Hierarchical models it is important to:
   a. Build accurate surfaces
   b. Understand normal and reversed hierarchy
   c. Decide between Kriging and (thin plate) splining
   d. Make sure you honor the boring data
2. With Geologic Indictor Kriging, the quick method gives good quick results.
3. The Probabilistic kriging approach is more rigorous and should be investigated
4. Horizontal-Vertical Anisotropy can dramatically alter GIK models.
5. The best approach (when possible) is to try all methods and pick the most appropriate one for your data. Rigorous investigation of the

geology of a site should generally include GIK in addition to hierarchical modeling (if appropriate).

6. If the resulting model shows clear evidence of layering it is a good indication that you should probably be using the hierarchical modeling approach instead!

In conclusion, for the 5 material complex geology data used in this workbook, the decision of which approach is best is perhaps a bit unclear. The greatest differences in the models are in areas where there is a lack of data. Perhaps the best conclusion we can draw about this data set is that it may be inadequate for some applications.

Ultimately a geologist familiar with the conditions at your site should help in the determination of modeling approach and settings. For other sites, the choice between the two approaches is often quite obvious, especially if hierarchical interpretation is too difficult or impossible.

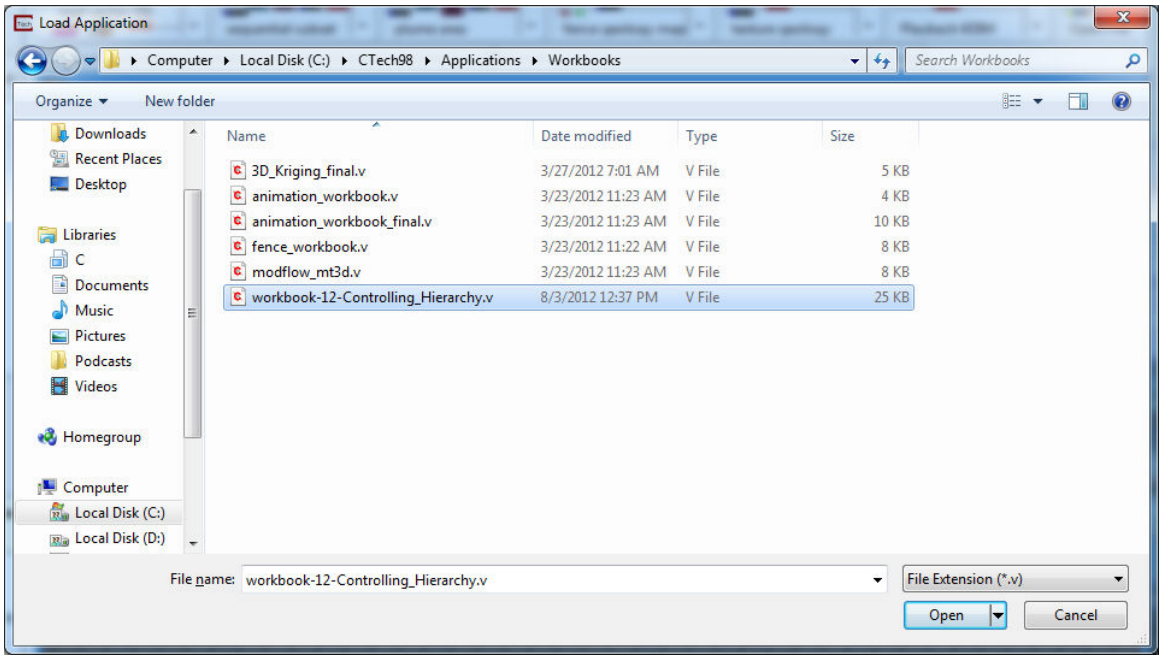
Controlling Geologic Hierarchy Index

WORKBOOKS

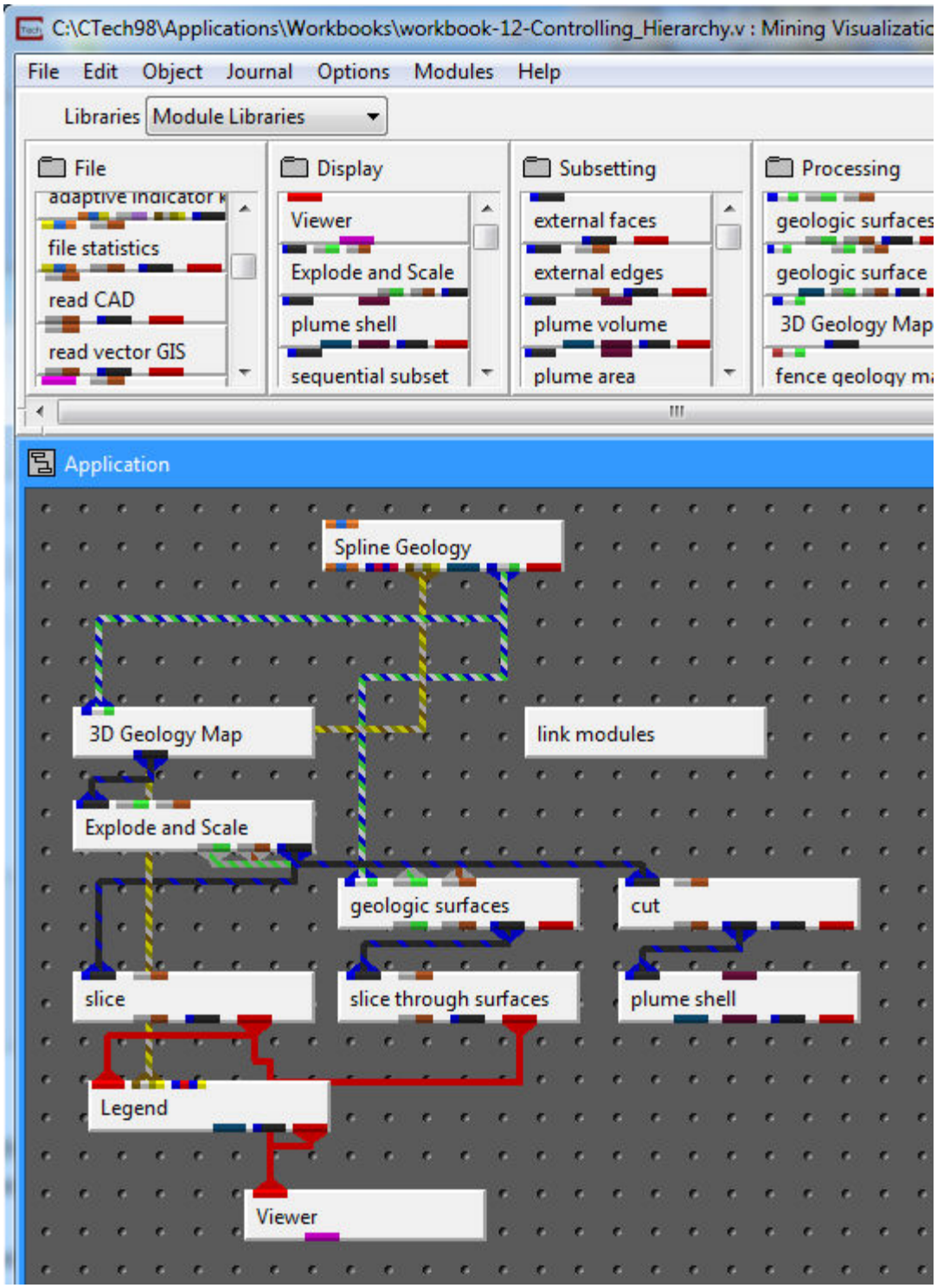
Understanding Our Starting Point

We'll begin this workbook by loading an application which has some interesting complications.

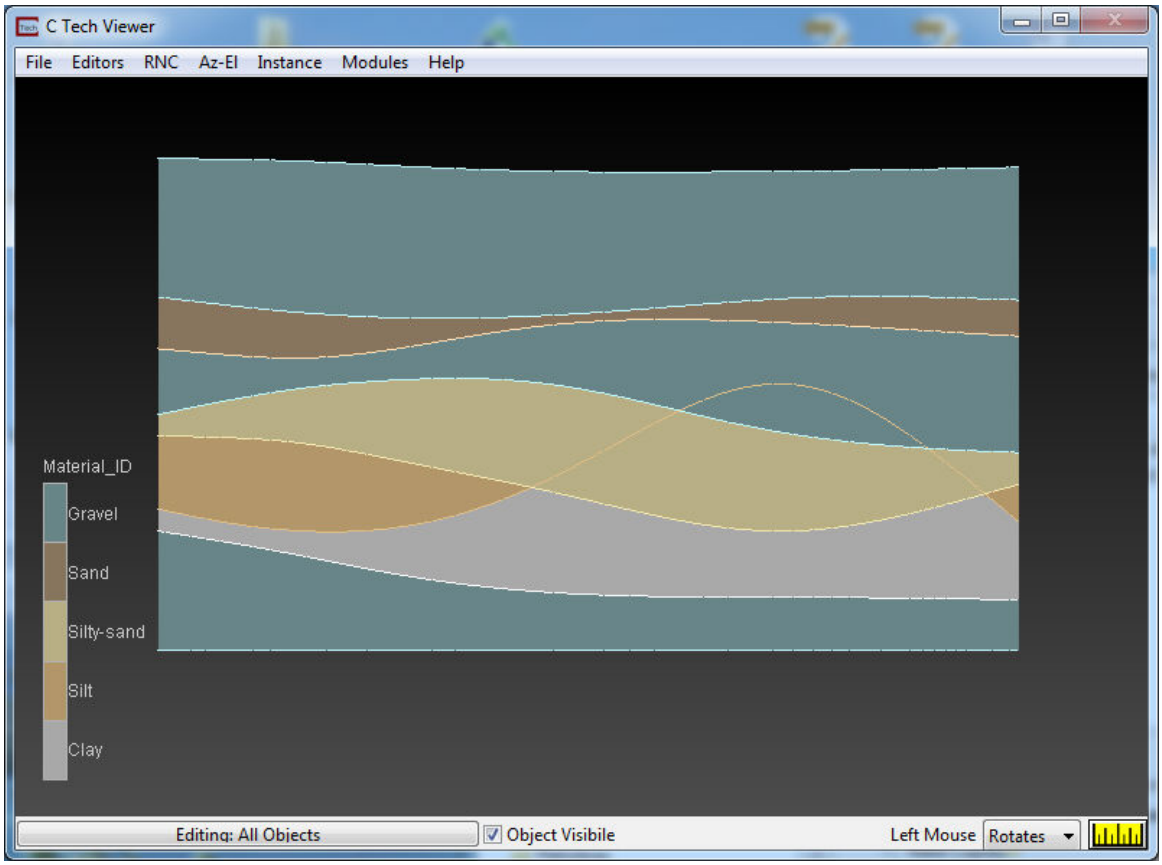
In the folder ctech\applications\workbooks load workbook-12-controlling_hierarchy.v



Within a few seconds the application should load:



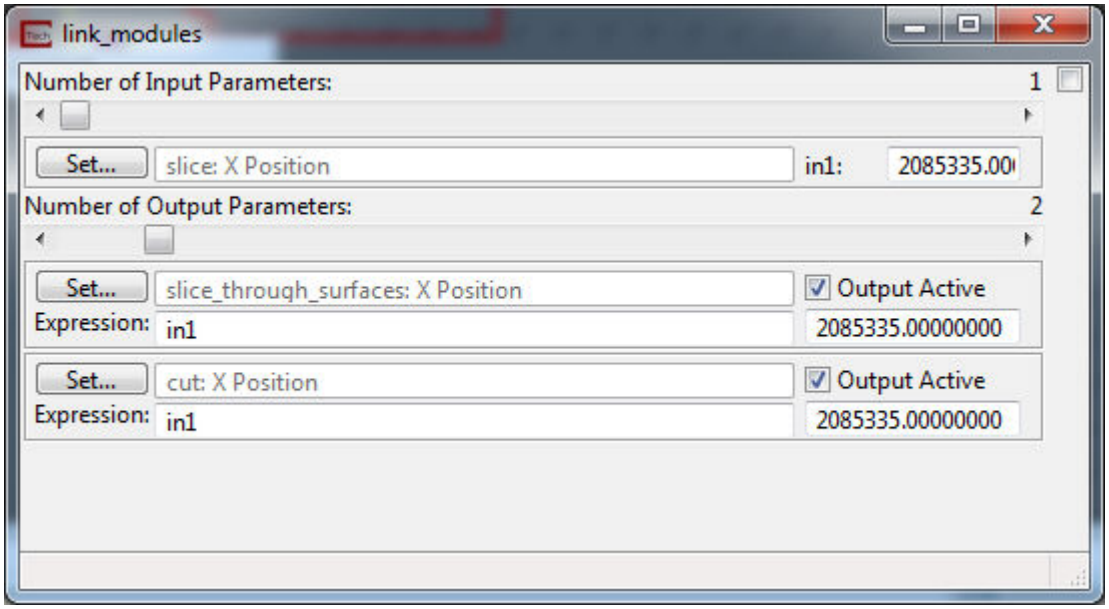
When it runs, the Viewer should show:



Let's discuss each of the more unusual modules in this application and its function:

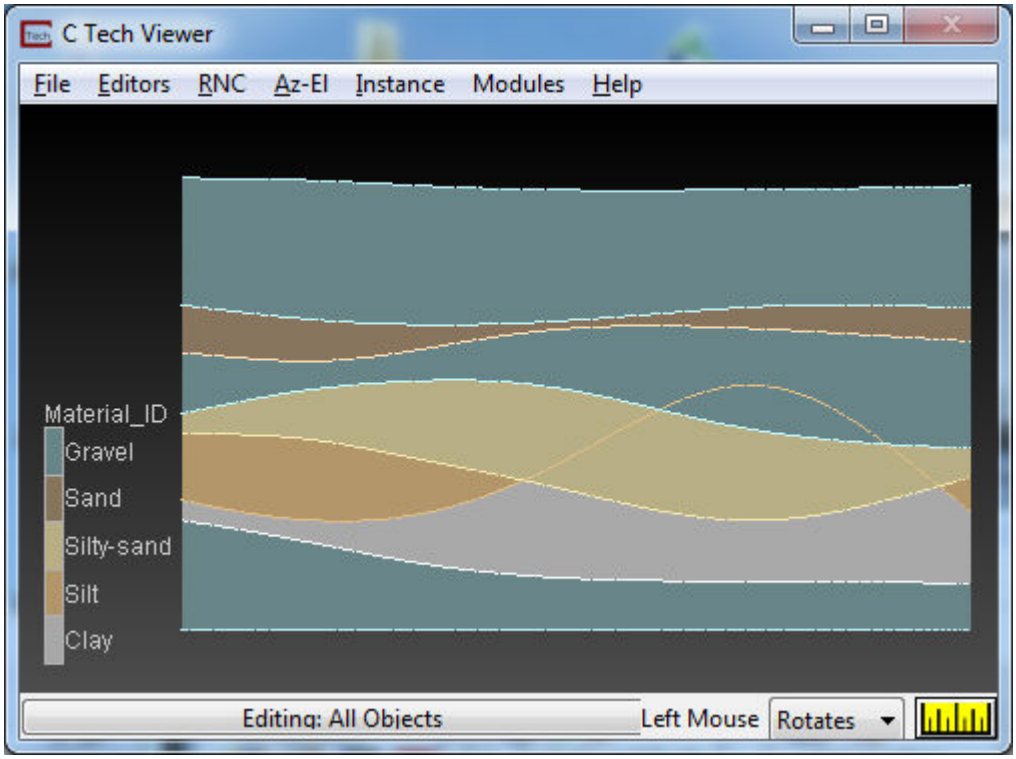
1. Spline_Geology is similar to Krig_3D_Geology, but uses the thin plate splining algorithm instead of kriging. We've selected it for this application because for geologic datasets with few defining points, it gives smooth surfaces that frequently intersect due to the extrapolation characteristics of this algorithm
2. 3D_Geology_Map, Explode_and_Scale, geologic_surfaces, Legend and Viewer are commonly used modules in many applications and examples
3. slice and cut are commonly used modules, but in this application, the position of the slice/cut is being shared by several modules because of link_modules
4. plume_shell is added after cut to remove the pinched out layers. When viewing a cross-section (e.g. slice) through our volumetric model, the very thin portions of layers are not visible. But if we explode the layers, these thin membranes must be removed or the pinched regions are not obvious. Our software could have been designed to remove them automatically, but that would provide less control over what thickness threshold to apply.
5. slice_through_surfaces is just a slice module that we've renamed to make its function in this application more obvious. It is slicing through the output of geologic_surfaces at the same easting position as the slice module. Since it is slicing through a set of surfaces, its output is a set of LINES. These lines are colored by the geologic material. The position of the slice (lines) is being shared by several modules because of link_modules

6. link_modules plays an important role in simplifying this application. You might ask, how does adding an extra module make things simpler. The answer is that the application is simpler to USE because link_modules synchronizes the position of cutting or slicing in three different modules (our two slice and one cut modules)



Normal and Reversed Hierarchy

Our application as loaded is showing us a slice at a specific X coordinate where the geologic surfaces or *horizons* can be quite different depending on how we choose to impose hierarchy rules.

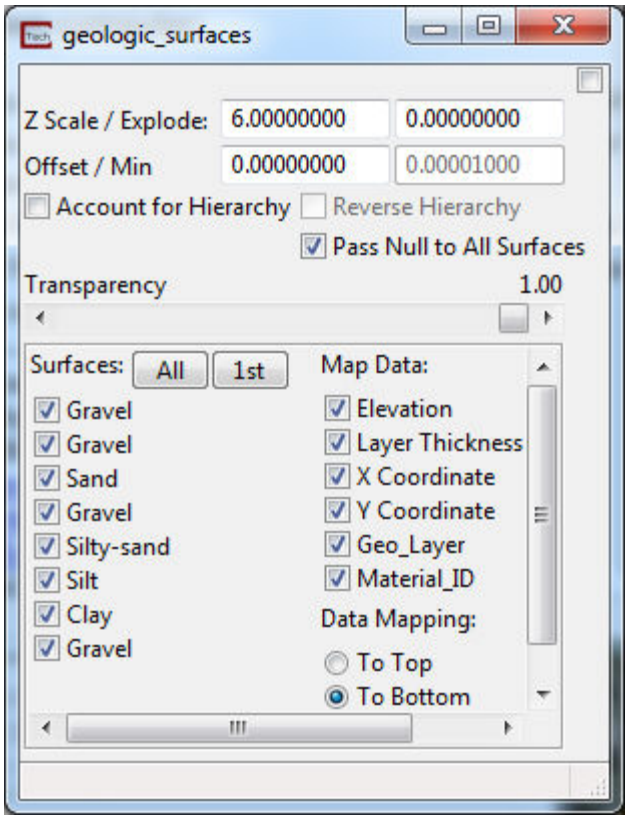


Along the left side of the slice, the ordering of the horizons is unambiguous. These horizons, highlighted by the bright lines are:

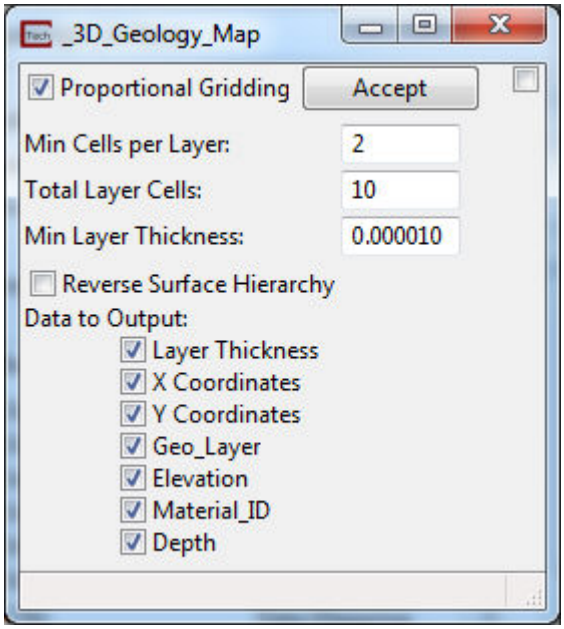
1. Ground Surface
2. Bottom of First Gravel
3. Bottom of Sand
4. Bottom of Second Gravel
5. Bottom of Silty Sand
6. Bottom of Silt
7. Bottom of Clay
8. Bottom of Third Gravel

As we move to the right, beyond the center of the slice, the ordering of these horizons becomes less obvious since the 6th horizon (bottom of slit) intersects and is above the two higher horizons (5 and ultimately 4).

The geologic_surfaces module in our application has its "Account for Hierarchy" toggle off

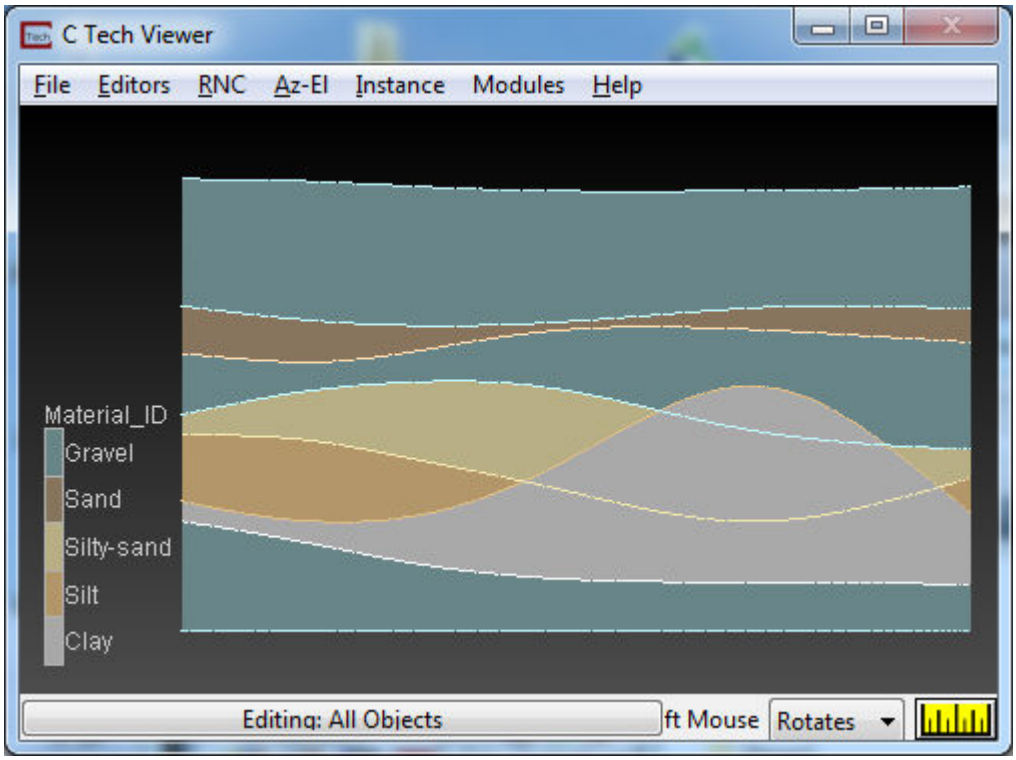


which is why the *normal* hierarchy as created by 3D_Geology_Map



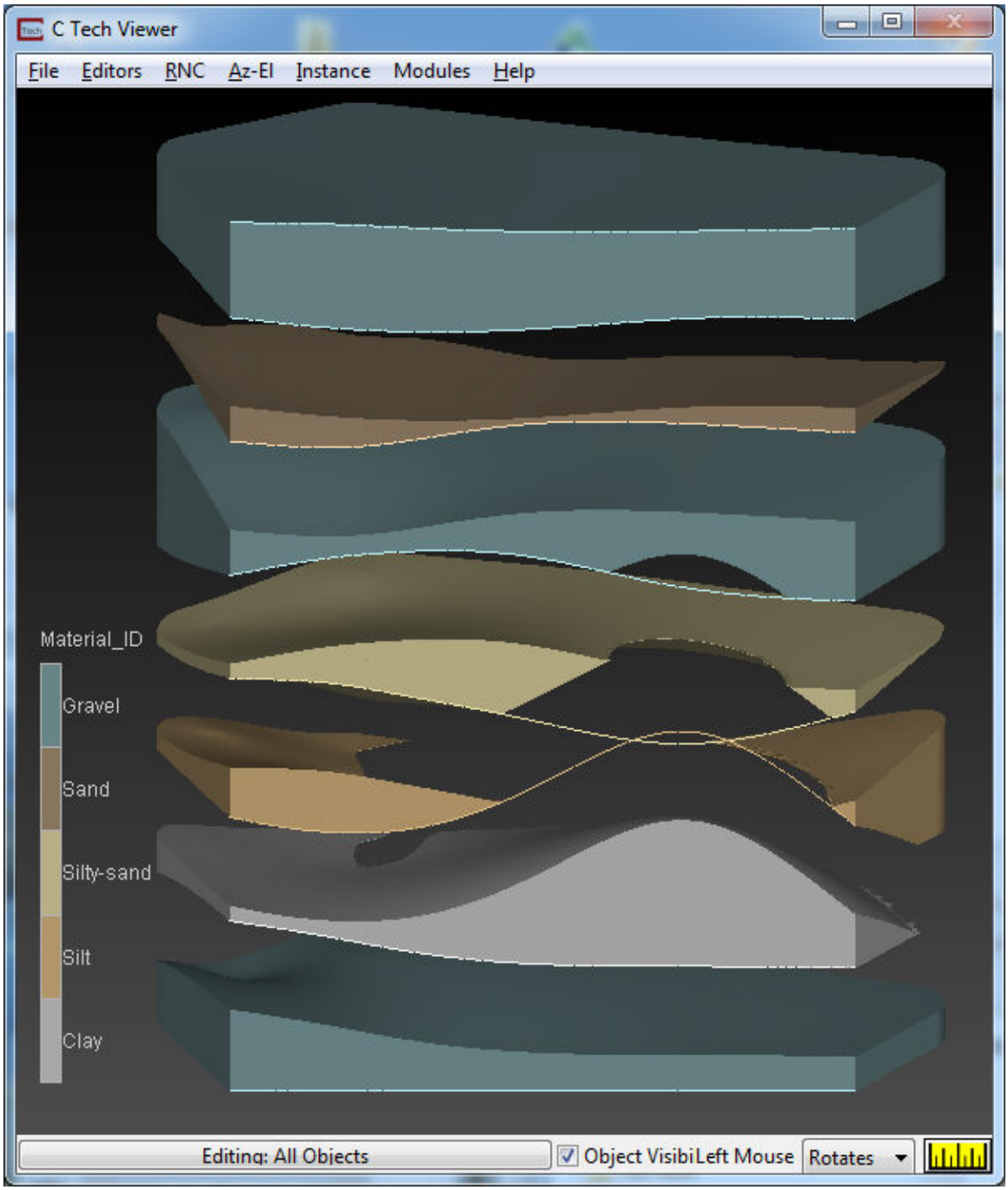
and shown with slice differs from the highlighting horizon lines.

If we turn on the "Reverse Surface Hierarchy" toggle, the layering in the slice is dramatically different:

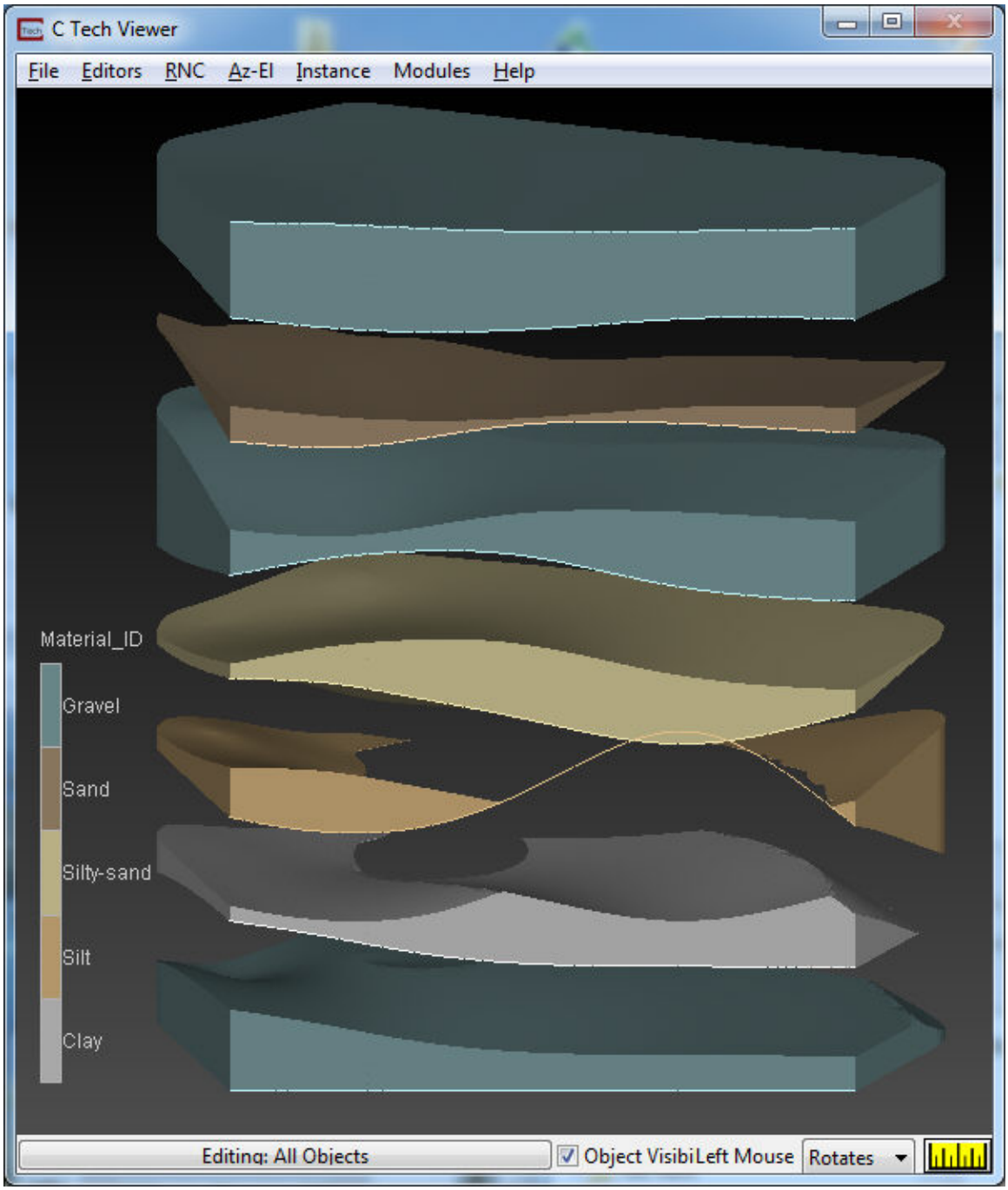


This reversed hierarchy favors the lower surfaces and allows the bottom of Silt to intrude into the two layers above (Silty Sand and Second Gravel), causing those two layers to thin and/or pinch out.

If we connect plume_shell to the Viewer and set the Explode Distance (in Explode_and_Scale) to 15, we can see the 3D representation of this model, in the REVERSED hierarchy:



and the NORMAL hierarchy


C Tech Viewer
File Editors RNC Az-El Instance Modules Help
Material_ID
Gravel
Sand
Silty-sand
Silt
Clay
Editing: All Objects
Object VisibiLeft Mouse
Rotates

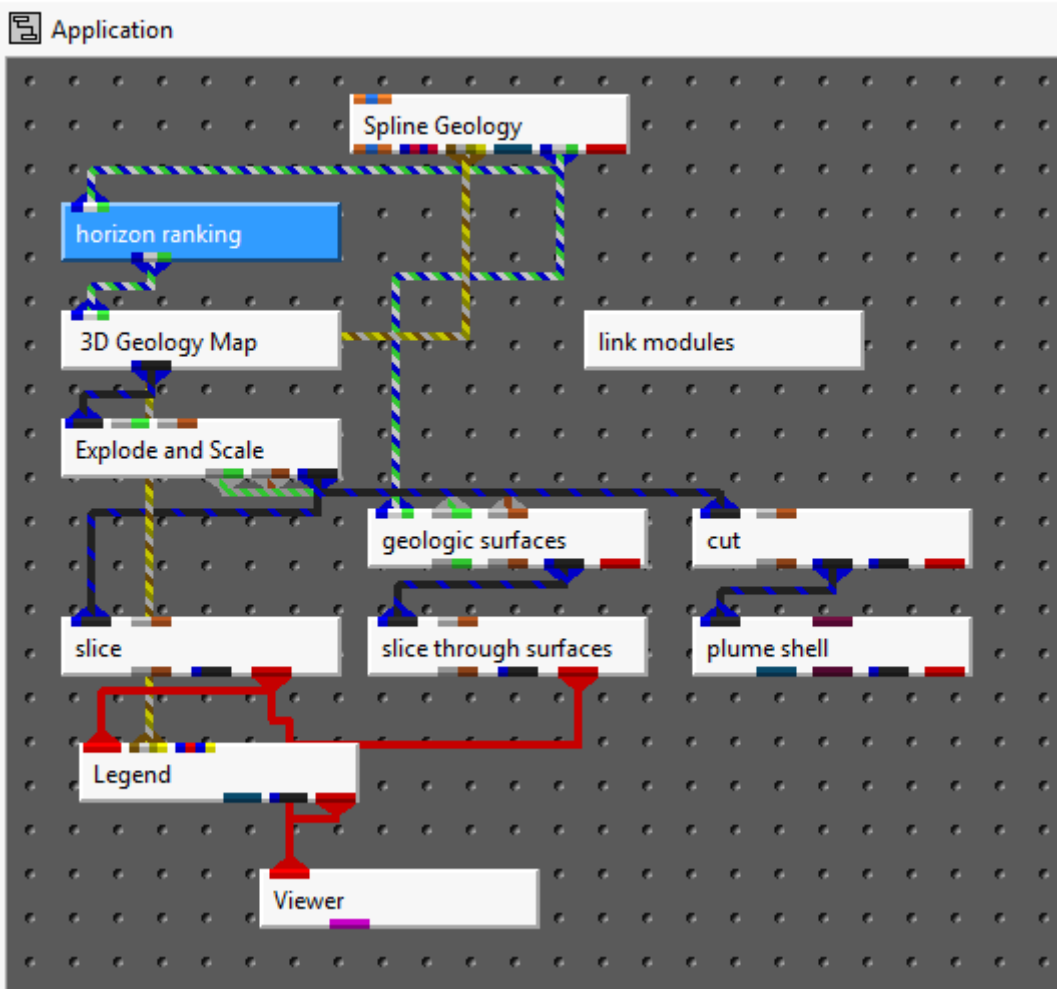
Controlling Geologic Hierarchy

When we consider the dramatic difference between the Normal and Reversed Hierarchy results from the preceding topic, it may seem that we don't need any more **control**, but that would be too limiting.

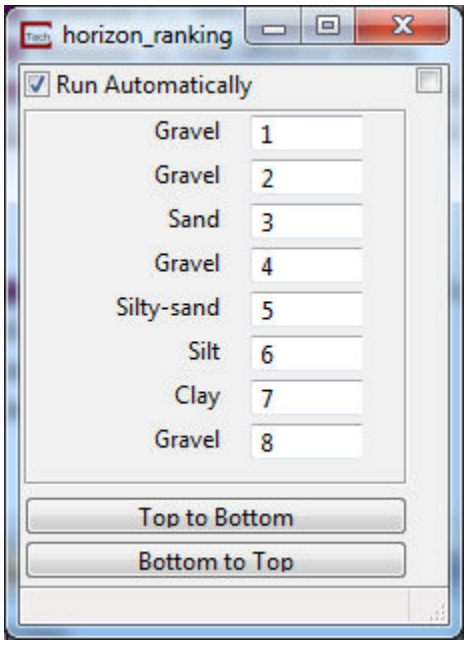
In our original starting application view, the bottom of Silt (top of Clay) intersect two horizons which are higher in the hierarchy. Normal hierarchy gives both of these surfaces higher priority and Reversed hierarchy makes both lower priority.

What if we want more control of the individual surface priorities or rankings?

The horizon_ranking module provides this extra control. It allows you to set the individual ranking or priority for all horizons. For the model in this workbook, insert it as shown below (also disconnect plume_shell and return Explode_Distance to zero.)



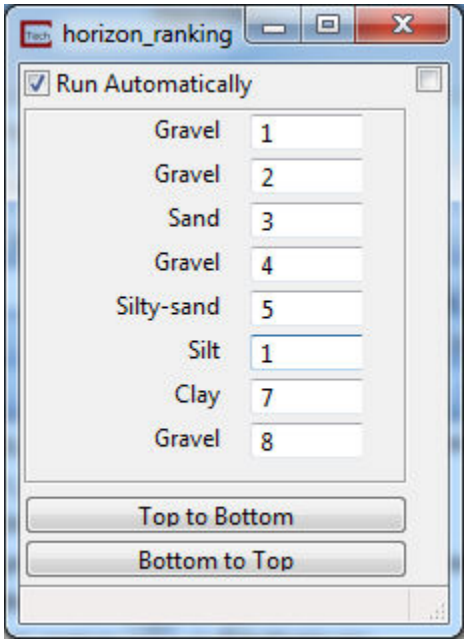
The default setting of horizon_ranking are equivalent to Normal hierarchy:



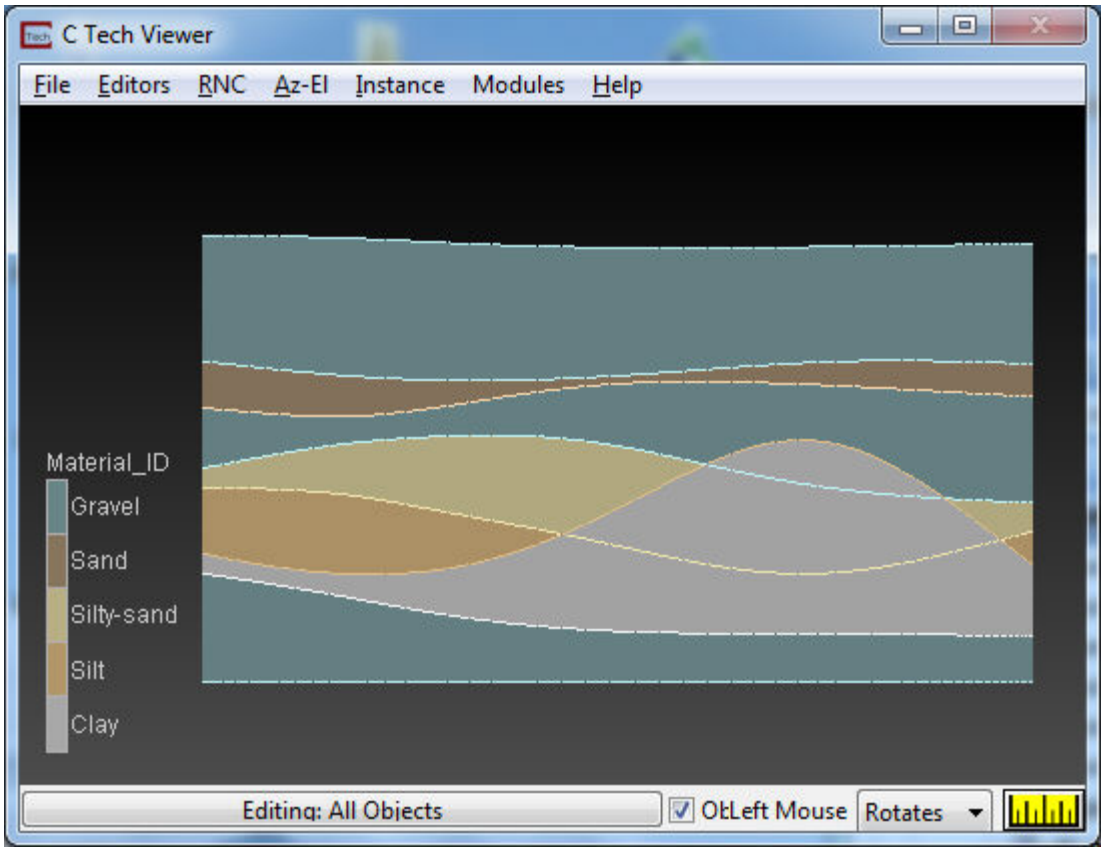
The Top to Bottom button returns the rankings to their default values as shown above (1 on top)

The Bottom to Top button reverses the ranking and is equivalent to reversed hierarchy

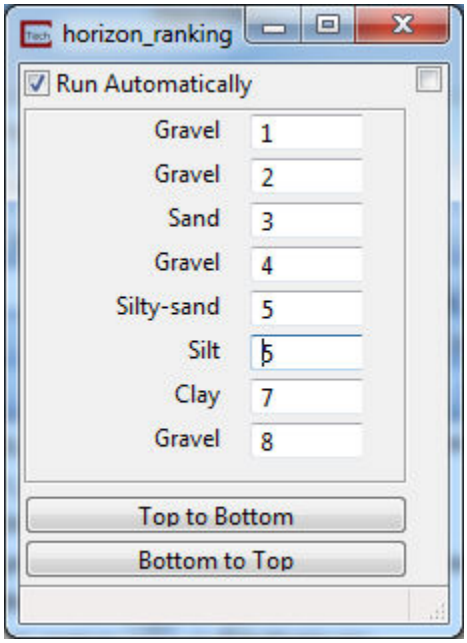
Setting the (bottom of) Silt horizon to a higher priority will allow it to intrude into higher layers. Setting the value to 1:

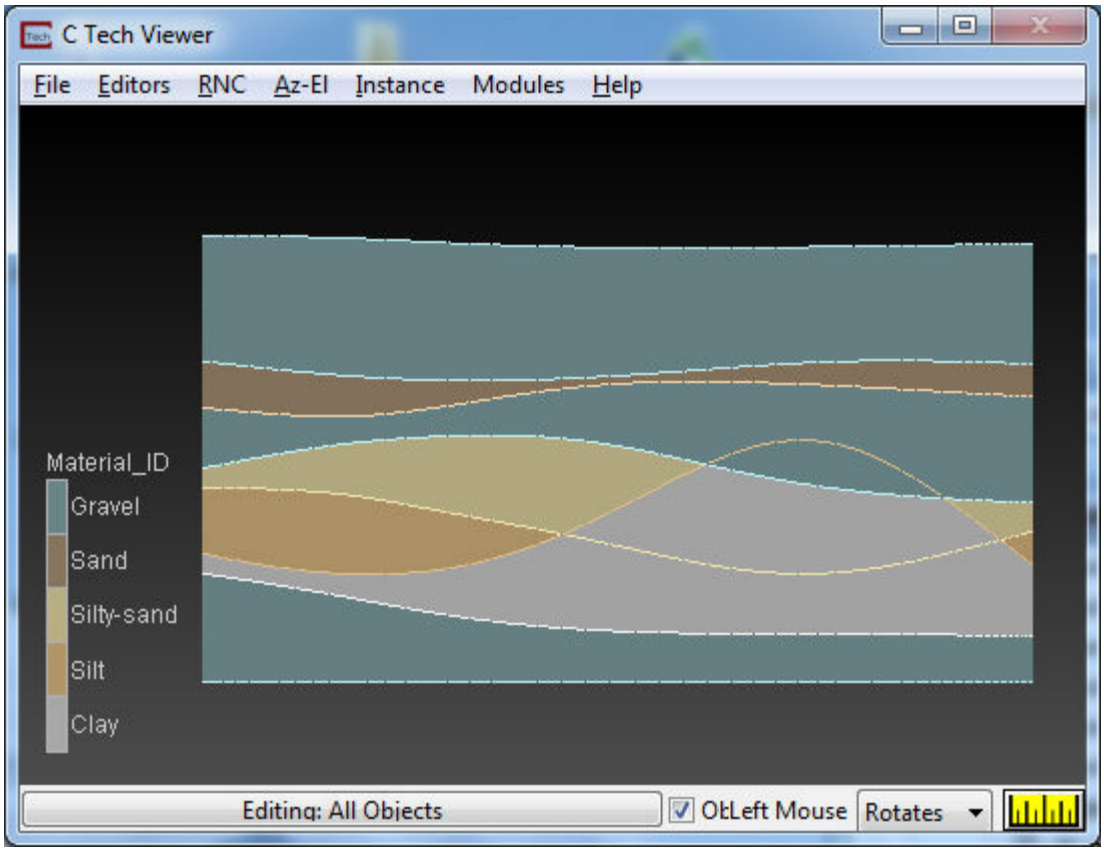


is equivalent to reversed hierarchy in this example:



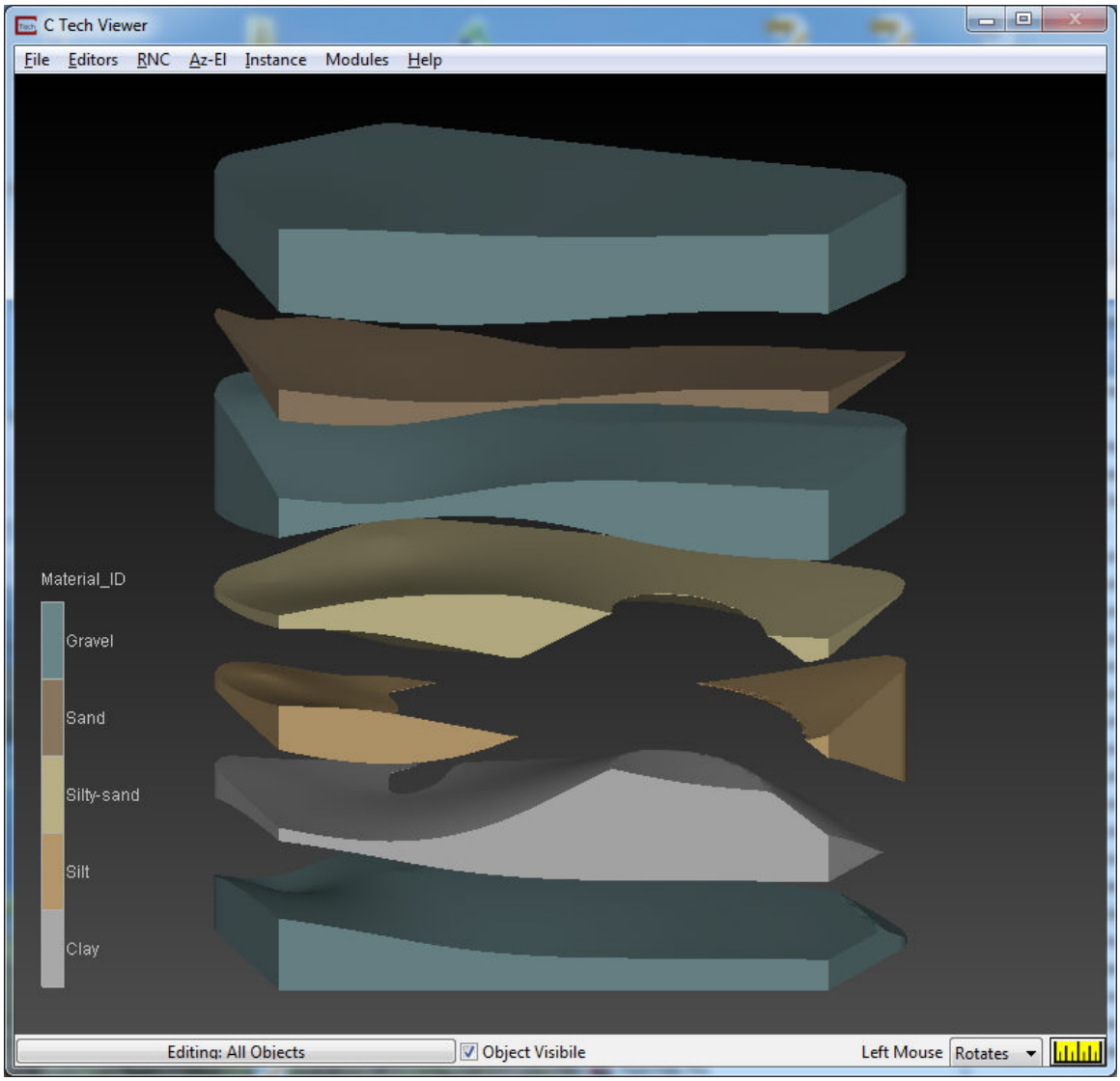
However, setting Silt to 5 provides a unique level of control:



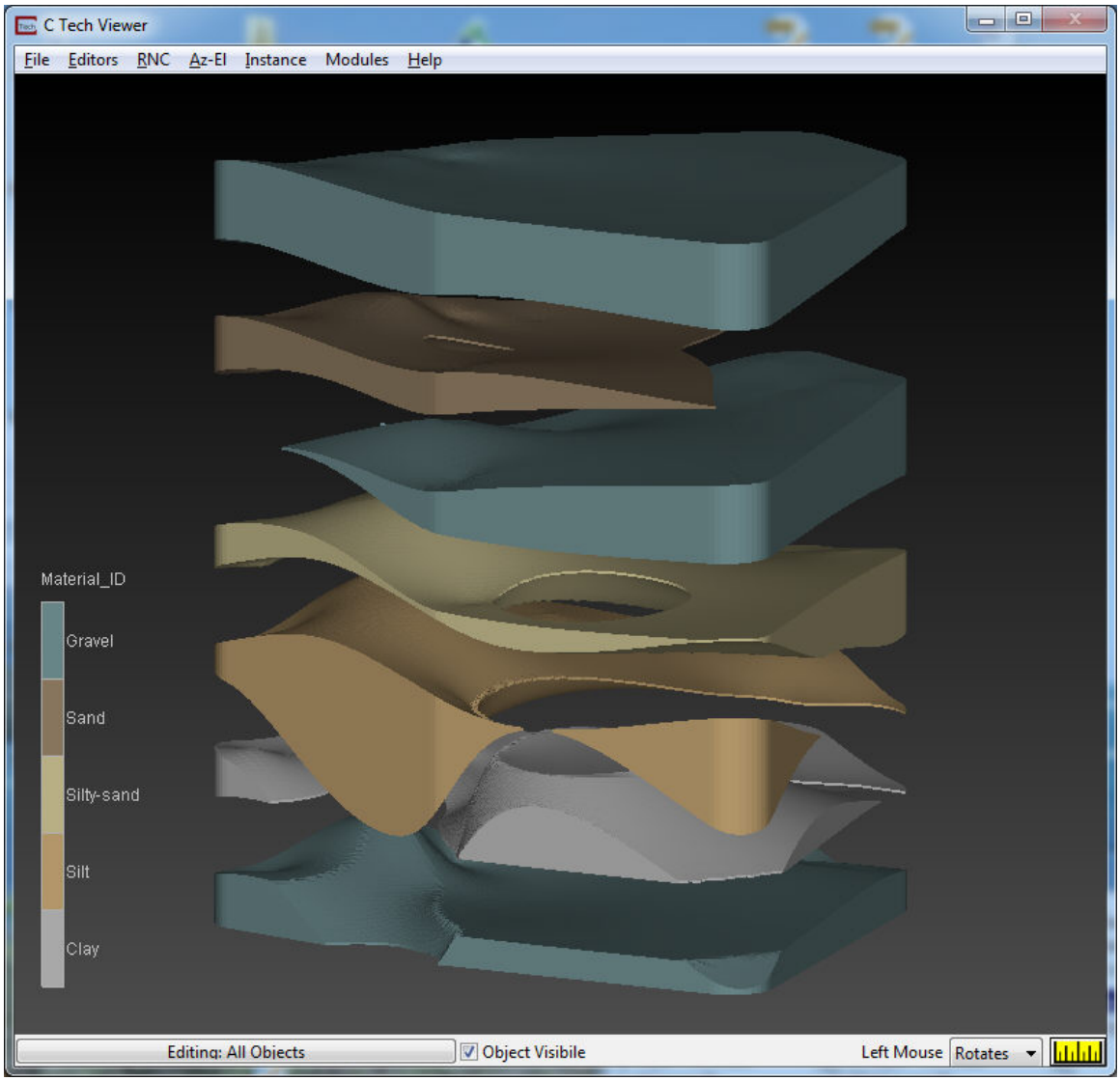


Note that with these settings, the clay layer penetrates the Silty Sand, but does not intrude into the Second Gravel.

This detailed control allows us to model a geologic system where clay and silty-sand were eroded before the second gravel was deposited.



Moving the cut surface so the entire model is visible and viewing from the South gives:


C Tech Viewer
File Editors RNC Az-El Instance Modules Help
Material_ID
Gravel
Sand
Silty-sand
Silt
Clay
Editing: All Objects
Object Visibile
Left Mouse
Rotates

Visualization Fundamentals

Visualizing Environmental Data

Summary

Environmental issues can quickly become extremely complex. When dealing with site assessments, environmental remediation design or monitoring, public hearings, or environmental litigation, the quantity of data involved can quickly become overwhelming. Maintaining and organizing that data is insufficient. Visualization is the only means for condensing and communicating vast quantities of spatially referenced data. Whether the data consumer is an environmental engineer, geologist, or the public, visualization provides an invaluable tool to communicate complex data in a form that makes it intelligible to all parties.

This paper addresses the key issues related to the visualization of environmental data. Compiling the site data and ensuring that it is consistent, adequate and complete enough to be used is the first step. Beyond the data, the paper addresses the myriad of methods that can be applied to visualize, analyze and model the data.

Conscientious data visualization should begin with direct data visualization that displays the data with a minimal amount of interpretation or assumptions. The next stage of investigation and display involves gridding the data and interpolating and extrapolating the data to the nodes and/or cells of the grid. This process can be complicated by the incorporation of geologic information, site features (such as roads, buildings, and bodies of water), and aerial photography. These issues are addressed in detail in the sections below. Once the data is gridded and mapped to the grid, numerous examples are given of how to present this data visually. Presenting the data generally requires finding ways to delve into the data in three-dimensions. Usually this is performed using one or more data subsetting techniques.

Visualization is visual communication. It is the art and science of compressing and presenting information in a useful form that we humans can easily comprehend. Whether the data collected at an environmental site represents a portion of a page or bookshelves of reports, visualization can benefit everyone involved by transforming words and numbers into comprehensible images.

This paper is dedicated to the complex task of visualizing environmental data. There are many ways of visualizing data ranging from simple graphs, plots and charts to more representational two and three-dimensional visualizations and time varying animations. The focus of this paper will be representational graphics that portray the spatial characteristics of data. To this end, the data requirements associated with this paper will necessitate that all data have spatial information.

Exploratory Data Analysis

When approaching any new project that involves environmental data visualization, the first step is to assess the form and content of the data. Data is commonly delivered in many forms (which will be discussed in the next section). Assessing the content of the data can be done visually or as simply as evaluating basic statistics such as the number of samples, minimum, average and maximum data values, and the spatial extent.

Data Sources

The form of the data is usually governed by the application in which it was authored and/or stored. Typical applications include database software, Geographic Information Systems (GIS), spreadsheet programs and ASCII editors.

Databases & GIS

Environmental databases can be built in generic database software such as Microsoft Access or Oracle, or products like Earthsoft's EQuIS or Integrate's TerraBase, which are specifically developed to provide environmental data management for both chemistry and geologic data.

GIS programs like ESRI's ArcView provide database functionality that is inherently tied to spatial information. For that reason, GIS systems provide an ideal platform for storing and retrieving data to be used for environmental visualization. GIS files may also include maps that contain features such as roads, building outlines, rivers and other geographic features. Most GIS systems also provide some level of visualization capability though it is generally two-dimensional or limited three-dimensional.

Environmental databases and GIS systems generally contain far more information than is needed for the task of data visualization. The addition of sample/measurement date, laboratory, analytical methods, quality assurance information, well construction details, and name(s) of companies or individuals are just a few examples of information which may not be necessary to the task of visualization. However this additional data is often crucial to the documentation and defense of visualization tasks.

Spreadsheets

Spreadsheet programs like Microsoft's Excel are commonly used as the repository for environmental data. Their ease of use and ability to perform many database-like functions enhance their appeal. Spreadsheets can import and export most common data file formats including many database files.

ASCII Files

ASCII (also known as text) files provide the most portable form for environmental data. Spreadsheets and databases can import and export ASCII files and they can be edited with a myriad of programs ranging from simple text editors (like Microsoft Notepad) to spreadsheets and databases. It is this portability that makes ASCII files a common choice as an input file format for environmental visualization.

CAD Files

Spatial environmental data if often contained in CAD (computer aided drafting) files. Many companies that take stereo-photography to provide topographic data deliver the results in CAD files. CAD files are also used to display roads, buildings and other site features. This data is most useful when the CAD files are drawn in the same, consistent coordinate system used for all other data.

Images

Image files (also known as raster or bitmaps) are digital photographs. These photographs can range from snapshots of features on the site to ortho-

rectified (requires that the image axes are parallel to the north-south and east-west coordinate axes) aerial photography. Snapshots merely provide documentation of site features and usually do not include any quantitative spatial information. Ortho-rectified photos can be used as texture maps. This allows them to be projected onto ground surfaces, geologic layers and/or buildings and other features in the visualization. In order to utilize a photo as a texture map we must be able to geo-reference the image. Some image file formats include geo-referencing information. This usually consists of the spatial coordinates of one corner of the image and the real-world size (both width and height) of a pixel. These formats include GeoTIFF, BMPW and TIFW (BMP & TIFF World Files). These file formats simplify the process of registering aerial photography. Some GIS software such as ESRI's ARC/INFO is capable of orthorectifying images that were not properly formatted.

Data Content Requirements

As defined above, our discussion of environmental data will be limited to data that includes spatial information. When spatial data is collected with a GPS (Global Positioning Satellite) system, the spatial information is often represented in latitude and longitude (Lat-Lon). Generally, before this data is visualized or combined with other data, it is converted to a Cartesian coordinate system. The process of converting from Lat-Lon to other coordinate systems is called projection. Many different projections and coordinate systems can be used. The single most important thing is maintaining consistency. Projecting this data is especially necessary for three-dimensional visualization because we want to maintain consistent units for x, y, and z coordinates. Latitude and longitude angle units (degrees, minutes and seconds) do not represent equal lengths and there is no equivalent unit for depth. Projections convert the angles into consistent units of feet or meters.

analyte (e.g. chemistry)

analyte (e.g. chemistry) data files must contain the spatial information (x, y, and optional z coordinates) as well as the measured analytical data. The file should specify the name of the analyte and should include information about the detection limits of the measured parameter. The detection limit is necessary because samples where the analyte was not detected are often reported as zero or "nd". It is generally not adequate (especially when logarithmically processing this data) to merely use a value of 0.0.

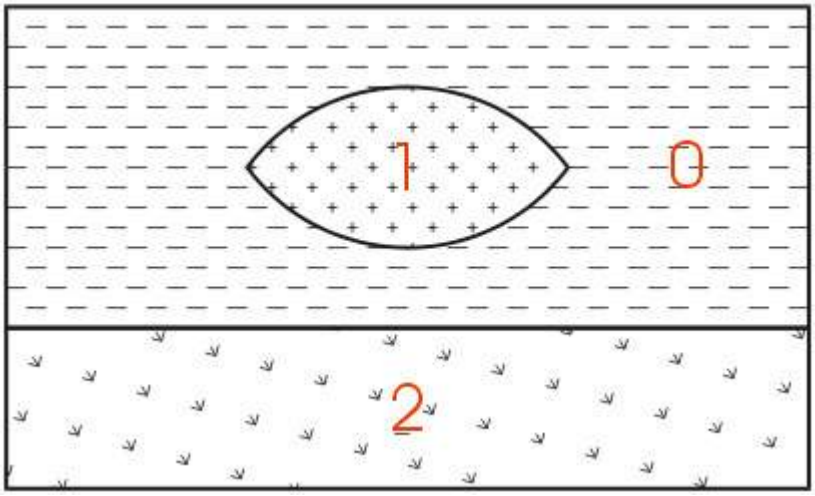
If we want to be able to create a graphical representation of the borings or wells from which the samples were taken, the analyte (e.g. chemistry) data file should also include the boring or well name associated with each sample and the ground surface elevation at the location of that boring.

The chapter on analyte (e.g. chemistry) Data Files includes an in-depth look at the format used by C Tech Development Corporation's Environmental Visualization System (EVS).

Geology

Geologic information is considerably more difficult to represent in a single, unified data format because of its nature and complexity. Geologic data files can be grouped into one of two classes, those representing interpreted geology and those representing boring logs. By some definitions, boring logs are interpreted since a geologist was required to assign materials based on core samples or some other quantitative measurements. However, for this discussion interpreted geology data will be defined as data organized into a geologic hierarchy.

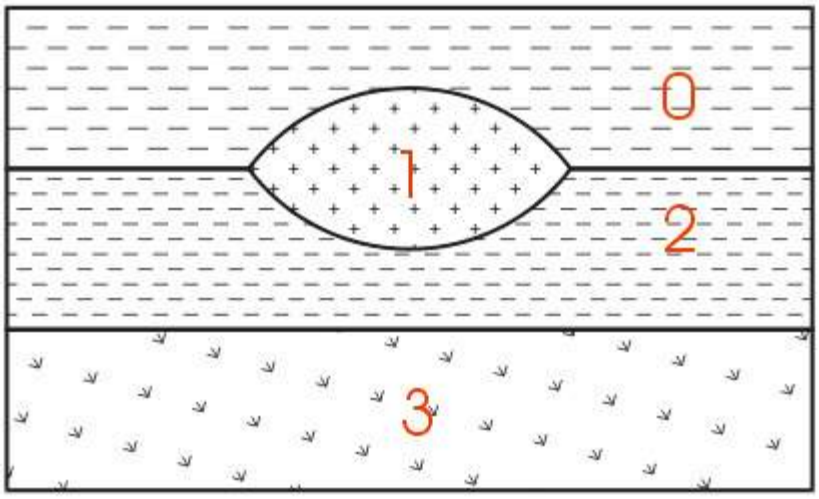
C Tech's software utilizes one of two different ASCII file formats for interpreted geologic information. These two file formats both describe points on each geologic surface (ground surface and bottom of each geologic layer), based on the assumption of a geologic hierarchy. Simply stated, geologic hierarchy requires that all geologic layers throughout the domain be ordered from top to bottom and that a consistent hierarchy be used for all borings. At first, it may not seem possible for a uniform layer hierarchy to be applicable for all borings. Layers often pinch out or exist only as localized lenses. Also layers may be continuous in one portion of the domain, but are split by another layer in other portions of the domain. However, all of these scenarios and many others can be usually be modeled using a hierarchical approach.



The easiest way to describe geologic hierarchy is with an example. Consider the example above of a clay lens in sand with gravel below.

Imagine borings on the left and right sides of the domain and one in the center. Those outside the center would not detect the clay lens. On the sides, it appears that there are only two layers in the hierarchy, but in the middle there are three materials and four layers.

EVS's & MVS's hierarchical geologic modeling approach accommodates the clay lens by treating every layer as a sedimentary layer. Because we can accommodate "pinching out" layers (making the thickness of layers ZERO) we are able to produce most geologic structures with this approach. Geologic layer hierarchy requires that we treat this domain as 4 geologic layers. These layers would be Upper Sand (0), Clay (1), Lower Sand (2) and Gravel (3).



If desired, both Upper and Lower Sand can have identical colors or hatching patterns in the final output.

Figure 0.1 Geologic Hierarchy of Clay Lens in Sand

When this geologic model is visualized in 3D, both Upper and Lower Sand can have identical colors or hatching patterns. Since the layers will fit together seamlessly, dividing a layer will not change the overall appearance (except when layers are exploded).

For sites that can be described using the above method, it is generally the best approach for building a 3D geologic model. Each layer has smooth boundaries and the layers (by nature of hierarchy) can be exploded apart to reveal the individual layer surface features. An example of a much more complex site is shown below in Figure 1.3. Sedimentary layers and lenses are modeled within the confines of a geologic hierarchy.

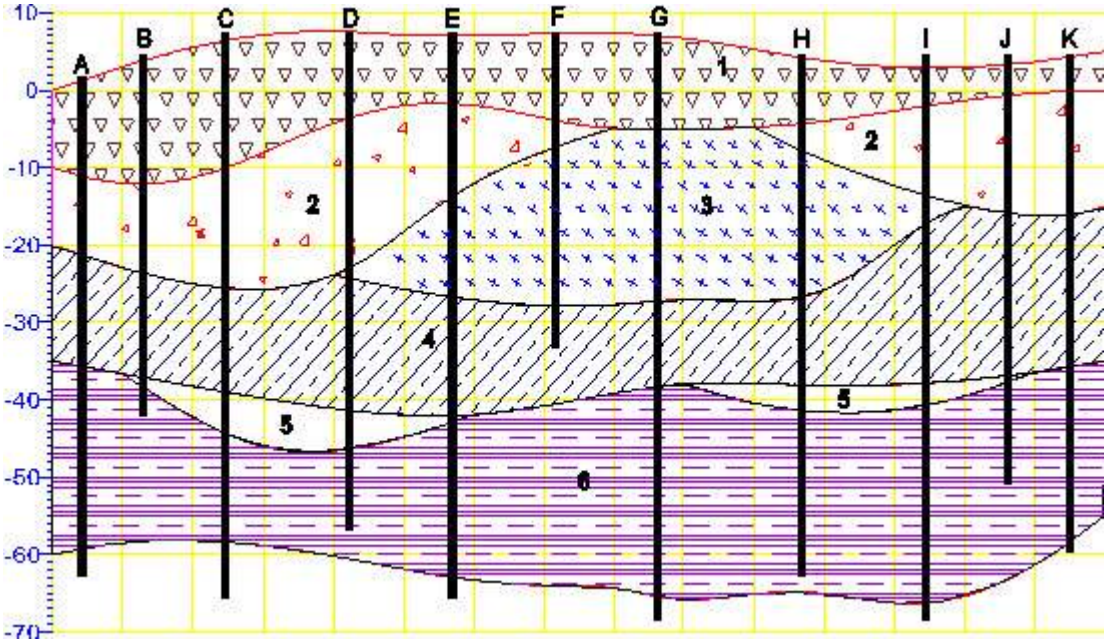


Figure 0.2 Complex Geologic Hierarchy

The hierarchical borehole based geology file format used for Figure 1.3 is described in the chapter on Borehole Geology Files.

With C Tech's EVS software, there are two other geology file formats. One of them is a more generic format for interpreted (hierarchical) geologic information. With that format; x, y, and z coordinates are given for each surface in the model. There is no requirement for the points on each surface to have coincident x-y coordinates or for each surface to be defined with the same number of points. The borehole geology file format described above could always be represented with this more generic file format.

The last file format is used to represent the materials observed in each boring. Borings are not required to be vertical, nor is there any requirement on the operator to determine a geologic hierarchy. C Tech refers to this file format as Pregeology referring to the fact that it is used to represent raw 3D boring logs. This format is also considered to be "uninterpreted". This is not meant to imply that no form of geologic evaluation or interpretation has occurred. On the contrary, it is required that someone categorizes the materials on the site and in each boring.

In C Tech's EVS software, the raw boring data can be used to create complex geologic models directly using a process called Geologic Indicator Kriging (GIK). The GIK process begins by creating a high-resolution grid constrained by ground surface and a constant elevation floor or some other meaningful geologic surface such as rockhead. For each cell in the grid, the most probable geologic material is chosen using the surrounding nearby borings. Cells of common material are grouped together to provide visibility and rendering control over each material.

The pregeology file format is discussed in this chapter.

Gridding and Dimensionality

Although there is great value in directly visualizing measured data; it does have many limitations. Without mapping sparse measured data to a grid, computation of contaminant areas or volumes is not possible. Further, the techniques available for visualizing the data are very limited. For these reasons and more, significant attention should be paid to the process of creating a grid into which the data will be interpolated and extrapolated.

For this paper, a grid is defined as a collection of nodes and cells. Nodes are points in two or three-dimensions with coordinates and usually one or more data values. The word "cell" and "element" are both used as a generic term to refer to geometric objects. The cell type and the nodes that comprise their vertices define these objects. Commonly used cell types are described in Table 1.1 and Figure 1.8.

Cell Type	Number of Nodes	Dimensionality
Point	1	0
Line	2	1
Triangle	3	2

Quadrilateral	4	2
Tetrahedron	4	3
Pyramid	5	3
Prism	6	3
Hexahedron	8	3

Table 0.1 Common Cell Types

Dimensionality refers to the space occupied by the cell. Points have do not have length, width, or height, therefore their dimensionality is zero (0). Lines are dimensionality "1" because they have length. Dimensionality 2 objects such as quadrilaterals (quad) and triangles have area and dimensionality 3 objects ranging from tetrahedrons (tet) to hexahedrons (hex) are volumetric. When creating a two-dimensional grid, areal cells are used and for three-dimensional grids, volumetric cells are used.

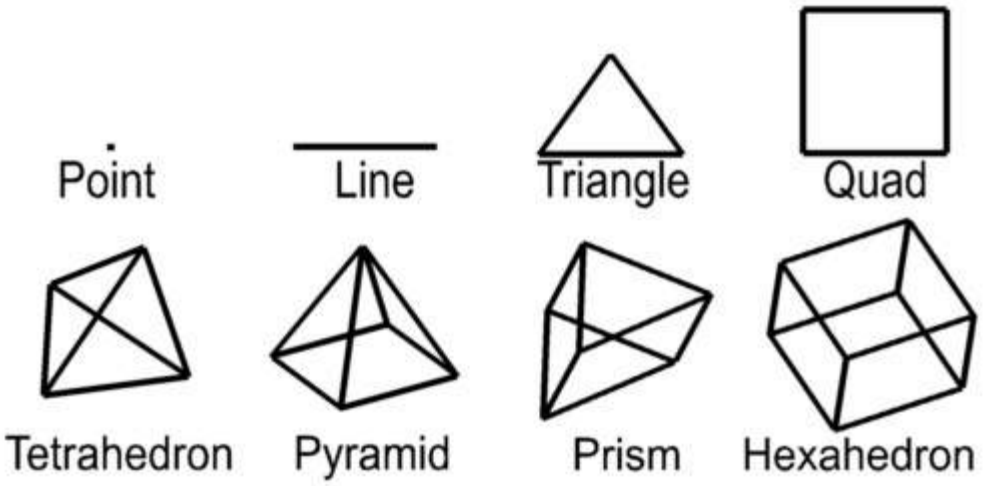


Figure0.7Common Cell Types

Direct Data Visualization

Many methods of environmental data visualization require mapping (interpolation and/or extrapolation) of sparse measured data onto some type of grid. Whenever this is done, the visualization includes assumptions and uncertainties introduced by both the gridding and interpolation processes. For these reasons, it is crucial to incorporate direct visualization of the data as a part of the entire process. It becomes the operator's responsibility to ensure that the gridding and interpolation methods accurately represent the underlying data.

A common means for directly visualizing environmental data is to use glyphs. A "glyph" refers to a graphical object that is used as a symbol to represent an object or some measured data. For the purposes of this paper, glyphs will be positioned properly in space and may be colored and/or sized according to some data value. For a graphics display, the simplest of all glyphs would be a single pixel. A pixel is a dot that is drawn on the computer screen or

rendered to a raster image. The issue of pixel size often creates confusion. Pixels (by definition) do not have a specific size. Their apparent size depends on the display (or printer) characteristics. On a computer screen, the displayed size of a pixel can be determined by dividing the screen width in inches or millimeters by the screen resolution in pixels. For example, a 19" computer monitor has a screen width of about 14.5 inches. If the "Desktop Area" is set to 1280 by 1024, the width of a pixel would be approximately 0.011 inches (~0.29 mm). If the "Desktop Area" were reduced, the apparent size of a pixel would increase.

There are virtually no limits to the type of glyph objects that may be used. Glyphs can be simple geometric objects (e.g. triangles, spheres, and cubes) or they can be representations of real-world objects like people, trees or animals.

Glyphs in 2D

For two-dimensional displays we generally use glyph objects which are two-dimensional (having no depth or z-coordinate information). Figure 1.4 is an example of such a display.

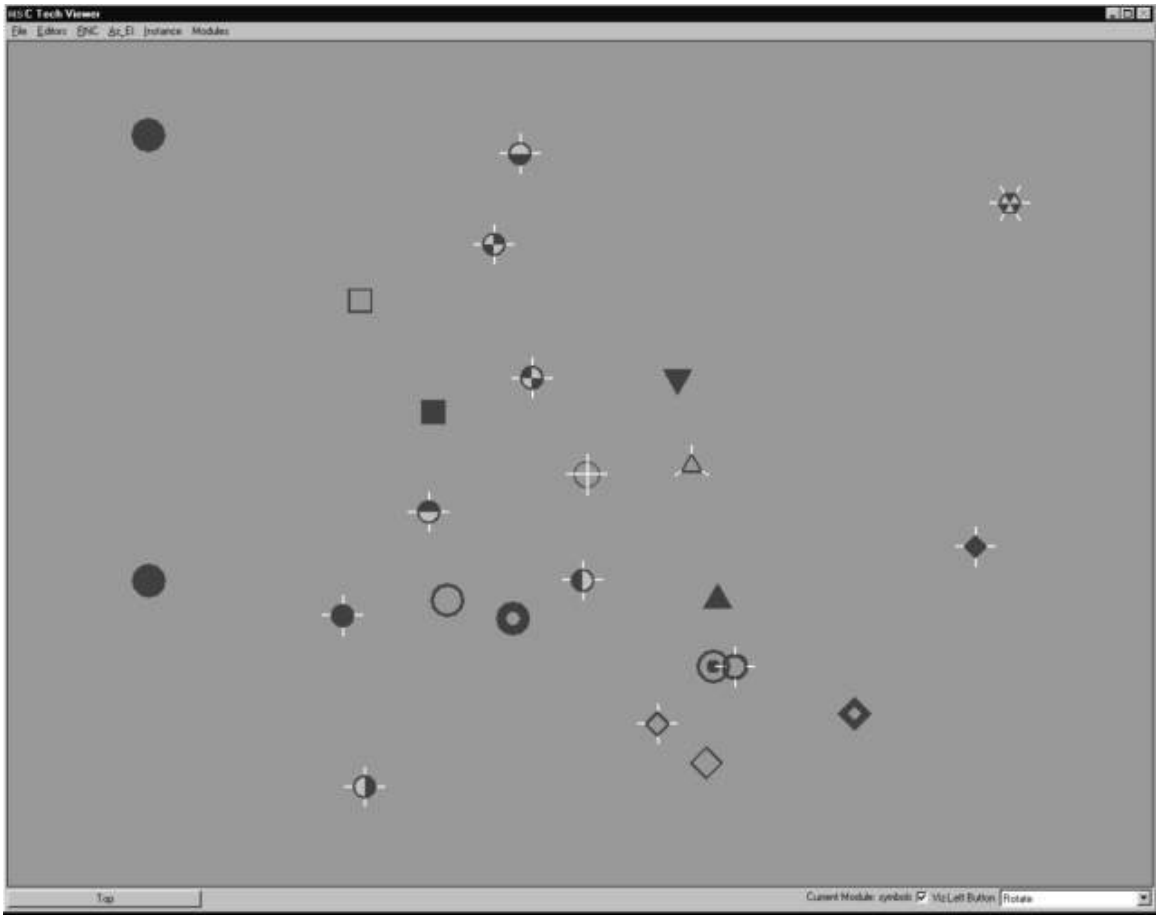


Figure 0.3 Two-Dimensional Glyphs

Glyphs in 3D

It is once we move to the three-dimensional world that glyphs become much more interesting. In Figure 1.5, cubes (hexahedron elements) are positioned, sized and colored to represent chemical measurements made in soil at a railroad yard in Sacramento, California. Axes were added to provide coordinate references and this picture was rendered with perspective effects turned on. This results in a visualization where parallel lines do not remain parallel and objects in the foreground appear larger than those in the background.

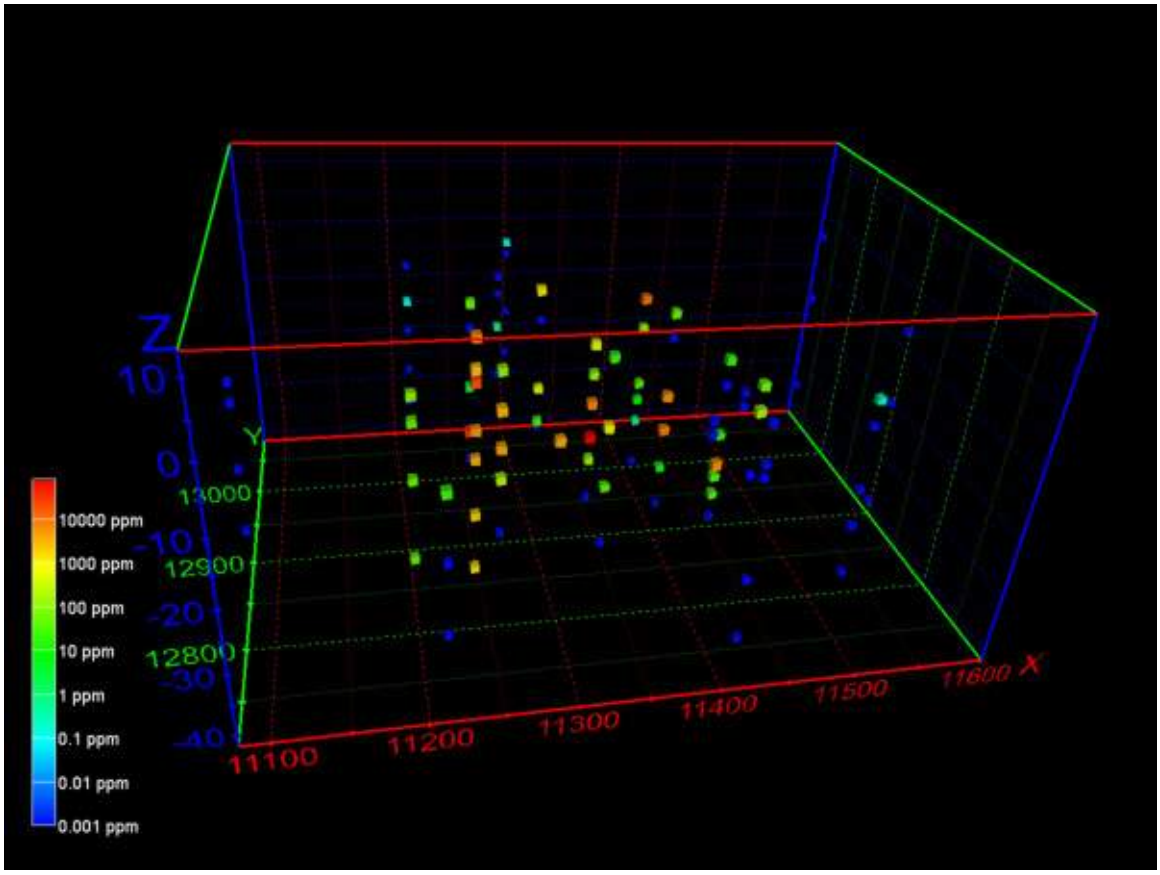


Figure 0.4 Three-Dimensional Cubic Glyphs

When representations of the borings are added, the figure becomes much more useful. Figure 1.6 shows the sample represented by colored spheres and tubes represent the borings. The tubes are colored alternating dark and light gray where the color changes on ten-foot intervals. This provides a reference to allow the viewer to quickly determine the approximate depth of the samples. The borings are also labeled with their designation. These last two figures both represent the same data, however it is clear which one provides the most useful information.

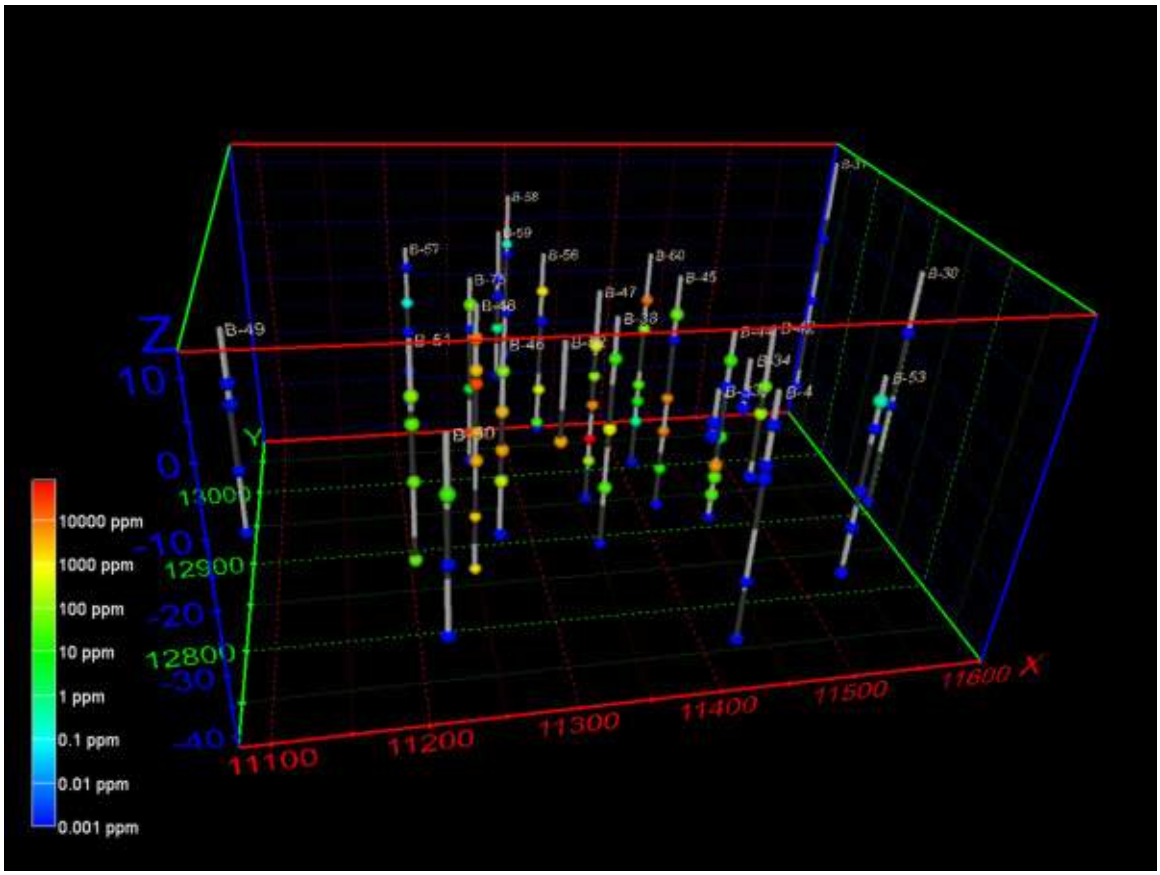


Figure 0.5 Three-Dimensional Glyphs with Boring Tubes

Glyphs can also be used to represent vector data. The most commonly encountered vector data represents ground water flow velocity. In this case, the glyph is not only colored and sized according to the magnitude of the velocity vector, but the glyph can also be oriented to point in the vector's direction. For this type of application, an assymetric glyph (as opposed to a sphere or cube) is used. Figure 1.7 uses a glyph that is referred to as "jet". It is an elongated tetrahedron that points in the direction of the vector. The data represented in this figure is predicted velocities output from a MODFLOW simulation to predict the groundwater flow field resulting from the dewatering of a gold mine pit.

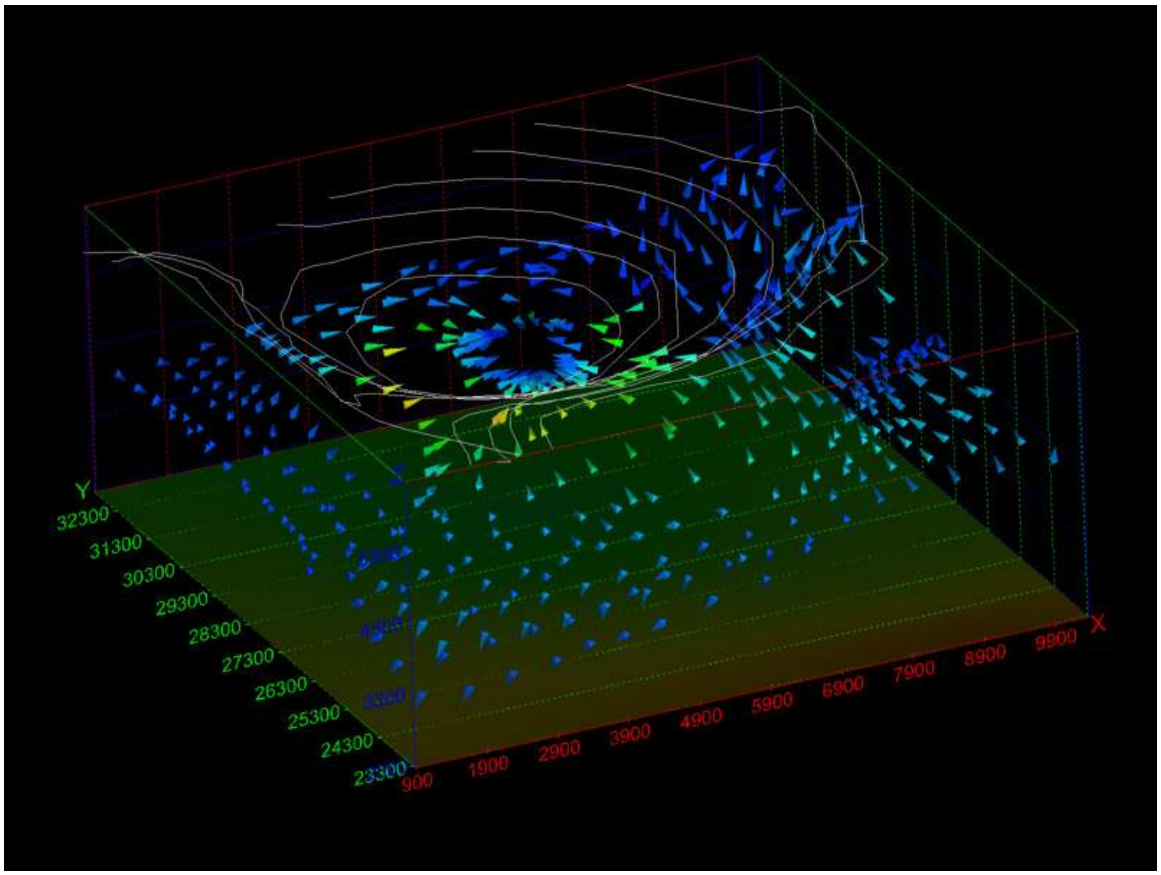


Figure 0.6 Three-Dimensional Glyphs Representing Vector Data

Rectilinear Grids

Rectilinear (a.k.a. uniform) grids are among the simplest type of grid. The grid axes are parallel to the coordinate axes and the cells are always rectangular in cross-section. The positions of all the nodes can be computed knowing only the coordinate extents of the grid (minimum and maximum x, y and optionally z). Two-dimensional rectilinear grids are comprised of quadrilateral cells. For a 2D grid with i nodes in the x direction and j nodes in the y direction, there will be a total of (i - 1)*(j - 1) cells.

The connectivity of the cells (the nodes that define each cell) can be implicitly determined because the nodes and cells are numbered in an orderly fashion. The advantages of rectilinear grids include the ease of creating them and the uniformity of cell area in 2D and cell volume in 3D. The disadvantages are that grid nodes are generally not coincident with the sample data locations and large areas of the grid may fall outside of the bounds of the data. A simple two-dimensional rectilinear grid is shown in Figure 1.9.

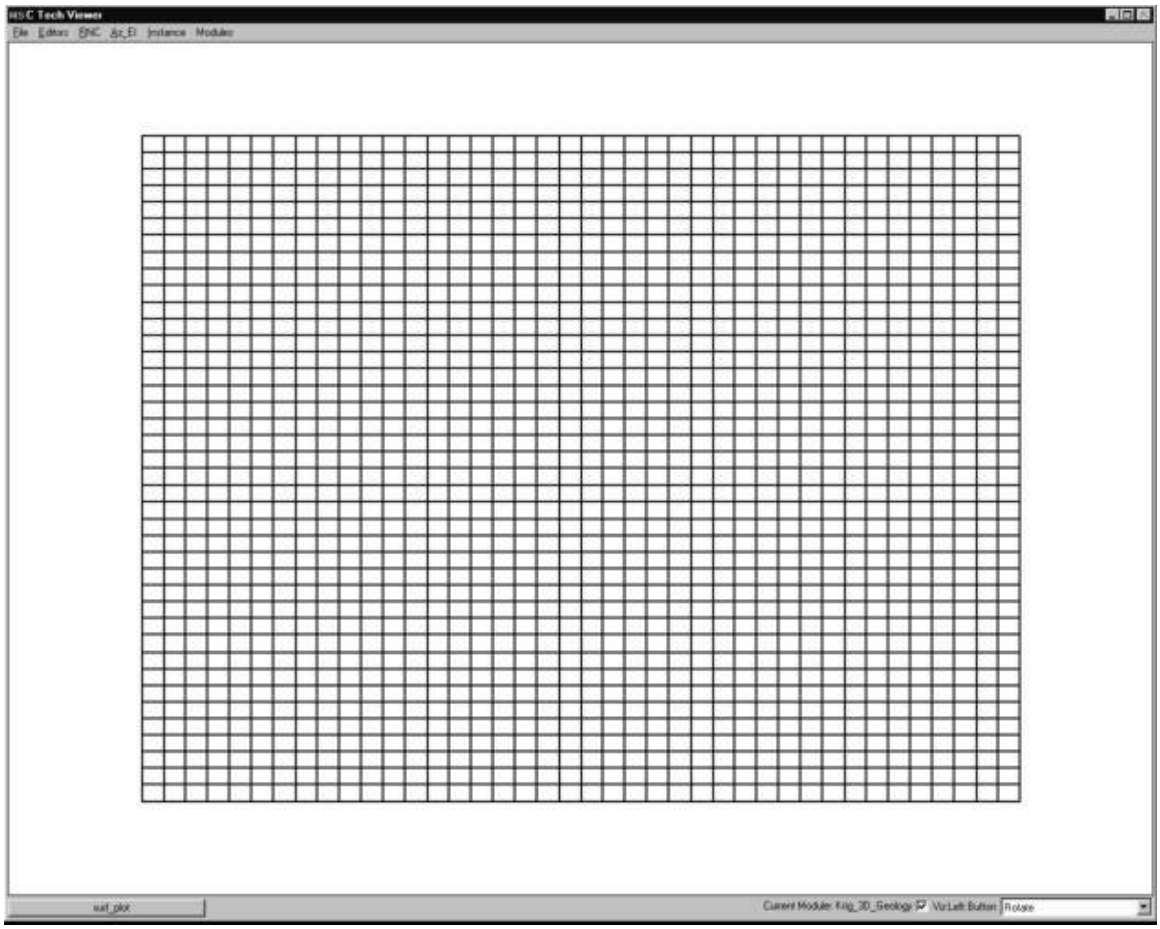


Figure 0.8 Two-Dimensional Rectilinear Grid

Three-dimensional rectilinear grids offer the simplest method for gridding a volume. They are constrained to rectangular parallel piped volumes and have hexahedral cells of constant size. (See Figure 1.10) For some processes and visualization techniques such as volume rendering, this is advantageous and may even be required. For a grid having i by j by k nodes there will be (i-1) * (j-1) * (k-1) hexahedron cells whose connectivity can be implicitly derived.

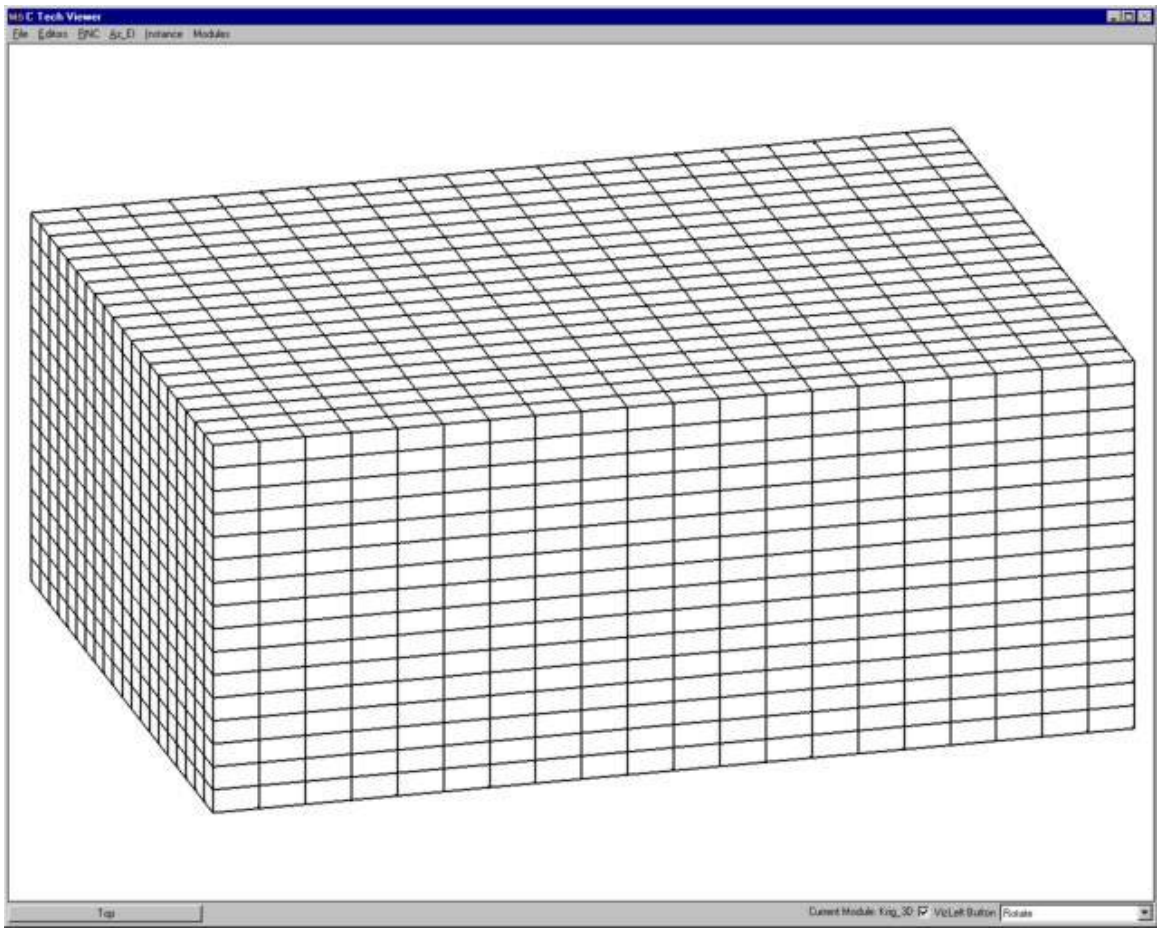


Figure 0.9 Three-Dimensional Rectilinear Grid

Finite Difference

The following type of grid derives its name from the numerical methods that it employs. Simulation software such as the USGS's MODFLOW utilizes a finite difference numerical method to solve equilibrium and transient ground water flow problems. This solution method requires a grid that contains only rectangular cells. However the cells need not be uniform in size. For two-dimensional grids, this results in rectangular cells, however it is possible that no two cells are precisely the same size. Some simulation software requires that finite difference grids be aligned with the coordinate axes. EVS does not impose this restriction, but it does provide a means to export the grid transformed so that the grid axes are aligned. Figure 1.11 shows a rotated 2D finite difference grid. Smaller cells are concentrated in areas of the model where there are significant gradients in the data. For groundwater simulations this is usually where wells are located. For environmental contamination it should be the location of spills or areas where DNAPL (dense non-aqueous phase liquids) contaminant plumes were detected. The smaller cells provide greater accuracy in estimating the parameter(s) of interest.

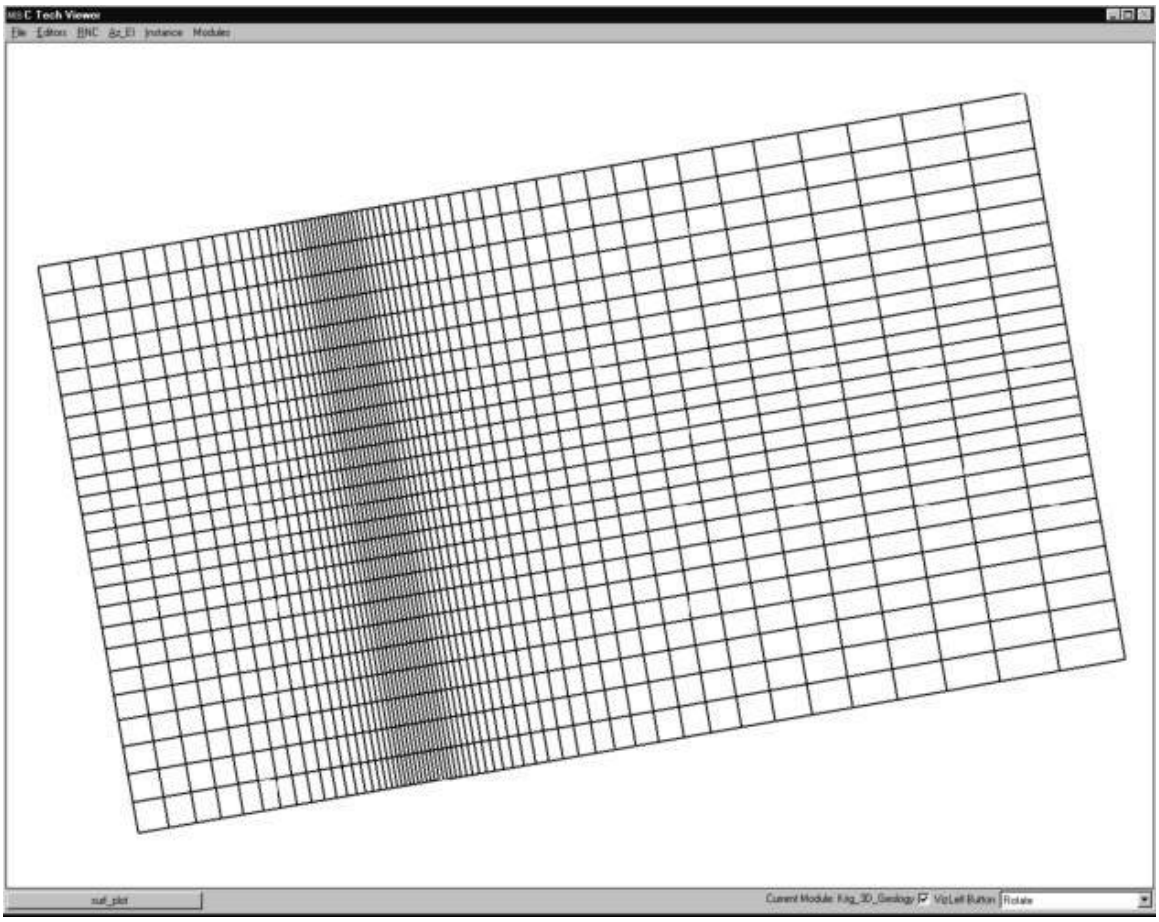


Figure 0.10 Two-Dimensional Rotated Finite Difference Grid

Three-dimensional finite difference grids have the same restrictions as 2D grids with respect to their x and y coordinates (cell width and length). However, the z coordinates of the grid (which define the cell thicknesses) are allowed to vary arbitrarily. This allows for creation of a grid that follows the contours of geologic surfaces. For a grid having i by j by k nodes there will be (i-1) * (j-1) * (k-1) hexahedron cells whose connectivity can be implicitly derived. However the coordinates of the nodes for this grid must be explicitly specified. Figure 1.12 shows the grid created to model the migration of a contaminant plume in a tidal basin.

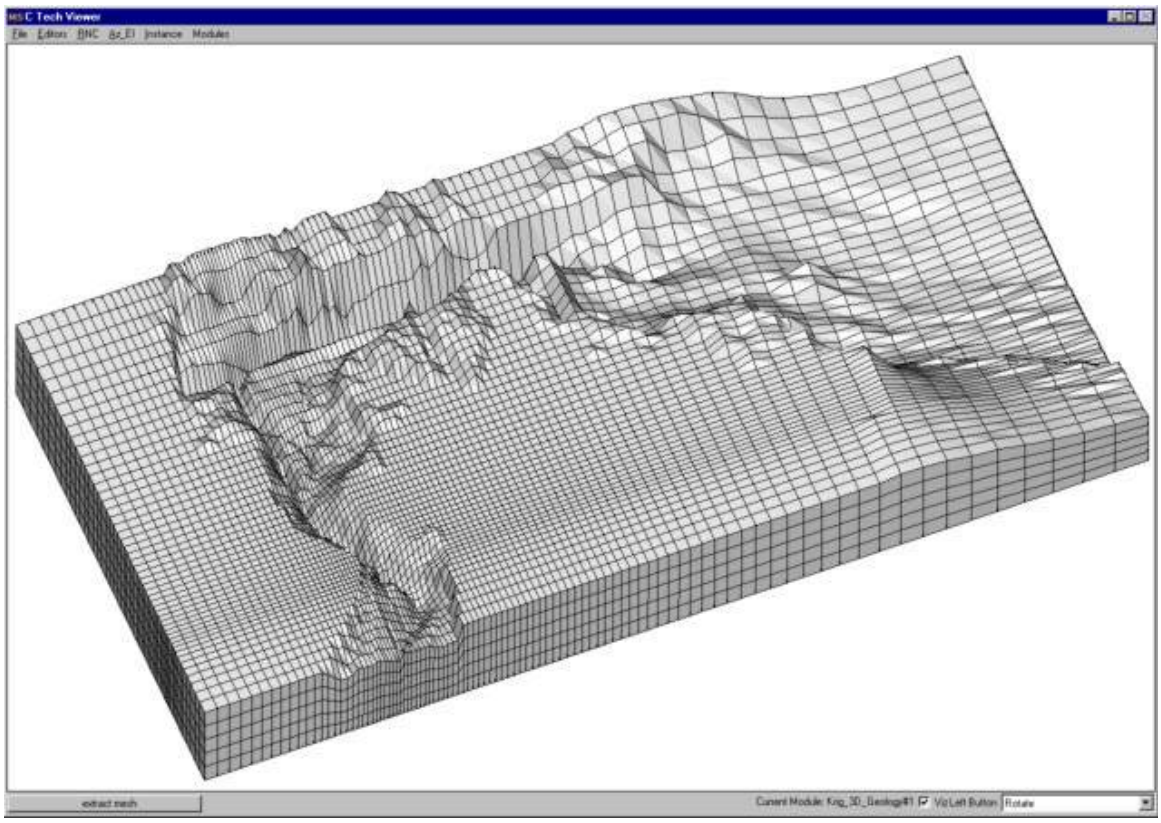


Figure 0.11 Three-Dimensional Finite Difference Grid

Convex Hull

The convex hull of a set of points in two-dimensional space is the smallest convex area containing the set. In the x-y plane, the convex hull can be visualized as the shape assumed by a rubber band that has been stretched around the set and released to conform as closely as possible to it. The area defined by the convex hull offers significant advantages. Within the convex hull all parameter estimates are interpolations. The convex hull best fits the spatial extent of the data. Remember that the convex hull defines an area. That area can be gridded in many ways. EVS grids convex hull regions with quadrilaterals. Smoothing techniques are used to create a grid that has reasonably equal area cells. A two-dimensional example of a convex hull grid is shown in Figure 1.13. In this example, the domain of the model was offset by a constant amount from the theoretical convex hull. This results in rounded corners and a model region that is larger than the convex hull.

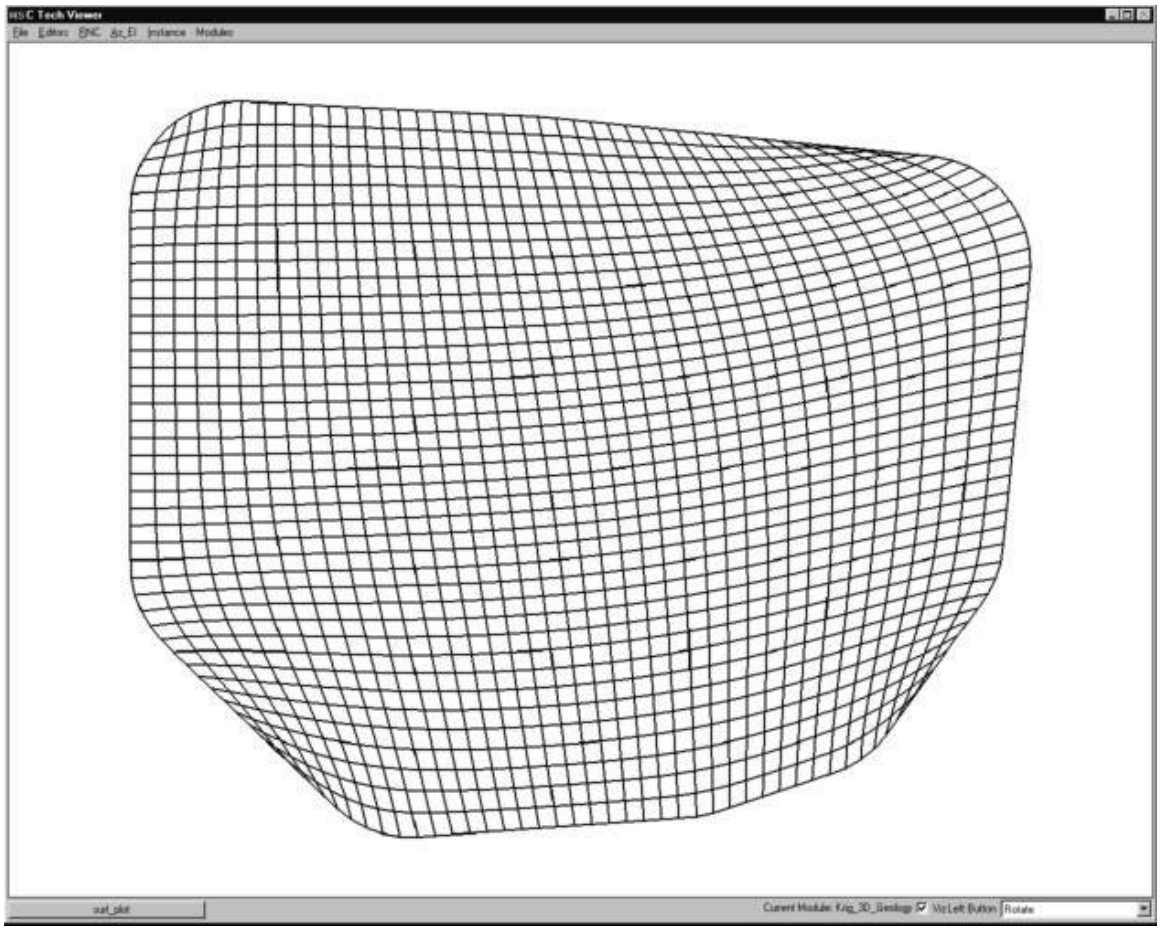


Figure 0.12 Convex Hull Grid with Offset

Adaptive Gridding

Adaptive gridding is the localized refinement of a grid to provide higher resolution in the areas or volumes surrounding measured sample data. Adaptive gridding or grid refinement can be accomplished in many different ways. In EVS, rectilinear, finite difference and convex hull grids can all be refined using a similar method. In two-dimensions a new node is placed precisely at the measured sample data location. Three additional nodes are placed to create a small quadrilateral cell within the cell to be refined. The corners of the small cell are connected to the corresponding corners of the cell being refined creating a total of five cells where the one previously was. The resulting nodal locations and grid connectivity must be explicitly defined.

Adaptively gridding offers many advantages. It assures that there will always be nodes at the precise coordinates of the sample data. This insures that the data minimum and maximum in the gridded model will match the sample data. It also provides greater fidelity in defining data trends in regions with high gradients. Figure 1.14 shows a two-dimensional adaptively gridded convex hull model. This model's area was also offset from the convex hull. Since each sample data point results in a refined region, and the sample points define the convex hull, the regions in each corner of the model contain adaptively gridded cells.

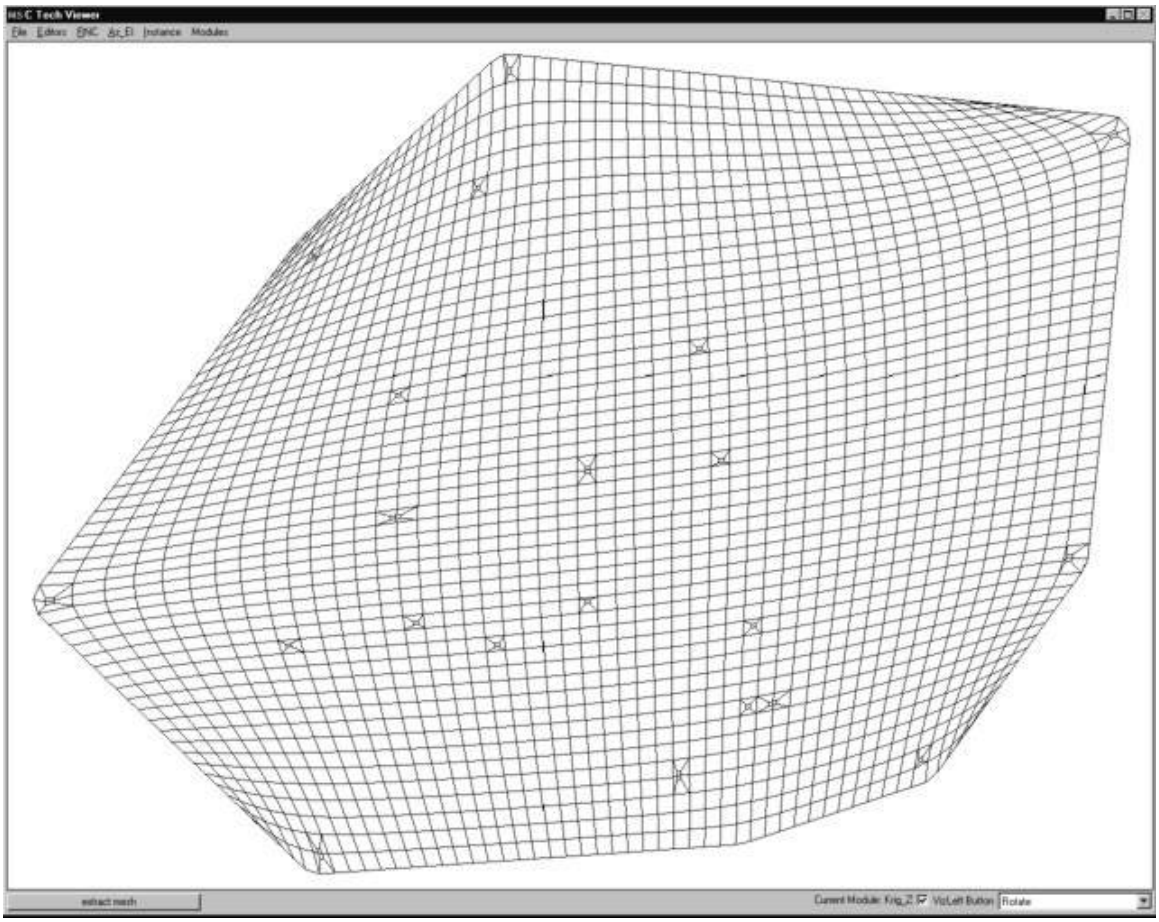


Figure 0.13 Adaptively Gridded Convex Hull Grid

Figure 1.15 is a close-up view of some refined cells near the lower right in Figure 1.14. It shows one of the special cases. If the point to be refined falls very near an existing cell edge, that edge is refined and the cells on either side of the edge are symmetrically refined. Since the edge must be broken into three segments, the cells on both sides must be affected.

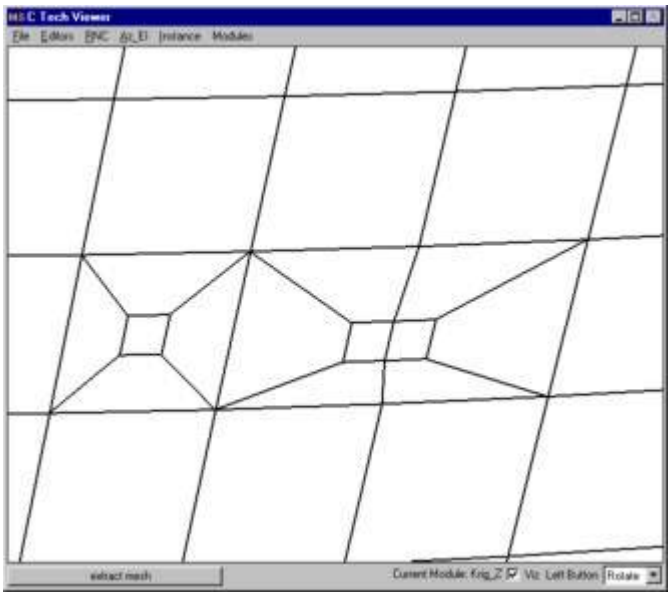


Figure 0.14 Close-up of Figure 1.14

The refinement process can also be applied to all types of 3D grids. When a sample falls in a hexahedron (hex) cell, a new much smaller hex cell is created with one of its' corners located precisely at the coordinates of the sample point. The eight corners of the small cell are connected to the corresponding corners of the parent cell. This creates 7 hex cells that fully occupy the volume of the original cell. Since the 3D-refinement process occurs internal to the volume of the model, it is more difficult to visualize the process. In order to see the refined cells, removing all cells in the grid with any nodes that were below a thresholded concentration level created Figure 1.16. By choosing the threshold properly, several of the refined cells become visible.

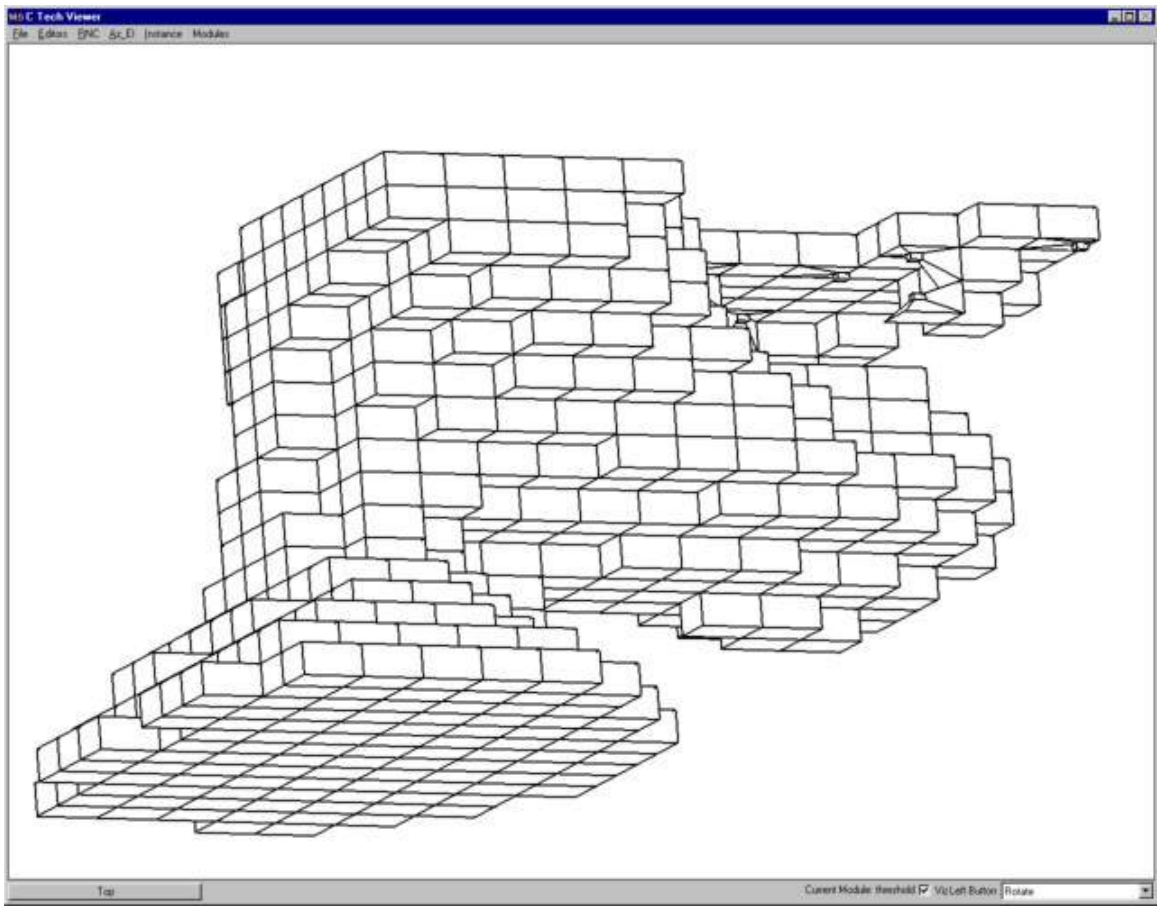


Figure 0.15 3D Adaptively Gridded Model

This figure (Figure 1.17) is an enlarged view of the upper right hand corner. It reveals the structure, relative sizes and connectivity resulting from 3D adaptive gridding.

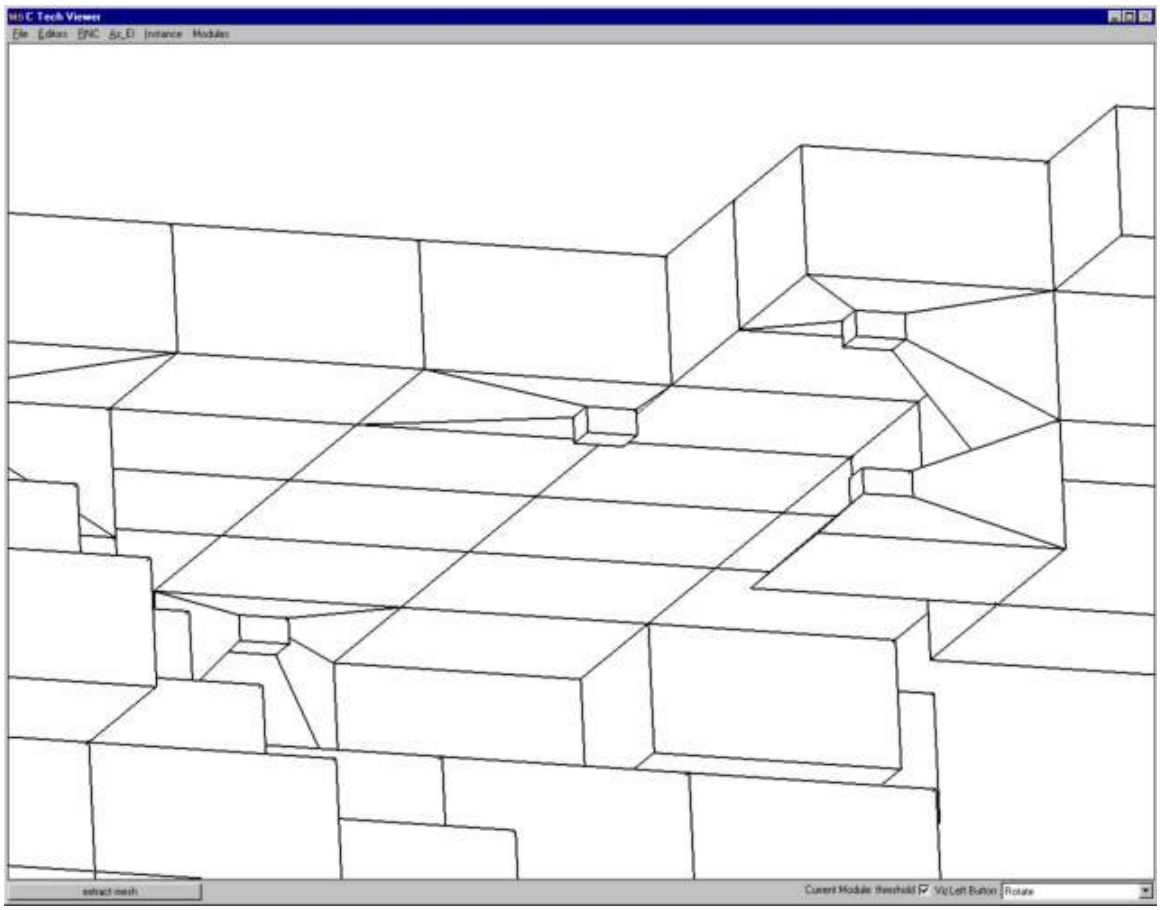


Figure 0.16 Close-up of Figure 1.16

Triangular Networks

Triangular networks are defined as grids of triangle or tetrahedron cells where all of the nodes in the grid are exclusively those in the sample data. For these types of grids, the cell connectivity must be explicitly defined. In two dimensions, these grids are referred to as Triangulated Irregular Networks or TINs. The 3D equivalent grids are Tetrahedral Irregular Networks.

Triangulated Irregular Networks – 2D

Delaunay triangulation is one of the most commonly used methods for creating TINs. By definition, 3 points form a Delaunay triangle if and only if the circle defined by them contains no other point. Focusing on creating Delaunay triangles produces triangles with fat (large) angles that have preferred rendering characteristics. The boundary edges on the Delaunay network form the convex hull, which is the smallest area convex polygon to contain all of the vertices.

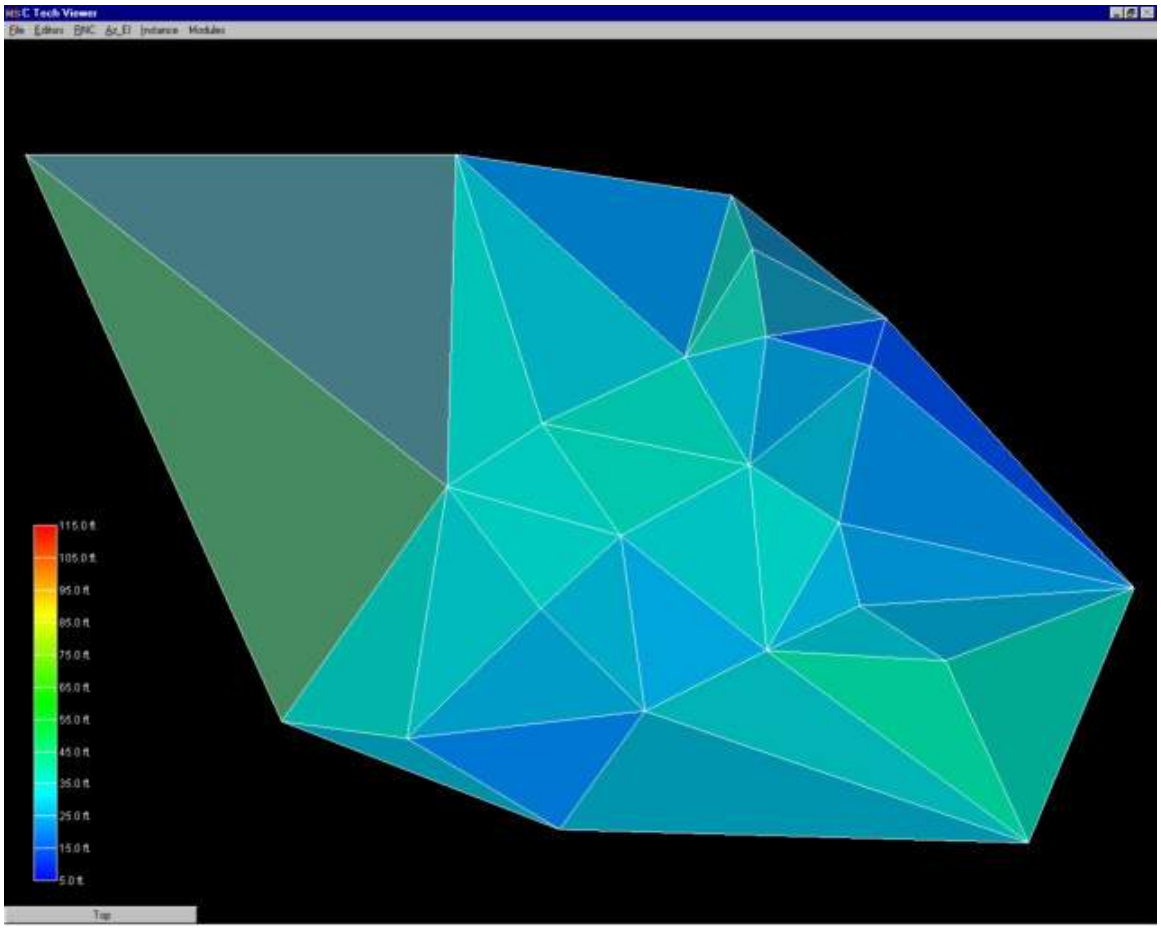


Figure 0.17 Flat-Shaded Delaunay TIN of Geologic Surface

The TIN surface above (Figure 1.18) has significant variation in the size of the triangles. This is a natural consequence of the grid's being created using only nodes from the input data file. When such a surface is rendered with data, having very large triangles can result in very objectionable visualization anomalies. These anomalies result from rendering large triangles that have a range of data values that span a significant fraction of the total data range. There are many methods that could be used to assign color to each triangle. These methods are referred to as surface rendering modes.

Two of the most commonly used rendering modes are flat shading and Gouraud shading. Flat shading assigns a single color to the entire triangle. The color is computed based on the average elevation (data value) for that triangle, lighting parameters and orientation to the viewer camera. In the upper left corner we have a large single triangle that spans a significant range of elevations. When it is assigned a color that corresponds to the mean elevation for that triangle, that color will be wrong. More precisely, the color does not fall within the color scale. Note the color of the triangle in the upper right corner of Figure 1.18 and the one below it. The color of these triangles is outside the range of our color scale.

The problem of large triangles is no better when using Gouraud shading. Gouraud shading assigns colors to each node of the triangle based on the

data values. This assures that the colors at the nodes (vertices of the triangles) **will** be correct. Colors are then interpolated over the area of the triangle based on lighting parameters and orientation to the viewer camera. Consider the triangle in the upper right hand corner of Figure 1.19. The upper right node is assigned the color blue (corresponding to a low value) and the upper left node is assigned the color red (corresponding to a high value). The color scale for this problem ranges from blue to cyan to green to yellow to red. However, for this anomalous situation the color that will be interpolated between blue and red along the uppermost edge will be magenta. Magenta is **not** a color in our range of colors.

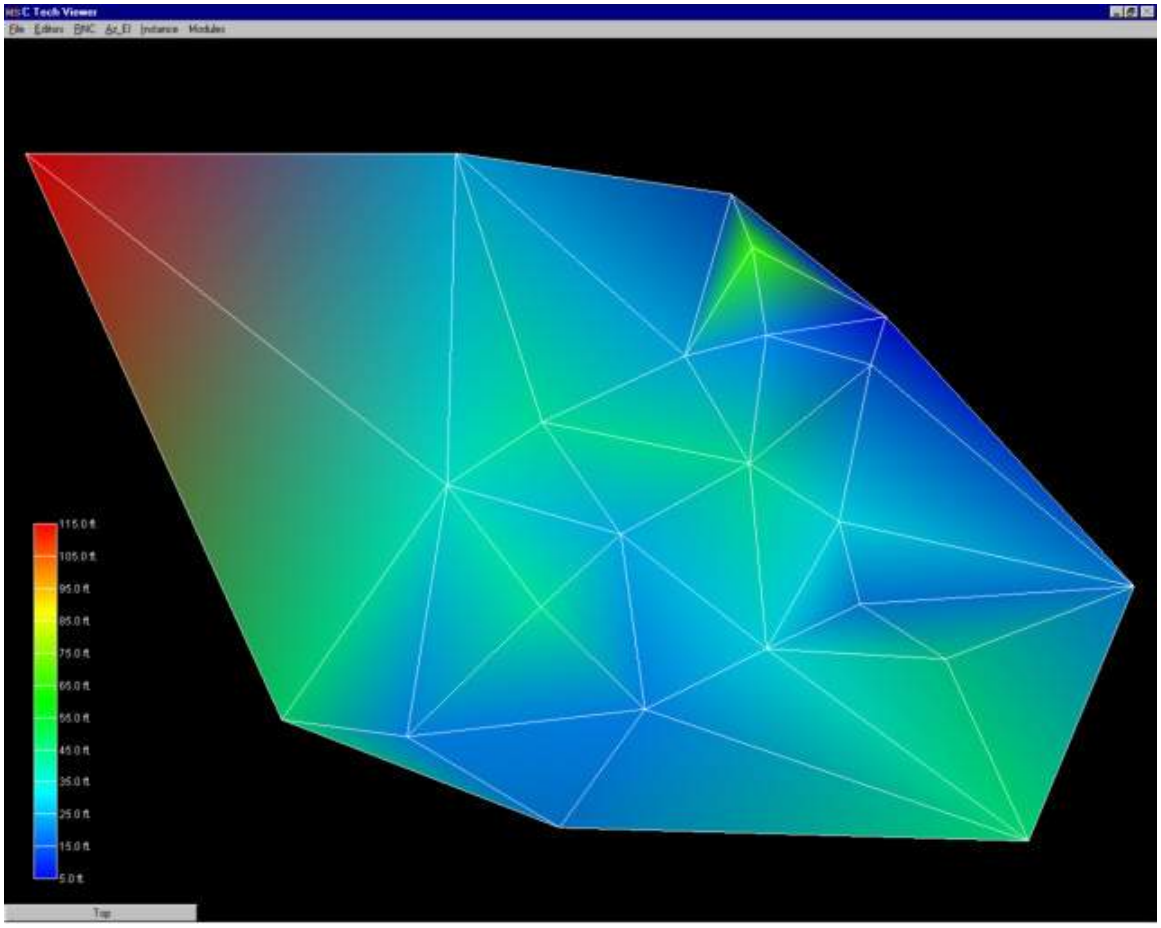


Figure 0.18 Gouraud-Shaded Delaunay TIN of Geologic Surface

To overcome the problems caused by large triangles, the triangles can be refined (subdivided) to create a grid that still contains points that honor the original input nodes, but has more uniform cell sizes. In Figure 1.20 (which has a spatial extent of 500 feet in x and 380 feet in y) it was specified that no triangle's edge may exceed 45 feet in length. We must interpolate the elevation values (or our data values) to these new nodes created as a result of the triangle subdivision. The simplest means of doing this is bilinear interpolation. The refined TIN grid with bilinear interpolation and flat shaded triangles is shown in Figure 1.21. Note that the all of the triangles have appropriate colors. To avoid the large cell coloring problem (this is a problem

with all cell types except points), no single cell should have data values at its nodes that span more than about 20 percent of the total data range.

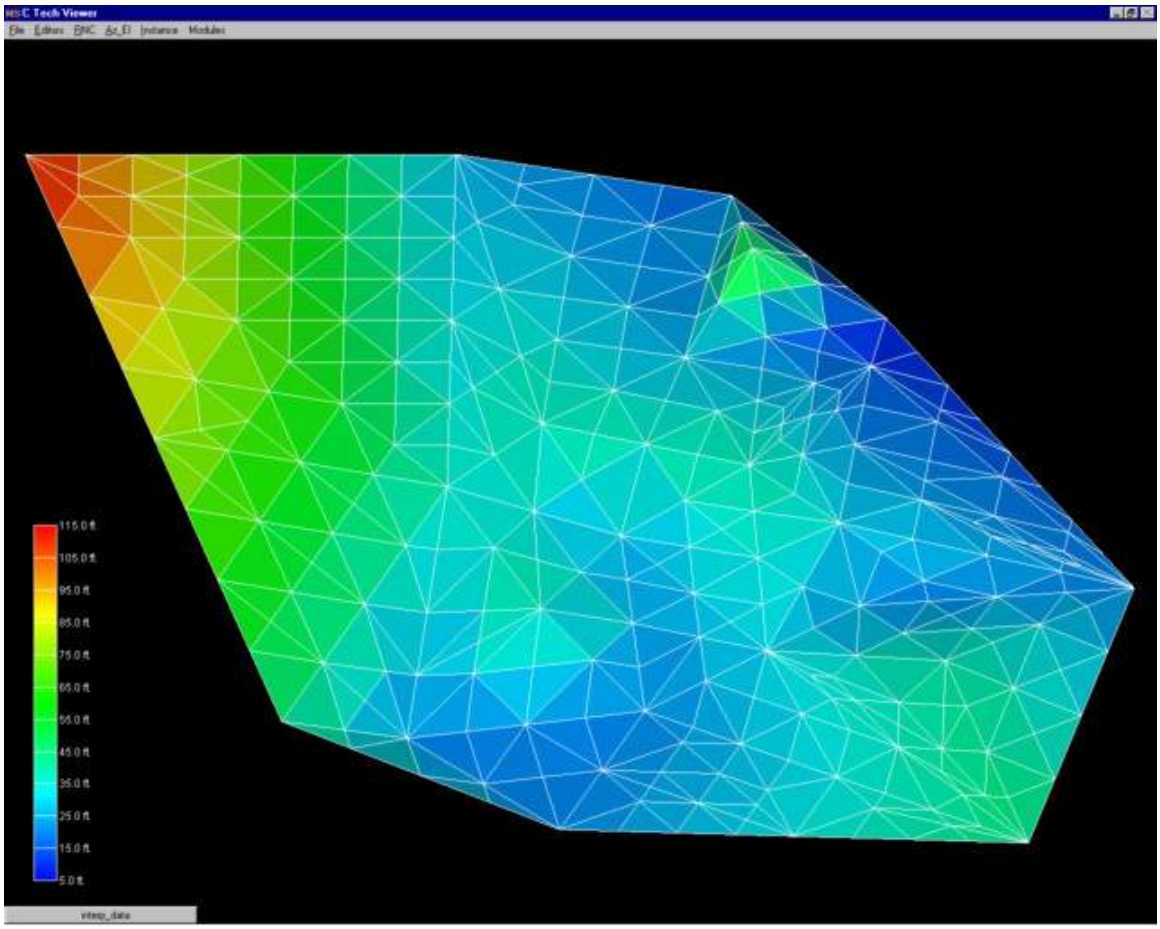


Figure 0.19 Flat-Shaded Subdivided TIN of Geologic Surface

If Gouraud shading is employed instead of flat shading, the resultant surface has a smoother appearance, however the fundamental linear interpolation along cell edges is still evident in the colors. If the maximum triangle size were made much smaller, the flat shaded model would approach the appearance of the Gouraud shaded model. However, without using a different interpolation approach the Gouraud-shaded model would not change dramatically.

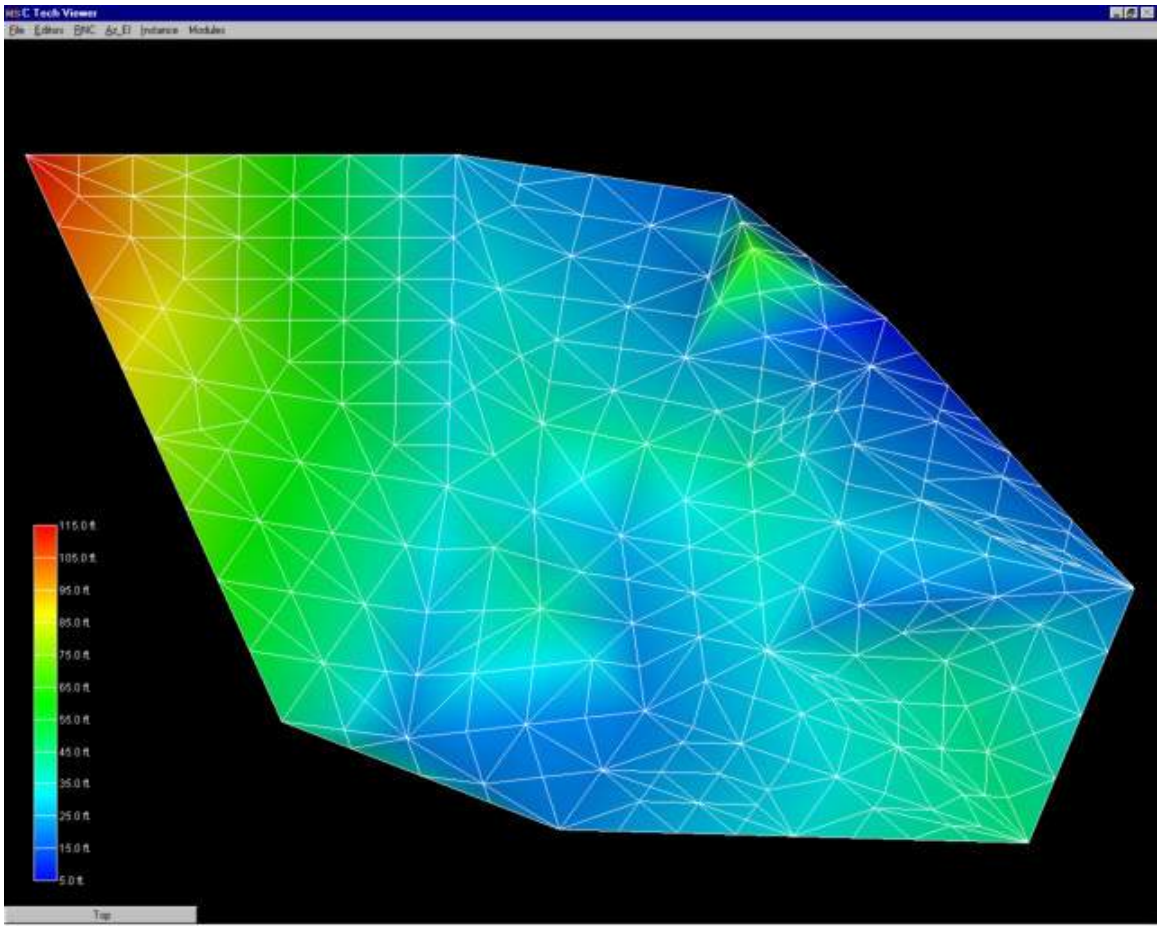


Figure 0.20 Gouraud-Shaded Subdivided TIN of Geologic Surface

EVS includes another technique for coloring surfaces. This method, called solid contours, assigns uniform color bands based on the data values. Figure 1.22 demonstrates this method that subdivides cells using bilinear interpolation. Because this method inherently includes triangle subdivision using bilinear interpolation, the figure would be identical whether the input grid was the large triangles from the original TIN surface or the refined smaller triangles. The boundaries of the colored bands are effectively isopachs (isolines) of constant elevation.

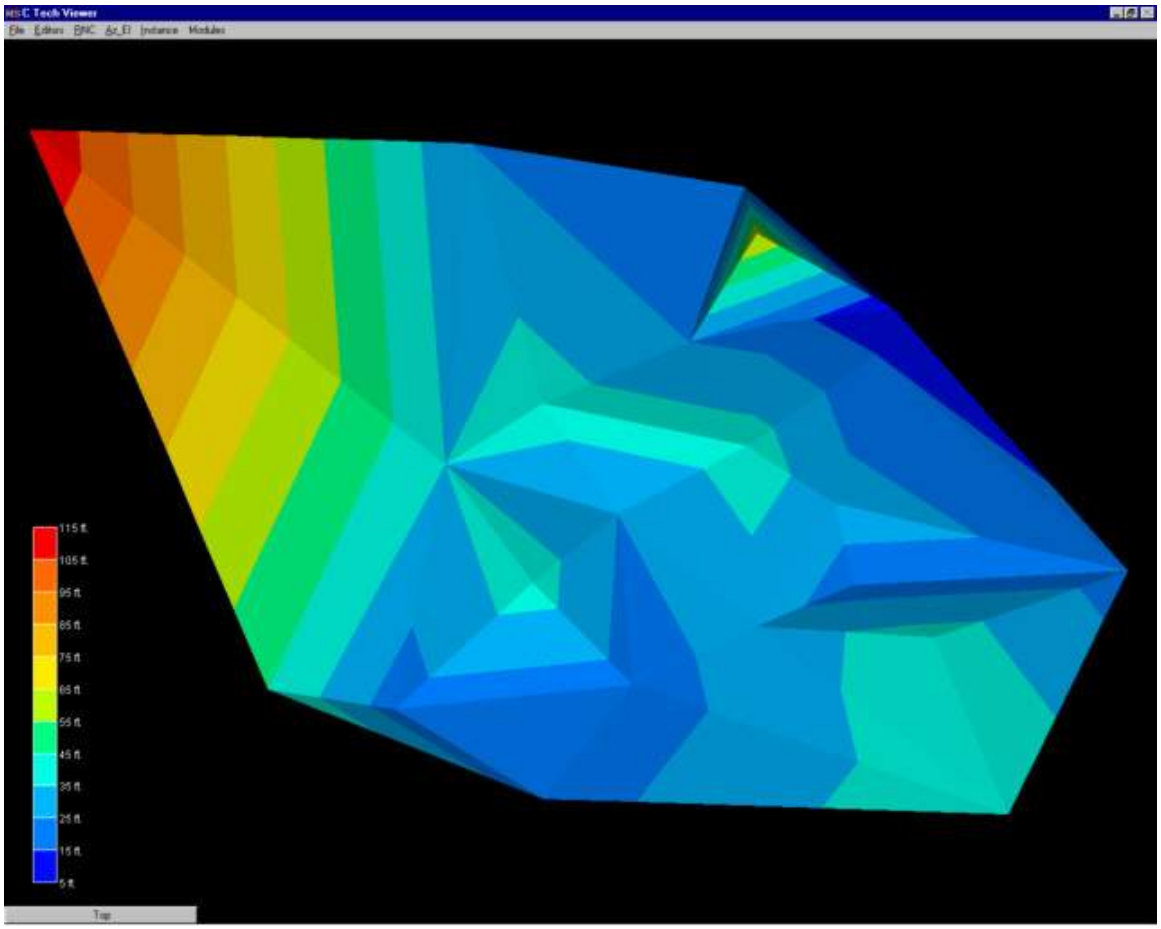


Figure 0.21 Solid Contour TIN of Geologic Surface

To complete this discussion and comparison of gridding and interpolation methods, the same data file was used to create a convex hull grid and the elevation data was estimated using EVS's two-dimensional kriging software. Kriging will be discussed in more detail in section 1.3.3. This technique honors all of the original data points, but creates much smoother distributions between the values. The result shown in Figure 1.23 is a more realistic and aesthetically superior surface. Labeled isolines on 10 foot intervals were added to this figure. Note that these isolines are similar, but much smoother than those in Figure 1.22.

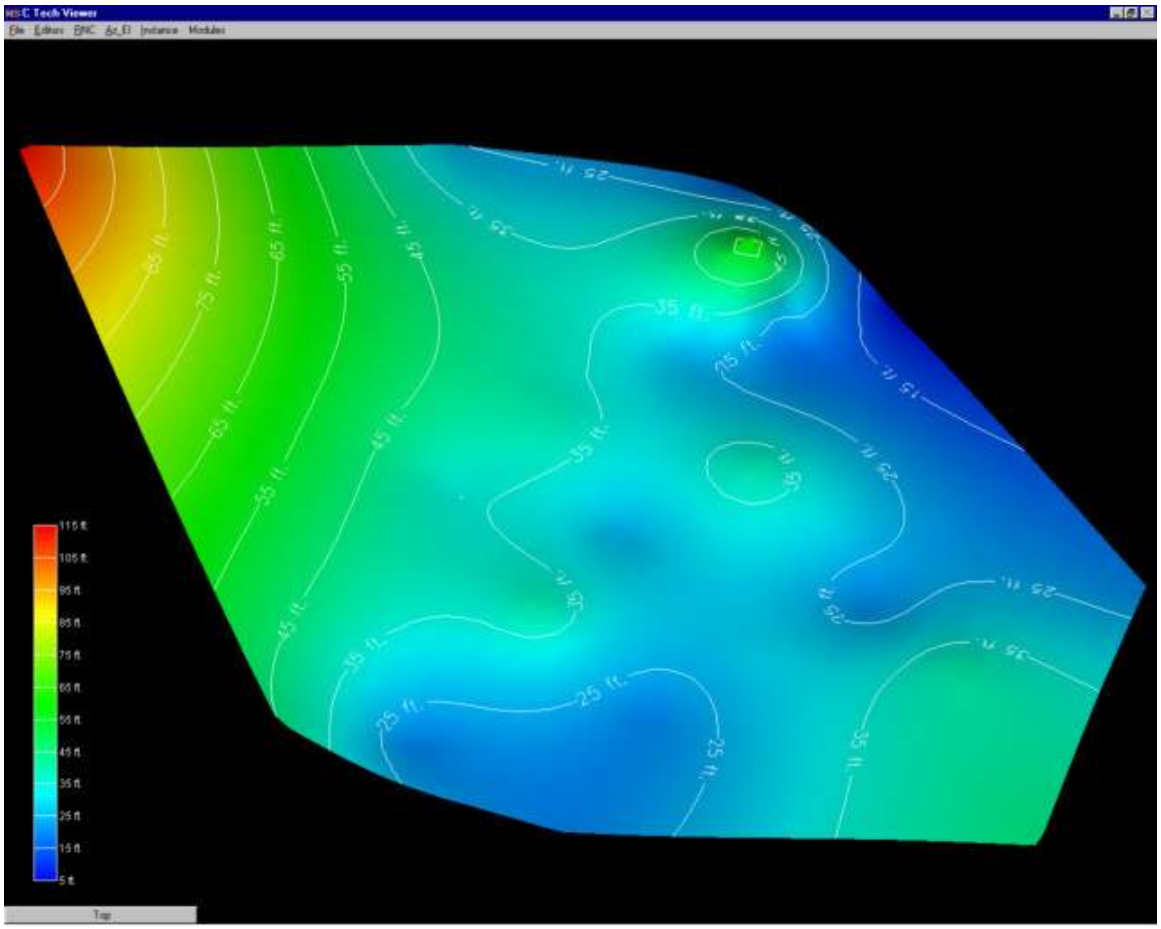


Figure 0.22 Kriged 2D Convex Hull Grid

Tetrahedral Irregular Networks – 3D

Tetrahedral Irregular Networks provide a method to create a volumetric representation of a three-dimensional set of points. As with a TIN, the nodes in the resulting grid are exclusively those in the original measured sample data. Tetrahedral Irregular Networks use tetrahedron cells to fill the three-dimensional convex hull of the data as shown in Figure 1.24. The result often contains cells of widely varying volumes having potentially large data variation across individual cells. For this and other reasons, this approach is not often used.

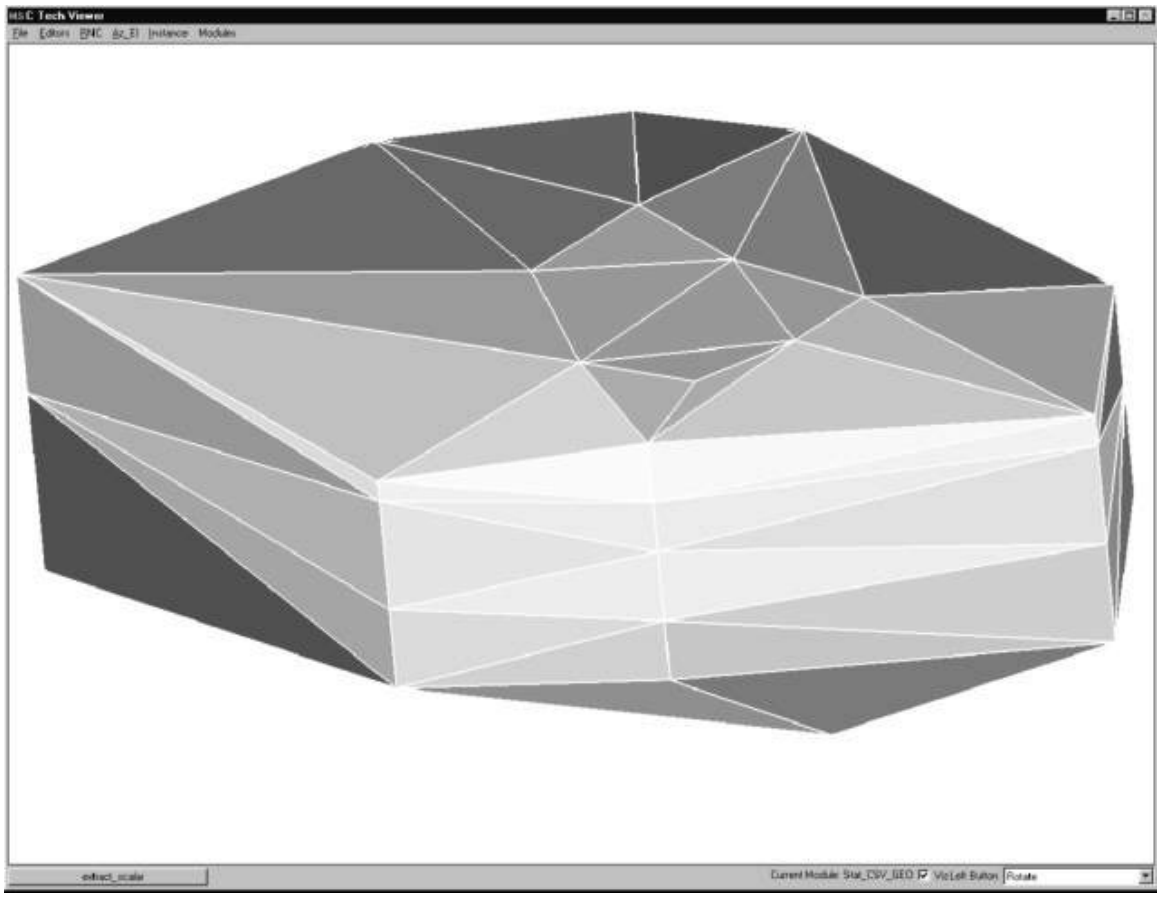


Figure 0.23 Tetrahedral Irregular Network

Interpolation Methods

Spatial interpolation methods are used to estimate measured data to the nodes in grids that do not coincide with measured points. The spatial interpolation methods differ in their assumptions, methodologies, complexity, and deterministic or stochastic nature.

Inverse Distance Weighted

Inverse distance weighted averaging (IDWA) is a deterministic estimation method where values at grid nodes are determined by a linear combination of values at known sampled points. IDWA makes the assumption that values closer to the grid nodes are more representative of the value to be estimated than samples further away. Weights change according to the linear distance of the samples from the grid nodes. The spatial arrangement of the samples does not affect the weights. IDWA has seen extensive implementation in the mining industry due to its ease of use. IDWA has also been shown to work well with noisy data. The choice of power parameter in IDWA can significantly affect the interpolation results. As the power parameter increases, IDWA approaches the nearest neighbor interpolation method where the interpolated value simply takes on the value of the closest sample point. Optimal inverse distance weighting is a form of IDWA where the power parameter is chosen on the basis of minimum mean absolute error.

Splining

Splining is a deterministic technique to represent two-dimensional curves on three-dimensional surfaces. Splining may be thought of as the mathematical equivalent of fitting a long flexible ruler to a series of data points. Like its physical counterpart, the mathematical spline function is constrained at defined points. Splines assume smoothness of variation. Splines have the advantage of creating curves and contour lines that are visually appealing. Some of splining's disadvantages are that no estimates of error are given and that splining may mask uncertainty present in the data. Splines are typically used for creating contour lines from dense regularly spaced data. Splining may, however, be used for interpolation of irregularly spaced data.

Geostatistical Methods (Kriging)

Kriging is a stochastic technique similar to inverse distance weighted averaging in that it uses a linear combination of weights at known points to estimate the value at the grid nodes. Kriging is named after D.L. Krige, who used kriging's underlying theory to estimate ore content. Kriging uses a variogram (a.k.a. semivariogram) which is a representation of the spatial and data differences between some or all possible "pairs" of points in the measured data set. The variogram then describes the weighting factors that will be applied for the interpolation. Unlike other estimation procedures investigated, kriging provides a measure of the error and associated confidence in the estimates. Cokriging is similar to kriging except it uses two correlated measured values. The more intensely sampled data is used to assist in predicting the less sampled data. Cokriging is most effective when the covariates are highly correlated. Both kriging and cokriging assume homogeneity of first differences. While kriging is considered the best linear unbiased spatial predictor (BLUP), there are problems of nonstationarity in real-world data sets.

Visualization Techniques

After selecting the combination of direct data visualization, gridding and interpolation methods to create the geometry and associated data, many different visualization techniques are available to visualize the model. With so many choices available for gridding, interpolation, and rendering of each object in the total model, there are virtually limitless possibilities for the final visualization.

Rendering Methods

Rendering methods determine the appearance of objects in the view. The methods that are chosen have a profound effect on the final product. The large number of rendering options and subsetting techniques allows for the same data to be presented in many different ways. This ability to customize visualizations provides a mechanism to emphasize certain aspects or objects in a model.

Points and Lines

Virtually all visualizations contain lines and points. Whether the visualization specifically includes cells that are lines and/or points or has 2D and/or 3D cells (e.g. triangles and hexahedrons) which have nodes and edges, line and

point rendering must be considered. Points can be rendered as individual pixels or as square blocks of multiple pixels. Points can also be represented by various glyphs as discussed in section 1.1.3. Lines can be rendered solid, or with patterns like dotted or dashed in any width (measured in pixels). Lines can also be rendered as cylindrical tubes with user specified radius. Both points and lines can be colored according to data values or their colors can be set to any shade to outline the cells that they represent. Of course, it is always an option to not display the points and/or lines.

Surfaces

The choice of surface rendering technique has a dramatic impact on model visualizations. Figure 1.25 is a dramatization that incorporates many common surface-rendering modes. These include Gouraud Shading, Flat Shading, Solid Contours, Transparency and Background Shading. In this figure, a plume is represented in each geologic layer of this model. The geologic layers are exploded and a unique rendering mode is used for each layer. This allows demonstrating five different surface rendering techniques. Section 1.2.5 included some discussion on surface rendering techniques. In the model, a very fine grid (in the x-y plane) was used and the flat shaded plume looks similar to the Gouraud shaded one. The solid contoured plume provides sharp color discontinuities at specific plume levels, however it provides no information about the variation of values within each interval.

The transparent plume was Gouraud shaded. Transparency could be applied to any of the surface rendering techniques except background shading. Transparency provides a means to see features or objects inside of the plume while still providing the basic shape of the plume. Objects inside a colored transparent object will have altered colors and the colors of the transparent object are affected by the color of the background and any other objects inside or behind the plume.

Background shading is a rather different approach. Each cell of the plume is colored the same color as the background. This makes the cell invisible, however the cell is still opaque. Objects that are behind the background shaded cells are not visible. In this example, the cell outlines are shown as lines colored by the concentration values. Background shading of the surfaces provides a "hidden line" rendering where the cells behind are not shown.

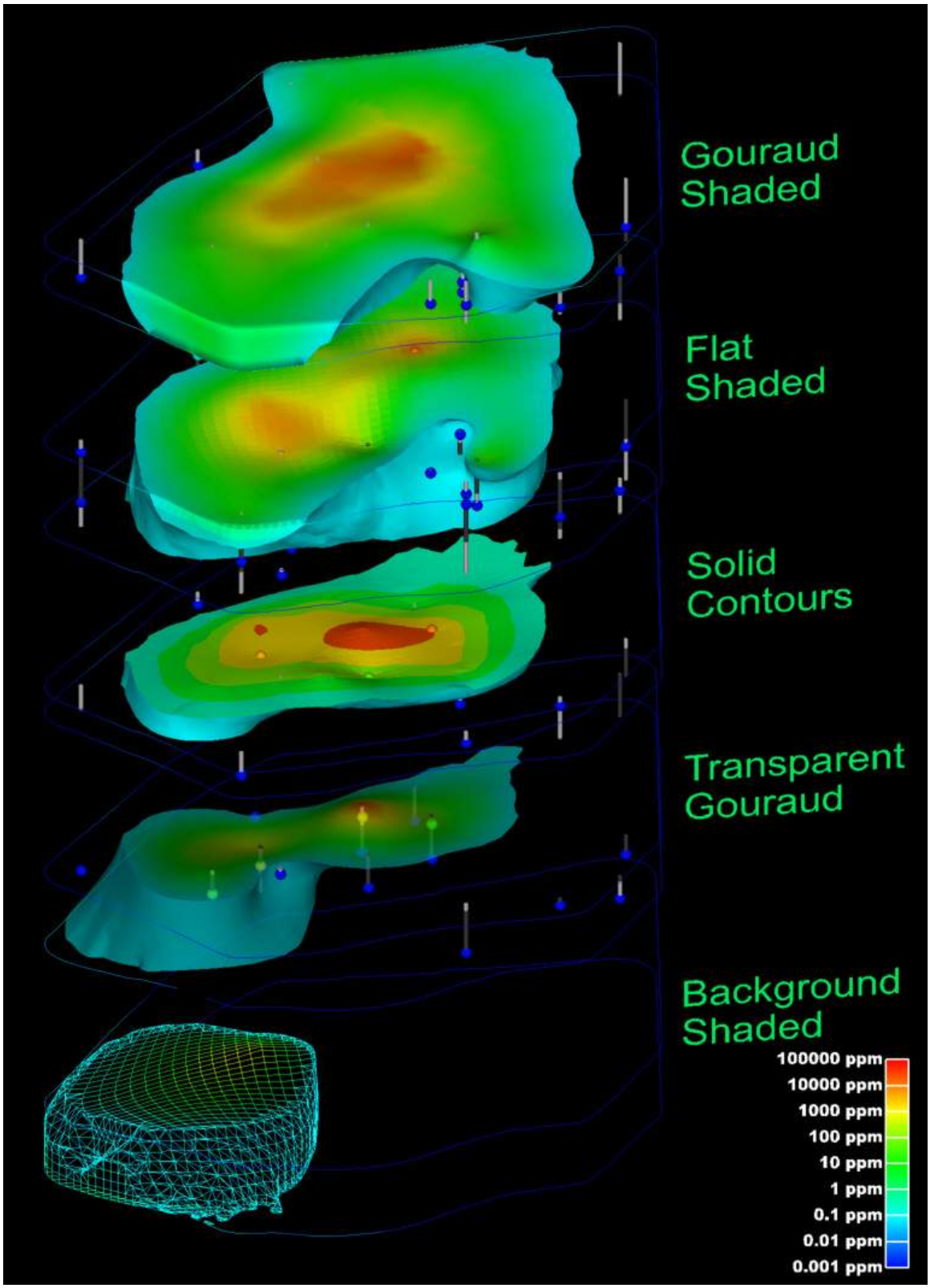


Figure 0.24 plume_shell Showing Various Shading Methods

An example of the rendering mode called "no lighting" has not been included in this paper. This technique renders cells as a single color (similar to flat shading), but with no lighting or shading effects. This eliminates all three-dimensional clues about the surface and usually produces an undesirable affect.

Texture mapping is a process of projecting a raster image onto one or more surfaces. The images should be geo-referenced (see section 1.1.1.5) to ensure that the image's features are placed in the correct spatial location. In Figure 1.26, a chlorinated hydrocarbon contaminant plume is shown at an industrial facility on the coast. Sand and rock geologic layers are displayed below the ocean layer. A color aerial photograph of the actual site was used to texture map and render the geologic layer that represents the ocean and was also applied to the three-dimensional representations of the site buildings as well as the ground surface.

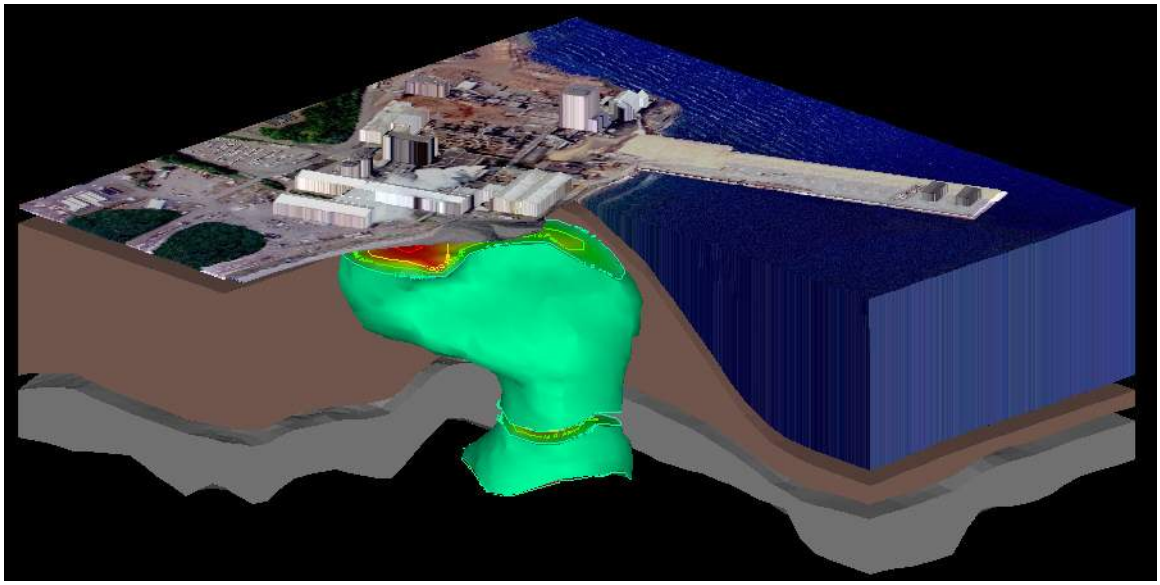


Figure 0.25 Coast Facility Showing Contaminant Plume, Geology with Texture Mapping

Volume Rendering

There are several volume rendering techniques. In Figures 1.27 and 1.28, two different methods are shown. For both methods, the data must be represented as a rectilinear grid (see section 1.2.1) and the data range is partitioned into two linear regions. The first region begins at the data minimum value and ends at a user-specified value (break point) somewhere between the minimum and maximum. The second region starts at the break point and continues to the data maximum. The opacity and color at each end of these regions is specified. This provides the parameters for mapping the data to colors and associated opacity.

The first method uses a ray tracing (a.k.a. ray casting or front-to-back) technique. Using ray tracing, the rectilinear grid is considered behind the projection plane. A ray is projected from each point in the projection plane

through the volume. The ray accumulates the opacity and color of each cell it passes through.

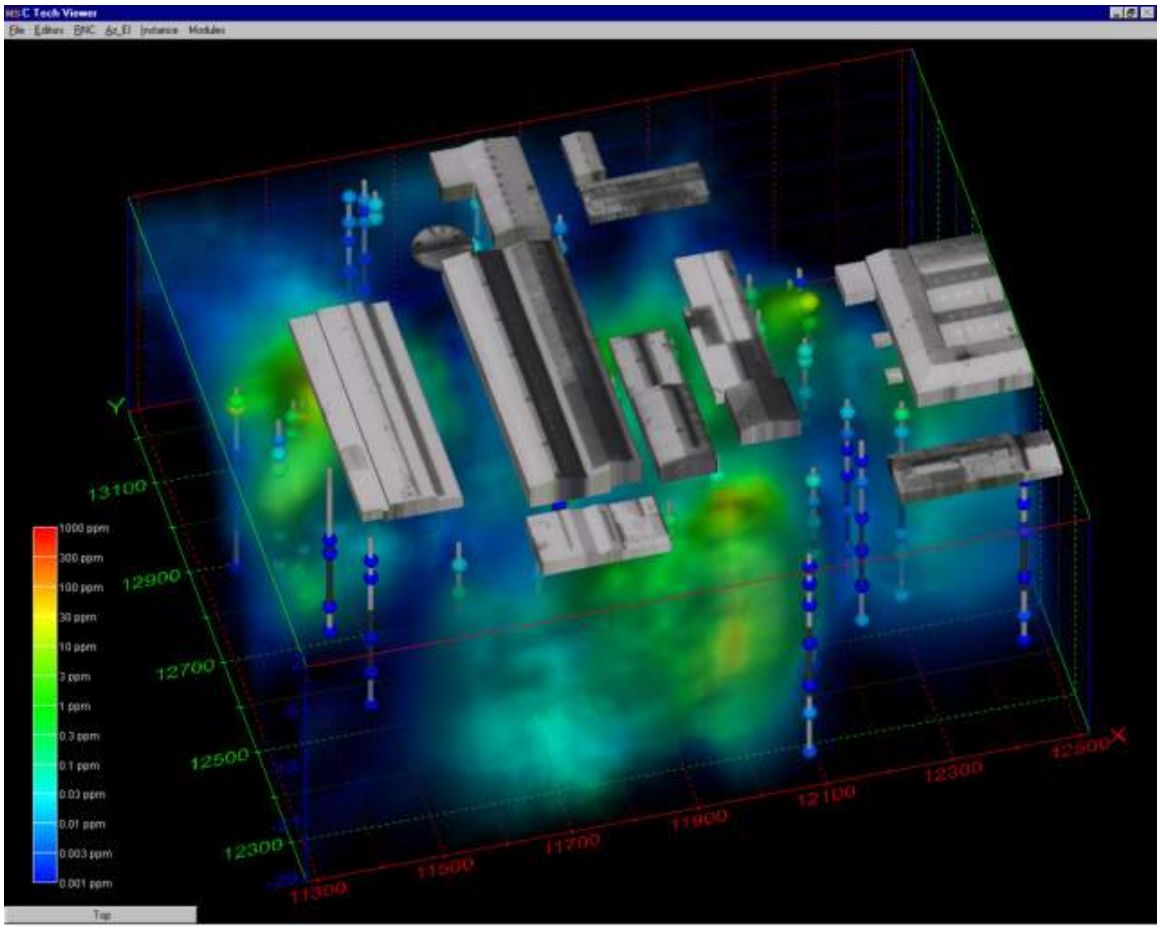


Figure 0.26 Ray Traced Volume Rendering

The other volume rendering technique is often referred to a Back-to-Front (BTF). This technique is supported in many OpenGL graphics cards. The hardware acceleration can make the rendering using BTF over 100 times faster than ray tracing. With back-to-front, the projection plane is behind the grid and the grid is represented as a series of slices. The slice direction through the volumetric grid is chosen to keep the slices as orthogonal as possible to the viewing direction. Each slice of the volume is projected on the projection plane, from the farthest plane to the nearest plane. The slices are colored and made transparent based on the properties described above.

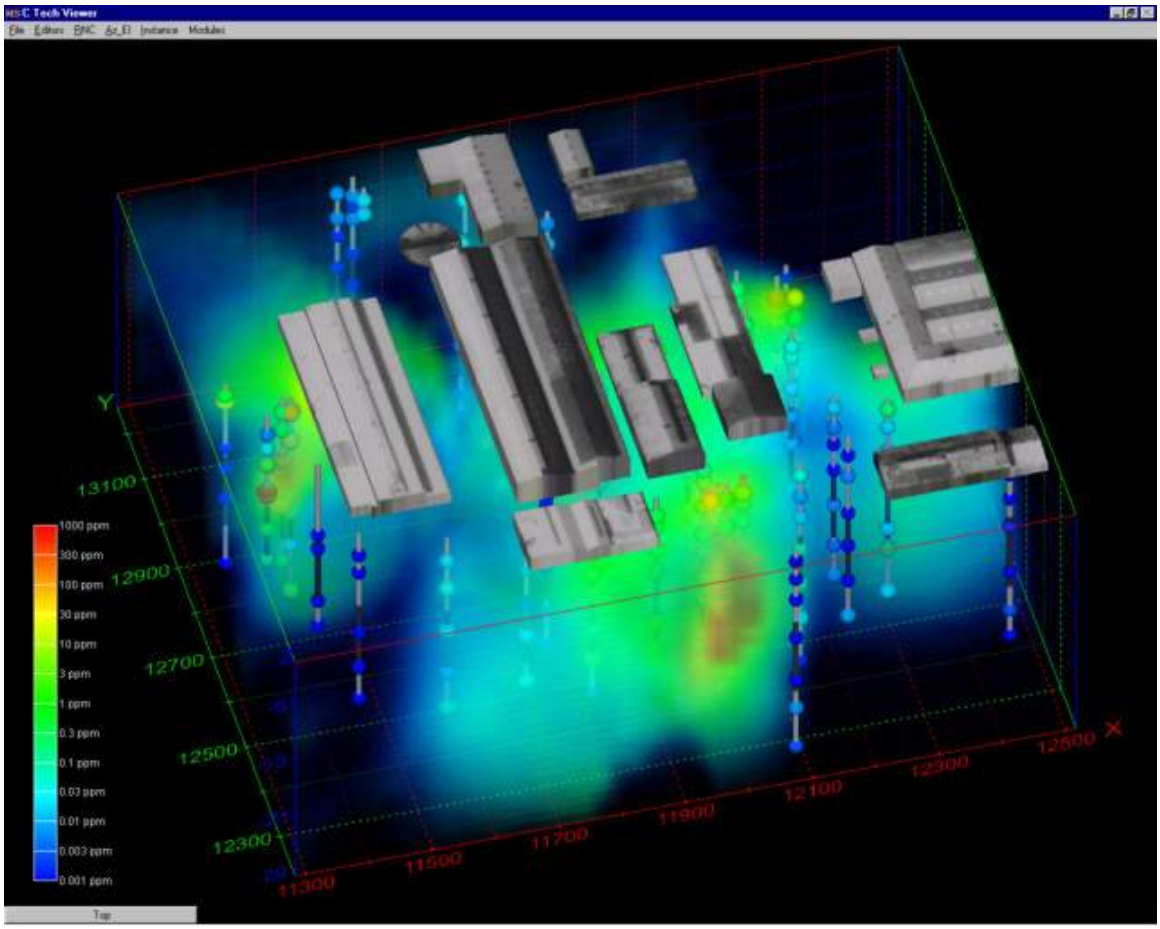


Figure 0.27 Back-to-Front Volume Rendering

Ray Tracing

Ray tracing is a very computationally intensive form of rendering. It is not uncommon that rendering individual images at modest resolution (640 by 480 pixels) may take 30 minutes or more. What ray tracing lacks in speed it makes up for in fidelity and detail. Ray tracing can display details like shadows and reflections (such an entire scene reflected in a shiny spherical object). Ray tracing is often used to produce photo-realistic scenes for architectural and landscape renderings. However, another consequence of photo-realism is that a substantial amount of information such as surface textures, reflectivity, specular reflection coefficients, etc., must be supplied for all objects in the view. Because of the complexity and very slow speed, ray tracing is not commonly used for environmental visualization.

Color

The choice of color(s) to be used in a visualization affects the scientific utility of the visualization and has a large psychological impact on the audience. Throughout this paper, a consistent color scale (a.k.a. datamap) has been used. This color scale associates low data values with the color blue and high data values with the color red. Values between the data minimum and maximum are mapped to hues that transition from red to yellow to green to

cyan (light blue) to blue. People are accustomed to interpreting blue as a "cold" color and red as a "hot" color. For this reason, lay persons more easily understand this color spectrum. It also provides a reasonably high degree of color fidelity, allowing discrimination of small changes in data values.

However, many times color scales with vivid colors like red are deemed too alarming. Since there is not a universally (or even scientifically) accepted standard for color spectrums used for data presentation, the use of softer shades of color and the elimination of red or other garish colors from the spectrum cannot be challenged on a scientific or legal basis. The consequence of this is the distinct possibility of two different visualizations that both communicate the same information with completely different colors. Often the choice of colors is made on aesthetic or political grounds, governed more by the party being represented and their role in the site than by scientific reasons.

Model Subsetting

Once the model of the site has been created, visually communicating the information about that site generally requires subsetting the model. Subsetting is a generic term used to convey the process of displaying only a portion of the information based on some criteria. The criteria could be "display all portions of the model with a y coordinate of 12,700. This would result in a slice at y = 12,700 through the model orthogonal to the y (or North) axis. As this slice passes through geologic layers and/or contaminated volumes, a cross-section of those objects would be visible on the slice. Without subsetting, only the exterior faces of the model will be visible.

When evaluating subsetting operations, the dimensionality of input and output should be considered. As an example, consider the slice described above. If a slice is passed through a volume, the output is a 2D planar surface. If that same slice passes through a surface, the result is a line. Slices reduce the dimensionality of the input by one. The sections below will discuss a few of the more common subsetting techniques.

Plume Visualization

Contaminant plume visualization employs one of the most frequently used subsetting operations. This is accomplished by taking the subset of all regions of a model where data values are above or below a threshold. This subset is also referred to as a volumetric subset and its threshold value as the subsetting level. When creating the objects that represent the plumes, two fundamentally different approaches can be employed. One approach creates one or more surfaces corresponding to all regions in the volume with data values exactly equal to the subsetting level and all portions of the external surfaces of the model where the data values exceed the subsetting level. This results in a closed but hollow representation of the plume. This method, which was used in Figure 1.26, has a dimensionality one less than the input dimensionality.

The other approach subsets the volumetric grid outputting all regions of the model (cells or portions thereof) that exceed the subsetting level. This

method has the same dimensionality output as input. The disadvantages of this approach are the need to compute and deal with the all interior volumetric cells and nodes. The advantages include the ability to perform additional subsetting and to compute volumetric or mass calculations on the subset volume.

Cutting and Slicing

Within C Tech's EVS software there is a significant distinction between the terms cut and slice. Slices create objects with dimensionality one less than the input dimensionality. If a volume is sliced the result is a plane. If a surface is sliced the result is one or more lines. If a line is sliced, one or more points are created. Figure 1.29 has three slice planes passing through a volume which has total hydrocarbon concentrations on a fine 3D grid. The horizontal slice plane is transparent and has isolines on ½ decade intervals.

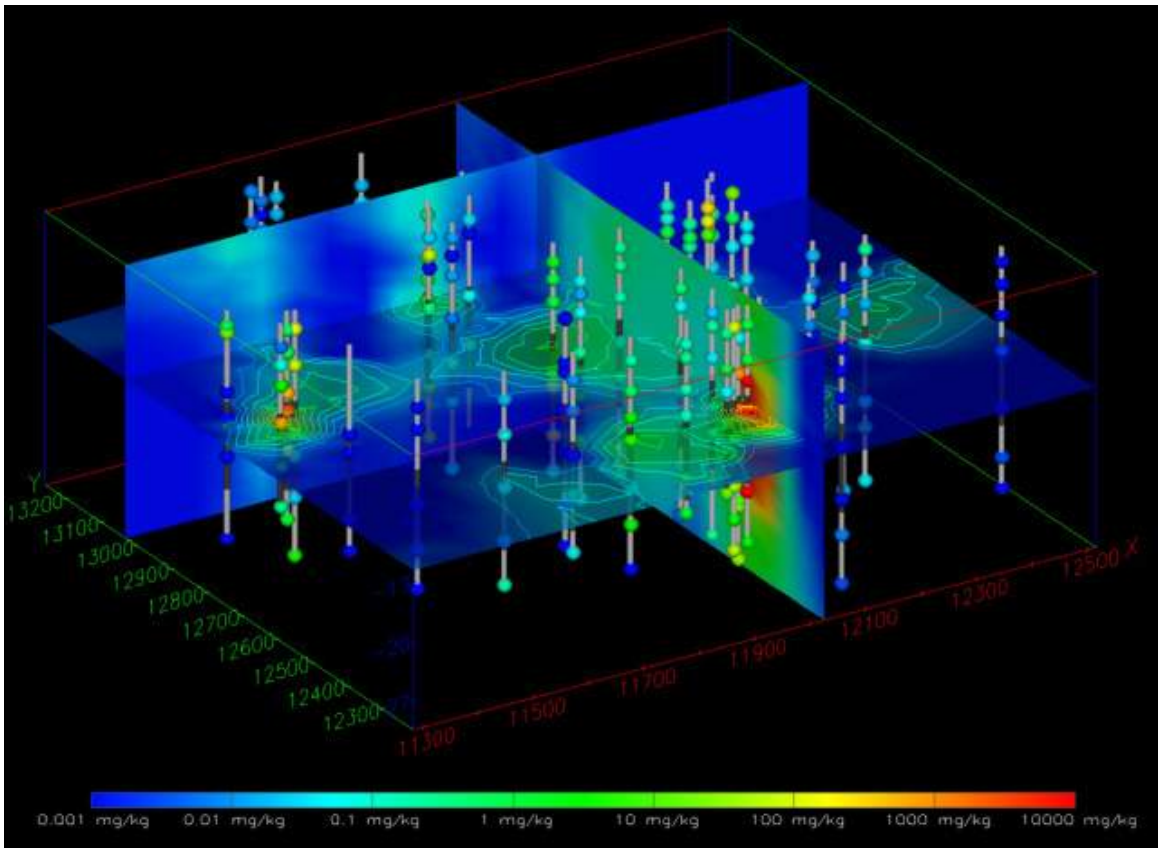


Figure 0.28 Three Slice Planes Passing Through a 3D Kriged Model

By comparison, cutting still uses a plane, but the dimensionality of input and output are the same. Cutting outputs all portions of the objects on one side of the cutting plane. If a volume is cut, a smaller volume is output. In Figure 1.30, the top half of the grid was cut away, but the plume at 1000 ppm is displayed in this portion of the volume. The lower half of the model also has labeled isolines on ½ decade intervals.

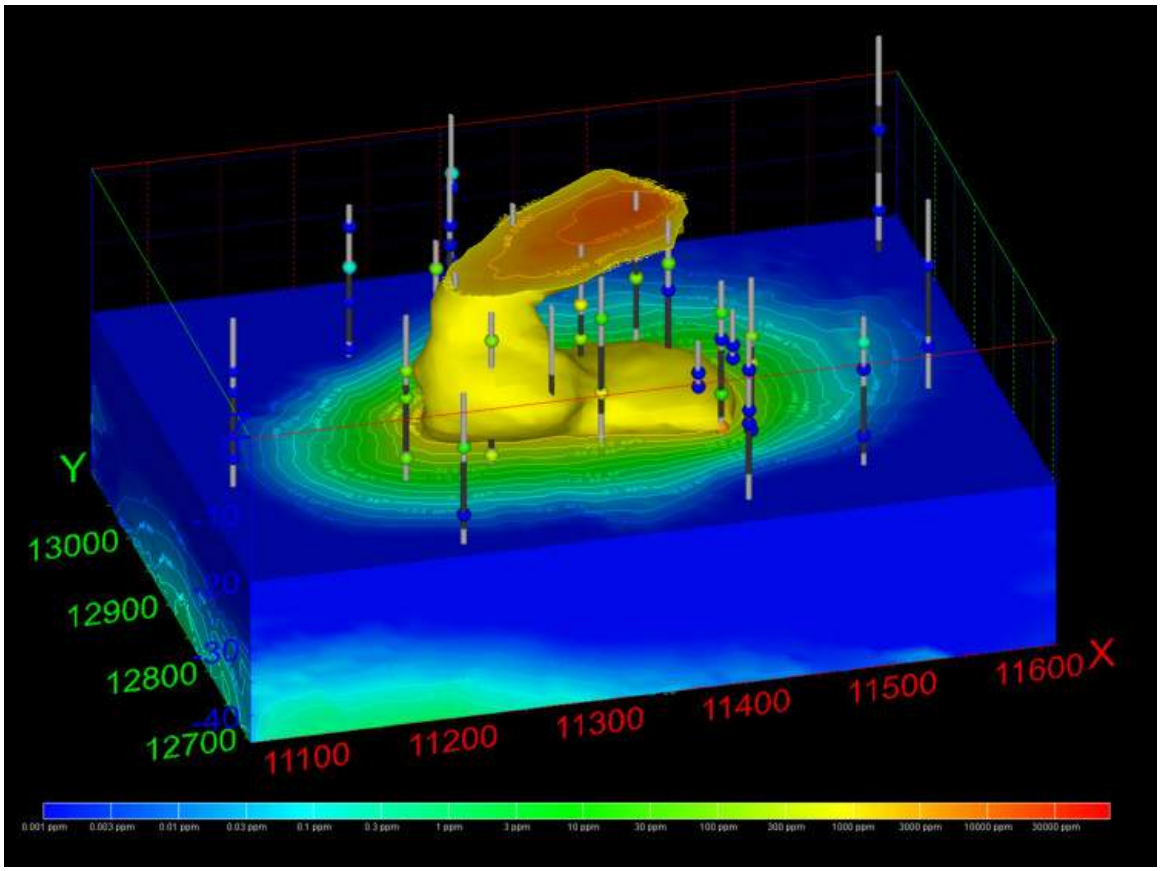


Figure 0.29 Cut 3D Kriged Model with Plume and Labeled Isolines

Isolines

Isolines (sometimes referred to as isopachs) have output dimensionalities that are one less than the input dimensionality. Surfaces with data result in isolines or contour lines that are paths of constant value on the surface(s). Isolines can be labeled or unlabeled. Various labeling techniques can be employed ranging from values placed beside the lines to labels that are incorporated into a break in the line path and mapped to the three-dimensional contours of the underlying surface. Examples of visualizations using isolines are shown in Figures 1.30 and 1.26.

Visualization Publishing

For the purposes of this paper, the term publishing refers to the creation of a visualization that can be delivered to an end-user and does not require that user to have the software that created the visual to view it. With this definition, we expand publishing to include formats used by word processing software, web publishing, animation and color printing. In today's world, even when the final use for a graphic is color printing (e.g. magazines, brochures, etc.), the image is usually delivered to the printer in an electronic format.

Raster Images

Raster images are the foundational format for visualization publishing. Raster images are the most common method of outputting visualization results.

Animations are constructed from a series of raster images and printing of digital graphics nearly always involves rasterization (conversion to a raster image) at some stage of the process.

Image Formats & Resolution

Raster images can be written in one of many different file formats. Common formats include Windows Bitmap (.bmp), TARGA (.tga), TIFF (.tif), JPEG (.jpg), and GIF (.gif). The critical issues to consider are color depth and whether the format uses lossy or lossless compression. For optimal quality, 24-bit true color should be used. The size of uncompressed true color raster images can be estimated in bytes as three times the product of the pixel resolution (width and height) plus ~30. For example, a 640x480 24-bit uncompressed image will be just under 922 Kbytes.

Windows Bitmap (.bmp) files do not use compression and support several color depths from 1 bit monochrome (black and white only) to 24-bit true color. File sizes tend to be large because of the lack of compression, but bmp files generally compress very well in archives (e.g. gzip, pkzip, tar, etc.). An additional advantage of Windows Bitmap files is that all Windows computers include software that can read .bmp files.

Portable Network Graphics (PNG), TARGA and TIFF formats both use lossless compression algorithms and support a wide range of color depths including 24-bit. Lossless compression does not compromise image quality. In contrast, lossy compression allows for modifying pixel colors in order to maximize compression. This usually results in the appearance of image noise. It is most pronounced near regions of high contrast such as the pixels adjacent to lines and edges. Lossless compression and uncompressed formats can be saved and converted any number of times without any degradation of image quality. PNG files also are quite small as they use a very appropriate compression for environmental data.

JPEG and GIF formats are both commonly used for graphics on the Internet. Their advantages are small file sizes and compatibility with virtually all computers and operating systems (provided they have web browsers or other image viewing software installed). Unfortunately, these are largely inappropriate for environmental data as they create compression artifacts which are highly undesirable.

Color Printing Issues

The following provides hints and tips for obtaining optimal quality when printing. This assumes you are using a color printer, but it is important to note that the user may print grayscale images with a black and white printer if desired. This would of course be best implemented by creating grayscale colormaps to eliminate ambiguities associated with different colors that have the same gray-scale representation.

Optimal printing of a raster image requires taking several factors into consideration. First, you must know the characteristics of the printer and the intended size of the printed image. Printers vary considerably and no single recommendation can be appropriate. Color printers fall into three primary

categories, inkjet, color laser, and dye sublimation. EVS, for example, produces raster images which are continuous tones with 256 shades each of red, green and blue for a total of 16.7 million possible colors (256 * 256 * 256). Color printers either produce continuous tones or approximate them using a pattern of primary colored pixels in an n-by-n grid.

Among these three printer categories there is considerable variation. Inkjet printers are generally capable of producing one of only eight primary colors for each printer pixel (or dot). These colors are white, black, cyan, magenta, yellow, red, green and blue. Inkjets must therefore use a grid of primary colored pixels to approximate continuous tones. The larger the grid (4 by 4 vs. 2 by 2) the better the color approximation. However, larger grids tend to create artifacts called jaggies that are visually undesirable. The challenge is to balance the need for smoother color rendition with the desire to have higher resolutions.

Dye sublimation printers are at the other end of the spectrum. Their ability to reproduce continuous tones makes the task of choosing a resolution easy. A typical dye-sub printer has a resolution of 300 dots per inch (dpi). If the intended size of the final printed image is 10 inches wide by 7 inches tall, then the optimal image size is 10*300 by 7*300 or 3000 x 2100 pixels. If quicker image creation and print times are desired, a compromise resolution would be exactly half or 1500 wide by 1050 high.

It is best to have an integer number of printer pixels for each "source" image pixel. When the image size is half of the printer pixel resolution, each source pixel gets a 2-by-2 grid. The n-by-n grid concept applies to all types of printers. This "rule" is actually a guideline for best results. Other resolutions (non-integer ratios) create banding artifacts that are usually objectionable.

For inkjet printers you should always allow for at least a 2x2 grid and usually 3x3 to 5x5 gives the best results. For an EPSON printer with 720x1440-dpi resolution you should use the smaller resolution number (720) for your calculations. The printer uses the additional resolution to better approximate the colors.

Example: For a printer with 720 dpi, to print an image 9 by 7.5 inches (landscape) we recommend that you start at a 4x4 grid which gives an effective printed resolution of 180 dpi. Your image width and height would therefore be:

Width = 9.0 * 180 = 9.0 * (720/4) = 1620

Height = 7.5 * 180 = 9.5 * (720/4) = 1350

Finally, color laser printers vary in their abilities to approximate continuous tones. This means that the rules to apply will be somewhere between dye-sub and inkjet properties.

Stereo Images

There are several different types of three dimensional viewing and printing technologies available. Anaglyphs which present the left and right eye images as red or blue monochromatic images are among the simplest technique, but do not allow the use of full color images. Shutter Goggles that switch the visibility of left and right eyes as in the IMAX Stereo Theaters and Virtual

Reality headsets allow for each eye to view a different image rendered by the computer. These methods require special viewer software and hardware to differentiate the left- and right-eye views. Our binocular vision enables us to view 3D objects because each eye sees a slightly different image of the scene from a slightly different viewpoint. In the brain, these two images are combined into a single stereoscopic perception, having the appearance of depth and volume.

Lenticular Images utilize a lenticular plastic sheet that has vertical ridges that form narrow cylindrical lenses. These lenses display a slightly different view to each eye depending on the angle between the eye and the lenticular sheet. The resulting lenticular images are autostereoscopic, meaning that they require no special viewers to display 3D imagery. A significant limitation of lenticular displays is that they display images with an apparent resolution that is governed by the width of the cylindrical lenses. However, the three-dimensional lenticular printouts have all the visual cues of an actual volume of space, including object positioning (foreground and background), parallax and shading, and 'look-around'. The hologram-like effect is quite remarkable, and the image can have a tremendous impact on the viewer. 3D COM, Inc. provides a commercial service printing stereo image sequences as lenticular printouts. Their URL is http://3dhardcopy.com/.

An alternative stereo inkjet printing technology that creates a high resolution, full color stereoscopic hard copy is called StereoJet (Scarpetti 1998 & 1996). In the StereoJet image the left and right views are printed onto opposite surfaces of a multi-layer sheet, using inks with polarizing properties. The polarizing axes of the two image-receiving layers are oriented at 90º to one another. By wearing polarized glasses viewers perceive the composite stereoscopic image in full depth. Each eye sees only the assigned image, and the brain processes the information to provide full binocular stereoscopic perception. Lenticular 3D prints do not have the resolution or clarity produced by this technology, however StereoJet prints require special glasses. StereoJet printing is quite expensive at about $300 per sq. ft., but the quality is very high. For additional technical information or a source for StereoJet printing, see http://www.rowland.org/stereo/ and http://www.slidefactory.com/SJmain.html.

Postscript

Postscript, developed by Adobe, is a relatively common printer language that can represent both raster images and vector (polygon) based representations of models. The advantage of vector representations is the elimination of jaggies and other objectionable results in printed output when the raster image being printed is too low resolution. The disadvantage of Postscript's vector format is its limited accommodation of rendering modes and shading methods.

VRML Models

Virtual Reality Modeling Language (VRML) is the only common non-raster output format that successfully captures most capabilities and features

common in environmental visualizations. VRML is an ASCII file format that provides a vector description of three-dimensional models. It includes information about grids, data, subsetting, rendering properties (e.g. transparency) and raster images that are used as texture maps. Generally, models output as VRML look nearly identical to the rendered model in modeling and visualization software like EVS.

Capabilities and Limitations

VRML models provide a great degree of functionality and support nearly all rendering modes and visualization techniques. Ray tracing and all types of volume rendering are not currently supported in VRML.

Customization

VRML models can be customized by hand editing or the use of VRML authoring and/or editing software. These packages provide a mechanism to add sounds, interactivity, and functionality to what are normally static VRML models.

VRML Viewer Browser Plug-ins

Viewing of VRML files is possible in most web browsers like Internet Explorer and Netscape Navigator. The processing and viewing of VRML models requires a browser plug-in or dedicated VRML viewer.

VRML is also used as a generic format for the printing of 3D models (see www.zprint.com)

Animations

Animations provide one of the most powerful means for communicating environmental data. They offer all of the advantages of images and many of the advantages of VRML models. Processing or playing back a sequence of individual raster images creates animations. Animations can be used to clarify three-dimensionality by rotating a model about one or more axes, however they can do much more. Animations can also include moving objects, changing plume levels, moving cut or slice planes. They can even be used to present temporal data such as measured water table variation over time. In fact all of the above techniques can be combined into a single animation.

Although animations cannot be manipulated by the user as a VRML model can, animations can incorporate complex model changes that are beyond the capabilities of VRML. Furthermore, VRML's flexibility can sometimes be a problem. The end user can become confused or can miss the point of the model since VRML viewers allow the user to manipulate the model without limitation. Animations follow a script. If the script is properly planned and executed, animations lead the viewer down a path communicating the proper message. Animations are not limited to rendered images. Title images can be used to introduce sequences and sound can be added. The addition of sound allows for background music for dramatic affect as well as narration.

Creation Process

The process of creating animations begins with the creation of the image sequences. In C Tech's EVS, a script is generated based on a series of key

frames (representing transition points in the animation) and interframe interpolation method. When the script is played, the software renders each image and writes them as a raster image. Conversion software is then used to encode the images into an animation file format.

Formats

There are many animation file formats available. The standard format for Windows computers is Audio-Video Interleaved (AVI). AVI files can employ many different Codecs (compression / decompression methods). Virtually all Codecs utilized by AVI converters employ lossy compression. Therefore if an individual frame of an animation were viewed or extracted, it would be noticeably degraded from the original. As with image file formats the compression artifacts are most pronounced in areas of high contrast such as lines and edges. Photographs of people and landscapes do not tend to have these high contrast regions (as compared with digitally created images) and therefore the compression is usually more acceptable. MPEG (developed by the Motion Picture Engineering Group) and Apple's QuickTime are competing animation formats. These also generally employ lossy compression.

A noticeable exception in animation file formats is the proprietary HAV file format developed by Gromada (http://www.gromada.com). HAV uses a lossless compression format that often creates files as small or smaller than lossy codecs. Gromada offers shareware converter software and freeware HAV player. Their player is also capable of playing image sequences without the need to convert them to an animation format. C Tech's EVS software includes an animation converter that creates AVI, MPEG and HAV files.

The creation of animations intended for playback on televisions requires consideration. The U.S. and foreign video formats (NTSC, PAL and SECAM) all use Interleaved Video fields. The first color TV broadcast system was implemented in the United States in 1953. This was based on the NTSC (**N**ational **T**elevision **S**ystem **C**ommittee) standard. Many countries on the American continent as well as many Asian countries including Japan use NTSC. NTSC runs on 525 lines/frame. The PAL (Phase Alternating Line) standard was introduced in the early 1960's and implemented in most European countries except for France. The PAL standard utilizes a wider channel bandwidth than NTSC that allows for better picture quality. PAL runs on 625 lines/frame. The SECAM (**S**equential **C**ouleur **A**vec **M**emoire or **S**equential **C**olor with **M**emory) standard was introduced in the early 1960's and implemented primarily in France. SECAM uses the same bandwidth as PAL but transmits the color information sequentially. SECAM runs on 625 lines/frame.

Interleaved video displays alternating odd and even scan lines every 60th of a second (50 Hz. for some PAL and all SECAM formats). These results in even and odd scan lines flickering at 30 Hz. When lines or edges are drawn only one pixel wide and are nearly horizontal, the lines will exist in only the odd or even fields. This causes the lines to flicker in a very objectionable manner. The best solution to this problem is to employ special processing techniques on the images before conversion to broadcast video. The processing reduces the contrast of lines and blurs them slightly to avoid having the entire line

being represented in only the odd or even fields. Although this may seem like it would introduce additional negative consequences, the results are usually excellent.

Visualization Applications

There are many applications for visualization of environmental data. Visualization enables technical personnel to better understand their work. It is often the only practical method of communicating complex results to a non-technical audience such as public hearings or litigation support. In these environments, animations are particularly useful and often employed.

Though GIS software often includes some visualization capability, it often falls short as compared with true 3D visualization software. However GIS software has the ability to store images and animations created with other software in their databases. These images or animations can be viewed by selecting them on a site-by-site basis. In this manner, GIS software can become the repository for detailed visualizations performed on small sites that are a portion of a much larger project. With GIS systems like ESRI's ArcView, images and animations can be hot-linked to objects in the GIS project allowing them to be viewed with a single mouse click.

References

1. Environmental Visualization System and EVS are registered trademarks of C Tech Development Corporation.

2. ArcView and ARC/INFO are registered trademarks of Environmental Systems Research Institute (ESRI).

3. Scarpetti, J.J., P.M Dubois, R.M. Friedhoff, and V.K. Walworth, Full-color 3-D Prints and Transparencies, *J. Imaging Sci. Tech*. **42**, 307-310 (1998).

4. Scarpetti, J.J., Coating Methods and Compositions for Production of Digitized Stereoscopic Polarizing Images, International Patent W/O 96/23663 (1996).

Information Resources

All of the figures and graphics in this paper were created using C Tech Development Corporation's Environmental Visualization System (EVS). Much of the material in this paper is based on material in the EVS reference manuals and tutorials. (http://www.ctech.com)

Biography

Reed D. Copsey is the President and CEO of C Tech Development Corporation, a software company exclusively engaged in visualization, modeling and analysis of environmental, geologic, mining and archaeological data.

EVS Data Input & Output Formats

Input

EVS conducts most of its analysis using input data contained in a number of ASCII files. These files can generally be created using the C Tech Data

Exporter (free) application downloadable at www.ctech.com. The Exporter will create C Tech's formats from from Microsoft Excel and Access files.

- **Requirement for Consistent Coordinate Systems**
- **Projecting File Coordinates**
- **3D analyte (e.g. chemistry) (.apdv) Format**
- **3D Groundwater analyte (e.g. chemistry) (.aidv) Format**
- **analyte (e.g. chemistry) Time Files (.sct and .gwt) Format**
- **Handling Non-Detects**
- **Pre-Geology File Format**
- **Borehole (.geo) Geology Format**
- **Geology File Example: Sedimentary Layers and Lenses**
- **Geology File Example: Outcrop of Dipping Strata**
- **Geology Multi-File (.gmf) Format**
- **Geology Files for Fence Diagrams**
- **Time Control File (TCF)**
- **EVS Field File Formats (.eff, .efz & .efb)**

IMPORTANT ENHANCEMENTSTO DATA FORMATS:

Note: You have added the ability to read .apdv, .aidv and .pgf file to create a single geologic layer model. This was not done as a preferred alternative to creating/representing your valid site geology. However, most sites have some ground surface topography variation. If Krig_3D is used without geology input, the resulting output will have flat top and bottom surfaces. The flat top surface may be below or above the actual ground surface at various locations. This can result in plume volumes that are inaccurate.

When a .apdv or .pgf is read by Krig_3D_Geology or Spline_Geology the files are interpreted as geology as follows:

1. If ***Top*** of boring elevations are provided in the file, these values are used to create the ground surface.
2. If ***Top*** of boring elevations **are not** provided in the file, the elevations of the highest sample in each boring are used to create the ground surface.
3. The bottom surface is created as a flat surface slightly below the lowest sample in the file. The elevation of the surface is computed by taking the lowest sample and subtracting 5% of the total z-extent of the samples.

Output

Because EVS runs under all versions of Microsoft Windows operating systems, there are numerous options for creating output.

Bitmap: EVS renders objects in the Viewer in a user defined resolution. That resolution refers to the number of pixels in the horizontal and vertical directions.

Images: EVS also includes the Output_Images module, which will produce virtually all types of bitmap images supported by Windows. The most common types are .png; .bmp; .tga; .jpg; and .tif. PNG is the recommended format because it has high quality **lossless** compression.

Bitmap Animations: By using Output_Images with the Animator module, EVS-PRO and MVS can create bitmap animations. Once a sequence of images is created, the Images_to_Animation module is used to convert these to a bitmap animation format such as .AVI, .MPG, or a proprietary format called .HAV.

Printed Output: The Viewer provides the ability to directly output to any Windows printer at a user defined resolution. Alternatively, images may be created (as in **a)** above) and printed.

Vector: EVS offers several vector output options. These include:

VRML: MVS creates VRML files which are a vector output format that allows for creation of 3D modules that model can be zoomed, panned and rotated and can represent most of the objects in the C Tech Viewer. VRML files must be played in a VRML viewer or used for creating 3D PDFs or 3D printing.

4DIM: EVS-PRO and MVS create 4DIMs, which unlike bitmap (image) based animations contain a **complete 3D model** at each frame of the animation. Each frame can be thought of as a VRML model (*though it is not*) and has similar functionality. Each frame of the model can be zoomed, panned and rotated as a static 3D model **or** you can interact with the 4DIM animation as it is playing.

2D and 3D Shapefiles: Shapefiles that are compatible with ESRI's ArcGIS program can be created in full three-dimensions. Nearly any object in your applications can be output as a shapefile. The primary limitations are associated with the limitations of shapefile. The most significant limitation is the lack of any volumetric elements.

AutoCAD .DXF Files: AutoCAD compatible DXF files can be created in full three-dimensions. Nearly any object in your applications can be output as a DXF file.

Archive: EVS and MVS offer several output options for archiving kriged results and/or geologic models. The preferred format is C Tech's fully documented EFF or EFB formats. Both of these file types can be read back into EVS or MVS eliminating the need to recreate the models by kriging or re-gridding. This saves time and provides a means to archive the data upon which analysis or visualization was based.

EVS Data Input & Output Formats

Input

EVS conducts most of its analysis using input data contained in a number of ASCII files. These files can generally be created using the C Tech Data

Exporter (free) application downloadable at www.ctech.com. The Exporter will create C Tech's formats from from Microsoft Excel and Access files.

- **Requirement for Consistent Coordinate Systems**
- **Projecting File Coordinates**
- **3D analyte (e.g. chemistry) (.apdv) Format**
- **3D Groundwater analyte (e.g. chemistry) (.aidv) Format**
- **analyte (e.g. chemistry) Time Files (.sct and .gwt) Format**
- **Handling Non-Detects**
- **Pre-Geology File Format**
- **Borehole (.geo) Geology Format**
- **Geology File Example: Sedimentary Layers and Lenses**
- **Geology File Example: Outcrop of Dipping Strata**
- **Geology Multi-File (.gmf) Format**
- **Geology Files for Fence Diagrams**
- **Time Control File (TCF)**
- **EVS Field File Formats (.eff, .efz & .efb)**

IMPORTANT ENHANCEMENTSTO DATA FORMATS:

Note: You have added the ability to read .apdv, .aidv and .pgf file to create a single geologic layer model. This was not done as a preferred alternative to creating/representing your valid site geology. However, most sites have some ground surface topography variation. If Krig_3D is used without geology input, the resulting output will have flat top and bottom surfaces. The flat top surface may be below or above the actual ground surface at various locations. This can result in plume volumes that are inaccurate.

When a .apdv or .pgf is read by Krig_3D_Geology or Spline_Geology the files are interpreted as geology as follows:

1. If ***Top*** of boring elevations are provided in the file, these values are used to create the ground surface.
2. If ***Top*** of boring elevations **are not** provided in the file, the elevations of the highest sample in each boring are used to create the ground surface.
3. The bottom surface is created as a flat surface slightly below the lowest sample in the file. The elevation of the surface is computed by taking the lowest sample and subtracting 5% of the total z-extent of the samples.

Output

Because EVS runs under all versions of Microsoft Windows operating systems, there are numerous options for creating output.

Bitmap: EVS renders objects in the Viewer in a user defined resolution. That resolution refers to the number of pixels in the horizontal and vertical directions.

Images: EVS also includes the Output_Images module, which will produce virtually all types of bitmap images supported by Windows. The most common types are .png; .bmp; .tga; .jpg; and .tif. PNG is the recommended format because it has high quality **lossless** compression.

Bitmap Animations: By using Output_Images with the Animator module, EVS-PRO and MVS can create bitmap animations. Once a sequence of images is created, the Images_to_Animation module is used to convert these to a bitmap animation format such as .AVI, .MPG, or a proprietary format called .HAV.

Printed Output: The Viewer provides the ability to directly output to any Windows printer at a user defined resolution. Alternatively, images may be created (as in **a)** above) and printed.

Vector: EVS offers several vector output options. These include:

VRML: MVS creates VRML files which are a vector output format that allows for creation of 3D modules that model can be zoomed, panned and rotated and can represent most of the objects in the C Tech Viewer. VRML files must be played in a VRML viewer or used for creating 3D PDFs or 3D printing.

4DIM: EVS-PRO and MVS create 4DIMs, which unlike bitmap (image) based animations contain a **complete 3D model** at each frame of the animation. Each frame can be thought of as a VRML model (*though it is not*) and has similar functionality. Each frame of the model can be zoomed, panned and rotated as a static 3D model **or** you can interact with the 4DIM animation as it is playing.

2D and 3D Shapefiles: Shapefiles that are compatible with ESRI's ArcGIS program can be created in full three-dimensions. Nearly any object in your applications can be output as a shapefile. The primary limitations are associated with the limitations of shapefile. The most significant limitation is the lack of any volumetric elements.

AutoCAD .DXF Files: AutoCAD compatible DXF files can be created in full three-dimensions. Nearly any object in your applications can be output as a DXF file.

Archive: EVS and MVS offer several output options for archiving kriged results and/or geologic models. The preferred format is C Tech's fully documented EFF or EFB formats. Both of these file types can be read back into EVS or MVS eliminating the need to recreate the models by kriging or re-gridding. This saves time and provides a means to archive the data upon which analysis or visualization was based.

Consistent Coordinate Systems

C Tech's software is designed to work with many types of data. However, because you are creating objects in a three-dimensional domain (x, y, and z extents) you must have all objects defined in a consistent coordinate system. Any coordinate projection may be used, but it is essential that all of your

data files (including world files to georeference images) be in the same coordinate system.

Furthermore, if volumes are to be calculated the units for all three axes (x, y, and z) must be the same. We strongly recommend working in feet or meters. Other units may be used (even microns!), but you may have to perform your own unit conversions when computing volumes with volumetrics.

Though all of your analysis must be performed in a consistent coordinate system, we do allow you to have data files with different units. If you choose to do this you must use the **reprojection** capabilities of the Projecting File Coordinates options in your data files.

Handling Non-Detects

It is important to understand how to properly handle samples that are classified as non-detects. A non-detect is an analytical sample where the concentration is deemed to be lower than could be detected using the method employed by the laboratory.

Non-detects should never be left out of the data file. They are critically important in determining the spatial extent of the contamination. Furthermore, it is important to understand what it means to have a sample that is *not-detected*. It is not the same as truly ZERO, or perfectly clean. In some cases samples may be non-detects but the detection limit may be so high that the sample should not be used in your data file.

As for WHY to use a fraction of the detection limit. At each point where a measurement was made and the result was a non-detect, we should use a fraction of the detection limit (such as one-tenth). If we were to use the detection limit, we would dramatically overestimate the actual concentrations. From a statistical point of view, when we have a non-detect on a site where the range of measurements varies over several orders of magnitude, it is far more probable that the actual measurement will be dramatically lower than the detection limit than just below it.

Non-detects are accommodated in EVS/MVS for analysis and visualization using a few **very** important parameters that should be well understood and carefully considered. These parameters control the clipping non-detect handling in all of the EVS modules that read chemistry (.apdv, or .aidv) files. The affected modules are Krig_3D, Krig_2D, Krig_Fence, post_samples, and file_statistics.

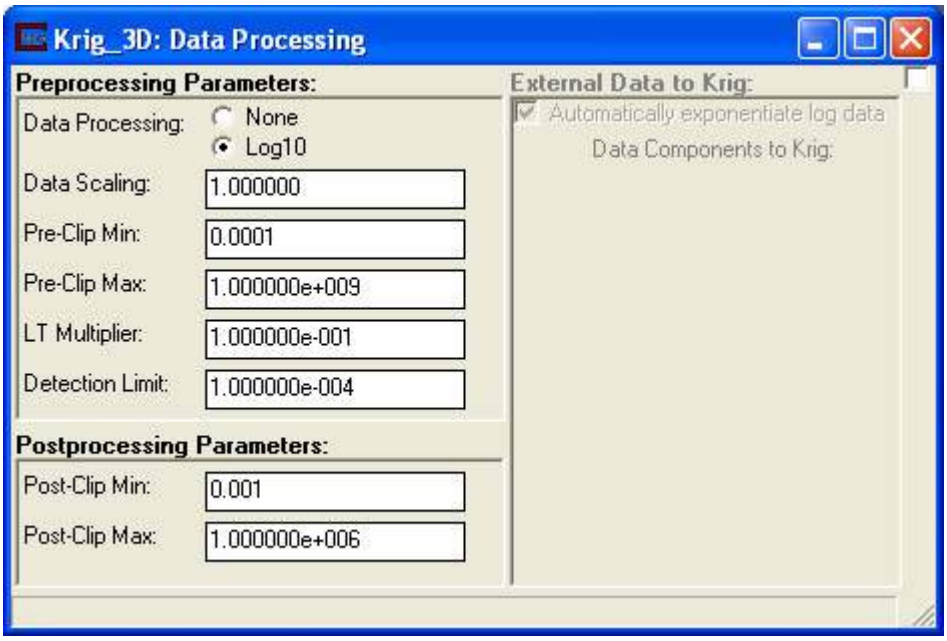
The parameters you need to understand (and perhaps modify) are:

- The number entered into the ***Pre-Clip Min*** input field will be used during preprocessing to replace any nodal property value that is less than the specified number. When log processing is being used, the value of Clip Min must be a positive, non-zero value. Generally, Clip Min should be set to a value that is one-half to one-tenth of the lowest detection limit in the data set, unless the user wishes to make the influence of not detected values stronger. As an example, if the lowest detection limit is 0.1 (which is present in the data set as a 0), and the user sets Clip Min to 0.0001, the clipped non-detected values forces

three orders of magnitude to be present between any detected value and the non-detected values.

- The ***Post-Clip Min*** parameter specifies the smallest nodal value that will be present in the data field output by Krig_3D. This parameter is useful for limiting or enhancing the effects of not detected values or outliers in a data set, and for optimizing the use of the dynamic color range used to represent the property distribution. Clip Min has a default value of 0.001, but can be set to any negative or positive value with magnitudes from -1.0 E09 to 1.0 E09. In general, good results are obtained by setting this value to the lowest property value or detection limit in the input data set. It is important to note that because not detected values are represented in the .apdv file as zeros, the Clip Min value can be used to strengthen the influence of small values or non-detects that are present in the data set. As an example, if the detection limit for a certain chemical analysis is 0.1 (which is entered into the .apdv file as 0), and the Clip Min is set to 0.0001, then this not detected value will have an effective influence on the kriged distribution near this data point that is three orders of magnitude stronger than the actual detection limit. The user should bear in mind that if mass or volume estimates are being made in the analysis, the Clip Min value could affect the estimates if considerable data are present that are being clipped. For volume estimates, if the specified lower bound of the concentrations of interest is well above the Clip Min value, then their will be no effects on the results. If the specified lower bound of concentrations for the volume estimate is lower than Clip Min, then the clipping will truncate the volume at the Clip Min value, and the estimated volume will be from the entire model domain.
- The ***LT Multiplier*** value affects any file value with a preceeding "<" character. It will multiply these values by the set value.
- The ***Detection Limit*** value affects any file values set with the "ND" or other non-detect flags (for a list of these flags open the help for the APDV file format). When the module encounters this flag in the file it will insert the a value equal to (Detection Limit * LT Multiplier).

The Data Processing panel for Krig_3D is shown below as an example.



Projecting File Coordinates

Discussion of File Coordinate Projection

Each file contains horizontal and vertical coordinates, which can be projected from one coordinate system to another given that the user knows which coordinates systems to project from and to. This is accomplished by adding the REPROJECT tag to the file. This tag is used in place of the coordinate unit definition and causes the file reader to look at the end of the file for a block of text describing the projection definitions. The definitions are a series of flags that listed below. **NOTE:** GMF files do not need the REPROJECT tag, the projection definitions can occur in a continuous block anywhere in the file.

Format (for REPROJECT flag):

APDV and AIDV files:

Line 2:Elevation/Depth Specifier:This line must contain the word*Elevation*or*Depth*(case insensitive)to denote whether sample elevations are true elevation or depth below ground surface. This should be followed by the ASCII string REPROJECT.

AN EXAMPLEFOLLOWS:

This is a comment line....not the header line - the next line is

X Y Z@@TOTHC Bore Top

Elevation 6.0 REPROJECT

PGF files:

- **Line 2**: Line 2 contains the declaration of Elevation or Depth, the definitions of Lithology IDs and Names, and coordinate units.

- **Elevation/Depth Specifier:** This line must contain the word *Elevation* or *Depth* (case insensitive) to specify whether well screen top and bottom elevations are true elevation or depth below ground surface.
    - **Depth** forces the otherwise optional ground surface elevation column to be required. Depths given in column 3 are distances below the ground surface elevation in the last column (column 6). If the top surface is omitted, a value of 0.0 will be assumed and a warning message will be printed to the EVS Status Window.
- **IDs and Names**: Line 2 should contain Lithology IDs and corresponding names for each material. Each Name is explicitly associated with its corresponding Lithology ID and the pairs are delimited by a pipe symbol "|".
    - Though it is generally advisable, IDs need not be sequential and may be any integer values. This allow for a unified set of Lithology IDs and Names to be applied to a large site where models create for sub-sites may not have all materials.
    - The number of (material) IDs and Names MUST be equal to the number of Lithology IDs specified in the data section. Each material ID present in the data section must have corresponding Lithology IDs and Names. If there are four materials represented in your .pgf file, there should be at least four IDs and Names on line two.
    - The order of Lithology IDs and Names will determine the order that they appear in Legends. The IDs do not need to be sequential.
    - You can specify additional IDs and Names, which are not in the data and those will appear on Legends.
- **Coordinate Units:** You should include the units of your coordinates (e.g. *feet* or *meters*). If this is included it must follow the names associated with each Lithology ID.
- The Btagmust follow the IDs & names forthematerials.

The first two lines of a PGF EXAMPLEFOLLOWS:

```
Pregeology file
Elevation 1|Silt 2|Fill 3|Clay 4|Sand 5|Gravel REPROJECT
```

GEO files:

Line 2: **Elevation/Depth Specifier:**

- The only REQUIRED item on this line in the Elevation or Depth Specifier.
    - This line should contain the word *Elevation* or *Depth* (case insensitive) to denote whether sample elevations are true elevation or depth below ground surface.
    - If set to Depth all surface descriptions for layer bottoms are entered as depths relative to the top surface. This is a common means of collecting sample coordinates for borings.
    - Note that the flags such as pinch or short are not modified.

- Line 2 SHOULD contain names for each geologic surface (and therefore the layers created by them).
    - There are some rules that must be observed.
    - The number of surface (layer) names MUST be equal to the number of surfaces. Therefore, if naming layers, the first name should correspond to the top surface and each subsequent name will refer to the surface that defines the bottom of that layer.
    - A name containing a space MUST be enclosed in quotation marks example ("Silty Sand"). Names should be limited to upper and lower case letters, numerals, hyphen "-" and underscore "_". The names defined on line two will appear as the cell set name in the Explode_and_Scale or select_cells modules. Names should be separated with spaces, commas or tabs.
- The REPROJECT tag must follow the names for the material numbers. It replaces the COORDINATE UNITS

AN EXAMPLE FOLLOWS:

X Y TOP BOT_1 BOT_2 BOT_3 BOT_4 BOT_5 BOT_6 BOT_7 Boring

-1 Top Fill SiltySand Clay Sand Silt Sand GravelREPROJECT

GMF files:

GMF files can have the projection block placed anywhere in the file.

Projection Block Flags:

**NOTE: Most flags defined below include arguments denoted by the '[' and ']' characters. These characters should not be included in the file. (Example: IN_XY meters)

PROJECTION: Indicates the start of the coordinate projection block

SWAP_XY:This will swap all coordinates in the x and y columns

UNITS[*string*]: This defines what your final coordinates for x, y, and z,will be.These units will be checked for in the file \data\special\unit_conversions.txt. If they are not found there they will be treated asequivalent tometers.

UNIT_SCALE[*double*]: The UNIT_SCALE flag sets the conversion factor between the final coordinates and meters. This is only necessary if you are defining units with the UNITS flagthat are not listed in the \data\special\unit_conversions.txt file.

IN_Z[*string*]: This flag sets what units your z or depth coordinates are. These units if different than the defined UNITS will be converted to the UNIT type. If UNITS arenot set then this will generate an error.

IN_X[*string*]: This flag sets whatunits your x coordinates are. These units if different than the defined UNITS will be converted to the UNIT type. If UNITS arenot set then this will generate an error.

IN_Y[*string*]: This flag sets whatunits your y coordinates are. These units if different than the defined UNITS will be converted to the UNIT type. If UNITS arenot set then this will generate an error.

IN_XY[*string*]: This flag sets what units your x and y coordinates are. These units if different than the defined UNITS will be converted to the UNIT type. If UNITS arenot set then this will generate an error.

PROJECT_FROM_ID[*int*]: This flag sets the EPSG ID value you wish to project from, you can look up what ID is appropriate for your location using the project_fieldmodule. To use this flag you must set the PROJECT_TO_ID or PROJECT_TO flag as well.

PROJECT_TO_ID[*int*]: This flag sets the EPSG ID value you wish to project to, you can look up what ID is appropriate for your location using theproject_field module. To use this flag you must set the PROJECT_FROM_ID or PROJECT_FROM flag as well.

PROJECT_FROM[*string*]: This flag sets the NAME of the location you wish to project from, you can look up what NAME is appropriate for your location using theproject_field module. To use this flag you must set the PROJECT_TO_ID or PROJECT_TO flag as well.**IMPORTANT**: The full name should be enclosed in quotation marks so that the full name will be read.

PROJECT_TO[*string*]: This flag sets the NAME of the location you wish to project to, you can look up what NAME is appropriate for your location using theproject_field module. To use thisflag you must set the PROJECT_FROM_ID or PROJECT_FROM flag as well.**IMPORTANT**: The full name should be enclosed in quotation marks so that the full name will be read.

TRANSLATE[*doubledoubledouble*]: This flag will translate each coordinate in the file by these values. It will translate x by the first value, y by the second, and all z values by the third.

END_PROJECTION: Denotes the end of the projection block and is required.

Example 1:

PROJECTION
PROJECT_FROM_ID 4267
PROJECT_TO "NAD83 / UTM zone 10N"
UNITS "meters"
SWAP_XY

END_PROJECTION

Example 2:
PROJECTION
UNITS "meters"
IN_XY "km"
IN_Z "ft"
END_PROJECTION

APDV: Analyte Point Data File Format

Discussion of analyte (e.g. chemistry) or Property Files

Analyte (e.g. chemistry) or property files contain horizontal and vertical coordinates, which describe the 3-D locations and values of properties of a system. For simplicity, these files will generally be referred to in this manual as analyte (e.g. chemistry) files, although they can actually contain any scalar property value of interest. Analyte (e.g. chemistry) files must be in ASCII format and can be delimited by commas, spaces, or tabs. They must have a .apdv suffix to be selected in the file browsers of EVS modules .The content and format of analyte (e.g. chemistry) files are the same, except that fence diagram files require some special subsetting and ordering. Each line of the analyte (e.g. chemistry) file contains the coordinate data for one sampling location and any number of (columns of) analyte (e.g. chemistry) or property values. There are no computational restrictions on the number of borings and/or samples that can be included in a analyte (e.g. chemistry) file, except that run times for execution of kriging do increase with the number of samples in the file.

Analyte (e.g. chemistry) data can be visualized independently or within a domain bounded by a geologic system. When a geologic domain is utilized for a 3-D visualization, a consistent coordinate system must be used in both the analyte (e.g. chemistry) and geology files. The boring and sample locations in 3-D analyte (e.g. chemistry) files do not have to correspond to those in the geology files, except that they must be contained within the spatial domain of the geology, or they will not be displayed in the visualization. If the posting of borings and sample locations are to honor the topography of a site, the analyte (e.g. chemistry) files also must contain the top surface elevation of the boring. As will be described in later sections, EVS uses tubes to show actual boring locations and depths, and spheres to show actual sample locations in three-space. In order for these entities to be correctly positioned in relation to a variable topography, the top elevation of the boring must be supplied to the program.

Format:

You may insert comment lines in C Tech analyte (e.g. chemistry) (.apdv) input files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

Line 1: You may include any header message here (that does not start with a '#' character) ***unless*** you wish to include analyte names for use by other EVS modules

(e.g. data component name). The format for line 1 to enable chemical names is as follows

A. Placing a pair of '@' symbols triggers the use and display of chemical names (example @@VOC). Any characters up to the @@ characters are ignored, and only the first analyte name needs @@, after that the chemical names must be delimited by spaces,

B. The following rules for commas are implemented to accommodate comma delimited files and also for using chemical names which have a comma within (example 1,1-DCA). Commas following a name will not become a part of the name, but a comma in the middle of a text string will be included in the name. The recommended approach is to put a space before the names.

C. If you want a space in your analyte name, you may use underscores and EVS will convert underscores to spaces (example: Vinyl_Chloride in a .aidv file will be converted to 'r;Vinyl Chloride." Or you may surround the entire name in quotation marks (example: "Vinyl Chloride").

The advantages of using chemical names (attribute names of any type) are the following:

- many modules use analyte names instead of data component numbers,
- when writing EVS Field files (.eff, .efb, etc.), you will get analyte names instead of data component numbers.
- when querying your data set with post_sample's mouse interactivity, the analyte name is displayed.
- time-series data can be used and the appropriate time-step can be displayed.

Line 2: **Specifications**

- **Elevation/Depth Specifier:** The first item on line 2 must be the word *Elevation* or *Depth* (case insensitive) to denote whether well screen top and bottom elevations are true elevation or depth below ground surface.
- **Coordinate Units:**After Depth/Elevation, include the units of your coordinates (e.g. *feet* or *meters*)

Line 3: **Specifications**

- The first integer (n) is the number of samples (rows of data) to follow.
- The second integer is the number of analyte (chemistry) values per sample.
- The units of each data analyte column (e.g. *ppm* or *mg/kg*).

Line 4: The first line of analyte point data must contain:

- X
- Y
- Elevation (or Depth) of sample
- (one or more) Analyte Value(s) (chemistry or property)

- Well or Boring name. The boring name cannot contain spaces (recommend underscore "_" instead).
- Elevation of the top of the boring.

  Boring name and top are are optional parameters, but are used by many modules and it is highly recommended that you include this information in your file if possible. They are used by post_samples for posting tubes along borehole traces and for generating tubes which start from the ground surface of the borehole. Both Krig_3D and Krig_3D_Geology will use this information to determing the Z spatial extent of your grids (Krig_3D_Geology will create a layer that begins at ground surface if this information is provided). Numbers and names can be separated by one comma and/or any number of spaces or tabs.

BLANK ENTRIES (CELLS) ARE NOT ALLOWED.

Please see the section on Handling Non-Detects for information on how to deal with samples whose concentration is below the detection limit. For any sample that is not detected you may enter any of the following. Please note that thefirst threeflag words are not case sensitive, but must be spelled exactly as shown below.

- A fraction of the actual detection limit for that sample (**best approach**). Normally the fraction should be 0.1 to 0.5 (10 to 50%).
- nondetect
- non-detect
- nd
- 0.0 (zero)

For files with multiple analytes such as the example below, if an analyte was not measured at a sample location, use any of the flags below to denote that this sample should be skipped for this analyte.Please note that these flag words are not case sensitive, but must be spelled exactly as shown below.

- missing
- unmeasured
- not-measured
- nm
- unknown
- unk
- na

Example Files are here:

APDV: Analyte Point Data File Format

Discussion of analyte (e.g. chemistry) or Property Files

Analyte (e.g. chemistry) or property files contain horizontal and vertical coordinates, which describe the 3-D locations and values of properties of a system. For simplicity, these files will generally be referred to in this manual as analyte (e.g. chemistry) files, although they can actually contain any scalar property value of interest. Analyte (e.g. chemistry) files must be in ASCII format and can be delimited by commas, spaces, or tabs. They must have a .apdv suffix to be selected in the file browsers of EVS modules .The content and format of analyte (e.g. chemistry) files are the same, except that fence diagram files require some special subsetting and ordering. Each line of the analyte (e.g. chemistry) file contains the coordinate data for one sampling location and any number of (columns of) analyte (e.g. chemistry) or property values. There are no computational restrictions on the number of borings and/or samples that can be included in a analyte (e.g. chemistry) file, except that run times for execution of kriging do increase with the number of samples in the file.

Analyte (e.g. chemistry) data can be visualized independently or within a domain bounded by a geologic system. When a geologic domain is utilized for a 3-D visualization, a consistent coordinate system must be used in both the analyte (e.g. chemistry) and geology files. The boring and sample locations in 3-D analyte (e.g. chemistry) files do not have to correspond to those in the geology files, except that they must be contained within the spatial domain of the geology, or they will not be displayed in the visualization. If the posting of borings and sample locations are to honor the topography of a site, the analyte (e.g. chemistry) files also must contain the top surface elevation of the boring. As will be described in later sections, EVS uses tubes to show actual boring locations and depths, and spheres to show actual sample locations in three-space. In order for these entities to be correctly positioned in relation to a variable topography, the top elevation of the boring must be supplied to the program.

Format:

You may insert comment lines in C Tech analyte (e.g. chemistry) (.apdv) input files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

Line 1: You may include any header message here (that does not start with a '#' character) ***unless*** you wish to include analyte names for use by other EVS modules (e.g. data component name). The format for line 1 to enable chemical names is as follows

A. Placing a pair of '@' symbols triggers the use and display of chemical names (example @@VOC). Any characters up to the @@ characters are ignored, and only the first analyte name needs @@, after that the chemical names must be delimited by spaces,

B. The following rules for commas are implemented to accommodate comma delimited files and also for using chemical names which have a comma within (example 1,1-DCA). Commas following a name will not become a part of the

name, but a comma in the middle of a text string will be included in the name. The recommended approach is to put a space before the names.

C. If you want a space in your analyte name, you may use underscores and EVS will convert underscores to spaces (example: Vinyl_Chloride in a .aidv file will be converted to 'r;Vinyl Chloride." Or you may surround the entire name in quotation marks (example: "Vinyl Chloride").

The advantages of using chemical names (attribute names of any type) are the following:

- many modules use analyte names instead of data component numbers,
- when writing EVS Field files (.eff, .efb, etc.), you will get analyte names instead of data component numbers.
- when querying your data set with post_sample's mouse interactivity, the analyte name is displayed.
- time-series data can be used and the appropriate time-step can be displayed.

Line 2: **Specifications**

- **Elevation/Depth Specifier:** The first item on line 2 must be the word *Elevation* or *Depth* (case insensitive) to denote whether well screen top and bottom elevations are true elevation or depth below ground surface.
- **Coordinate Units:**After Depth/Elevation, include the units of your coordinates (e.g. *feet* or *meters*)

Line 3: **Specifications**

- The first integer (n) is the number of samples (rows of data) to follow.
- The second integer is the number of analyte (chemistry) values per sample.
- The units of each data analyte column (e.g. *ppm* or *mg/kg*).

Line 4: The first line of analyte point data must contain:

- X
- Y
- Elevation (or Depth) of sample
- (one or more) Analyte Value(s) (chemistry or property)
- Well or Boring name. The boring name cannot contain spaces (recommend underscore "_" instead).
- Elevation of the top of the boring.

Boring name and top are are optional parameters, but are used by many modules and it is highly recommended that you include this information in your file if possible. They are used by post_samples for posting tubes along borehole traces and for generating tubes which start from the ground surface of the borehole. Both Krig_3D and Krig_3D_Geology will use this information to determing the Z spatial

extent of your grids (Krig_3D_Geology will create a layer that begins at ground surface if this information is provided). Numbers and names can be separated by one comma and/or any number of spaces or tabs.

BLANK ENTRIES (CELLS) ARE NOT ALLOWED.

Please see the section on Handling Non-Detects for information on how to deal with samples whose concentration is below the detection limit. For any sample that is not detected you may enter any of the following. Please note that thefirst threeflag words are not case sensitive, but must be spelled exactly as shown below.

- A fraction of the actual detection limit for that sample (**best approach**). Normally the fraction should be 0.1 to 0.5 (10 to 50%).
- nondetect
- non-detect
- nd
- 0.0 (zero)

For files with multiple analytes such as the example below, if an analyte was not measured at a sample location, use any of the flags below to denote that this sample should be skipped for this analyte.Please note that these flag words are not case sensitive, but must be spelled exactly as shown below.

- missing
- unmeasured
- not-measured
- nm
- unknown
- unk
- na

Example Files are here:

Three Dimensional Analyte Point Data File Example

An actual .apdv file could look like the following:

X	Y	ELEV	@@1-DCA	1-DCE	TCE	VC	SITE_I
Elevation feet							
50	4		mg/kg	ug/kg	ug/kg	mg/kg	
12008	12431	22.9	22	missing	500	nd	CSB-3
12008	12431	18.9	nd	nd	2800	nd	CSB-3
12008	12431	13.4	nd	nd	290	nd	CSB-3
12008	12431	8.4	nd	nd	9.7	nd	CSB-3

12008	12431	7.9	nd	nd	23	nd	CSB-3
12008	12431	1.9	nd	nd	24	nd	CSB-3
11651	13184	28.5	nd	nd	nd	nd	CSB-4
11651	13184	26	nd	nd	nd	nd	CSB-4
11427	12781	28.8	0.28	0.02	0.78	nd	CSB-4
11427	12781	24.8	nd	0.02	0.76	nd	CSB-4
11427	12781	17.3	nd	nd	0.01	nd	CSB-4
11427	12781	14.6	nd	nd	0.01	nd	CSB-4
11427	12781	9.8	nd	nd	nd	nd	CSB-4
11427	12781	3.3	0.64	0.14	1.5	0.19	CSB-4
11410	12725	29.6	0.01	nd	0.01	nd	CSB-4
11410	12725	23.6	0.08	nd	0.02	nd	CSB-4
11410	12725	21.6	0.04	nd	0.01	nd	CSB-4
11410	12725	12.1	0.1	nd	nd	0.13	CSB-4
11410	12725	6.1	0.06	nd	nd	0.05	CSB-4
11417	12819	28.2	0.01	nd	0.03	nd	CSB-4
11417	12819	24.2	0.04	nd	0.04	nd	CSB-4
11417	12819	16.2	0.43	0.04	0.04	nd	CSB-4
11417	12819	11.2	1.1	nd	nd	nd	CSB-4
11417	12819	9.2	nd	nd	nd	nd	CSB-4
11417	12819	6.2	nd	nd	nd	nd	CSB-4
11417	12819	2.2	0.06	nd	nd	nd	CSB-4
11402	12898	28.5	nd	nd	nd	nd	CSB-4
11402	12898	24.5	nd	nd	nd	nd	CSB-4
11402	12898	14.5	0.79	nd	1.7	nd	CSB-4
11402	12898	9	nd	nd	11	nd	CSB-4
11402	12898	2	0.18	nd	0.01	0.11	CSB-4
11260	12819	28.4	nd	nd	nd	nd	CSB-4
11260	12819	22.4	nd	nd	nd	nd	CSB-4
11260	12819	16.9	nd	nd	nd	nd	CSB-4
11260	12819	11.9	nd	nd	nd	nd	CSB-4
11260	12819	2.9	nd	nd	nd	nd	CSB-4
11340	12893	24.6	nd	nd	nd	nd	CSB-4
11340	12893	20.1	nd	nd	nd	nd	CSB-4

```
11340     12893     14.6     0.15          nd            nd          nd          CSB-4
11340     12893     9.1      nd            nd            nd          1.1         CSB-4
11340     12893     5.1      nd            nd            nd          nd          CSB-4
11249     12871     27.8     90            0.07          0.32        nd          CSB-4
11249     12871     23.3     0.16          nd            nd          nd          CSB-4
11249     12871     21.3     2.1           nd            nd          nd          CSB-4
11249     12871     13.3     nd            nd            nd          nd          CSB-4
11249     12871     8.3      nd            nd            nd          nd          CSB-4
11087     12831     28.3     nd            nd            0.01        nd          CSB-4
11087     12831     24.8     nd            nd            nd          nd          CSB-4
11087     12831     14.8     nd            nd            nd          nd          CSB-4
11087     12831     4.8      nd            nd            nd          nd          CSB-4
```

This file uses z coordinates (versus depth) for all samples, therefore line 2 has the word Elevation. There are 50 samples and 5 analytes (chemicals) per sample.

Another example using depths from the top surface is:

```
X Coord          Y Coord          Depth     @@TOTHC        Boring          Top
Depth            feet
37               1                          ppm
11856.72         12764.01         1         .057           CSB_67          1.7
11856.72         12764.01         8         .134           CSB_67          1.7
11856.72         12764.01         16        .081           CSB_67          1.7
11856.72         12764.01         20        .292           CSB_67          1.7
11856.72         12764.01         26        .066           CSB_67          1.7
11889.60         12772.20         2         1.762          CSB_23          1.3
11889.60         12772.20         4         .853           CSB_23          1.3
11889.60         12772.20         7         .941           CSB_23          1.3
11889.60         12772.20         15        10.467         CSB_23          1.3
11889.60         12772.20         16        488.460        CSB_23          1.3
11889.60         12772.20         22        410.900        CSB_23          1.3
11889.60         12772.20         26        .140           CSB_23          1.3
11939.19         12758.45         6         .175           CSB_70          3.7
11939.19         12758.45         15        .100           CSB_70          3.7
11939.19         12758.45         18        .430           CSB_70          3.7
```

```
11939.19        12758.45        26        .100          CSB_70        3.7
12002.80        12759.80        2         .321          CSB_24        1.2
12002.80        12759.80        4         .296          CSB_24        1.2
12002.80        12759.80        8         .179          CSB_24        1.2
12002.80        12759.80        13        0.000         CSB_24        1.2
12002.80        12759.80        17        .711          CSB_24        1.2
12002.80        12759.80        23        .864          CSB_24        1.2
12002.80        12759.80        28        .311          CSB_24        1.2
12085.15        12749.01        2         .104          CSW_71        4.6
12085.15        12749.01        6         .154          CSW_71        4.6
12085.15        12749.01        16        .732          CSW_71        4.6
12085.15        12749.01        26        .065          CSW_71        4.6
12146.70        12713.21        1         .027          CSB-72        2.1
12146.70        12713.21        7         .251          CSB-72        2.1
12146.70        12713.21        23        1.176         CSB-72        2.1
12199.70        12709.80        2         .043          CSB-12        6.0
12199.70        12709.80        4         .055          CSB-12        6.0
12199.70        12709.80        8         .031          CSB-12        6.0
12199.70        12709.80        12        .014          CSB-12        6.0
12199.70        12709.80        16        .018          CSB-12        6.0
12199.70        12709.80        23        .466          CSB-12        6.0
12199.70        12709.80        27        .197          CSB-12        6.0
```

This file has 37 samples in 7 boreholes. Since depth below the top surface is used instead of "Z" coordinates, line 2 contains the word Depth. Note that in this example there is only one analyte (e.g. chemistry) (property) value per line, but up to 300 could be included in which case line three of the file would read "37 300" and we would have 299 more columns of numbers in this file!.

A analyte (e.g. chemistry) fence diagram file has the exact same format, except that the samples from each boring must occur in the order of connectivity along the fence, and they should be sorted by increasing depth at each sample location.

Discussion of analyte (e.g. chemistry) Files for Fence Sections

analyte (e.g. chemistry) files to be used to create fence diagrams using the older Krig_Fence module, must contain only those borings that the user wishes to include on an individual cross section of the fence, in the order that they will be connected along the section. The result is that one .apdv file is produced for each cross section that will be included in the fence diagram, and the data for borings at which the fences will intersect are included in each of the intersecting cross section files. When geology is included on

the fence diagrams, the order of the borings in the analyte (e.g. chemistry) files must be identical to those in the geology files for each section. Generally, it is easiest to create the analyte (e.g. chemistry) file for a complete dataset, and then subset the fence diagram files from the complete file.

AIDV: Analyte Interval Data File Format

This format allows you to specify the top and bottom elevations of well screens and one or more concentrations that were measured over that interval. This new format (.aidv) will allow you to quickly visualize well screens in post_samples and automatically convert well screens to intelligently spaced samples along the screen interval for 3D (and 2D) kriging.

Format:

You may insert comment lines in C Tech Groundwater analyte (e.g. chemistry) (.aidv) input files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

Line 1: You may include any header message here (that does not start with a '#' character) ***unless*** you wish to include analyte names for use by other EVS modules (e.g. data component name). The format for line 1 to enable chemical names is as follows

A. Placing a pair of '@' symbols triggers the use and display of chemical names (example @@VOC). Any characters up to the @@ characters are ignored, and only the first analyte name needs @@, after that the chemical names must be delimited by spaces,

B. The following rules for commas are implemented to accommodate comma delimited files and also for using chemical names which have a comma within (example 1,1-DCA). Commas following a name will not become a part of the name, but a comma in the middle of a text string will be included in the name. The recommended approach is to put a space before the names.

C. If you want a space in your analyte name, you may use underscores and EVS will convert underscores to spaces (example: Vinyl_Chloride in a .aidv file will be converted to 'r;Vinyl Chloride." Or you may surround the entire name in quotation marks (example: "Vinyl Chloride").

The advantages of using chemical names (attribute names of any type) are the following:

- many modules use analyte names instead of data component numbers,
- when writing EVS Field files (.eff, .efb, etc.), you will get analyte names instead of data component numbers.
- when querying your data set with post_sample's mouse interactivity, the analyte name is displayed.
- time-series data can be used and the appropriate time-step can be displayed.

Line 2: **Specifications**

- **Elevation/Depth Specifier:** The first item on line 2 must be the word *Elevation* or *Depth* (case insensitive) to denote whether well screen top and bottom elevations are true elevation or depth below ground surface.

- **Maximum Gap:** The second parameter in this line is a real number (not an integer) specifying the Max-Gap. Max-gap is the maximum distance between samples for kriging. When a screen interval's total length is less than max-gap, a single sample is placed at the center of the interval. If the screen interval is longer than max-gap, two or more equally spaced samples are distributed within the interval. The number of samples is equal to the interval divided by max-gap rounded up to an integer.
    - [note: if you set max gap too small, you effectively create over-sampling in z (relative to x-y) for your data. On the other hand, if you have multiple screen intervals with different z extents and depths, choosing the proper value for max-gap will ensure better 3D distributions. If max-gap is set very large, only one sample is placed at the center of each screen interval. If the screens are small relative to the thickness of the aquifer, a large max gap is OK. If the screens are long (30% or more) of the local thickness and there are nearby screens with different depths/lengths, you will need a smaller max-gap value. Viewing your screen intervals with the spheres ON will help assess the optimal value.
- **Coordinate Units:** After Depth/Elevation, include the units of your coordinates (e.g. *feet* or *meters*)

Line 3: **Specifications**

- The first integer (n) is the number of well screens (rows of data) to follow.
- The second integer is the number of analyte (chemistry) values per well screen.
- The units of each data analyte column (e.g. *ppm* or *mg/l*).

Line 4: The first line of analyte interval (well screen) data must contain:

- X
- Y
- Well Screen Top
- Well Screen Bottom
- (one or more) Analyte Value(s) (chemistry or property)
- Well or Boring name. The boring name cannot contain spaces (recommend underscore "_" instead).
- Elevation of the top of the boring.

  Boring name and top are are optional parameters, but are used by many modules and it is highly recommended that you include this information in your file if possible. They are used by post_samples for posting tubes along borehole traces and for generating tubes which start from the ground surface of the borehole. Both Krig_3D and Krig_3D_Geology will use this information to determing the Z spatial extent of your grids (Krig_3D_Geology will create a layer that begins at ground surface if this information is provided). Numbers and names can be separated by one comma and/or any number of spaces or tabs.

  BLANK ENTRIES (CELLS) ARE NOT ALLOWED.

Please see the section on Handling Non-Detects for information on how to deal with samples whose concentration is below the detection limit. For any sample that is not detected you may enter any of the following. Please note that the first three flag words are not case sensitive, but must be spelled exactly as shown below.

- A fraction of the actual detection limit for that sample (**best approach**). Normally the fraction should be 0.1 to 0.5 (10 to 50%).
- nondetect
- non-detect
- nd
- 0.0 (zero) or

For files with multiple analytes such as the example below, if an analyte was not measured at a sample location, use any of the flags below to denote that this sample should be skipped for this analyte. Please note that these flag words are not case sensitive, but must be spelled exactly as shown below.

- missing
- unmeasured
- not-measured
- nm
- unknown
- unk
- na

Example Files are here:

AIDV: Analyte Interval Data File Format

This format allows you to specify the top and bottom elevations of well screens and one or more concentrations that were measured over that interval. This new format (.aidv) will allow you to quickly visualize well screens in post_samples and automatically convert well screens to intelligently spaced samples along the screen interval for 3D (and 2D) kriging.

Format:

You may insert comment lines in C Tech Groundwater analyte (e.g. chemistry) (.aidv) input files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

Line 1: You may include any header message here (that does not start with a '#' character) ***unless*** you wish to include analyte names for use by other EVS modules (e.g. data component name). The format for line 1 to enable chemical names is as follows

> **A.** Placing a pair of '@' symbols triggers the use and display of chemical names (example @@VOC). Any characters up to the @@ characters are ignored, and only the first analyte name needs @@, after that the chemical names must be delimited by spaces,

B. The following rules for commas are implemented to accommodate comma delimited files and also for using chemical names which have a comma within (example 1,1-DCA). Commas following a name will not become a part of the name, but a comma in the middle of a text string will be included in the name. The recommended approach is to put a space before the names.

C. If you want a space in your analyte name, you may use underscores and EVS will convert underscores to spaces (example: Vinyl_Chloride in a .aidv file will be converted to 'r;Vinyl Chloride." Or you may surround the entire name in quotation marks (example: "Vinyl Chloride").

The advantages of using chemical names (attribute names of any type) are the following:

- many modules use analyte names instead of data component numbers,
- when writing EVS Field files (.eff, .efb, etc.), you will get analyte names instead of data component numbers.
- when querying your data set with post_sample's mouse interactivity, the analyte name is displayed.
- time-series data can be used and the appropriate time-step can be displayed.

Line 2: **Specifications**

- **Elevation/Depth Specifier:** The first item on line 2 must be the word *Elevation* or *Depth* (case insensitive) to denote whether well screen top and bottom elevations are true elevation or depth below ground surface.
- **Maximum Gap:** The second parameter in this line is a real number (not an integer) specifying the Max-Gap. Max-gap is the maximum distance between samples for kriging. When a screen interval's total length is less than max-gap, a single sample is placed at the center of the interval. If the screen interval is longer than max-gap, two or more equally spaced samples are distributed within the interval. The number of samples is equal to the interval divided by max-gap rounded up to an integer.
    - [note: if you set max gap too small, you effectively create over-sampling in z (relative to x-y) for your data. On the other hand, if you have multiple screen intervals with different z extents and depths, choosing the proper value for max-gap will ensure better 3D distributions. If max-gap is set very large, only one sample is placed at the center of each screen interval. If the screens are small relative to the thickness of the aquifer, a large max gap is OK. If the screens are long (30% or more) of the local thickness and there are nearby screens with different depths/lengths, you will need a smaller max-gap value. Viewing your screen intervals with the spheres ON will help assess the optimal value.
- **Coordinate Units:** After Depth/Elevation, include the units of your coordinates (e.g. *feet* or *meters*)

Line 3: **Specifications**

- The first integer (n) is the number of well screens (rows of data) to follow.

- The second integer is the number of analyte (chemistry) values per well screen.
- The units of each data analyte column (e.g. *ppm* or *mg/l*).

Line 4: The first line of analyte interval (well screen) data must contain:

- X
- Y
- Well Screen Top
- Well Screen Bottom
- (one or more) Analyte Value(s) (chemistry or property)
- Well or Boring name. The boring name cannot contain spaces (recommend underscore "_" instead).
- Elevation of the top of the boring.

  Boring name and top are are optional parameters, but are used by many modules and it is highly recommended that you include this information in your file if possible. They are used by post_samples for posting tubes along borehole traces and for generating tubes which start from the ground surface of the borehole. Both Krig_3D and Krig_3D_Geology will use this information to determing the Z spatial extent of your grids (Krig_3D_Geology will create a layer that begins at ground surface if this information is provided). Numbers and names can be separated by one comma and/or any number of spaces or tabs.

  BLANK ENTRIES (CELLS) ARE NOT ALLOWED.

Please see the section on Handling Non-Detects for information on how to deal with samples whose concentration is below the detection limit. For any sample that is not detected you may enter any of the following. Please note that the first three flag words are not case sensitive, but must be spelled exactly as shown below.

- A fraction of the actual detection limit for that sample (**best approach**). Normally the fraction should be 0.1 to 0.5 (10 to 50%).
- nondetect
- non-detect
- nd
- 0.0 (zero) or

For files with multiple analytes such as the example below, if an analyte was not measured at a sample location, use any of the flags below to denote that this sample should be skipped for this analyte. Please note that these flag words are not case sensitive, but must be spelled exactly as shown below.

- missing
- unmeasured
- not-measured
- nm
- unknown

- unk
- na

Example Files are here:

AIDV File Examples

An actual .aidv file could look like the following:

```
# This is a comment line....any line that starts with # is
ignored
X                   Y                   Ztop    Zbot    @@TOTHC         Bore    Top
Elevation           6.0                 feet
10                  1                   mg/l
11086.52            12830.67            -13     -26     2.000           W-49    4.5
11199.04            12810.16            -18     -30     2.000           W-51    4
11298.00            12808.63            -12     -38     3600.           W-52    3
11566.34            12850.59            -14     -25     0.000           W-30    7.5
11251.30            12929.27            -24     -30     33000           W-75    2
11248.75            12870.91            -17     -22     5004.8          W-48    3
11340.49            12892.61            -11     -16     120.0           W-47    2.5
11340.49            12892.61            -22     -28     320.0           W-47    2.5
11338.00            12830.80            -13     -20     640.0           W-38    4
11401.73            12897.77            -36     -40     0.300           W-45    4
```

This example file above (10_well_screens.aidv) has 10 well screens in 9 boreholes. Well W-47 has two different screen intervals. Note that line 2 contains the word Elevation and the number 6.0 which is the max-gap parameter. There are 10 rows of data and there is only one analyte value per line, but up to 300 could be included in a single file.

Analyte Time Files Format

Discussion of Analyte Time Files

Analyte time files contain 3-D coordinates (x, y, z) describing the locations of samples and values of one or more analytes or properties taken over a series of different times. Time files must conform to the ASCII formats described below and individual entries (coordinates or measurements) can be delimited by commas, spaces, or tabs. They must have either a **.sct** (Soil Chemistry Time) or **.gwt** (Ground Water Time) suffix to be selected in the file browsers of EVS modules. Each line of the file contains the coordinate data for one sampling location, or well screen, and any number of chemistry or property values. There are no limits on the number of borings and/or samples that can be included in these files, except that run times for execution of kriging do increase with a greater number of samples in the file.

Time data can be visualized independently (without geology data) or within a domain bounded by a geologic system. When a geologic domain is utilized for a 3-D visualization, a consistent coordinate system (the same projection and overlapping spatial extents) must be used for both the chemistry and geology. The boring and sample locations in the time files do not have to

correspond to those in the geology files, except that only those contained within or proximal to the spatial domain of the geology will be used for the kriging.

If the posting of borings and sample locations are to honor the topography of the site, the chemistry files also must contain the top surface elevation of each boring.

Format:

You may insert comment lines anywhere in Analyte time files. Comments must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

The format of chemistry time files is substantially different from other analyte file formats (.apdv or .aidv) used in EVS/MVS. These differences include**required**analyte name and unitson line one (no other information allowed), and no need to specify the number of samples or number of analytesandtimes.

Line 1: This line contains the name of each analyte. After every analyte has been listed the analyte units are then required for each analyte. Analyte Units are **REQUIRED** for time chemistry files.

Line 2: This line contains the mapping of the analytes to a specific date. This is done by listing the analyte name followed by a pipe character "|" and then followed by the sampling date. There should be one of these mappings for every column of data in the file. If you want a space in your analyte name you may enclose the entire name and date in quotation marks (example: "Vinyl Chloride|6/1/2004"). Optionally the analyte name may be omitted and just a date used, in this case the first analyte name listed on line one will be used.

It is required that the order of analyte-date columns be from oldest to newest for each analyte.

The date format is dependent on your REGIONAL SETTINGS on your computer (control panel).

C Tech uses the SHORT DATE and SHORT TIME formats.

If the date/time works in Excel it will likely work in EVS.

For most people in the U.S., this would not be 24 hour clock so you would need:

"m/d/yyyy hh:mm:ss AM" or "m/d/yyyy hh:mm:ss PM"

Also, you MUST put the date/time in quotes if you use more than just date (i.e. if there are spaces in the total date/time).

Line 3: This line must contain the word Elevation or Depth to denote whether sample elevations are true elevation or depth below ground surface. If actual elevations are used (a right-handed coordinate system), then this parameter should be *Elevation*; if depths below the top surface elevation are used, then this parameter should be *Depth*.

> FOR GWT FILESONLY:the second parameter in this line is a real number (not an integer) specifying the Max-Gap in the same units as your coordinate data. Max-gap is the maximum distance between

samples for kriging. When a screen interval's total length is less than max-gap, a single sample is placed at the center of the interval. If the screen interval is longer than max-gap, two or more equally spaced samples are distributed within the interval. The number of samplesis equal to theinterval divided by max-gap roundedupto an integer.

The last value on this line should be the units of your coordinates (e.g. feet or meters), or the flag word reproject.

Lines 4+: *The lines of sample data:*The content of these lines varies whether the files is a SCT or GWT file. GWT files have an additional column of elevation (Z) data to allow for specification of the top and bottom of each screen interval, whereas SCT files specify the location of a POINT sample (requiring only a single elevation).

X, **Y**, **Z** (for Chemistry files or Well Screen Top), **Well Screen Bottom** for groundwater chemistry files) , (one or more) **Analyte Value**(s) (chemistry or property), **Boring name**, and **Elevation of the Top Of The Boring** (optional).

There are several flag words available for **missing values** these include:

a. unmeasured
b. not-measured
c. nm
d. missing
e. unknown
f. unk
g. na

For **non-detect** samples the following flag words are available:

a. nondetect
b. nd
c. non-detect

The boring name cannot contain spaces (recommend underscore "_" instead), unless surrounded by quotation marks (example: "B 1"). The optional boring name and top are needed only by the post_samples module for posting tubes along borehole traces and for generating tubes which start from the ground surface of the borehole. Numbers and names can be separated by one comma and/or any number of spaces or tabs.**BLANK ENTRIES (CELLS) ARE NOT ALLOWED.**

When Top of Boring elevations are given, they must be provided for all lines of the file.

```
#Soil Chemistry Time File Example (SCT)
"ethane""ethylene""mg/kg""ug/kg"
```

```
"ethane|6/8/1976""ethylene|6/8/1976""ethane|1/12/1979" "ethylene|1/12/1979"
"ethylene|3/16/1981"
Elevation meters
12008 12431 22.9 22 Unk 21 500 0 CSB-39 30.4
11271 13105 18.9 0 0 0 2800 0 CSB-40 35.9
10652 13857 23.4 0 0 0 290 0 CSB-41 28.1
9904 14522 18.4 0 0 0 Unk Unk CSB-42 22.8
9029 15283 37.9 0 0 0 23 0 CSB-43 30.1
```

For the GWT file below, those items that are unique to GWT (vs. SCT) are in BLUE.

```
#Ground WaterChemistry Time File Example (GWT)
"ethane""ethylene""mg/kg""ug/kg"
"ethane|6/8/1976""ethylene|6/8/1976""ethane|1/12/1979" "ethylene|1/12/1979"
"ethylene|3/16/1981"
Elevation3.0meters
12008 12431 22.9 15.2 22 Unk 21 500 0 CSB-39 30.4
11271 13105 18.9 12.5 0 0 0 2800 0 CSB-40 35.9
10652 13857 23.4 19.0 0 0 0 290 0 CSB-41 28.1
9904 14522 18.4 11.8 0 0 0 Unk Unk CSB-42 22.8
9029 15283 37.9 30.3 0 0 0 23 0 CSB-43 30.1
```

Time Domain Analyte Data

We recommend that analyte files which represent data collected over time use either the APDV or AIDV format and include data for only a single analyte

We do not recommend using the SCT or GWT formats.

When using APDV or AIDV files for time domain data, the following rules apply:

- Include data for only a **single** analyte
- Group measurements taken over a few days or even weeks into the same DATE GROUP. If your entire site is re-sampled every 3 months, do not separately list each day when a particular well is sampled.
- The "analyte name" for each column of data representing a Date Group should be the average date for that sampling event. The date must be in the Windows standard short date format. In the United States that is typically MM/DD/YYYY (e.g. 11/08/2003 for November 8, 2003)
- The data file cannot specify the actual analyte name (e.g. benzene). However, the modules which deal with time domain data have the ability to specify the actual name and units.
- Date groups need not be at equal time intervals.

Time Domain AIDV Example File

x	y	ztop	zbot	@@1/1/2001	5/1/2001	8/1/2001
Elevation	10	m				
98	5			mg/l	mg/l	mg/l
2772536.7	331635.8	886.5	866.5	6	5	5
2772554.6	331635.2	987.4	967.4	0.71	5	5
2772601.5	333091.7	862.1	852.1	0.71	5	5
2772610.4	333100.5	950.6	930.6	0.71	1	1
2772830.1	336800.0	853.5	833.5	190	130	125
2772982.4	333214.1	955.3	935.3	5	5	5
2773014.8	331825.0	954.0	934.0	180	nm	nm
2773014.8	331825.0	881.9	861.9	150	nm	nm
2773069.9	332631.8	888.1	868.1	35	36	40
2773076.0	332138.7	959.5	949.5	48	48	55
2773087.1	332138.3	994.4	974.4	0.71	1	10
2773091.3	332611.7	784.4	684.4	5	5	5
2773104.2	332134.5	887.6	867.6	440	480	500
2773129.1	332136.9	736.0	686.0	0.71	5	5
2773146.2	333741.7	862.5	842.5	300	330	240
2773149.9	333225.7	1020.1	990.1	2650	2500	2350
2773156.3	333244.4	1017.8	987.8	750	690	13500
2773156.6	333219.8	1002.0	982.0	200	200	200
2773157.7	333579.1	946.1	941.1	0.71	2	5
2773159.4	333587.1	1006.4	986.4	0.71	1	1
2773165.1	333262.3	1013.1	993.1	10000	10000	30000
2773182.8	333309.7	1009.2	989.2	45000	43000	53500
2773192.1	333368.0	796.2	779.2	5	5	5
2773192.5	333361.4	870.7	853.7	19	11	22
2773196.2	333647.9	936.4	921.4	29	100	130
2773236.4	333568.8	1016.6	1016.6	10	9	nm
2773253.6	333567.2	1017.0	1017.0	800	800	770
2773266.3	335344.6	908.3	888.3	6	nm	nm
2773290.3	335351.9	833.0	813.0	610	nm	nm
2773307.6	333207.6	1005.5	985.5	2000	1900	1500
2773308.9	333198.4	945.6	940.6	180	180	200
2773323.3	333554.5	1016.3	996.3	750	510	7700
2773324.5	333353.1	947.0	942.0	750	750	675
2773325.8	333349.2	1009.5	989.5	100	91	85
2773326.6	333529.3	1012.4	992.4	1100	1000	810
2773328.0	333518.5	1021.1	1001.1	800	730	700
2773439.9	333202.0	994.0	974.0	90	88	80
2773441.7	333077.6	1009.3	989.3	410	410	400

2773446.4	333203.9	946.0	941.0	5	5	5
2773457.6	333081.2	890.2	870.2	400	380	275
2773462.8	333364.4	1000.7	980.7	11000	11000	10550
2773477.3	333524.2	941.8	936.8	5	5	5
2773480.4	333449.2	1010.0	980.0	7000	6600	5750
2773480.5	333522.5	1006.9	986.9	350	350	375
2773482.1	333669.2	946.5	931.5	0.71	1	5
2773541.1	333784.9	876.4	826.4	230	240	290
2773570.2	333713.2	1013.2	989.9	0.71	1	5
2773571.6	333770.9	853.5	833.5	100	110	160
2773572.2	332825.6	1008.8	988.8	25	26	27
2773573.4	332844.1	903.4	883.4	125	120	175
2773575.8	333740.1	738.3	688.3	0.71	5	5
2773620.0	332116.7	1019.5	996.5	5	5	5
2773630.2	332116.9	959.4	939.4	1	1	5
2773663.4	332966.1	1003.8	983.8	700	610	625
2773672.4	332971.5	889.9	869.9	75	65	240
2773688.4	332956.9	743.3	693.3	5	5	5
2773689.4	333385.8	997.9	977.9	370	190	420
2773692.6	333066.4	882.0	862.0	800	750	950
2773708.8	333065.2	1007.8	987.8	250000	220000	260000
2773713.9	333494.8	860.6	849.1	100	270	190
2773714.1	333523.8	1006.5	986.5	36	36	35
2773717.9	333532.7	941.2	936.2	31	31	30
2773730.5	331660.3	906.0	886.0	0.71	nm	nm
2773732.8	331687.1	950.3	930.3	0.71	nm	nm
2773735.5	333543.7	784.5	734.5	0.71	5	5
2773760.8	333319.1	936.3	931.3	8	8	8
2773763.3	333330.4	997.1	977.1	59262	57805	56348
2773765.6	333309.4	1013.0	963.0	770	820	890
2773797.1	333060.9	1008.8	988.8	97	97	95
2773899.9	333080.3	967.1	957.1	10	12	12
2773902.7	333097.7	915.8	905.8	5	9	12
2774022.9	333742.9	882.9	832.9	46	95	77
2774033.8	333513.5	986.9	974.9	2	2	2
2774051.8	333512.9	1027.5	1005.5	2100	2100	2100
2774065.2	333730.6	983.5	963.5	5	250	5
2774073.1	333738.4	858.5	838.5	0.71	0.71	3
2774073.7	334671.8	947.1	937.1	0.71	1	4
2774076.5	333728.3	823.7	823.7	0.71	2	2
2774083.0	332103.9	866.4	856.4	98	85	100
2774085.3	333736.6	996.9	973.5	16	25	37
2774087.2	334674.8	792.4	782.4	22	20	19

2774094.7	333745.8	936.3	924.5	16	14	50
2774186.2	331604.2	873.9	853.9	0.71	5	5
2774187.3	333087.0	911.3	891.3	16	22	25
2774194.8	333100.9	973.6	953.6	5	5	5
2774324.1	334101.7	922.3	912.3	0.71	1	5
2774332.3	333623.1	881.4	861.4	0.71	3	5
2774338.3	333327.8	998.8	981.5	0.71	2	5
2774341.9	333638.3	1022.6	999.4	5	5	5
2774344.3	333870.5	862.2	852.2	5	5	6
2774352.8	333882.0	898.1	888.1	0.71	1	4
2774664.2	334463.8	845.0	835.0	0.71	1	5
2774677.0	334462.1	961.0	951.0	130	120	135
2774820.0	333352.3	883.5	863.5	0.71	5	5
2774995.8	336287.5	694.9	644.9	0.71	5	5
2774995.9	336310.6	831.8	811.8	30	31	34
2775092.1	334397.8	946.4	936.4	10	9	10
2777126.6	336231.0	809.7	789.7	0.71	5	5

Analyte Time Files Format

Discussion of Analyte Time Files

Analyte time files contain 3-D coordinates (x, y, z) describing the locations of samples and values of one or more analytes or properties taken over a series of different times. Time files must conform to the ASCII formats described below and individual entries (coordinates or measurements) can be delimited by commas, spaces, or tabs. They must have either a **.sct** (Soil Chemistry Time) or **.gwt** (Ground Water Time) suffix to be selected in the file browsers of EVS modules. Each line of the file contains the coordinate data for one sampling location, or well screen, and any number of chemistry or property values. There are no limits on the number of borings and/or samples that can be included in these files, except that run times for execution of kriging do increase with a greater number of samples in the file.

Time data can be visualized independently (without geology data) or within a domain bounded by a geologic system. When a geologic domain is utilized for a 3-D visualization, a consistent coordinate system (the same projection and overlapping spatial extents) must be used for both the chemistry and geology. The boring and sample locations in the time files do not have to correspond to those in the geology files, except that only those contained within or proximal to the spatial domain of the geology will be used for the kriging.

If the posting of borings and sample locations are to honor the topography of the site, the chemistry files also must contain the top surface elevation of each boring.

Format:

You may insert comment lines anywhere in Analyte time files. Comments must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

The format of chemistry time files is substantially different from other analyte file formats (.apdv or .aidv) used in EVS/MVS. These differences include**required**analyte name and unitson line one (no other information allowed), and no need to specify the number of samples or number of analytesandtimes.

Line 1: This line contains the name of each analyte. After every analyte has been listed the analyte units are then required for each analyte. Analyte Units are **REQUIRED** for time chemistry files.

Line 2: This line contains the mapping of the analytes to a specific date. This is done by listing the analyte name followed by a pipe character "|" and then followed by the sampling date. There should be one of these mappings for every column of data in the file. If you want a space in your analyte name you may enclose the entire name and date in quotation marks (example: "Vinyl Chloride|6/1/2004"). Optionally the analyte name may be omitted and just a date used, in this case the first analyte name listed on line one will be used.

It is required that the order of analyte-date columns be from oldest to newest for each analyte.

The date format is dependent on your REGIONAL SETTINGS on your computer (control panel).

C Tech uses the SHORT DATE and SHORT TIME formats.

If the date/time works in Excel it will likely work in EVS.

For most people in the U.S., this would not be 24 hour clock so you would need:

"m/d/yyyy hh:mm:ss AM" or "m/d/yyyy hh:mm:ss PM"

Also, you MUST put the date/time in quotes if you use more than just date (i.e. if there are spaces in the total date/time).

Line 3: This line must contain the word Elevation or Depth to denote whether sample elevations are true elevation or depth below ground surface. If actual elevations are used (a right-handed coordinate system), then this parameter should be *Elevation*; if depths below the top surface elevation are used, then this parameter should be *Depth*.

> FOR GWT FILESONLY:the second parameter in this line is a real number (not an integer) specifying the Max-Gap in the same units as your coordinate data. Max-gap is the maximum distance between samples for kriging. When a screen interval's total length is less than max-gap, a single sample is placed at the center of the interval. If the screen interval is longer than max-gap, two or more equally spaced samples are distributed within the interval. The number of samplesis equal to theinterval divided by max-gap roundedupto an integer.

The last value on this line should be the units of your coordinates (e.g. feet or meters), or the flag word reproject.

Lines 4+: *The lines of sample data:*The content of these lines varies whether the files is a SCT or GWT file. GWT files have an additional column of elevation (Z) data to allow for specification of the top and bottom of each screen interval, whereas SCT files specify the location of a POINT sample (requiring only a single elevation).

X, **Y**, **Z** (for Chemistry files or Well Screen Top), **Well Screen Bottom** for groundwater chemistry files) , (one or more) **Analyte Value**(s) (chemistry or property), **Boring name**, and **Elevation of the Top Of The Boring** (optional).

There are several flag words available for **missing values** these include:

a. unmeasured
b. not-measured
c. nm
d. missing
e. unknown
f. unk
g. na

For **non-detect** samples the following flag words are available:

a. nondetect
b. nd
c. non-detect

The boring name cannot contain spaces (recommend underscore "_" instead), unless surrounded by quotation marks (example: "B 1"). The optional boring name and top are needed only by the post_samples module for posting tubes along borehole traces and for generating tubes which start from the ground surface of the borehole. Numbers and names can be separated by one comma and/or any number of spaces or tabs.**BLANK ENTRIES (CELLS) ARE NOT ALLOWED.**

When Top of Boring elevations are given, they must be provided for all lines of the file.

```
#Soil Chemistry Time File Example (SCT)
"ethane""ethylene""mg/kg""ug/kg"
"ethane|6/8/1976""ethylene|6/8/1976""ethane|1/12/1979" "ethylene|1/12/1979"
"ethylene|3/16/1981"
Elevation meters
12008 12431 22.9 22 Unk 21 500 0 CSB-39 30.4
11271 13105 18.9 0 0 0 2800 0 CSB-40 35.9
10652 13857 23.4 0 0 0 290 0 CSB-41 28.1
9904 14522 18.4 0 0 0 Unk Unk CSB-42 22.8
9029 15283 37.9 0 0 0 23 0 CSB-43 30.1
```

For the GWT file below, those items that are unique to GWT (vs. SCT) are in BLUE.

```
#Ground WaterChemistry Time File Example (GWT)
"ethane""ethylene""mg/kg""ug/kg"
"ethane|6/8/1976""ethylene|6/8/1976""ethane|1/12/1979" "ethylene|1/12/1979"
"ethylene|3/16/1981"
Elevation3.0meters
12008 12431 22.9 15.2 22 Unk 21 500 0 CSB-39 30.4
11271 13105 18.9 12.5 0 0 0 2800 0 CSB-40 35.9
10652 13857 23.4 19.0 0 0 0 290 0 CSB-41 28.1
9904 14522 18.4 11.8 0 0 0 Unk Unk CSB-42 22.8
9029 15283 37.9 30.3 0 0 0 23 0 CSB-43 30.1
```

Pre Geology File

The ASCII pregeology file name must have a .pgf suffix to be selected in the module's file browser. This file type represents raw (uninterpreted) 3D boring logs. This format is used by:

- make_geo_hierarchy
- post_samples
- Krig_3D_Geology (to extract a top and bottom surface to build a single layer)
- Indicator_Geology (both PRO and MVS only) for Geologic Indicator Kriging (GIK).
- adaptive_indicator_krig (MVS only)

You may insert comment lines in C Tech Pre Geology (.pgf) input files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

The pre-geology file format is used to represent raw 3D boring logs. We also refer to this geologic data format as "uninterpreted". This is not meant to imply that no form of geologic evaluation or interpretation has occurred. On the contrary, it is required that someone categorizes the materials on the site and in each boring.

Data Concept:

- A PGF file can be considered a **group of file sections** where each section represents **each boring** (well).
- It is essential to use the same ID for the ground surface (first line) as for the bottom of the first observed material (second line) in each section (boring). If a different material ID is used a synthetic point will be added between the ground and first observed material. This will be reported for the first five occurrences.

    - Think about the PGF file as a shorthand way of specifying intervals. The first line is the FROM. The second is the TO.
- Please note that the data for each boring must be sorted (by you) from beginning to end (normally top to bottom).
    - We cannot sort this data for you because some borings may turn to horizontal or even upwards. It is your responsibility to make sure that the data is in the proper order.
    - If the data is unsorted, and within a boring the direction between two values varies by more than 90 degrees, an error will be reported.

FILE FORMAT:

- **Line 1:** May contain any header message, but cannot be left blank or commented. There is no information content in this line.
- **Line 2**: Line 2 contains the declaration of Elevation or Depth, the definitions of Lithology IDs and Names, and coordinate units.
    - **Elevation/Depth Specifier:** This line must contain the word *Elevation* or *Depth* (case insensitive) to specify whether well screen top and bottom elevations are true elevation or depth below ground surface.
        - **Depth** forces the otherwise optional ground surface elevation column to be required. Depths given in column 3 are distances below the ground surface elevation in the last column (column 6). If the top surface is omitted, a value of 0.0 will be assumed and a warning message will be printed to the EVS Status Window.
    - **IDs and Names**: Line 2 should contain Lithology IDs and corresponding names for each material. Each Name is explicitly associated with its corresponding Lithology ID and the pairs are delimited by a pipe symbol "|".
        - Though it is generally advisable, IDs need not be sequential and may be any integer values. This allow for a unified set of Lithology IDs and Names to be applied to a large site where models create for sub-sites may not have all materials.
        - The number of (material) IDs and Names MUST be equal to the number of Lithology IDs specified in the data section. Each material ID present in the data section must have corresponding Lithology IDs and Names. If there are four materials represented in your .pgf file, there should be at least four IDs and Names on line two.
        - The order of Lithology IDs and Names will determine the order that they appear in Legends. The IDs do not need to be sequential.
        - You can specify additional IDs and Names, which are not in the data and those will appear on Legends.
    - **Coordinate Units:** You should include the units of your coordinates (e.g. *feet* or *meters*). If this is included it must follow the names associated with each Lithology ID.

- **Line 3:** Must be the number of lines of data (n) to follow. For each boring, there is one line for the ground surface and one line for the bottom of each observed lithologic unit. Therefore the total number of lines in the file should be equal to the number of borings PLUS the sum of the number of materials observed in each boring.
- **Line 4:** First line of sample data. X, Y, Z, "Lithology ID", Boring name, and Ground surface elevation. The Ground surface elevation is an optional parameter which is required if *Depth* is specified on line 2. If depths are used (instead of elevations) the top surface should be in the same coordinate system. Depths are relative to the Ground surface (which is assumed at 0.0 if the Ground surface is not defined). The boring name cannot contain spaces unless the entire name is surrounded in quotation marks (example "Boring 1D"). One comma and/or any number of spaces or tabs can separate numbers and name.
- **Line 3+n:** is the last line of the file.

AN EXAMPLE FILE FOLLOWS:

```
Pregeology    file

Elevation     1|Silt      2|Fill      3|Clay      4|Sand      5|Gravel   ft

17

11086.5       12830.7     4.5         1           B-49

11086.5       12830.7     -3.8        1           B-49

11086.5       12830.7     -21         2           B-49

11086.5       12830.7     -26         3           B-49

11086.5       12830.7     -42         4           B-49

11086.5       12830.7     -55         5           B-49

11199         12600       4           1           B-51

11199         12600       -5          1           B-51

11199         12600       -20         2           B-51

11199         12600       -25         3           B-51

11199         12600       -39         4           B-51

11199         12600       -53         5           B-51

11259.7       12819.3     2           1           B-46

11259.7       12819.3     -7.5        1           B-46

11259.7       12819.3     -27         3           B-46

11259.7       12819.3     -40         4           B-46

11259.7       12819.3     -53         5           B-46
```

Pre Geology File

The ASCII pregeology file name must have a .pgf suffix to be selected in the module's file browser. This file type represents raw (uninterpreted) 3D boring logs. This format is used by:

- make_geo_hierarchy
- post_samples
- Krig_3D_Geology (to extract a top and bottom surface to build a single layer)
- Indicator_Geology (both PRO and MVS only) for Geologic Indicator Kriging (GIK).
- adaptive_indicator_krig (MVS only)

You may insert comment lines in C Tech Pre Geology (.pgf) input files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

The pre-geology file format is used to represent raw 3D boring logs. We also refer to this geologic data format as "uninterpreted". This is not meant to imply that no form of geologic evaluation or interpretation has occurred. On the contrary, it is required that someone categorizes the materials on the site and in each boring.

Data Concept:

- A PGF file can be considered a **group of file sections** where each section represents **each boring** (well).
- It is essential to use the same ID for the ground surface (first line) as for the bottom of the first observed material (second line) in each section (boring). If a different material ID is used a synthetic point will be added between the ground and first observed material. This will be reported for the first five occurrences.
    - Think about the PGF file as a shorthand way of specifying intervals. The first line is the FROM. The second is the TO.
- Please note that the data for each boring must be sorted (by you) from beginning to end (normally top to bottom).
    - We cannot sort this data for you because some borings may turn to horizontal or even upwards. It is your responsibility to make sure that the data is in the proper order.
    - If the data is unsorted, and within a boring the direction between two values varies by more than 90 degrees, an error will be reported.

FILE FORMAT:

- **Line 1:** May contain any header message, but cannot be left blank or commented. There is no information content in this line.
- **Line 2**: Line 2 contains the declaration of Elevation or Depth, the definitions of Lithology IDs and Names, and coordinate units.
    - **Elevation/Depth Specifier:** This line must contain the word *Elevation* or *Depth* (case insensitive) to specify whether well screen top and bottom elevations are true elevation or depth below ground surface.
        - **Depth** forces the otherwise optional ground surface elevation column to be required. Depths given in column 3 are distances below the ground surface elevation in the last column (column 6). If the top surface is omitted, a value of 0.0 will be assumed and a warning message will be printed to the EVS Status Window.
    - **IDs and Names**: Line 2 should contain Lithology IDs and corresponding names for each material. Each Name is explicitly associated with its corresponding Lithology ID and the pairs are delimited by a pipe symbol "|".
        - Though it is generally advisable, IDs need not be sequential and may be any integer values. This allow for a unified set of Lithology IDs and Names to be applied to a large site where models create for sub-sites may not have all materials.
        - The number of (material) IDs and Names MUST be equal to the number of Lithology IDs specified in the data section. Each material ID present in the data section must have corresponding Lithology IDs and Names. If there are four materials represented in your .pgf file, there should be at least four IDs and Names on line two.
        - The order of Lithology IDs and Names will determine the order that they appear in Legends. The IDs do not need to be sequential.
        - You can specify additional IDs and Names, which are not in the data and those will appear on Legends.
    - **Coordinate Units:** You should include the units of your coordinates (e.g. *feet* or *meters*). If this is included it must follow the names associated with each Lithology ID.
- **Line 3:** Must be the number of lines of data (n) to follow. For each boring, there is one line for the ground surface and one line for the bottom of each observed lithologic unit. Therefore the total number of lines in the file should be equal to the number of borings PLUS the sum of the number of materials observed in each boring.
- **Line 4:** First line of sample data. X, Y, Z, "Lithology ID", Boring name, and Ground surface elevation. The Ground surface elevation is an optional parameter which is required if *Depth* is specified on line 2. If depths are used (instead of elevations) the top surface should be in the same coordinate system. Depths are relative to the Ground surface (which is assumed at 0.0 if the Ground surface is not defined). The boring name cannot contain spaces unless the entire name is surrounded in quotation marks (example "Boring 1D"). One comma and/or any number of spaces or tabs can separate numbers and name.
- **Line 3+n:** is the last line of the file.

AN EXAMPLE FILE FOLLOWS:

```
Pregeology    file

Elevation     1|Silt      2|Fill      3|Clay      4|Sand      5|Gravel   ft
17

11086.5       12830.7     4.5         1           B-49

11086.5       12830.7     -3.8        1           B-49

11086.5       12830.7     -21         2           B-49

11086.5       12830.7     -26         3           B-49

11086.5       12830.7     -42         4           B-49

11086.5       12830.7     -55         5           B-49

11199         12600       4           1           B-51

11199         12600       -5          1           B-51

11199         12600       -20         2           B-51

11199         12600       -25         3           B-51

11199         12600       -39         4           B-51

11199         12600       -53         5           B-51

11259.7       12819.3     2           1           B-46

11259.7       12819.3     -7.5        1           B-46

11259.7       12819.3     -27         3           B-46

11259.7       12819.3     -40         4           B-46

11259.7       12819.3     -53         5           B-46
```

PGF File Examples

In the (very short) example file below, please note that the Lithology IDs and Names are not ordered by increasing ID number. The order that you specify the Lithology IDs and Names determines the order that is used for exploding the lithologic materials and the ordering in Legends. Also notice that Lithology ID 3 is specified in line 2, but not present in the data. Silty-Sand will be shown in the Legend, but will not be present in the borings displayed with post_samples nor any model created with this data.

```
EAST NORTH TOP-BOT MATERIAL-ID WELL_ID
Elevation  4|Sand  5|Gravel  1|Clay  2|Silt  3|Silty-sand meters
11
2085487.12    322869.95    31    4    AW-3
2085487.12    322869.95    -1    4    AW-3
2085487.12    322869.95    -3    2    AW-3
2085108.47    323363.89    32    4    MW-10A
2085108.47    323363.89    20    4    MW-10A
2085108.47    323363.89    12    5    MW-10A
2085079.22    323361.25    32    4    MW-10B
2085079.22    323361.25    20    4    MW-10B
2085266.93    323410.05    32    4    MW-11A
2085266.93    323410.05    14    4    MW-11A
2085266.93    323410.05    7     1    MW-11A
```

In the realistic example below, IDs are listed in ascending order and this order on Line 2 will be the order used for exploding materials and Legends.

```
Pre-Geology File for Initial
Painting Facility soil investigation
Elevation1|Silt 2|Fill3|Clay4|Sand  5|Gravel    ft
144
11086.5  12830.74.5   2     B-49
11086.5  12830.7-3.8  2     B-49
11086.5  12830.7-21.0 1     B-49
11086.5  12830.7-26.0 3     B-49
11086.5  12830.7-42.0 5     B-49
11086.5  12830.7-55.0 4     B-49
11199.0  12810.24.0   2     B-51
11199.0  12810.2-5.0  2     B-51
11199.0  12810.2-20.0 1     B-51
11199.0  12810.2-25.0 3     B-51
11199.0  12810.2-39.0 5     B-51
11199.0  12810.2-53.0 4     B-51
11259.7  12819.32.0   2     B-46
```

```
11259.7  12819.3-7.5  2     B-46
11259.7  12819.3-20.5 1     B-46
11259.7  12819.3-27.0 3     B-46
11259.7  12819.3-40.0 5     B-46
11259.7  12819.3-53.0 4     B-46
11298.0  12808.63.0   2     B-52
11298.0  12808.6-6.0  2     B-52
11298.0  12808.6-19.0 1     B-52
11298.0  12808.6-25.8 3     B-52
11298.0  12808.6-41.8 5     B-52
11298.0  12808.6-55.0 4     B-52
11414.4  12781.12.0   2     B-34
11414.4  12781.1-6.0  2     B-34
11414.4  12781.1-20.5 1     B-34
11414.4  12781.1-28.0 3     B-34
11414.4  12781.1-42.0 5     B-34
11414.4  12781.1-55.0 4     B-34
11427.0  12780.96.5   2     B-42
11427.0  12780.9-7.0  2     B-42
11427.0  12780.9-23.0 1     B-42
11427.0  12780.9-28.5 3     B-42
11427.0  12780.9-38.5 5     B-42
11427.0  12780.9-51.0 4     B-42
11496.3  12753.61.5   2     B-53
11496.3  12753.6-7.5  2     B-53
11496.3  12753.6-20.0 1     B-53
11496.3  12753.6-28.8 3     B-53
11496.3  12753.6-38.8 5     B-53
11496.3  12753.6-51.0 4     B-53
11209.4  12993.92.0   2     B-57
11209.4  12993.9-3.0  2     B-57
11209.4  12993.9-23.0 1     B-57
11209.4  12993.9-27.5 3     B-57
11209.4  12993.9-37.5 5     B-57
11209.4  12993.9-51.0 4     B-57
11251.3  12929.32.0   2     B-75
11251.3  12929.3-2.5  2     B-75
11251.3  12929.3-22.0 1     B-75
11251.3  12929.3-28.0 3     B-75
11251.3  12929.3-40.0 5     B-75
11251.3  12929.3-53.0 4     B-75
11248.8  12870.93.0   2     B-48
11248.8  12870.9-3.5  2     B-48
```

```
11248.8  12870.9-22.0 1     B-48
11248.8  12870.9-28.5 3     B-48
11248.8  12870.9-36.3 5     B-48
11248.8  12870.9-50.0 4     B-48
11211.9  12710.82.0   2     B-50
11211.9  12710.8-6.5  2     B-50
11211.9  12710.8-22.5 1     B-50
11211.9  12710.8-27.5 3     B-50
11211.9  12710.8-37.5 5     B-50
11211.9  12710.8-51.0 4     B-50
11302.0  13079.74.5   2     B-58
11302.0  13079.7-3.5  2     B-58
11302.0  13079.7-21.9 1     B-58
11302.0  13079.7-26.0 3     B-58
11302.0  13079.7-38.0 5     B-58
11302.0  13079.7-51.0 4     B-58
11286.8  13026.72.0   2     B-59
11286.8  13026.7-5.0  2     B-59
11286.8  13026.7-23.0 1     B-59
11286.8  13026.7-29.0 3     B-59
11286.8  13026.7-37.0 5     B-59
11286.8  13026.7-50.0 4     B-59
11309.0  12949.04.0   2     B-56
11309.0  12949.0-2.5  2     B-56
11309.0  12949.0-22.0 1     B-56
11309.0  12949.0-28.3 3     B-56
11309.0  12949.0-38.3 5     B-56
11309.0  12949.0-52.0 4     B-56
11340.5  12892.62.5   2     B-47
11340.5  12892.6-2.5  2     B-47
11340.5  12892.6-20.0 1     B-47
11340.5  12892.6-23.0 3     B-47
11340.5  12892.6-38.0 5     B-47
11340.5  12892.6-52.0 4     B-47
11338.0  12830.84.0   2     B-38
11338.0  12830.8-8.8  2     B-38
11338.0  12830.8-23.0 1     B-38
11338.0  12830.8-26.5 3     B-38
11338.0  12830.8-36.5 5     B-38
11338.0  12830.8-50.0 4     B-38
11393.5  12948.93.5   2     B-60
11393.5  12948.9-3.8  2     B-60
11393.5  12948.9-23.0 1     B-60
```

```
11393.5  12948.9-27.0 3     B-60
11393.5  12948.9-39.0 5     B-60
11393.5  12948.9-52.0 4     B-60
11401.7  12897.84.0   2     B-45
11401.7  12897.8-2.0  2     B-45
11401.7  12897.8-22.0 1     B-45
11401.7  12897.8-27.5 3     B-45
11401.7  12897.8-37.5 5     B-45
11401.7  12897.8-51.0 4     B-45
11416.9  12819.52.5   2     B-44
11416.9  12819.5-5.0  2     B-44
11416.9  12819.5-21.0 1     B-44
11416.9  12819.5-28.5 3     B-44
11416.9  12819.5-38.5 5     B-44
11416.9  12819.5-51.0 4     B-44
11381.7  12747.51.5   2     B-33
11381.7  12747.5-4.0  2     B-33
11381.7  12747.5-21.5 1     B-33
11381.7  12747.5-25.8 3     B-33
11381.7  12747.5-42.8 5     B-33
11381.7  12747.5-56.0 4     B-33
11410.3  12724.70.5   2     B-43
11410.3  12724.7-4.5  2     B-43
11410.3  12724.7-22.9 1     B-43
11410.3  12724.7-25.0 3     B-43
11410.3  12724.7-36.0 5     B-43
11410.3  12724.7-49.0 4     B-43
11566.3  12850.62.5   2     B-30
11566.3  12850.6-5.0  2     B-30
11566.3  12850.6-21.0 1     B-30
11566.3  12850.6-28.5 3     B-30
11566.3  12850.6-38.5 5     B-30
11566.3  12850.6-51.0 4     B-30
11586.3  13050.611.5  2     B-31
11586.3  13050.61.0   2     B-31
11586.3  13050.6-11.0 1     B-31
11586.3  13050.6-18.5 3     B-31
11586.3  13050.6-26.5 5     B-31
11586.3  13050.6-47.0 4     B-31
11086.3  13090.68.5   2     B-32
11086.3  13090.6-1.0  2     B-32
11086.3  13090.6-14.0 1     B-32
11086.3  13090.6-23.5 3     B-32
```

```
11086.3  13090.6-32.5 5     B-32
11086.3  13090.6-48.0 4     B-32
```

PGF File Example with Depth

```
Easting     Northing     Depth     Lithology_ID     Boring_ID
    Ground
Depth  0|Overburden  1|Lava  2|Sulfide  3|Rhyolite
 4|Mafic_Intrusion    m
29
192731.10     1389503.04     0.00       0     1      2132.53
192731.10     1389503.04     6.75       0     1      2132.53
192731.10     1389503.04     101.00     1     1      2132.53
192731.10     1389503.04     383.10     3     1      2132.53
192731.10     1389503.04     403.70     2     1      2132.53
192731.10     1389503.04     490.00     4     1      2132.53
192674.55     1389639.67     0.00       0     22     2126.28
192674.55     1389639.67     4.30       0     22     2126.28
192674.55     1389639.67     100.60     1     22     2126.28
192674.55     1389639.67     156.00     3     22     2126.28
192674.55     1389639.67     209.40     2     22     2126.28
192674.55     1389639.67     496.20     4     22     2126.28
192987.12     1389624.87     0.00       0     13     2130.64
192987.12     1389624.87     6.98       0     13     2130.64
192987.12     1389624.87     91.40      1     13     2130.64
192987.12     1389624.87     397.40     2     13     2130.64
192987.12     1389624.87     425.80     4     13     2130.64
192930.95     1389745.48     0.00       0     14     2128.68
192930.95     1389745.48     6.70       0     14     2128.68
192930.95     1389745.48     80.40      1     14     2128.68
192930.95     1389745.48     246.40     3     14     2128.68
192930.95     1389745.48     250.60     2     14     2128.68
192930.95     1389745.48     459.60     4     14     2128.68
192582.47     1389677.63     0.00       0     23     2123.62
192582.47     1389677.63     6.80       0     23     2123.62
192582.47     1389677.63     101.20     1     23     2123.62
192582.47     1389677.63     138.70     3     23     2123.62
192582.47     1389677.63     160.00     2     23     2123.62
192582.47     1389677.63     499.60     4     23     2123.62
```

Borehole Geology (.geo) File Format

Discussion of Geology Files

Geology data files basically contain horizontal and vertical coordinates, which describe the geometry of geologic features of the region being modeled. The files must be in ASCII format and can be delimited by commas, spaces, or tabs. Borehole Geology files must have a .geo suffix to be selected in the file browsers of EVS modules. The z values in .geo files can represent either

elevation or depth, although elevation is generally the easiest to work with. When chemistry or property data is to be utilized along with geologic data for a 3-D visualization, a consistent coordinate system must be used in both sets of data.

Geology files should also specify the geologic layer material (color) number and layer names. This provides a mechanism to color multiple (not necessarily adjacent) layers as the same material.

Borehole Geology files (.geo suffix) must have the same number of entries for each boring location, so that every geologic layer in the system is represented in each boring. However, EVS allows flags to be included in the .geo files to allow automated processing of data in systems where geologic layers are not present in all locations (i.e., the layers "pinch out"). Also, EVS accommodates borings that were not extended deep enough to encounter layers that the scientist knows are present in the system. The use of these flags greatly facilitates the production of .geo data files, and minimizes the amount of manual interpretation the scientist must do before using EVS to analyze, understand, and refine a geologic model. For layers that pinch out, a flag of *pinch* can be used for automated estimation of the "depth" to the bottom of that layer. Entering this flag is essentially equivalent to entering the bottom depth of the layer directly above the pinched out layer (which is also an acceptable way to prepare the file). When EVS encounters this flag in a file, it assigns the pinched out layer a zero thickness at this location. For borings that do not extend to the depths of geologic layers in the system, a flag of *short* is included in the file for all layers below the depth of the boring. Including this flag notifies EVS to ignore the presence of this boring when kriging the surface of the layers below the total depth of the boring.

Format:

The file name must have a .geo suffix to be selected in the module's file browser. The format below is the same for all EVS modules which read geology files:

You may insert comment lines in C Tech Geology (.geo) files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file. There is an important exception. The first non-commented line of the file is the header line (line 1 described below).

Line 1: Any header message: Except that:

- $W or $G as the first two characters signifies a special geology file which contains unrelated surfaces such as historical water tables. These flags turn off checking for corrupt geology file formats (situations where lower surfaces are above higher surfaces) and automatically turn off kriging in thickness space.
- Line one cannot be BLANK

Line 2: **Elevation/Depth Specifier:**

- The only REQUIRED item on this line in the Elevation or Depth Specifier.
    - This line should contain the word *Elevation* or *Depth* (case insensitive) to denote whether sample elevations are true elevation or depth below ground surface.
    - If set to Depth all surface descriptions for layer bottoms are entered as depths relative to the top surface. This is a common means of collecting sample coordinates for borings.
    - Note that the flags such as pinch or short are not modified.
- Line 2 SHOULD contain names for each geologic surface (and therefore the layers created by them).
    - There are some rules that must be observed.
    - The number of surface (layer) names MUST be equal to the number of surfaces. Therefore, if naming layers, the first name should correspond to the top surface and each subsequent name will refer to the surface that defines the bottom of that layer.
    - A name containing a space MUST be enclosed in quotation marks example ("Silty Sand"). Names should be limited to upper and lower case letters, numerals, hyphen "-" and underscore "_". The names defined on line two will appear as the cell set name in the Explode_and_Scale or select_cells modules. Names should be separated with spaces, commas or tabs.
- Line 2: After the names, include the units of your coordinates (e.g. *feet* or *meters*). It must follow the names for each material number.

Line 3: The first integer (n) is the number of lines to follow. The second integer (m) is the number of geologic layer depths plus one (for the top surface). The 3rd and subsequent numbers are the colors for each surface in your model. Layers are colored by the color of the surface that defines their bottoms. The first two color numbers should be the same (top and bottom of the first layer).

When used with Fence_Geology, the order of the borings determines the connectivity of the fence diagram and must match the chemistry file for Krig_Fence.

Note that X and Y corresponding to Eastings and Northings are used. Be careful not to reverse these.

Line 4: First line of sample data. X, Y, top surface, and "m" depths or elevations to the bottom of each geologic layer. Coordinates, elevations (depths) and boring name can be separated by one comma and/or any number of spaces or tabs.

Two different flag parameters are included to accommodate special conditions. These flags are

A: Boring terminates early or surface information is missing. This flag class is used to identify that a boring did not continue deep enough to find the bottom of a geologic layer, OR that a section of

a core sample is missing (lost, damaged, etc.) and that no determination of the location of this surface can be made from this boring. This is distinctly different than a surface (layer) that is not present because it has been determined that it has pinched out. The flags that are used for this class are [note: all flags are case insensitive, but spelling is critical]:

- missing
- unknown
- unk
- na
- short
- terminated
- term

In the sample file below, BOR-24 was not deep enough to reach to the bottom of the Lsand (lower sand) layer or the gravel layer. Rather than use the bottom of the boring (a meaningless number), the short flag is used so that this boring will not be used to determine the bottom of these two layers. Similarly BOR-72 is not deep enough to be used in determining the bottom of the last (Gravel) layer. The flags that are used for this class are [note: all flags are case insensitive, but spelling is critical]:

B: This flag class is used to identify that a geologic layer is not present because it has pinched out for this particular boring. It can be "thought of" as numerically equivalent to using the value one column to the left. However, now that Krig_3D_Geology includes special treatment for the *pinch*flag, using the value to the left is not strictly equivalent.

- pinch
- pinched
- pinch-out

Note that several layers pinch out in borings WEL-67, BOR-23, BOR-70 and BOR-24, so the ***pinch*** flag was used for these layer's entries instead of any numerical value.

IMPORTANT: There are two important things to consider when using the flags above:

1. It is wholly inappropriate to have a pinch follow a short. Pinch denotes that the layer above is zero thickness. It is equivalent to using the numeric value to the left. However if it were to follow a short (unknown) it would be meaningless since the short is interpreted to be missing information.

2. If your last defined surface has fewer than 3 numeric values (with all the rest being missing/short), it will be poorly defined since it takes 3 points to define a plane. **If there are no numeric valuesthe surfacecannot be created.**

...

Line 3+n is the last line of the file.

AN EXAMPLE FILE FOLLOWS:

X	Y	TOP	BOT_1	BOT_2	BOT_3	BOT_4	BOT_5	BOT_6	BOT_7	Boring
Depth	Top	Fill	SiltySand	Clay	Sand	Silt	Sand	Gravel	feet	
7	8	5	5	3	1	4	2	4	6	
11856.7	12764.0	0	5.0	18.2	23.5	pinch	pinch	69.0	105.0	WEL-67
11889.6	12772.2	0	1.5	17.0	22.0	pinch	pinch	63.0	105.0	BOR-23
11939.1	12758.4	0	2.5	16.0	25.5	pinch	pinch	63.0	105.0	BOR-70
12002.8	12759.8	0	1.0	17.0	27.0	pinch	47.0	short	short	BOR-24
12085.1	12749.0	0	1.0	17.5	25.7	45.7	pinch	68.0	105.0	WEL-71
12146.7	12713.2	0	1.0	18.0	26.5	32.5	39.5	65.0	short	BOR-72
12199.7	12709.8	0	1.0	16.5	22.5	27.5	35.5	70.0	105.0	WEL-12

This file has 7 boreholes with 8 entries (surfaces) per borehole, corresponding to the top surface and the bottom depths of 7 geologic layers. Note that the fourth and sixth layers are both designated to be material 4. This allows you to easily create layers with the same material the same color.

Other Examples of Geologic Input Files

Example of a .geo file for sedimentary layers and lenses (containing pinchouts)

Example of a .geo file for Dipping Strata

Geologic_File_Example_Outcrop_of_Dipping_Strata

Borehole Geology (.geo) File Format

Discussion of Geology Files

Geology data files basically contain horizontal and vertical coordinates, which describe the geometry of geologic features of the region being modeled. The files must be in ASCII format and can be delimited by commas, spaces, or tabs. Borehole Geology files must have a .geo suffix to be selected in the file browsers of EVS modules. The z values in .geo files can represent either elevation or depth, although elevation is generally the easiest to work with. When chemistry or property data is to be utilized along with geologic data for a 3-D visualization, a consistent coordinate system must be used in both sets of data.

Geology files should also specify the geologic layer material (color) number and layer names. This provides a mechanism to color multiple (not necessarily adjacent) layers as the same material.

Borehole Geology files (.geo suffix) must have the same number of entries for each boring location, so that every geologic layer in the system is represented in each boring. However, EVS allows flags to be included in the .geo files to allow automated processing of data in systems where geologic layers are not present in all locations (i.e., the layers "pinch out"). Also, EVS accommodates borings that were not extended deep enough to encounter layers that the scientist knows are present in the system. The use of these flags greatly facilitates the production of .geo data files, and minimizes the amount of manual interpretation the scientist must do before using EVS to analyze, understand, and refine a geologic model. For layers that pinch out, a flag of *pinch* can be used for automated estimation of the "depth" to the bottom of that layer. Entering this flag is essentially equivalent to entering the bottom depth of the layer directly above the pinched out layer (which is also an acceptable way to prepare the file). When EVS encounters this flag in a file, it assigns the pinched out layer a zero thickness at this location. For borings that do not extend to the depths of geologic layers in the system, a flag of *short* is included in the file for all layers below the depth of the boring. Including this flag notifies EVS to ignore the presence of this boring when kriging the surface of the layers below the total depth of the boring.

Format:

The file name must have a .geo suffix to be selected in the module's file browser. The format below is the same for all EVS modules which read geology files:

You may insert comment lines in C Tech Geology (.geo) files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file. There is an important exception. The first non-commented line of the file is the header line (line 1 described below).

Line 1: Any header message: Except that:

- $W or $G as the first two characters signifies a special geology file which contains unrelated surfaces such as historical water tables. These flags turn off checking for corrupt geology file formats (situations where lower surfaces are above higher surfaces) and automatically turn off kriging in thickness space.
- Line one cannot be BLANK

Line 2: **Elevation/Depth Specifier:**

- The only REQUIRED item on this line in the Elevation or Depth Specifier.
    - This line should contain the word *Elevation* or *Depth* (case insensitive) to denote whether sample elevations are true elevation or depth below ground surface.

    - If set to Depth all surface descriptions for layer bottoms are entered as depths relative to the top surface. This is a common means of collecting sample coordinates for borings.
    - Note that the flags such as pinch or short are not modified.
- Line 2 SHOULD contain names for each geologic surface (and therefore the layers created by them).
    - There are some rules that must be observed.
    - The number of surface (layer) names MUST be equal to the number of surfaces. Therefore, if naming layers, the first name should correspond to the top surface and each subsequent name will refer to the surface that defines the bottom of that layer.
    - A name containing a space MUST be enclosed in quotation marks example ("Silty Sand"). Names should be limited to upper and lower case letters, numerals, hyphen "-" and underscore "_". The names defined on line two will appear as the cell set name in the Explode_and_Scale or select_cells modules. Names should be separated with spaces, commas or tabs.
- Line 2: After the names, include the units of your coordinates (e.g. *feet* or *meters*). It must follow the names for each material number.

Line 3: The first integer (n) is the number of lines to follow. The second integer (m) is the number of geologic layer depths plus one (for the top surface). The 3rd and subsequent numbers are the colors for each surface in your model. Layers are colored by the color of the surface that defines their bottoms. The first two color numbers should be the same (top and bottom of the first layer).

When used with Fence_Geology, the order of the borings determines the connectivity of the fence diagram and must match the chemistry file for Krig_Fence.

Note that X and Y corresponding to Eastings and Northings are used. Be careful not to reverse these.

Line 4: First line of sample data. X, Y, top surface, and "m" depths or elevations to the bottom of each geologic layer. Coordinates, elevations (depths) and boring name can be separated by one comma and/or any number of spaces or tabs.

Two different flag parameters are included to accommodate special conditions. These flags are

A: Boring terminates early or surface information is missing. This flag class is used to identify that a boring did not continue deep enough to find the bottom of a geologic layer, OR that a section of a core sample is missing (lost, damaged, etc.) and that no determination of the location of this surface can be made from this boring. This is distinctly different than a surface (layer) that is not present because it has been determined that it has pinched out.

The flags that are used for this class are [note: all flags are case insensitive, but spelling is critical]:

- missing
- unknown
- unk
- na
- short
- terminated
- term

In the sample file below, BOR-24 was not deep enough to reach to the bottom of the Lsand (lower sand) layer or the gravel layer. Rather than use the bottom of the boring (a meaningless number), the short flag is used so that this boring will not be used to determine the bottom of these two layers. Similarly BOR-72 is not deep enough to be used in determining the bottom of the last (Gravel) layer. The flags that are used for this class are [note: all flags are case insensitive, but spelling is critical]:

B: This flag class is used to identify that a geologic layer is not present because it has pinched out for this particular boring. It can be "thought of" as numerically equivalent to using the value one column to the left. However, now that Krig_3D_Geology includes special treatment for the *pinch*flag, using the value to the left is not strictly equivalent.

- pinch
- pinched
- pinch-out

Note that several layers pinch out in borings WEL-67, BOR-23, BOR-70 and BOR-24, so the ***pinch*** flag was used for these layer's entries instead of any numerical value.

IMPORTANT: There are two important things to consider when using the flags above:

1. It is wholly inappropriate to have a pinch follow a short. Pinch denotes that the layer above is zero thickness. It is equivalent to using the numeric value to the left. However if it were to follow a short (unknown) it would be meaningless since the short is interpreted to be missing information.
2. If your last defined surface has fewer than 3 numeric values (with all the rest being missing/short), it will be poorly defined since it takes 3 points to define a plane. **If there are no numeric valuesthe surfacecannot be created.**

...

Line 3+n is the last line of the file.

AN EXAMPLE FILE FOLLOWS:

X	Y	TOP	BOT_1	BOT_2	BOT_3	BOT_4	BOT_5	BOT_6	BOT_7	Boring
Depth	Top	Fill	SiltySand	Clay	Sand	Silt	Sand	Gravel	feet	
7	8	5	5	3	1	4	2	4	6	
11856.7	12764.0	0	5.0	18.2	23.5	pinch	pinch	69.0	105.0	WEL-67
11889.6	12772.2	0	1.5	17.0	22.0	pinch	pinch	63.0	105.0	BOR-23
11939.1	12758.4	0	2.5	16.0	25.5	pinch	pinch	63.0	105.0	BOR-70
12002.8	12759.8	0	1.0	17.0	27.0	pinch	47.0	short	short	BOR-24
12085.1	12749.0	0	1.0	17.5	25.7	45.7	pinch	68.0	105.0	WEL-71
12146.7	12713.2	0	1.0	18.0	26.5	32.5	39.5	65.0	short	BOR-72
12199.7	12709.8	0	1.0	16.5	22.5	27.5	35.5	70.0	105.0	WEL-12

This file has 7 boreholes with 8 entries (surfaces) per borehole, corresponding to the top surface and the bottom depths of 7 geologic layers. Note that the fourth and sixth layers are both designated to be material 4. This allows you to easily create layers with the same material the same color.

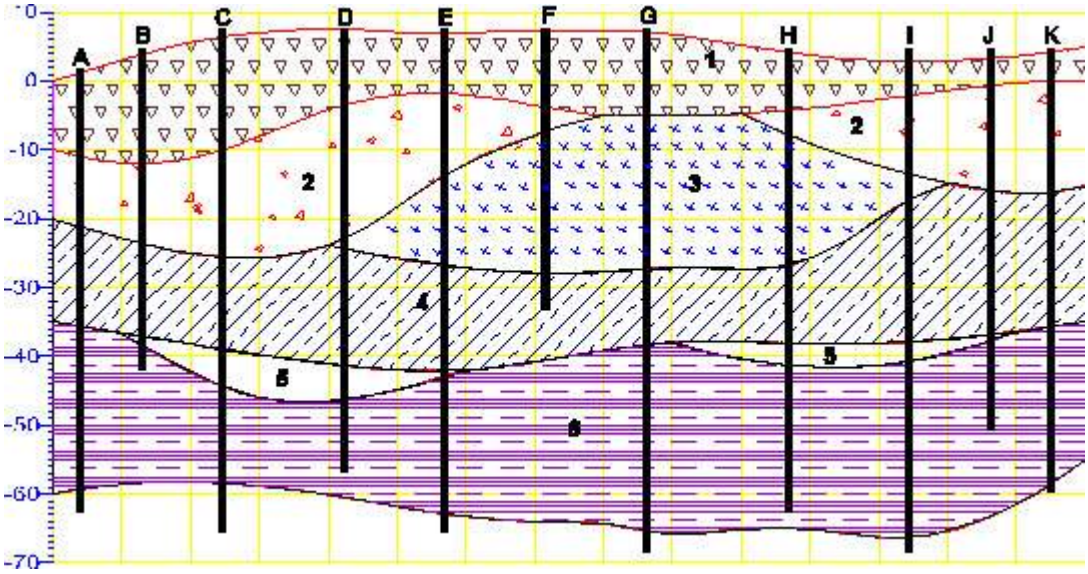
Other Examples of Geologic Input Files

Example of a .geo file for sedimentary layers and lenses (containing pinchouts)

Example of a .geo file for Dipping Strata

Geologic_File_Example_Outcrop_of_Dipping_Strata

Geologic File Example: Sedimentary Layers & Lenses



Both example files below represent valid forms for the geology file associated with the above figure. For file 1, line 2 of the file is "1", therefore all surface

elevations are entered as actual elevations relative to a fixed reference such as sea level (not depths) and the relationship between x, y, and elevation must be a right handed coordinate system. Note that X and Y corresponding to Eastings and Northings are used. Be careful not to reverse these.

Two special flags are used to accommodate special conditions. These flags are pinch and short. Pinch is used to identify that a geologic layer is not present (pinched out) for a particular boring. It is equivalent to using the value one column to the left. Short is used to identify that a boring did not extend to the bottom of a geologic layer. In the sample file below, boring C was not deep enough to reach to the bottom of the layer 3 or any subsequent layers. Rather than use the bottom of the boring (a meaningless number), the *short* flag is used so that this boring will not be used to determine the bottom of these layers.

File 1:

X	Y	TOP	BOT_1	BOT_2	BOT_3	BOT_4	BOT_5	BOT_6	NAME
Elev		Top	FILL	SH	SS	SD	SLS	GR	feet
11	7	1	1	2	3	4	5	6	
5	3	3	-11.5	-22	pinch	-36	pinch	-59	A
13	5	3.5	-12	-22.5	pinch	-36.8	-37.5	short	B
24	7	5	-11	-24	pinch	-38.5	-43	-58.6	C
42	2	8	-3	-22	-23	-41.5	-46	short	D
57	11	7	-2	-13	-26.5	-42	-43.5	-63	E
72	14	7	-3	-8	-27.6	short	short	short	F
85	19	5.7	-5	pinch	-26.6	-38.3	pinch	-65	G
107	23	4.2	-5	-8	-26	-38	-41	short	H
123	35	2.2	-3	-13	-16.9	-37.5	-41	-66	I
136	24	3	-1.5	-15	pinch	-37	-37.5	short	J
145	18	4	0	-15.7	pinch	-36.3	pinch	-58	K

For file 2 line 2 of the file is Depth", therefore all surface descriptions for layer bottoms are entered as depths relative to the top surface elevations. This is a common means of collecting sample coordinates for borings. Note that the flags (pinch and short) are not affected by using depths versus elevations.

File 2:

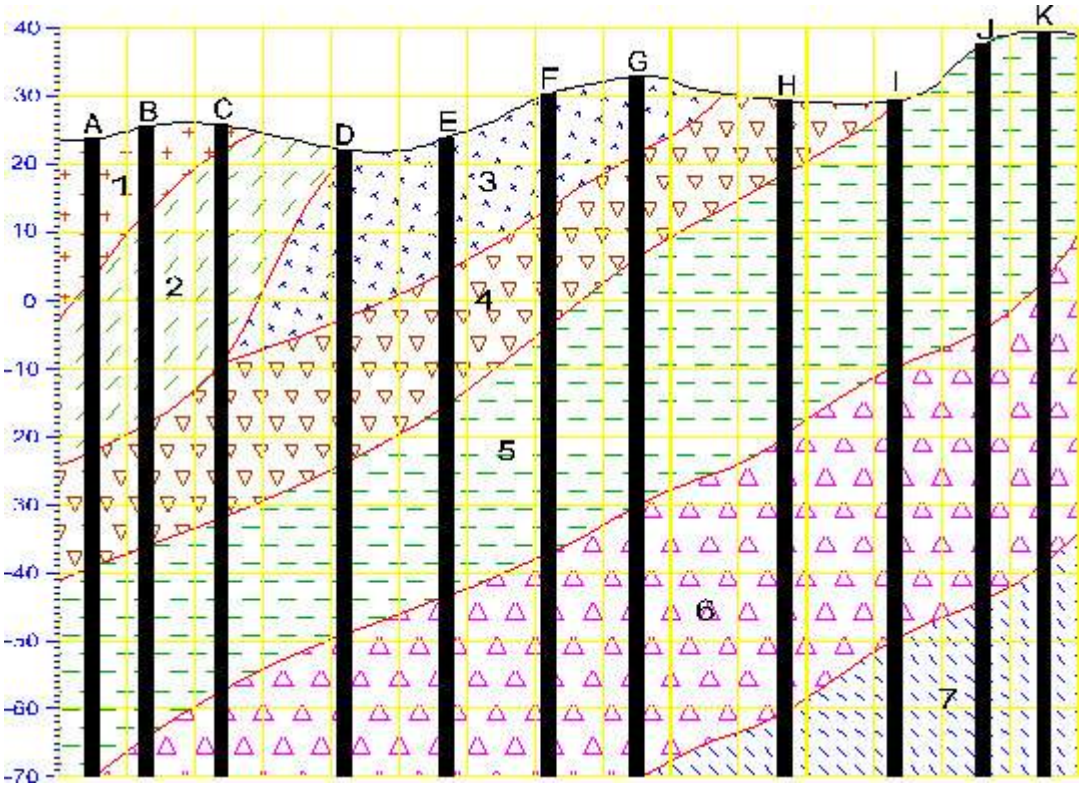
X	Y	TOP	BOT_1	BOT_2	BOT_3	BOT_4	BOT_5	BOT_6	NAME
Depth		Top	FILL	SH	SS	SD	SLS	GR	feet
11	7	1	1	2	3	4	5	6	
5	3	3	14.5	25	pinch	39	pinch	62	A

13	5	3.5	15.5	26	pinch	40.3	41	short	B
24	7	5	16	29	pinch	43.5	48	63.6	C
42	2	8	11	30	31	49.5	54	short	D
57	11	7	9	20	33.5	49	50.5	70	E
72	14	7	10	15	34.6	short	short	short	F
85	19	5.7	10.7	pinch	32.3	44	pinch	70.7	G
107	23	4.2	9.2	12.2	30.2	42.2	45.2	short	H
123	35	2.2	5.2	15.2	19.1	39.7	43.2	68.2	I
136	24	3	4.5	18	pinch	40	40.5	short	J
145	18	4	4	19.7	pinch	40.3	pinch	62	K

There is no numerical equivalent to using the short flag. It causes the kriging modules to select only those borings with valid data for computing the surfaces of each layer.

Geologic File Example: Outcrop of Dipping Strata

EVS is not limited to sedimentary layers or lenses. The figure below shows a cross-section through an outcrop of dipping geologic strata. EVS easily model the layers truncating on the top ground surface.



The file below represents the geology file associated with the above figure. Line 2 of the file is "Elevation", therefore all surface elevations are entered as

elevations (not depths) and the relationship between x, y, and elevation must be a right handed coordinate system. The *pinch* flag is used extensively to identify that a geologic layer is not present (pinched out) for a particular boring. It is equivalent to using the value one column to the left. The file was created with the assumption that there was no desire to model any layers below -70 foot elevation and that all borings extend to/beyond that depth.

Also, we have assigned the following material layer colors (numbers) to the 7 layers.

Layer # Material Abbreviation Material Color

1 Shale SH 5

2 Silty-sand SS 2

3 Sand SD 1

4 Sandy-silt SLS 3

5 Silty-sand SS 2

6 Sandy-silt SLS 3

7 Silt SL 4

X	Y	TOP	BOT_1	BOT_2	BOT_3	BOT_4	BOT_5	BOT_6	BOT_7	NAME
Elevation	Top	SH	SH	SS	SD	SLS	SS	SLS	SL	feet
44	8	5	5	2	1	3	2	3	4	
5	3	23.5	4	-22	pinch	-39	-70	-70	-70	A
13	5	26	13	-18	pinch	-36	-64	-70	-70	B
24	7	26	22	-9	-9.5	-32	-57.5	-70	-70	C
42	2	22	pinch	pinch	-3	-24	-50	-70	-70	D
57	6	24	pinch	pinch	4	-15	-43.5	-70	-70	E
72	7	30.5	pinch	pinch	14	-4	-37	-70	-70	F
85	3	33	pinch	pinch	21.5	6	-30	-70	-70	G
107	4	29.5	pinch	pinch	pinch	19	-20	-60	-70	H
123	6	29.5	pinch	pinch	pinch	28.5	-10	-49.5	-70	I
136	3	38	pinch	pinch	pinch	pinch	-4	-44	-70	J
145	0	39.5	pinch	pinch	pinch	pinch	-3	-39	-70	K
3.11	28.18	25.93	3.96	-20.99	pinch	-39.01	-70	-70	-70	A1
16.85	37.97	24.85	15.61	-20.7	pinch	-35.7	-61.92	-70	-70	B1
25.99	32.02	23.05	23.34	-6.11	-6.41	-31.53	-59.17	-70	-70	C1
41.05	25.13	24.26	pinch	pinch	-1.22	-25.57	-47.06	-70	-70	D1
54.43	34.94	26.56	pinch	pinch	1.36	-14.66	-45.49	-70	-70	E1
67.29	29.3	28.3	pinch	pinch	16.45	-6.49	-37.22	-70	-70	F1
88.89	25.31	32.92	pinch	pinch	19.17	6.16	-27.28	-70	-70	G1
104.17	30.58	30.13	pinch	pinch	pinch	19.76	-22.25	-62.18	-70	H1
121.87	30.26	30.76	pinch	pinch	pinch	27.84	-7.81	-49.67	-70	I1

136.99	29.61	35.95	pinch	pinch	pinch	pinch	-6.02	-44.8	-70	J1
149.67	29.33	37.59	pinch	pinch	pinch	pinch	-4.09	-40.17	-70	K1
4.06	62.03	23.47	3.46	-22.43	pinch	-38.05	-70	-70	-70	A2
12.09	64.15	25.26	13.42	-19.11	pinch	-33.89	-59.06	-70	-70	B2
30.73	66.42	25.81	26.1	-3.46	-3.76	-28.81	-58.62	-70	-70	C2
40.43	49.79	26.12	pinch	pinch	-0.5	-27.73	-46.67	-70	-70	D2
54.5	65.51	27.88	pinch	pinch	1.79	-15.51	-43.8	-70	-70	E2
66.41	52.9	25.48	pinch	pinch	16.96	-7.18	-35.22	-70	-70	F2
93.58	50.18	34.29	pinch	pinch	21.62	6.46	-28.76	-70	-70	G2
106.13	55.44	30.39	pinch	pinch	pinch	20.9	-23.47	-60.65	-70	H2
126.19	63.43	28.78	pinch	pinch	pinch	27.64	-8.31	-48.85	-70	I2
138.39	62.4	36.52	pinch	pinch	pinch	pinch	-5.72	-47.12	-70	J2
144.91	52.79	40.49	pinch	pinch	pinch	pinch	-4.66	-37.23	-70	K2
6.77	86.15	21.09	2.83	-22.62	pinch	-36.05	-70	-70	-70	A3
16.91	98.53	22.86	10.95	-17.19	pinch	-31.33	-57.46	-70	-70	B3
35.07	87.05	25.39	25.81	-2.37	-2.67	-30.89	-59.85	-70	-70	C3
36.37	77.38	26.62	pinch	pinch	-2.19	-27.56	-43.87	-70	-70	D3
51.5	94.86	27.26	pinch	pinch	4.57	-15.51	-46.35	-70	-70	E3
71.23	73.19	26.45	pinch	pinch	16.19	-9.22	-38.04	-70	-70	F3
93.09	79.15	33.93	pinch	pinch	19.64	9.37	-28.16	-70	-70	G3
110.18	76.02	27.4	pinch	pinch	pinch	20.63	-21.81	-63.39	-70	H3
127.9	90.62	31.64	pinch	pinch	pinch	29.56	-8.26	-45.96	-70	I3
139.27	96.26	37.57	pinch	pinch	pinch	pinch	-8.29	-47.67	-70	J3
143.52	75.62	38.22	pinch	pinch	pinch	pinch	-6.59	-37.51	-70	K3

Geology Files for Production of a Fence Diagram

Discussion of Geology Files for Fence Sections

Files used to create fence diagrams contain only those borings that the user wishes to include on an individual cross section of the fence, in the order that they will be connected along the section. The resulting set of files includes one .geo file for each cross section that will be included in a fence diagram. The order of the boring listings determines the connectivity of the fence diagram, and must match the order of the borings in the associated chemistry file when chemistry is to be displayed on the diagram. The data for the boring(s) at which individual sections will be joined to produce the fence diagram are included in each of the cross section files that will intersect. Generally, it is easiest to create the geology file for the complete 3-D dataset, and then cut and paste the individual section files from the complete file. Examples of a 3-D geology file and a typical set of fence diagram files are presented below.

The format of the data in the file is exactly the same as for 3-D geology files. Material colors are not supported for fence diagrams.

An example set of files for producing a fence diagram with two merged cross sections are shown below:

```
Geology File for Cross Section A-A'

Elevation feet

7         8

11086.52  12830.67  2500  2496  2484  2479  2420  2417.5  2415  2395  BOR-49
11199.04  12810.16  2501  2492  2482  2473  2420  2414.5  2409  2397  BOR-51
11259.67  12819.29  2502  2492  2479  2467  2425  2419.5  2414  2399  BOR-46
11298     12808.63  2503  2492  2492  2480  2424  2413.5  2403  2392  BOR-52
11414.4   12781.1   2504  2491  2482  2471  2420  2416.3  2412  2396  BOR-34
11427     12780.9   2501  2493  2477  2467  2424  2415.0  2406  2397  BOR-42
11496.34  12753.59  2502  2492  2480  2465  2422  2416.5  2411  2400  BOR-53


Geology File for Cross Section B B'

Elevation feet

5         8

11209.35  12993.94  2502  2492  2481  2462  2423  2415  2410  2400  BOR 57
11251.30  12929.27  2503  2493  2474  2465  2422  2414  2406  2397  BOR 75
11248.75  12870.91  2501  2492  2483  2472  2421  2416  2411  2396  BOR 48
11199.04  12810.16  2501  2492  2482  2473  2420  2414  2409  2397  BOR 51
11211.87  12710.75  2503  2493  2480  2468  2422  2420  2415  2399  BOR 50
```

This example fence diagram contains two cross sections, with elevations for the surface and the bottoms of seven layers of geology in each. Section A-A' has seven borings that will be used to define it, and Section B-B' has five borings. Neither of the sections contains layers that pinch out, and all of the borings extend to the depth of the fence. Note that the entries for location BOR-51 are identical in each file, and are placed such that the sections will cross at the second location in the A-A' file, and the fourth location in the B-B' file. The user will typically use a basemap to plan the orientations and intersections of the fences. EVS does not impose any restrictions on the number of borings in or placement of sections in fence diagrams, but planning should be done to assure that most sections of the fence can be viewed from a chosen viewpoint.

Geology Multi-File

Geology Multi-Files: Unlike the .geo file format, the .gmf format is not based on boring observations with common x,y coordinates. The multi-file format allows for description of individual geologic surfaces by defining a set of x,y,z coordinates (separated by spaces, tabs, and/or commas). Geologic hierarchy still applies for definition of complex geologic structures.

This file format allows for creation of geologic models when the data available for the top surface and one or more of the subsurface layers are uncorrelated (in number or x,y location). For example, a gmf file may contain

1000 x,y,z measurements for the ground surface, but only 12 x,y,z measurements for other lithologic surfaces. This format also allows for specification of the geologic material color (layer material number).

You SHOULD include the units of your coordinates (e.g. *feet* or *meters*). If this is included it must be on a line following the word *units*.

Note: there are no special flags (e.g. short, pinch, etc.) used in GMF files. Since each surface stands on its own (does not refer to a prior surface) pinched-out layers are accomplished by duplicating the elevations (x,y,z points) on two consecutive surfaces. The "short" flags are not needed since those points are merely excluded from a surface's definition.

The name for a surface can be a date or date & time if the data represents surface points at different times (e.g. changing groundwater elevations. The date format is dependent on your REGIONAL SETTINGS on your computer (control panel).

C Tech uses the SHORT DATE and SHORT TIME formats.

If the date/time works in Excel it will likely work in EVS.

For most people in the U.S., this would not be 24 hour clock so you would need:

"m/d/yyyy hh:mm:ss AM" or "m/d/yyyy hh:mm:ss PM"

Also, you MUST put the date/time in quotes if you use more than just date (i.e. if there are spaces in the total date/time).

Format: The following is a geology multi-file which is included with EVS. This file begins with the line starting with a "#".

```
# Lines beginning with a "#" character are comments.
# Each geologic surface begins with a line: surface x
# The number after surface is the layer material color number.
# Each surface can have different x,y coords and number of points
units ft
surface 2 Top
11086.5 12830.7 4.5
11199.0 12810.2 4
# Comment lines can be placed anywhere in a multi-file
11259.7 12819.3 2
11298.0 12808.6 3
11414.4 12781.1 2
11427.0 12780.9 6.5
11496.3 12753.6 1.5
11209.4 12993.9 2
11251.3 12929.3 2
11248.8 12870.9 3
11211.9 12710.8 2
11302.0 13079.7 4.5
```

```
11286.8 13026.7 2
11309.0 12949.0 4
11340.5 12892.6 2.5
11338.0 12830.8 4
11393.5 12948.9 3.5
11401.7 12897.8 4
11416.9 12819.5 2.5
11381.7 12747.5 1.5
11410.3 12724.7 0.5
11566.3 12850.6 2.5
11586.3 13050.6 11.5
11086.3 13090.6 8.5
surface 2 Fill
11086.5 12830.7 -3.8
11199.0 12810.2 -5
11259.7 12819.3 -7.5
11298.0 12808.6 -6
11414.4 12781.1 -6
11427.0 12780.9 -7
11496.3 12753.6 -7.5
11209.4 12993.9 -3
11251.3 12929.3 -2.5
11248.8 12870.9 -3.5
11211.9 12710.8 -6.5
11302.0 13079.7 -3.5
11286.8 13026.7 -5
11309.0 12949.0 -2.5
11340.5 12892.6 -2.5
11338.0 12830.8 -8.8
11393.5 12948.9 -3.8
11401.7 12897.8 -2
11416.9 12819.5 -5
11381.7 12747.5 -4
11410.3 12724.7 -4.5
11566.3 12850.6 -5
11586.3 13050.6 1
11086.3 13090.6 -1
surface 1 Silt
11086.5 12830.7 -21
11199.0 12810.2 -20
11259.7 12819.3 -20.5
11298.0 12808.6 -19
```

```
11414.4 12781.1 -20.5
11427.0 12780.9 -23
11496.3 12753.6 -20
11209.4 12993.9 -23
11251.3 12929.3 -22
11248.8 12870.9 -22
11211.9 12710.8 -22.5
11302.0 13079.7 -21.9
11286.8 13026.7 -23
11309.0 12949.0 -22
11340.5 12892.6 -20
11338.0 12830.8 -23
11393.5 12948.9 -23
11401.7 12897.8 -22
11416.9 12819.5 -21
11381.7 12747.5 -21.5
11410.3 12724.7 -22.9
11566.3 12850.6 -21
11586.3 13050.6 -11
11086.3 13090.6 -14
surface 3 Clay
11086.5 12830.7 -26
11199.0 12810.2 -25
11259.7 12819.3 -27
11298.0 12808.6 -25.8
11414.4 12781.1 -28
11427.0 12780.9 -28.5
11496.3 12753.6 -28.8
11209.4 12993.9 -27.5
11251.3 12929.3 -28
11248.8 12870.9 -28.5
11211.9 12710.8 -27.5
11302.0 13079.7 -26
11286.8 13026.7 -29
11309.0 12949.0 -28.3
11340.5 12892.6 -23
11338.0 12830.8 -26.5
11393.5 12948.9 -27
11401.7 12897.8 -27.5
11416.9 12819.5 -28.5
11381.7 12747.5 -25.8
11410.3 12724.7 -25
```

```
11566.3 12850.6 -28.5
11586.3 13050.6 -18.5
11086.3 13090.6 -23.5
surface 5 Gravel
11086.5 12830.7 -42
11199.0 12810.2 -39
11259.7 12819.3 -40
11298.0 12808.6 -41.8
11414.4 12781.1 -42
11427.0 12780.9 -38.5
11496.3 12753.6 -38.8
11209.4 12993.9 -37.5
11251.3 12929.3 -40
11248.8 12870.9 -36.3
11211.9 12710.8 -37.5
11302.0 13079.7 -38
11286.8 13026.7 -37
11309.0 12949.0 -38.3
11340.5 12892.6 -38
11338.0 12830.8 -36.5
11393.5 12948.9 -39
11401.7 12897.8 -37.5
11416.9 12819.5 -38.5
11381.7 12747.5 -42.8
11410.3 12724.7 -36
11566.3 12850.6 -38.5
11586.3 13050.6 -26.5
11086.3 13090.6 -32.5
surface 4 Sand
11086.5 12830.7 -55
11199.0 12810.2 -53
11259.7 12819.3 -53
11298.0 12808.6 -55
11414.4 12781.1 -55
11427.0 12780.9 -51
11496.3 12753.6 -51
11209.4 12993.9 -51
11251.3 12929.3 -53
11248.8 12870.9 -50
11211.9 12710.8 -51
11302.0 13079.7 -51
11286.8 13026.7 -50
```

```
11309.0 12949.0 -52
11340.5 12892.6 -52
11338.0 12830.8 -50
11393.5 12948.9 -52
11401.7 12897.8 -51
11416.9 12819.5 -51
11381.7 12747.5 -56
11410.3 12724.7 -49
11566.3 12850.6 -51
11586.3 13050.6 -47
11086.3 13090.6 -48
end
```

Geology Multi-File

Geology Multi-Files: Unlike the .geo file format, the .gmf format is not based on boring observations with common x,y coordinates. The multi-file format allows for description of individual geologic surfaces by defining a set of x,y,z coordinates (separated by spaces, tabs, and/or commas). Geologic hierarchy still applies for definition of complex geologic structures.

This file format allows for creation of geologic models when the data available for the top surface and one or more of the subsurface layers are uncorrelated (in number or x,y location). For example, a gmf file may contain 1000 x,y,z measurements for the ground surface, but only 12 x,y,z measurements for other lithologic surfaces. This format also allows for specification of the geologic material color (layer material number).

You SHOULD include the units of your coordinates (e.g. *feet* or *meters*). If this is included it must be on a line following the word *units*.

Note: there are no special flags (e.g. short, pinch, etc.) used in GMF files. Since each surface stands on its own (does not refer to a prior surface) pinched-out layers are accomplished by duplicating the elevations (x,y,z points) on two consecutive surfaces. The "short" flags are not needed since those points are merely excluded from a surface's definition.

The name for a surface can be a date or date & time if the data represents surface points at different times (e.g. changing groundwater elevations. The date format is dependent on your REGIONAL SETTINGS on your computer (control panel).

C Tech uses the SHORT DATE and SHORT TIME formats.

If the date/time works in Excel it will likely work in EVS.

For most people in the U.S., this would not be 24 hour clock so you would need:

"m/d/yyyy hh:mm:ss AM" or "m/d/yyyy hh:mm:ss PM"

Also, you MUST put the date/time in quotes if you use more than just date (i.e. if there are spaces in the total date/time).

Format: The following is a geology multi-file which is included with EVS. This file begins with the line starting with a "#".

```
# Lines beginning with a "#" character are comments.
# Each geologic surface begins with a line: surface x
# The number after surface is the layer material color number.
# Each surface can have different x,y coords and number of points
units ft
surface 2 Top
11086.5 12830.7 4.5
11199.0 12810.2 4
# Comment lines can be placed anywhere in a multi-file
11259.7 12819.3 2
11298.0 12808.6 3
11414.4 12781.1 2
11427.0 12780.9 6.5
11496.3 12753.6 1.5
11209.4 12993.9 2
11251.3 12929.3 2
11248.8 12870.9 3
11211.9 12710.8 2
11302.0 13079.7 4.5
11286.8 13026.7 2
11309.0 12949.0 4
11340.5 12892.6 2.5
11338.0 12830.8 4
11393.5 12948.9 3.5
11401.7 12897.8 4
11416.9 12819.5 2.5
11381.7 12747.5 1.5
11410.3 12724.7 0.5
11566.3 12850.6 2.5
11586.3 13050.6 11.5
11086.3 13090.6 8.5
surface 2 Fill
11086.5 12830.7 -3.8
11199.0 12810.2 -5
11259.7 12819.3 -7.5
11298.0 12808.6 -6
11414.4 12781.1 -6
11427.0 12780.9 -7
11496.3 12753.6 -7.5
11209.4 12993.9 -3
11251.3 12929.3 -2.5
```

```
11248.8 12870.9 -3.5
11211.9 12710.8 -6.5
11302.0 13079.7 -3.5
11286.8 13026.7 -5
11309.0 12949.0 -2.5
11340.5 12892.6 -2.5
11338.0 12830.8 -8.8
11393.5 12948.9 -3.8
11401.7 12897.8 -2
11416.9 12819.5 -5
11381.7 12747.5 -4
11410.3 12724.7 -4.5
11566.3 12850.6 -5
11586.3 13050.6 1
11086.3 13090.6 -1
surface 1 Silt
11086.5 12830.7 -21
11199.0 12810.2 -20
11259.7 12819.3 -20.5
11298.0 12808.6 -19
11414.4 12781.1 -20.5
11427.0 12780.9 -23
11496.3 12753.6 -20
11209.4 12993.9 -23
11251.3 12929.3 -22
11248.8 12870.9 -22
11211.9 12710.8 -22.5
11302.0 13079.7 -21.9
11286.8 13026.7 -23
11309.0 12949.0 -22
11340.5 12892.6 -20
11338.0 12830.8 -23
11393.5 12948.9 -23
11401.7 12897.8 -22
11416.9 12819.5 -21
11381.7 12747.5 -21.5
11410.3 12724.7 -22.9
11566.3 12850.6 -21
11586.3 13050.6 -11
11086.3 13090.6 -14
surface 3 Clay
11086.5 12830.7 -26
```

```
11199.0 12810.2 -25
11259.7 12819.3 -27
11298.0 12808.6 -25.8
11414.4 12781.1 -28
11427.0 12780.9 -28.5
11496.3 12753.6 -28.8
11209.4 12993.9 -27.5
11251.3 12929.3 -28
11248.8 12870.9 -28.5
11211.9 12710.8 -27.5
11302.0 13079.7 -26
11286.8 13026.7 -29
11309.0 12949.0 -28.3
11340.5 12892.6 -23
11338.0 12830.8 -26.5
11393.5 12948.9 -27
11401.7 12897.8 -27.5
11416.9 12819.5 -28.5
11381.7 12747.5 -25.8
11410.3 12724.7 -25
11566.3 12850.6 -28.5
11586.3 13050.6 -18.5
11086.3 13090.6 -23.5
surface 5 Gravel
11086.5 12830.7 -42
11199.0 12810.2 -39
11259.7 12819.3 -40
11298.0 12808.6 -41.8
11414.4 12781.1 -42
11427.0 12780.9 -38.5
11496.3 12753.6 -38.8
11209.4 12993.9 -37.5
11251.3 12929.3 -40
11248.8 12870.9 -36.3
11211.9 12710.8 -37.5
11302.0 13079.7 -38
11286.8 13026.7 -37
11309.0 12949.0 -38.3
11340.5 12892.6 -38
11338.0 12830.8 -36.5
11393.5 12948.9 -39
11401.7 12897.8 -37.5
```

```
11416.9 12819.5 -38.5
11381.7 12747.5 -42.8
11410.3 12724.7 -36
11566.3 12850.6 -38.5
11586.3 13050.6 -26.5
11086.3 13090.6 -32.5
surface 4 Sand
11086.5 12830.7 -55
11199.0 12810.2 -53
11259.7 12819.3 -53
11298.0 12808.6 -55
11414.4 12781.1 -55
11427.0 12780.9 -51
11496.3 12753.6 -51
11209.4 12993.9 -51
11251.3 12929.3 -53
11248.8 12870.9 -50
11211.9 12710.8 -51
11302.0 13079.7 -51
11286.8 13026.7 -50
11309.0 12949.0 -52
11340.5 12892.6 -52
11338.0 12830.8 -50
11393.5 12948.9 -52
11401.7 12897.8 -51
11416.9 12819.5 -51
11381.7 12747.5 -56
11410.3 12724.7 -49
11566.3 12850.6 -51
11586.3 13050.6 -47
11086.3 13090.6 -48
end
```

ctech_example.gmf

```
# Database Generated GMF File (Creation at 7/22/2003 5:36:07 PM)
#
#
# Surface 1: 25 Coordinates
# Database Columns [GMF_Surface0 (Ground Surface)]: X, Y, Top
surface 1 Sand
11566.34 12850.59 2.5
11586.34 13050.59 11.5
```

```
11086.3 13090.6 8.5
      .
      .
      .
      .
11393.47 12948.9 3.5
11251.3 12929.27 2
# Surface 1 Complete
#
# Surface 2: 24 Coordinates (Added at 7/22/2003 5:37:04 PM)
# Database Columns [GMF_Surface1]: X, Y, Z
surface 1 Sand
11566.34 12850.59 -5
11586.34 13050.59 1
11086.3 13090.6 -1
      .
      .
      .
      .
11393.47 12948.9 -3.8
11251.3 12929.27 -2.5
# Surface 2 Complete
#
# Surface 3: 24 Coordinates (Added at 7/22/2003 5:38:18 PM)
# Database Columns [GMF_Surface2]: X, Y, Z
surface 1 Sand
11566.34 12850.59 -21
11586.34 13050.59 -11
11086.3 13090.6 -14
      .
      .
      .
11393.47 12948.9 -23
11251.3 12929.27 -22
# Surface 3 Complete
#
units ft
end
# Database Generated GMF File (Finalization at 7/22/2003 5:39:06
PM)
```

EMT File Format

The EMT (EVS Multi-Text) format is used to place 3D text (labels) with user adjustable font and alignment.

The format is:

- Lines beginning with "#" are comments
- Lines beginning with "FONT" are font specification lines (more later)
- Lines beginning with "END" specify the end of the file (this is optional, but if you want to have anything after the last command or data line, precede it with an "END" statement.
- All other lines are DATA lines specifying the x-y-z coordinates of a string and the text for that string.
- Blank lines are ignored.
- The FONT specification lines contain the following information **in this order**:
  - Size: The font size is the height of a typical Capitol letter in true user units
  - Justification: The justification options are the same as in post_samples
  - Plane: The plane options are the same as in post_samples
  - Orientation: The orientation options are the same as in post_samples
  - Red, Green, Blue: These 3 numbers determine the font color.
  - Resolution: The resolution parameter is the same as in post_samples
  - Depth: The parameter is the same as in post_samples
  - Bevel%: The Bevel percentage isthe same as in post_samples
  - Font Face: The Font Face options are the same as in post_samples
- The DATA lines contain four columns of information:

  1. X coordinate
  2. Y coordinate
  3. Z coordinate
  4. Text: Everything on the line after the z coordinate (and trailing spaces) is the text to be placed at the above coordinate.

Below is an example EMT File

```
# FONT Size Just. Plane Orient R G B Resolution Depth Bevel% Font
Face
```

```
FONT, 4, MC, XZ, +X, 0.8, 0.8, 0.8, 3, 0, 0, Arial
# X, Y, Z, Bore
11566.34, 12850.59, 8.5, B-30
11586.34, 13050.59, 12.5, B-31
11381.7, 12747.5, 2.5, B-33
11414.4, 12781.1, 3, B-34
11410.29, 12724.69, 4.5, B-4
11427, 12780.9, 7.5, B-42
11086.52, 12830.67, 5.5, B-49
11211.87, 12710.75, 3, B-50
11199.04, 12810.16, 5, B-51
11496.34, 12753.59, 2.5, B-53
11209.35, 12993.94, 3, B-57
11301.97, 13079.66, 5.5, B-58
11286.77, 13026.7, 3, B-59

# FONT Size Just. Plane Orient R G B Resolution Depth Bevel% Font
Face
FONT, 6, MC, XZ, +X, 1, 0.5, 0.5, 3, 0.1, 0, Arial
11393.47, 12948.9, 4.5, B-60
11309.03, 12948.99, 5, B-56
11248.75, 12870.91, 4, B-48
11259.67, 12819.29, 3, B-46
11298, 12808.63, 4, B-52
11338, 12830.8, 5, B-38
11401.73, 12897.77, 5, B-45
11416.9, 12819.45, 3.5, B-44

# FONT, Size, Justification, Plane, Orientation, Red, Green,
Blue, Resolution, Depth, Bevel%, Font Face
FONT, 8, MC, XZ, +X, 1, 0, 0, 3, .3, 0, Arial Bold
11340.49, 12892.61, 3.5, B-47
11251.3, 12929.27, 3, B-75
END
```

4D Interactive Model Animation Player

- Animation Using 4D Interactive Models
- Open 4D File
- 4D Interactive Model Player
- File Menu Options
- Transformations with the Mouse
- Azimuth and Elevation Panel

4D Interactive Model Animation Player

Animation Using 4D Interactive Models

MVSand EVS-PRO have two distinctly different animation concepts. Our traditional animations consist of a sequence of bitmap images that have been encoded into an animation file (AVI, MPG, or HAV) using one of several different CODECS. This type of animation has some distinct advantages including the ability to include a soundtrack (background music and/or narration) and the animations are playable with a variety of common and (usually) free players.

C Tech also offers a second animation concept that we refer to as 4DIM, or Four-Dimensional Interactive Model Animations. The name 4DIM (referring to four dimensions instead of only 3) was chosen because these models represent a 3D scene changing in time (the fourth dimension). Unlike bitmap (image) based animations 4DIMs contain a **complete 3D model** at each frame of the animation. Each frame can be thought of as a VRML model (*though it is not*) and has similar functionality. Each frame of the model can be zoomed, panned and rotated as a static 3D model **or** you can interact with the 4DIM animation as it is playing.

The 4DIM Player is offered as a module (Playback_4DIM) in all versions of our software and as a stand-alone application.

The 4DIM files (which have a .4d suffix) created by EVS-PRO must be played back in a licensed copy of C Tech software or a licensed 4DIM Player (standalone). 4DIM Files created in MVS automatically contain a password that allows them to be played with a demo player. In other words, **4DIM files created in MVS can be played on free players.**

Open 4D File

Let's open a 4DIM model file and interact with the 4DIM Player. The file we'll choose is an extremely simple 4DIM file that has 35 frames. Though it is a very simplistic example, it is also an amazingly small file that is only 30 Kbytes (yes Kbytesnot Mbytes!).

To begin, run the stand-alone player or instance the Playback_4DIM module from the File sublibrary. The player window appears in the upper left corner of your desktop.

From the 4D Interactive Model Player window, choose the File pull-down menu and select *Open*. The default folder should be ctech\data\4dim (if you installed to ctech). From this folder select the file thermal.4d

It is now time to use the 4D Interactive Model Player.

NOTE:IN THE STANDALONE 4DIM PLAYER, you may also open 4D files from Windows Explorer or a browser (e.g. Internet Explorer or Netscape) by clicking (or double clicking) on the files. Furthermore, if you have an open file and choose another in this manner, it will open in the current 4DIM player. This also applies to A4D files discussed in a few more topics.

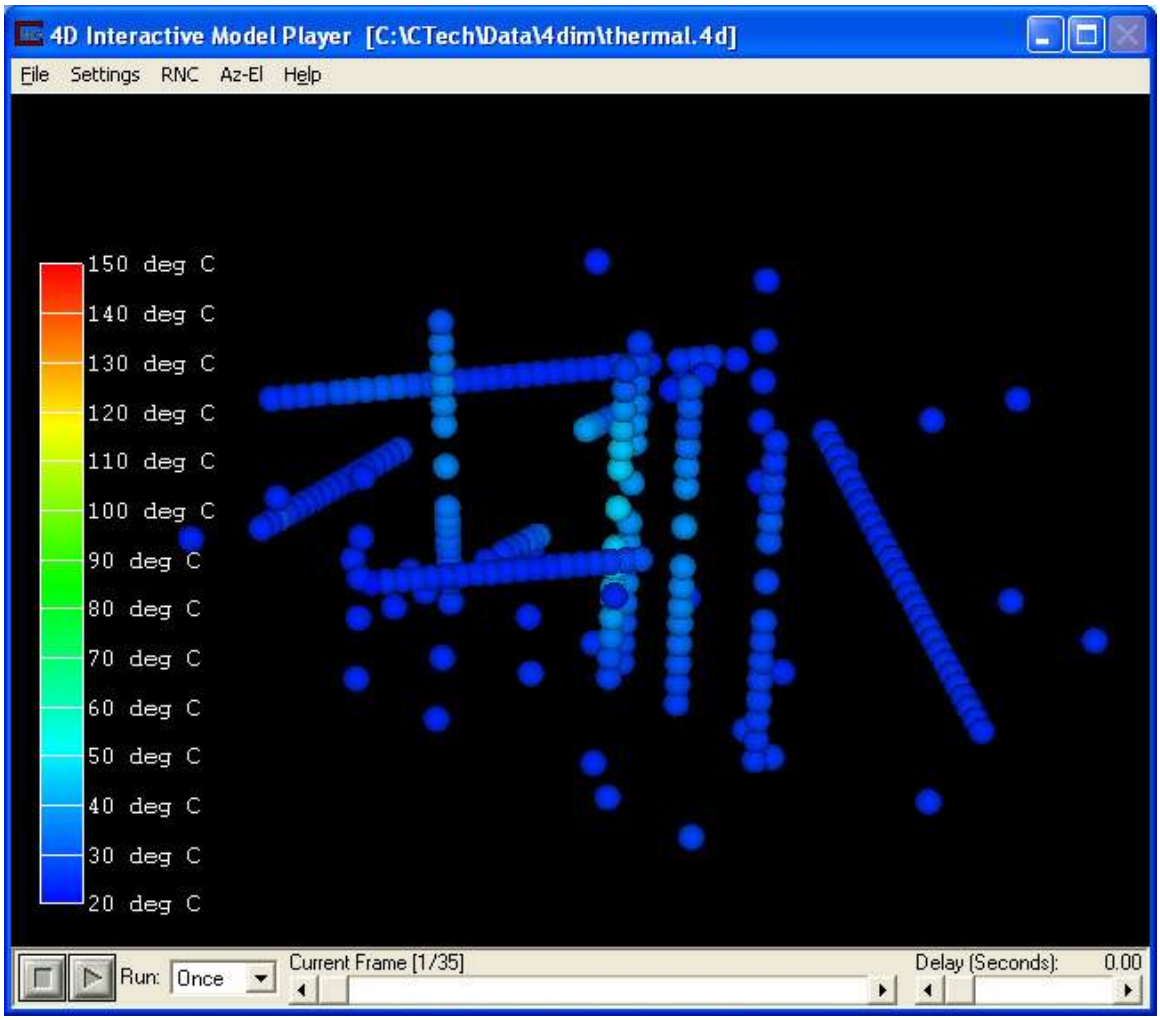
4D Interactive Model Player

The module for the player integrated in C Tech software is:

The Playback _4DIM module and the stand-alone 4DIM Player are identical in functionality with very minor exceptions. The players include an integrated C

Tech Viewer with many of its features. Mouse interactions are nearly identical to the Viewer's. The bottom of the player window includes script controls and more.

After loading the file specified in the previous topic, the player's window should look like:



Shortcuts

There are a few keyboard shortcuts worth noting. These will give you quick control over the player.

- CTRL-F sets the player to FULL SCREEN mode. This is not equivalent to the maximize button in the upper right corner since this removes the normal borders.
- With the player as the active window, ESC(ape) exits Full Screen mode
- CTRL-H toggles the Auto-Hide mode.

The VCR type buttons on the player panel allows you to run or pause the selected scipt. The function of each button is:

 **Stop**: Stop script from playing.

 **Pause**: Pause script.

 **Play**: The play button will open the Choose 4DIM View window. This window selects the current script to be played by the 4DIM player. There are standard scripts that can be played with every 4DIM or complex scripts that can be created and bundled for specific 4DIM's. If a 4DIM has been paused while playing a script this button will instead resume the playing of the script.

Start by hitting the Play button.Select "Play Each Frame" from the Standard Script Library and then select the "Play Script" button.Notice that the animation plays from the beginning to the end only one time. There are options to change this behavior.

The **Run** option menu allows for specifying three different ways of playing the animation:

Once displays the script one scene at a time.

Cycle will display the script one scene at a time, when the script is complete it will repeat this process.

Bounce also displays the script one scene at a time, however, when the script is complete it will then play the scenes in the reverse order. It will continue to bounce until stopped.

Change to the **Cycle** option and Play again. Note now that the animation runs until you Pause or Stop. In this mode it runs from start to finish and then resets to the beginning and runs again.

Change to the **Bounce** option and Play again. In this mode it plays from start to finish and then from finish to start and so on.

The **Current Frame** slider provides a visual display of the current frame, and more importantly provides a direct method of examining each frame. The slider button (moving square) can be dragged forward or backward to advance or retreat the current frame in real time.

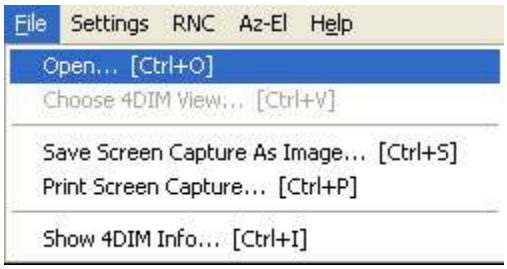
Now let's control the animation using the slider. You should have noticed that it moved when you play with any of the modes above. Press Stop (or Pause) and "click and hold" on the square moving button on the slider. As you move the slider, the animation responds by displaying the frame that you specify by your motion.

You can use the **Delay** slider to slow down the replay. The slider controls the amount of time between frames in seconds. You can set the amount of time from one-hundredth of a second to two seconds.

Now that you know how to play the animation, the real fun begins.

File Menu Options

The file menu for the Playback 4DIM module and the Standalone 4DIM Player have the following options:



Open 4DIM file.

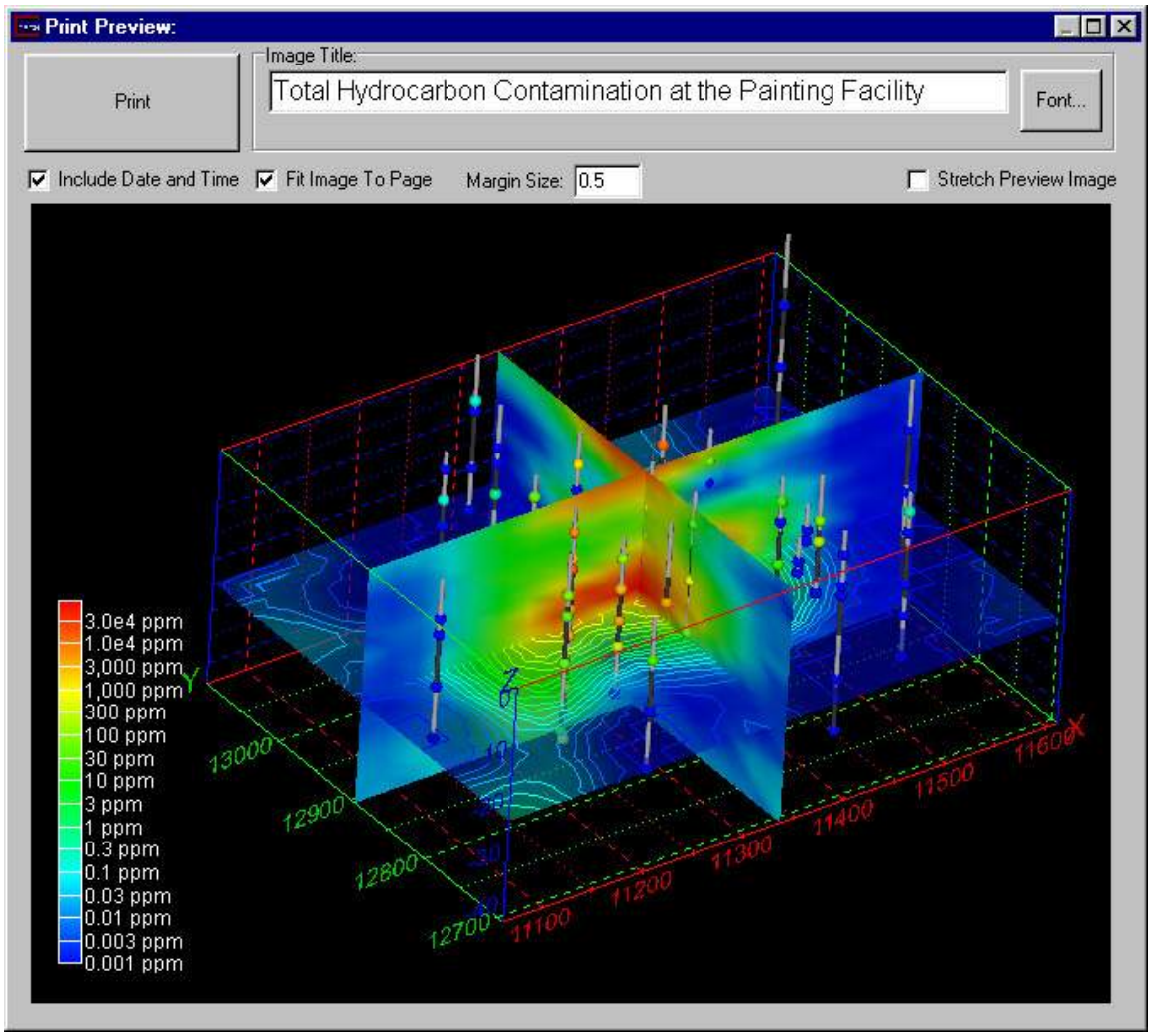
Choose 4DIM View sets the current 4DV script or A4D file. These files can have basic operations such as setting a certain view, or stepping through every frame in a 4DIM, or they can be complex animations.

Save Screen Capture as Image provides a means to save the image in the 4DIM player window (not the border) at whatever resolution (size) your 4DIM player is at the time. Though you can always create high resolution images with Output_Images, this provides a very quick and easy way to save images that are often adequate resolution. When this option is selected you will be prompted for a file name. Supported file types are Windows .BMP (the most common uncompressed file format) and PNG, the highest compression *lossless* file format. A confirmation window will pop-up when the file writing is complete.

Print Screen Capture provides a means to print the image in the 4DIM player window (not the border) at whatever resolution (size) your 4DIM player is at the time. Though you can always print high resolution images with the Print Editor, this provides a very quick and easy way to save print the contents of the 4DIM player **exactly as it appears** at a resolution that is often adequate. When this option is chosen, a window (as shown below) will appear.

You have the following options:

1. Adding an *Image Title* and specifying the font type, size and color.
2. Including the date and time
3. Fitting the Image to the Page, in order to have the image scaled to fill out to the margins (and additional buffer for the title).
4. Specifying Margin Size in inches.
5. *Stretch Preview Image*. This option **stretches** the preview image to fit the preview window. It is most useful for the preview in the Print Editor where the output resolution may be so large that you cannot see the entire image. It may (probably will) cause distortion of the image aspect ratio during preview, but does not affect the output.
6. The Print button opens a dialog allowing you to specify the printer that you wish to use. You may set additional printer settings such as "Landscape" that will affect the final size and orientation.



Show 4DIM Info opens a window that lists te 4DIM name, author and notes. This information is stored in the 4DIM during the creation process.

Transformations with the Mouse

Now that you know how to play the animation, let's interact with itthat is why it is called 4D *Interactive* Model Player. At every frame of the animation, **or** while it is playing, you can interact with the model and perform rotations, scaling or translations. The commands to interact with your mouse are:

Rotate the model.

Move the mouse to a location within the viewer portion of the 4DIM Player's window.

Hold down the left mouse button and move the mouse pointer in various directions. The model rotates.

NOTE: The best way to understand mouse rotations is to consider the viewer screen as a ball floating in a bowl of water. Where you start with the mouse and where you end (release) determine the angle of and amount of rotation.

Zoom in or out on the model.

Press the '+' or '-' key on the keyboard. This will zoom in on the model. Alternatively either the middle mouse button on a 3 button mouse can be clicked and dragged to change the zoom level, or a wheel button will also affect the zoom.

Move (Translate or Pan) the model.

Hold down the right mouse button and drag the object up, down, and around, then center the model.

Try interacting with the model when it is paused and running. Note how much more informative it is to interact with the animation while it is playing.

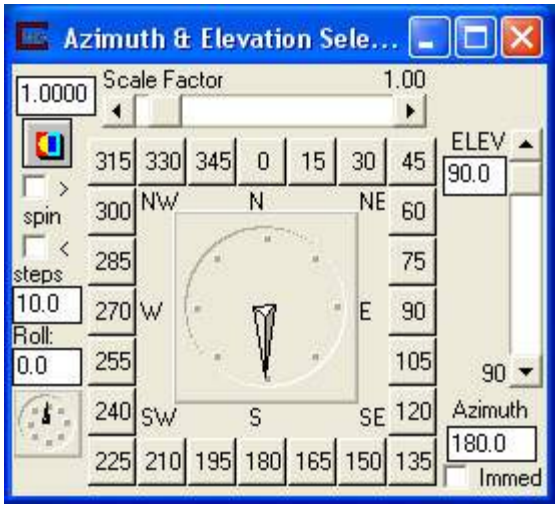
There are two other ways to transform (interact with) models. The first one we will try is using the Az-El Panel.

Transformations with the Az-El Panel

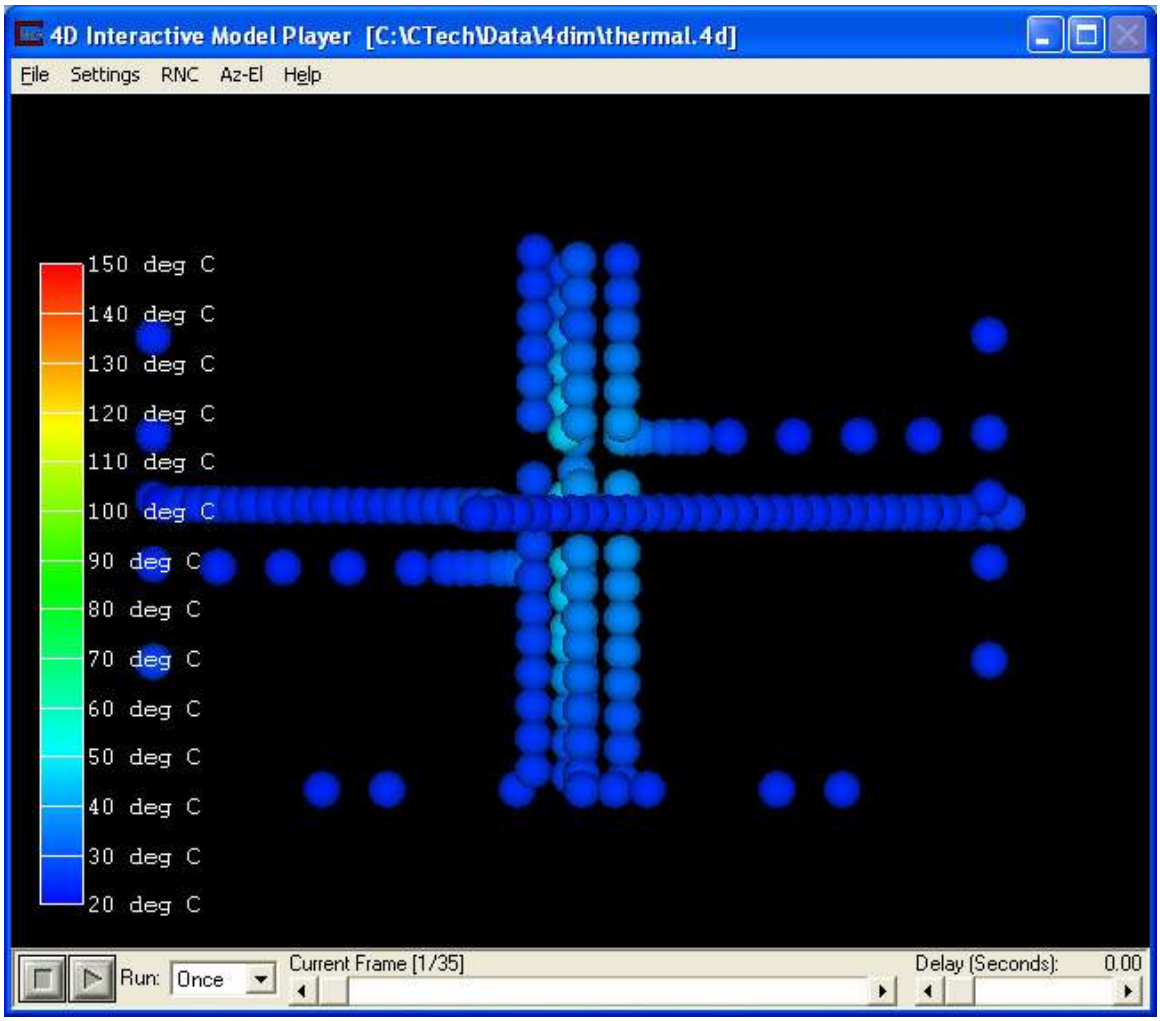
The 4DIM Player gives us another more precise way to transform (scale, and rotate) a model: through the Az-El panel.

Transform the model with the Azimuth and Elevation Panel

Click on Az-El on the 4DIM Player's main menu.



The first time you do this the model will snap to a **top** view. Your screen should look like:



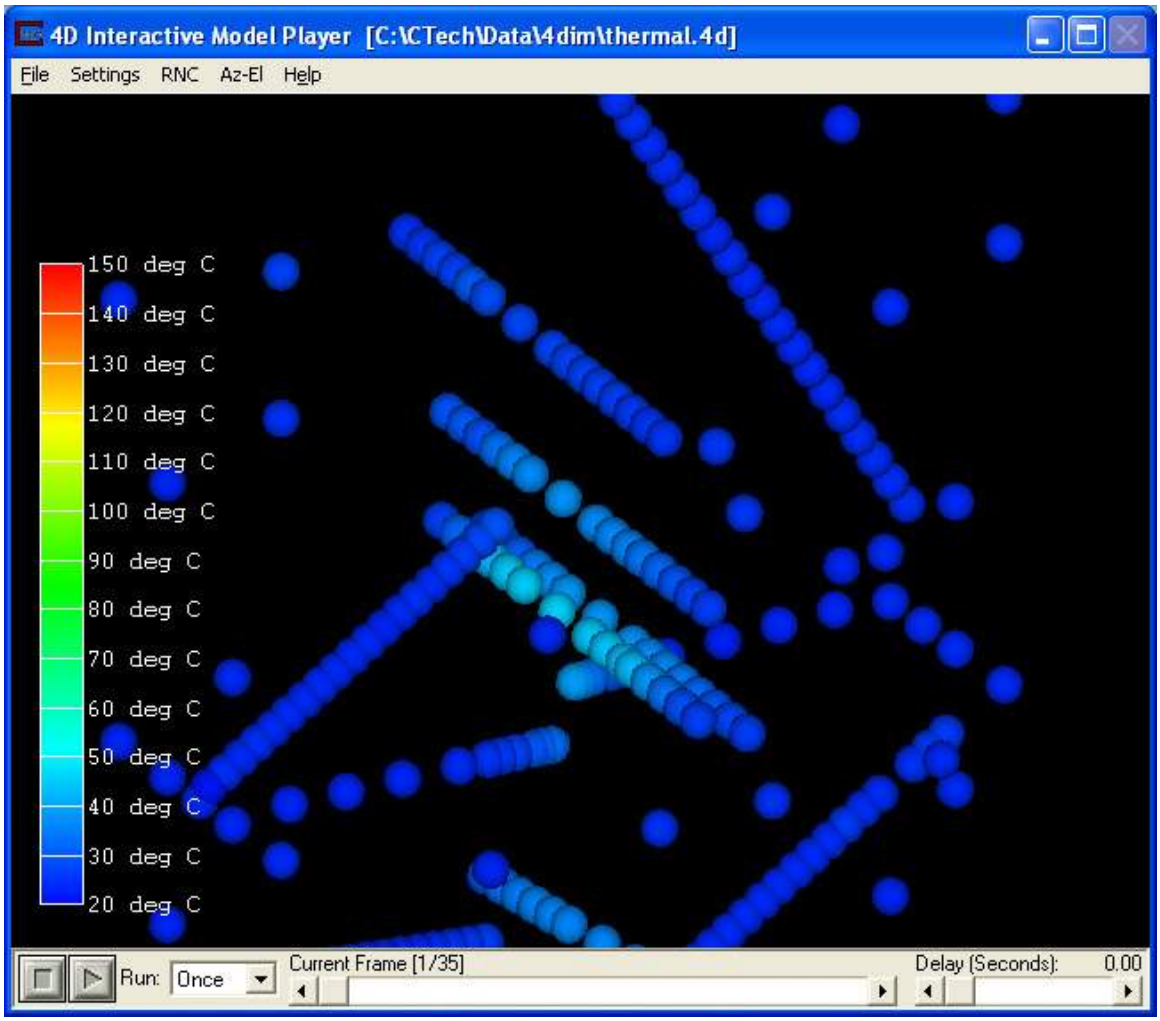
The Az-El panel provides the ability to set specific views using a variety of controls. These include:

- A combined **Azimuth** dial or button array The panel is arranged with an array of azimuth selection buttons in 15 degree increments around the compass, and a dial for setting any azimuthal location. There is also a type-in box for setting a precise azimuth. An interesting feature is that if you move the dial with your mouse, as you cross North, it will add or subtract 360 degrees from the button values. This is important during animations since rotations crossing north must be properly specified to determine the direction and number of revolutions of your model.
- An **Elevation** slider (and type-in) sets the angle of the view from horizontal plane. An elevation value of +90 will result in a view from directly above the object (looking straight down the +Z axis), while an elevation value of -90 will result in a view from directly below the object (looking straight up the -Z axis). For example, a value of +20, provides a view looking at the object from 20 degrees above the horizontal

plane. Note that changing any of the settings on the Az-El panel updates the current view immediately.

- A **Scale Factor** slider (and accompanying type-in) allows the user to specify how much of the Viewer width the object(s) will occupy in the specified perspective view. A Scale Factor of 0.7 will therefore produce a display with the object occupying approximately 70% of the Viewer. Values above 1.0 will generally cause some of the model to be outside of the viewer window.
- The **RNC** (Reset-Normalize-Center) button with the tri-color icon in the upper left corner of the panel forces a recentering of the model and sets the view to the current Az-El settings. Furthermore, this button causes the Viewer to RENORMALIZE, which means that it recomputes the internal system wide scale factor required to fit all objects connected to the Viewer into the current view. Please note that the RNC pull-down menu on the Viewer includes a "Top View" option that is equivalent to this button PLUS resetting all Az-El parameters to their default settings.
- An important feature for creating simple animations and automating viewer manipulations is the **Spin** toggles and it's associated '**steps**' type-in box. The type-in specifies the number of degrees to spin the viewer azimuth and the '>' and '<' toggles determine the direction of spin. Use of this functionally essentially activates a loop whereby checking the '>' or '<' toggle increments the viewer azimuth according to the prescribed 'steps' type-in. For example, if you wish to automate views incrementing every 10 degrees with a clockwise rotation, you should type-in 10.0 (steps) and check the '>' toggle. The viewer will automatically step through from the beginning azimuth in 10.0 degree increments, in a clockwise direction. This spin can be stopped (interrupted) at any time by unchecking the '>' toggle. HINT: You may connect an output_images module to the viewer, then check the dynamic toggle to save a frame of each viewer position for use in building an animation.
- The **Perspective** toggle turns on Perspective mode. When it is turned on, an additional type-in for FOV appears. Additional settings for perspective are under the Camera editor.
- **FOV** (Field of View) sets the camera's included angle when in Perspective mode. The default value of 45 results in a camera with a 45 degree included field of view.
- The **Immediate** toggle turns on immediate mode for scale and elevation sliders and the Azimuth dial. In immediate mode the view updates in real time as you move these controls.
- The **Roll** type-in and very small dial provide control over the object roll axis. This is a rotation that is not normally used since it causes vertical objects to not be vertical. It can be interesting for some fly-through animations or for unusual circumstances.

If you choose a scale of 1.0, an elevation of 25 degrees and an azimuth of 210 degrees you should get something like this:



Play Animation Script Files

The A4D file has been superceded by the 4DV file. Although the newer 4DIM players will play A4D files the newer 4DV files are more feature rich and flexible. The author of 4DIM files is encouraged to take advantage of the ability to bundle any number of 4DV files into their 4DIMs and minimize the burden on the end-consumer (user) of the 4DIMs.

The best way to appreciate the power of bundled scripts is to play one. Several preset views and animation script files that work with most all 4DIM files are built into the Standard 4DIM Script library. To access them, from the 4D Interactive Model Player window, choose the File pull-down menu and select *Choose 4DIM View*. On this window there is a radio selector to choose between Standard Library and Bundled Views and Scripts. All 4DIM files will have access to the Standard Library, whereas Bundled scripts are only available if they are created and bundled by the 4DIM author.

Select *Standard Library* and chose *Fancy Rotation*.

Unfortunately it isn't possible to show you in the help what happens, but what you should see is an animation that begins with a TOP view, rotates down to 20 degree elevation, spins around 360 degrees and then returns to the top view.

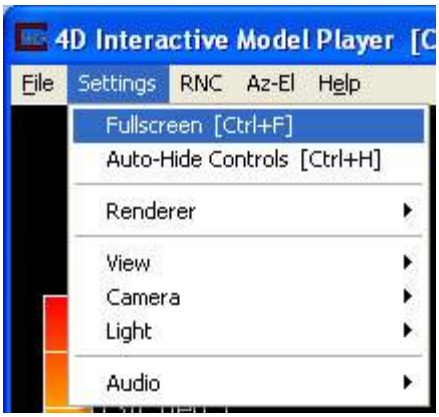
A4D Files: This help discussion is still included for backwards compatibility.

From the 4D Interactive Model Player window, choose the File pull-down menu and select *Choose 4DIM View*. On this window there is a button in the lower right to *Play .A4D File*. Browse to the .a4d file to select it. If the .a4d file is an older style fly through, once it finishes it leaves the Viewer in a non-standard mode. Attempts to interact with the model will give unexpected results. To reset the Viewer after playing an A4D file, choose *Reset After A4D Fly-Through* on the RNC pull-down menu.

Note: If you have a Multimedia keyboard with Play/Pause and Stop buttons, these will control the 4DIM player during script playback the same as the buttons at the bottom of the player.

Explore 4DIM Player's Settings

Explore 4DIM Player's Settings menus. The tables and figures below summarize 4DIM Player's settings.



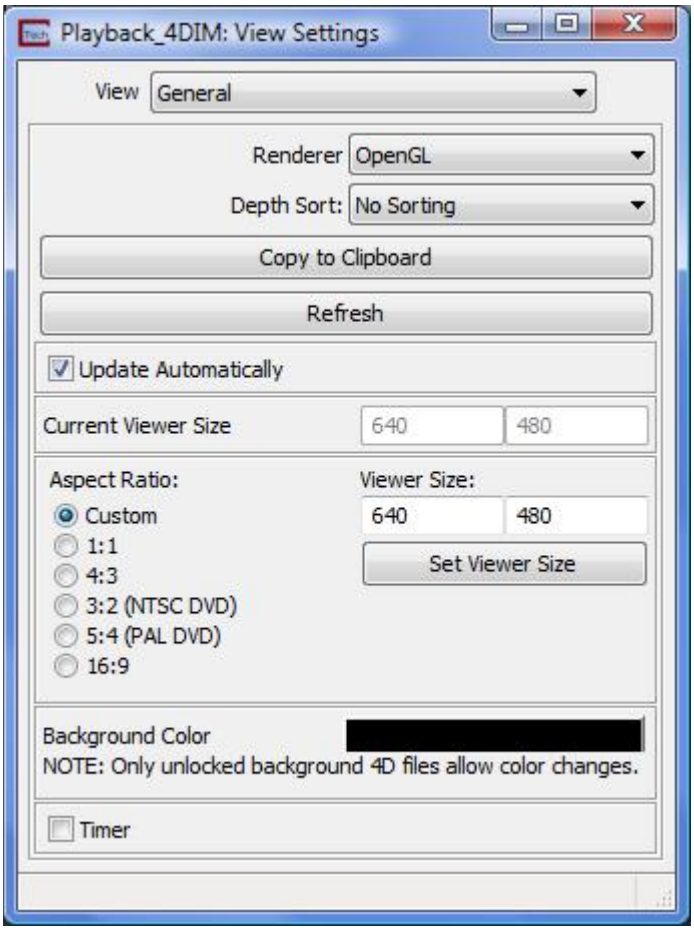
Fullscreen [Ctrl+F] toggles the 4DIM viewer into and out of full screen mode.

Auto-Hide Controls [Ctrl+H] will toggle the play control and player settings to hide unless the mouse moves over the correct area.

Renderer selection (OpenGL or Software).

View menu's Advanced Settings provides control over several view-related settings. These include:

1. Renderer Selection of OpenGL or Software
2. Depth Sorting (Software Renderer only)
3. Copy Viewer image to clipboard
4. Viewer Size and Aspect Ratio
5. Background Color (if a locked multi-color background was not used when the 4DIM file was saved)
6. Turn on Timer to test your computer's graphics speed



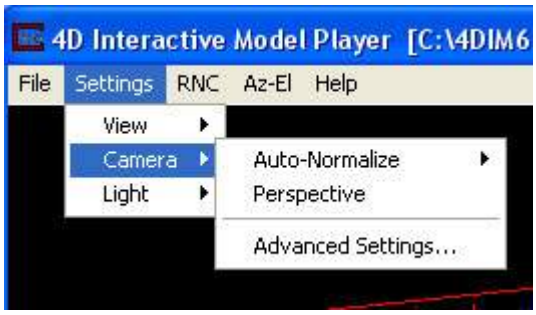
The OpenGL renderer is a high performance renderer which is native to the Windows operating system, and is the default option. For most objects, OpenGL rendering provides much faster rendering performance than the software renderer, and is the default setting.

However, the Software renderer provides some functionality that the OpenGL renderer does not, and therefore is the desired option for some operations. Specifically, spheres are calculated in the software renderer much more quickly and efficiently then OpenGL which renders a many faceted object. If you are visualizing a large number of spheres you should choose the software rendering option. The other important difference is that the Software Renderer support Depth Sorting when your view has multiple transparent objects.

When there are two or more transparent objects in your view, without depth sorting the appearance of which object is in front or behind can be improperly displayed. Properly displaying multiple transparent objects requires sorting the object's order of display based on their apparent position relative to the observer. There are two different depth sorting options, each requires progressively more computation and is therefore increasingly slower than normal rendering.

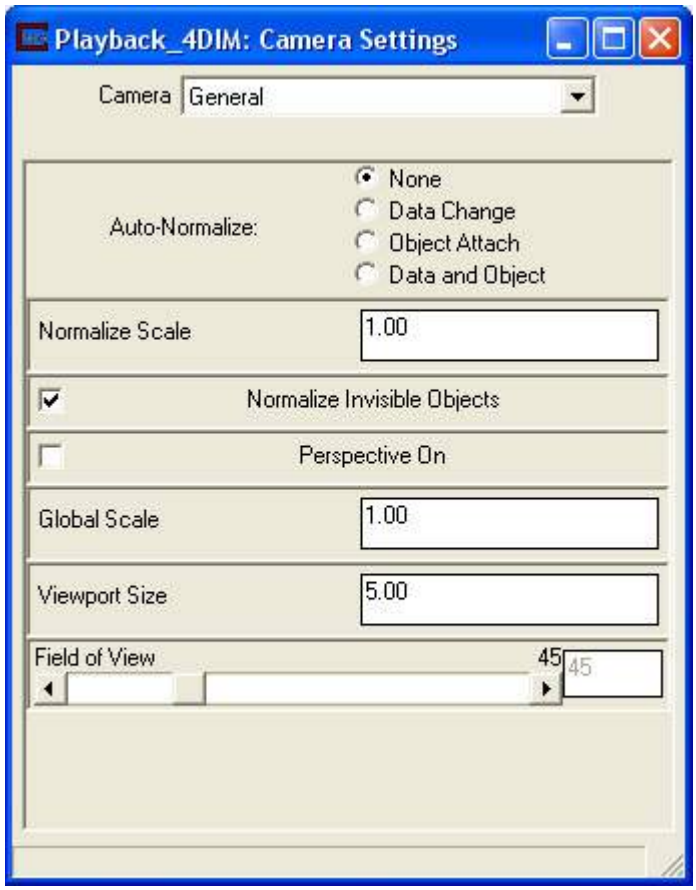
Remember that these options are not available in OpenGL since none of the graphics card manufacturers support this level of sophistication.

_____



Camera menu provides control over many settings that affect the appearance of your model. These include:

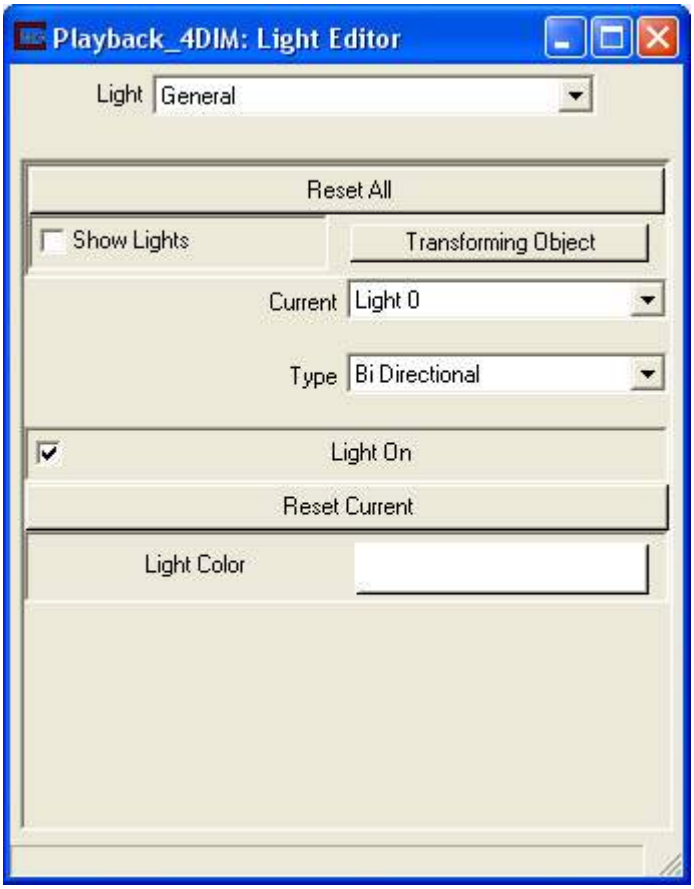
1. Turning Auto-Normalize on or off
2. Turning Perspective on or off
3. Advanced Settings lets you
    a. Control additional auto-normalize behavior
    b. Set auto-normalize scale
    c. Control the field of view when in perspective mode



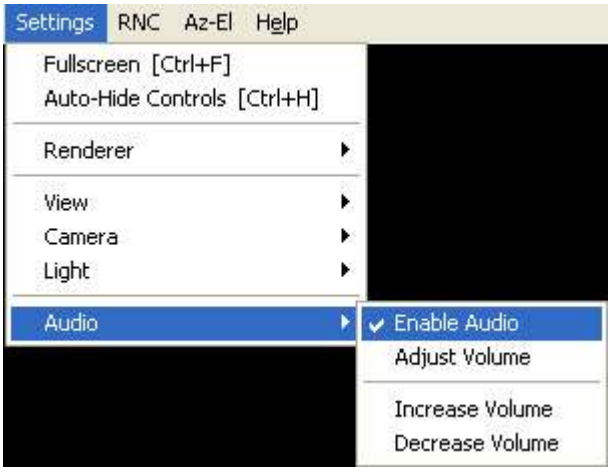
_____

Light menu allows you to control how lighting affects the shading and color of objects in your view. The options include:

1. Whether to show lights
2. Whether to transform lights or objects with your mouse
3. Advanced Settings lets you
    a. Set the properties of individual lights
    b. Turn lights on/off
    c. Set light's color



_____



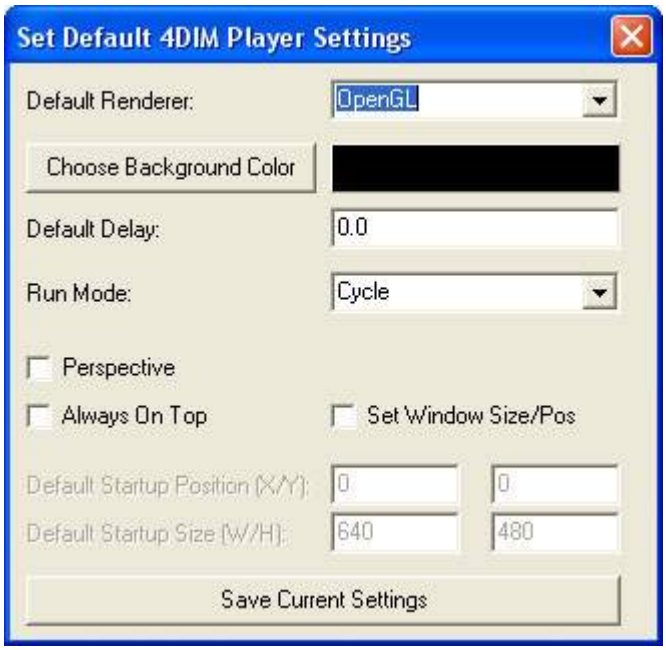
Audio menu allows you to control whether to include audio, and the audio volume

1. Whether to enable Audio.
2. Adjust Volume using a slider.
3. Increase Volume.
4. Decrease Volume.

Configure 4DIM Player

When you install the 4DIM player, a shortcut for the program that allows you to customize the start-up behavior of the player is also included.

When you run this a window appears that allows you to set start-up conditions for the player.



This program allows you to set the default:

1. Renderer
2. Background color
3. Delay time
4. Run Mode
5. Setting for Perspective
6. Always on top
7. Window size and position (NOTE: The window size/position is the size position of the ENTIRE window, not the renderable section.)

When you save these settings they will apply to the player each time you start it in the future. Note that this does not change the settings of an open player.

Exit 4DIM Player

The ability to exit is only available on the stand-alone 4DIM Player. To exit the 4DIM Player, select the File->Exit pull-down command.

The 4DIM Player exits after displaying a confirmation message.

Overview of Module Libraries

EVS/MVS Modules

EVS & MVS modules can each be considered software applications that can be combined together by the user to form high level customized applications performing analysis and visualization. These modules have input and output ports and user interfaces.

File Modules

Modules that read or write data files and convert them to/from EVS grids with coordinates and/or data.

Display Modules

Modules whose primary purpose is visualization rather than subsetting, processsing or manipulation of data.

Subsetting Modules

Modules that subsetting EVS grids with coordinates and/or data. Subsetting includes:

1. selective subsetting of data components (e.g. extract_component)
2. reduction of dimensionality (e.g. slice)
3. spatial subsetting (e.g. cut)
4. data-based subsetting (e.g. plume_volume)

Processing Modules

Modules whose primary purpose is processsing EVS grids with coordinates and/or data using a particular mathematical or analytical method. This includes modules that facilitate subsetting (e.g. area_cut) but do not actually perform the subsetting.

Image Modules

Image modules include all modules that read, write, modify or work with raster (bitmap) images.

Animation Modules

Animation modules that facilitate the creation of animations or produce animated effects.

Tools Modules

Tools modules include the nuts-and-bolts to complete your applications and include a collection of modules that do not fit in any of the categories above.

Cell Data Modules

Cell data modules act on a field's Cell_Data and/or subset the grid such that the mesh output from the module has a restricted domain or smaller extent or lesser dimensionality.

Overview of Module Libraries

EVS/MVS Modules

EVS & MVS modules can each be considered software applications that can be combined together by the user to form high level customized applications performing analysis and visualization. These modules have input and output ports and user interfaces.

File Modules

Modules that read or write data files and convert them to/from EVS grids with coordinates and/or data.

Display Modules

Modules whose primary purpose is visualization rather than subsetting, processsing or manipulation of data.

Subsetting Modules

Modules that subsetting EVS grids with coordinates and/or data. Subsetting includes:

1. selective subsetting of data components (e.g. extract_component)
2. reduction of dimensionality (e.g. slice)
3. spatial subsetting (e.g. cut)
4. data-based subsetting (e.g. plume_volume)

Processing Modules

Modules whose primary purpose is processsing EVS grids with coordinates and/or data using a particular mathematical or analytical method. This includes modules that facilitate subsetting (e.g. area_cut) but do not actually perform the subsetting.

Image Modules

Image modules include all modules that read, write, modify or work with raster (bitmap) images.

Animation Modules

Animation modules that facilitate the creation of animations or produce animated effects.

Tools Modules

Tools modules include the nuts-and-bolts to complete your applications and include a collection of modules that do not fit in any of the categories above.

Cell Data Modules

Cell data modules act on a field's Cell_Data and/or subset the grid such that the mesh output from the module has a restricted domain or smaller extent or lesser dimensionality.

Module Status Icons

EVS & MVS modules have four icons which provide visual indication of the module status.

The four icons are:

1. **X** 

: This icon signifies that the module has not yet run and has no output. For the Viewer module, this icon is visible if no modules are connected.
2. **Solid Octagon** 

: This icon signifies that the module has not yet run, has no output, and its RUN toggle is OFF.
3. **No Icon** 

: The lack of an icon signifies that the module has run and has output. When you load an application, once all modules are "icon-

free", you have visual confirmation that the application has likely completed and is working correctly.

4. **Hollow Octagon** 

: This icon signifies that the module has run, has output, but its RUN toggle is OFF.

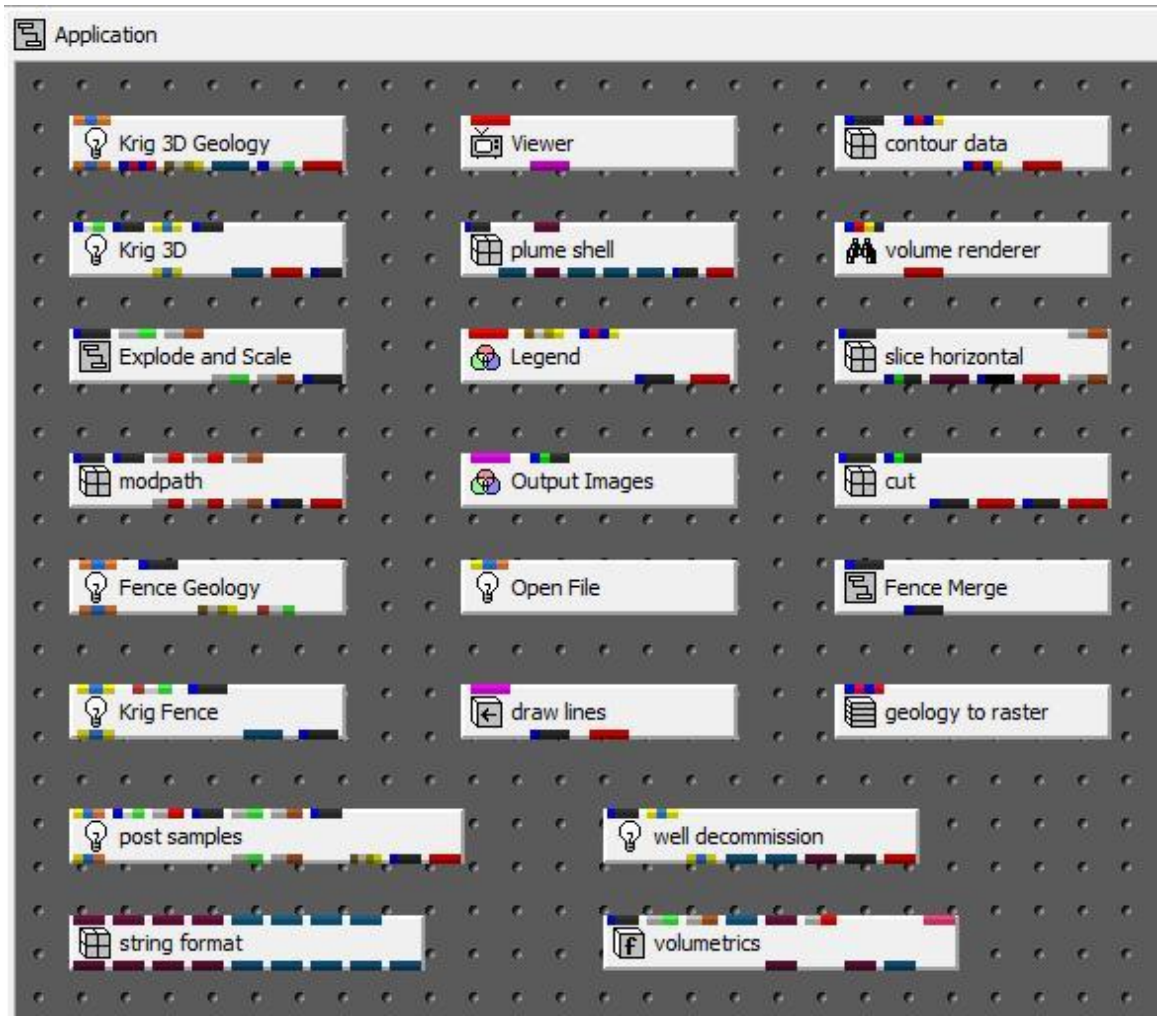
Not all modules have all four icons, and a few modules have none. Those modules generally have no options or do not have output ports.

ModuleInput & Output Ports

1. Red – **Renderable object**  This commonly used port connects to various modules, most notably the Viewer. It contains the grids, data and rendering information.
2. Blue-Black – **EVS field**  This is the most common port which passes your grids and data (nodal or cell) between modules that create these fields and those that subset or modify them.
3. Yellow-Blue-Yellow: **Chemistry files**  This data port is color coded to signify that it should contain chemistry file names only.
4. Yellow-Blue-Orange:**Chemistry or Geology Files**  This data port is color coded to signify that it should contain Chemistry or Geology file names only.
5. Orange-Blue-Orange: **Geology Files**  This data port is color coded to signify that it should contain Geology file names only.
6. Orange-Blue- Yellow: **Pre-Geology Files**  This data port is color coded to signify that it should contain Pre-Geology file names only.
7. Blue-White-Green: **Geology Output**  This common port is used to pass raw geology surfaces between various modules. Created only by Krig_3D_Geology.
8. Grey-Brown: **Z-Scale**  This port passes the z-exaggeration factor. Used by many modules such as Explode_and_Scale.
9. Grey-Green: **Explode Factor**  This port passes the explode factor. Used by many modules such as Explode_and_Scale.
10. Purple: **View**:  This port is a Viewer output port and input to many other modules. It is used to pass information to render images, allow for interactive drawing (Click_Sketch), and auto-size (add_logo).
11. Dark-Pink: **int**  This port passes integer values. The port may be vertically oriented as in the "int" module.
12. Dark Brown: **double**  This port passes floating point double precision values. The port may be vertically oriented as in the "double" module.
13. Greyish Blue: **string**  This port passes character string (text) information. The port may be vertically oriented as in the "string" module. Other modules using this include Titles and field_math.

14. Blue-Yellow-Red: **Multi-surface**  This is a special port to pass information to the geology_to_vistas module
15. Blue-Red-Blue-Red: **Grid-info**  This is a special port to pass information to the geology_to_vistas module
16. Blue-Green-Black: **Image or 2D Field**  This port passes image data (which is a 2D field) or simple flat surfaces (e.g. slice planes).
17. Blue-Red-Blue-Yellow: **Contour levels**  This is a special port to pass Contour level information between modules such as solid_3d_set, isolines, color_legend and Datamap_Editor.
18. Blue-Red-Yellow-Black: **Uniform Field**  This is a special port to pass uniform fields into modules such as "volume_renderer". You can connect to this port with port 2 (blue-black) if the output is a uniform field.
19. Brown-White-Green: **Fence Geology**  This port passes geologic information between Fence Geology and Krig_Fence.
20. Grey-Red **Time/Date**  This port passes time and/or date information
21. Brown-Grey-Light Brown-Beige  **Geology Material Names** This port passes Geology Material Name information from modules like Krig_3D_Geology or post_samples to modules such as Legend.

Below is a collection of modules (a small subset of all EVS/MVS modules) that includes most of the above input/output ports.



When trying to determine which port on the module icon corresponds to which documented item, remember:

A) Ports are documented from left to right.

(i.e., the first input port documented on the reference page is the leftmost port on the module, the second input port documented is second port from the left on the module, and so on.)

B) You can always use the module's Info panel to see the name and class of a port:

With the cursor on the module, press the right mouse button to activate the pulldown menu.

Select Info.

The first input port listed is the leftmost port on the icon, the second port listed is the second port from the left on the icon, and so on.

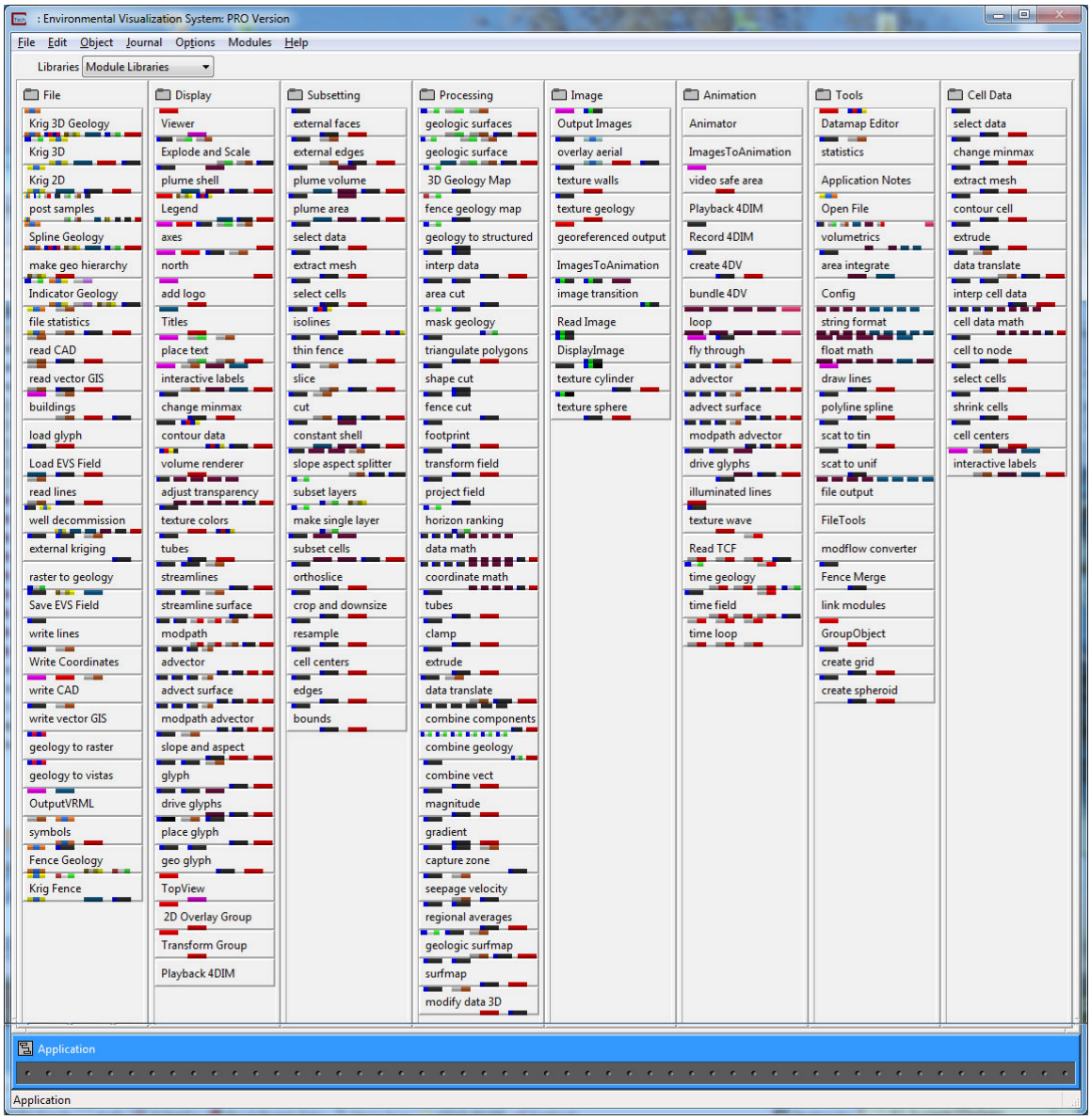
C) You can always select use your right mouse button to click on a module and choose "Help" for more detail.

EVS-PROModule Libraries

The figure below shows all of the modules available in EVS-PRO (Environmental Visualization System-PRO). There are 184 unique modules in the following libraries:

1. 28 File modules
2. 31 Display modules
3. 22 Subsetting modules
4. 32 Processing modules
5. 11 Image modules
6. 19 Animation modules
7. 21 Tools modules
8. 13 Cell_Data modules

Note: some modules are duplicated in other libraries because of their functionality.



C Tech Main Help

MVSModule Libraries

The figure below shows all of the modules available in MVS (Mining Visualization System). There are 207 unique modules in the following libraries:

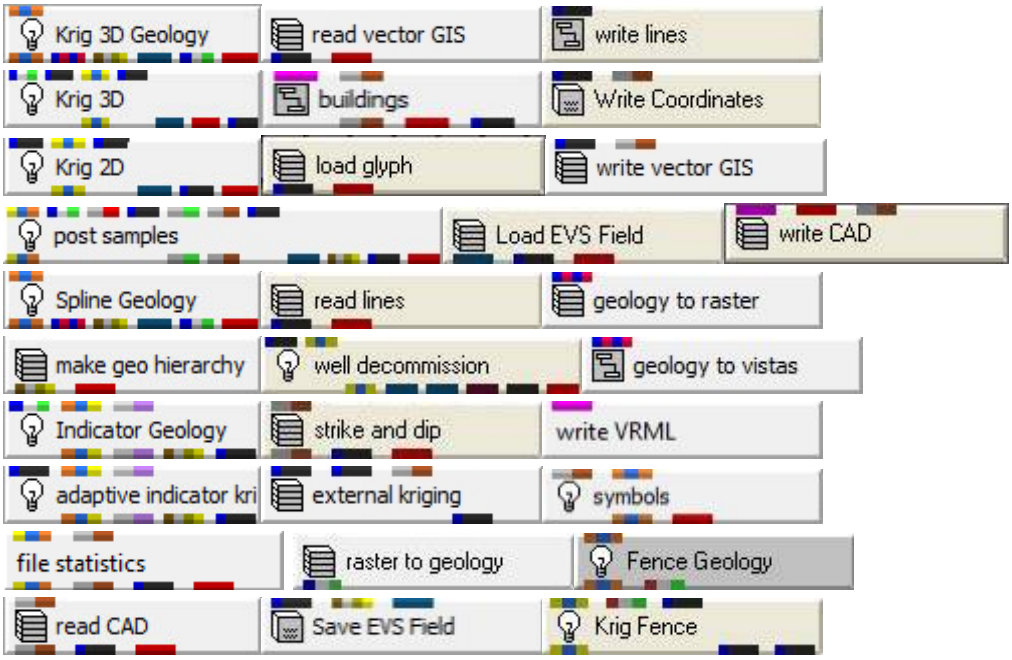
1. 30 File modules
2. 34 Display modules
3. 24 Subsetting modules
4. 39 Processing modules
5. 11 Image modules
6. 20 Animation modules

7. 26 Tools modules
8. 13 Cell_Data modules

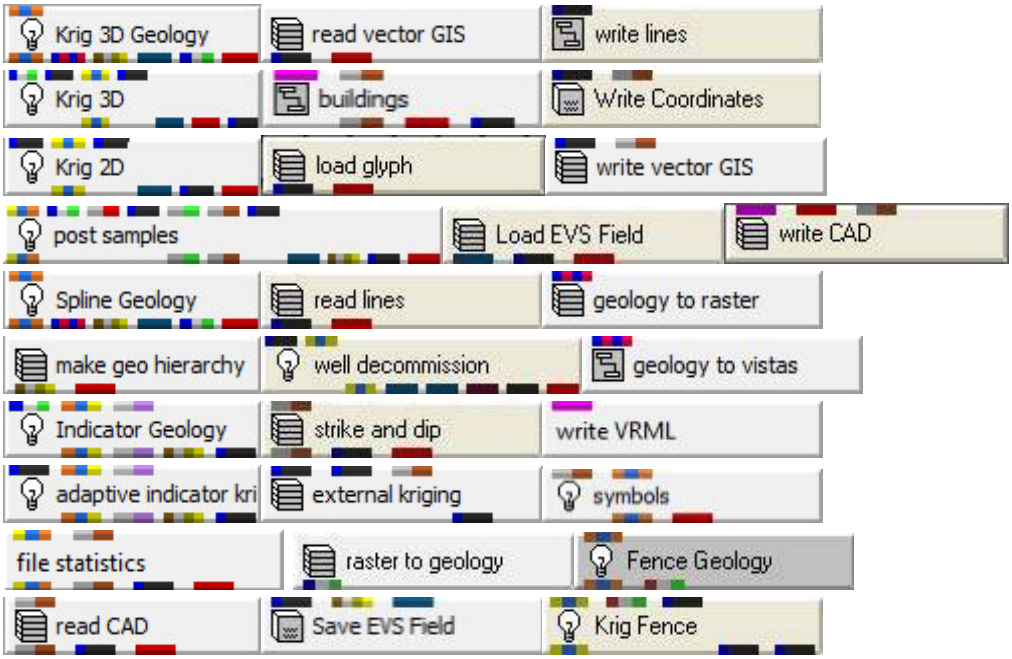
Note: some modules are duplicated in other libraries because of their functionality.

C Tech Main Help

File Modules



File Modules



Krig_3D_Geology

General Module Function

The Krig_3D_Geology module uses data in .geo files and .gmf files to model the surfaces of geologic layers that will provide the geologic framework for

three-dimensional geologic modeling and parameter estimation. Krig_3D_Geology creates layers of quadrilateral (4 node) elements in which each node is assigned the kriged elevation for the specific geologic surface at that point in space. The output of Krig_3D_Geology is a data field that can be sent to the Krig 3D and 3D Geology Map modules where the quadrilateral elements are connected to the element nodes in adjacent geologic surfaces to create layers of hexahedral (8-node) elements. The output of Krig_3D_Geology can also be sent to the Geologic Surface module which will allow visualization of the individual layers of quadrilateral elements (the surfaces) that comprise the surface layer and the bottoms of each geologic layer. Krig_3D_Geology has the capability to produce layer surfaces within the convex hull of the data domain, within a rectilinear domain with equally spaced nodes, or within a rectilinear domain with specified internodal distances such as a finite-difference model grid. The finite-difference gridding capabilities allows the user to visually design a grid with variable spacing, and then krige the geologic layer elevations directly to the finite difference grid nodes. This functionality facilitates setup of finite difference and or hexahedral element finite element model input files. Krig_3D_Geology also provides geologic surface definitions to the post_samples module to allow exploding of boreholes and samples by geologic layer.

Note: Krig_3D_Geology has the ability to read .apdv, .aidv and .pgf file to create a single geologic layer model. This was not done as a preferred alternative to creating/representing your valid site geology. However, most sites have some ground surface topography variation. If Krig_3D is used without geology input, the resulting output will have flat top and bottom surfaces. The flat top surface may be below or above the actual ground surface at various locations. This can result in plume volumes that are inaccurate.

When a .apdv or .pgf is read by Krig_3D_Geology or Spline_Geology the files are interpreted as geology as follows:

1) If ***Top*** of boring elevations are provided in the file, these values are used to create the ground surface.

2) If ***Top*** of boring elevations **are not** provide in the file, the elevations of the highest sample in each boring are used to create the ground surface.

3) The bottom surface is created as a flat surface slightly below the lowest sample in the file. The elevation of the surface is computed by taking the lowest sample and subtracting 5% of the total z-extent of the samples.

When reading these files, you will get a single layer which goes to either the Top column (if it exists) otherwise, the top sample in each boring, and 5% below the lowest sample in the file (flat bottom). This allows you to create a convex hull around data without having geology info. It also provide a topographic top surfaces if your analyte (e.g. chemistry) or PGF file has Tops (grounds surface elevations). Also nice for doing indicator kriging (since a single, well-defined pgf can give you an entire indicator model now). Be aware that if Top is specified, but all values are exactly 0.0, the top sample elevation for each boring will be used.

Module Input Ports

Krig_3D_Geology has an Orange-Blue-Orange input port that can receive the geology file name.

Module Output Ports

Krig_3D_Geology has six output ports.

1) Read_geo_data_file (Orange-Blue-Orange) : Supplies the geology file name.

2) Vistas Group (blue-magenta-blue-magenta) : Provides input to the Geology_to_Vistas module.

3) Geology Material Names (Brown-Grey-Light Brown-Beige) : Provides geologic material information for the Legend module.

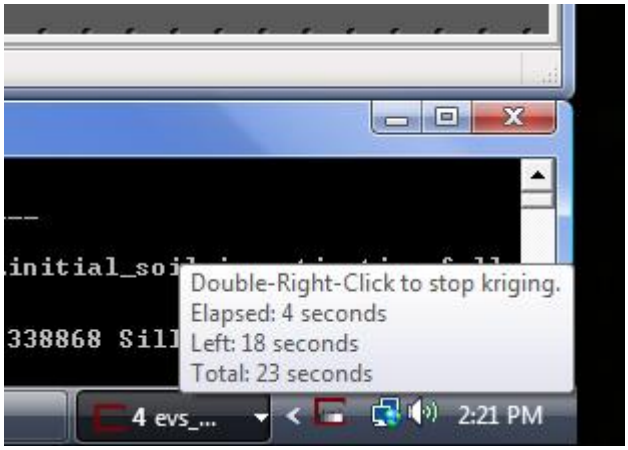
4) status_out(blue/green) : Outputs a string containing status updates while the module is running.

5) output0 (blue-white-green) : The primary output field. The primary output field port can be connected to the Krig_3D, 3D_Geology Map, and Geologic Surface modules.

6) Grid_out (red) : Outputs a renderable geometry of a finite difference grid.

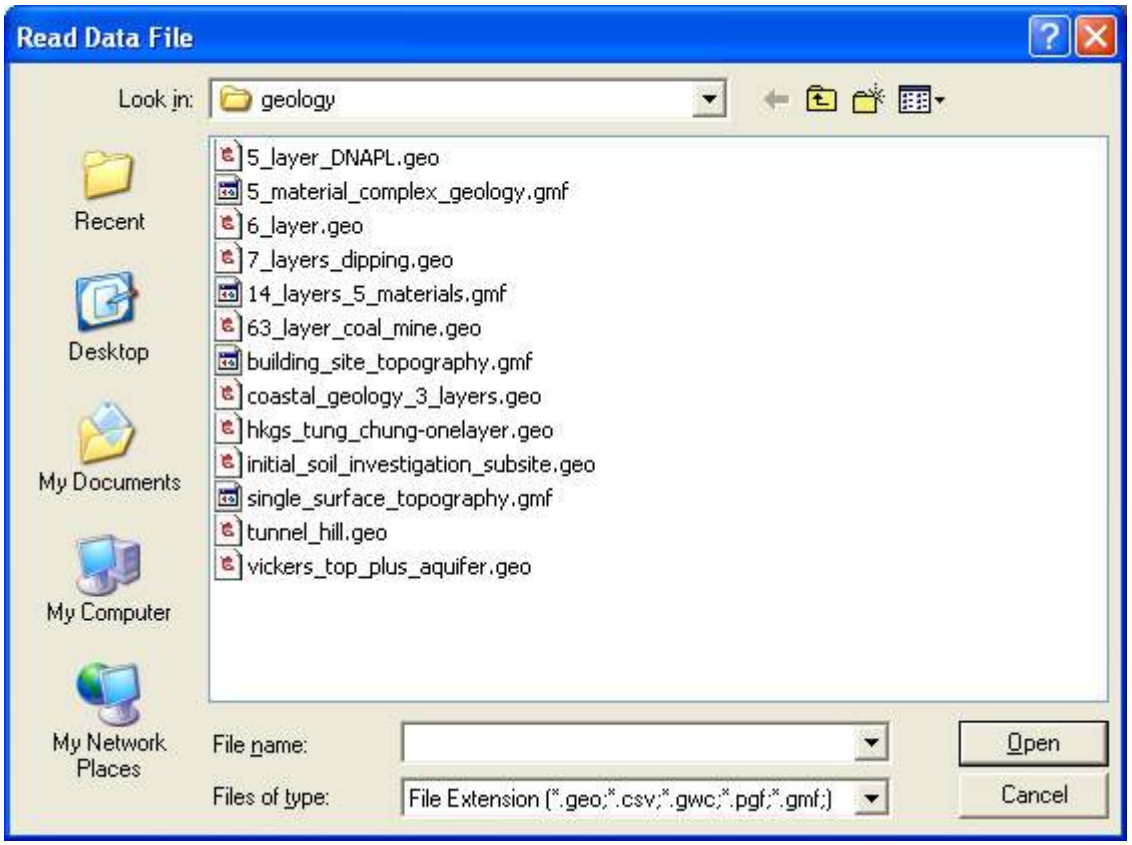
Module Status: Interruptible

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel

The control panel of Krig_3D_Geology is shown in the figure above. The **ReadDataFile** button opens the *File Browser* panel shown in the figure below.



The file browser is a standard window browser that allows the user to specify the directory in which the files reside, and the extension used to filter the displayed available file names. For Krig_3D_Geology, the default file extension is .geo, .apdv, .aidv, .pgf, or .gmf. The format of .geo files is described in the geo_file_format Help topic. The format of .gmf files is described in the gmf_file_format Help topic.

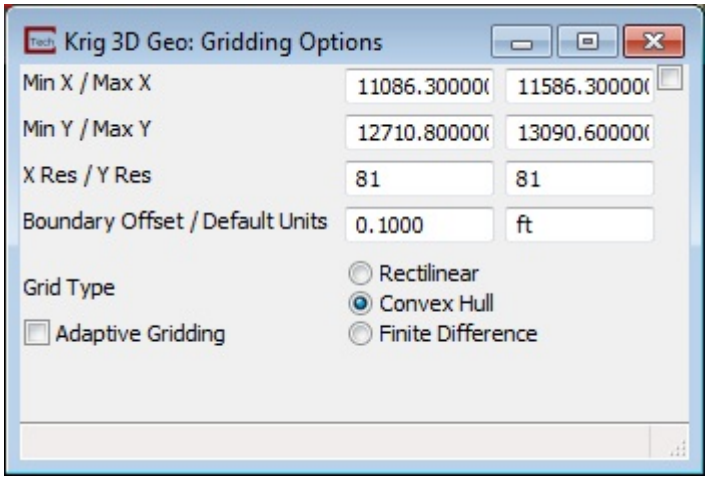
Note that this module will begin running only when a file is selected and the **Accept All Current Values** button is pushed. However, if a finite difference grid domain is to be setup for the kriging, then a geology file should **not** be selected until the grid domain has been setup.

The "Run" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed. When it is off, the module will not run.

Both .geo and .gmf geology files support inclusion of parameters to specify geologic layer material number. Within EVS, layers are numbered beginning with the top layer as zero. Previously, there was no other parameter associated specifically with layers. Now, each layer can be assigned a material number. This allows for specifying layer material independent of layer number.

Module Parameter Subpanels

Krig_3D_Geology has three subpanels, which allow the user to set the parameters used for creating the kriging grid, the semivariogram production, and for creating compututational surfaces. Clicking on either the check boxes next to the subpanel names, or on the names themselves will bring up the subpanel data entry screens. Note that the subpanels have to be closed by clicking on the box in the **Module Control Panel**.



The **Gridding Options** subpanel of Krig_3D_Geology is shown above.

Although it is at the bottom of the subpanel, the first option that should be set by the user is ***Grid Type***, as this will determine the nature of the domain that will be kriged. The ***Rectilinear*** option is used when the user wishes to produce a model that contains estimated values everywhere inside a user specified rectangular domain. The ***Convex Hull*** Boundary Option is used when the user wishes to produce a model that can have an irregular boundary that is defined by the distribution of measured data points. The Convex Hull of a data set can be thought of as the domain that would be outlined by stretching a rubber band around the external data points in the data set. The Convex Hull boundary option effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points. The ***Finite Difference*** domain allows the user to krige the elevations of the geologic layers directly to the finite difference grid node locations that were setup in the control panel of Krig_3D_Geology.

Note that the finite difference grid must be designed and displayed before the user selects a .geo file and clicks the **Accept all Current Values** button.

Adaptive Gridding causes the grid nodes to be shifted up to 30% of a cell width to cause the nodes to align with your input data values. By having grid node coincident with surface points, the grid will exactly honor the surface at that location. This help maintain accuracy with coarser grids.

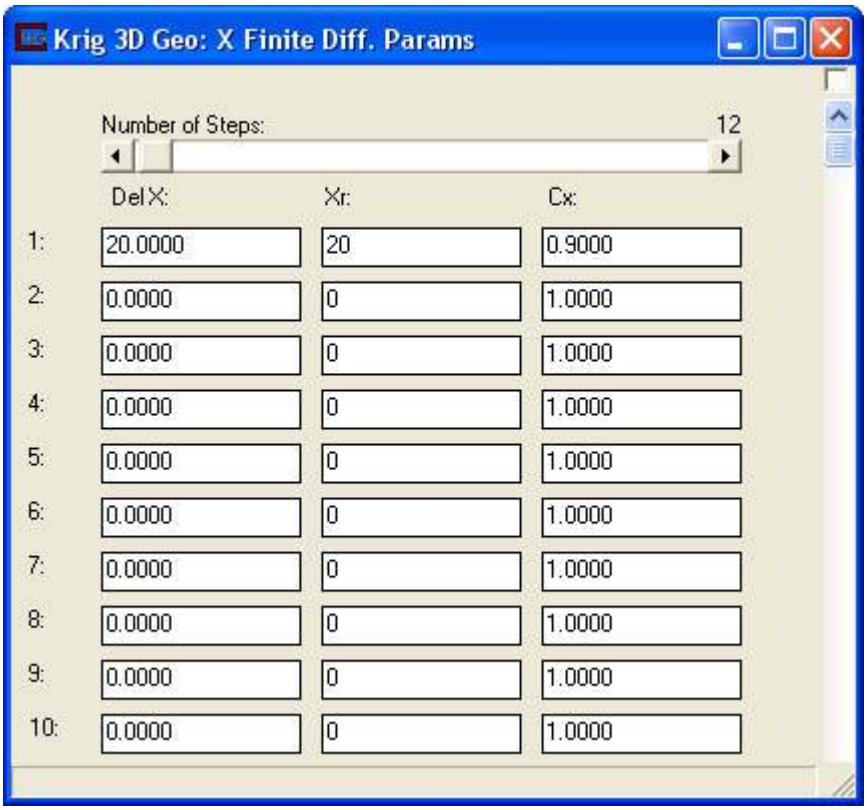
The **Min X, Max X, Min Yand *Max Y*** inputs allow the user to define the horizontal domain within the data set in which kriging of the geologic surfaces will be completed. The Min and Max values are only used when the Rectilinear Option is selected. The Min (x-y) values are also used to set the origin for the Finite Difference Grid Type option. A value of 0 is the default for these parameters, which results in a model domain that is defined by the entire data set when the module is run. When kriging within a finite difference domain, the Min (x-y) values are used to set the origin (lower left hand corner before grid rotation). If the user is uncertain of the X and Y limits of the data domain, the module should be run with the default 0 values, and upon completion of execution, the values in the X and Y input fields will be the min and max values of X and Y in the data set. Alternatively, the File_Statistics module can be used to examine the data characteristics. There is no specification of the Z grid dimensions, as Krig_3D_Geology outputs 2-D surfaces of each geologic interface. The modules that use the output of Krig_3D_Geology define the Z grid dimensions as specified by the user in those modules.

The ***X Res and Y Res*** parameters specify the number of grid nodes that will be included within the model domain. The number of grid elements along either axis of the model is simply the X Res or Y Res value minus one, as every element has two bounding nodes. The default value for these parameters is 41, but the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user. The robust kriging algorithms in EVS generally produce reasonable modeled distributions with a fewer number of grid nodes than the user may be used to, so the recommended procedure for setting the X and Y Res parameters is to start with less, and then increase the value until an acceptable model fidelity is obtained.

The ***Boundary Offset*** parameter sets the distance that the convex hull for the kriging domain will be set outside of the actual convex hull of the data. This parameter allows the user to specify the distance outside of the actual data in which the parameter values will be extrapolated. The distance is a percentage of the diagonal extent in the X-Y plane. The default is 0.10 (10%). For example: if data extent is 100 in x and 100 in y, diagonal distance is 144. It will make the convex hull offset 14.4 total by offsetting 7.2 on all sides. This eliminates dangerously large offsets when data extent is small.

The ***Default Units*** parameter allows the user to set the coordinate units of the model if the file being read does not contain them. If the file does contain coordinate units, this parameter is ignored.

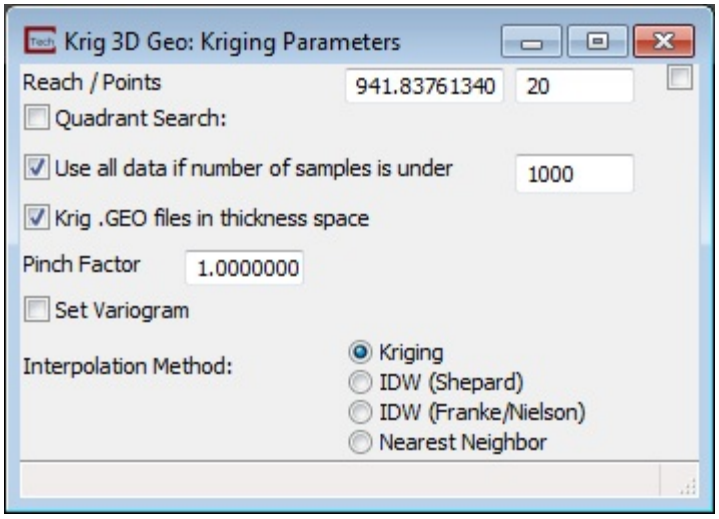
To set up a finite difference grid, the Gridding options checkbox should be checed and the Finite Difference Grid Type should be selected, which will bring up the X and Y check boxes, and the Angle data input field. The angle input field allows the user to specify an angle in degrees counter-clockwise from true north that the resulting grid will be rotated. The finite difference grid corresponding to the input parameters can be visualized at any time by clicking the **Accept All Current Values** button (before a .geo file has been specified in the file browser). The *Finite Difference Gridding* subpanels for X and Y allow the user to constrain the modeled domain to be within the finite difference grid setup using the Krig_3D_Geology control panel. Note the X and Y values specified in these subpanels override any other specified values for Min and Max X and Y in the Gridding Options. Clicking on the X or Y subpanels in finite difference gridding brings up a subpanel similar to the one shown in the figure below.



The first parameter is a slider to specify the total ***Number of Steps*** (grid regions). There is no limit to the number of steps (regions) and each region can any number of elements (cells) and can have constant size cells, decreasing (Cx < 1.0) or increasing (Cx>1.0).

The user builds the finite difference model grid by specifying the distance between the first and second grid nodes (or the element width) in the Del X input field. The number of additional nodes to be placed at this spacing is then specified in the Xr (standing for "X repeat") input field. If a gradually varying node spacing is desired between the number of nodes specified in Xr, then a multiplication factor is specified in the Cx (standing for "Change X") input field. Values greater than 1.0 create an increasing element size, and

values less than 1.0 create diminishing element sizes. The grid design process is identical for the Y grid panel. These panels allow the user to quickly design and visualize a rectilinear model grid.



The **KrigingParameters** subpanel of Krig_3D_Geology is shown in the figure above.

The ***Reach*** input field defines the radial distance (in user units) from any given model node that the kriging module will look for data points to be included in the estimation of the model parameter at that node. The default value of reach is 0, which results in the module calculating a reach value, which is approximately two-thirds of the longest distance between any two data points in the data set.

The ***Points*** parameter defines the maximum number of data points (within the specified reach) that will be considered for the parameter estimation at a model node. The default value for points is 20, which generally provides reasonably smooth modeled parameter distributions. The effects of decreasing and increasing the values for reach and points on the model output are somewhat similar, but for different reasons. If the data have a fairly even spatial distribution throughout the domain, then increasing these values will generally include more of the input data points that will be used to krige the value for a given model node, and thus will result in smoother modeled data distributions.

Decreasing the values of reach and points (in an evenly distributed data set) results in fewer input data points being used to calculate the parameter estimates at a given model node, and result in modeled distributions with greater variations across smaller areas. The user should consider both the spatial distribution and the range of values in the input data set when deciding upon values for the reach and points parameters. If the specified reach is too small to allow the kriging module to locate at least one point within the search area, then no kriging will be completed at that model node, the nodal value will be set to 0, and the confidence level will be set to <0.1%. Note that this nodal value is generally inappropriate, and the regions of the model receiving the 0 values should be subsetted out of the by using an plume_volume module with a confidence isocomponent of 1%. If the user

specifies a large number of points (that are within the specified reach), then the output will be smoother, but the execution time for the kriging can increase significantly. By posting the input data using the post_samples module, and looking at the characteristics of the resulting kriged data using the Statistics module, the user can quickly analyze the characteristics and distribution of the kriging output for a given set of parameters, and test the effects of changing the kriging parameter values.

The ***Quadrant Search*** toggle changes the method by which data sample points are selected for inclusion in the kriging matrix. If this is on, the "Points" parameter switches to "Max Points in Quadrant". Searching is performed for each of the four quadrants surrounding the point to be kriged. Within each quadrant a maximum number of points (up to one-half of the total points) are selected. Then, points are taken sequentially from each quadrant up to the maximum number of total points or until all quadrant's points have been used. The panel display changes when this option is selected as shown above.

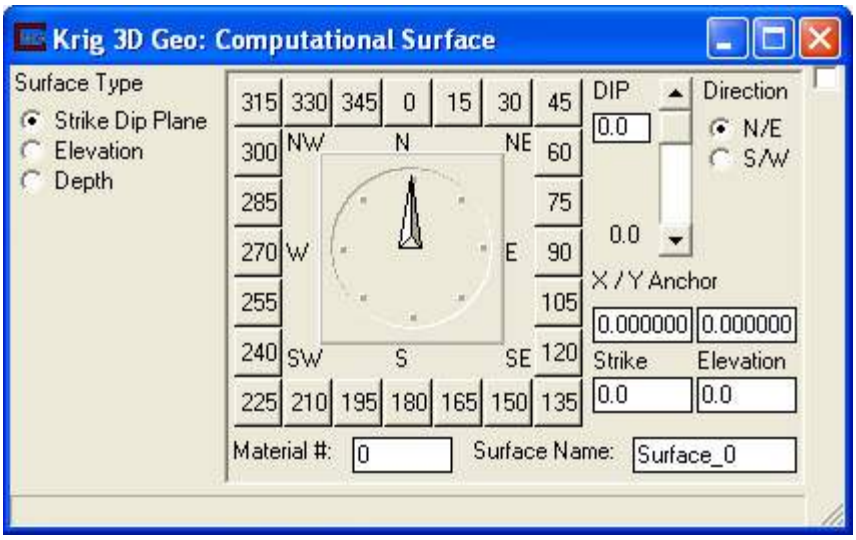
The type-in value for ***Use all data if number of samples is under*** is off by default, but this option gives the smoothest surfaces since all data is used for the kriging process. Sometimes using all points results in faster computation since only one (large) kriging matrix must be solved.

The ***Krig .GEO files in thickness space*** toggle off causes GEO files to be kriged like GMF files. Each surface get kriged independently of the other surface instead of being kriged in thickness space. This only applies to GEO files without the $W/$G flags.

The ***Pinch Factor*** parameter provides the ability to control where pinching occurs between positive thicknesses and borings having the pinch flag. It defaults to 1.0 which causes pinching to occur approximately half-way between positive thicknesses and borings having the pinch flag. When older applications created before version 9.5 are loaded they will have a value of 0.0 for backwards compatibility.

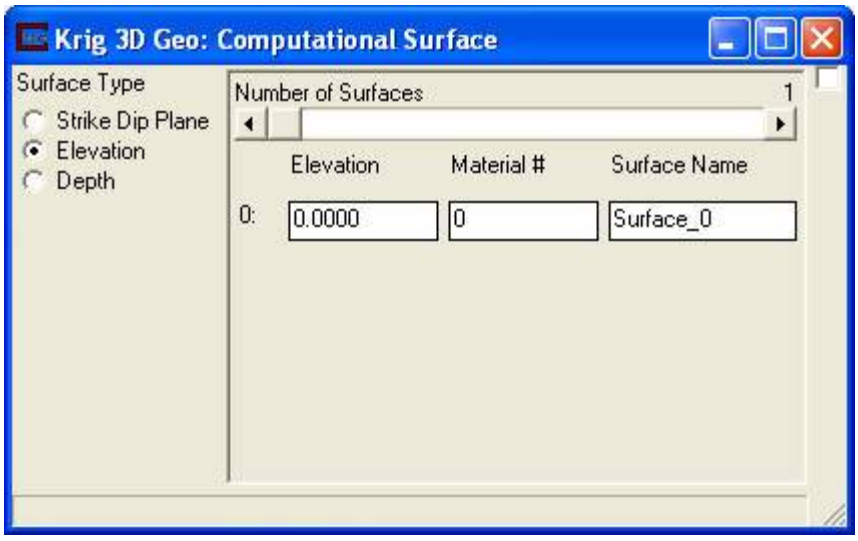
Interpolation Method:

- Kriging (this is the default and the only option available up to Version 9.0)
- IDW (Shepard)
- IDW (Franke/Nielson)
- Nearest Neighbor

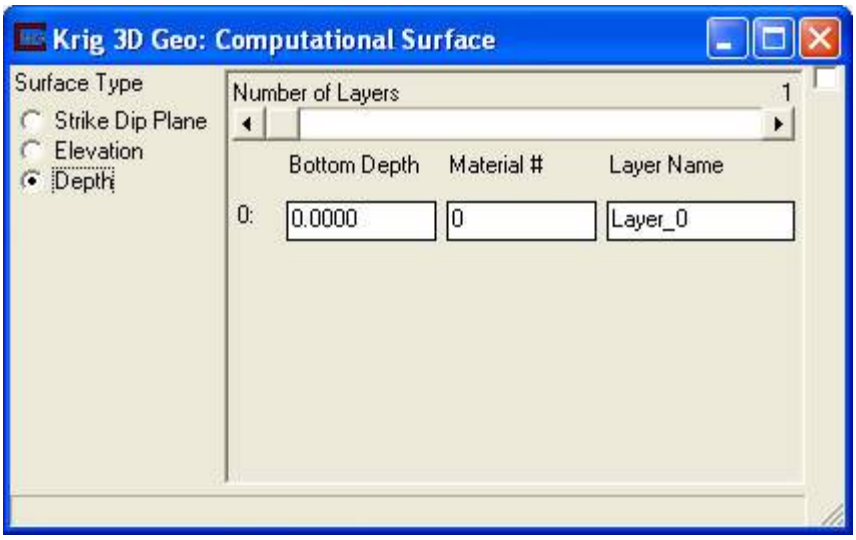


The **Create Computational Surface** panel is shown above. This method will create three different types of surfaces based upon the created grid. A file is required for a computational surface to be created. The first step is to create the desired grid using the Gridding Options subpanel discussed above.

The *Surface Type* radio box is the next step. There are three different types of surfaces that can be created. The Strike Dip Plane option consists of a single surface that can be rotated to match both strike and dip values. This option also requires an 3D coordinate point to center the rotation plane. This coordinate point is entered into the *X / Y Anchor* fields as well as the *Elevation* field. A *MaterialNumber* and *Surface Name* can also be entered for this computed surface.



The *Elevation* suface parameters can be seen in the image above. This surface type will create any number of surfaces all with the same X, Y coordinates created by the gridding options, but at set elevations. The *Material Number* and the *Surface Name* can be set for each surface.



The Depth surface paramets can be seen above. This method will Krige the selected file onto the grid created using the gridding options. It will then create any number of layers at specified depths below the top surface. The *Material Number* and *Layer Name* can be set for each layer.

Explanations of the various parameter settings and their effects on the visualization are provided in Workbook 2.

Krig_3D

General Module Function

Krig_3D models three-dimensional parameter distributions within rectilinear domains defined by the limits of the data set, or by the convex hull, rectilinear, or finite-difference grid extents of a geologic system modeled by Krig_3D_Geology. It creates a hexahedral finite-element grid with kriged nodal values of any scalar property and its kriged confidence level, and outputs a geometry whose elements can be filtered, sliced and rendered to view the color scaled parameter distribution on the element surfaces and/or edges. Krig_3D provides several convenient options for pre- and post-processing the input parameter values, and allows the user to consider anisotropy in the medium containing the property.

Krig_3D has the ability for this module to create uniform fields, and the ability to choose which data components you want to include in the output. There are a couple significant requirements for uniform fields. First, there cannot be geologic input (otherwise the cells could not be rectangular blocks). Second, Adaptive_Gridding must be turned off (otherwise the connectivity is not implicit).

Module Input Ports

1. **input0** (Blue/Grey/Green): This port can accept a data field from Krig_3D_Geology to krig data into geologic layers. Please note that if

any portions of the input geology is NULL, these cells will be omitted from the grid that is created. This can save memory and provide a means to cut (in a Lego fashion) along boundaries

2. **External_Grid** (Blue/Black): (Available only in MVS) This port (blue/black) allows the user to import a previously created grid. All data will be kriged to this grid.
3. **Read__data_file** (Yellow/Blue/Yellow): This port allows the sharing of file names between similar modules.
4. **External_Data** (Blue/Black): (Available only in MVS) This port allows the user to import a field contain data. This data will be kriged to the grid instead of using file data..

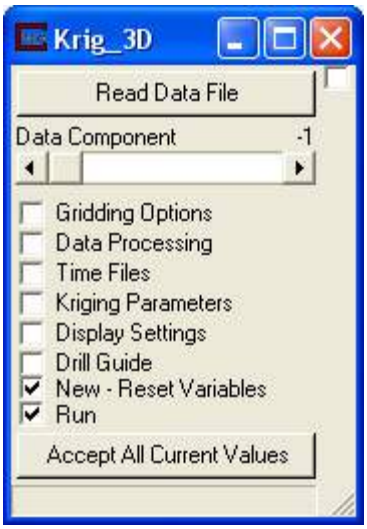
Module Output Ports

1. **Read__data_file** (Yellow/Blue/Yellow): This port allows the sharing of file names between similar modules.
2. **status_out** (Blue-Green) : This port outputs a string that has the current status message from the module. This string is updated as the module runs.
3. **Sphere_Out** (Red): This port outputs a renderable sphere to the view, this sphere represents the location of maximum uncertainty.
4. **Out1** (Blue/Black): This port outputs a 3D data field which can be input to any of the Subsetting and Processing modules which have the same color port

When the Plot Semivariogram option is selected three additional output ports will appear as in the above figure.

1. **Out2** (Red/Grey): This port outputs a renderable geometry of the calculated semivariogram.
2. **VG_fld_out**(Blue/Black): This port outputs a 3D data field representing the lines (semivariogram cloud pairs) and is provided primarily for connection to the axes module.
3. **VG_scale_fact** (Grey/Brown): This port is the Z_Scale of the semivariogram plot and is also provided primarily for connection to the axes module.

Module Control Panel



The control panel of Krig_3D is shown in the figure above.

The **Read Data File** button opens a *File Browser* as described for Krig_3D_Geology, but which lists the *.apdv;*.aidv; or*.vdf files that are present in the current directory shown in the directory window. The format of .apdv files is described in the apdv_file_format help topic.

Note that this module will not begin running until a valid analyte (e.g. chemistry) file has been selected, or a valid field containing data has been imported, and the **Accept All Current Values** button is pushed.

The **Data Component** slider allows the user to select which of the property values in the analyte (e.g. chemistry) file will be kriged by execution of the module. The default value is -1, which results in kriging of all of the data components in the file during a single execution of the module.

The "***Run***" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed (or the geology input changes). When it is off, the module will not run.

The "***New-Reset Variables***" toggle (on by default) resets all expert system calculated variables to zero before each run. This allows multiple calculations to be performed without tedious manual resetting of these variables. To change an expert system calculated variable, the toggle must be off.

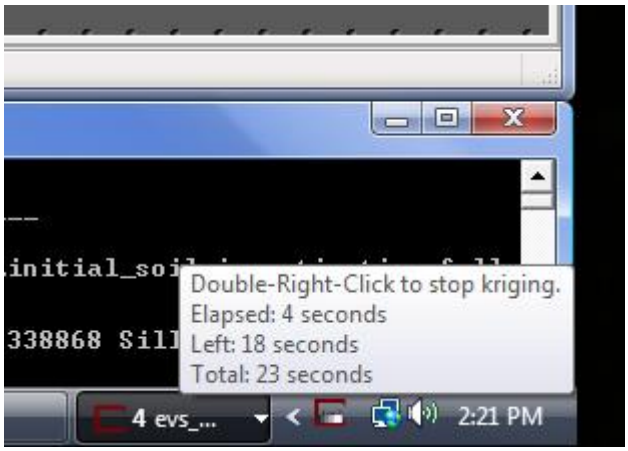
Module Parameter Subpanels

Krig_3D has six subpanels which allow the user to set the parameters used for preprocessing the input data, producing the semivariogram, executing kriging, and post processing kriged output. Clicking on either the check boxes next to the subpanel names, or on the names themselves will bring up the subpanel parameter screens.

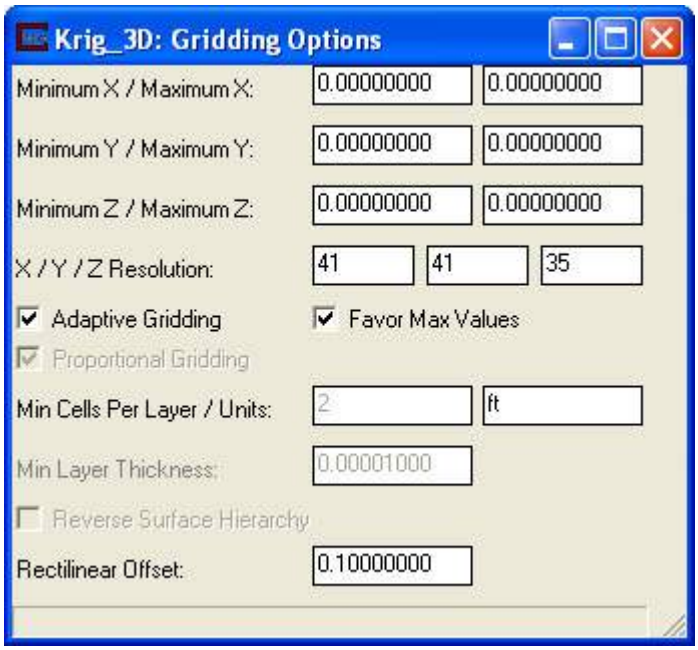
Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-

Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



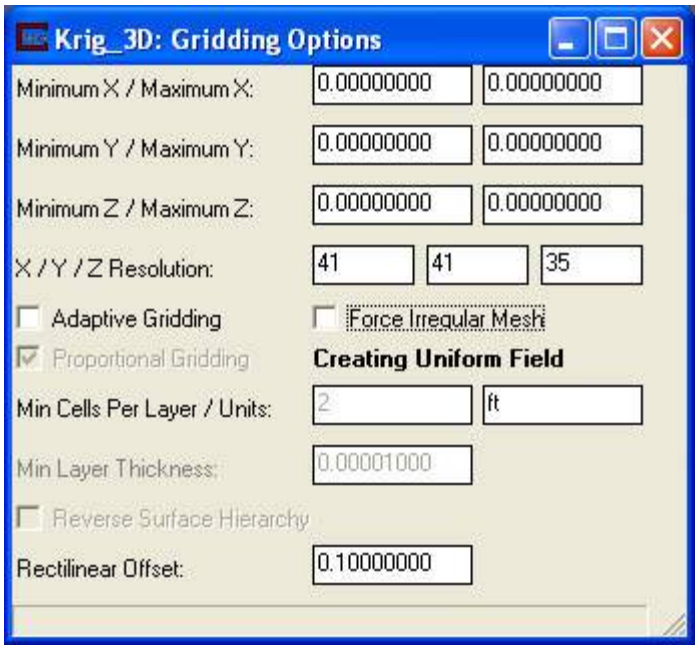
Gridding Options



The ***Gridding Options*** subpanel is shown in the figure above. It is used to specify all parameters that affect the grid exported from Krig_3D. The availability of many options depends on whether geologic input is available or if the grid is created totally within Krig_3D. The window above shows the options with no geologic input. NOTE: If an external grid is imported into Krig_3D this panel will be blank.

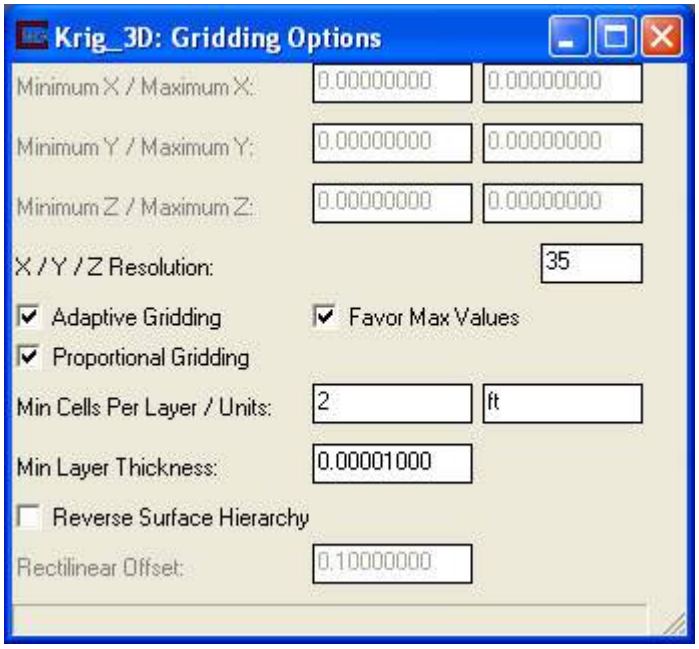
The ***Minimum X, Maximum X, Minimum Y, Maximum Y, Minimum Z and Maximum Z*** parameters allow the user to define the horizontal and vertical domain within the data set in which kriging of the parameter distribution will be completed. A value of 0 is the default for these parameters, which results in a model domain that is defined by the rectilinear bounds of the entire data

set when the module is run. Krig_3D utilizes a model domain that is bounded by the limits of the data set, unless the user specifies a different domain by setting the Min and Max Values for X, Y, and Z, or it is passed a model (or finite-difference) domain from Krig_3D_Geology. Utilizing the default extents effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points. Note that when a geologic model domain is passed to Krig_3D, the kriging domain is restricted to that domain regardless of what X, Y, or Z values are set in the Krig_3D inputs. The geologic model domain can be bounded by the convex hull of the data set, or by some specified offset (see discussion in Krig_3D_Geology). If the user is uncertain of the X, Y and Z limits of the data domain, the module should be run with the default 0 values, and upon completion of execution, the values in the X, Y, and Z input fields will be the min and max values of these parameters in the data set. The File_Statistics module can also be used to investigate the limits and distribution of values in the input data set, and the statistics module can be used to output the distribution of values in the kriged model.



Krig_3D has the ability for this module to create uniform fields, and the ability to choose which data components you want to include in the output. There are a couple significant requirements for uniform fields. First, there cannot be geologic input (otherwise the cells could not be rectangular blocks). Second, Adaptive_Gridding must be turned off (otherwise the connectivity is not implicit). Finally, when both of these conditions are satisfied, you turn off the toggle "***Force Irregular Mesh***", which is on by default (as shown below). This results in the output field being a uniform field. Uniform fields are more memory efficient and allow the use of several "field" specific modules like: orthoslice. They also have the ability to use the volume_renderer directly on the output!

The ***X / Y / Z Resolution*** parameters specify the number of grid nodes that will be included within the model domain. If Indicator_Geology is being passed a model domain from Krig_3D_Geology, then the x-y values are not used . The number of grid elements along any axis of the model is simply the axis Res value minus one, as every element has two bounding nodes along an axis. The default value for the X and Y Res parameters is 41, and the default value for the Z Res is 35. [note: these values were increased substantially as of version 6.0 in response to ever increasing computer speeds] However, the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user.



If geologic layers are being passed to Krig_3D, then the number of nodes specified in Z Res will be distributed over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain.

NOTE: If Proportional Gridding is toggled off then only Min Cells per Layer is enabled. In this case, only the Min Cells per Layer input will be used for each layer of the model.

The Adaptive Grid toggle (This feature available only in EVS PRO and MVS) refines the grid automatically to place grid nodes at all of the measured data points. The adaptive grid will result in a kriged parameter distribution that honors all of the measured data points exactly. This is the default domain used by Krig_3D. EVS PRO's improved Adaptive Gridding uses the following options and logic. It will perform grid refinement on any cell containing one or more samples and when there is more than one sample it will (usually) refine the two samples having the highest and lowest values. This assures that your kriged grid will always honor the total range of your data and ensures that data where low values are most important (e.g. pH) are handled optimally.

Proportional Gridding - A toggle for activating the same algorithm for apportioning nodes in the Z direction as used in Krig_3D. The number of nodes specified for the Z Resolution will be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) of the Krig_3D domain will be placed in each geologic layer.

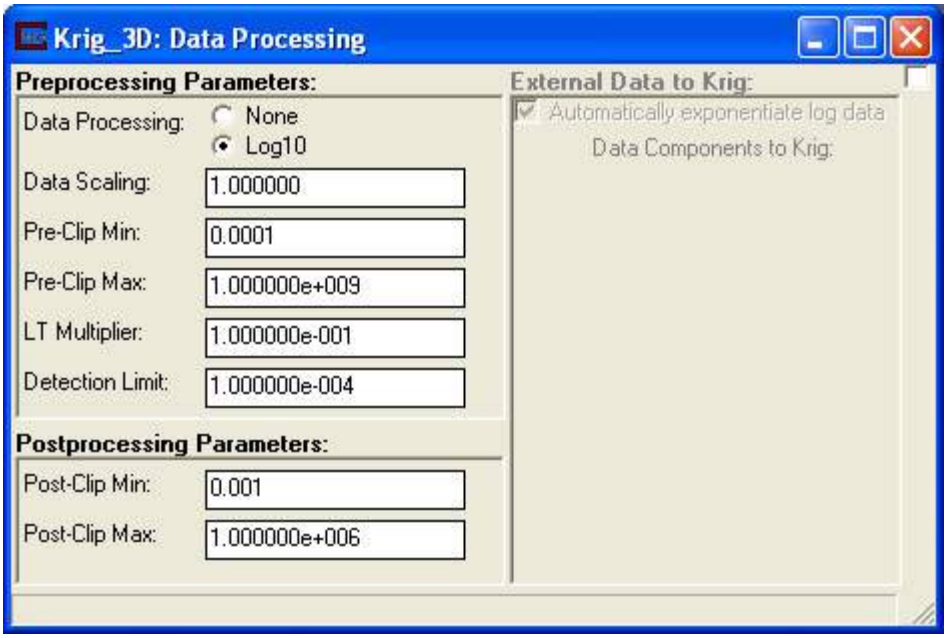
Min Cells per layer - A Type in for establishing a minimum number of cells (in the Z direction) per layer. The default is 2 cells which results in 3 nodes.

Units – This field contains the default coordinate units for the grid, if no coordinate units are specified in the file being read, these units will be used.

Reverse Surface Hierarchy – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

The ***Rectilinear Offset*** parameter is used to create an automatically generated rectilinear domain which is larger than the input data set by a user input percentage along all three axes. If 10% (0.1 default value) is used, the limits will be offset by 5% on all 6 sides of the rectilinear volume.

Data Processing



The ***Data Processing*** parameters subpanel is shown in the figure above. It is divided into three groups which are Preprocessing. Postprocessing, and External Data to Krig.

It is important to note that all preprocessing actions are applied directly to the data in memory, and that the original data file is not altered. However, all of the functions within Krig_3D and the modules downstream of Krig_3D will be using the preprocessed (and kriged) parameter distribution. The user

should refer back to the preprocessing subpanel when setting values for the filtering and display of the kriged data (i.e., to correctly specify whether the data has been log transformed, scaled, and/or clipped).

The ***Data Processing*** radio buttons allow the user to specify whether the data will be used as is, or will be processed to compute the log (base 10) of the parameter value before kriging. Note that if the log10 of the data is taken, a Clip Min value (which must be greater than 0) must be used to replace values in the data below the specified minimum value because the log function is undefined for values equal or less than zero.

The ***Data Scaling*** input field is used to specify a value by which all nodal data values will be multiplied before kriging. The default value is 1, but the user can specify any negative or positive value. This option is most commonly used to convert the units of the property being kriged, such as to convert concentrations in ppb to ppm or visa-versa. The user should bear in mind that any scaling of the data that is completed in Krig_3D will affect all downstream modules, which is especially important when using the Volume and Mass module downstream of Krig_3D.

The number entered into the ***Pre-Clip Min*** input field will be used during preprocessing to replace any nodal property value that is less than the specified number. When log processing is being used, the value of Clip Min must be a positive, non-zero value. Generally, Clip Min should be set to a value that is one-half to one-tenth of the lowest detection limit in the data set, unless the user wishes to make the influence of not detected values stronger. As an example, if the lowest detection limit is 0.1 (which is present in the data set as a 0), and the user sets Clip Min to 0.0001, the clipped non-detected values forces three orders of magnitude to be present between any detected value and the non-detected values.

The number entered into the ***Pre-Clip Max*** input field will be used during preprocessing to replace any nodal property value that is greater than the specified number. The clipping values can be used to lessen the importance placed on extremes of the data, or outlier data values, before kriging. The preprocessing functions can be used in various ways to investigate the kriging confidence levels within specified ranges or limits of the data sets. An example of this type of analysis is provided in the Sample Networks section.

The ***LT Multiplier*** value affects any file value with a preceeding "<" character. It will multiply these values by the set value.

The ***Detection Limit*** value affects any file values set with the "ND" or other non-detect flags (for a list of these flags open the help for the APDV file format). When the module encounters this flag in the file it will insert the a value equal to (Detection Limit * LT Multiplier).

Postprocessing of the data from Krig_3D affects the calculation of the kriging confidence and uncertainty estimates that are produced by the model, and allows the user to apply a filter for the data passed to all modules downstream of Krig_3D. Note that the postprocessing **does not** affect any of the other Semivariogram or kriging algorithms that execute in Krig_3D, so the user can experiment with different values of postprocessing parameters

to obtain the type of display desired, independent of the internal kriging process.

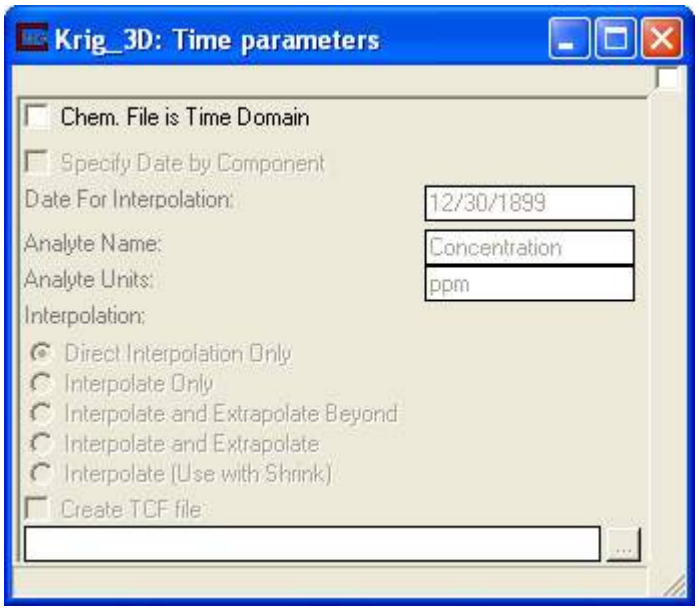
The ***Post-Clip Min*** parameter specifies the smallest nodal value that will be present in the data field output by Krig_3D. This parameter is useful for limiting or enhancing the effects of not detected values or outliers in a data set, and for optimizing the use of the dynamic color range used to represent the property distribution. Clip Min has a default value of 0.001, but can be set to any negative or positive value with magnitudes from -1.0 E09 to 1.0 E09. In general, good results are obtained by setting this value to the lowest property value or detection limit in the input data set. It is important to note that because not detected values are represented in the .apdv file as zeros, the Clip Min value can be used to strengthen the influence of small values or non-detects that are present in the data set. As an example, if the detection limit for a certain chemical analysis is 0.1 (which is entered into the .apdv file as 0), and the Clip Min is set to 0.0001, then this not detected value will have an effective influence on the kriged distribution near this data point that is three orders of magnitude stronger than the actual detection limit. The user should bear in mind that if mass or volume estimates are being made in the analysis, the Clip Min value could affect the estimates if considerable data are present that are being clipped. For volume estimates, if the specified lower bound of the concentrations of interest is well above the Clip Min value, then their will be no effects on the results. If the specified lower bound of concentrations for the volume estimate is lower than Clip Min, then the clipping will truncate the volume at the Clip Min value, and the estimated volume will be from the entire model domain.

The ***Post-Clip Max*** parameter specifies the largest nodal value that will be present in the data field output by Krig_3D. Again this parameter can be used to optimize the use of the dynamic color range when a data set has a few extremely high values, but the user is most interested in seeing the detailed changes in the distribution in some lower range of the data. It can also be used to investigate the kriging confidence or uncertainty levels near certain threshold values of a property. As an example, if the regulatory threshold for a parameter is 10 units, then the Clip Max parameter can be set to 10, and all kriged values greater than 10 will have the same strength in the calculation of the uncertainty levels in the kriged distribution. Clip Max has a default value of 1,000,000, but can be set to any negative or positive value with magnitude form -1.00E09 to 1.00E09.

When external data is being imported into the module the External Data to Krig window becomes active. This window allows the user to select which data components to krige by switching on or off the toggles associated with that data component. The ***Automatically exponentiate log data*** toggle will detect if the data being imported has been log processed and if it has

exponentiate it. This avoids double log processing the data with the Preprocessing option.

Time Files



(EVS PRO and MVS Only)

This panel is for analyte (e.g. chemistry) files (*.apdv, *.aidv) that are set up as time domain files (i.e. one analyte whose values are recorded over time).

The ***Chem. File is Time Domain*** toggle turns on date interpolation for time domain analyte (e.g. chemistry) files.

The ***Specify Dateby component*** toggle causes the module to ignore the user specifed date and instead use the component slider to select the date.

The ***Date for interpolation*** field is the date being interpolated to, for example if you have an analyte value of 2 on 1/01/05 and a value of 4 on 1/03/05 and the date is set to 1/02/05 with Direct Interpolation the value should be set to 3. The Date can be either set by hand or imported. This makes the module useful in a time loop.

In time domain files there is no place to set either the analyte name that has been kriged over time or the analyte units; these can be set in the ***Analyte Name*** and ***Analyte Units*** fields.

There are 5 different ***Interpolation*** methods that are available, each interpolation method is used to define how to interpolate when given unsampled times in a file.

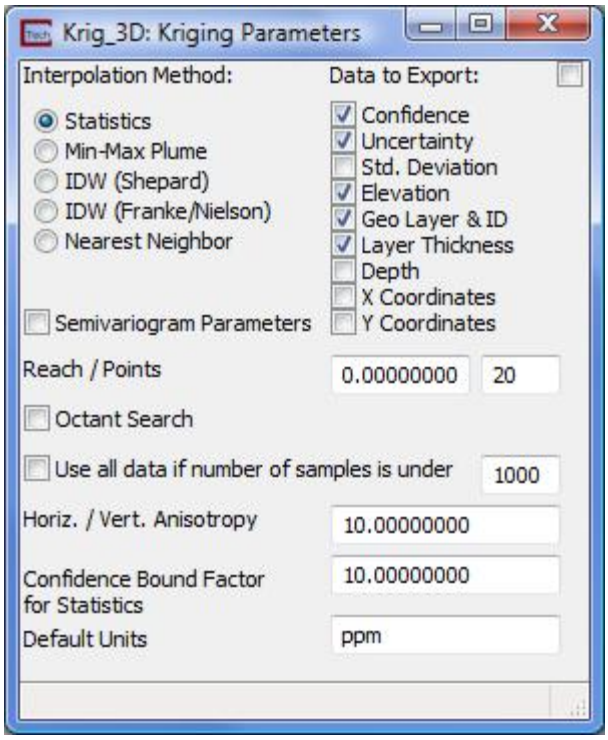
1. *Direct Interpolation Only:* This is the most basic interpolation method, and the most accurate in terms of representing the data as it has been entered. This method looks at the two dates surrounding the input Date. If either date is unsampled, the value for that sample will remain

unsampled and no interpolation will occur. If both dates have sampled values it will interpolate between them.

2. *Interpolate Only:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date.. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. If either search fails to find a useable dates then it will set the value for that time to a unsampled value. Else it will interpolate between the two found dates. This is useful for files that have a small amount of unsampled values.
3. *Interpolate and Extrapolate Beyond:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. If it does not find a sampled date after the input Date, it will extrapolate beyond the last useable date to the input Date.
4. *Interpolate and Extrapolate:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date. If it fails to find one it will extrapolate the first value backwards to the input Date. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. It will also extrapolate beyond the last valid date in the file.
5. *Interpolate(Use with Shrink):* This method uses the same interpolation method algorithm as Direct Interpolation. It is used for clarity when there is a post_samples or file_statistics module in the network that is using the Interpolate and Shrink to Unsampled option on the same file.

The ***Create TCF File*** toggle will cause the Krig_3D module to run in a loop, going through each component and creating an EFB file at that time. These EFB files will be linked together in the specified TCF file and can be used with the Read_TCF module for animation purposes. NOTE: This function will not work unless a TCF name is selected as well.

Kriging Parameters



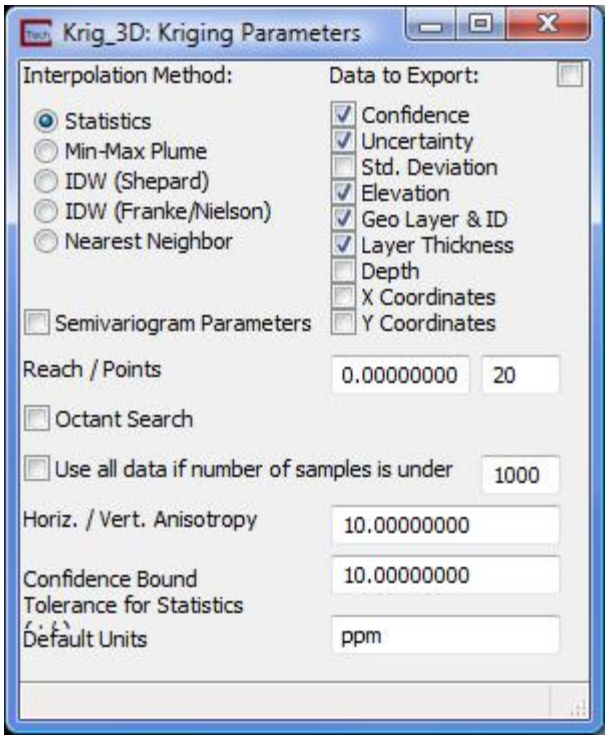
The ***Kriging Parameters*** subpanel of Krig_3D is shown in the figure above.

The ***Interpolation Method*** radio buttons determine:

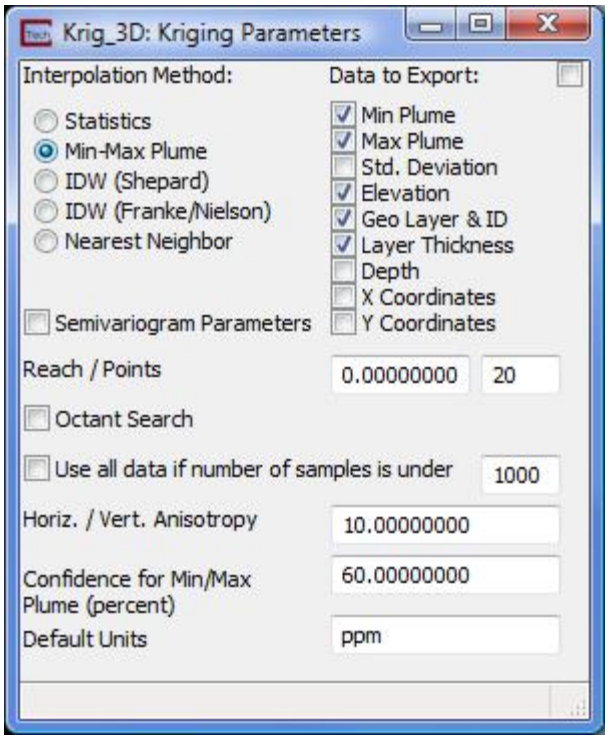
1. The type of statistical information which will be included in the nodal data components (statistics vs Min-Max Plume option)
2. IDW Shepard
3. IDW Franke/Nielson
4. Nearest Neighbor.

If ***Statistics*** is chosen, Each Concentration (for every chemical in the .apdv file) will have a corresponding Confidence (based on the Confidence Bound parameter) and Uncertainty. The display when Statistics is selected and the data is log processed is shown above.

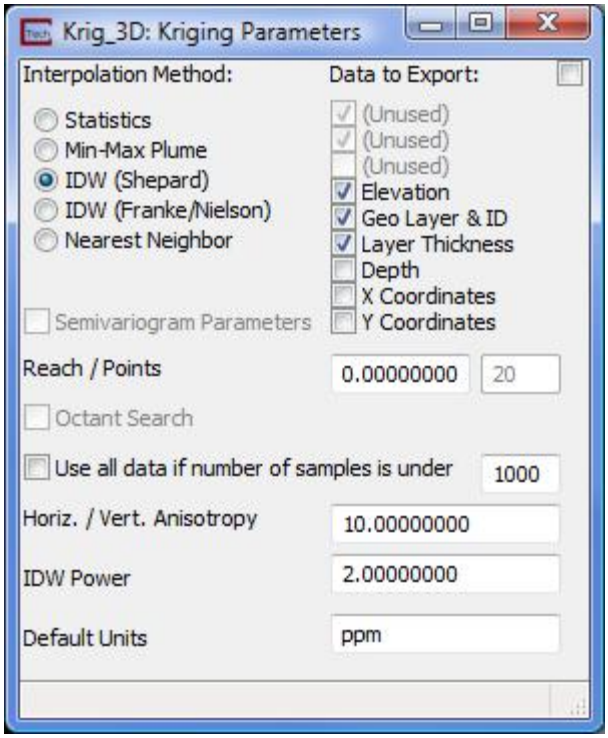
The display when Statistics is selected and the data is NOT log processed is shown below. In this case, the confidence bound is a tolerance vs. a factor.



If ***Min-Max Plume*** is chosen (This feature available only in EVS PRO and MVS) three different Concentration components will be calculated, the Nominal, Minimum and Maximum. These different distributions are determined based on the standard deviation and nominal concentration at each node based on the *Confidence* parameter which is expressed in percent (nominally 60%). Note that at a "confidence" of 50%, the nominal, minimum and maximum concentrations are identical.



If either ***Inverse Distance Weighted (IDW)***estimation method (Shepard and Franke/Nielson) is selected, kriging is not performed and one of these algorithms is used as an alternative to kriging in Krig_3D. This also affects the available parameters as show in the figure below:



Note: The "new" toggle on the main panel (on by default) resets all expert system calculated variables to zero before each run. This allows multiple calculations to be performed without tedious manual resetting of these variables. To change an expert system calculated variable, the toggle must be off.

The ***Reach*** input field defines the radial distance (in user units) from any given model node that the kriging module will look for data points to be included in the estimation of the model parameter at that node. The default value of reach is 0, which results in the module calculating a reach value which is approximately two-thirds of the longest distance between any two data points in the data set.

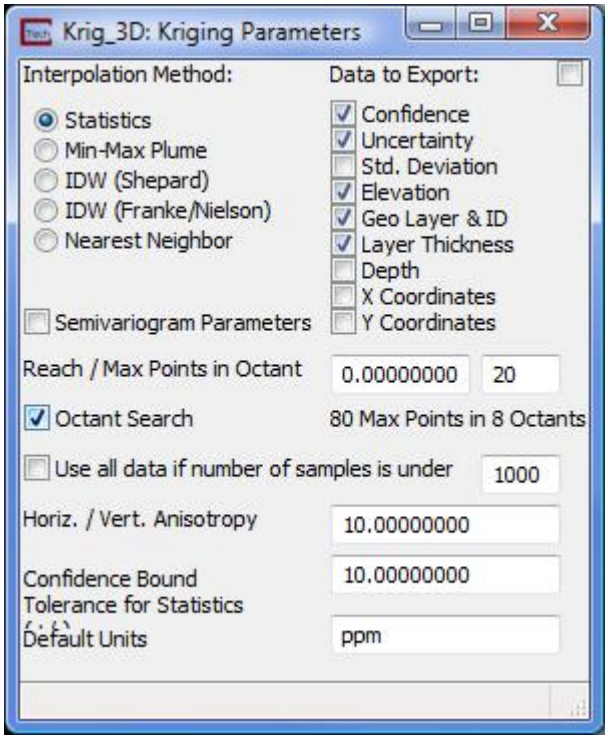
NOTE: If your Reach value is set so small that for some points in your grid, no samples fall within the Reach ellipsoid (it is a sphere distorted by the Horizontal/Vertical Anisotropy), you can receive an error message that will abort the kriging process. The error is *Matrix Singular in one or more areas.* This is unlikely to occur if you do not reset the Reach unless you set an extremely high Horizontal/Vertical Anisotropy. Highly clustered data with large regions without sampling can cause this to occur.

The only solutions are to increase Reach, reduce Horizontal/Vertical Anisotropy (if that is suspected) or revise the extent of your model in x-y and/or z.

The ***Points*** parameter defines the maximum number of data points (within the specified reach) that will be considered for the parameter estimation at a model node. The default value for points is 20, which generally provides reasonably smooth modeled parameter distributions. The effects of decreasing and increasing the values for reach and points on the model output are somewhat similar, but for different reasons. If the data have a fairly even spatial distribution throughout the domain, then increasing these values will generally include more of the input data points that will be used to krige the value for a given model node, and thus will result in smoother modeled data distributions. Decreasing the values of reach and points (in an evenly distributed data set) results in fewer input data points being used to calculate the parameter estimates at a given model node, and result in modeled distributions with greater variations across smaller areas.

The user should consider both the spatial distribution and the range of values in the input data set when deciding upon values for the reach and points parameters. If the specified reach is too small to allow the kriging module to locate at least one point within the search area, then no kriging can be done and the module will terminate with an error message..

If the user specifies a large number of points (that are within the specified reach), then the output will be smoother, but the execution time for the kriging can sometimes increase significantly. By posting the input data using the *post_samples* module, and looking at the characteristics of the resulting kriged data using the *statistics* module, the user can quickly analyze the characteristics and distribution of the kriging output for a given set of parameters, and test the effects of changing the kriging parameter values.



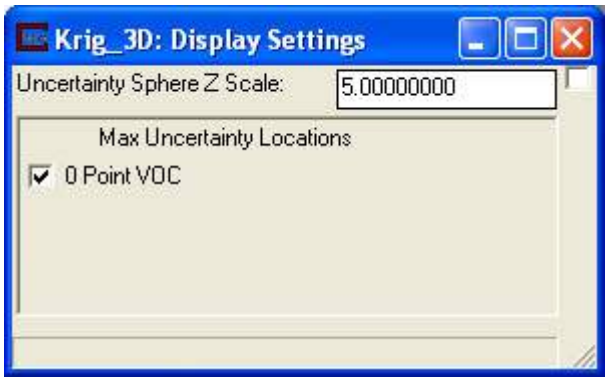
The ***Octant Search*** toggle changes the method by which data sample points are selected for inclusion in the kriging matrix. If this is on, the "Points"

parameter switches to "***Max Points in Octant***". Searching is performed for each of the eight Octants surrounding the point to be kriged. Within each octant a maximum number of points (up to one-fourth of the total points) are selected. Then, points are taken sequentially from each octant up to the maximum number of total points or until all octant's points have been used. The panel display changes when this option is selected as shown above. Octant Search is applicable for the Statistics and Min-Max Plume kriging modes.

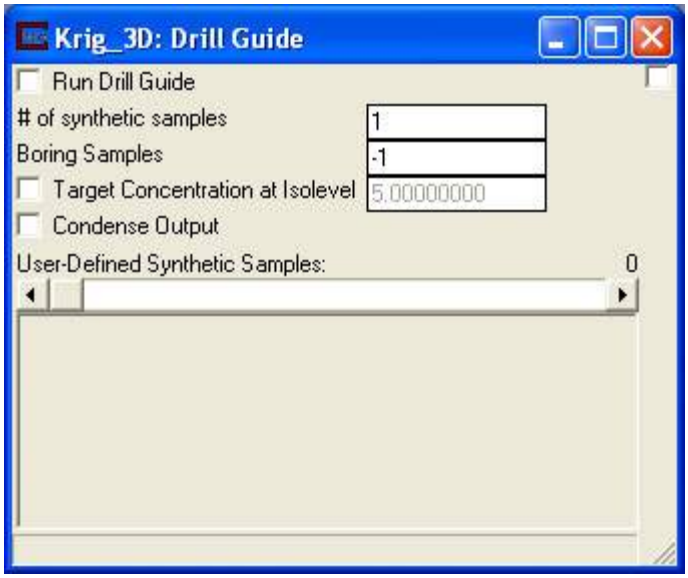
The type-in value for ***Use all data if number of samples is under***is off by default, but this option gives the smoothest plumes since all data is used for the kriging process. Sometimes using all points results in faster computation since only one (large) kriging matrix must be solved.

The ***Horiz./Vert. Anisotropy Ratio*** parameter allows the user to consider the effects of anisotropy in the conductivity of soil matrices to fluid flow. In most cases, geologic materials are deposited with platy clay minerals oriented horizontally, and thus flow of water in both the saturated and unsaturated zone can be slower in the vertical direction than in the horizontal direction. Also, ore deposition can occur along horizontal or vertical fault or fracture systems. Chemical constituents being transported with flowing fluids may therefore show a larger degree of spreading in one or the other direction. The Horiz./Vert. Anisotropy Ratio basically tells the kriging algorithm what multiplication factor should be used to apply biased weighting on data points in horizontal and vertical directions away from a given model node. The default value is 10, which allows data points in a horizontal direction away from a model node to influence the kriged value at that node 10 times more than data points an equal distance away in a vertical direction. However, the user can specify any positive number with a magnitude up to 100,000. When the property being modeled is not related to fluid flow or other processes that might be affected by matrix anisotropy, then this parameter should be set to 1.

The ***Confidence Bound*** parameter is used to specify what interval around the kriged model estimates the kriging confidence or uncertainty will apply to. The default value is 10, which essentially produces the confidence and uncertainty that the kriged data are within one order of magnitude of the "true" value. As an example, if the Confidence Bound is 10, the kriged property value at a node is 5, and the kriged confidence level at the node is 0.9, then 90% of the time, the "true" value of the kriged property at that node will be in the range of from 0.5 to 50 units. Additional discussions of confidence and uncertainty are provided in the Sample Networks.



The Display Settings window is shown in the image above. This window controls the display of the uncertainty sphere location. By default there is one uncertainty sphere for every analyte. The visibility of the spheres can be toggled on and off by click on the check box next to the name of the analyte. These spheres can also be vertically exaggerated to match any downstream scaling of the field itself.



The DrillGuide© panels is shown above. When the ***Run Drill Guide*** toggle is selected Krig_3D will run in a loop creating a synthetic boring at the maximum uncertainty location until it has reached the set ***# of synthetic samples***.

The ***Boring Samples*** field indicates how many samples create at each synthetic boring that is created.

The ***Target Concentration at subsetting level*** toggle, when selected, changes the location of the uncertainty sphere based upon the specified target concentration.

The ***Condense Output*** toggle, when selected, will minimize the number of strings displayed in the EVS console window.

The ***User-Defined Synthetic Samples*** slider is used to add points to the grid for drill guide purposes. The purpose of this slider is to help eliminate

areas of high uncertainty that cannot be drilled, for example under a building.

Krig_3D Module Hints

Manually adjusting kriging parameters

The user must ***un***check the new toggle in the main window in order to make manual adjustments to most parameters in this module. Otherwise the values will be calculated by EVS according to the spatial extents or distribution of the data.

Krig_3D

General Module Function

Krig_3D models three-dimensional parameter distributions within rectilinear domains defined by the limits of the data set, or by the convex hull, rectilinear, or finite-difference grid extents of a geologic system modeled by Krig_3D_Geology. It creates a hexahedral finite-element grid with kriged nodal values of any scalar property and its kriged confidence level, and outputs a geometry whose elements can be filtered, sliced and rendered to view the color scaled parameter distribution on the element surfaces and/or edges. Krig_3D provides several convenient options for pre- and post-processing the input parameter values, and allows the user to consider anisotropy in the medium containing the property.

Krig_3D has the ability for this module to create uniform fields, and the ability to choose which data components you want to include in the output. There are a couple significant requirements for uniform fields. First, there cannot be geologic input (otherwise the cells could not be rectangular blocks). Second, Adaptive_Gridding must be turned off (otherwise the connectivity is not implicit).

Module Input Ports

1. **input0** (Blue/Grey/Green): This port can accept a data field from Krig_3D_Geology to krig data into geologic layers. Please note that if any portions of the input geology is NULL, these cells will be omitted from the grid that is created. This can save memory and provide a means to cut (in a Lego fashion) along boundaries
2. **External_Grid** (Blue/Black): (Available only in MVS) This port (blue/black) allows the user to import a previously created grid. All data will be kriged to this grid.
3. **Read__data_file** (Yellow/Blue/Yellow): This port allows the sharing of file names between similar modules.
4. **External_Data** (Blue/Black): (Available only in MVS) This port allows the user to import a field contain data. This data will be kriged to the grid instead of using file data..

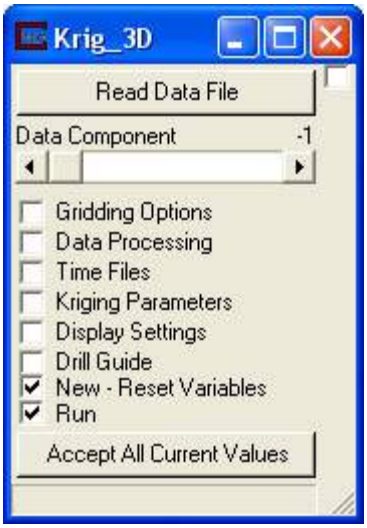
Module Output Ports

1. **Read__data_file** (Yellow/Blue/Yellow): This port allows the sharing of file names between similar modules.
2. **status_out** (Blue-Green) : This port outputs a string that has the current status message from the module. This string is updated as the module runs.
3. **Sphere_Out** (Red): This port outputs a renderable sphere to the view, this sphere represents the location of maximum uncertainty.
4. **Out1** (Blue/Black): This port outputs a 3D data field which can be input to any of the Subsetting and Processing modules which have the same color port

When the Plot Semivariogram option is selected three additional output ports will appear as in the above figure.

1. **Out2** (Red/Grey): This port outputs a renderable geometry of the calculated semivariogram.
2. **VG_fld_out**(Blue/Black): This port outputs a 3D data field representing the lines (semivariogram cloud pairs) and is provided primarily for connection to the axes module.
3. **VG_scale_fact** (Grey/Brown): This port is the Z_Scale of the semivariogram plot and is also provided primarily for connection to the axes module.

Module Control Panel



The control panel of Krig_3D is shown in the figure above.

The **Read Data File** button opens a *File Browser* as described for Krig_3D_Geology, but which lists the *.apdv;*.aidv; or*.vdf files that are

present in the current directory shown in the directory window. The format of .apdv files is described in the apdv_file_format help topic.

Note that this module will not begin running until a valid analyte (e.g. chemistry) file has been selected, or a valid field containing data has been imported, and the **Accept All Current Values** button is pushed.

The **Data Component** slider allows the user to select which of the property values in the analyte (e.g. chemistry) file will be kriged by execution of the module. The default value is -1, which results in kriging of all of the data components in the file during a single execution of the module.

The "***Run***" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed (or the geology input changes). When it is off, the module will not run.

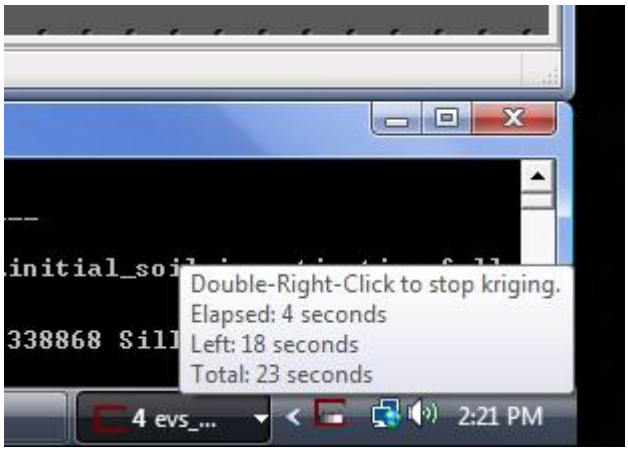
The "***New-Reset Variables***" toggle (on by default) resets all expert system calculated variables to zero before each run. This allows multiple calculations to be performed without tedious manual resetting of these variables. To change an expert system calculated variable, the toggle must be off.

Module Parameter Subpanels

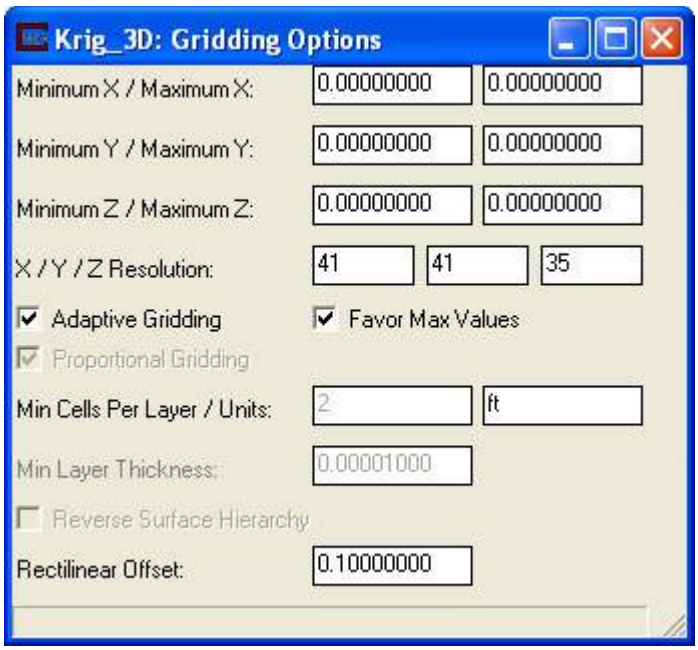
Krig_3D has six subpanels which allow the user to set the parameters used for preprocessing the input data, producing the semivariogram, executing kriging, and post processing kriged output. Clicking on either the check boxes next to the subpanel names, or on the names themselves will bring up the subpanel parameter screens.

Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.

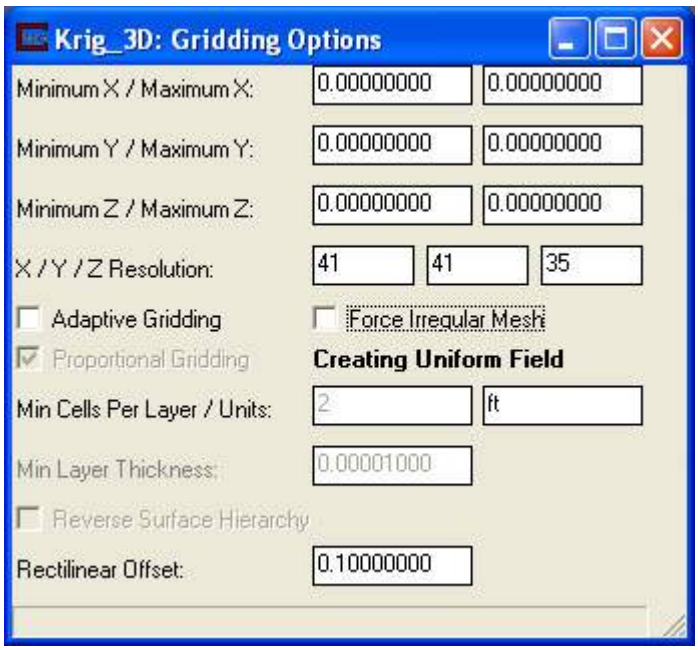


Gridding Options



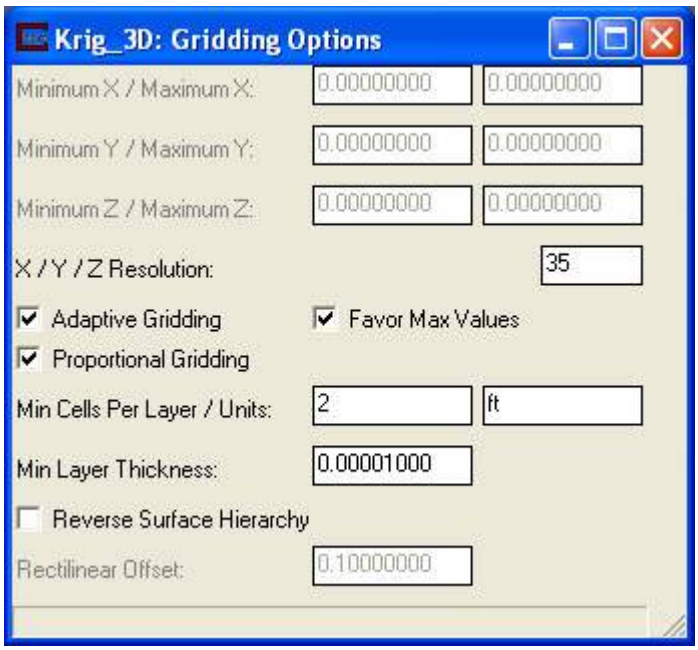
The ***Gridding Options*** subpanel is shown in the figure above. It is used to specify all parameters that affect the grid exported from Krig_3D. The availability of many options depends on whether geologic input is available or if the grid is created totally within Krig_3D. The window above shows the options with no geologic input. NOTE: If an external grid is imported into Krig_3D this panel will be blank.

The ***Minimum X, Maximum X, Minimum Y, Maximum Y, Minimum Z and Maximum Z*** parameters allow the user to define the horizontal and vertical domain within the data set in which kriging of the parameter distribution will be completed. A value of 0 is the default for these parameters, which results in a model domain that is defined by the rectilinear bounds of the entire data set when the module is run. Krig_3D utilizes a model domain that is bounded by the limits of the data set, unless the user specifies a different domain by setting the Min and Max Values for X, Y, and Z, or it is passed a model (or finite-difference) domain from Krig_3D_Geology. Utilizing the default extents effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points. Note that when a geologic model domain is passed to Krig_3D, the kriging domain is restricted to that domain regardless of what X, Y, or Z values are set in the Krig_3D inputs. The geologic model domain can be bounded by the convex hull of the data set, or by some specified offset (see discussion in Krig_3D_Geology). If the user is uncertain of the X, Y and Z limits of the data domain, the module should be run with the default 0 values, and upon completion of execution, the values in the X, Y, and Z input fields will be the min and max values of these parameters in the data set. The File_Statistics module can also be used to investigate the limits and distribution of values in the input data set, and the statistics module can be used to output the distribution of values in the kriged model.



Krig_3D has the ability for this module to create uniform fields, and the ability to choose which data components you want to include in the output. There are a couple significant requirements for uniform fields. First, there cannot be geologic input (otherwise the cells could not be rectangular blocks). Second, Adaptive_Gridding must be turned off (otherwise the connectivity is not implicit). Finally, when both of these conditions are satisfied, you turn off the toggle "***Force Irregular Mesh***", which is on by default (as shown below). This results in the output field being a uniform field. Uniform fields are more memory efficient and allow the use of several "field" specific modules like: orthoslice. They also have the ability to use the volume_renderer directly on the output!

The ***X / Y / Z Resolution*** parameters specify the number of grid nodes that will be included within the model domain. If Indicator_Geology is being passed a model domain from Krig_3D_Geology, then the x-y values are not used . The number of grid elements along any axis of the model is simply the axis Res value minus one, as every element has two bounding nodes along an axis. The default value for the X and Y Res parameters is 41, and the default value for the Z Res is 35. [note: these values were increased substantially as of version 6.0 in response to ever increasing computer speeds] However, the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user.



If geologic layers are being passed to Krig_3D, then the number of nodes specified in Z Res will be distributed over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain.

NOTE: If Proportional Gridding is toggled off then only Min Cells per Layer is enabled. In this case, only the Min Cells per Layer input will be used for each layer of the model.

The Adaptive Grid toggle (This feature available only in EVS PRO and MVS) refines the grid automatically to place grid nodes at all of the measured data points. The adaptive grid will result in a kriged parameter distribution that honors all of the measured data points exactly. This is the default domain used by Krig_3D. EVS PRO's improved Adaptive Gridding uses the following options and logic. It will perform grid refinement on any cell containing one or more samples and when there is more than one sample it will (usually) refine the two samples having the highest and lowest values. This assures that your kriged grid will always honor the total range of your data and ensures that data where low values are most important (e.g. pH) are handled optimally.

Proportional Gridding - A toggle for activating the same algorithm for apportioning nodes in the Z direction as used in Krig_3D. The number of nodes specified for the Z Resolution will be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) of the Krig_3D domain will be placed in each geologic layer.

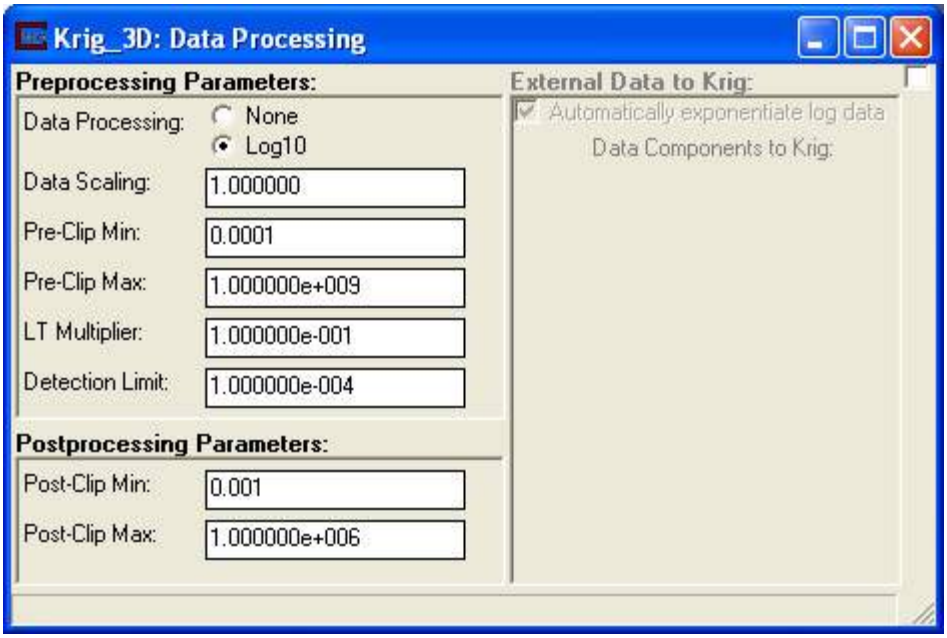
Min Cells per layer - A Type in for establishing a minimum number of cells (in the Z direction) per layer. The default is 2 cells which results in 3 nodes.

Units – This field contains the default coordinate units for the grid, if no coordinate units are specified in the file being read, these units will be used.

Reverse Surface Hierarchy – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

The ***Rectilinear Offset*** parameter is used to create an automatically generated rectilinear domain which is larger than the input data set by a user input percentage along all three axes. If 10% (0.1 default value) is used, the limits will be offset by 5% on all 6 sides of the rectilinear volume.

Data Processing



The ***Data Processing*** parameters subpanel is shown in the figure above. It is divided into three groups which are Preprocessing. Postprocessing, and External Data to Krig.

It is important to note that all preprocessing actions are applied directly to the data in memory, and that the original data file is not altered. However, all of the functions within Krig_3D and the modules downstream of Krig_3D will be using the preprocessed (and kriged) parameter distribution. The user should refer back to the preprocessing subpanel when setting values for the filtering and display of the kriged data (i.e., to correctly specify whether the data has been log transformed, scaled, and/or clipped).

The ***Data Processing*** radio buttons allow the user to specify whether the data will be used as is, or will be processed to compute the log (base 10) of the parameter value before kriging. Note that if the log10 of the data is taken, a Clip Min value (which must be greater than 0) must be used to replace values in the data below the specified minimum value because the log function is undefined for values equal or less than zero.

The ***Data Scaling*** input field is used to specify a value by which all nodal data values will be multiplied before kriging. The default value is 1, but the

user can specify any negative or positive value. This option is most commonly used to convert the units of the property being kriged, such as to convert concentrations in ppb to ppm or visa-versa. The user should bear in mind that any scaling of the data that is completed in Krig_3D will affect all downstream modules, which is especially important when using the Volume and Mass module downstream of Krig_3D.

The number entered into the ***Pre-Clip Min*** input field will be used during preprocessing to replace any nodal property value that is less than the specified number. When log processing is being used, the value of Clip Min must be a positive, non-zero value. Generally, Clip Min should be set to a value that is one-half to one-tenth of the lowest detection limit in the data set, unless the user wishes to make the influence of not detected values stronger. As an example, if the lowest detection limit is 0.1 (which is present in the data set as a 0), and the user sets Clip Min to 0.0001, the clipped non-detected values forces three orders of magnitude to be present between any detected value and the non-detected values.

The number entered into the ***Pre-Clip Max*** input field will be used during preprocessing to replace any nodal property value that is greater than the specified number. The clipping values can be used to lessen the importance placed on extremes of the data, or outlier data values, before kriging. The preprocessing functions can be used in various ways to investigate the kriging confidence levels within specified ranges or limits of the data sets. An example of this type of analysis is provided in the Sample Networks section.

The ***LT Multiplier*** value affects any file value with a preceeding "<" character. It will multiply these values by the set value.

The ***Detection Limit*** value affects any file values set with the "ND" or other non-detect flags (for a list of these flags open the help for the APDV file format). When the module encounters this flag in the file it will insert the a value equal to (Detection Limit * LT Multiplier).

Postprocessing of the data from Krig_3D affects the calculation of the kriging confidence and uncertainty estimates that are produced by the model, and allows the user to apply a filter for the data passed to all modules downstream of Krig_3D. Note that the postprocessing **does not** affect any of the other Semivariogram or kriging algorithms that execute in Krig_3D, so the user can experiment with different values of postprocessing parameters to obtain the type of display desired, independent of the internal kriging process.

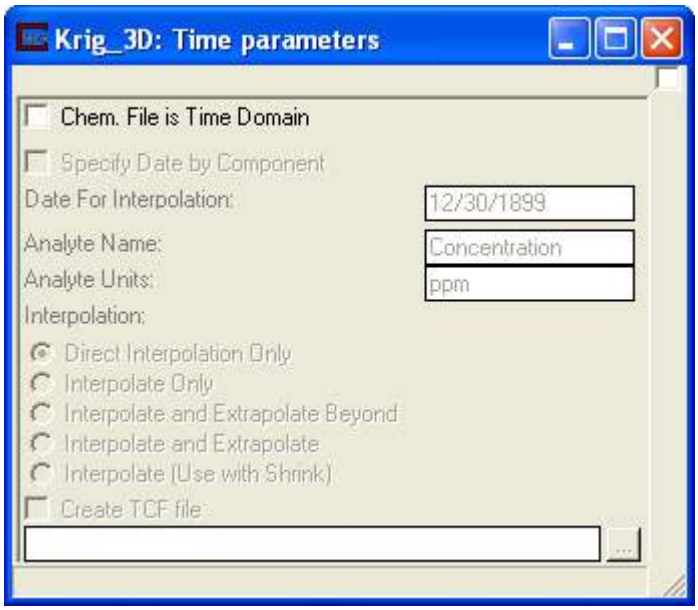
The ***Post-Clip Min*** parameter specifies the smallest nodal value that will be present in the data field output by Krig_3D. This parameter is useful for limiting or enhancing the effects of not detected values or outliers in a data set, and for optimizing the use of the dynamic color range used to represent the property distribution. Clip Min has a default value of 0.001, but can be set to any negative or positive value with magnitudes from -1.0 E09 to 1.0 E09. In general, good results are obtained by setting this value to the lowest property value or detection limit in the input data set. It is important to note that because not detected values are represented in the .apdv file as zeros, the Clip Min value can be used to strengthen the influence of small values or

non-detects that are present in the data set. As an example, if the detection limit for a certain chemical analysis is 0.1 (which is entered into the .apdv file as 0), and the Clip Min is set to 0.0001, then this not detected value will have an effective influence on the kriged distribution near this data point that is three orders of magnitude stronger than the actual detection limit. The user should bear in mind that if mass or volume estimates are being made in the analysis, the Clip Min value could affect the estimates if considerable data are present that are being clipped. For volume estimates, if the specified lower bound of the concentrations of interest is well above the Clip Min value, then their will be no effects on the results. If the specified lower bound of concentrations for the volume estimate is lower than Clip Min, then the clipping will truncate the volume at the Clip Min value, and the estimated volume will be from the entire model domain.

The ***Post-Clip Max*** parameter specifies the largest nodal value that will be present in the data field output by Krig_3D. Again this parameter can be used to optimize the use of the dynamic color range when a data set has a few extremely high values, but the user is most interested in seeing the detailed changes in the distribution in some lower range of the data. It can also be used to investigate the kriging confidence or uncertainty levels near certain threshold values of a property. As an example, if the regulatory threshold for a parameter is 10 units, then the Clip Max parameter can be set to 10, and all kriged values greater than 10 will have the same strength in the calculation of the uncertainty levels in the kriged distribution. Clip Max has a default value of 1,000,000, but can be set to any negative or positive value with magnitude form -1.00E09 to 1.00E09.

When external data is being imported into the module the External Data to Krig window becomes active. This window allows the user to select which data components to krige by switching on or off the toggles associated with that data component. The ***Automatically exponentiate log data*** toggle will detect if the data being imported has been log processed and if it has exponentiate it. This avoids double log processing the data with the Preprocessing option.

Time Files



(EVS PRO and MVS Only)

This panel is for analyte (e.g. chemistry) files (*.apdv, *.aidv) that are set up as time domain files (i.e. one analyte whose values are recorded over time).

The ***Chem. File is Time Domain*** toggle turns on date interpolation for time domain analyte (e.g. chemistry) files.

The ***Specify Dateby component*** toggle causes the module to ignore the user specifed date and instead use the component slider to select the date.

The ***Date for interpolation*** field is the date being interpolated to, for example if you have an analyte value of 2 on 1/01/05 and a value of 4 on 1/03/05 and the date is set to 1/02/05 with Direct Interpolation the value should be set to 3. The Date can be either set by hand or imported. This makes the module useful in a time loop.

In time domain files there is no place to set either the analyte name that has been kriged over time or the analyte units; these can be set in the ***Analyte Name*** and ***Analyte Units*** fields.

There are 5 different ***Interpolation*** methods that are available, each interpolation method is used to define how to interpolate when given unsampled times in a file.

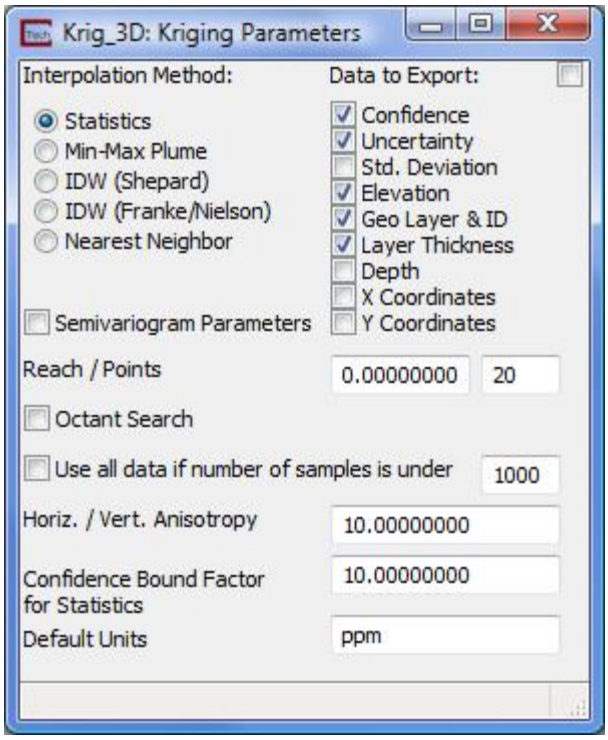
1. *Direct Interpolation Only:* This is the most basic interpolation method, and the most accurate in terms of representing the data as it has been entered. This method looks at the two dates surrounding the input Date. If either date is unsampled, the value for that sample will remain unsampled and no interpolation will occur. If both dates have sampled values it will interpolate between them.
2. *Interpolate Only:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date.. If the date after the input Date is unsampled it

will look forward through the time columns until it finds a sampled date. If either search fails to find a useable dates then it will set the value for that time to a unsampled value. Else it will interpolate between the two found dates. This is useful for files that have a small amount of unsampled values.

3. *Interpolate and Extrapolate Beyond:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. If it does not find a sampled date after the input Date, it will extrapolate beyond the last useable date to the input Date.
4. *Interpolate and Extrapolate:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date. If it fails to find one it will extrapolate the first value backwards to the input Date. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. It will also extrapolate beyond the last valid date in the file.
5. *Interpolate(Use with Shrink):* This method uses the same interpolation method algorithm as Direct Interpolation. It is used for clarity when there is a post_samples or file_statistics module in the network that is using the Interpolate and Shrink to Unsampled option on the same file.

The ***Create TCF File*** toggle will cause the Krig_3D module to run in a loop, going through each component and creating an EFB file at that time. These EFB files will be linked together in the specified TCF file and can be used with the Read_TCF module for animation purposes. NOTE: This function will not work unless a TCF name is selected as well.

Kriging Parameters



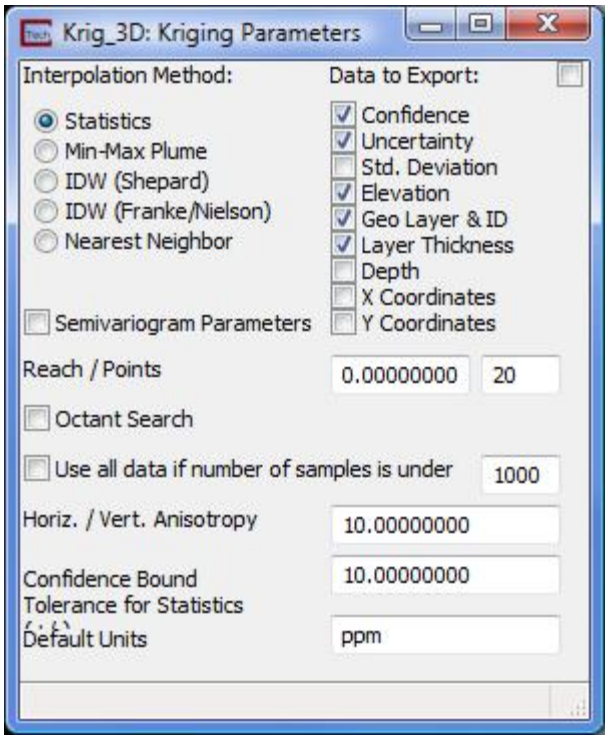
The ***Kriging Parameters*** subpanel of Krig_3D is shown in the figure above.

The ***Interpolation Method*** radio buttons determine:

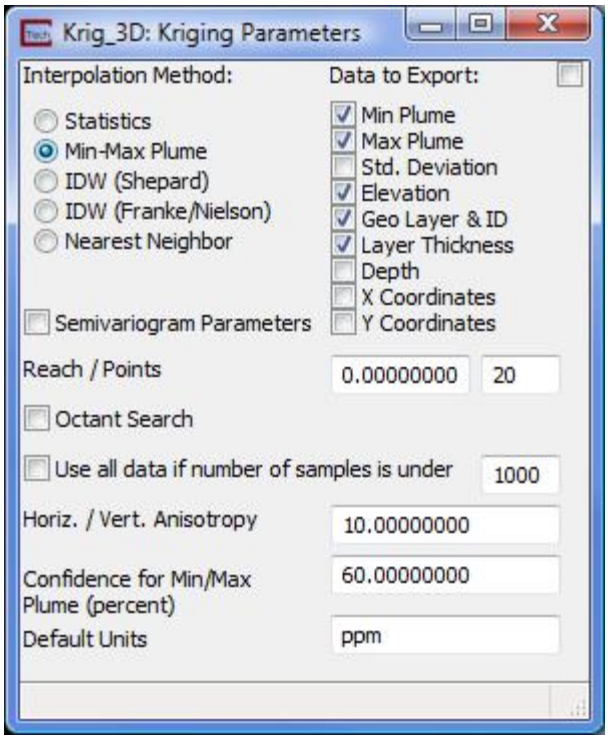
1. The type of statistical information which will be included in the nodal data components (statistics vs Min-Max Plume option)
2. IDW Shepard
3. IDW Franke/Nielson
4. Nearest Neighbor.

If ***Statistics*** is chosen, Each Concentration (for every chemical in the .apdv file) will have a corresponding Confidence (based on the Confidence Bound parameter) and Uncertainty. The display when Statistics is selected and the data is log processed is shown above.

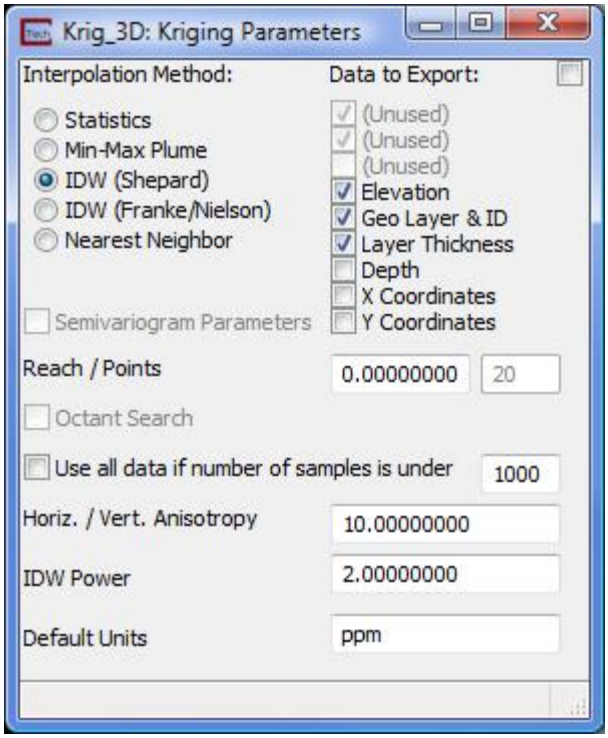
The display when Statistics is selected and the data is NOT log processed is shown below. In this case, the confidence bound is a tolerance vs. a factor.



If ***Min-Max Plume*** is chosen (This feature available only in EVS PRO and MVS) three different Concentration components will be calculated, the Nominal, Minimum and Maximum. These different distributions are determined based on the standard deviation and nominal concentration at each node based on the *Confidence* parameter which is expressed in percent (nominally 60%). Note that at a "confidence" of 50%, the nominal, minimum and maximum concentrations are identical.



If either ***Inverse Distance Weighted (IDW)***estimation method (Shepard and Franke/Nielson) is selected, kriging is not performed and one of these algorithms is used as an alternative to kriging in Krig_3D. This also affects the available parameters as show in the figure below:



Note: The "new" toggle on the main panel (on by default) resets all expert system calculated variables to zero before each run. This allows multiple calculations to be performed without tedious manual resetting of these variables. To change an expert system calculated variable, the toggle must be off.

The ***Reach*** input field defines the radial distance (in user units) from any given model node that the kriging module will look for data points to be included in the estimation of the model parameter at that node. The default value of reach is 0, which results in the module calculating a reach value which is approximately two-thirds of the longest distance between any two data points in the data set.

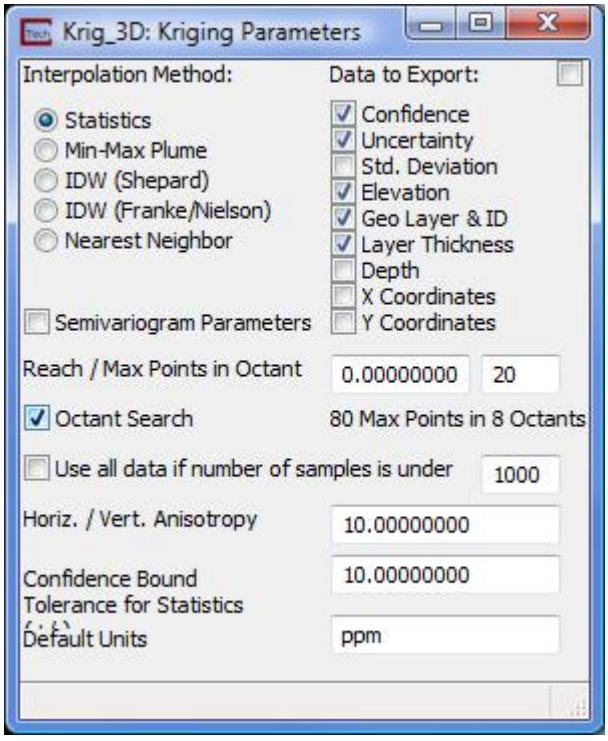
NOTE: If your Reach value is set so small that for some points in your grid, no samples fall within the Reach ellipsoid (it is a sphere distorted by the Horizontal/Vertical Anisotropy), you can receive an error message that will abort the kriging process. The error is *Matrix Singular in one or more areas.* This is unlikely to occur if you do not reset the Reach unless you set an extremely high Horizontal/Vertical Anisotropy. Highly clustered data with large regions without sampling can cause this to occur.

The only solutions are to increase Reach, reduce Horizontal/Vertical Anisotropy (if that is suspected) or revise the extent of your model in x-y and/or z.

The ***Points*** parameter defines the maximum number of data points (within the specified reach) that will be considered for the parameter estimation at a model node. The default value for points is 20, which generally provides reasonably smooth modeled parameter distributions. The effects of decreasing and increasing the values for reach and points on the model output are somewhat similar, but for different reasons. If the data have a fairly even spatial distribution throughout the domain, then increasing these values will generally include more of the input data points that will be used to krige the value for a given model node, and thus will result in smoother modeled data distributions. Decreasing the values of reach and points (in an evenly distributed data set) results in fewer input data points being used to calculate the parameter estimates at a given model node, and result in modeled distributions with greater variations across smaller areas.

The user should consider both the spatial distribution and the range of values in the input data set when deciding upon values for the reach and points parameters. If the specified reach is too small to allow the kriging module to locate at least one point within the search area, then no kriging can be done and the module will terminate with an error message..

If the user specifies a large number of points (that are within the specified reach), then the output will be smoother, but the execution time for the kriging can sometimes increase significantly. By posting the input data using the *post_samples* module, and looking at the characteristics of the resulting kriged data using the *statistics* module, the user can quickly analyze the characteristics and distribution of the kriging output for a given set of parameters, and test the effects of changing the kriging parameter values.



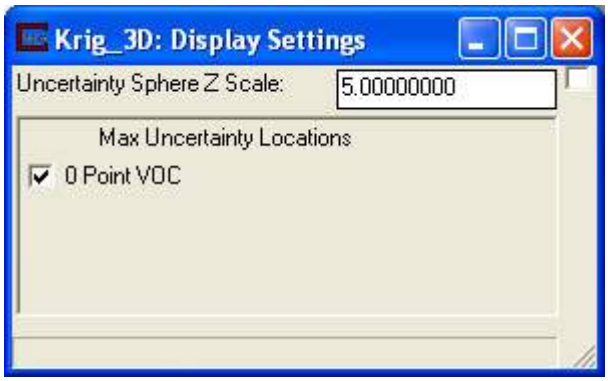
The ***Octant Search*** toggle changes the method by which data sample points are selected for inclusion in the kriging matrix. If this is on, the "Points"

parameter switches to "***Max Points in Octant***". Searching is performed for each of the eight Octants surrounding the point to be kriged. Within each octant a maximum number of points (up to one-fourth of the total points) are selected. Then, points are taken sequentially from each octant up to the maximum number of total points or until all octant's points have been used. The panel display changes when this option is selected as shown above. Octant Search is applicable for the Statistics and Min-Max Plume kriging modes.

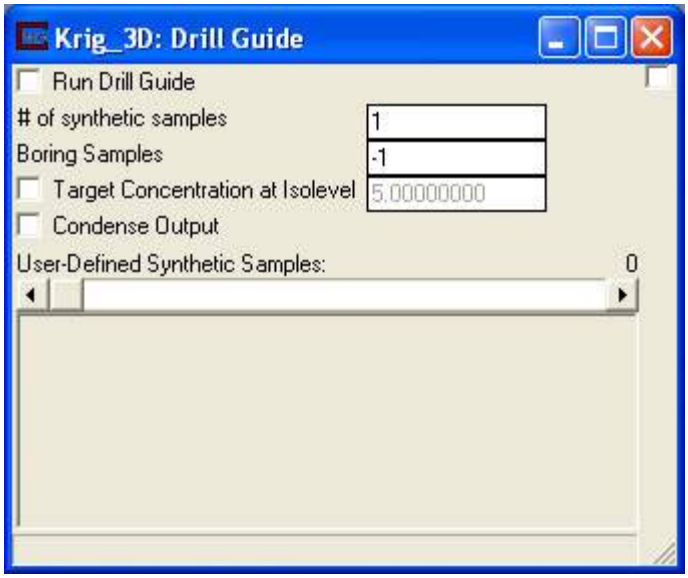
The type-in value for ***Use all data if number of samples is under***is off by default, but this option gives the smoothest plumes since all data is used for the kriging process. Sometimes using all points results in faster computation since only one (large) kriging matrix must be solved.

The ***Horiz./Vert. Anisotropy Ratio*** parameter allows the user to consider the effects of anisotropy in the conductivity of soil matrices to fluid flow. In most cases, geologic materials are deposited with platy clay minerals oriented horizontally, and thus flow of water in both the saturated and unsaturated zone can be slower in the vertical direction than in the horizontal direction. Also, ore deposition can occur along horizontal or vertical fault or fracture systems. Chemical constituents being transported with flowing fluids may therefore show a larger degree of spreading in one or the other direction. The Horiz./Vert. Anisotropy Ratio basically tells the kriging algorithm what multiplication factor should be used to apply biased weighting on data points in horizontal and vertical directions away from a given model node. The default value is 10, which allows data points in a horizontal direction away from a model node to influence the kriged value at that node 10 times more than data points an equal distance away in a vertical direction. However, the user can specify any positive number with a magnitude up to 100,000. When the property being modeled is not related to fluid flow or other processes that might be affected by matrix anisotropy, then this parameter should be set to 1.

The ***Confidence Bound*** parameter is used to specify what interval around the kriged model estimates the kriging confidence or uncertainty will apply to. The default value is 10, which essentially produces the confidence and uncertainty that the kriged data are within one order of magnitude of the "true" value. As an example, if the Confidence Bound is 10, the kriged property value at a node is 5, and the kriged confidence level at the node is 0.9, then 90% of the time, the "true" value of the kriged property at that node will be in the range of from 0.5 to 50 units. Additional discussions of confidence and uncertainty are provided in the Sample Networks.



The Display Settings window is shown in the image above. This window controls the display of the uncertainty sphere location. By default there is one uncertainty sphere for every analyte. The visibility of the spheres can be toggled on and off by click on the check box next to the name of the analyte. These spheres can also be vertically exaggerated to match any downstream scaling of the field itself.



The DrillGuide© panels is shown above. When the ***Run Drill Guide*** toggle is selected Krig_3D will run in a loop creating a synthetic boring at the maximum uncertainty location until it has reached the set ***# of synthetic samples***.

The ***Boring Samples*** field indicates how many samples create at each synthetic boring that is created.

The ***Target Concentration at subsetting level*** toggle, when selected, changes the location of the uncertainty sphere based upon the specified target concentration.

The ***Condense Output*** toggle, when selected, will minimize the number of strings displayed in the EVS console window.

The ***User-Defined Synthetic Samples*** slider is used to add points to the grid for drill guide purposes. The purpose of this slider is to help eliminate

areas of high uncertainty that cannot be drilled, for example under a building.

Krig_3D Module Hints

Manually adjusting kriging parameters

The user must ***un***check the new toggle in the main window in order to make manual adjustments to most parameters in this module. Otherwise the values will be calculated by EVS according to the spatial extents or distribution of the data.

Krig_2D

General Module Function

The Krig_2D module models two-dimensional parameter distributions within domains defined by the convex hull of a data set, within a rectilinear domain with equally spaced nodes, or a technique called *Adaptive Grid*(This feature available only in EVS-PRO and MVS) which uses the convex hull option, and then subdivides individual elements to place a "kriged" node at the location of each input data sample. This guarantees that the output will accurately reflect the input at all measured locations (i.e. the maximum in the output will be the maximum of the input). This option is now the default gridding option. Krig_2D creates a layer of quadrilateral (4 node) elements in which each node is assigned the kriged parameter value, and optionally its associated kriging confidence level and uncertainty. Upon execution of the module, Krig_2D produces a new input data file with a synthetic boring at the location of maximum uncertainty calculated from the previous kriging estimates, which can then be rerun to find the next area of highest uncertainty. The naming of the "DrillGuide©" file which is created when Krig_2D is run with all types of analyte (e.g. chemistry) files ends in apdv1, apdv2, apdv3, etc. the output file name will be .apdv2, apdv3, apdv4.... There are no limits to the number of cycles that may be run.

The use of Krig_2D to perform analytically guided site assessment is covered in detail in Workbook 2: DrillGuide© Analytically Guided Site Assessment.

This process can be continued as many times as desired to define the number and placement of additional borings that are needed to reduce the maximum uncertainty in the modeled domain to a user specified level. The features of Krig_2D make it particularly useful for optimizing the benefits obtained from environmental sampling or ore drilling programs. Krig_2D also provides some special data processing options that are unique to it, which allow it to extract 2-dimensional data sets from input data files that contain three-dimensional data. This functionality allows it to use the same .apdv files as all of the other EVS input and kriging modules, and allows detailed analyses of property characteristics along 2-dimensional planes through the data set. Krig_2D also provides the user with options to magnify or distort the resulting grid by the kriged value of the property at each grid node. Krig_2D also allows the user to automatically clamp the data distribution to a

specified level along a boundary that can be offset from the convex hull of the data domain by a user defined amount.

Module Input Ports

1. External Grid (Blue/Black): (Available only in MVS) This port allows a previously created grid to be imported into the module. All data will be kriged onto this grid and the input file will be ignored for gridding purposes.
2. Filename (Yellow/Blue/Yellow): This port allows for the sharing of filenames between related modules.
3. External Data (Blue/Black): (Available only in MVS) This port allows a previously created data field to be imported into the module. This data will be kriged onto the grid and the input file will be ignored for data purposes.

Module Output Ports

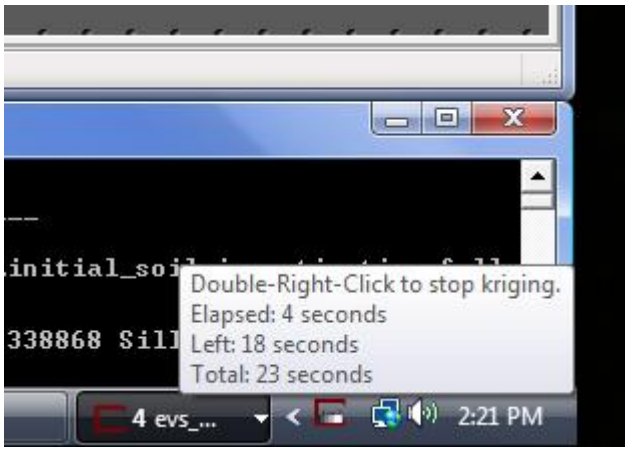
1. Filename (Yellow/Blue/Yellow): This port allows for the sharing of filenames between related modules.
2. Surface Output (Blue/Black): This port outputs a 2D data field that can be input to any of the Subsetting and Processing modules that have the same color input port.
3. Surf_Out (Red): This port outputs a geometry of the component surface which can be input directly to the viewer. Connecting the geometry port directly to the viewer allows use of the clamping functions in Krig_2D Data Processing to quickly investigate the distributions of components within the specified range.

When the Plot Semivariogram option is selected three additional output ports will appear as in the above figure.

1. Out2 (Red/Grey): This port outputs a geometry of the calculated semivariogram.
2. status_out (Blue-Green) : This port outputs a string that has the current status message from the module. This string is updated as the module runs.
3. VG_fld_out (Blue/Black): This port outputs a 3D data field representing the lines (semivariogram cloud pairs) and is provided primarily for connection to the axes module.
4. VG_scale_fact (Grey/Brown): This port is the Z_Scale of the semivariogram plot and is also provided primarily for connection to the axes module.

Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel

The control panel of Krig_2D is shown in the figure above.

The **Read Data File** button opens a *File Browser* which lists the *.apdv;*.aidv; or*.geo files that are present in the current directory shown in the directory window. The format of .apdv files is described in the apdv_file_format help topic.

Note that this module will not begin running until a valid analyte (e.g. chemistry) file has been selected, or a valid field containing data has been imported, and the **Accept All Current Values** button is pushed.

The **Data Component** slider allows the user to select which of the property values in the analyte (e.g. chemistry) file will be kriged by execution of the

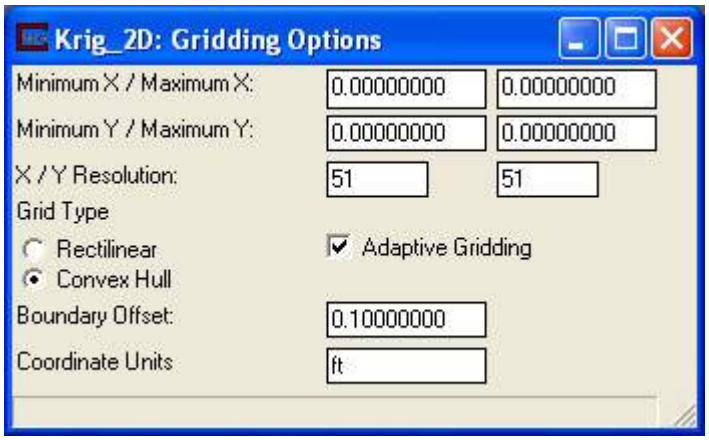
module. The default value is -1, which results in kriging of all of the data components in the file during a single execution of the module.

The "***Run***" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed (or the geology input changes). When it is off, the module will not run.

The "***New-Reset Variables***" toggle (on by default) resets all expert system calculated variables to zero before each run. This allows multiple calculations to be performed without tedious manual resetting of these variables. To change an expert system calculated variable, the toggle must be off.

Module Parameter Subpanels

Krig_2D has six subpanels which allow the user to set the parameters used for preprocessing the input data, producing the semivariogram, executing kriging, and post processing kriged output. Clicking on either the check boxes next to the subpanel names, or on the names themselves will bring up the subpanel parameter screens.



The ***Gridding Options*** subpanel is shown in the figure above. It is used to specify all parameters that affect the grid exported from Krig_2D. The availability of many options depends on whether geologic input is available or if the grid is created totally within Krig_2D. The window above shows the options with no geologic input. NOTE: If an external grid is imported into Krig_2D this panel will be blank.

The ***Minimum X, Maximum X, Minimum Y,andMaximum Y*** parameters allow the user to define the horizontal domain within the data set in which kriging of the parameter distribution will be completed. A value of 0 is the default for these parameters, which results in a model domain that is defined by the rectilinear bounds of the entire data set when the module is run. Krig_2D utilizes a model domain that is bounded by the limits of the data set, unless the user specifies a different domain by setting the Min and Max Values for X and Y. Utilizing the default extents effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points. If the user is uncertain of the X and Y limits of the data domain, the module should be run with the default 0 values, and upon completion of execution, the values in the X and Y input fields will be

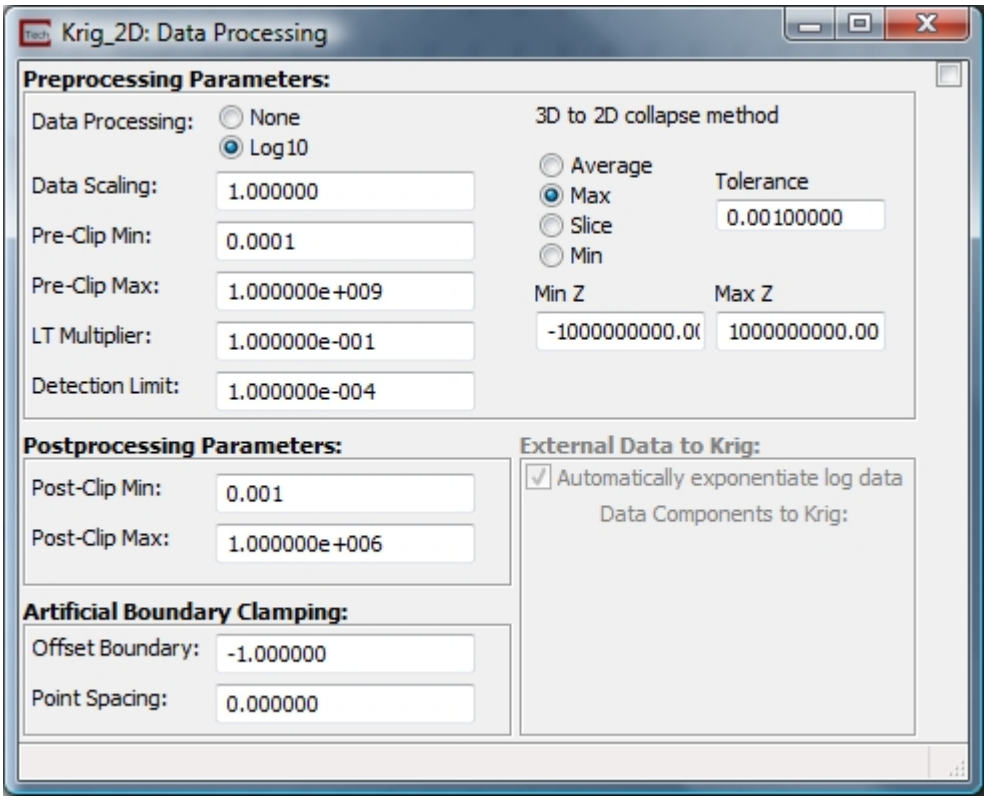
the min and max values of these parameters in the data set. The File Statistics module can also be used to investigate the limits and distribution of values in the input data set, and the statistics module can be used to output the distribution of values in the kriged model.

The ***X / YResolution*** parameters specify the number of grid nodes that will be included within the model domain. The number of grid elements along any axis of the model is simply the axis Resolution value minus one, as every element has two bounding nodes along an axis. The default value for the X and Y Res parameters is 51. However, the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user.

The **Grid Type** radio buttons (***Rectilinear*** / ***Convex Hull***) allows the user to select the type of domain in which the kriging will be completed. A convex hull boundary should be utilized when the user wishes to produce a model that can have irregular boundaries that are defined by the extent of the measured data points. The convex hull of a data set can be thought of as the domain that would be outlined by stretching a rubber band around the external data points in the data set. The convex adaptive grid (This feature available only in EVS PRO and MVS) is the same as the convex hull grid, except that the grid is automatically adapted to place grid nodes at all of the measured data points. The adaptive grid will result in a kriged parameter distribution that honors all of the measured data points exactly. This is the default domain used by Krig_2D. Utilizing a convex hull boundary effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points, or by the specified Boundary offset.

The **Boundary Offset** parameter sets the distance in user units that the grid coordinate range for the kriging domain will be set outside of the actual coordinate range of the data. This parameter allows the user to specify the distance outside of the actual data in which the parameter values will be extrapolated. A value for the Boundary Offset of 0 units, specifies that the true (not offset) convex hull of the data set will be used.

CoordinateUnits - This field contains the default coordinate units for the grid, if no coordinate units are specified in the file being read, these units will be used.



The ***Data Processing*** parameters subpanel is shown in the figure above. It is divided into four groups: Preprocessing; Postprocessing; Artificial Boundary Clamping; and External Data to Krig.

It is important to note that all preprocessing actions are applied directly to the data in memory, and that the original data file is not altered. However, all of the functions within Krig_2D and the modules downstream of Krig_2D will be using the preprocessed (and kriged) parameter distribution. The user should refer back to the preprocessing subpanel when setting values for the filtering and display of the kriged data (i.e., to correctly specify whether the data has been log transformed, scaled, and/or clipped).

The ***Data Processing*** radio buttons allow the user to specify whether the data will be used as is, or will be processed to compute the log (base 10) of the parameter value before kriging. Note that if the log10 of the data is taken, a Clip Min value (which must be greater than 0) must be used to replace values in the data below the specified minimum value because the log function is undefined for values equal or less than zero.

The ***Data Scaling*** input field is used to specify a value by which all nodal data values will be multiplied before kriging. The default value is 1, but the user can specify any negative or positive value. This option is most commonly used to convert the units of the property being kriged, such as to convert concentrations in ppb to ppm or visa-versa. The user should bear in mind that any scaling of the data that is completed in Krig_2D will affect all downstream modules.

The number entered into the ***Pre-Clip Min*** input field will be used during preprocessing to replace any nodal property value that is less than the

specified number. When log processing is being used, the value of Clip Min must be a positive, non-zero value. Generally, Clip Min should be set to a value that is one-half to one-tenth of the lowest detection limit in the data set, unless the user wishes to make the influence of not detected values stronger. As an example, if the lowest detection limit is 0.1 (which is present in the data set as a 0), and the user sets Clip Min to 0.0001, the clipped non-detected values forces three orders of magnitude to be present between any detected value and the non-detected values.

The number entered into the ***Pre-Clip Max*** input field will be used during preprocessing to replace any nodal property value that is greater than the specified number. The clipping values can be used to lessen the importance placed on extremes of the data, or outlier data values, before kriging. The preprocessing functions can be used in various ways to investigate the kriging confidence levels within specified ranges or limits of the data sets. An example of this type of analysis is provided in the Sample Networks section.

The ***LT Multiplier*** value affects any file value with a preceeding "<" character. It will multiply these values by the set value.

The ***Detection Limit*** value affects any file values set with the "ND" or other non-detect flags (for a list of these flags open the help for the CSV file format). When the module encounters this flag in the file it will insert the a value equal to (Detection Limit * LT Multiplier).

The **3D to 2D collapse method** allows the user to extract data from within the interval specified in **Min Z** and **Max Z** using four different methods. It should be noted that if multiple samples occur at the exact same x, y, and z coordinates (duplicate samples), the average of these samples will be calculated first to provide a single data point for that coordinate, and then the extract method will be applied to other samples in the specified depth interval. If the **Average** radio button is selected, then Krig_2D will search the vertical interval in each boring and calculate the average of all the values found for input to the kriging analysis. If the **Max** radio button is selected (the default selection), then Krig_2D will extract the maximum property value that exists within the specified Z interval. When the **Slice** radio button is chosen, Krig_2D will look for the closest data point above the specified Min Z value, and the closest data point below the specified Min Z value, and calculate the average of these two points for input to the kriging analysis. The Slice Extract Method is provided as a convenient method for the user to assemble a 2-D data set that will contain at least one data point from each sampling location. If only one data point exists for a sampling location, then that value will be extracted for that location whether it is actually located above or below the specified Slice Position. If the **Min** radio button is selected, then Krig_2D will extract the minimum property value that exists within the specified Z interval.

The ***Tolerance*** parameter further extends the definition of coincident points. It allows all points within a +/- x-y box to be treated as coincident. This is not the same a true euclidean distance, since two points which were different in both x and y by 0.9999 would be considered coincident if the tolerance

were 1, even though the true distance between the points was over 1.4 (square root of the sum of the squares).

Postprocessing of the data from Krig_3D affects the calculation of the kriging confidence and uncertainty estimates that are produced by the model, and allows the user to apply a filter for the data passed to all modules downstream of Krig_2D. Note that the postprocessing **does not** affect any of the other Semivariogram or kriging algorithms that execute in Krig_2D, so the user can experiment with different values of postprocessing parameters to obtain the type of display desired, independent of the internal kriging process.

The ***Post-Clip Min*** parameter specifies the smallest nodal value that will be present in the data field output by Krig_2D. This parameter is useful for limiting or enhancing the effects of not detected values or outliers in a data set, and for optimizing the use of the dynamic color range used to represent the property distribution. Clip Min has a default value of 0.001, but can be set to any negative or positive value with magnitude from -1.0 E09 to 1.0 E09. In general, good results are obtained by setting this value to the lowest property value or detection limit in the input data set. It is important to note that because not detected values are represented in the .apdv file as zeros, the Clip Min value can be used to strengthen the influence of small values or non-detects that are present in the data set. As an example, if the detection limit for a certain chemical analysis is 0.1 (which is entered into the .apdv file as 0), and the Clip Min is set to 0.0001, then this not detected value will have an effective influence on the kriged distribution near this data point that is three orders of magnitude stronger than the actual detection limit. The user should bear in mind that if mass or volume estimates are being made in the analysis, the Clip Min value could affect the estimates if considerable data are present that are being clipped. For volume estimates, if the specified lower bound of the concentrations of interest is well above the Clip Min value, then their will be no effects on the results. If the specified lower bound of concentrations for the volume estimate is lower than Clip Min, then the clipping will truncate the volume at the Clip Min value, and the estimated volume will be from the entire model domain.

The ***Post-Clip Max*** parameter specifies the largest nodal value that will be present in the data field output by Krig_2D. Again this parameter can be used to optimize the use of the dynamic color range when a data set has a few extremely high values, but the user is most interested in seeing the detailed changes in the distribution in some lower range of the data. It can also be used to investigate the kriging confidence or uncertainty levels near certain threshold values of a property. As an example, if the regulatory threshold for a parameter is 10 units, then the Clip Max parameter can be set to 10, and all kriged values greater than 10 will have the same strength in the calculation of the uncertainty levels in the kriged distribution. Clip Max has a default value of 1,000,000, but can be set to any negative or positive value with magnitude form -1.00E09 to 1.00E09.

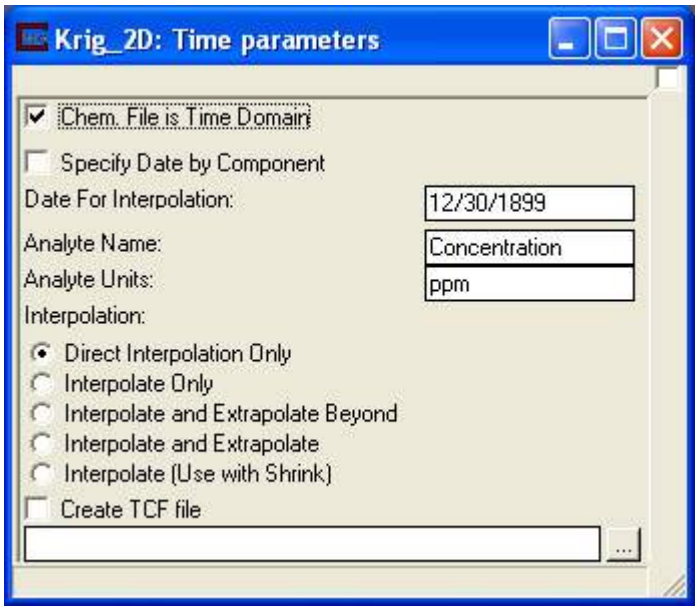
Artificial Boundary Clamping creates a ring of artificial points outside the grid coordinate range to control how extrapolation is done outside of the data set. The artificial points are not used in the production of the semivariogram.

The ***Offset Boundary*** parameter specifies the distance outside of the coordinate range of the data set that a boundary (or ring) of synthetic data points, which have property values that are equivalent to the specified Preprocessing Clip Min, will be automatically generated for kriging.

The ***Point Spacing*** parameter sets the linear distance along this clamp boundary at which the synthetic data points will be placed. The default value for Offset Boundary is -1.0, which specifies that no clamping boundary should be used. The user can set this value to any distance in user units that is within the kriging Reach distance specified, to modify how sharply the kriged data distribution will transition to the clamped value. The default Point Spacing parameter value is 0.0, which specifies that no points will be placed along the clamp boundary. Again, the user can input any value in user units that will provide the desired modeled data distribution.

The user should be aware that the synthetic data points **do not** affect the kriging semivariogram production, but they **do** affect the kriging confidence and uncertainty values that are output by Krig_2D, as they are considered by the kriging algorithms to be real data points. Therefore, a clamping boundary should not be used when confidence or uncertainty values are of importance to the analysis. The clamping boundary is provided primarily as a display control tool, and will produce a very even and regular synthetic boundary along the outside of the data domain.

When external data is being imported into the module the External Data to Krig window becomes active. This window allows the user to select which data components to krige by switching on or off the toggles associated with that data component. The ***Automatically exponentiate log data*** toggle will detect if the data being imported has been log processed and if it has exponentiate it. This avoids double log processing the data with the Preprocessing option.



(EVS PRO and MVS Only)

This panel is for analyte (e.g. chemistry) files (*.apdv, *.aidv) that are set up as time domain files (i.e. one analyte whose values are recorded over time).

The ***Chem. File is Time Domain*** toggle turns on date interpolation for time domain analyte (e.g. chemistry) files.

The ***Specify Dateby Component*** toggle will cause the module to ignore the set Date for Interpolation and instead use the component slider to select the date

The ***Datefor Interpolation*** field is the date being interpolated to, for example if you have an analyte value of 2 on 1/01/05 and a value of 4 on 1/03/05 and the date is set to 1/02/05 with Direct Interpolation the value should be set to 3. The Date can be either set by hand or imported. This makes the module useful in a time_loop.

In time domain files there is no place to set either the analyte name that has been kriged over time or the analyte units; these can be set in the ***Analyte Name*** and ***Analyte Units*** fields.

There are 5 different ***Interpolation*** methods that are available, each interpolation method is used to define how to interpolate when given unsampled times in a file.

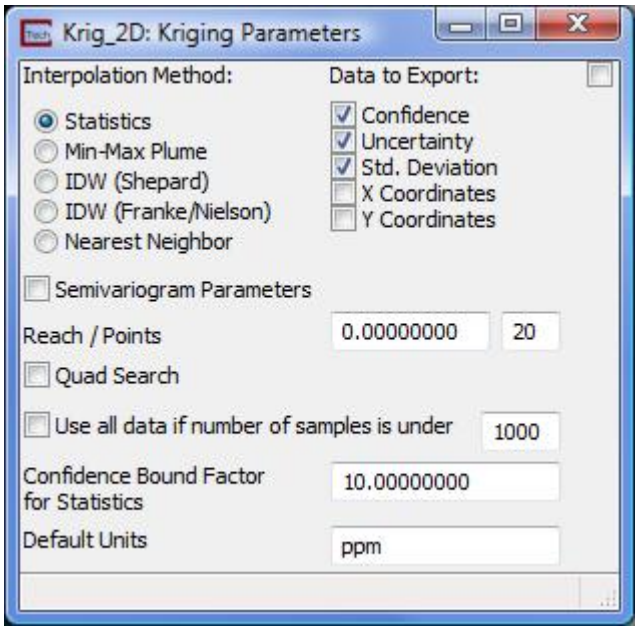
1. *Direct Interpolation Only:* This is the most basic interpolation method, and the most accurate in terms of representing the data as it has been entered. This method looks at the two dates surrounding the input Date. If either date is unsampled, the value for that sample will remain unsampled and no interpolation will occur. If both dates have sampled values it will interpolate between them.
2. *Interpolate Only:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it

finds a sampled date.. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. If either search fails to find a useable dates then it will set the value for that time to a unsampled value. Else it will interpolate between the two found dates. This is useful for files that have a small amount of unsampled values.

3. *Interpolate and Extrapolate Beyond:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. If it does not find a sampled date after the input Date, it will extrapolate beyond the last useable date to the input Date.
4. *Interpolate and Extrapolate:* This method will look at the two dates surrounding the input Date. If the date before the input Date is an unsampled date it will continue to look backwards through each time column until it finds a sampled date. If it fails to find one it will extrapolate the first value backwards to the input Date. If the date after the input Date is unsampled it will look forward through the time columns until it finds a sampled date. It will also extrapolate beyond the last valid date in the file.
5. *Interpolate(Use with Shrink):* This method uses the same interpolation algorithm as method one, but should be used for clarity when there is a post_samples or file_statistics module in the network that is using the Interpolate and Shrink to Unsampled option.

The ***Create TCF File*** toggle will cause the Krig_2D module to run in a loop, going through each component and creating an EFB file at that time. These EFB files will be linked together in the specified TCF file and can be used with the Read_TCF module for animation purposes. NOTE: This function will not work unless a TCF name is selected as well.

Kriging Parameters



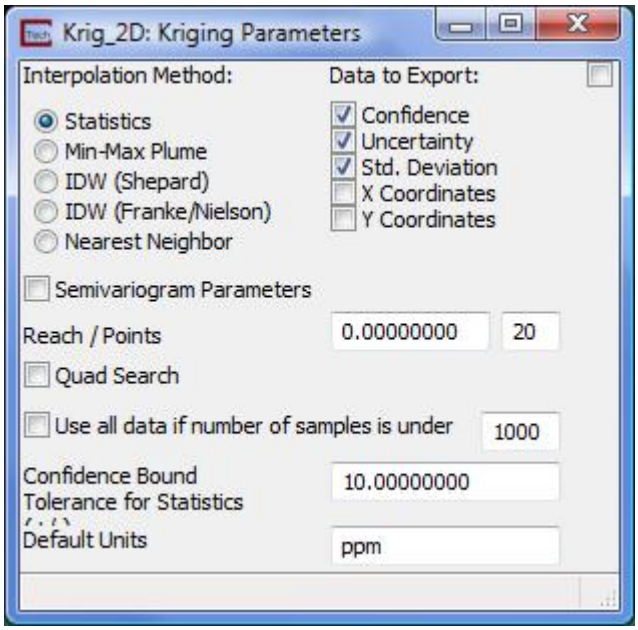
The ***Kriging Parameters*** subpanel of Krig_2D is shown in the figure above.

Note: The "new" toggle on the main panel (on by default) resets all expert system calculated variables to zero before each run. This allows multiple calculations to be performed without tedious manual resetting of these variables. To change an expert system calculated variable, the toggle must be off.

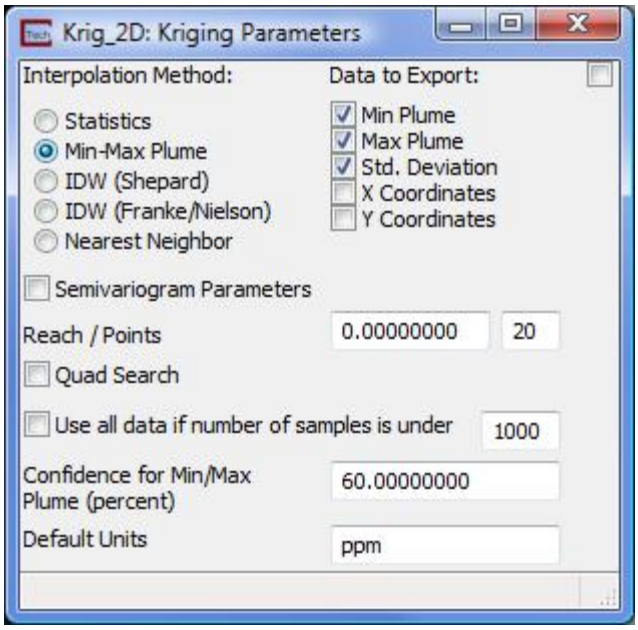
The ***Interpolation Method*** radio buttons determine the type of statistical information which will be included in the nodal data components output from Krig_2D or to select IDW vs. kriging.

If ***Statistics*** is chosen, Each Concentration (for every chemical in the .apdv file) will have a corresponding Confidence (based on the Confidence Bound parameter) and Uncertainty. The display when Statistics is selected and the data is log processed is shown above.

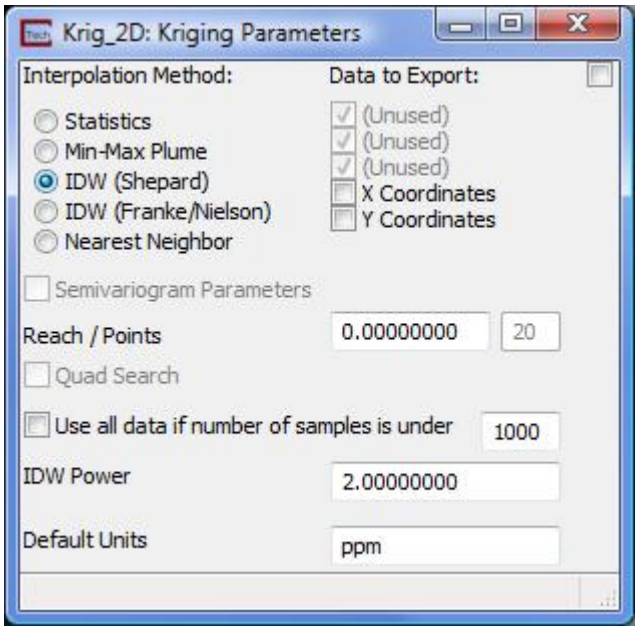
The display when Statistics is selected and the data is NOT log processed is shown below. In this case, the confidence bound is a tolerance vs. a factor.



If ***Min-Max Plume*** is chosen (This feature available only in EVS PRO and MVS) three different Concentration components will be calculated, the Nominal, Minimum and Maximum. These different distributions are determined based on the standard deviation and nominal concentration at each node based on the *Confidence* parameter which is expressed in percent (nominally 60%). Note that at a "confidence" of 50%, the nominal, minimum and maximum concentrations are identical.



If either ***Inverse Distance Weighted (IDW)***estimation method (Shepard and Franke/Nielson) is selected, kriging is not performed and one of these algorithms is used as an alternative to kriging in Krig_2D. This also affects the available parameters as show in the figure below:



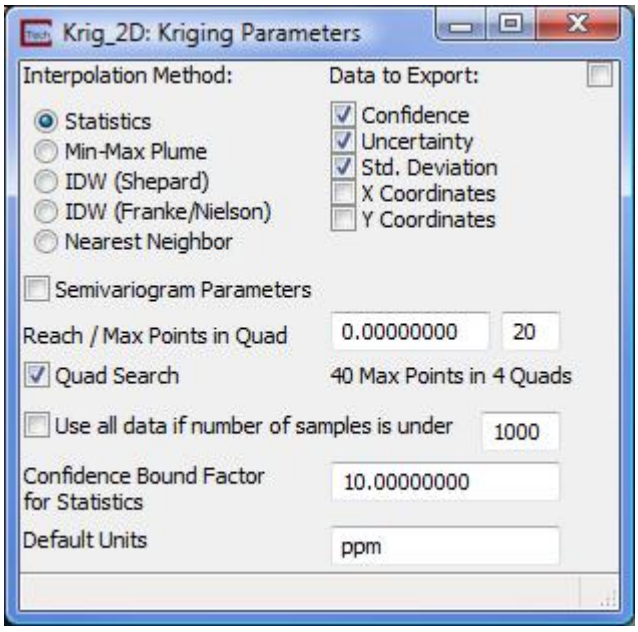
The ***Reach*** input field defines the radial distance (in user units) from any given model node that the kriging module will look for data points to be included in the estimation of the model parameter at that node. The default value of reach is 0, which results in the module calculating a reach value which is approximately two-thirds of the longest distance between any two data points in the data set.

The ***Points*** parameter defines the maximum number of data points (within the specified reach) that will be considered for the parameter estimation at a model node. The default value for points is 20, which generally provides reasonably smooth modeled parameter distributions. The effects of decreasing and increasing the values for reach and points on the model output are somewhat similar, but for different reasons. If the data have a fairly even spatial distribution throughout the domain, then increasing these values will generally include more of the input data points that will be used to krige the value for a given model node, and thus will result in smoother modeled data distributions. Decreasing the values of reach and points (in an evenly distributed data set) results in fewer input data points being used to calculate the parameter estimates at a given model node, and result in modeled distributions with greater variations across smaller areas.

The user should consider both the spatial distribution and the range of values in the input data set when deciding upon values for the reach and points parameters. If the specified reach is too small to allow the kriging module to locate at least one point within the search area, then no kriging can be performed and the module will terminate with an error message to the Status Console.

If the user specifies a large number of points (that are within the specified reach), then the output will be smoother, but the execution time for the kriging can increase significantly. By posting the input data using the *post_samples* module, and looking at the characteristics of the resulting kriged data using the *statistics* module, the user can quickly analyze the

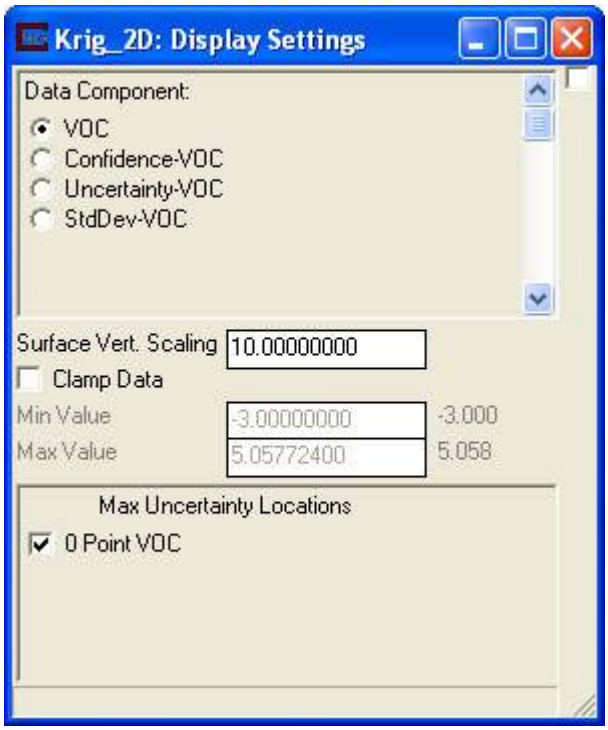
characteristics and distribution of the kriging output for a given set of parameters, and test the effects of changing the kriging parameter values.



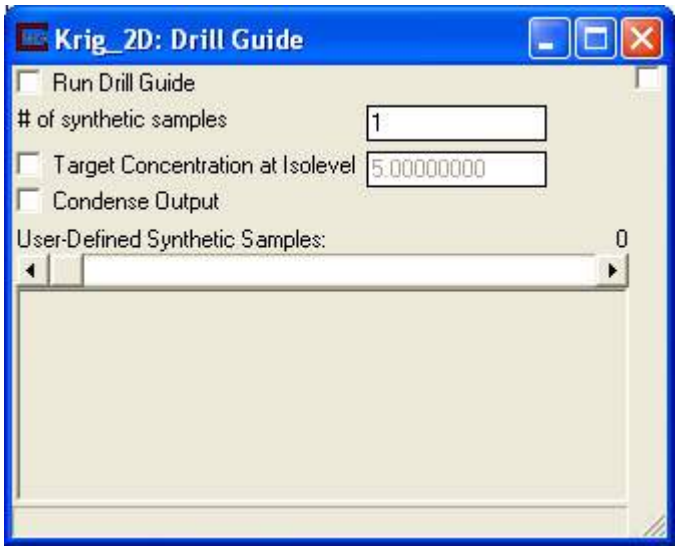
The ***Quad Search*** toggle changes the method by which data sample points are selected for inclusion in the kriging matrix. If this is on, the "Points" parameter switches to "***Max Points in Quad***". Searching is performed for each of the quadrilaterals surrounding the point to be kriged. Within each quadrilateral a maximum number of points (up to one-fourth of the total points) are selected. Then, points are taken sequentially from each quadrilateral up to the maximum number of total points or until all octant's points have been used. The panel display changes when this option is selected as shown above. Quad Search is applicable for the Statistics and Min-Max Plume kriging modes.

The type-in value for ***Use all data if number of samples is under*** is off by default, but this option gives the smoothest surfaces since all data is used for the kriging process. Sometimes using all points results in faster computation since only one (large) kriging matrix must be solved.

The ***Confidence Bound*** parameter is used to specify what interval around the kriged model estimates the kriging confidence or uncertainty will apply to. The default value is 10, which essentially produces the confidence and uncertainty that the kriged data are within one order of magnitude of the "true" value. As an example, if the Confidence Bound is 10, the kriged property value at a node is 5, and the kriged confidence level at the node is 0.9, then 90% of the time, the "true" value of the kriged property at that node will be in the range of from 0.5 to 50 units. Additional discussions of confidence and uncertainty are provided in the Sample Networks.



The Display Settings window is shown in the image above. This window controls the display of the uncertainty sphere location. By default there is one uncertainty sphere for every analyte. The visibility of the spheres can be toggled on and off by click on the check box next to the name of the analyte. These spheres can also be vertically exaggerated to match any downstream scaling of the field itself.



The DrillGuide© panel is shown above. When the ***Run Drill Guide*** toggle is selected Krig_2D will run in a loop creating a synthetic boring at the maximum uncertainty location until it has reached the set ***# of synthetic samples***.

The ***Boring Samples*** field indicates how many samples create at each synthetic boring that is created.

The ***Target Concentration at subsetting level*** toggle, when selected, changes the location of the uncertainty sphere based upon the specified target concentration.

The ***Condense Output*** toggle, when selected, will minimize the number of strings displayed in the EVS console window.

The ***User-Defined Synthetic Samples*** slider is used to add points to the grid for drill guide purposes. The purpose of this slider is to help eliminate areas of high uncertainty that cannot be drilled, for example under a building.

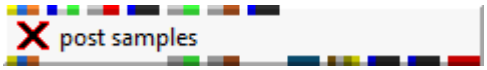
Krig_2D Module Hints

Manually adjusting kriging and semivariogram parameters

The user must ***un-***check the new toggle in the main window in order to make manual adjustments to most parameters in this module. Otherwise the values will be calculated by EVS according to the spatial extents or distribution of the data.

When using the set semivariogram option, all other semivariogram parameters are ignored. But to go back to the expert system parameters the user must make all type-ins 0,00 (in set semivariogram), then check the new toggle on the main Krig_2D panel.

post_samples



(This module has some features available only in EVS PRO and MVS)

General Module Function

The post_samples module is used to visualize:

1. Sampling locations and the values of the properties in .apdv files
2. The lithology specified in a .geo file
3. The location and values of well screens in a .aidv file

along with a representation of the borings from which the samples/data were collected. The post_samples module has the capability to process property values to make the posted data values consistent with data used in kriging modules. Data can be represented as spheres or any user specified glyph. The sampling locations may be colored and sized according to the magnitude of the property value, and labels can be applied to the sampling locations with several different options.

Each sampling location can probed for data by holding the alt button and left-clicking on the sample location.

The post_samples module can also represent downhole geophysical logs or Cone Penetration Test (CPT) logs with colored and sized tube diameters according to the magnitude of the data. It can display nonvertical borings and data values collected along their length, and can also explode borings

and sample locations to show their correct position within exploded geologic layering.

When used to read geology files, post_samples will place surface indicators at the top (ground) surface and the bottom of each geologic layer that are colored according to the layer they depict. When a geology file (.geo or .gmf) is exploded without using geologic surface input from Krig_3D_Geology (or Spline_Geology) there will be surface indicators at the top and bottom of each layer. You may color the borings by lithology.

Module Input Ports

(For EVS PRO and MVS)

The post_samples module has seven input ports.

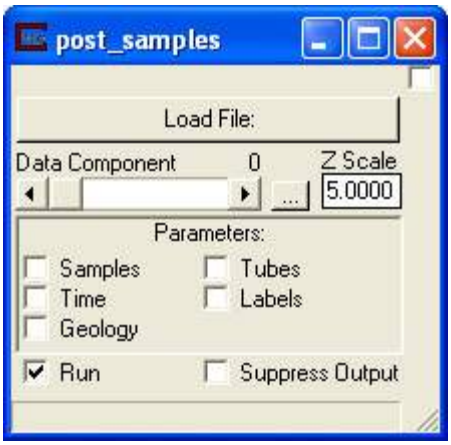
1. The first (leftmost) Yellow-Blue-Orange port allows the sharing of file names between similar modules.
2. The second port can only be connected to Krig 3D Geology, which provides the geologic layering information that allows the boreholes and spheres to be exploded into their appropriate geologic layers (NOTE: If a *.geo file is being read, the samples can be exploded using the values in the file itself with no additional input.)
3. The third port is a float value representing a date, this value can be imported from a time_loop module and is used for interpolating between dates in a *.apdv (or *.aidv) time domain file format.
4. The fourth port (Blue-Black) allows a 2D surface or (1D) line to be passed into the module:
    1. The surface is used to subset the data read in from the file.
    2. The line may consist of several segments and is used to project the borings for 2D displays
5. The fifth port is the Z-exaggeration factor
6. The sixth port is the explode distance, which can both be imported from the explode_and_scale module.
7. The seventh Blue-Black port is an optional port used to input a glyph to be placed at each sample location (by default spheres are placed at each sample location).

Module Output Ports

The post_samples module has eight output ports.

1. The first (leftmost) Yellow-Blue-Orange port allows the sharing of file names between similar modules.
2. The second port is the Explode factor.
3. The third port is the Z-exaggeration factor.
4. The fourth port is a string (text) representing the analyte being displayed

5. The fifth port exports geologic layer names when reading .geo, .gmf or .pgf files. This can be used by the Legend module for layer naming.
6. The sixth port (Blue-Black) exports a field containing the borings tubes. This can be exported as a shapfile or EFB (with EnterVol) for use in ArcGIS..
7. The seventh port (Blue-Black) exports a field containing the data associated with each boring. The output contents depends on what type of objects are visible:
    1. If Colored Tubes are present, only they will be output....otherwise
    2. If Colored Spheres are present, only they will be output....otherwise
    3. If Well Screen (Interval Data) are present, they will be output.
8. The last port (Red) exports a renderable geometry that can be connected the Viewer for visualization.



Module Control Panel

The control panel of post_samples is shown in the figure above.

The **Load File** button opens a *File Browser* which lists the *.apdv, *.aidv, *.geo, *.pgf or *.gmf files that are present in the current directory shown in the directory window.

To display nonvertical boreholes, the only requirement is that samples from any given borehole have the same location name, with the actual coordinates of each sampling location. The **Data Component** slider allows the user to select which of the property values in a file will be displayed by post_samples. For example, when reading CPT data the user may choose either tip resistance, sleeve friction or friction ratio using the slider. When reading contaminant data the user may choose among the various contaminants listed in the file. The default value is 0, which selects the first value that is present in the file. When reading .geo files, the data for all of the geologic layer tops will be read and displayed.

The **Z Scale** field displays the current Z exaggeration factor.

The **Run** toggle is used to turn on and off automatic execution of the module whenever a parameter is changed or a connection is made.

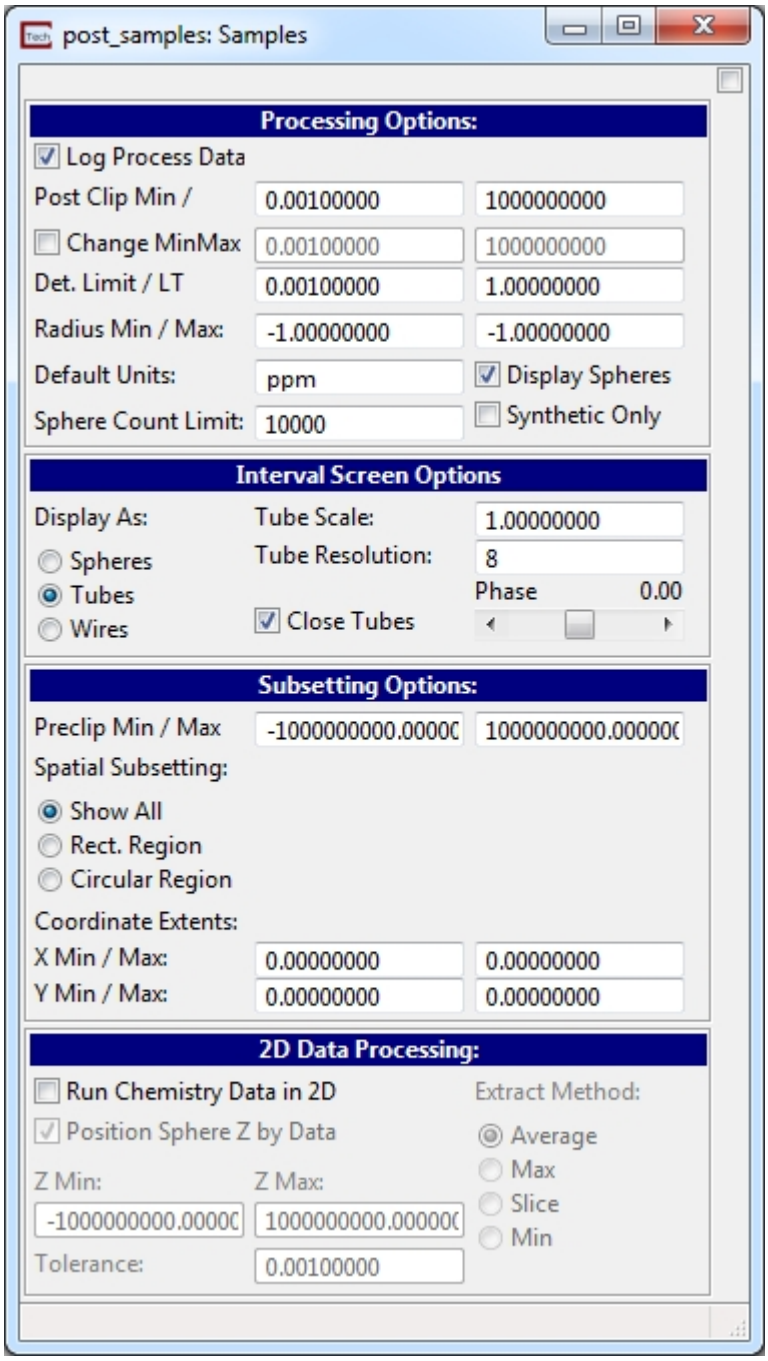
The **Suppress Output** toggle is used to prevent messages from the post_samples module from appearing in the Command Console.

Module Parameter Subpanels:

The post_samples module has five subpanels. These allow the user to:

1. set the parameters used while processing the input data
2. to specify the characteristics of the tube used to represent geology
3. to control how time data is displayed
4. to control the display of tubes, lines, colored tubes
5. to specify the types of labeling that will be displayed.

Clicking on the check boxes next to the subpanel names will bring up the subpanel parameter screens.



The **Samples** subpanel is shown in the figure above. It is important to note that all data processing actions are applied directly to the data in memory, and that the original data file is not altered. The data processing within post_samples applies only to the Sample indicators or tube coloring in the display. When setting data processing parameters in post_samples, the user should refer back to the preprocessing and/or postprocessing parameters set in any other modules that will contribute to the final display to assure that consistent parameters are being used (i.e., to correctly specify whether the data has been log transformed, scaled, and/or clipped). This is particularly

important when a color scale is being used to display parameter value distributions on a 2D or 3D subset and spheres in the same display.

***Processing**:*

- The number entered into the ***Post Clip Min*** input field will be used after data processing to replace any sample property value that is less than the specified number.
  - The default value for Clip Min is 0.001 when the ***Log Process Data*** toggle is on (default), but reverts to -1.0+e9 when it is off. The number entered into the
- ***Post Clip Max*** input field will be used after processing to replace any nodal property value that is greater than the specified number.
  - The default value for Clip Max is 1.0+e9 (so it won't clip anything), but the user can enter any number less than your data maximum.

Notes:

1. The above CLIP values do change your data and how it will be displayed and labeled.
2. Setting the clip max outside your data maximum will not shift the data mapping outside your data range. If duplicate data values exist in files (data points with exact X, Y, and Z coordinates), these values are averaged to produce one data point for each unique coordinate.
3. If the log10 of the data is taken, the Clip Min value must be used to replace any zero values in the data with a specified Clip Min value that is greater than zero, to eliminate possible errors associated with attempting to take the log of 0 or negative number (which is undefined). The ***Log Process Data*** toggle is used to indicate that the log10 of the data read from the file is to be used.

- The ***ChangeMin/Max*** toggle (and associated min and max type-ins) behave like the change_minmax module and sets the min and max values that are used by the datamap. This affects colors, but does not affect the actual data.
- The ***Radius Min / Max*** input fields are used to specify the minimum and maximum radii of the spheres (or other sample indicators in user units), that will be used to scale the sample indicators to the value of the property being represented. If these values are set to zero the scaling procedure considers the extent in X-Y of the domain of the file being read by post_samples, to produce sample indicators that are easily seen on the display. When posting geophysical logs or CPT logs (with apdv files), the Radius Min / Max are commonly adjusted to different values to display the relative magnitude of the deflections along the borehole trace.

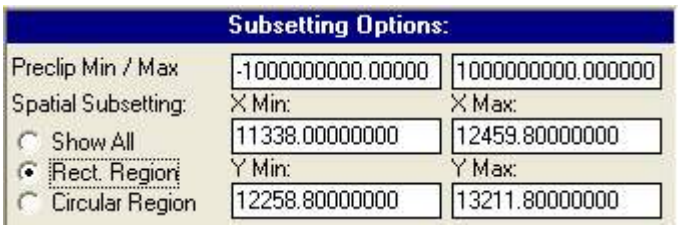
- The ***Default Units*** field indicates in which units the data are to be considered if not specified in file.
- The ***Sphere Count Limit*** field determines the maximum number of spheres to create. If the number of samples (spheres to be generated) exceeds this value, the module will turn spheres off and generate (more efficient) colored tubes instead.

Interval Screen Options:

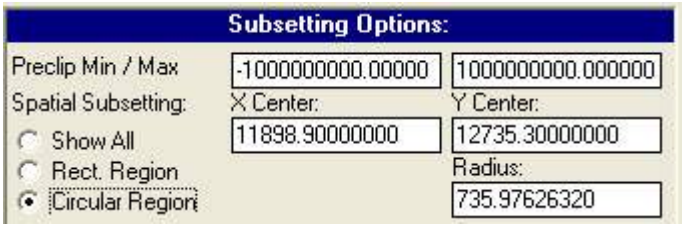
- The ***Display As*** radio box controls how the well screens will be displayed. If the *Spheres* option is selected, spheres are evenly distributed through the well screen based off the *Max Gap* parameter in the AIDV data file. If Tubes is selected the well screen will be displayed as a tube from the top of the well screen to the bottom. If Wires is selected the well screen will be displayed as a line from the top of the well screen to the bottom.
- The ***Tube Scale*** field allows the user to input a value to scale the radius of the well screen tube by.
- The ***Tube Resolution*** field changes the number of divisions used to make each well screen tube. The higher the resolution the smother or more circular each tube will be.
- The ***Close Tubes*** toggle if turned on will place a disc, of the same resolution as the tube, inside the top and bottom of the tube. This gives the well screen tubes a more solid visualization.
- The ***Phase*** slider will rotate the tube about its axis. This is more useful when using tubes of low resolution (e.g. flat-2 or triangular 3)

Subsetting Options:

- The ***PreClip Min*** and ***PreClip Max*** fields allow you to view a subset of your data samples based on the numbers entered. The default value for Clip Min is -1.00e9, and for ***PreClip Max*** it is 1.00e9.
- The purpose of ***Spatial Subsetting*** is to eliminate samples from being processed. This can be accomplished in two ways. The first is by connecting a surface to the subset surface port of the module (in EVS PRO and MVS), this can eliminate all samples within, or outside of the boundaries of the surface. The second is to define the type of region and extents of the region you wish to include in the processing. This eliminates all samples that fall out of the defined area.

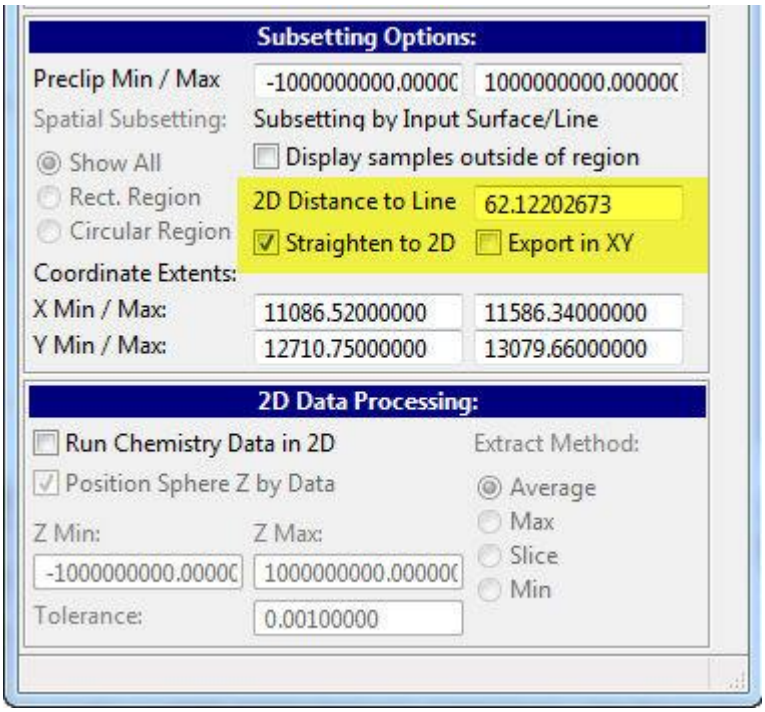


To subset by a rectangular region the XY extents of the region must be defined.



To subset by a circular region the XY center of the circle must be defined as well as the radius.

- The **Cut inside surface** toggle is used when a surface is input to the 4th input port. This toggle, when OFF gives you all borings inside of the surface. When ON gives you all samples outside of the surface.



The **Straighten to 2D** toggle shown above is used to project your 3D borings to a 2D representation. It creates a subset of your data which falls within the **Distance to Line** distance from a line segment input to the 4th input port. This functionality is typically used in conjunction with the related

2D straightening in the thin_fence module. You would use the same line input as you're using on thin_fence in that case.

The **Export in XY** toggle projects the data onto the XY plane vs. the XZ plane. This is the more useful option if the results are to be used in ESRI's ArcMAP.

> Please note that the coordinate extents of straightened borings will be to the right of the origin (0,0). It is best to create 2D projections in a separate Viewer from any other objects in your application.

The ***Coordinate Extents*** are read only and are there for reference when defining your subset area.

2D Data Processing:

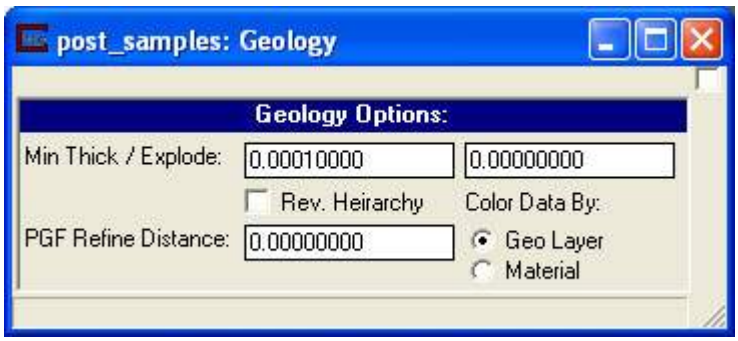
Checking the ***Run analyte (e.g. chemistry) Data in 2D*** toggle will cause any 3D data you have to be turned into 2D data based upon the ***Extract Method***.

- The 3D->2D subsetting will use the same rules and methods used by Krig_2D, so that the results from post_samples will show the actual values which Krig_2D would use internally in the kriging process.
- Any values in your data entered as "<x.xx" will be multiplied by the current LT (multiplier) and the resultant value will be used for all subsetting operations.

There are four different types of extraction methods.

1. The first, **Average**, will take the average data values for all samples sharing an (X,Y) coordinate, it will then place one sample with that average at the (X,Y) location.
2. The second method,**Max**, will take the maximum value for all samples sharing an (X,Y) location and then place a sample with that value at that (X,Y) location.
3. The third method, Slice allows the user to select a z coordinate for the slice position as shown below. It will then find the closest samples to that z location that share (X,Y) coordinates. If there is a sample at that exact location, it will use the data from that sample to create a point at that (X,Y) location. If it is between two samples it will interpolate the data between them based on their distance from the slice position. It will use that interpolated data to create a new point.
4. The final method, **Min**, will take the minimum value for all samples sharing an (X,Y) location and then place a sample with that value at that (X,Y) location.

The ***Position Sphere Z by Data*** toggle will set the Z component of the sphere to the value of the first data component.



GeologyPanel

The Geology is useful when reading in a geology based file (*.geo, *.pgf, *.gmf).

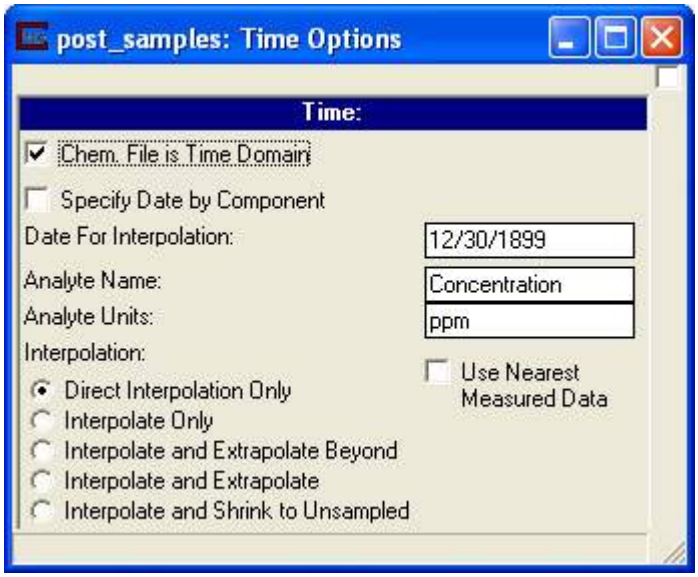
The ***Min Thickness*** is the smallest thickness, in user coordinate units, that a geological layer can have.

The ***Explode***(EVS PRO and MVS Only) distance can be set here or referenced by connection from the Explode_and_Scale module. NOTE: For *.geo files, the samples will be exploded based upon their layer designations unless a different geology is imported into the module.

The ***Rev. Hierarchy*** toggle will reverse the hierarchy of the geology. Normal hierarchy will push down any layer that tries to rise through the one above it. Reverse hierarchy will pull up any layer that dips into the one below it.

The ***PGF Refine Distance*** is used when pgf files are displayed as spheres. At minimum there is a sphere at the top and bottom of each layer, the refine distance will add spheres between those two spheres until no sphere is more than the refine distance apart.

The ***Color Data By*** radio box will color the sample spheres by either the Geological layer or the Material value of that layer.



Time Panel

(EVS PRO and MVS Only)

This panel is for analyte (e.g. chemistry) files (*.apdv, *.aidv) that are set up as time domain files (i.e. one analyte whose values are recorded over time).

The ***Chem. File is Time Domain*** toggle turns on date interpolation for time domain analyte (e.g. chemistry) files.

The ***Specify Date by Component*** toggle will cause the Date for Interpolation field to be ignored and the date to be selected using the component slider.

The ***Datefor Interpolation*** field is the date being interpolated to, for example if you have an analyte value of 2 on 1/01/05 and a value of 4 on 1/03/05 and the date is set to 1/02/05 with Direct Interpolation the value should be set to 3. The Date can be either set by hand or imported. This makes the module useful in a time_loop.

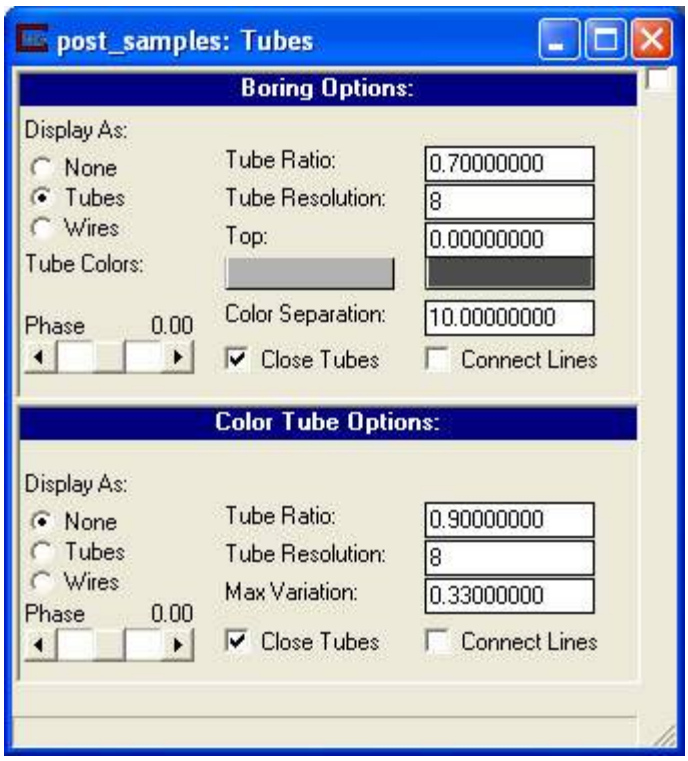
In time domain files there is no place to set either the analyte name that has been kriged over time or the analyte units; these can be set in the ***Analyte Name*** and ***Analyte Units*** fields.

The ***Use Nearest Measured Data*** toggle causes the sample at the interpolated date, to have the same value as the nearest measured date

There are 5 different ***Interpolation*** methods that are available, each interpolation method is used to define how to interpolate when given non-detects in a file.

1. *Direct Interpolation Only:* This is the most basic interpolation method, and the most accurate in terms of representing the data as it has been entered. This method looks at the two dates surrounding the input Date. If either date, or both dates, have non-detects as values, the value for that sample will be a non-detect and no interpolation will occur. If neither date has a non-detect value it will interpolate between them.
2. *Interpolate Only:* This method will look at the two dates surrounding the input Date. If the date before the input Date is a non-detect it will continue to look backwards through each time column until it finds a date with a value that is not a non-detect. If the date after the input Date is a non-detect I will look forward through the time columns until it finds a date that is not a non-detect. If either search fails to find a useable dates then it will set the value for that time to a non-detect value. Else it will interpolate between the two found dates. This is useful for files that have a small amount of non-detects.
3. *Interpolate and Extrapolate Beyond:* This method will look at the two dates surrounding the input Date. If the date before the input Date is a non-detect it will continue to look backwards through each time column until it finds a date with a value that is not a non-detect. If the date after the input Date is a non-detect I will look forward through the time columns until it finds a date that is not a non-detect. If it does not find a date with a non-detect after the input Date, it will extrapolate beyond the last useable date to the input Date.

4. *Interpolate and Extrapolate:* This method will look at the two dates surrounding the input Date. If the date before the input Date is a non-detect it will continue to look backwards through each time column until it finds a date with a value that is not a non-detect. If it fails to find one it will extrapolate the first value backwards to the input Date. If the date after the input Date is a non-detect I will look forward through the time columns until it finds a date that is not a non-detect. It will also extrapolate beyond the last valid date in the file.
5. *Interpolate and Shrink to Non-Detect:* This method will look at the two dates surrounding the input Date. It will shrink the size of the sphere when interpolating to a non-detect from a valid value, and will grow the size of the sample sphere when interpolating to a valid value from a non-detect. If the ***Non-Linear Shrink toggle*** is selected the size of the sphere will be determined by a cosine interpolation instead of a linear one. In the case of interpolating to a non-detect from a valid value, this causes the size of the sphere to shrink very slowly at first and then to shrink more quickly as the date approaches the non-detect date.



Tubes panel

The Tubes panel, shown above, handles the options for boring tubes and for creating color tubes. Both of these tube types are based on the boring ID's in a file. However, the Boring tubes will rise all the way to either the value in the Top field or the Top value specified in the file, while the Color Tubes will start at the highest sample location for each boring.

Boring Options

The ***Display As*** radio box change how the Boring tube is displayed, you can either turn off the tube entirely, display it as a tube, or display it as a wire.

The ***Tube Ratio*** field is a double value between 0.0 and 1.0 that sets what percentage of the sample sphere minimum radius the tube radius will be. For example if you have sample spheres with a minimum radius of 10 and the Tube ratio is set to .7, then the radius of each tube will be 7.

The ***Tube Resolution*** indicates how many faces will be used to create each tube; increasing this number will result in rounder tubes, but will use more memory. The minimum value for this field is 2 which creates plane like tubes, 3 will create triangular shaped tubes, 4 will be square, etc...

The ***Top*** value will set top elevation for each tube. If this value has been specified in the file, the file value will be used instead of this value.

Tubes are broken up into cells of two colors to help give a visual clue as to scale. These colors can be set by using the ***Tube Colors*** buttons. The height of each cell can be set using ***the Color Separation*** field. **NOTE:** Having a small Color Separation on a large model can impact performance because of memory demands.

The ***Phase*** slider will rotate a tube about its axis.

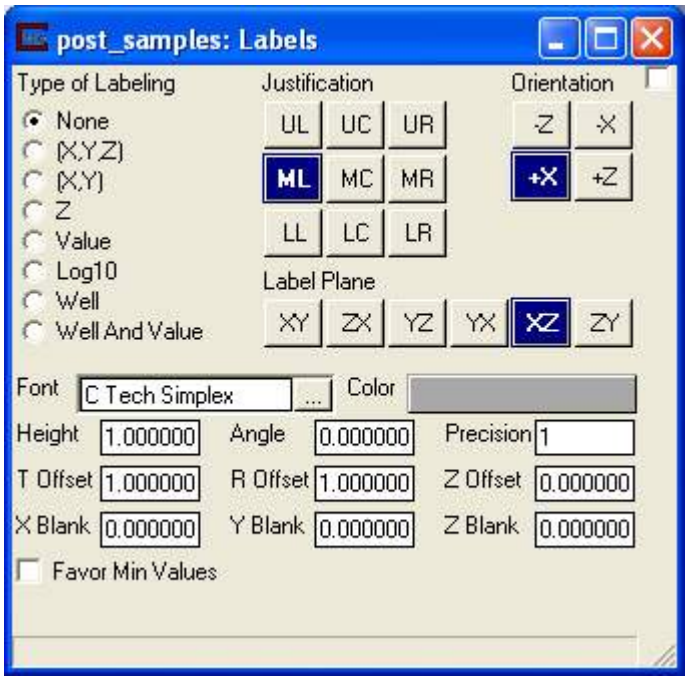
To put a cap on the top and bottom of each Boring Tube check the ***Close Tubes*** toggle.

The ***Connect Lines*** toggle is used for non-vertical borings. The best way to visualize this function is to use an example of 3 non-vertical samples. A cell will be created between the first and second sample and another cell will be created between the second and third sample. Because these are necessarily separate cells the ends between the cells may not be coincident. This is especially evident when looking at borings with large changes in angle between samples, or kinks. To merge these cells into a single tube the separate ends are "averaged" causing a distorted, but more efficient and seamless tube.

Color Tube Options:

The functions of the Color Tubes match those functions of Boring Tubes with the same names. Where Boring Tubes are separated by two distinct bands of color for a visual scale clue, Color Tubes are colored by the nodal data component.

The ***Max Variation*** value is a tool used to correct color blending between samples. Color tubes is actually interpolating the color of the tube between two samples based upon the color of the spheres, not upon their data. For example, if one sphere has a value at your data max and is colored red and the next sphere in the boring has a value at your data min and is blue, it would color the tube at the midpoint between these spheres purple, a color which does not exist in the default datamap. By forcing new points to be generated between these, we can prevent these discolorations from occurring. A value of .33, the default, states that any samples which lie next to each other along the tube and whose data varies by more than 33% of the total data range will automatically get extra points inserted between them.



Labels panel

The ***Type of Labeling*** radio box will allow the user to select what to label for each sample

1. None - Do not label samples.
2. (X, Y, Z) - Label each sample with its x, y and z coordinates.
3. (X, Y) - Label each sample with its x and y coordinates.
4. - Label each sample with its z coordinates.
5. (Value) - Label each sample with its analyte value.
6. (Log10) - Label each sample with the log base 10 of its analyte value.
7. (Well) - Label the top of each well or boring with the well or boring id.
8. (Well And Value) - Label the top of each well or boring with the well or boring id, and label each sample with its value.

The ***Justification*** buttons define how the labels will be justified around the samples.

The ***Label Plane*** options determine in which plane to write the labels.

The ***Orientation*** buttons determine in which direction in the label plane to write the labels.

The ***Font*** for the labels is by default C Tech Simplex, which is a low memory type font. For a font that will always be facing the view regardless of rotation, switch to Forward Facing font.

The ***Color*** button will change the color of the labels.

The ***Height*** field will change the height of the labels.

The ***Angle*** field determines where in a circle around the sample to place the label. For example a value of zero means place the labels to the direct right of the sample, a value of 90 would place the label directly above the sample.

The ***Precision*** field indicates to what precision to display any float or double values being displayed as labels.

The ***T Offset*** field is the title (the well or boring ID label) offset in the z direction. This can raise or lower your title and is added to the Z offset to determine the z location of the title.

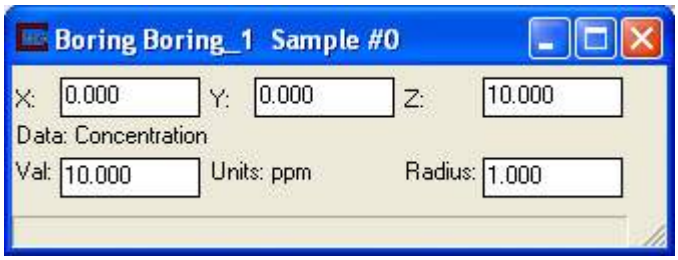
The ***R Offset*** field is the radial offset, this determines how far from the sample to place the label, a value of 1 means to add the radius of the sphere to the XYZ location of the sample and place the label there.

The ***Z Offset*** is added to the Z coordinate in labeled sample to raise or lower the labels in the Z direction.

The **X Blank**, ***Y Blank***, and ***Z Blank*** fields create a 3D box that will allow only the label at the highest value to be displayed inside. If there are multiple samples with the same max value within the box, the label that is displayed is based upon the order processed, which is not necessarily the order defined in the file. The ***Favor Min Values*** toggle will switch this behavior to display the label at the minimum value.

Interactive Querying of Your Data

The post_samples module allows you to probe any sphere and determine information about that sample and its borehole ID. To probe a sample, first verify that you have spheres **on**, click on any sphere with Left Mouse Button while holding down the Alt key. The following window will appear in the upper right hand corner of your desktop.



X (Easting)

Y (Northing)

Z (Elevation)

Data: (Current sample's nodal data name)

Val: (Sample Value)

Units: (Sample Units)

Radius: (Sphere Radius)

Spline_Geology

Spline_Geologyis an alternative to Krig_3D_Geology that uses the*thin plate spline*method for estimating surfaces from scattered data points.We have found that this method creates wonderfully smooth surfaces that honor your data.It is reasonably fast and is often preferable to kriging with small datasets (less than 50 points) but tends to**slow down rapidly with large datasets**.The nature of the solver makes it impossible to predict the time to completion. If running with datasets with more than 50 samples, be prepared for long run times with no progress feedback.

Module Input Ports

Spline_Geology has an Orange-Blue-Orange input port that can receive the geology file name.

Module Output Ports

Spline_Geology has six output ports.

1) Read_geo_data_file (Orange-Blue-Orange) : Supplies the geology file name.

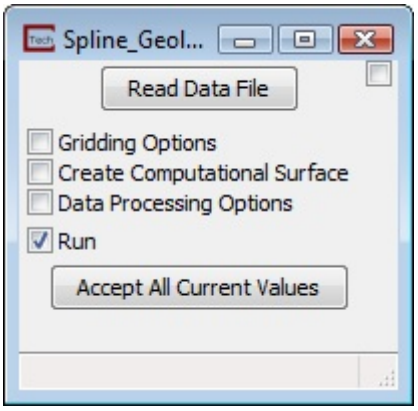
2) Vistas Group (blue-magenta-blue-magenta) : Provides input to the Geology_to_Vistas module.

3) (brown-grey-green/brown-yellow/brown) : Provides geologic material information for the Legend module.

4) status_out(blue/green) : Outputs a string containing status updates while the module is running.

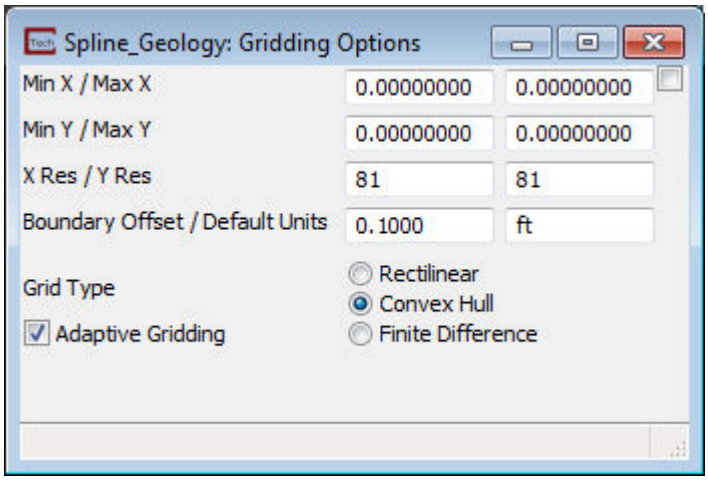
5) output0 (blue-white-green) : The primary output field. The primary output field port can be connected to the Krig_3D, 3D_Geology Map, and Geologic Surface modules.

6) Grid_out (red) : Outputs a renderable geometry of a finite difference grid.



General Module Function

Spline_Geology has fewer options than Krig_3D_Geology. It always uses all of your sample locations and therefore does not have parameters to adjust it's interpolation process.



The **Gridding Options** subpanel of Spline_Geology is shown above.

Although it is at the bottom of the subpanel, the first option that should be set by the user is ***Grid Type***, as this will determine the nature of the domain that will be kriged. The ***Rectilinear*** option is used when the user wishes to produce a model that contains estimated values everywhere inside a user specified rectangular domain. The ***Convex Hull*** Boundary Option is used when the user wishes to produce a model that can have an irregular boundary that is defined by the distribution of measured data points. The Convex Hull of a data set can be thought of as the domain that would be outlined by stretching a rubber band around the external data points in the data set. The Convex Hull boundary option effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points. The ***Finite Difference*** domain allows the user to krige the elevations of the geologic layers directly to the finite difference grid node locations that were setup in the control panel of Krig_3D_Geology. Note that the finite difference grid must be designed and displayed before the user selects a .geo file and clicks the **Accept all Current Values** button.

Adaptive Gridding causes the grid nodes to be shifted up to 30% of a cell width to cause the nodes to align with your input data values. By having grid node coincident with surface points, the grid will exactly honor the surface at that location. This help maintain accuracy with coarser grids.

The **Min X, Max X, Min Yand *Max Y*** inputs allow the user to define the horizontal domain within the data set in which kriging of the geologic surfaces will be completed. The Min and Max values are only used when the Rectilinear Option is selected. The Min (x-y) values are also used to set the origin for the Finite Difference Grid Type option. A value of 0 is the default for these parameters, which results in a model domain that is defined by the entire data set when the module is run. When kriging within a finite difference domain, the Min (x-y) values are used to set the origin (lower left hand corner before grid rotation). If the user is uncertain of the X and Y limits of the data domain, the module should be run with the default 0 values, and upon completion of execution, the values in the X and Y input fields will be the min and max values of X and Y in the data set. Alternatively, the File_Statistics module can be used to examine the data

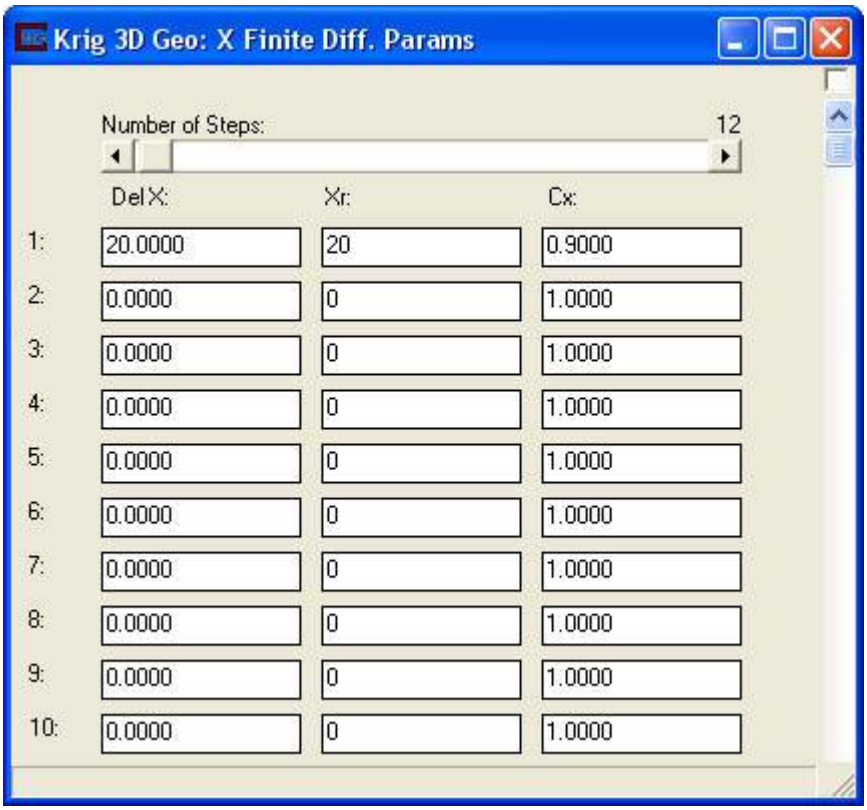
characteristics. There is no specification of the Z grid dimensions, as Krig_3D_Geology outputs 2-D surfaces of each geologic interface. The modules that use the output of Krig_3D_Geology define the Z grid dimensions as specified by the user in those modules.

The ***X Res and Y Res*** parameters specify the number of grid nodes that will be included within the model domain. The number of grid elements along either axis of the model is simply the X Res or Y Res value minus one, as every element has two bounding nodes. The default value for these parameters is 41, but the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user. The robust kriging algorithms in EVS generally produce reasonable modeled distributions with a fewer number of grid nodes than the user may be used to, so the recommended procedure for setting the X and Y Res parameters is to start with less, and then increase the value until an acceptable model fidelity is obtained.

The ***Boundary Offset*** parameter sets the distance that the convex hull for the kriging domain will be set outside of the actual convex hull of the data. This parameter allows the user to specify the distance outside of the actual data in which the parameter values will be extrapolated. The distance is a percentage of the diagonal extent in the X-Y plane. The default is 0.10 (10%). For example: if data extent is 100 in x and 100 in y, diagonal distance is 144. It will make the convex hull offset 14.4 total by offsetting 7.2 on all sides. This eliminates dangerously large offsets when data extent is small.

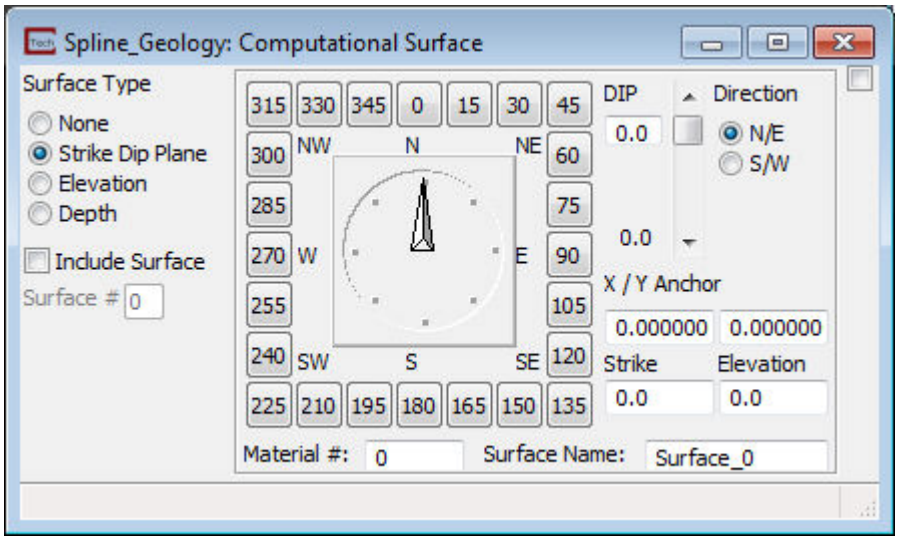
The ***Default Units*** parameter allows the user to set the coordinate units of the model if the file being read does not contain them. If the file does contain coordinate units, this parameter is ignored.

To set up a finite difference grid, the Gridding options checkbox should be checed and the Finite Difference Grid Type should be selected, which will bring up the X and Y check boxes, and the Angle data input field. The angle input field allows the user to specify an angle in degrees counter-clockwise from true north that the resulting grid will be rotated. The finite difference grid corresponding to the input parameters can be visualized at any time by clicking the **Accept All Current Values** button (before a .geo file has been specified in the file browser). The *Finite Difference Gridding* subpanels for X and Y allow the user to constrain the modeled domain to be within the finite difference grid setup using the Krig_3D_Geology control panel. Note the X and Y values specified in these subpanels override any other specified values for Min and Max X and Y in the Gridding Options. Clicking on the X or Y subpanels in finite difference gridding brings up a subpanel similar to the one shown in the figure below.



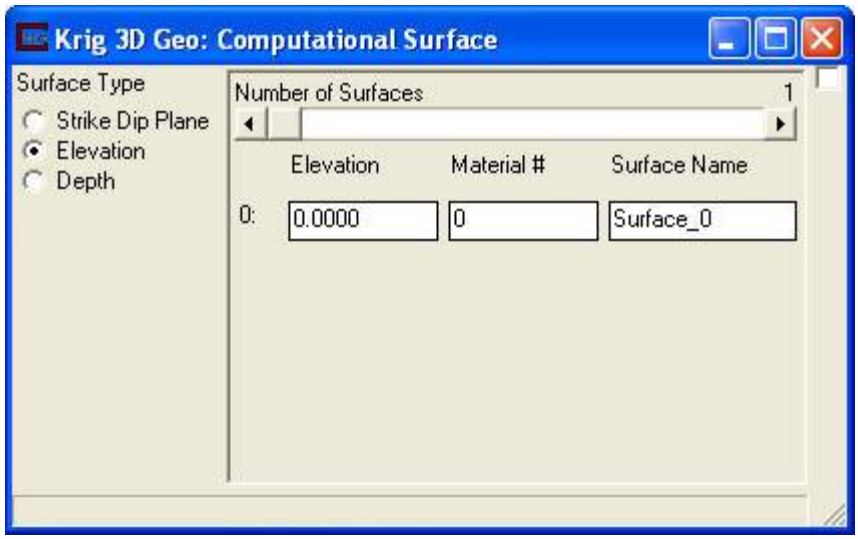
The first parameter is a slider to specify the total ***Number of Steps*** (grid regions). There is no limit to the number of steps (regions) and each region can any number of elements (cells) and can have constant size cells, decreasing (Cx < 1.0) or increasing (Cx>1.0).

The user builds the finite difference model grid by specifying the distance between the first and second grid nodes (or the element width) in the Del X input field. The number of additional nodes to be placed at this spacing is then specified in the Xr (standing for "X repeat") input field. If a gradually varying node spacing is desired between the number of nodes specified in Xr, then a multiplication factor is specified in the Cx (standing for "Change X") input field. Values greater than 1.0 create an increasing element size, and values less than 1.0 create diminishing element sizes. The grid design process is identical for the Y grid panel. These panels allow the user to quickly design and visualize a rectilinear model grid.

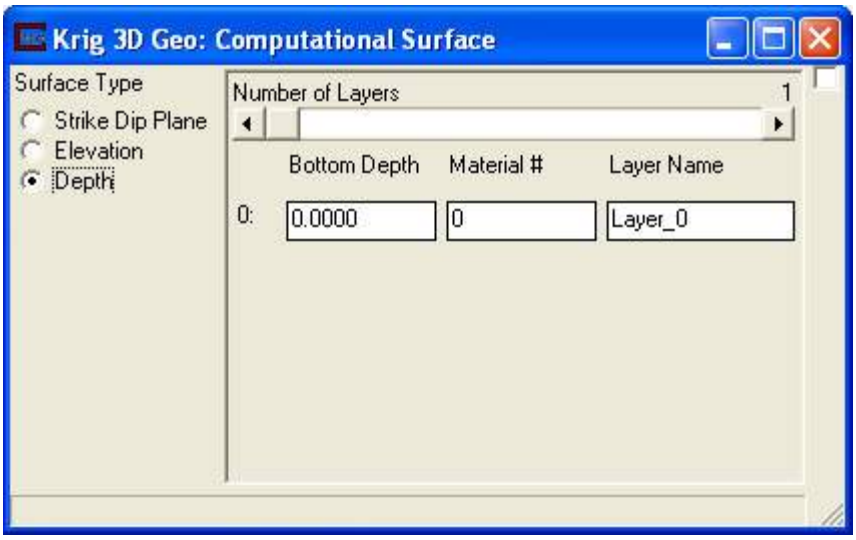


The **Create Computational Surface** panel is shown above. This method will create three different types of surfaces based upon the created grid. A file is required for a computational surface to be created. The first step is to create the desired grid using the Gridding Options subpanel discussed above.

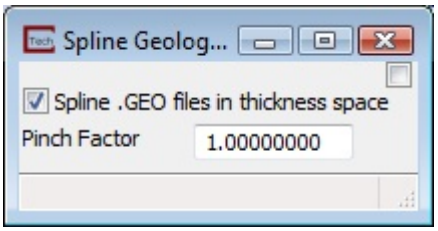
The *Surface Type* radio box is the next step. There are three different types of surfaces that can be created. The Strike Dip Plane option consists of a single surface that can be rotated to match both strike and dip values. This option also requires an 3D coordinate point to center the rotation plane. This coordinate point is entered into the *X / Y Anchor* fields as well as the *Elevation* field. A *MaterialNumber* and *Surface Name* can also be entered for this computed surface.



The *Elevation* suface parameters can be seen in the image above. This surface type will create any number of surfaces all with the same X, Y coordinates created by the gridding options, but at set elevations. The *Material Number* and the *Surface Name* can be set for each surface.



The Depth surface paramets can be seen above. This method will Krige the selected file onto the grid created using the gridding options. It will then create any number of layers at specified depths below the top surface. The *Material Number* and *Layer Name* can be set for each layer.



Data Processing Options

The only parameters that affect the output shape of the resultant surfaces are:

- The ***Pinch Factor*** parameter provides the ability to control where pinching occurs between positive thicknesses and borings having the pinch flag. It defaults to 1.0 which causes pinching to occur approximately half-way between positive thicknesses and borings having the pinch flag. When older applications created before version 9.5 are loaded they will have a value of 0.0 for backwards compatibility.
- The ***Krig .GEO files in thickness space*** toggle off causes GEO files to be kriged like GMF files. Each surface get kriged independently of the other surface instead of being kriged in thickness space. This only applies to GEO files without the $W/$G flags.

The Spline_Geology module uses the same gridding and computational surface functionality as Krig_3D_Geology. Some of the images above are from Krig_3D_Geology because their function is identical.

make_geo_hierarchy

General Module Function

The make_geo_hierarchy module reads a special input file format called a pgf file, and then allows the user to build geologic surfaces based on the input file's geologic surface intersections. This process is carried out visually (in the EVS Viewer) with the use of the make_geo_hierarchy user interface. The surface hierarchy can either be generated automatically for simple geology models or for every layer for complex models. When the user is finished creating surfaces the gmf file can be finalized and converted into a *.GEO file.

Module Input Ports

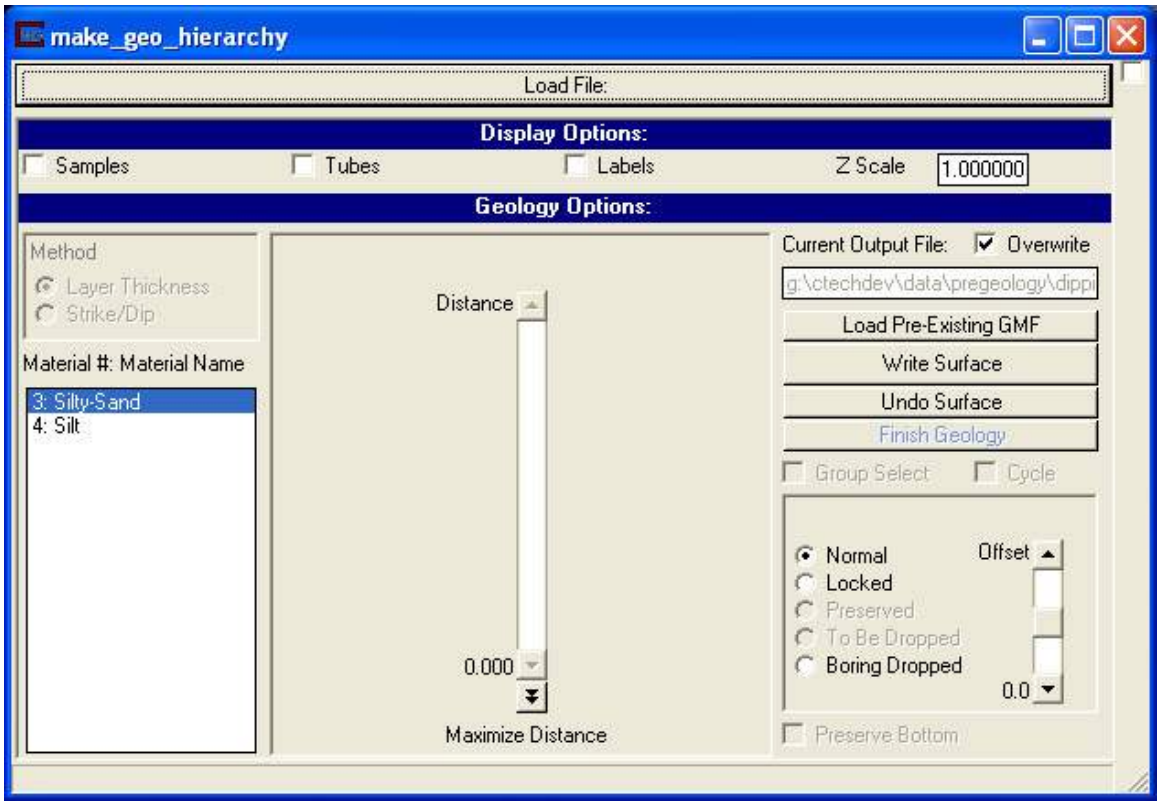
This module is a stand-alone module and has no input ports.

Module Output Ports

make_geo_hierarchy has two output port.

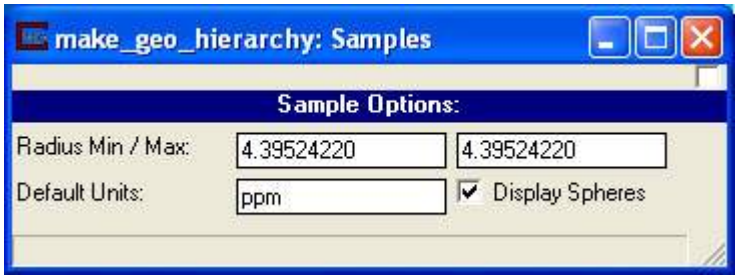
1) geologyInfoOut (Brown, Grey, Light Brown, Dark Yellow) : This port outputs geologic material values or geologic layer names to the Legend module.

2) out_obj (Red): This port outputs a renderable geometry to the viewer.



Module Control Panel

The main panel for make_geo_hierarchy is shown above. The Load File button activates a browser for selecting any file with a .pgf extension.

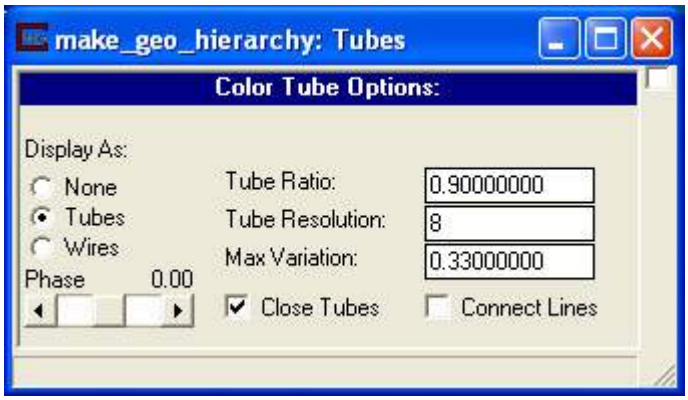


Display Options:

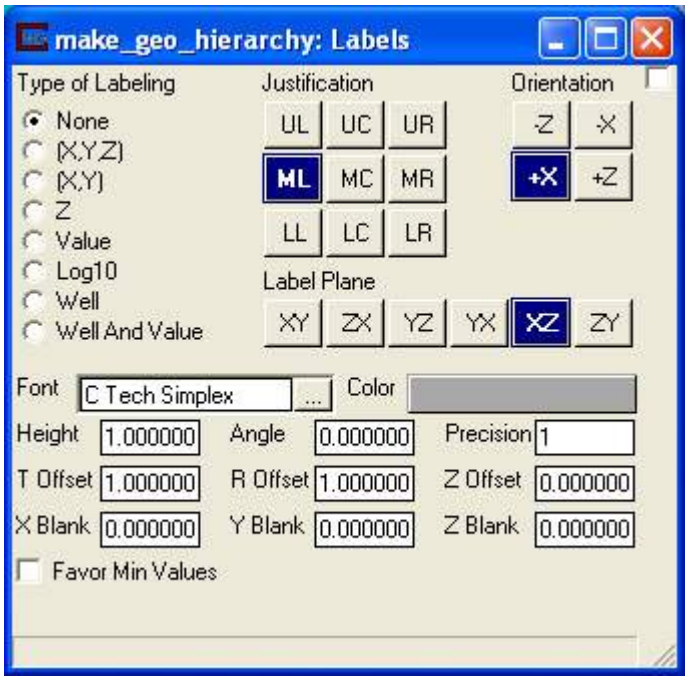
The Sample toggle will bring up the window above when selected.

The **Radius Min / Max** fields control the radius of the spheres.

The **Display Spheres** toggle when selected will render a sphere at each sample location.



The Tubes toggle will bring up the above window. This window has the same options as the Color Tube Options window in post_samples and the help for it can be found there.

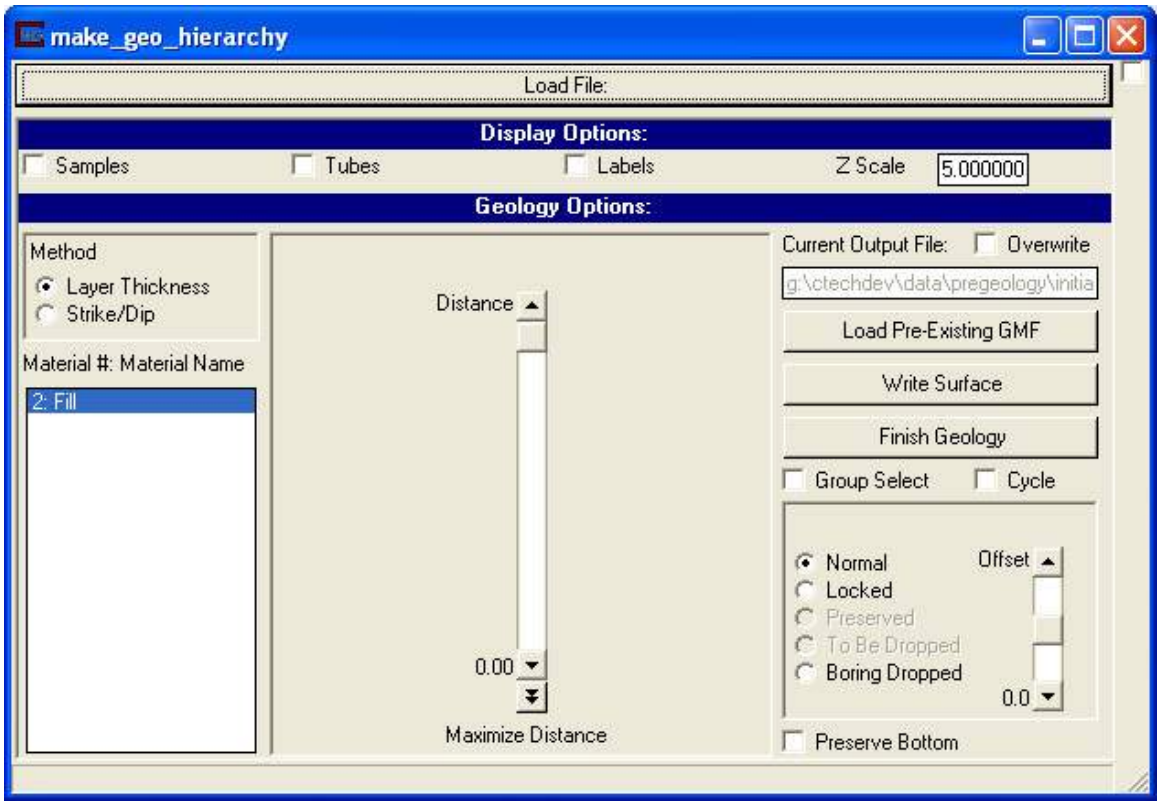


The Labels toggle will bring up the above window. This window has the same options as the Label Options window in post_samples and the help for it can be found there.

The **Z Scale** field affects the veritical exaggeration of the rendered model.

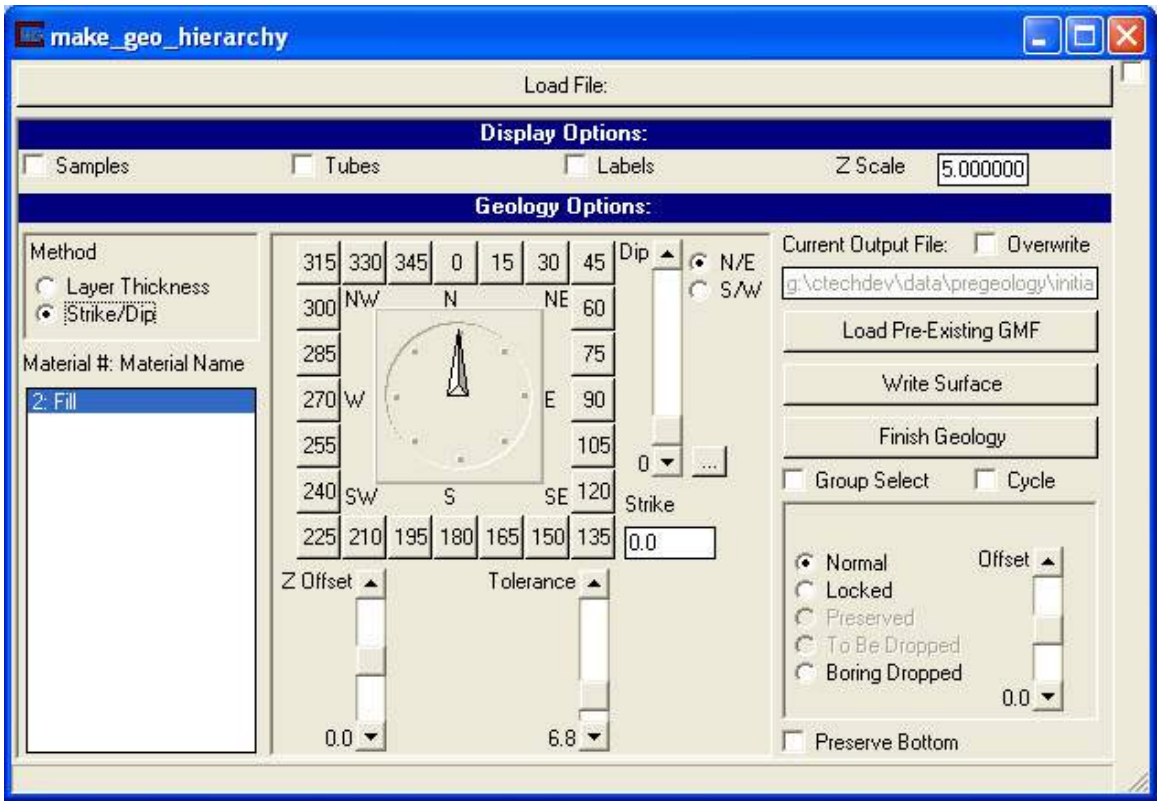
Geology Options:

The first step to creating a geologic hierarchy is to load a *.pgf file. The next step is to either load a pre-existing GMF or to select the material of the top surface and then write the top surface. The **Material #: Material Name** list box is used to select the material for the current suface. Only materials that are at the position of the current surface will be available to be selected. Until at least one surface has been created or loaded none of the hierarchy options will be available. Once the top surface has been created the user should select a **Method** for generating subsequent surfaces.



The *Layer Thickness* option is shown in the image above. This method generates layers based on the thickness of the chosen material. The benefit of this method is that it can quickly make simple geology models. The drawback to this method is that complex models can take a large amount of time.

The **Distance** slider reflects the position of the surface being created. The algorithm will look at the material of each boring at the current position. It will then see how far down each boring it can go until it finds the bottom of the boring or a different material. The double down arrow button below the slider will set the slider position to its maximum.



The *Strike/Dip* option is shown in the image above. This method generates layers based on their distance to a generated strike and dip plane. This method is useful for generating complex hierarchy models.

The Z Offset slider is used for visualization purposes only. It allows the user to change the position of the plane so that the strike and dip are easier to match if they are not known.

The Tolerance slider indicates the distance from the strike/dip plane that the bottom of current material can be. If the material continues below the plane past the tolerance value. The material will be split at that location.

The **Current OutputFile** text field contains the GMF filename that is being created. By default this name is created based upon the PGF filename selected. However a pre-existing GMF file can be read in using the **Load Pre-Existing GMF File** button. **NOTE:** Only GMF files that were created using this module, or those that have been altered to fit the format of this module's GMF output files should be loaded in this way.

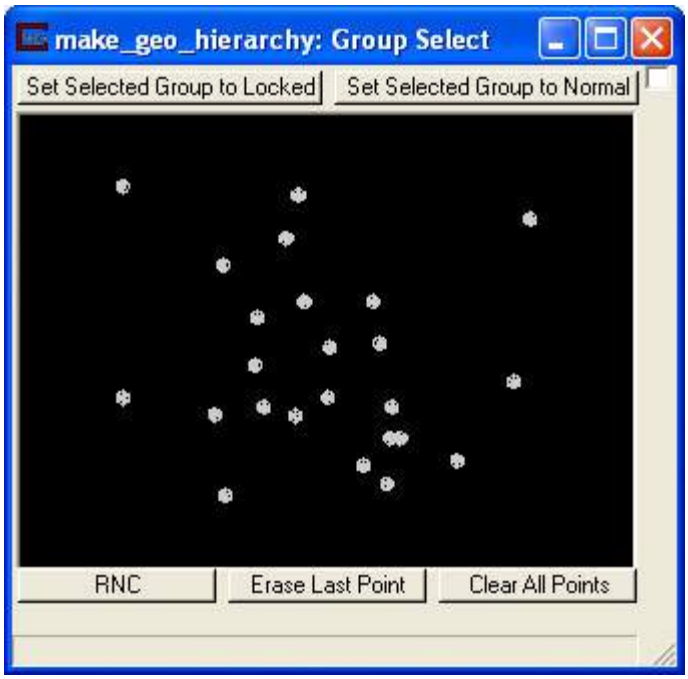
The **Overwrite** toggle causes the GMF file in the Current Output File text field to be overwritten when the Write Layer button is selected. This toggle is on be default but automatically turned off when a GMF file is loaded using the Load Pre-Existing GMF button, or after the Write Layer button is selected.

The **WriteSurface** button will write the position of every boring in the TIN surface and the current selected material to the GMF file. If the Overwrite toggle is on it will overwrite the current file, else it will append the surface to the GMF file. It will also look at the position of the surface at each boring, if the surface is at the bottom of a material and there is another material below

it, it will add the material below to the current selected material list. If the surface is at the bottom of the boring it will remove that boring from the TIN surface. **NOTE:** If a boring is at the bottom of a surface but other borings continue, a commented flag will be placed in the GMF file. This commented flag will not affect the output of the GMF file but will be taken into account when creating a GEO file with the Finish Geology button, where these flagged values are turned into SHORT flags. Finally if there are any locked borings (discussed below) then they will be unlocked.

The **Undo Surface** button will remove the last surface from the file. **NOTE:** This button affects file contents not just the memory resident surface.

The **Finish Geology** button should be used when the GMF file is complete. This will convert the GMF to a GEO file.

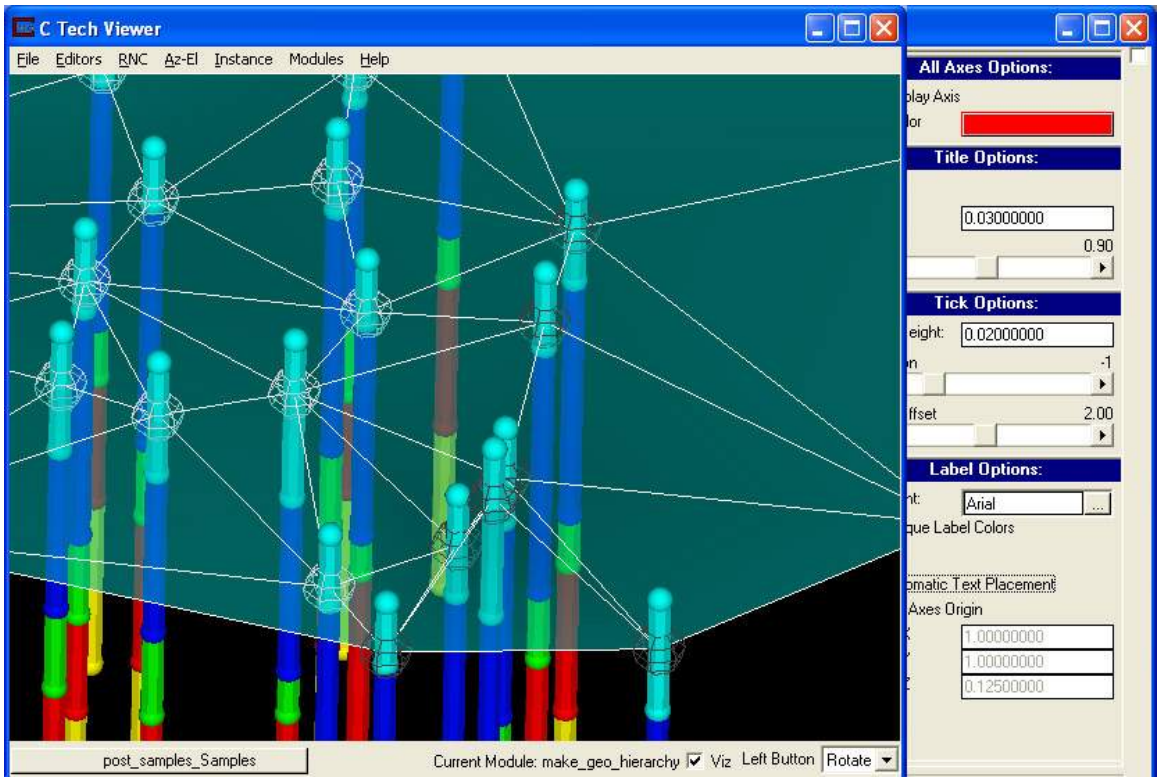


The **Group Select** toggle will bring up the window shown above. This window gives a Top View of the pgf file samples and is useful for selecting groups of borings to lock or unlock. Use the Alt + Right Mouse button to draw a polygon enclosing a group of spheres and then set their state to either Locked or Normal. The state of an individual boring can toggled using the Alt + Left Mouse button. A white sphere means the boring is in the Normal state and a grey sphere means it has been Locked. This view can also be translated by clicking and dragging inside the window.

The **Cycle** toggle will cause the module to automatically generate layers using the set parameters until it reaches a stopping condition. The stopping conditions include that it has reached the bottom of the model or that the next layer it creates would be of the same material as the one above it. The latter case can occur when using the strike/dip plane method with a tolerance that is to small. For the layer thickness method the algorithm will set the distance to the maximum extent and then write out that layer. It will

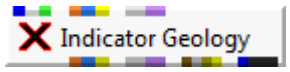
then pick a material from the available materials at that location and continue.

The window above appears when any sample along a boring is selected. By locking a boring, the user is keeping the surface from descending along that boring. The locked status of all boring will be cleared when the Write Layer button has been selected. All borings have a white where mesh sphere on them to indicate where they are a part of the TIN surface being created. When the boring is locked the mesh sphere will turn grey as shown in the picture below



The boring can also be dropped from the TIN entirely, or have its current position changed.

Indicator_Geology



This is an EVS-PRO & MVS module only.

Indicator_Geology is an alternative geologic modeling concept that uses the *indicator* method for assigning each cell's lithologic material as defined in a pregeology (.pgf) file, to cells in a 3D volumetric grid.

General Module Function

Module Input Ports

Indicator_Geology has an Orange-Blue-Yellow input port that can receive the geology file name and a (gray-magenta) port outputs the refine distance to Pre_Geology.

Indicator_Geology has three input ports.

The leftmost (blue/white/green) port can accept a data field from Krig_3D_Geology to constrain indicator kriging between two topographic surfaces.

The second (Orange-Blue-Yellow) port receives the pre-geology file name.

The third port (gray-magenta) port receives the refine distance.

Module Output Ports

Indicator_Geology has three output ports.

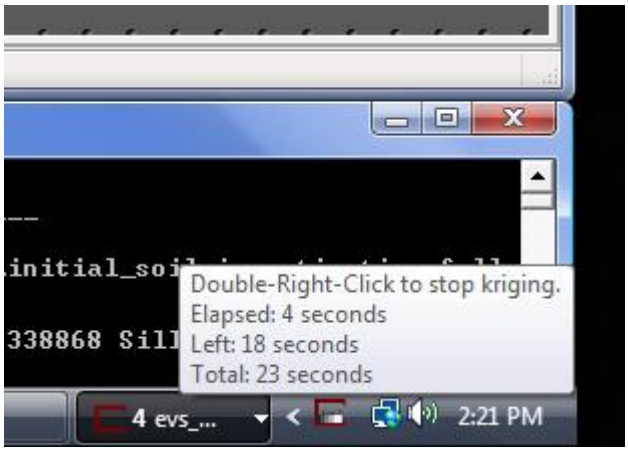
The first (Orange-Blue-Yellow) port supplies the pre-geology file name to Pre_Geology.

The second (gray-magenta) port outputs the refine distance to Pre_Geology.

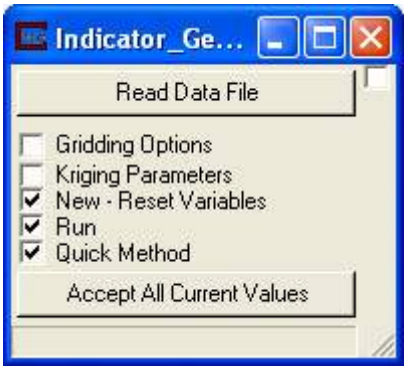
The third port (blue-black) is the primary output field containing cell data representing geologic materials.

Module Status: **Interruptible**

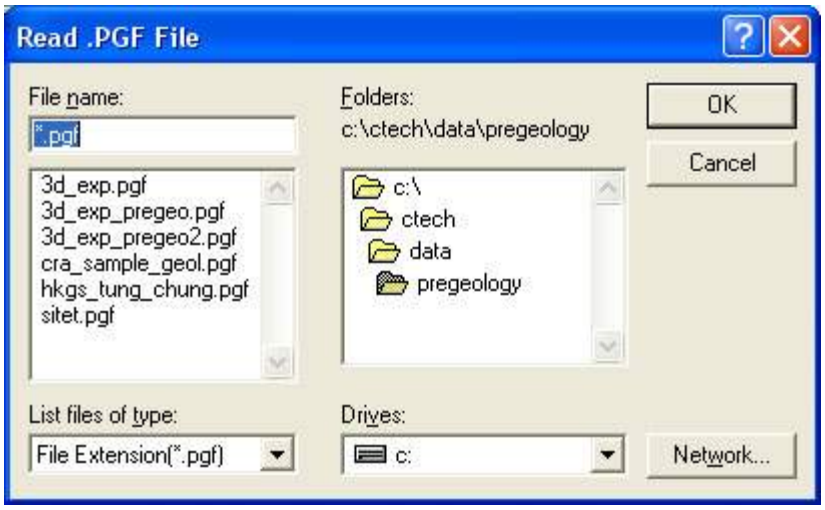
This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel



The control panel of Indicator_Geology is shown in the figure above. The **ReadDataFile** button opens the *File Browser* panel shown in the figure below.

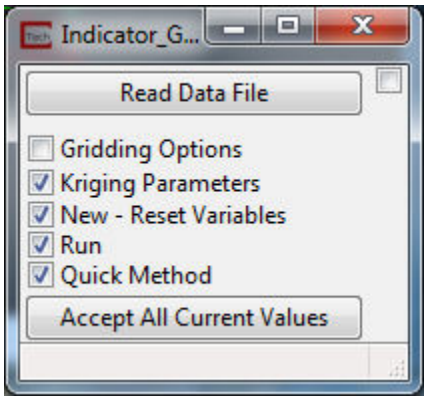


The file browser is a standard window browser that allows the user to specify the directory in which the files reside, and the extension used to filter the displayed available file names. For Indicator_Geology, the only file extension is .pgf. The format of .pgf files is described in the pgf_file_format Help topic.

The "*Run*" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed. When it is off, the module will not run.

The *Quick Method* toggle removes Semivariogram Parameters as an option and utilizes a substantially simplified algorithm for indicator kriging. In many cases the results are comparable to the more rigorous method and it is **always much quicker**.

Note: The "new" toggle on the main panel (on by default) resets all expert system calculated variables to zero before each run. This allows multiple calculations to be performed without tedious manual resetting of these variables. To change an expert system calculated variable, the toggle must be off.

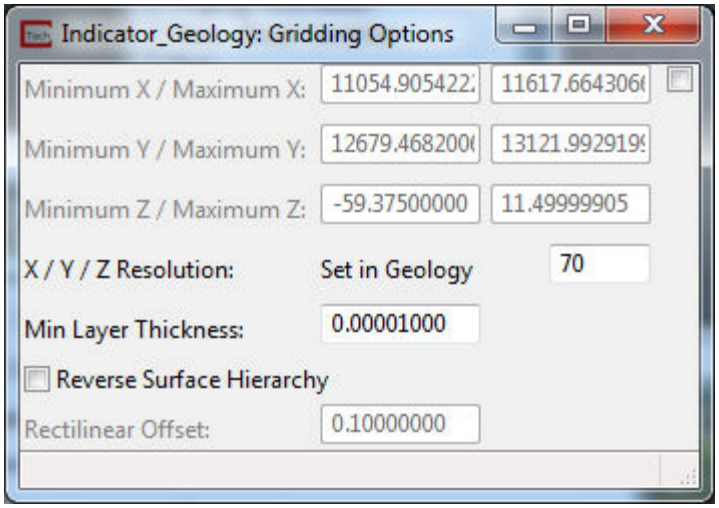


1) The *Quick Method* assigns the lithologic material cell data based on the nearest lithologic material (in anisotropic space) to your PGF borings. This is now done based on the cell center (coordinates) and an enhanced refinement scheme for the PGF borings. In general the **Quick Methodshould not be used for final results.**

2)For the morerigorous probabilistic approach to geologic indicator kriging, the probability for each material is computed for each cell center of your grid. The material with the highest probability is assigned to the cell. All of the individual material probabilities are provided as additional cell data components. This will allow you to identify regions where the material assignment is somewhat ambiguous. Needless to say, this approach is much slower (especially with many materials), but often yields superior results and interesting insights.

Module Parameter Subpanels

Indicator_Geology has two subpanels which allow the user to set the parameters used for *Gridding Options* and setting the *KrigingParameters*. Clicking on either the check boxes next to the subpanel names, or on the names themselves will bring up the subpanel data entry screens.



The ***Gridding Options*** subpanel of Indicator_Geology is shown in the figure above.

The ***Minimum X, Maximum X, Minimum Y, Maximum Y, Minimum Z and Maximum Z*** parameters allow the user to define the horizontal and vertical domain within the data set in which kriging of the parameter distribution will be completed. A value of 0 is the default for these parameters, which results in a model domain that is defined by the rectilinear bounds of the entire data set when the module is run. Indicator_Geology utilizes a model domain that is bounded by the limits of the data set, unless the user specifies a different domain by setting the Min and Max Values for X,Y, and Z, or it is passed a model (or finite-difference) domain from Krig_3D_Geology. Utilizing the default extents effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points. Note that when a geologic model domain is passed to Indicator_Geology, the kriging domain is restricted to that domain regardless of what X, Y, or Z values are set in the Indicator_Geology inputs. The geologic model domain can be bounded by the convex hull of the data set, or by some specified offset (see discussion in Krig_3D_Geology). If the user is uncertain of the X, Y and Z limits of the data domain, the module should be run with the default 0 values, and upon completion of execution, the values in the X, Y, and Z input fields will be the min and max values of these parameters in the data set. The File Statistics module can also be used to investigate the limits and distribution of values in the input data set, and the statistics module can be used to output the distribution of values in the kriged model.

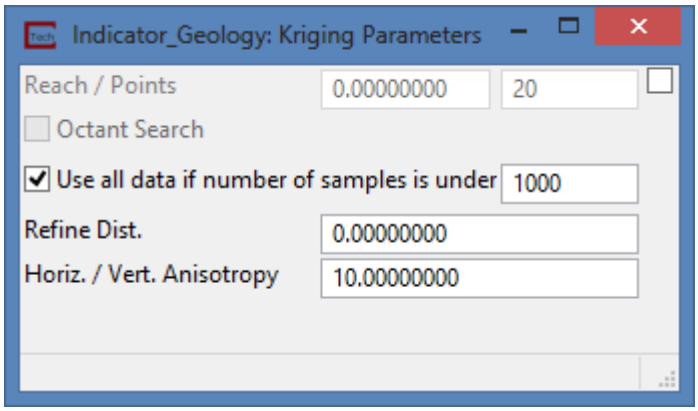
The ***X / Y / Z Resolution*** parameters specify the number of grid nodes that will be included within the model domain. If Indicator_Geology is being passed a model domain from Krig_3D_Geology, then the x-y values are not used . The number of grid elements along any axis of the model is simply the axis Res value minus one, as every element has two bounding nodes along an axis. The default value for the X and Y Res parameters is 41, and the default value for the Z Res is 35. However, the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user.

If geologic layers are being passed to Indicator_Geology, then the number of nodes specified in Z Res will be distributed over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. However, in all cases, at least three layers of nodes (2 layers of elements) of the Indicator_Geology domain will be placed in each geologic layer.

Reverse Surface Hierarchy - This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

Minimum Layer Thickness forces layers thinner than the specified value to have a minimum thickness and therefore not pinch-out completely.

The ***Rectilinear Offset*** parameter is used to create an automatically generated rectilinear domain which is larger than the input data set by a user input percentage along all three axes. If 10% (0.1 default value) is used, the limits will be offset by 5% on all 6 sides of the rectilinear volume.



The ***Kriging Parameters*** subpanel of Indicator_Geology is shown in the figure above.

The ***Reach*** input field defines the radial distance (in user units) from any given model node that the kriging module will look for data points to be included in the estimation of the model parameter at that node. The default value of reach is 0, which results in the module calculating a reach value which is approximately two-thirds of the longest distance between any two data points in the data set.

The ***Points*** parameter defines the maximum number of data points (within the specified reach) that will be considered for the parameter estimation at a model node. The default value for points is 20, which generally provides reasonably smooth modeled parameter distributions. The effects of decreasing and increasing the values for reach and points on the model output are somewhat similar, but for different reasons. If the data have a fairly even spatial distribution throughout the domain, then increasing these values will generally include more of the input data points that will be used to krige the value for a given model node, and thus will result in smoother modeled data distributions. Decreasing the values of reach and points (in an evenly distributed data set) results in fewer input data points being used to calculate the parameter estimates at a given model node, and result in modeled distributions with greater variations across smaller areas.

The user should consider both the spatial distribution and the range of values in the input data set when deciding upon values for the reach and points parameters. If the specified reach is too small to allow the kriging module to locate at least one point within the search area, then no kriging will be completed and the following error will be printed to the Status Window:

Matrix Singular. Increase REACH. Your reach is so small that no samples fall within the search radius.

If the user specifies a large number of points (that are within the specified reach), then the output will be smoother, but the execution time for the kriging can increase significantly. By posting the input data using the *post_samples* module, and looking at the characteristics of the resulting kriged data using the *Statistics* module, the user can quickly analyze the characteristics and distribution of the kriging output for a given set of parameters, and test the effects of changing the kriging parameter values.

The ***Octant Search*** toggle changes the method by which data sample points are selected for inclusion in the kriging matrix. If this is on, the "Points" parameter switches to "Max Points in Octant". Searching is performed for each of the eight Octants surrounding the point to be kriged. Within each octant a maximum number of points (up to one-fourth of the total points) are selected. Then, points are taken sequentially from each octant up to the maximum number of total points or until all octant's points have been used. The panel display changes when this option is selected as shown above. Octant Search is applicable for the Statistics and Min-Max Plume kriging modes. *Octant Search cannot be used with the Quick Method.*

Understanding the ***Refine Dist***(ance) parameter requires an understanding of the basics of indicator kriging with a .PGF file as input. Since a PGF file has only a single point defining the bottom of intervals of a specific material, it is necessary to create an intermediate dataset that is used for the actual kriging. Since each interval represents a line segment with constant material characteristics, we approximate it by refining the interval into a set of points. The spacing between these points affects accuracy. Each segment will receive at least two points near the end points of the interval, but longer segments will broken up into segments no larger than the refine distance. Setting this value too small will result in much greater run times, as well as requiring higher values for the points parameter. Setting this value too large can create artificial inaccuracies in the prediction because the center of the intervals can have a lack of sample points, causing points in other borings to appear closer. If you leave the default setting of 0.00 for this parameter the expert system determines the spacing for you by searching the original PGF file for the shortest segment in any boring. The calculated refine distance will be twice the shortest segment length or one half the height of the first cell in the grid, whichever is larger. This tends to give reasonable results with most datasets. You can also use the Pre_Geology module to visualize the refined points directly by setting its refine distance parameter to be the same as the computed one in Indicator_Geology.

Since a PGF file has only a single point defining the bottom of intervals of a specific material, it is necessary to create an intermediate *dataset* that is used for the actual kriging. Since each interval represents a line segment with constant material characteristics, we approximate it by refining the interval into a set of points. The spacing between these points affects accuracy. If you leave the default setting of 0.00 for this parameter the expert system determines the spacing for you by searching the original PGF file for the shortest segment. This minimum distance will receive two points surrounding the center of the interval and longer segments will receive many more. Setting this value too small will result in much greater run times.

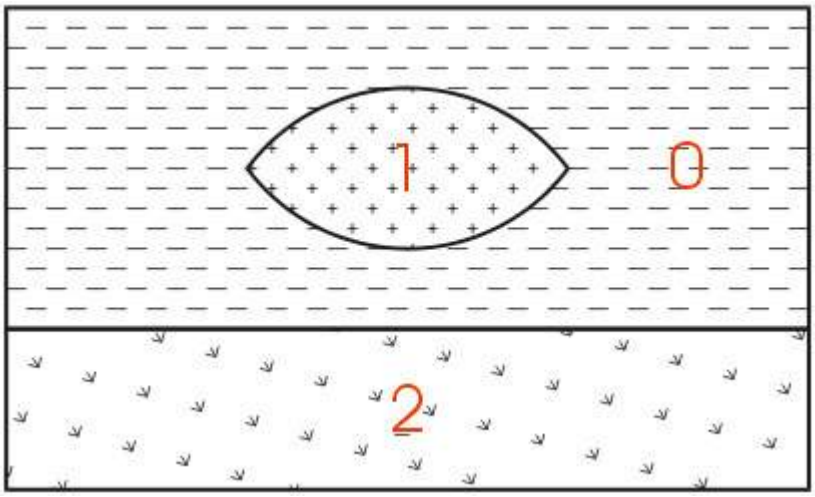
The ***Horiz./Vert. Anisotropy Ratio*** parameter allows the user to consider the effects of anisotropy in the conductivity of soil matrices to fluid flow. In most cases, geologic materials are deposited with platy clay minerals oriented horizontally, and thus flow of water in both the saturated and unsaturated zone can be slower in the vertical direction than in the horizontal direction. Also, ore deposition can occur along horizontal or vertical fault or

fracture systems. Chemical constituents being transported with flowing fluids may therefore show a larger degree of spreading in one or the other direction. The Horiz./Vert. Anisotropy Ratio basically tells the kriging algorithm what multiplication factor should be used to apply biased weighting on data points in horizontal and vertical directions away from a given model node. The default value is 10, which allows data points in a horizontal direction away from a model node to influence the kriged value at that node 10 times more than data points an equal distance away in a vertical direction. However, the user can specify any positive number with a magnitude up to 100,000. When the property being modeled is not related to fluid flow or other processes that might be affected by matrix anisotropy, then this parameter should be set to 1.

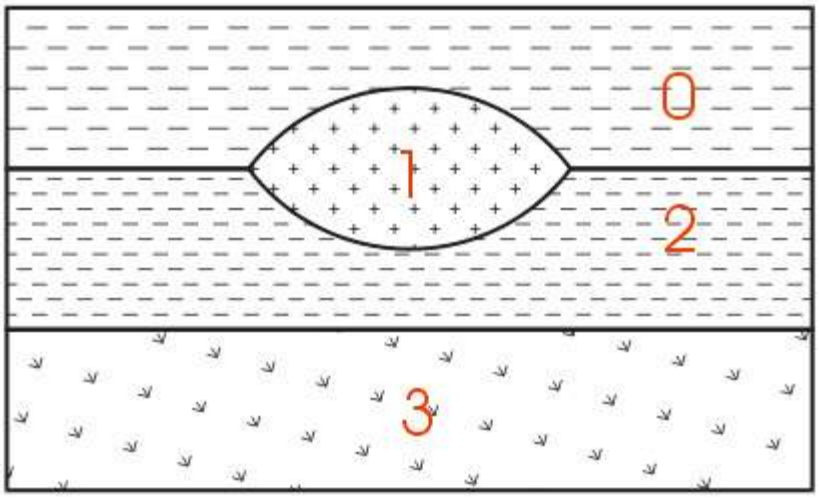
General Discussion of Geologic Indicator Kriging vs. Hierarchical Geologic Modeling

C Tech's original method for 3D geologic modeling utilizes one of two different ASCII file formats (.geo and .gmf) which contain "interpreted" geologic information. These two file formats both describe points on each geologic surface (ground surface and bottom of each geologic layer), based on the assumption of a geologic hierarchy.

The easiest way to describe geologic hierarchy is with an example. Consider the example below of a clay lens in sand with gravel below. Some borings will see only sand above the gravel, while others will reveal an upper sand, clay, and lower sand.

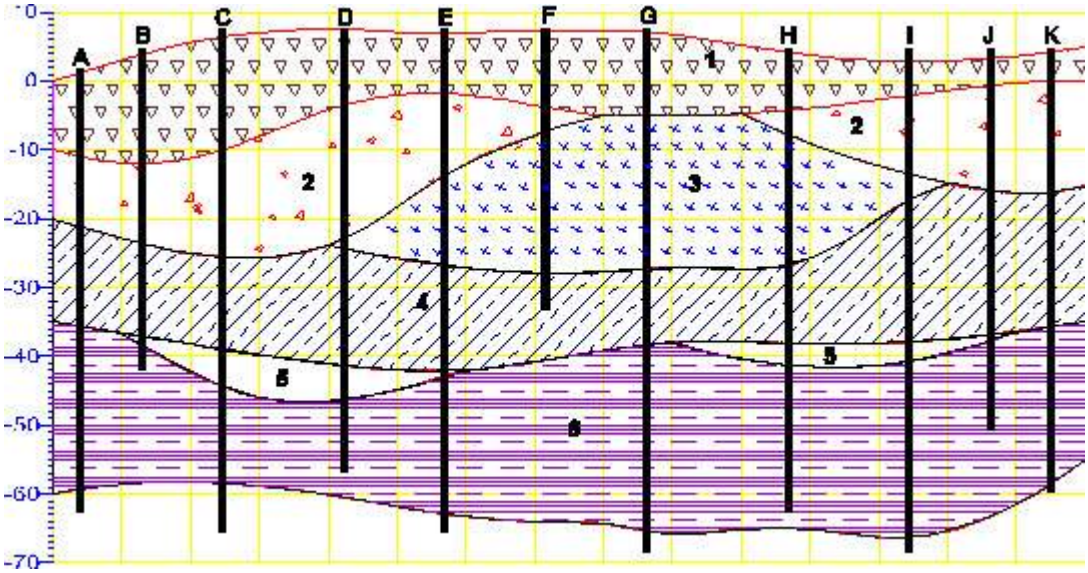


The geologic hierarchy for this site will be upper sand, clay, lower sand, and gravel. This requires that the borings with only sand (above the gravel) be described as upper sand, clay, and lower sand, with the clay described as being zero thickness. For this simple example, determining the hierarchy is straightforward. For some sites (as will be discussed later) it is very difficult or even impossible.



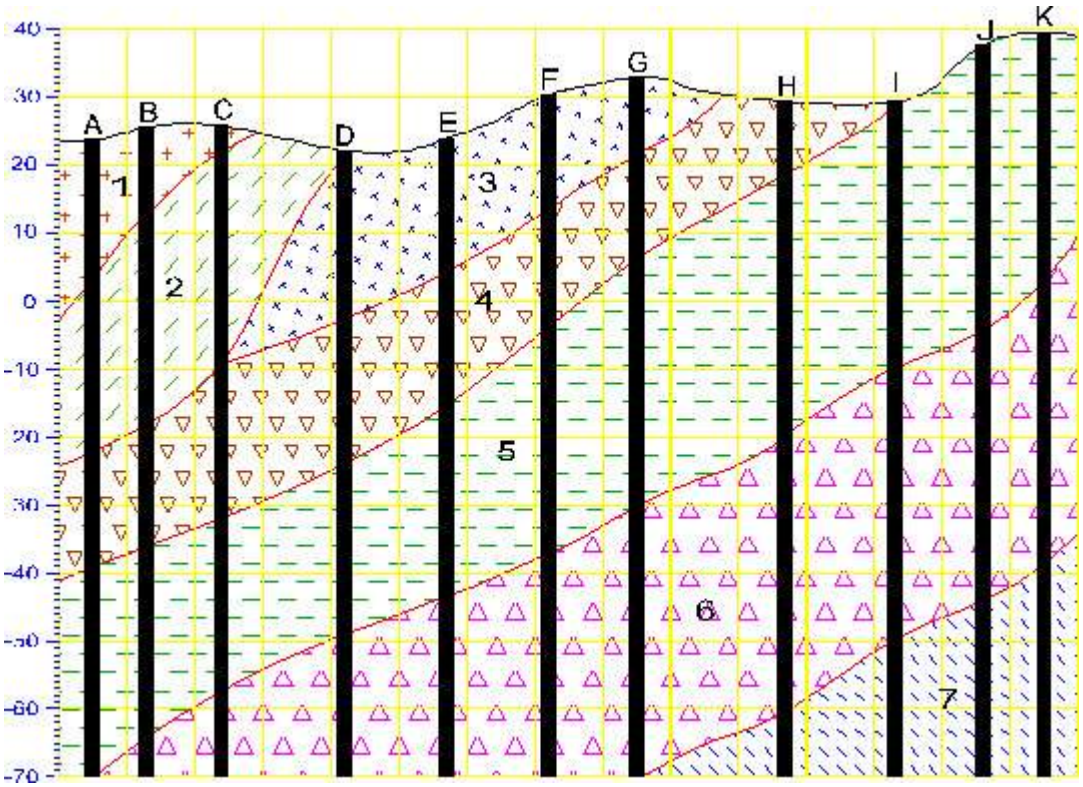
For those sites that can be described using the above method, it remains the best approach for building a 3D geologic model. Each layer has smooth boundaries and the layers (by nature of hierarchy) can be exploded apart to reveal the individual layer surface features. In the above example, the numbers represent the layer numbers for this site (even though layers 0 and 2 are both sand). Two examples of much more complex sites that are best described by this original approach are shown below.

Geologic Example: Sedimentary Layers and Lenses



Geology Example & Figure: Outcrop of Dipping Strata

EVS is not limited to sedimentary layers or lenses. The figure below shows a cross-section through an outcrop of dipping geologic strata. EVS can easily model the layers truncating on the top ground surface.

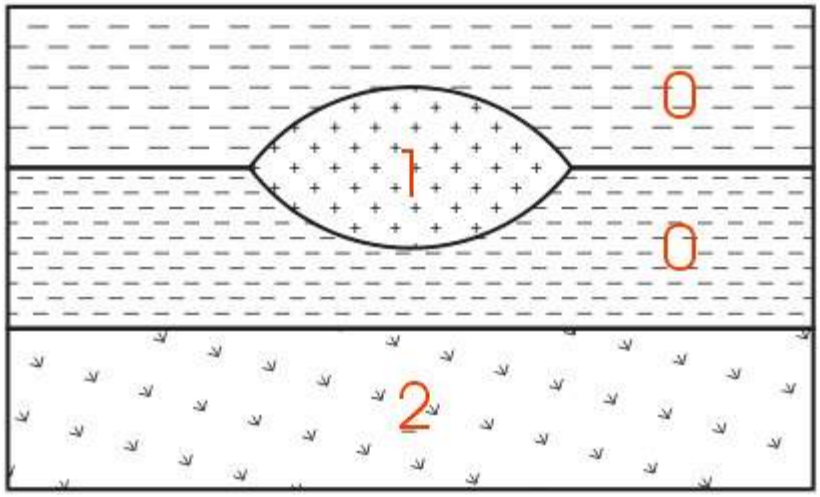


However, many sites have geologic structures (plutons, karst geology, sand channels, etc.) that do not lend themselves to description within the context of hierarchical layers. For these sites, Geologic Indicator Kriging (GIK) offers the ability to build extremely complex models with a minimum of effort (and virtually no interpretation) on the part of the geologist. GIK can also be a useful check of geologic hierarchies developed for sites that do lend themselves to a model based upon hierarchical layers.

GIK uses raw, uninterpreted 3D borings logs as the input file. The .pgf (pre-geology file) format is used to represent these logs. PGF files contain descriptions of each boring with x,y, & z coordinates for ground surface and the bottom of each observed geologic unit. Consecutive integer values (e.g. 0 through n-1, for n total observed units in the site) are used to describe each material observed in the entire site.

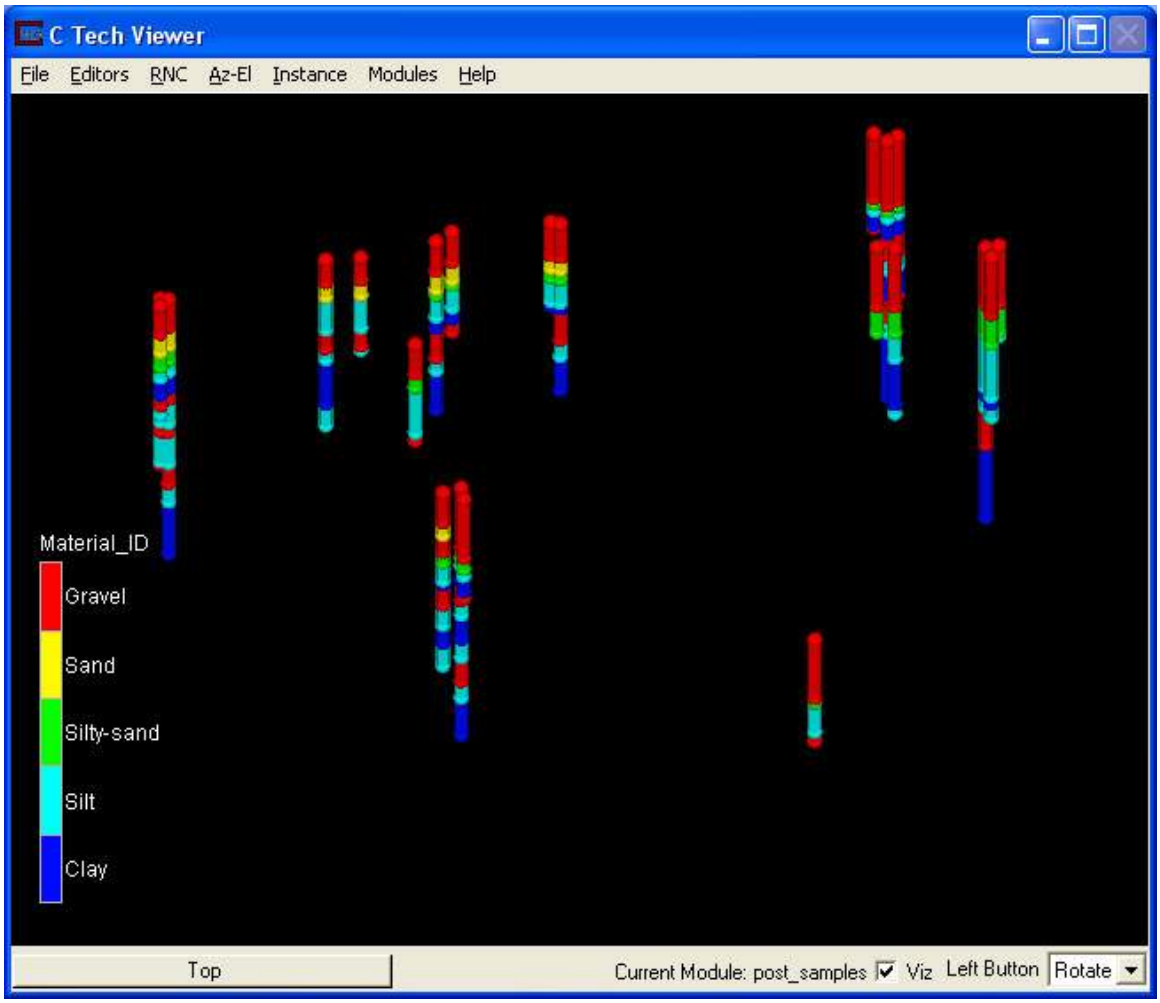
NOTE: It is important to start your material ID numbering at zero (0) instead of 1.

Usually, materials are numbered based upon a logical classification (such as porosity or particle size), however the numbering can be arbitrary as long as the numbers are consecutive (don't leave numbers out of the sequence). For the example given above, we could number the materials as shown in the figure below (even though it is not a numbering sequence based on porosity or particle size).



For a .pgf file, borings that do not see the clay (material 2 in the figure) would not need to consider the sand as being divided into upper and lower. Rather, every boring is merely a simple ASCII representation of the raw borings logs. The only interpretation involves classification of the observed soil types in each boring and assigning an associated numbering scheme.

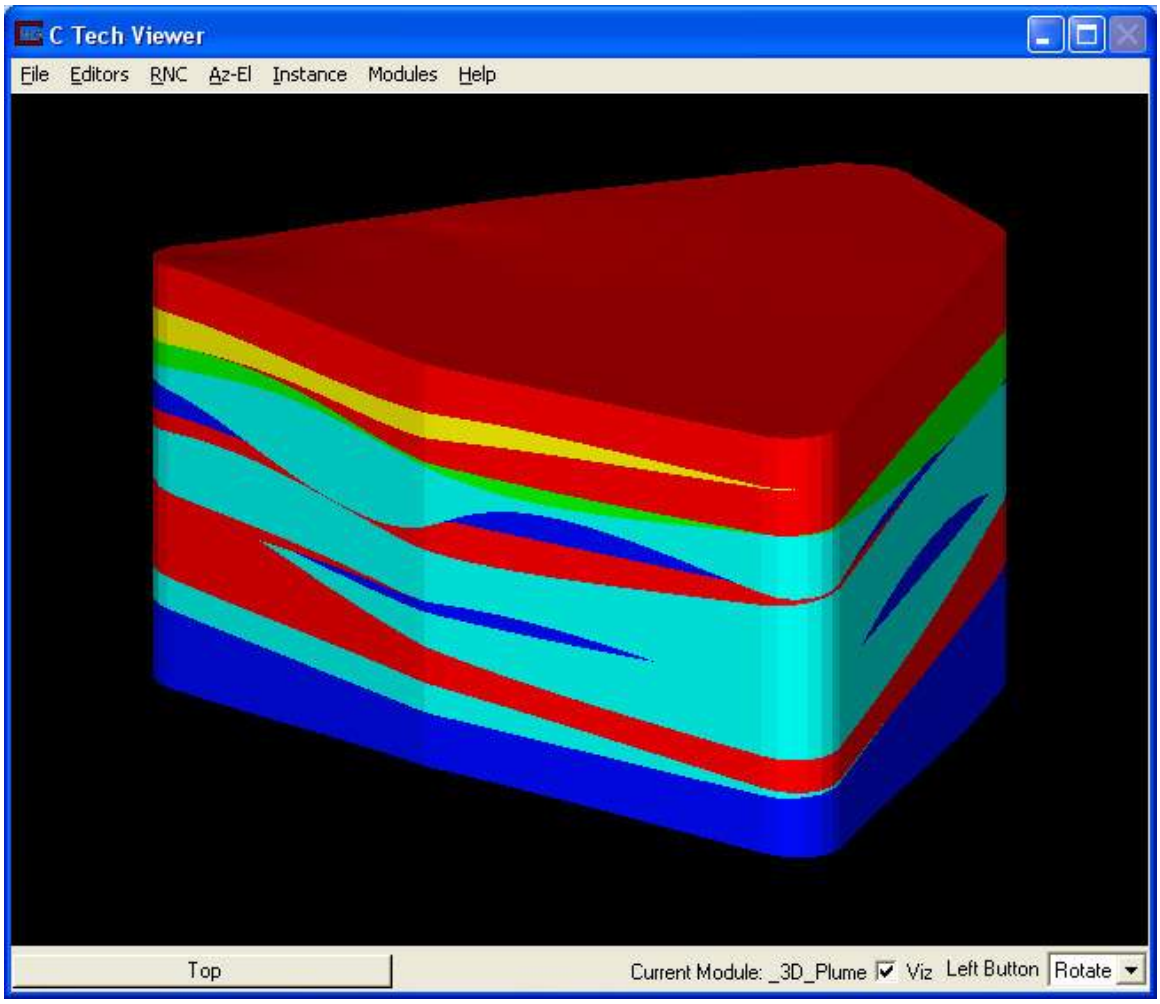
As an example, we have taken the Site T geologic data and created a model using both methods. This is a moderately complex site with geology that can be represented with hierarchical layering. However the complexity at this site does require a significant amount of interpretation (by a geologist or analyst) that lends a subjective element to the resulting model. The figure below shows the .pgf file for this site representing 5 unique materials.



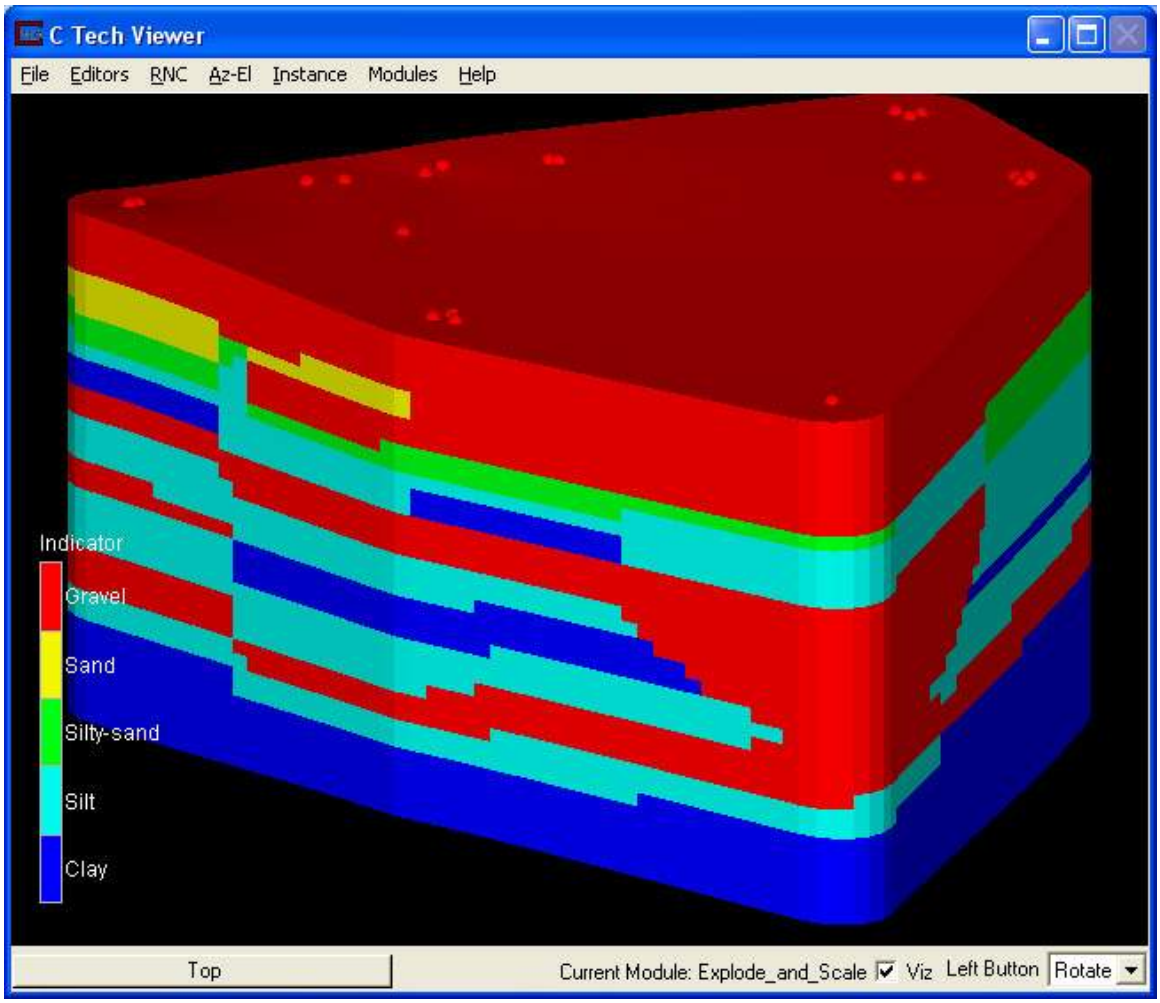
Note that there is a significant difference in the depths of some of the borings. Most notably, AW-3 (right foreground) is much shallower than the rest and is located quite a distance from any neighboring borings. With both techniques, extrapolation of the geology to the region below AW-3 will be low confidence and we expect that the two methods (Geologic Hierarchy and Geologic Indicator Kriging) will yield different results.

However both techniques should (and will) result in the same layering in the vicinity of the borings. This is a common theme in EVS for both geology and chemistry data. EVS will honor your measured data.

The figure below is the geologic hierarchy model for Site T. Note that the layers have relatively smooth boundaries and transitions (in thickness) from one boring to the next. Also note that the extrapolated geology below AW-3 is a logical extension from the neighboring borings. Please remember that a significant effort was required to interpret the .pgf file and create an interpreted geologic data file (.gmf file) that was used to create this model.



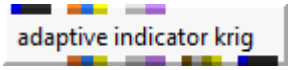
This next figure shows the model produced using GIK. The most notable difference is that the boundaries between layers are blocky. This is due to the totally different approach. With GIK, a default grid is created (per a user-specified fineness). This grid can be (and is in this case) defined between two geologic surfaces (e.g. ground surface and the bottom of gravel). The finer the grid (especially in the vertical direction) the more accurate the representation. GIK assigns the most probable material to each CELL in the model. The assignment to a cell is a significant distinction as compared with all other interpolation performed in EVS/MVS. If geologic data were assigned to nodes (vs. cells), in areas where there is a transition between two non-sequential material id's, colors would be interpolated between them that would correspond to intermediate materials. Using the cell data paradigm, we can have regions that transition from Clay to Gravel without seeing any other materials interpolated in between.



Also note that regions produced with GIK can have vertical transitions and folds. It is not encumbered by the layered approach of geologic hierarchy.

In conclusion, for this site, the decision of which approach is best is unclear. The greatest differences in the models are in areas where there is a lack of data. Ultimately a geologist familiar with the conditions should make that determination. For other sites, the choice between the two approaches is often quite obvious (especially if interpretation is too difficult).

adaptive_indicator_krig



This is an MVS module only.

General Module Function

Thisworks on an external grid, preserves nodal data,andcreates cell data.The input cell setsare not preserved, as it will create two cell sets per material, allowing you to remove specific materials with select_cells or Explode_and_Scale. Also,there aresomeknown, albeit rare,stability issues inside theadaptive gridding loops.

adaptive_indicator_krigis an alternativegeologic modeling concept thatuses the*indicator*method forassigning each node's material in a grid based on data

in a pregeology (.pgf) file. It is an extension of the technology in Indicator_Geology for several reasons:

1. Material assignments are done on a nodal versus cell basis providing additional inherent resolution
2. Gridding is handled by outside modules. This allows for assigning material data based on a PGF file after kriging analyte (e.g. chemistry) or other parameter data with Krig_3D.
3. Though it does not provide material boundaries that are as smooth as Krig_3D_Geology or Spline_Geology, it does provide much smoother interfaces than Indicator_Geology's *Lego-like* material structures.

There are two fundamental differences between Indicator_Geology and adaptive_indicator_krig

1. Geology / Grid input:
    1. Indicator_Geology expects input from modules like Krig_3D_Geology (which is a set of surfaces) and it builds you grid for you just as Krig_3D does.
    2. adaptive_indicator_krig is more like the "Kriging to an external grid" option in Krig_3D. You need to create the 3D grid (which doesn't need to have any data) that it will use. It will take that grid as a starting point for material assignments and later smoothing.
2. Lithologic Material Assignment
    1. Indicator_Geology assigns whole cells to cell sets and sets CELL data which is Material_ID.
    2. adaptive_indicator_krig takes the external grid and further refines it by splitting whole cells along all boundaries between two or more materials to create smoother interfaces.

Module Input Ports

adaptive_indicator_krig has three input ports.

The leftmost (blue/black) field port can accept a data field from Krig_3D, 3D_Geology_Map or other modules that have already created a grid containing volumetric cells.

The second (Orange-Blue-Yellow) port receives the pre-geology file name.

The third port (gray-magenta) port receives the refine distance.

Module Output Ports

adaptive_indicator_krig has three output ports.

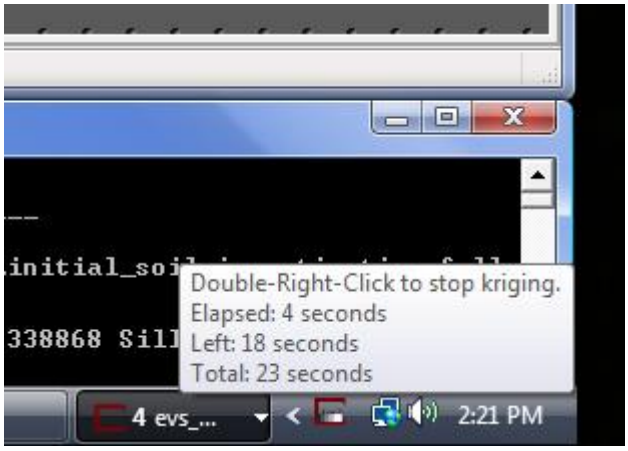
The first (Orange-Blue-Yellow) port supplies the pre-geology file name to Pre_Geology.

The second (gray-magenta) port outputs the refine distance to Pre_Geology.

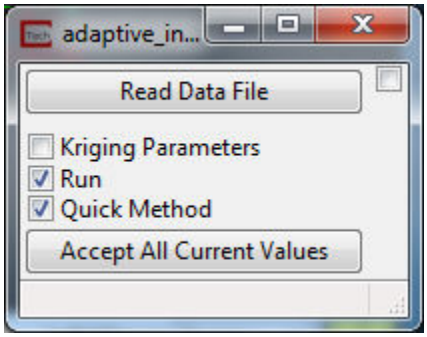
The third port (blue-black) is the primary output field containing nodal data and a refined grid representing geologic materials.

***Module Status*: Interruptible**

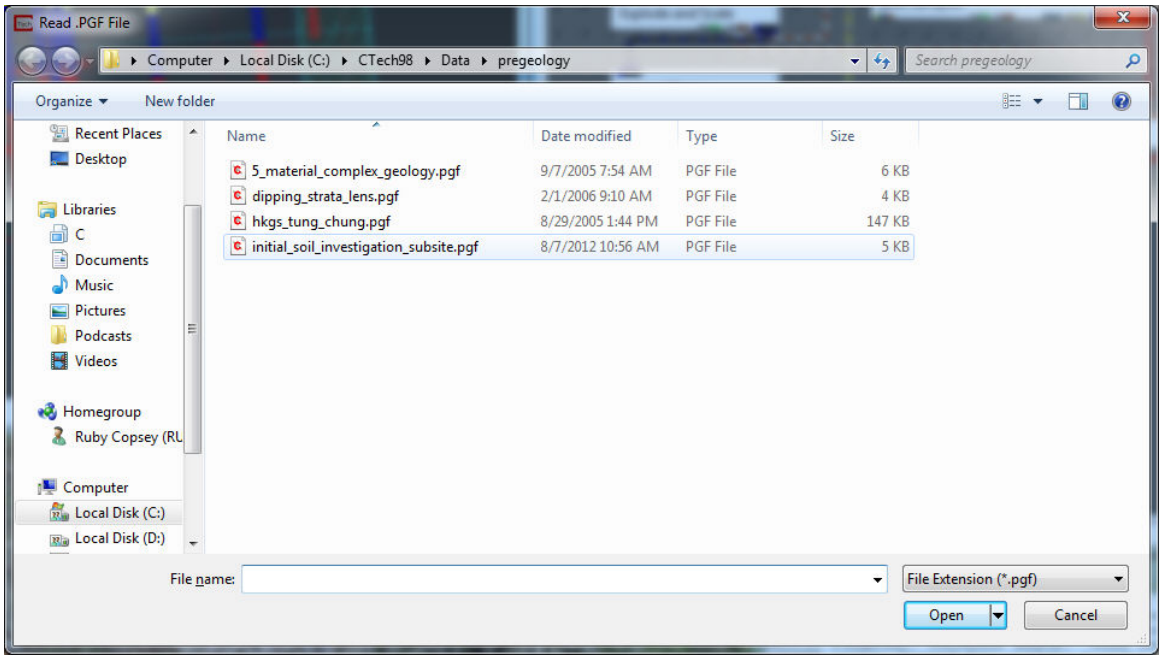
This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel



The control panel of adaptive_indicator_krig is shown in the figure above. The **Read Data File** button opens the *File Browser* panel shown in the figure below.



For Indicator_Geology, the only file extension is .pgf. The format of .pgf files is described in the pgf_file_format Help topic.

The "*Run*" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed. When it is off, the module will not run.

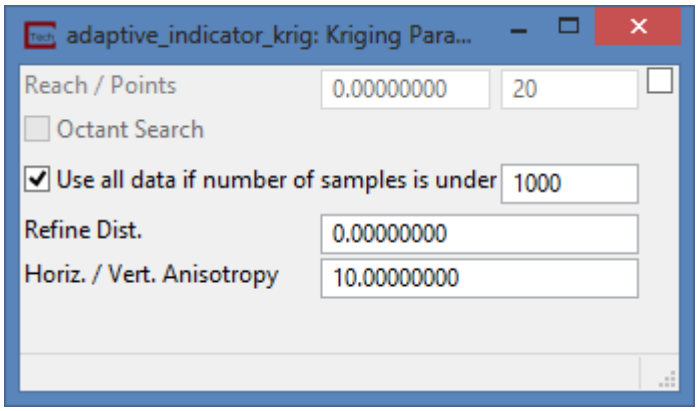
The *Quick Method* toggle determines the kriging method.

The*Quick Method*is similar (but improved) to the previous Geologic Indicator Kriging that was in Krig_3D. It assigns the geologic material data based on the nearest geologic material (in anisotropic space) to your PGF borings. This is done ona nodal basisand an enhanced refinement scheme for the PGF borings.

The ***Kriging Parameters*** subpanel with Quick Method selected is shown in the figure below.

For the morerigorous probabilistic approach to geologic indicator kriging, theindividualmaterialprobabilitiesfor eachnodeis computed. The material with the highest probability is assigned to thenode. All of the individual material probabilities are provided as additionalnodaldata components. This will allow you to identify regions where the material assignment is somewhat ambiguous. Needless to say, this approach is much slower (especially with many materials), but often yields superior results and interesting insights.

The ***Kriging Parameters*** subpanel with Quick Method selected is shown in the figure below.



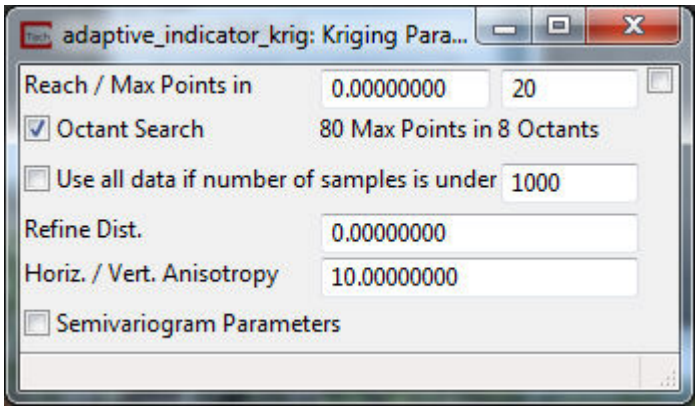
Note that the *Quick Method* toggle deactivates the Reach, Points and Octant Search options and utilizes a substantially simplified algorithm for indicator kriging.

The ***Reach*** input field defines the radial distance (in user units) from any given model node that the kriging module will look for data points to be included in the estimation of the model parameter at that node. The default value of reach is 0, which results in the module calculating a reach value which is approximately two-thirds of the longest distance between any two data points in the data set.

The ***Points*** parameter defines the maximum number of data points (within the specified reach) that will be considered for the parameter estimation at a model node. The default value for points is 20, which generally provides reasonably smooth modeled parameter distributions. The effects of decreasing and increasing the values for reach and points on the model output are somewhat similar, but for different reasons. If the data have a fairly even spatial distribution throughout the domain, then increasing these values will generally include more of the input data points that will be used to krige the value for a given model node, and thus will result in smoother modeled data distributions. Decreasing the values of reach and points (in an evenly distributed data set) results in fewer input data points being used to calculate the parameter estimates at a given model node, and result in modeled distributions with greater variations across smaller areas.

The user should consider both the spatial distribution and the range of values in the input data set when deciding upon values for the reach and points parameters. If the specified reach is too small to allow the kriging module to locate at least one point within the search area, then no kriging will be completed at that model node, the nodal value will be set to 0, and the confidence level will be set to <0.1%. Note that this nodal value is generally inappropriate, and the regions of the model receiving the 0 values should be subsetted out of the domain by using an plume_volume module with a confidence isocomponent of 1% or more. If the user specifies a large number of points (that are within the specified reach), then the output will be smoother, but the execution time for the kriging can increase significantly. By posting the input data using the *post_samples* module, and looking at the characteristics of the resulting kriged data using the *Statistics* module, the user can quickly analyze the characteristics and distribution of the kriging

output for a given set of parameters, and test the effects of changing the kriging parameter values.



The ***Octant Search*** toggle changes the method by which data sample points are selected for inclusion in the kriging matrix. If this is on, the "Points" parameter switches to "Max Points in Octant". Searching is performed for each of the eight Octants surrounding the point to be kriged. Within each octant a maximum number of points (up to one-fourth of the total points) are selected. Then, points are taken sequentially from each octant up to the maximum number of total points or until all octant's points have been used. The panel display changes when this option is selected as shown above. Octant Search is applicable for the Statistics and Min-Max Plume kriging modes.

Understanding the ***Refine Dist***(ance) parameter requires an understanding of the basics of indicator kriging with a .PGF file as input. Since a PGF file has only a single point defining the bottom of intervals of a specific material, it is necessary to create an intermediate dataset that is used for the actual kriging. Since each interval represents a line segment with constant material characteristics, we approximate it by refining the interval into a set of points. The spacing between these points affects accuracy. Each segment will receive at least two points near the end points of the interval, but longer segments will broken up into segments no larger than the refine distance. Setting this value too small will result in much greater run times, as well as requiring higher values for the points parameter. Setting this value too large can create artificial inaccuracies in the prediction because the center of the intervals can have a lack of sample points, causing points in other borings to appear closer. If you leave the default setting of 0.00 for this parameter the expert system determines the spacing for you by searching the original PGF file for the shortest segment in any boring. The calculated refine distance will be twice the shortest segment length or one half the height of the first cell in the grid, whichever is larger. This tends to give reasonable results with most datasets. You can also use the Pre_Geology module to visualize the refined points directly by setting its refine distance parameter to be the same as the computed one in Indicator_Geology.

Since a PGF file has only a single point defining the bottom of intervals of a specific material, it is necessary to create an intermediate *dataset* that is used for the actual kriging. Since each interval represents a line segment

with constant material characteristics, we approximate it by refining the interval into a set of points. The spacing between these points affects accuracy. If you leave the default setting of 0.00 for this parameter the expert system determines the spacing for you by searching the original PGF file for the shortest segment. This minimum distance will receive two points surrounding the center of the interval and longer segments will receive many more. Setting this value too small will result in much greater run times.

The ***Horiz./Vert. Anisotropy Ratio*** parameter allows the user to consider the effects of anisotropy in the conductivity of soil matrices to fluid flow. In most cases, geologic materials are deposited with platy clay minerals oriented horizontally, and thus flow of water in both the saturated and unsaturated zone can be slower in the vertical direction than in the horizontal direction. Also, ore deposition can occur along horizontal or vertical fault or fracture systems. Chemical constituents being transported with flowing fluids may therefore show a larger degree of spreading in one or the other direction. The Horiz./Vert. Anisotropy Ratio basically tells the kriging algorithm what multiplication factor should be used to apply biased weighting on data points in horizontal and vertical directions away from a given model node. The default value is 10, which allows data points in a horizontal direction away from a model node to influence the kriged value at that node 10 times more than data points an equal distance away in a vertical direction. However, the user can specify any positive number with a magnitude up to 100,000. When the property being modeled is not related to fluid flow or other processes that might be affected by matrix anisotropy, then this parameter should be set to 1.

file_statistics

General Module Function

The file_statistics module is used to check the format of: *.apdv; *.aidv; *.geo; *.gmf; *.vdf; and *.pgf files, and to calculate and display statistics about the data contained in these files. This module also calculates a frequency distribution of properties in the file. During execution, file_statistics reads the file, displays an error message if the file contains errors in format or numeric values, and then displays the statistical results in the EVS console window.

ModuleInputPorts

1. Read_CSV_data_file (Yellow/Blue/Orange): This port receives data file names.
2. Z_Exag (Grey/Brown): This port receives the Z-Scale factor.

Module Output Ports

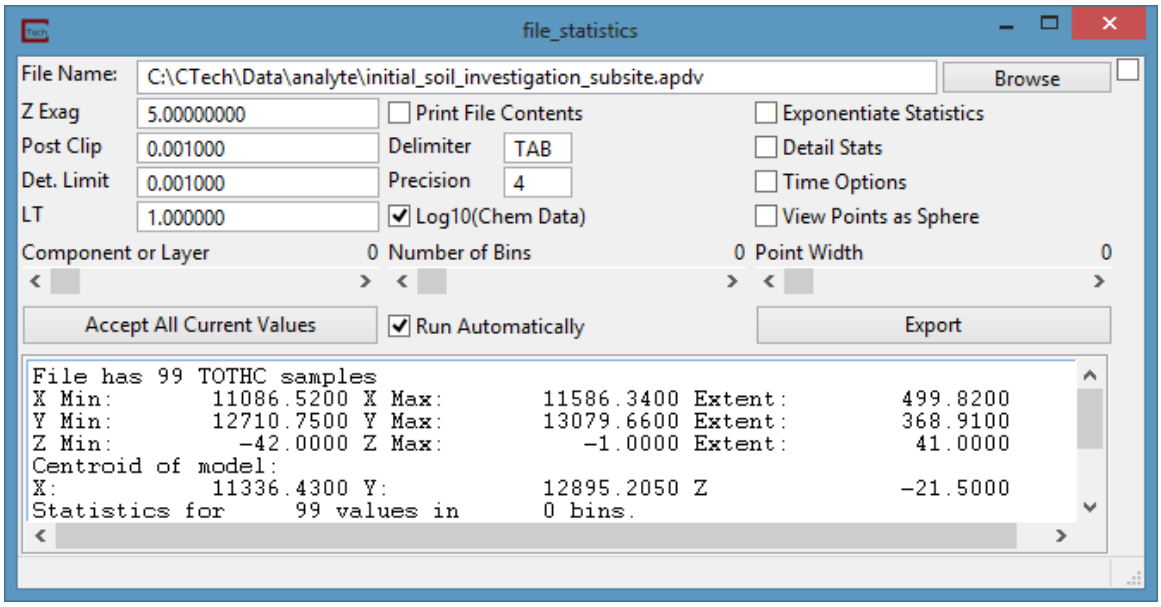
The output ports available from file_statistics depend on whether the Export button has been pushed. Prior to exporting the additional ports, there are four output ports:

1. Read_CSV_data_file (Yellow/Blue/Orange): This port exports data file names.
2. Z_Exag (Grey/Brown): This port exports the Z_Scale factor.
3. out_field (Blue/Black): This port is used to output point data to other modules such as scat_to_tin and scat_to_unif. This provides a means to perform linear or higher order interpolation using methods other than EVS's kriging or IDW algorithms.
4. out_obj (Red): This port outputs a renderable geometry that consists of a cloud of points representing your data locations and values.

After Export, there are a total of 10 ports (including the four above). These are (from left to right):

1. Read_CSV_data_file (Yellow/Blue/Orange): This port exports data file names.
2. Z_Exag (Grey/Brown): This port exports the Z_Scale factor.
3. out_field (Blue/Black): This port is used to output point data to other modules such as scat_to_tin and scat_to_unif. This provides a means to perform linear or higher order interpolation using methods other than EVS's kriging or IDW algorithms.
4. mean
5. median
6. min
7. max
8. number of points
9. The string (characters) representing the total output from Statistics.
10. out_obj (Red): This port outputs a renderable geometry that consists of a cloud of points representing your data locations and values.

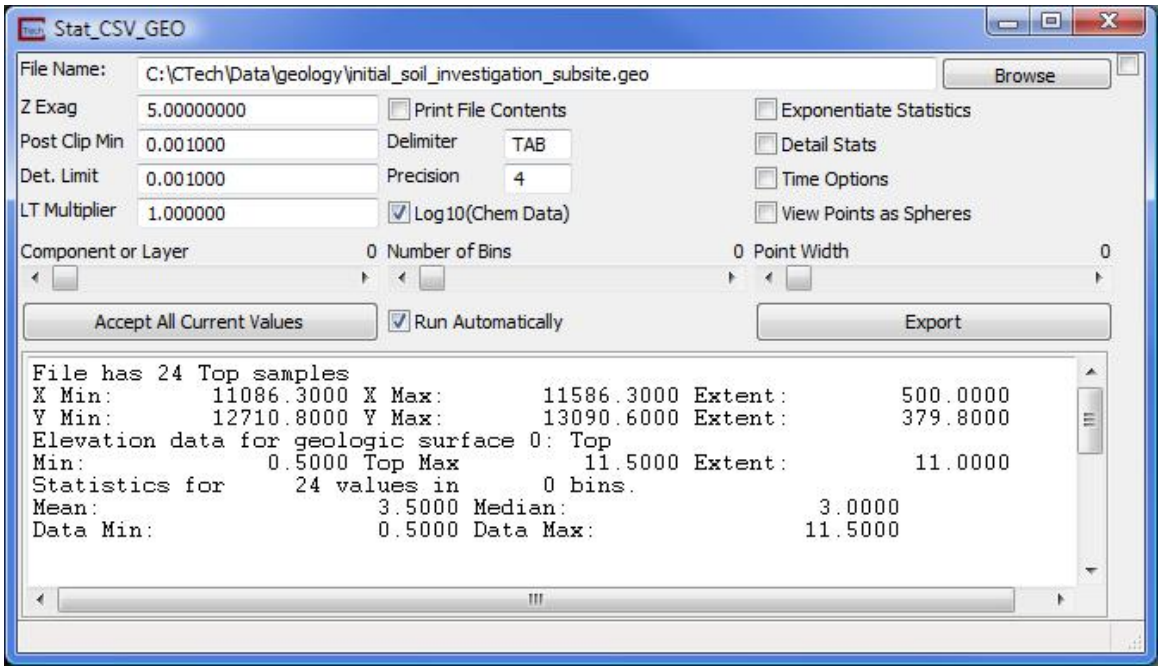
Module Control Panel



1. The control panel of file_statistics is shown in the figure above. The **Browse** button opens a *File Browser* which lists the files that are present in the current directory shown in the directory window. The file may also be typed into the type-in field and will be displayed there if selected in the browser.
2. The ***Z_Exag*** field is the vertical exaggeration of the model. This value can either be set in this field or imported from a different module.
3. The number entered into the ***Post Clip Min*** input field will be used during data processing to replace any sample property value that is less than the specified number. The default value for Post Clip Min is 0.001, but the user can enter any number. This is important if you wish to perform statistics following log10 processing.
4. The ***Det. Limit*** refers to the detection limit used when creating the input file. Any non-detect flag in the file (please see the help file for the file type being read for a list of non-detect flags) will be replaced the Det. Limit value.
5. The ***LT Multiplier*** field is used to input the Less Than Multiplier. This value is used whenever the 'r;<' character preceeds a file value. The file value will be replaced by the product of the file value and the LT Multiplier.
6. The ***Delimiter*** field is a text field that represents how the statistics will be displayed in the modules statistics window, and in the printed file contents. This field can contain any string. Using the string "TAB", the default, causes the text to be tab delimited. This is the only specially handled string. Any other text (e.g. a space or a comma) will be literally used.

7. The ***Print File Contents*** check box allows the user to have the lines of data in the file printed to the console window, along with the results of the statistical calculations.
8. The ***Log10*** toggle (only applicable for chem data files) causes the log of all data values to be taken after applying the "Post Clip Min" criteria. This parameter should be set to be consistent with the data processing that is being performed by the kriging or other modules that are contributing to the display.
9. The ***Detail Stats*** toggle causes the module to display the original statistical information along with more detailed statistics.
10. The ***Exponentiate Statistics*** toggle is used to exponentiate the statistics of log processed data.
11. The ***Time Options*** toggle displays the Time Options panel, which is discussed in the post_samples help.
12. The ***View Points as Spheres*** toggle causes all generated points to be displayed as spheres. When this option is selected the Point Width slider is replaced by a ***Sphere Radius*** field. Entering a value of less than zero in this field will allow the module to compute the radius of the spheres based on the coordinate extents of the module. If a value greater than 1 is placed in this field, that value will be used as the radius of the spheres.
13. The ***Component or Layer*** slider allows the user to select which property component (in a chem file) or geologic layer (in a geology file) will be considered during the current execution of the module. Note that the first component in the file is considered to be component 0 on this slider.
14. The ***Number of Bins*** slider allows the user to specify the bins that will be used to calculate the frequency histogram. The default value of 0 does not calculate a frequency distribution during execution of the module. The user can select any number of bins from 2 to 255 (note that a value of 1 is not valid).
15. The ***Point Width*** slider sets the size of the rendered pixels. The default is 0 which is equivalent to 1.
16. The ***Export*** button causes 6 additional output ports to be visible. These are described above.
17. The ***Accept All Current Values*** button causes the module to re-run.

The output for execution of file_statistics is shown in the figure below.



For this run of the module, no errors were detected in the sample data file initial_soil_investigation_subsite.geo, and both basic statistics and a frequency distribution of 10 bins were calculated and printed. As the Print File Contents box was not checked, the contents of the file were not displayed in the window.

Related Modules

The statistics module performs similar functions on fields of data output from kriging modules.

read_CAD

General Module Function

The read_CAD module will read all versions of AutoCAD DWG and DXF files.

This module provides the user with the capability to integrate site plans, buildings, and other 2D or 3D features into the EVS visualization, to provide a frame of reference for understanding the three dimensional relationships between the site features, and characteristics of geologic, hydrologic, and chemical features. The drawing entities are treated as three dimensional objects, which provides the user with a lot of flexibility in the placement of DXF objects in relation to EVS objects in the visualization. The surfmap and geologic_surfmap modules allow the user to drape DXF line-type entities (not 3D-Faces) onto three dimensional surfaces.

Virtually all AutoCAD object types are supported including points, lines (of all types), 3D surface objects and 3D volumetric objects.

AutoCAD drawings can be drawn in model space (MSPACE) or paper space (PSPACE). Drawings in paper space have a defined viewport which has

coordinates near the origin. When read into EVS or MVS this creates objects which are far from your true model coordinates. For this reason, all drawings for use in our software should be in model space.

Polylines with WIDTH are converted by Read_CAD into triangle strips of the specified width. As you zoom in on polylines with width, the apparent width will change, whereas the apparent width of lines DOES NOT change. However, once they are triangles, they DO NOT define a closed area and therefore would not work with triangulate_polygons.

Module Input Ports

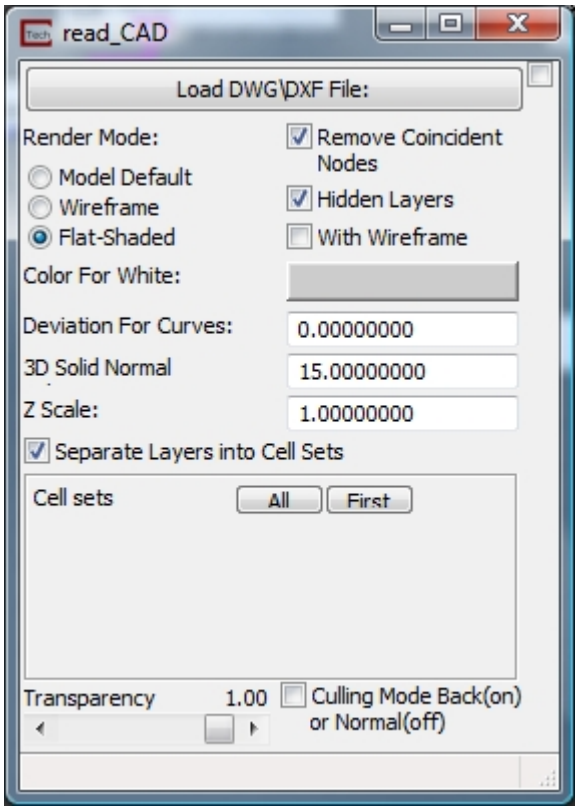
The read_CAD module is shown above. read_CAD has one input port for the Z Scale.

Module Output Ports

The first (left) port outputs the Z Scale.

The second port outputs a typical data field which can be input to surfmap, external edges and/or any of the Subsetting and Processing modules which have the same color port.

The right port outputs a renderable geometry, and can only be connected to the viewer.



Module Control Panel

The parameter input panel for Read DXF is shown above.

Clicking the ***Load DWG\DXF File*** button opens a standard windows style file browser which allows the user to select an AutoCAD DWG or DXF file from a specified directory. The module runs as soon as you read the file.

The Render Mode options allow you to choose how the CAD data will be rendered. The options and their consequences are:

a)*Model Default:*Newer versions of AutoCAD can save the mode by which surface element are diplayed. This will honor that setting

b) *Wireframe* displays the surface objects as a wireframe display

c) *Flat-Shaded*displays surface objects as flat-shaded surface elements

The ***With Wireframe*** toggle outlines surfaces. It is equivalent to setting Line Rendering to "regular"

The ***Remove Coincident Nodes*** toggle causes the reader to process the data, removing coincident nodes. This takes longer to process but creates a smaller more efficient representation of your CAD data.

The***Deviation for Curves***fieldallows the user to specify the accuracy with which analytical curves (arcs, circles) are displayed. The default value of 0.0 employs an expert system algorithm to determine a reasonable value based on the overall size of your CAD model. We recommend that you use this value, especially the first time a file is read.

The ***Z Scale*** field allows the user to scale the Z coordinates. The default scale factor is 1.0, but the user can input any value, by which the Z coordinates will be multiplied. The scale factor must be used to correctly place the entities when Z exaggeration or other scale factors are being applied to the kriged data distributions to produce effective visualizations. In this case, the scale factor should normally be input from the Explode and Scale module, and an equivalent, or nearly equivalent factor should be applied in the Read DXF scale factors (note that the scale factors usually only affect the Z axis scaling).

The ***Separate Layers into Cell Sets*** toggle causes each layer in the CAD file to be output as a separate cell set in EVS.

The ***Cell Sets*** subpanel allows you to control visibility of cell sets in CAD file. When the above toggle is on, each layer may be represented by up to two cell sets, since each cell set can only contain a single type of cell such as points, lines, triangles, etc. All objects are represented as lines or faces in EVS. Points are represented as zero length lines. Volumetric CAD objects are output as the external faces of the object, not as a true 3D volumetric representation.

Cell set visibility can be toggled on-off in this panel.

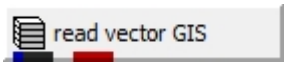
The ***Culling Mode*** toggle controls whether back facing surface are visible. Generally you will want this ON when making the object(s) transparent.

The ***Transparency*** slider determines the opacity of the objects.

Related Modules

-> write_CAD

read_vector_GIS



General Module Function

The read_vector_GIS module reads the following vector file formats: ESRI Shapefile (*.shp); Arc/Info E00 (ASCII) Coverage (*.e00); Atlas BNA file (*.bna); GeoConcept text export (*.gxt); GMT ASCII Vectors (*.gmt); and the MapInfo TAB (*.tab) format.

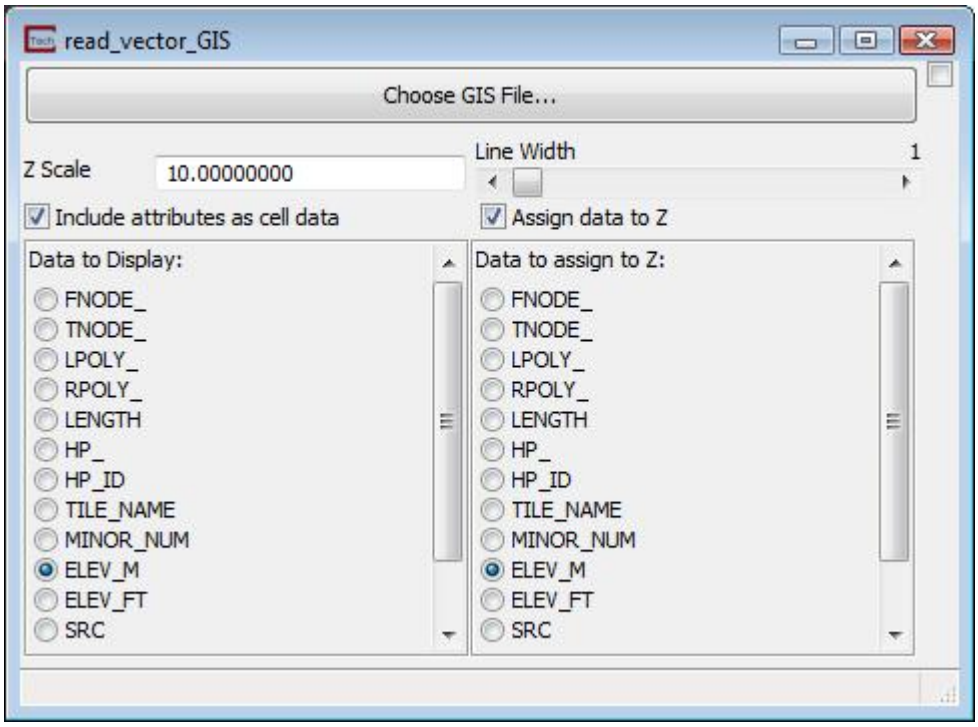
Module Input Ports

No input Ports

Module Output Ports

1. out_field (Blue-Black): This port outputs the field created from reading the file.
2. out_obj (Red): This port outputs a renderable geometry of the output field.

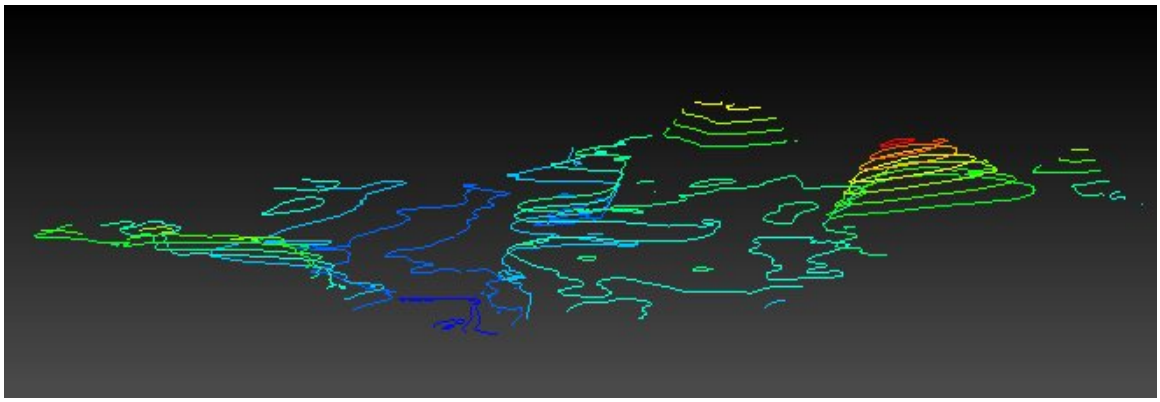
Module Control Panel



The user interface for read_vector_GIS is shown above.

- The **Choose GIS File** button selects the vector file to be read.
- If a line object is read in, the **Line Width** slider will control the width of the renderable line.
- If the **Include attributes as cell data** toggle is turned on, all attribute data will turned into cell data. This includes text data which will be assigned a data value based upon the content of the string.

- The **Data to assign to Z** toggle allows you to specify the attribute (cell data component) to assign to your Z coordinate for 2D shapefiles which represent 3D data. See the figure below as an example of elevation contours.
- The attribute data to be used to color your data can be selected in the Data to Display frame.



buildings

NOTE: This module has increased functionality in EVS-PRO and MVS.

General Module Function

The buildings module reads C Tech's .BLDG file and creates various 3D objects (boxes, cylinders, wedge-shapes for roofs, simple houses etc.), and provides a means for scaling the objects and/or placing the objects at user specified locations. The objects are displayed based on x, y & z coordinates supplied by the user in a .bldg file, with additional scaling option controls on the buildings user interface.

Each object is made up of 3D volumetric elements. This allows for the output of buildings to be cut or sliced to reveal a cross section through the buildings.

Module Input Ports

buildings has two input ports (PRO & MVS)

1) The first (purple port) connects to the Viewers only output port.

2) The second (grey-brown) connects to Explode_and_Scale to inherit the Z_Exaggeration factor.

Module Output Ports

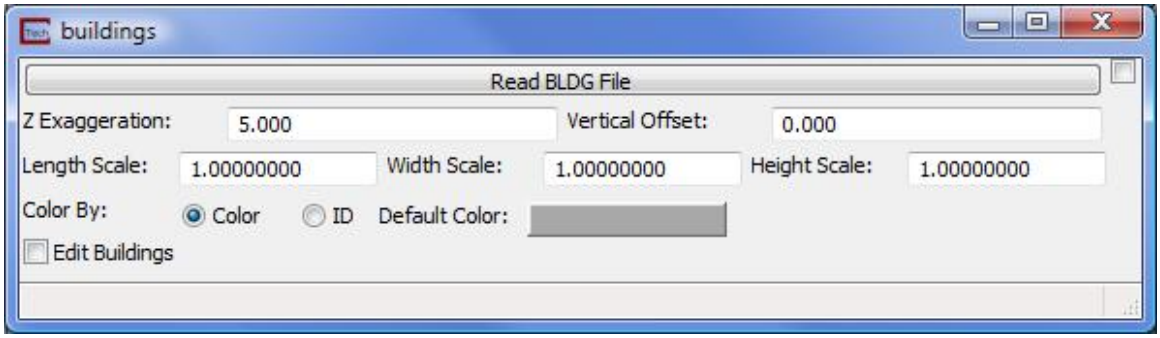
buildings has three output ports.

1) The first outputs the Z_Exaggeration factor.

2) The second port sends a renderable geometry that connects directly to the Viewer.

3) The third outputs the mesh data which can be passed on for other mapping uses such as overlay_aerial or interp_data.

Module Control Panel



The control panel for buildings is shown above. For EVS-PRO and MVS, selecting the "Edit Buildings" toggle will open an additional section which provides the ability to interactively create 3D buildings in your project. This is shown below.

Read .BLDG File: Use to select the BLDG formatted data file. The file sample.bldg provides a number of header lines (denoted with a # character) which describe the various building/object types available, parameters for each column, etc. In general terms this file requires a specification of x, y, z coordinates, length, width, height, angle and building type. Additional optional parameters include color (rgb) and building ID number. Viewing the file sample.bldg in the viewer is a good way to inspect the provided objects and to practice manipulation of the objects.

The **Z Exaggeration** changes the 'Z' placement (starting 'Z' coordinates) for each objects specification in the *.bldg file. Therefore, if an object sits at an elevation of 100 ft. msl., and the user selects a Z Exaggeration of 2.00, the object's base will be moved to 200 ft. msl.

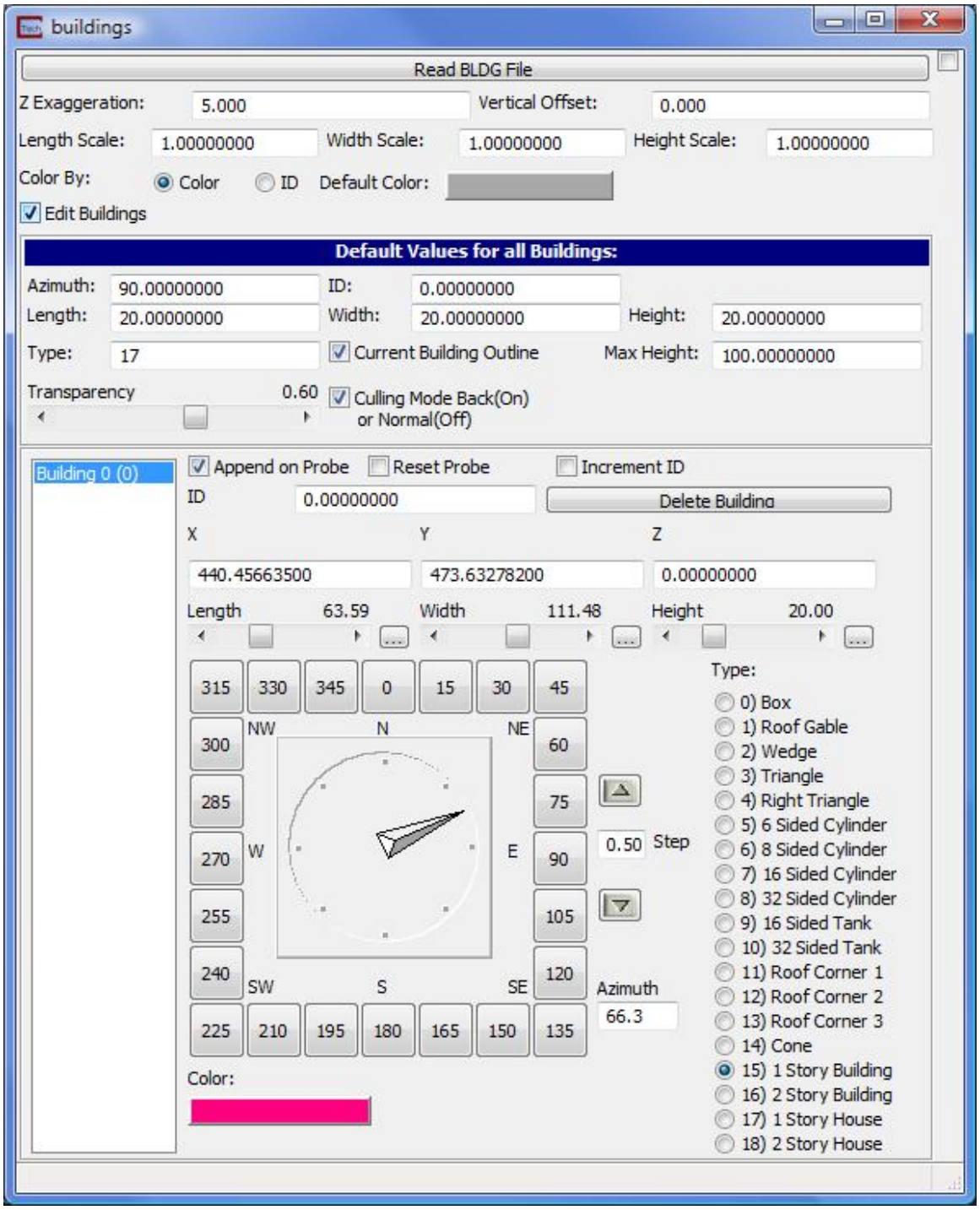
The **Vertical Offset** allows you to move the buildings in the vertical direction. For more elaborate transformations use transform_field.

The **Building Length**, **Building Width** and **Building Height Scale** type-ins change the dimensions of all building objects.

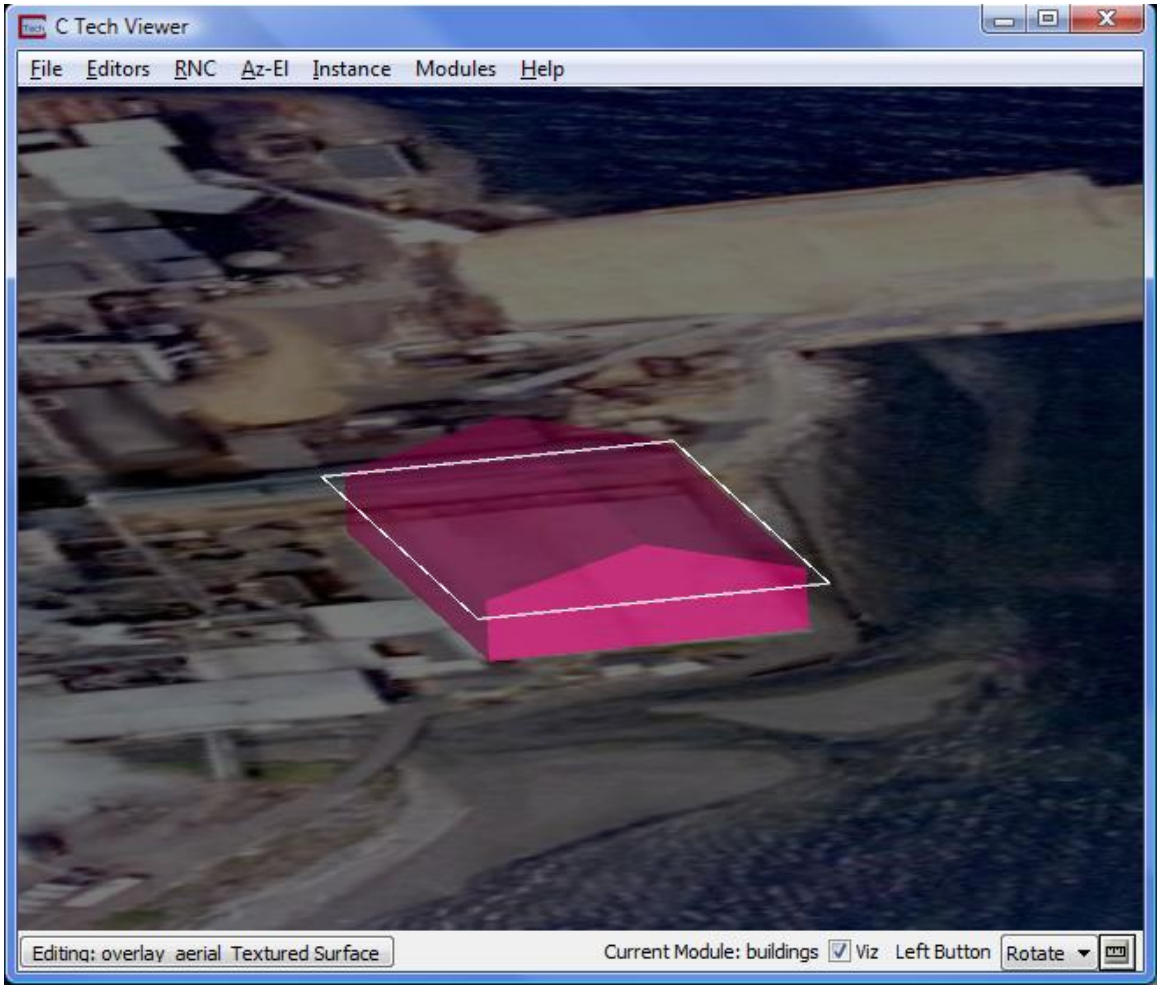
The **Color By** radio buttons allow you to choose whether to color by *Color* (the specified RGB color or the Default Color) or by the ID number in the .BLDG file.

The **Default Color** button and radio buttons allow you to choose the color for any buildings in your file that do not have a specified color.

Selecting the "Edit Buildings" toggle expands the control panel to show:



These new controls work with the probing function in the Viewer to allow you to interactively select the location for new buildings and to set the building type, length, width, height, rotation, and color. These functions can also be used to edit existing buildings that may not fit perfectly with your aerial photo. The figure below shows the Viewer with the building settings above being edited.



Sample Buildings File

Below is an example buildings file. Note that the last 4 columns are optional and contain RGB color values (three numbers from zero to 1.0) and/or a building ID number that can be used for coloring. If only color values are supplied (3 numbers) the ID is automatically determined by the row number. If four numbers are provided it is assumed that the last one is the ID. If only one number is provided it is the ID.

The file below is shown in a table (with dividing lines) for clarity only. The first uncommented line is the number 16 which defines the number of rows of buildings data. The actual file is a simple ASCII file with separators of *space, comma and/or tab*.

```
#  EVS/MVS
#  Copyright (c) 1994-2008 by
#  C Tech Development Corporation
```

```
#  All Rights Reserved
#
# This software comprises unpublished confidential information of
# C Tech Development Corporation and may not be used, copied or
made
# available to anyone, except in accordance with the license
# under which it is furnished.
#
#
# C Tech 3D Building file
# Building 0 is a unit box with base at z=0.0 centered at origin
x,y
# Building 1 is a gabled roof for the unit box
# (to make it a house) with base at z=0.0 centered at origin x,y
# Building 2 is a wedge roof for the unit box
# (to make it a house) with base at z=0.0 centered at origin x,y
# Building 3 is a Equilateral (or Isoseles) Triangular Building 3
side
# Building 4 is a Right Triangular Building 3 side
# Building 5 is a Hexagonal (6 side) cylinder
# Building 6 is a Octagonal (8 side) cylinder
# Building 7 is a 16 side cylinder
# Building 8 is a 32 side cylinder
# Building 9 is a 16 sided horiz. cylindrical tank (Height &
Width equal diameter, Length is along x)
# Building 10 is a 32 sided horiz. cylindrical tank (Height &
Width equal diameter, Length is along x)
# Building 11 is a right angle triangle, height only at right
angle
# Building 12 is a right angle triangle, height at non-right
angle
# Building 13 is a right angle triangle, height at right angle
and 1 non-right angle
# Lines beginning with "#" are comments
# First uncommented line is number of buildings
# X   Y Z LengthWidthHeight Angle Bldg_Type Color and/orID
16
0     0     10    50    50    20    0     0     1
0     100   0     50    50    30    30    0     2
0     100   30    60    50    20    30    1     2
0     200   0     50    50    30    10    0     3
0     200   30    50    50    25    10    2     3
```

200	0	0	50	50	50	0	3	4			
100	100	0	40	40	20	15	4	5			
200	100	0	40	40	30	30	5	6			
200	200	0	50	50	50	0	6	7			
100	200	0	40	60	20	-45	7	8			
100	0	0	50	50	40	0	8	9			
300	0	0	60	20	20	-45	9	0.8	0.6	0.4	10
300	100	0	50	50	30	0	10	0.4	0.6	0.4	11
0	300	0	50	50	50	0	11	1.0	0.4	0.4	12
100	300	0	50	50	50	0	12	0.4	1.0	0.4	13
200	300	0	50	50	50	0	13	0.4	0.4	1.0	14

load_glyph

General Module Function

load_glyph replaces the Glyphs sub-library that was in the tools library. It reads glyphs saved in any of the three primary EVS/MVS field file formats and allows you to modify the shape and orientation of the glyph to allow it to be used in various modules that emply glyphs in slightly different ways. These include glyph, geo_glyph, place_glyph, drive_glyph, drive_glyphs, advector, post_samples, etc. Most modules EXCEPT post_samples will use the glyphs without chaning the default alignment. The supported file formats are:

1) .eff ASCII format, best if you want to be able to open the file in an editor or print it

2) .efz GNU Zip compressed ASCII, same as .eff but in a zip archive

3) .efb binary compressed format, the smallest & fastest format due to its binary form

For a description of the .EFF file formats click here.

The objects saved in the .efx files should be simple geometric objects ideally designed to fit in a unit box centered at the origin (0,0,0). For optimal performance the objects should not include nodal or cell data. You may create your own objects or use any of the ones that C Tech supplies in the ctech\data\glyphs folder.

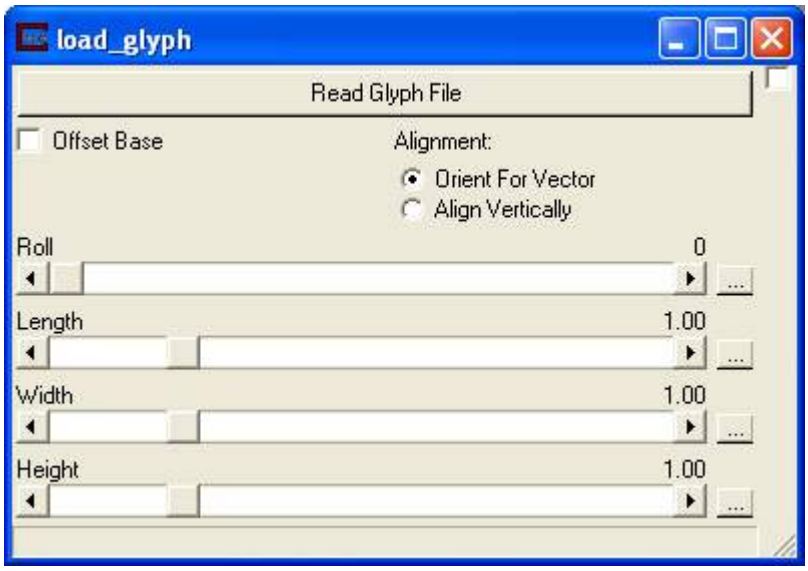
Module Input Ports

The load_glyph module is shown above. It has no input ports. It obtains the EVS Field input data by reading a file with a file browser.

Module Output Ports

load_glyph has two output ports. The first port (closest to the left) passes the mesh and data components to other modules which accept Field data

types. The second port is used to send data directly to the viewer for rendering.



Module Control Panel

The user interface for load_glyph consists of:

1) The *Read Glyph File* buttton which opens a file browser to select the glyph file. The file browser is used for selecting the file with the following suffixes *.eff;*.efz; or *.efb. Double clicking on a filename or selecting a file name and choosing OK reads the selected file into memory and closes the file browser. After the file has been read into memory, the mesh and is modified by the other parameters in the module and then passed to any downstream modules in the application. Selecting a different file, after one has been read in, will replace the first file's mesh a with the new file's mesh and all modules will be updated with the new glyph.

2) *Offset Base* is a toggle that changes the origin from the centroid of the glyph to the base. This is useful for modules like advector or glyph if you want the glyph to be centered about its base vs. its centroid.

3) *Roll* is a slider that lets you control the rotation about the roll axis (glyph directional axis)

4) *Length* is a slider that lets you scale the length of the glyph. The default scale value and default length are 1.0.

5) *Width* is a slider that lets you scale the widthof the glyph. The default scale value and default length are 1.0.

6) *Height* is a slider that lets you scale the height of the glyph. The default scale value and default length are 1.0.

Related Modules

Load_EVS_Field

Load_EVS_Field

General Module Function

Load_EVS_Field supplants the need for Read_UCD, Read_netCDF and Read_Field by incorporating all of their functionality and more in a single module. It reads a dataset from any of six different EVS/MVS compatible file formats into an EVS field, including the new EVS Field Formats:

1) .eff ASCII format, best if you want to be able to open the file in an editor or print it

2) .efz GNU Zip compressed ASCII, same as .eff but in a zip archive

3) .efb binary compressed format, the smallest & fastest format due to its binary form

The EVS Field Formats *.eff;*.efz;and *.efb support **all** types of EVS field output including:

1) Uniform fields

2) Geology (from Krig_3D_Geology and Spline_Geology)

3) Structured fields (such as irregular fields read in from Read_Field)

4) Unstructured Cell Data (UCD format) general grids with nodal and/or cell data

5) Special fields containing spheres (which are points with radii)

6) Special fields containing color data (such as from Read_DXF)

Note: Because the .eff, .efz and .efb formats better handle all types of EVS/MVS output, these three formats are recommended for use over UCD, netCDF or Field.

For a description of the .EFF file formats click here.

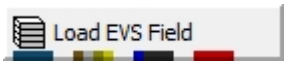
Module Input Ports

The Load_EVS_Field module is shown above. It has no input ports. It obtains the EVS Field input data by reading a file with a file browser.

Module Output Ports

Load_EVS_Field has three or four output ports.

If the field file being read was the output of modules like Krig_3D_Geology or Spline_Geology, there will be four output ports and the module will look like:

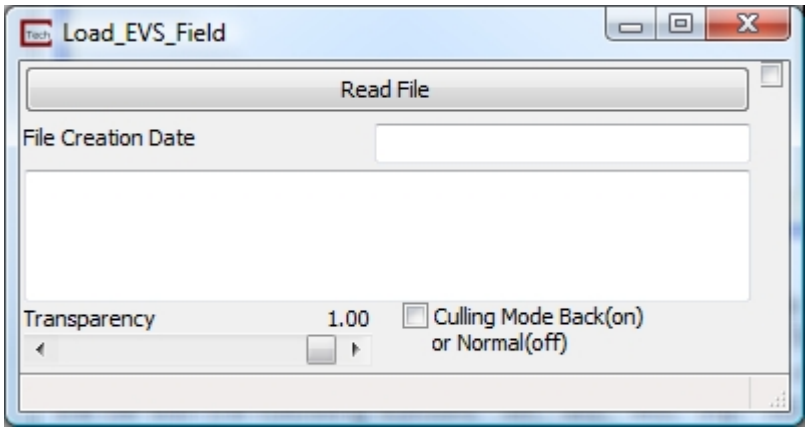


1. The first port (closest to the left) passes a string representing any notes that were found in the file. These notes exist only in field file and are not necessary for the module to run.
2. The second port (Brown-Grey-Light Brown-Beige) outputs geology material info similar to the same color ports on Krig_3D_Geology or Spline_Geology.

3. The third port outputs the mesh and data components to other modules which accept Field data types.
4. The fourth (red) port is used to send data directly to the viewer for rendering.

Otherwise the ports will be:

1. The first port (closest to the left) passes a string representing any notes that were found in the file. These notes exist only in field file and are not necessary for the module to run.
2. The second port outputs the mesh and data components to other modules which accept Field data types.
3. The third port is used to send data directly to the viewer for rendering.



Module Control Panel

The user interface for Load_EVS_Field consists of the file browser shown above, a window showing the file creation date, and any notes that were in the file. The notes are only in EVS Field files (*.eff;*.efz;*.efb) and are not necessary for the module to run. The file browser is used for selecting the file with the following suffixes *.eff;*.efz;*.efb;*.inp;*.fld; or *.cdf. Double clicking on a filename or selecting a file name and choosing OK reads the selected file into memory and closes the file browser. After the file has been read into memory, the mesh and data components are passed to any downstream modules in the application. Selecting a different file, after one has been read in, will replace the first file's mesh and data components with the new file's mesh and data components and all modules will be updated with the new data.

The ***Culling Mode*** toggle controls whether back facing surface are visible. Generally you will want this ON when making the object(s) transparent.

The ***Transparency*** slider determines the opacity of the objects.

Related Modules

Read_netCDF

Read_UCD

Read_Field

EVS Field File Formats and Examples

EVS Field file formats supplant the need for UCD, netCDF, Field (.fld), EVS_Geology by incorporating all of their functionality and more in a new file format with three mode options.

1) .eff ASCII format, best if you want to be able to open the file in an editor or print it
2) .efz GNU Zip compressed ASCII, same as .eff but in a zip archive
3) .efb binary compressed format, the smallest & fastest format due to its binary form

Here are the tags available in an EVS field file, in the appropriate order. Note that **no** file will contain ALL these tags, as some are specific to the type of field (based on definition). The binary file format is undocumented and exclusively used by C Tech's Save_EVS_Field module.

If the file is written compressed, the .efz file (and any split, extra data files) will all be compressed. The compression algorithm is compatible with the free gzip/gunzip programs or WinZip, so the user can uncompress a .efz file and get an .eff file at will. The .efb file is also compressed (hence its very small size), but uncompressing this file will not make it human-readable.

EVS Field Files

EVS Field Files consist of file tags that delineate the various sections of the file(s) and data (coordinates, nodal and/or cell data, and connectivity). The file tags are discussed below followed by portions of a few example files.

FILE TAGS:

The file tags for the ASCII file formats (shown in ***Bold Italics***) are discussed below with a representative example. They are given in the appropriate order. If you need assistance creating software to write these file formats, please contact support@ctech.com.

DATE_CREATED(optional) 7/16/2004 1:57:55 PM

The creation date of the file.

EVS_FIELD_FILE_NOTES_START (optional)

Insert your Field file notes here.

EVS_FIELD_FILE_NOTES_END

This is the file description block. These notes are used to describe the contents of the Field file. The entire block is optional, however if you wish to use notes then both the starting and end tag are required.

DEFINITION Mesh+Node_Data

This is the type of field we are creating. Typically options are:

1) Mesh+Node_Data
2) Mesh+Cell_Data
3) Mesh+Node_Data+Cell_Data
4) Mesh_Struct+Node_Data (Geology)
5) Mesh_Unif+Node_Data (Uniform field)

NSPACE 3

nspace of the output field. Typically 3, but 2 in the case of geology or an image

NNODES 66355

Number of nodes. Not used for Mesh_Struct of Mesh_Unif

NDIM 2

Number of dimensions in a Mesh_Struct or Mesh_Unif

DIMS 41 41

The dimensions for a mesh_struct or uniform field

POINTS 11061.528999 12692.304504 -44.049999 11611.330994 13098.105469 11.500000

The lower left and upper right corner of a uniform field (Mesh_Unif only)

COORD_UNITS "ft"

Coordinate Units

NUM_NODE_DATA 7

Number of nodal data components

NUM_CELL_DATA 1

Number of cell data components

NCELL_SETS 5

Number of cell sets

NODES FILE "test_split.xyz" ROW 1 X 1 Y 2 Z 3

Nodes section is starting. If it says "NODES IN_FILE", the nodes follow (x/y/z) on the next nnodes rows, otherwise, the line will say FILE "filename" ROW 1 X 1 Y 2 Z 3, which is the file to get

the coordinates, the row to start at (1 is first line of file), and the columns containing your X, Y, and Z values

NODE_DATA_DEF 0 "TOTHC" "log_ppm" MINMAX -3 4.592 FILE "test_split.nd" ROW 1 COLS 1

NODE_DATA_DEF specifies the definition of a nodal data component. The second word is the data component number, the third is the name, the 4th is the units, then it will either say IN_FILE (which means that it will start after a NODE_DATA_START tag) or the file information. Other options are:

1) MINMAX - two numbers follow which are the data minimum and maximum. This behaves much like the set_min_max module.

2) If this is vector data, there will be a VECLEN 3 tag in there, and COLS will need to have 3 numbers following it (for each component of the vector)

3) NODE_DATA_START. All the node data components that are specified IN_FILE are listed in order after this tag.

CELL_SET_DEF 0 8120 Hex "Fill" MINMAX 1 14 FILE "test_split.conn" ROW 1

Definition of a cell set. 2nd word is cell set number, 3rd is number of cells, 4th is type, 5th is the name, then its either IN_FILE (which means they will be listed in order by cell set), or the FILE "filename" section and a row to begin reading from. Other options are:

1) MINMAX - two numbers follow which are the data minimum and maximum. This behaves much like the cell_set_min_max module.

2) CELL_START. Start of all the cell set definitions that are specified IN_FILE.

CELL_DATA_DEF 0 "Indicator" "Discreet Unit" FILE "test_split.cd" ROW 1 COLS 1

Definition of cell data. Same options as NODE_DATA_DEF

CELL_DATA_START

Start of all cell data that is specified as IN_FILE

LAYER_NAMES "Top" "Fill" "Silt" "Clay" "Gravel" "Sand"

Allows you to specify the names associated with surfaces (layers)

MATERIAL_MAPPING "1|Silt" "2|Fill" "3|Clay" "4|Sand" "5|Gravel"

Allows you to specify the Material_ID and the associated material names. Note that each number/name pair is in quotes, with the name separated from the number by the pipe "|" symbol.

END

Marks the end of the data section of the file. (Allows us to put a password on .eff files)

EVS Field File Examples:

Because EVS Field Files can contain so many different types of grids, it is beyond the scope of our help system to include every variant.

Krig_3D - EFF file representing a uniform field: The file below is an abbreviated example of writing the output of Krig_3D having kriged a uniform field (which can be volume rendered). Large sections of the data regions of this file are omitted to save space. This is represented by sections of the file with "*** omitted ***" replacing many lines of data.

```
DEFINITION Mesh_Unif+Node_Data
NSPACE 3
NDIM 3
DIMS 41 41 35
COORD_UNITS "ft"
NUM_NODE_DATA 7
POINTS 11281.910004 12211.149994 -29.900000 12515.890015
13259.449951 0.900000
NODE_DATA_DEF 0 "VOC" "log_ppm" IN_FILE
NODE_DATA_DEF 1 "Confidence-VOC" "linear_%" IN_FILE
NODE_DATA_DEF 2 "Uncertainty-VOC" "linear_Unc" IN_FILE
NODE_DATA_DEF 3 "Geo_Layer" "linear_" IN_FILE
NODE_DATA_DEF 4 "Elevation" "linear_ft" IN_FILE
NODE_DATA_DEF 5 "Layer Thickness" "linear_ft" IN_FILE
NODE_DATA_DEF 6 "Material_ID" "linear_" IN_FILE
NODE_DATA_START
-2.357487 34.455845 2.325005 0.000000 -29.900000 30.799999
0.000000
-3.000000 34.977974 0.000000 0.000000 -29.900000 30.799999
0.000000
-3.000000 35.603794 0.000000 0.000000 -29.900000 30.799999
0.000000
***** OMITTED *****
```

```
-3.000000 30.056839 0.000000 0.000000 0.900000 30.799999
0.000000
-3.000000 29.858747 0.000000 0.000000 0.900000 30.799999
0.000000
-3.000000 29.673925 0.000000 0.000000 0.900000 30.799999
0.000000
END
```

Krig_3D - EFF Split file representing a uniform field: The file below is a complete example of writing the output of Krig_3D having kriged a uniform field (which can be volume rendered). Note that the .EFF file is quite small, but references the data in a separate file named krig_3d_uniform_split.nd.

```
DEFINITION Mesh_Unif+Node_Data
NSPACE 3
NDIM 3
DIMS 41 41 35
COORD_UNITS "ft"
NUM_NODE_DATA 7
POINTS 11281.910004 12211.149994 -29.900000 12515.890015
13259.449951 0.900000
NODE_DATA_DEF 0 "VOC" "log_ppm" FILE
"krig_3d_uniform_split.nd" ROW 1 COLS 1
NODE_DATA_DEF 1 "Confidence-VOC" "linear_%" FILE
"krig_3d_uniform_split.nd" ROW 1 COLS 2
NODE_DATA_DEF 2 "Uncertainty-VOC" "linear_Unc" FILE
"krig_3d_uniform_split.nd" ROW 1 COLS 3
NODE_DATA_DEF 3 "Geo_Layer" "linear_" FILE
"krig_3d_uniform_split.nd" ROW 1 COLS 4
NODE_DATA_DEF 4 "Elevation" "linear_ft" FILE
"krig_3d_uniform_split.nd" ROW 1 COLS 5
NODE_DATA_DEF 5 "Layer Thickness" "linear_ft" FILE
"krig_3d_uniform_split.nd" ROW 1 COLS 6
NODE_DATA_DEF 6 "Material_ID" "linear_" FILE
"krig_3d_uniform_split.nd" ROW 1 COLS 7
END
```

Large sections of the data regions of the data file krig_3d_uniform_split.nd are omitted below to save space. This is represented by sections of the file with "*** omitted ***" replacing many lines of data.

```
-2.357487 34.455845 2.325005 0.000000 -29.900000 30.799999
0.000000
-3.000000 34.977974 0.000000 0.000000 -29.900000 30.799999
0.000000
```

```
-3.000000 35.603794 0.000000 0.000000 -29.900000 30.799999
0.000000
***** OMITTED *****
-3.000000 30.056839 0.000000 0.000000 0.900000 30.799999
0.000000
-3.000000 29.858747 0.000000 0.000000 0.900000 30.799999
0.000000
-3.000000 29.673925 0.000000 0.000000 0.900000 30.799999
0.000000
```

Krig_3D_Geology & Krig_3D - EFF file representing multiple geologic layers with analyte (e.g. chemistry): The file below is an abbreviated example of writing the output of Krig_3D having kriged analyte (e.g. chemistry) data with geology input. Large sections of the data regions of this file are omitted to save space. This is represented by sections of the file with "*** omitted ***" replacing many lines of data.

```
NSPACE 3
NNODES 66355
COORD_UNITS "ft"
NUM_NODE_DATA 7
NCELL_SETS 5
NODES IN_FILE
11153.998856 12722.725708 2.970446
11161.871033 12715.198792 2.783408
11169.743210 12707.671875 2.594242
***** OMITTED *****
11250.848221 12865.266907 -42.575920
11248.750000 12870.909973 -42.000000
11243.389938 12870.020935 -42.474934
NODE_DATA_DEF 0 "TOTHC" "log_mg/kg" IN_FILE
NODE_DATA_DEF 1 "Confidence-TOTHC" "linear_%" IN_FILE
NODE_DATA_DEF 2 "Uncertainty-TOTHC" "linear_Unc" IN_FILE
NODE_DATA_DEF 3 "Geo_Layer" "Linear_" IN_FILE
NODE_DATA_DEF 4 "Elevation" "Linear_ft" IN_FILE
NODE_DATA_DEF 5 "Layer Thickness" "Linear_ft" IN_FILE
NODE_DATA_DEF 6 "Material_ID" "Linear_" IN_FILE
NODE_DATA_START
-0.777059 27.239126 15.861248 0.000000 2.970446 8.270601
2.000000
-0.661227 27.349216 16.503609 0.000000 2.783408 8.270658
2.000000
-0.288564 27.512394 18.822187 0.000000 2.594242 8.261375
2.000000
***** OMITTED *****
```

```
2.886921 69.551514 1.128253 4.000000 -42.575920 13.628321
4.000000
3.113943 99.999977 0.000000 4.000000 -42.000000 13.654032
4.000000
3.070153 72.869553 0.841437 4.000000 -42.474934 13.646055
4.000000
CELL_SET_DEF 0 8120 Hex "Fill" IN_FILE
CELL_SET_DEF 1 14680 Hex "Silt" IN_FILE
CELL_SET_DEF 2 6502 Hex "Clay" IN_FILE
CELL_SET_DEF 3 11284 Hex "Gravel" IN_FILE
CELL_SET_DEF 4 14412 Hex "Sand" IN_FILE
CELL_START
0 1 42 41 1681 1682 1723 1722
1 2 43 42 1682 1683 1724 1723
2 3 44 43 1683 1684 1725 1724
***** OMITTED *****
54462 54503 66349 66348 56143 56184 66353 66352
54503 54502 66350 66349 56184 56183 66354 66353
54502 54461 66347 66350 56183 56142 66351 66354
END
```

Post_samples - EFF file representing spheres: The file below is a complete example of writing the output of post_samples' blue-black field port having read the file initial_soil_investigation_subsite.apdv. This data file has 99 samples with data that was log processed. If this file is read by Load_EVS_Field. It creates all 99 spheres colored and sized as they were in Post_samples. The tubes and any labeling are not included in the field port from which this file was created.

```
DEFINITION Mesh+Node_Data
NSPACE 3
NNODES 99
COORD_UNITS "units"
NUM_NODE_DATA 2
NCELL_SETS 1
NODES IN_FILE
11566.340027 12850.590027 -10.000000
11566.340027 12850.590027 -70.000000
11566.340027 12850.590027 -160.000000
11586.340027 13050.589966 -10.000000
11586.340027 13050.589966 -70.000000
11586.340027 13050.589966 -160.000000
11381.700012 12747.500000 -15.000000
11381.700012 12747.500000 -25.000000
```

```
11414.399994 12781.099976 -15.000000
11414.399994 12781.099976 -25.000000
11338.000000 12830.799988 -10.000000
11338.000000 12830.799988 -65.000000
11338.000000 12830.799988 -115.000000
11338.000000 12830.799988 -165.000000
11410.290009 12724.690002 -5.000000
11410.290009 12724.690002 -35.000000
11410.290009 12724.690002 -45.000000
11410.290009 12724.690002 -125.000000
11410.290009 12724.690002 -175.000000
11427.000000 12780.900024 -10.000000
11427.000000 12780.900024 -30.000000
11427.000000 12780.900024 -80.000000
11416.899994 12819.450012 -10.000000
11416.899994 12819.450012 -30.000000
11416.899994 12819.450012 -70.000000
11416.899994 12819.450012 -95.000000
11416.899994 12819.450012 -105.000000
11416.899994 12819.450012 -120.000000
11416.899994 12819.450012 -140.000000
11401.730011 12897.770020 -10.000000
11401.730011 12897.770020 -30.000000
11401.730011 12897.770020 -80.000000
11401.730011 12897.770020 -110.000000
11401.730011 12897.770020 -145.000000
11401.730011 12897.770020 -180.000000
11259.670013 12819.289978 -10.000000
11259.670013 12819.289978 -40.000000
11259.670013 12819.289978 -70.000000
11259.670013 12819.289978 -95.000000
11259.670013 12819.289978 -140.000000
11340.489990 12892.609985 -30.000000
11340.489990 12892.609985 -55.000000
11340.489990 12892.609985 -80.000000
11340.489990 12892.609985 -110.000000
11340.489990 12892.609985 -130.000000
11340.489990 12892.609985 -165.000000
11248.750000 12870.909973 -10.000000
11248.750000 12870.909973 -35.000000
11248.750000 12870.909973 -45.000000
11248.750000 12870.909973 -85.000000
```

```
11248.750000 12870.909973 -110.000000
11248.750000 12870.909973 -160.000000
11248.750000 12870.909973 -210.000000
11086.519997 12830.669983 -15.000000
11086.519997 12830.669983 -30.000000
11086.519997 12830.669983 -80.000000
11086.519997 12830.669983 -130.000000
11211.869995 12710.750000 -30.000000
11211.869995 12710.750000 -80.000000
11211.869995 12710.750000 -135.000000
11199.039993 12810.159973 -20.000000
11199.039993 12810.159973 -40.000000
11199.039993 12810.159973 -85.000000
11199.039993 12810.159973 -150.000000
11298.000000 12808.630005 -60.000000
11496.339996 12753.590027 -10.000000
11496.339996 12753.590027 -30.000000
11496.339996 12753.590027 -80.000000
11496.339996 12753.590027 -110.000000
11496.339996 12753.590027 -150.000000
11309.029999 12948.989990 -10.000000
11309.029999 12948.989990 -35.000000
11309.029999 12948.989990 -95.000000
11309.029999 12948.989990 -125.000000
11309.029999 12948.989990 -130.000000
11209.350006 12993.940002 -5.000000
11209.350006 12993.940002 -35.000000
11209.350006 12993.940002 -60.000000
11209.350006 12993.940002 -95.000000
11209.350006 12993.940002 -125.000000
11301.970001 13079.660034 -20.000000
11301.970001 13079.660034 -30.000000
11301.970001 13079.660034 -85.000000
11301.970001 13079.660034 -125.000000
11286.769989 13026.699951 -30.000000
11286.769989 13026.699951 -45.000000
11286.769989 13026.699951 -75.000000
11286.769989 13026.699951 -120.000000
11393.470001 12948.900024 -20.000000
11393.470001 12948.900024 -45.000000
11393.470001 12948.900024 -95.000000
11393.470001 12948.900024 -110.000000
```

```
11393.470001 12948.900024 -130.000000
11393.470001 12948.900024 -170.000000
11251.300003 12929.270020 -10.000000
11251.300003 12929.270020 -30.000000
11251.300003 12929.270020 -80.000000
11251.300003 12929.270020 -120.000000
11251.300003 12929.270020 -145.000000
NODE_DATA_DEF 0 "TOTHC" "log_mg/kg" IN_FILE
NODE_DATA_DEF 1 "" "" ID 668 IN_FILE
NODE_DATA_START
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
1.322219 4.998203
2.806180 4.998203
1.602060 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
1.845098 4.998203
2.278754 4.998203
-3.000000 4.998203
1.296665 4.998203
-3.000000 4.998203
1.278754 4.998203
3.716003 4.998203
1.623249 4.998203
1.505150 4.998203
-3.000000 4.998203
1.707570 4.998203
-3.000000 4.998203
3.770852 4.998203
```

```
3.869232 4.998203
1.113943 4.998203
-3.000000 4.998203
2.025306 4.998203
3.434569 4.998203
3.594039 4.998203
2.454845 4.998203
-3.000000 4.998203
2.740363 4.998203
2.079181 4.998203
3.806180 4.998203
4.908485 4.998203
2.176091 4.998203
-3.000000 4.998203
3.792392 4.998203
3.362897 4.998203
4.255272 4.998203
3.699387 4.998203
3.518514 4.998203
3.301030 4.998203
3.113943 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
1.361728 4.998203
-3.000000 4.998203
-3.000000 4.998203
2.000000 4.998203
1.643453 4.998203
1.732394 4.998203
1.643453 4.998203
3.556303 4.998203
-0.522879 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
3.079181 4.998203
-3.000000 4.998203
2.633468 4.998203
1.505150 4.998203
```

```
-3.000000 4.998203
-3.000000 4.998203
-0.920819 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-0.886057 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-3.000000 4.998203
-0.096910 4.998203
-3.000000 4.998203
4.000000 4.998203
2.000000 4.998203
1.602060 4.998203
1.000000 4.998203
-0.301030 4.998203
-3.000000 4.998203
1.785330 4.998203
-3.000000 4.998203
0.431364 4.998203
4.518514 4.998203
-3.000000 4.998203
CELL_SET_DEF 0 99 Point "" IN_FILE
CELL_START
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

END

read_lines

General Module Function

The read_lines module is used to visualize a series of points with data connected by lines. read_lines accepts three different file formats, with the APDV file format the lines are connected by boring ID, with the ELF (EVS Line File) format each line is made by defining the points that make up the line, and with the SAD (Strike and Dip) file format, there is a choice to connect each sample by ID or by Data Value.

Module Input Ports

read_lines has one input port that receives the Z-Scale factor

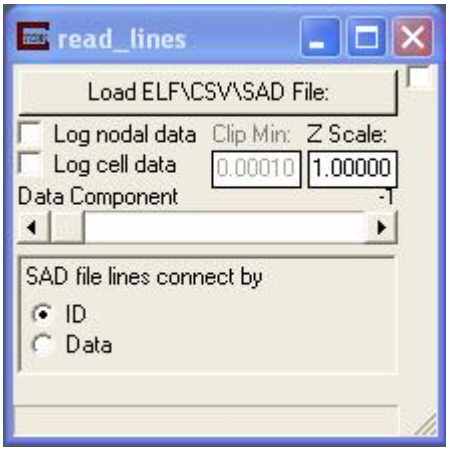
Module Output Ports

read_lines has three output ports:

The leftmost grey-beige port outputs the Z-Scale factor

The center (blue-black) port outputs a field comprised of lines with all defined data components.

The rightmost (red) port outputs a geometry that can be input to the Viewer for rendering.



Module Control Panel

The control panel of read_lines is shown in the figure above. The Load ELF\APDV\SAD File button opens a *File Browser* which lists the *.elf, *.sad, and *.apdv files that are present in the current directory shown in the directory window. The format of each of these files is described in the help section.

Note that this module will not begin running until a valid file has been selected.

Log nodal data – This toggle sets the nodal data values to the log value of the input node data.

Log cell data – This toggle sets the cell data values to the log value of the input cell data.

Clip Min – This is only active if either Log processing for nodal or cell data is selected and sets the minimum value for each data value so that they can be correctly processed.

Data Component – This slider controls which data value to color the lines by.

SAD files connect by ID – If a *.sad file has been read the lines will be connected by ID.

SAD files connect by Data – If a *.sad file has been read the lines will be connected by the data component.

EVS Line File Example

Discussion of EVS Line Files

EVS line files contain horizontal and vertical coordinates, which describe the 3-D locations and values of properties of a system. Line files must be in ASCII format and can be delimited by commas, spaces, or tabs. They must have an .elf suffix to be selected in the file browsers of EVS modules. Each line of the EVS line file contain the coordinate data for one sampling location and up to 300 (columns of) property values. There are no computational restrictions on the number of lines that can be included in a file.

EVS Line Files

EVS **Line** Files consist of file tags that delineate the various sections of the file(s) and data (coordinates, nodal and/or cell data). The file tags are discussed below followed by portions of a few example files.

FILE TAGS:

The file tags for the ASCII file formats (shown in ***Bold Italics***) are discussed below with a representative example. They are given in the appropriate order. If you need assistance creating software to write these file formats, please contact support@ctech.com.

COORD_UNITS "ft"

Defines the coordinate units for the file. These should be consistent in X, Y, and Z.

NUM__DATA 7 1

Number of nodal data components followed by the number of cell data components.

NODE_DATA_DEF 0 "TOTHC" "log_ppm"

NODE_DATA_DEF specifies the definition of a nodal data component. The second value is the data component number, the third is the name, and the 4th is the units.

CELL_DATA_DEF 0 "Indicator" "Discreet Unit"

Definition of cell data. Same options as NODE_DATA_DEF

LINE 12 1

Beginning of a line segment is followed on the same line by the cell data values.

Following this line should be the points making up the line in the following format:

X, Y, Z coordinates followed by nodal data values.

64718.310547 37500.000000 -1250.000000 1 -1250.000000

63447.014587 35101.682129 -2000.000000 2 -2000.000000

CLOSED

This flag is used at the end of a line definition to indicate the end of the line should be connected to the beginning of the line.

END

Marks the end of the data section of the file. (Allows us to put a password on .eff files)

EXAMPLE FILE

NUM_DATA 2 0
NODE_DATA_DEF 0 "Node_Number" "Linear_ID"
NODE_DATA_DEF 1 "Distance" "Linear_ft"
LINE
1900297.026154 677367.319824 72.000000 0.000000 0.000000
1900314.256775 677438.611328 72.000000 1.000000 73.344208
1900314.687561 677442.703522 72.000000 2.000000 77.459015
1900316.410645 677447.011261 72.000000 3.000000 82.098587
1900319.641266 677447.442017 72.000000 4.000000 85.357796
1900345.487030 677441.411530 72.000000 5.000000 111.897774
1900360.563782 677439.472870 72.000000 6.000000 127.098656
1900363.579193 677447.226807 72.000000 7.000000 135.418289
1900365.517822 677447.226807 72.000000 8.000000 137.356918
1900365.948608 677438.396118 72.000000 9.000000 146.198105
1900379.733032 677436.888245 72.000000 10.000000 160.064758
1900405.578766 677432.150055 72.000000 11.000000 186.341217
1900497.331879 677416.427002 72.000000 12.000000 279.431763
1900511.331512 677414.919464 72.000000 13.000000 293.512329

1900525.762268 677411.257721 72.000000 14.000000 308.400421
1900527.269775 677405.442444 72.000000 15.000000 314.407898
1900524.900696 677399.411926 72.000000 16.000000 320.887085
1900522.531311 677391.012024 72.000000 17.000000 329.614746
1900517.362366 677357.196808 72.000000 18.000000 363.822754
1900501.854828 677266.951569 72.000000 19.000000 455.390686
1900501.639282 677262.213379 72.000000 20.000000 460.133789
1900500.777710 677255.321014 72.000000 21.000000 467.079773
1900496.470306 677250.151733 72.000000 22.000000 473.808472
1900487.208862 677241.751816 72.000000 23.000000 486.311798
1900450.378204 677201.906097 72.000000 24.000000 540.572083
1900403.568481 677152.368134 72.000000 25.000000 608.727478
1900356.758759 677102.830177 72.000000 26.000000 676.882874
1900309.949036 677053.292221 72.000000 27.000000 745.038269
1900286.257172 677028.523243 72.000000 28.000000 779.313721
1900278.718445 677022.923517 72.000000 29.000000 788.704651
1900269.672546 677024.431061 72.000000 30.000000 797.875305
1900217.334717 677035.200397 72.000000 31.000000 851.309631
1900232.196075 677097.230453 72.000000 32.000000 915.095154
1900247.057434 677159.260513 72.000000 33.000000 978.880615
1900252.226715 677179.937317 72.000000 34.000000 1000.193787
1900267.159851 677242.326401 72.000000 35.000000 1064.345215
1900282.093018 677304.715485 72.000000 36.000000 1128.496460
1900297.026154 677367.104584 72.000000 37.000000 1192.647827
END

well_decommission

(Available only in MVS)

General Module Function

Groundwater contamination sites worldwide are engaged in regular sampling of monitoring wells with a typical cost of $1,500 per well per sampling event. Many of these wells are redundant or geostatisticaly insignificant and can be decommissioned. The well_decommission module analyzes all available data and quantifies the impact to site assessment quality of removing each well. The well_decommission module offers the following functionality:

1. Provides an easy to use method to determine which, if any, wells can be decommissioned.
2. Performs baseline analysis using all data.
3. Provides a justifiable approach for determining candidate wells for decommissioning.

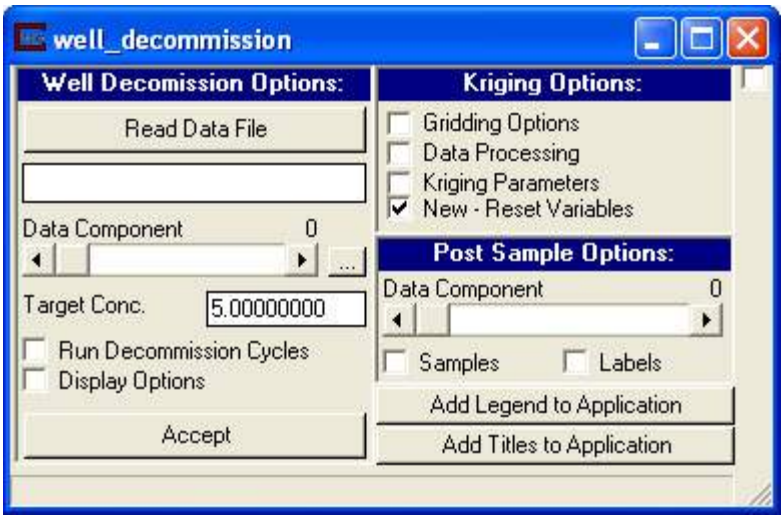
4. Can save thousands of dollars per year for each well identified for decommission.

Module Input Ports

1. external_grid (Blue/Black): This port allows a grid to be imported into the module for the purposes of kriging. If a grid is imported the selected file will be ignored for gridding purposes.
2. Filename (Yellow/Blue/Yellow): This port allows for the sharing of filenames between modules.

Module Output Ports

1. decommission_file (Yellow/Blue/Yellow): This port allows for the sharing of the name of the well decommission file that is created after the analysis at each cycle is complete.
2. status_string (Dark Blue): This port exports a string representing the status of the module while it is running. This string is useful for the making of titles.
3. analyte_name (Dark Blue): This port exports a string representing the analyte name. This string is useful for the making of titles.
4. target_conc (Dark Red): This port exports a float representing the target concentration. This float is useful for the making of titles.
5. display_name (Black): This port exports a string representing the type of display. The type of display is set in the Display Options panel.
6. Output_object(Red): This port outputs a renderable geometry.



Module Control Panel

The control panel for well_decommission is shown in the figure above. This module is a macro of two customized and integrated modules:

Krig_2D

post_samples

A complete description of the windows and options for each of these modules is given in their respective help and is not repeated here. In general you may find some options that are not present in Well Decommission and those were removed because they are not relevant for the function of Well Decommission.

Otherwise the function of each of the sub-modules is nearly identical to their counterparts listed above.

Well Decommission Options

The ***Read Data File*** button allows the user to read in the selected *.apdv, *.aidv, or *.geo file.

The ***Data Component*** slider selects which data component from the file to run the well_decommission algorithm on.

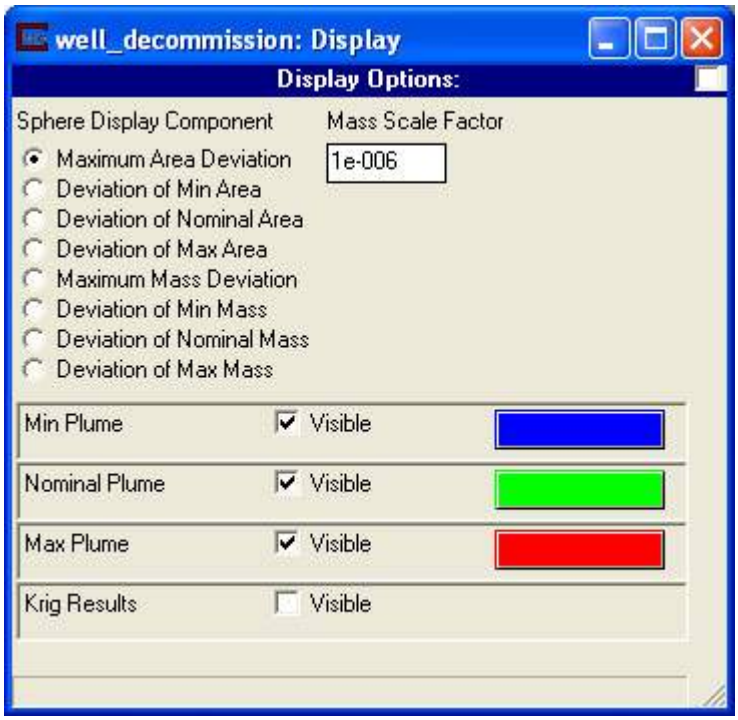
The ***Target Conc.*** is the subsetting level (in real not log units) that is to be used for the decommissioning analysis. This should normally be the EPA reporting level.

The ***Run Decommission Cycles*** toggle when unchecked will run a single baseline analysis. With it checked it will run through a series of calculations that sequentially drop each of the monitoring wells in the input data file.

The **Add Legend to Application** button will automatically instance a Legend module and connect it both to the well_decommission module and to the Viewer. If no Viewer is instanced it will report an error that it cannot find the Viewer.

The **Add Titles to Application** button will automatically instance both a string_format module and a Titles module. And connect these both to the well_decommission module and the Viewer.If no Viewer has been instanced it will report an error that it cannot find the Viewer.

The ***Display Options*** toggle will bring up the Display Options window shown below.



Display Options

The ***Sphere Display Component*** will change which component the spheres from post_samples are colored by.

The ***Mass Scale Factor*** is applied to the Deviation of the Min Mass, the Nominal Mass, and the Max Mass. It is used to scale these values, into more useful units.

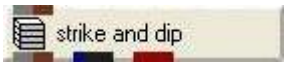
The Min Plume can have its ***Visible*** state toggled on or off, as well as having its color set.

The Nominal Plume can have its ***Visible*** state toggled on or off, as well as having its color set.

The Max Plume can have its ***Visible*** state toggled on or off, as well as having its color set.

The kriging results can be displayed by checking the **Visible** toggle after its name, this option is off by default.

strike_and_dip



(This module is available only in MVS)

General Module Function

The strike_and_dip module is used to visualize sampled locations. It places a disk, oriented by strike and dip, at each sample location. Each disk is probable and can be colored by a picked color, by Id, or by data value. If an ID is present, such as a boring ID, then there is an option to place tubes between connected disks, or those disks with similar Id's.

Strike and dip refer to the orientation of a geologic feature. The strike is a line representing the intersection of that feature with the horizontal plane (though this is often the ground surface). Strike is represented with a line segment parallel to the strike line. Strike can be given as a compass direction (a single three digit number representing the azimuth) or basic compass heading (e.g. N, E, NW).

The dip gives the angle of descent of a feature relative to a horizontal plane, and is given by the number (0°-90°) as well as a letter (N,S,E,W, NE, SW, etc.) corresponding to the rough direction in which feature bed is dipping.

We do not support the Right-Hand Rule, therefore all dip directions must have the direction letter(s).

Module Input Ports

strike_and_dip has a one input port (gray-brown) to provide the *Z Scale* factor

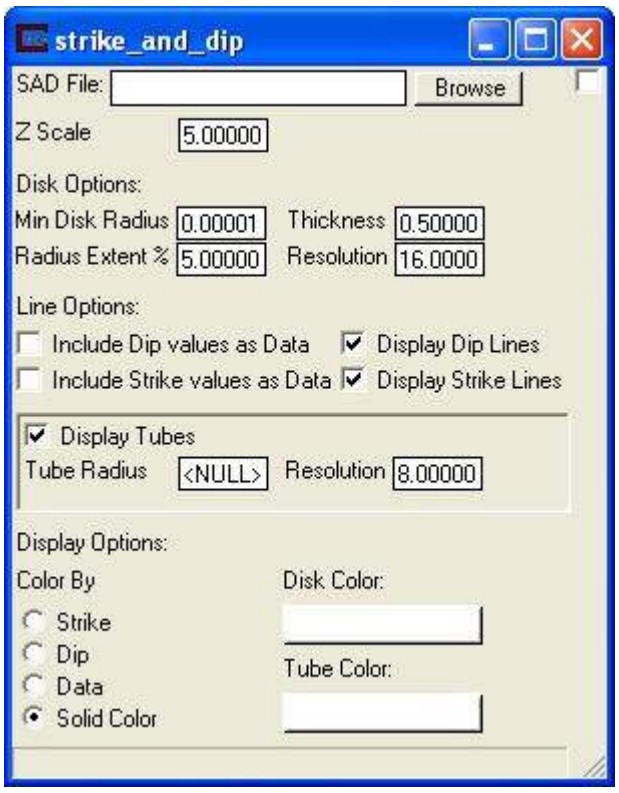
Module Output Ports

strike_and_dip has three output ports:

The leftmost (gray-brown) port is the *Z Scale* factor.

The second (blue-black) port outputs a field comprised of three cell data components : strike, dip, and Data

The third (red) port outputs a geometry, along with tubes, that can be connected to the Viewer for rendering.



Module Control Panel

The control panel of strike_and_dip is shown in the figure above. The **Browse** button opens a *File Browser* which lists the *.sad files that are present in the current directory shown in the directory window. The format of .sad files is described in the SAD file format help topic.

Note that this module will not begin running until a valid SAD file has been selected.

Disk Options:

Min Disk Radius: This sets the minimum radius for each disk. The coordinates are in
model coordinates.

Thickness : This value sets the thickness of each disk. The coordinates are in model coordinates.

Radius Extent: The disk radius is based upon a percentage of the x, y, and z extents
of the model.

Resolution: Number of prisms used to make each disk. The greater the number of
of prisms the more "round" the disk. Resolutions based on a factor of
of four are recommended as they will allow the dip and strike lines to
end at the extents of the disk.

Line Options:

Include Dip values as Data: This allows you to color the Dip lines based upon the dip angle.

Include Strike values as Data: This allows you to color the Strike lines based upon the strike
angle.

Display Dip Lines: This toggle turns the Dip lines on and off.

Display Strike Lines: This toggle turns the Strike lines on and off.

Tube Options:

Display Tubes: This toggle turns the tubes on and off.

Tube Radius: This is the radius for the tubes, set by default to be 2 percent of the extents of the model

Resolution: This alters the number of sides each tube has, the more sides the more "round" the tubes will look, and possibly the slower they will display.

Display Options:

Color by Strike: Colors the disks and tubes by the Strike angle.

Color by Dip - Colors the disks and tubes by the Dip angle.

Color by Data - Colors the disks and tubes by the Data value.

Color by Solid Color: Allows the user to select a constant colors for both disks and tubes.

Strike and Dip File Example

Discussion of Strike and Dip Files

Strike and dip files consist of 3D coordinates along with two orientation values called strike and dip. A simple disk is placed at the coordinate location and then the disk is rotated about Z to match the strike and then rotated about Y to match the dip. An optional id and data value can be used to color the disk.

Format:

You may insert comment lines in C Tech Strike and Dip (.sad) input files. Comments can be inserted anywhere in a file and must begin with a '**#**' character.

Strike can be defined in the following ways :

1) For strikes running along an axis:

   N, S, NS, SN are all equivalent to 90 or 270

   E, W, EW, WE are all equivalent to 0, 180 or 360

   NE, SW are both equivalent to 45 or 225

   NW, SE are both equivalent to 135 or 315

2) For all other strikes: any azimuth direction between 0 and 360 degrees

Dip can be defined only as a degree followed by a direction such as 35E.

There is no required header for this file type.

Each line of the file must contain:

X, Y, Z, Strike, Dip, ID (optional), and Data (optional).

NOTE: The ID can only contain spaces if enclosed in quotation marks (ex "ID 1").

EXAMPLE FILE

```
# x        y        z      strike     dip
51.967  10.948 26.127 -5.205   59.803188E
50.373  33.938 26.127 13.048  68.499864E
51.654  60.213 26.127 -9.108  76.742125E
50.529  83.203 26.127 -13.50  62.945989E
64.358  76.634 11.471 -14.23  80.386294E
```

66.430 33.938 -6.849 -1.421 60.385837E
75.901 50.360 -21.505 -0.141 72.88960E
72.943 7.663 -21.505 5.255 65.512417E
101.90 30.654 -72.801 -7.675 65.952504E
81.339 50.360 -43.489 -4.285 70.707799E
72.263 73.350 -21.505 -2.929 69.314259E
89.897 73.350 -61.809 -4.531 55.678350E
END

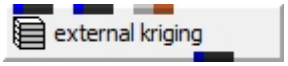
FILE TAGS:

The file tags for the ASCII file formats (shown in ***Bold Italics***) are discussed below with a representative example. They are given in the appropriate order. If you need assistance creating software to write these file formats, please contact support@ctech.com.

COORD_UNITS "ft"
Defines the coordinate units for the file. These should be consistent in X, Y, and Z.

END (this is optional, but should be used if any lines will follow your actual data lines)

external_kriging



This module is available only in EVS-PRO and MVS

General Module Function

The external_kriging will output both data and a grid in the simple GeoEAS format. This data can be kriged externally from EVS/MVS and the results in the same format can then be read into EVS and added to the input grid.

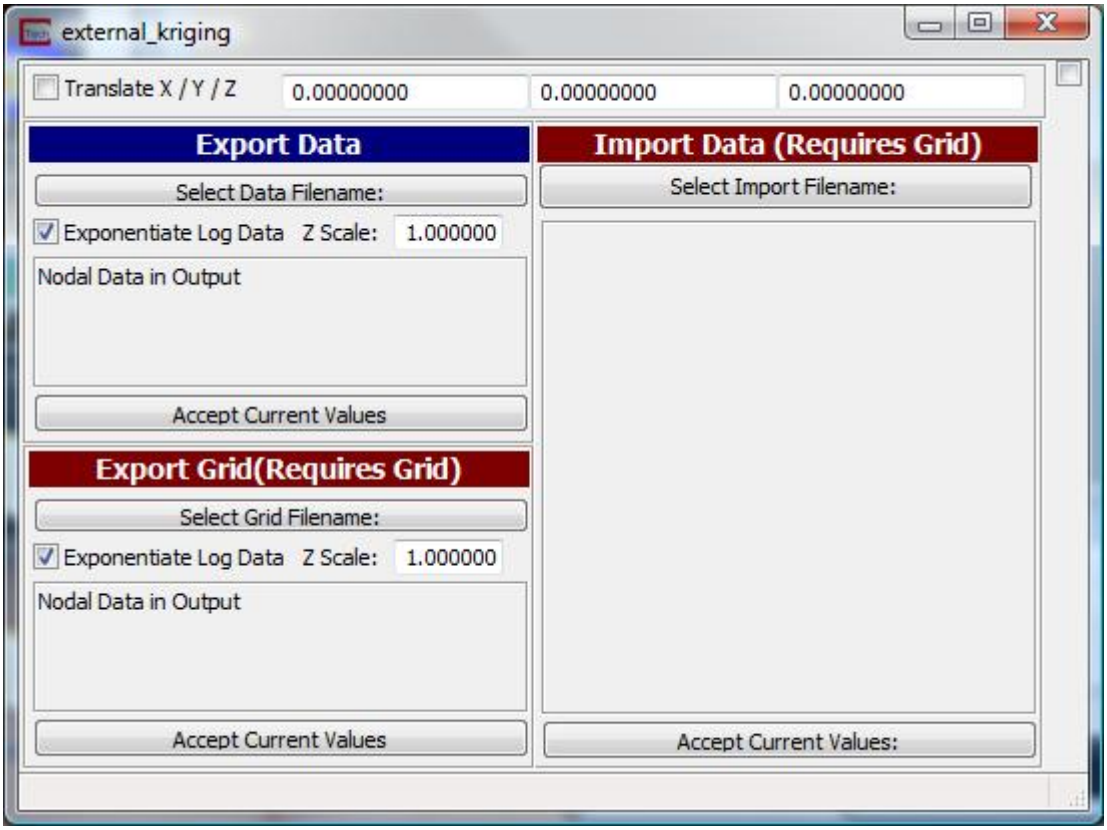
Module Input Ports

1) in_data (Blue-Black): This port takes any field with nodal data.
2) in_grid (Blue-Black): This port takes a grid.
3) z_scale (Gray-Brown): This ports takes a float value that is the z_exaggeration of the input coordinates.

Module Ouput Ports

1) out_fld (Blue-Black): This port outputs a grid containing the selected imported data.

Module Control Panel



The external_kriging control panel is shown above.

The **Translate X / Y / Z** toggle when turned on, translates the entire model back to the origin based on the values input. If the input values are left at 0.0 the values will be calculated based on the centroid of the model.

Upon importing your model, all imported coordinates will be translated by the same values to return your model to the proper coordinate system.

NOTE: This toggle should be used if experiencing precision issues with other software.

Export Data and Export Grid Panels:

- The Select Data/Grid Filename button will set the output GeoEAS file name for the nodal/grid data.

- Selecting the Exponentiate Log Data toggle will cause any data that has been log processed to be exponentiated before being written out.
- When the Accept Current Values button is selected the file will be written out.

Import Data Panel:

- Select Import Filename allows the user to select the data file they wish to import.
- The data columns from the imported file can be toggled on or off, and the data can be log processed if necessary.
- When the Accept Current Values button is selected the module will try to match each X,Y,Z coordinate found in the file with a node coordinate from the input grid. If these coordinates do not match an error will be reported.

raster_to_geology

General Module Function

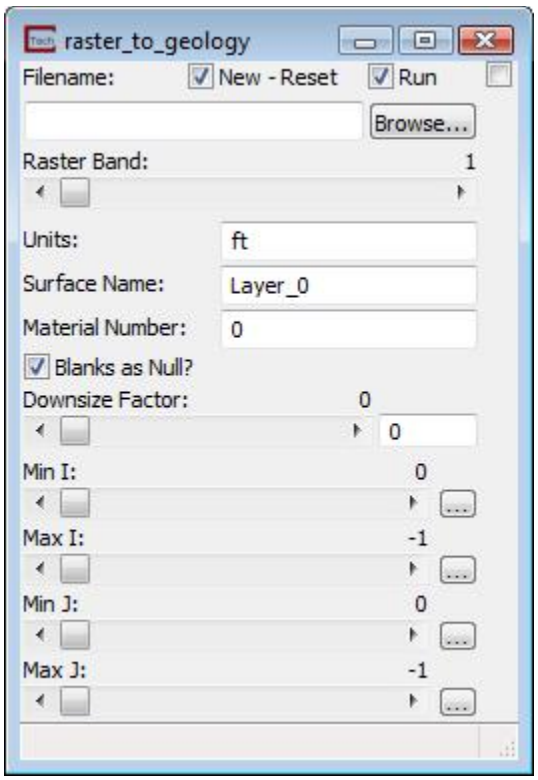
The raster_to_geology module reads several different raster format files in EVS Geology format. These formats include DEMs, Surfer grid files, Mr. Sid files, ADF files, etc.. Multiple raster_to_geology can be combined with combine_geology into a 3D geologic model. Alternatively, a single file can be displayed as a surface (with geologic_surfaces) or you can export its coordinates (with Write_Coordinates) to use the values in a GMF file.

Module Input Ports

raster_to_geology has no input ports since it builds its output from a raster file.

Module Output Ports

1) out (Blue-Grey-Green): This output port is identical to the port on Krig_3D_Geology or Spline_Geology.



Module Control Panel

The control panel for raster_to_geology is shown in the figure above.

The Raster Band slider is used to select a single raster band when reading image formats such as jpeg or tif.

The ***Units***, ***Surface Name***, and ***Material Number*** type-ins allow you to set the values for the surface you are reading. This provides equivalent information as to what is specified in a .Geo or .GMF file and passed through by Krig_3D_Geololgy.

The **Blanks as Null?** option is used to specify that points marked to be blanked out in the raster will be left out of the geologic model when it is created. It is highly recommended to leave this checked.

Additional controls for ***cropping*** and ***downsizing*** the grid during reading are provided to dramatically reduce memory usage and increase speed of multi-layer 3D models.

Save_EVS_Field

General Module Function

The Save_EVS_Field module creates a file in one of 5 different formats containing all the mesh and nodal and/or cell data component information sent to the input port.

This module is useful for writing the output of modules which manipulate or interpolate data (Krig_3D , Krig_2D , etc.) so that the data will not need to

be processed in the future. The processed data can be read using Load_EVS_Field, which is much faster than reprocessing the data.

For a description of the .EFF file formats click here.

Save_EVS_Field supplants the need for Write_UCD, Write_netCDF and Write_EVS_Geology by incorporating all of their functionality and more in a single module. It saves (writes) a dataset in any of five different EVS/MVS compatible file formats, including the new EVS Field Formats:

1) .eff ASCII format, best if you want to be able to open the file in an editor or print it

2) .efz GNU Zip compressed ASCII, same as .eff but in a zip archive

3) .efb binary compressed format, the smallest & fastest format due to its binary form

The EVS Field Formats *.eff;*.efz;and *.efb support **all** types of EVS field output including:

1) Uniform fields

2) Geology (from Krig_3D_Geology and Spline_Geology)

3) Structured fields (such as irregular fields read in from Read_Field)

4) Unstructured Cell Data (UCD format) general grids with nodal and/or cell data

5) Special fields containing spheres (which are points with radii)

6) Special fields containing color data (such as from Read_DXF)

Note: Because the .eff, .efz and .efb formats better handle all types of EVS/MVS output, these three formats are recommended for use over UCD, netCDF or Field.

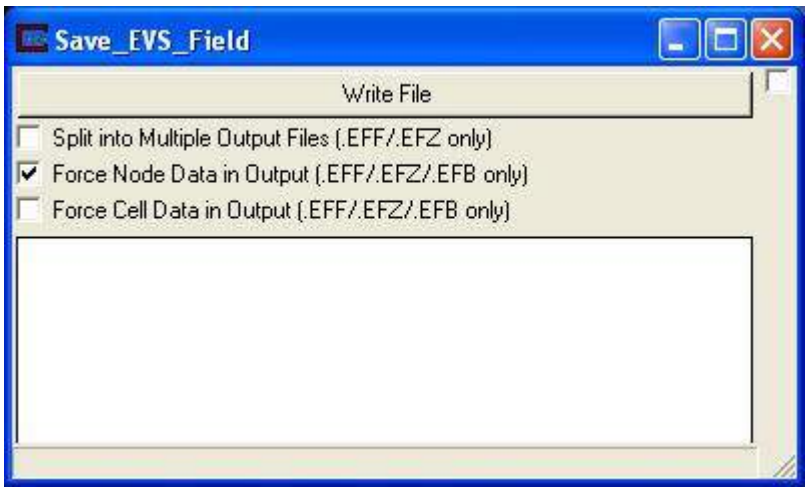
Module Input Ports

The Save_EVS_Field module is shown above.

1. The module's first input port (the leftmost one) accepts any mesh with optional nodal and/or cell data.
2. This port (Brown-Grey-Light Brown-Beige) receives geology info from those modules that output geologic data (Krig_3D_Geology or Spline_Geology).
3. The third port accepts a string that will be saved in the file as notes. These notes are used to describe the content of the file, but are not necessary for the file to be loaded.

Module Output Ports

Save_EVS_Field has no output ports.



Module Control Panel

The Save_EVS_Field control panel is shown above.

Clicking on the Save_EVS_Field's *Write File* button opens a standard windows file browser. The filename and location can be specified in this browser. The suffix selected determines the format in which the field will be saved. The supported suffixes are *.eff;*.efz;*.efb;*.inp; and *.cdf.

There are also three options with the following meaning:

1) Split into Multiple Output Files: Splits the output into separate files. The corresponding .EFF or .EFZ file will only contain the tag information (see below). Any required nodes, data, or connectivity will be written into separate files.

2) Force Node Data in Output: Causes the field definition to always contain Node_Data. If this is off, and there are no node data components, a simpler, Mesh-Only style field will be saved. However, when you load those fields into EVS or MVS, many modules will not work correctly with them. This will cause a mesh like that to save as a Mesh+Node_Data (with 0 data components), which will allow many modules to work with this data.

3) Force Cell Data in Output: Same as above, but with cell data. Very few modules require cell data, so this is not on by default, but is provided as an option.

The window below these options contains the string of notes that will be inserted into the saved file. These notes do not affect the processing of the file in any way, and are there for the sole purpose of providing a description of the file contents.

Related Modules

- Read_UCD
- Write_UCD
- Read_netCDF
- Write_netCDF

- Read_Field

write_lines

General Module Function

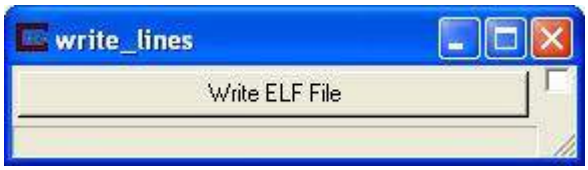
The write_lines module is used to save a series of points with data connected by lines. These lines are stored in the EVS Line File format.

Module Input Ports

write_lines has one input port that takes in a field consisting of lines. For example the output of polyline_spline.

Module Output Ports

write_lines has no output ports.



Module Control Panel

The control panel of write_lines is shown in the figure above. The Write ELF button opens a *File Browser* which allows you to enter the name of the new *.elf file.

Write_Coordinates

General Module Function

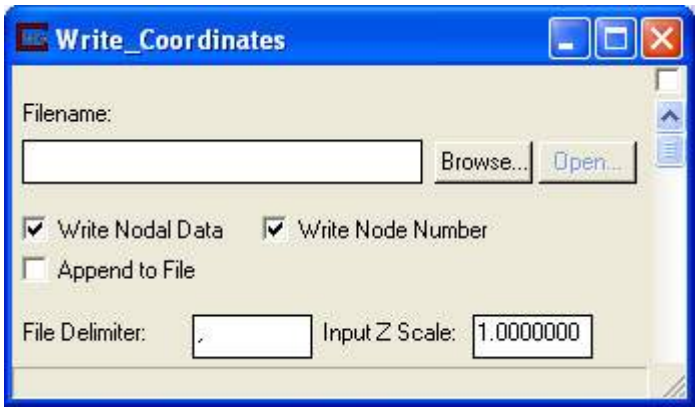
Write_Coordinates provides a means to export an ASCII file containing the coordinates (and optionally the data) of any object in EVS/MVS. The output contains a header line and one row for each node in the input field. Each row contains the x, y, & z coordinates and optionally node number and nodal data.

Module Input Ports

Write_Coordinates has two input ports.

This port accepts the output from any module with a blue/black (field) output port.

The second port accepts a Z exaggeration factor. All coordinates written to the output file will unscale their z coordinates by this factor.



Module Control Panel

The control panel for Write_Coordinates is shown in the figure above.

The *Write Nodal Data* toggle causes all nodal data to be included in the file.

The *Write Node Number* toggle causes the node numbers to be included in the file.

The *Append to File* toggle causes the data to be appended to an existing file and suppresses the header line for the appended data section.

The *Open* button allows you to open the file you've created in your default editor.

File Delimiter allows you to specify the delimited for the .txt file. The default value is a comma. There is one special delimiter the tag "<TAB>" which will tab delimit the text file.

You can specify a GMF file or a APDV file as well as a text file.

1. If you choose GMF, you'll get "surface 0 Unknown" prepended instead of the standard header, which allows you to create (and append to) a valid GMF file. This makes it easy to convert data from other sources (like a surfer grid or shapefile) into a gmf file for kriging.
2. If you choose APDV, you cannot append, but the correct APDV header is written, and all the nodal data gets specified as data components. This makes it easy to convert data from other sources into a APDV file. This happens automatically based on the file extension chosen.

Related Modules

-> Read_UCD

-> Read_DXF

write_vector_GIS

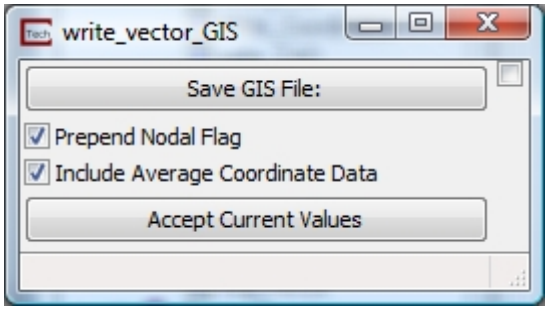
General Module Function

The write_vector_GIS module will create a file in one of the following vector formats: ESRI Shapefile (*.shp); GMT ASCII Vectors (*.gmt); and MapInfo TAB (*.tab). Module Input Ports

1. in_field (Blue-Black): This port takes a mesh as input.
2. z_scale (Gray-Brown): This port takes in a float value related to the z exaggeration of the input mesh.

Module Output Ports
No output ports.

Module Control Panel



The user interface for write_vector_GIS is shown above.

- The ***Save GIS File*** button will select the file name to be output. The file extension used will determine the type of file written.
  This module outputs files whose formats allow for only cell data. Turning the Prepend Nodal Flag toggle on will add the prefix "n_" before each nodal data name when writing, indicating that the nodal data has been averaged across the cell.

- Turning the ***Include Average Coordinate Data*** toggle will add four cell data components to the output file. These are the absolute cell number average coordinate of each cell in X, Y, and Z.

- When the ***Accept Current Values*** button has been selected the module will examine the type of cells passed to the input field. It will generate a file for each cell type (i.e. points, lines, and tris/quads). If there is only one cell type in the input field the filename will remain as selected. If there are multiple types of cells then the filename will be appended with the input cell type.

write_CAD

General Module Function

write_CAD provides EVS with vector output of graphical results in industry standard AutoCAD DWG or DXF format.

write_CAD will output one or more individual objects (red port) or your complete model (purple input port from the Viewer). Volumetric objects in EVS are converted to surface and line type objects.

write_CAD preserves the colors of all cells and objects by assigning **cell** colors to each AutoCAD surface or line entity according to the following procedure:

a) If nodal data is present, the first nodal data component is averaged to the cells and that color is applied. This is equivalent to the appearance of surfaces in EVS with flat shading mode applied.

b) If no nodal data is present, but cell data is, that color is applied. This is equivalent to the appearance of surfaces in EVS with flat shading mode applied.

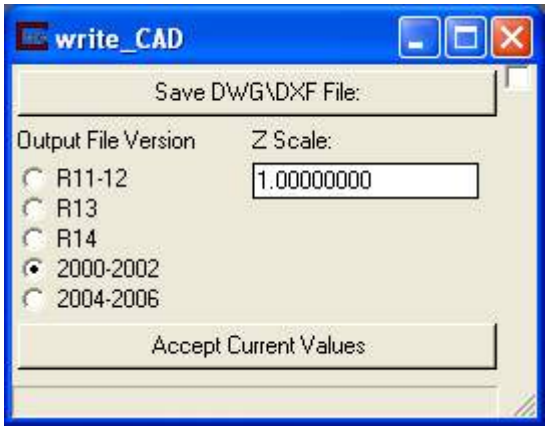
c) If neither nodal or cell data is present the object's color is used.

The results should look virtually identical to the Viewer in EVS except for the lack of gouraud shading support in AutoCAD.

All "objects" in EVS are converted to separate layers based upon the EVS object name (as shown in the Viewer's Object_Selector).

Module Input Ports

write_CAD has three input ports. The left most (purple) port can connect only to the Viewer and will result in your complete Viewer contents being written. The second (red) port can accept any number of objects from one or more modules in your application. This allows you to write out a select subset of your application's objects. The third (grey-brown) port receives the Z Scale.



Module Control Panel

The control panel for write_CAD is shown in the figure above.

Clicking the ***Load DWG\DXF File*** button opens a standard windows style file browser which allows the user to select an AutoCAD DWG or DXF file in a specified directory. The module runs as soon as you specify the file.

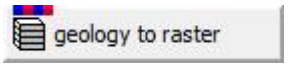
Output File Version allows you to specify which AutoCAD file format you require.

The ***Z Scale*** type-in **unscales** your objects if you wish to have the CAD output with no z exaggeration.

Related Modules

-> read_CAD

geology_to_raster



General Module Function

geology_to_raster is used in conjunction with Krig_3D_Geology or Spline_Geology. Both of these modules can create rectilinear grids based on your geologic data. A large number of formats are supported such as Surfer and ESRI grids. For some formats, each cell in your grid should be the same size. This will require you to adjust the extents of your grid and set the grid resolution according to:

Cell size = (Max:xy - Min:xy) / (grid-resolution -1)

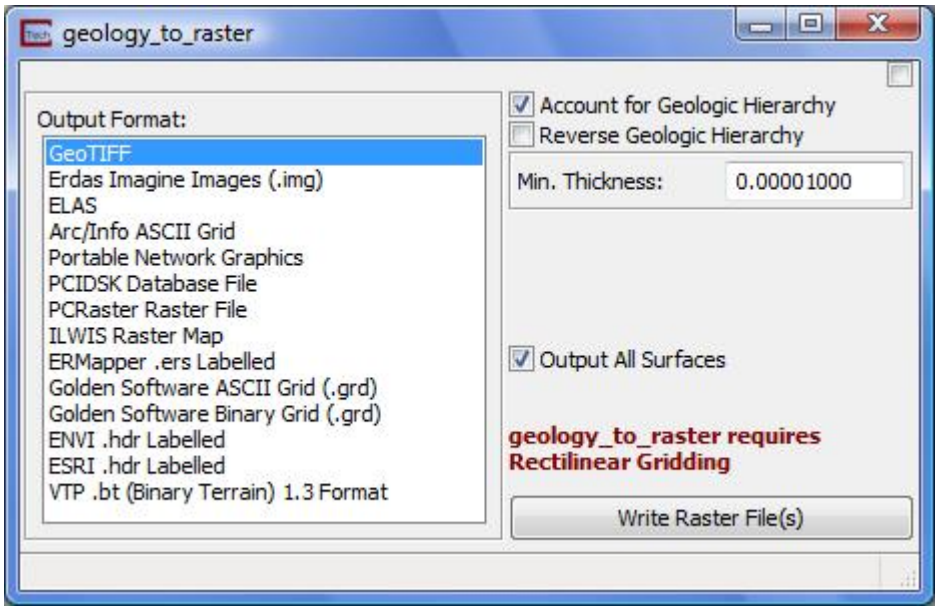
NOTE: YOU MUST SELECT RECTILINEAR GRIDDING IN KRIG_3D_GEOLOGY OR SPLINE_GEOLOGY.

Module Input Ports

geology_to_raster has one input port which accepts all required grid information from either Krig_3D_Geology or Spline_Geology.

Module Output Ports

geology_to_raster has no output ports.

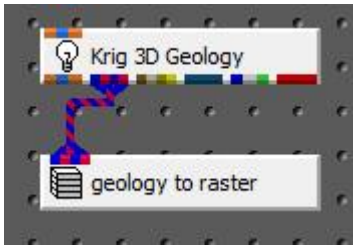


ModuleControl Panel

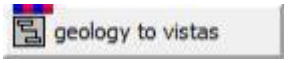
The geology_to_raster panel is shown above. The parameters are:

1. ***Output Format:*** allows you to select the desired output type
2. ***Account for Geologic Hierarchy*** is a toggle that forces hierarchy on multiple surfaces. Use this IF the surfaces represent a geologic hierarchy.

3. ***Reverse Surface Hierarchy*** – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 13.
4. ***Min(imum Layer) Thickness*** forces layers thinner than the specified value to have a minimum thickness and therefore not pinch-out completely.
5. ***Output All Surfaces*** is a toggle that determines if only the first surface is output or whether all should be included.
6. The ***Write Raster File(s)*** button opens a file browser and causes the output to be created.



geology_to_vistas



General Module Function

geology_to_vistas is used in conjunction with Krig_3D_Geology or Spline_Geology. Both of these modules can create finite difference grids based on your geologic data.

It writes the fundamental geologic grid information to a file format that Ground Water Vistas can read.

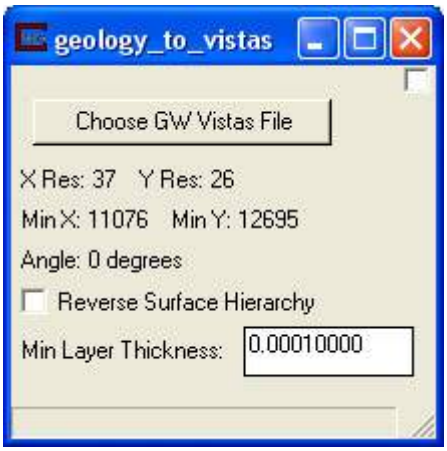
The output includes the x,y origin; rotation; and x-y resolutions in addition to descriptive header lines proceeded by a "#".

Module Input Ports

geology_to_vistas has one input port which accepts all required grid information from either Krig_3D_Geology or Spline_Geology.

Module Output Ports

geology_to_vistas has no output ports.

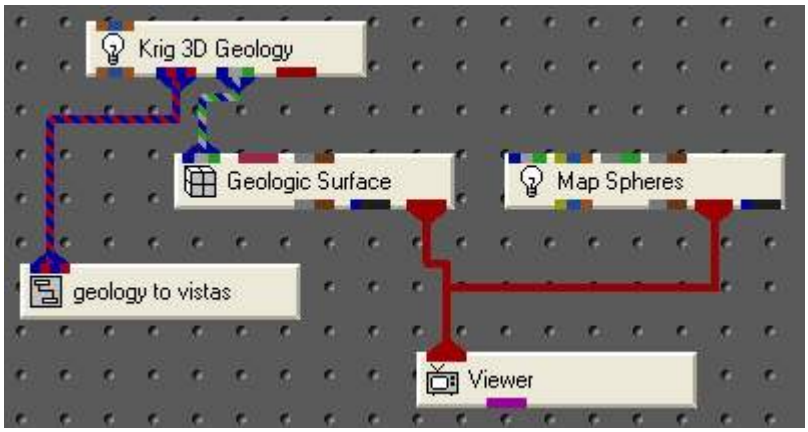


ModuleControl Panel

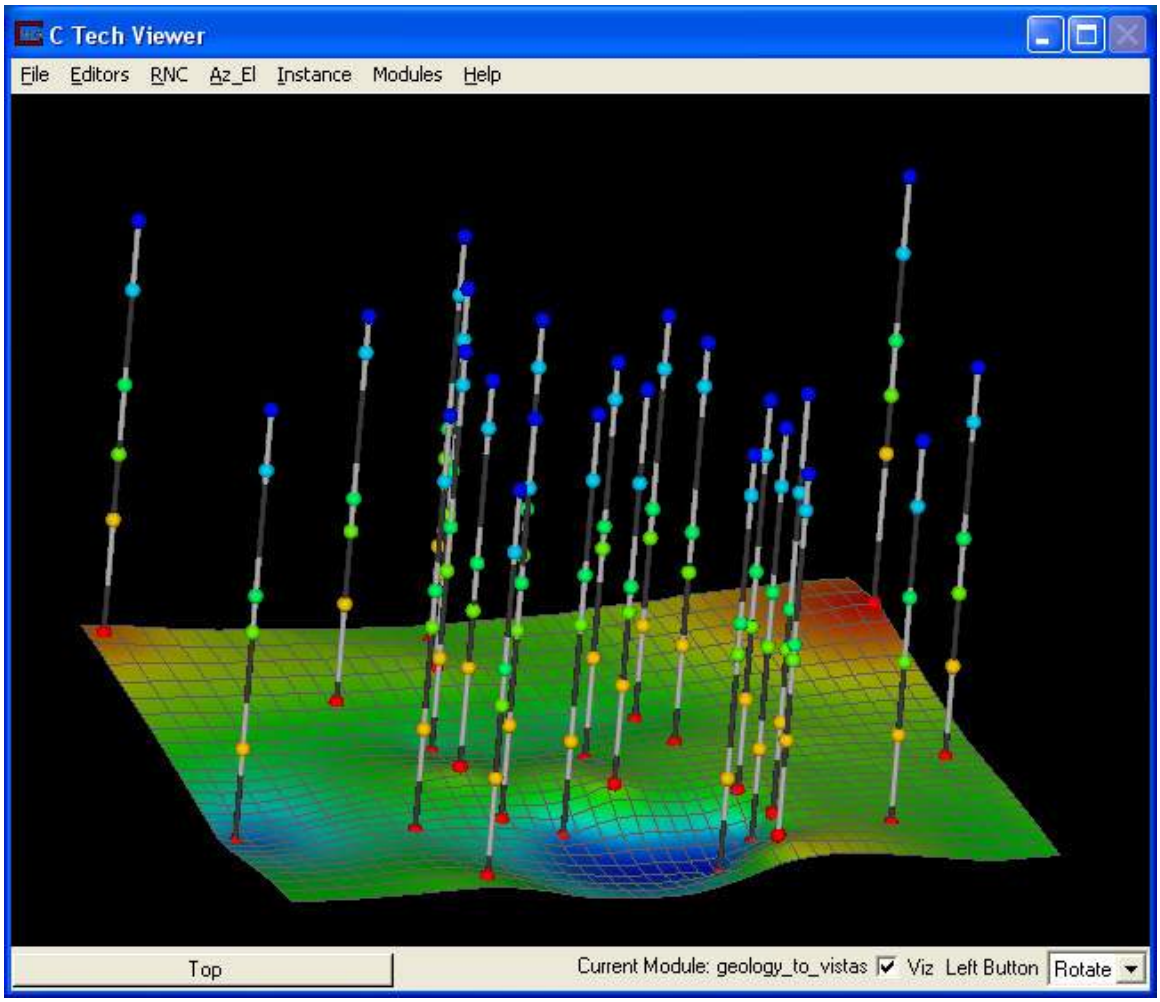
The geology_to_vistas panel is shown above. There are only two parameters (besides the output file name).

Reverse Surface Hierarchy – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

Minimum Layer Thickness forces layers thinner than the specified value to have a minimum thickness and therefore not pinch-out completely.



An example of an application using this module is shown above. The output from this application is shown below.



write_VRML

General Module Function

This module is only available in MVS.

The write_VRML module is able to output most graphics objects in the Viewer to a VRML-formatted file.

VRML is a network transparent protocol for communicating 3D graphics over the World Wide Web. It has fallen out of favor on the web, though it is still a standard for 3D model output.

We provide VRML output for two primary purposes:

1. Export of 3D models for conversion to 3D PDF
2. Export of 3D models for 3D Printing

Known Issues

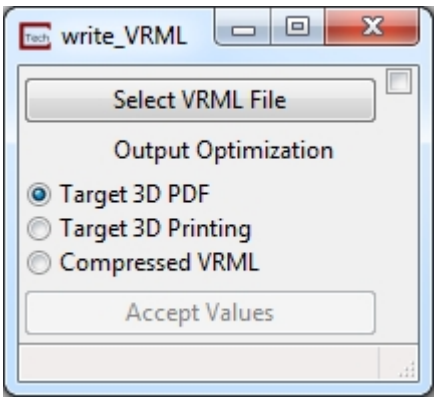
- Turn on the "Use Unlocked Background" option in the Viewer->Background editor when writing VRML files, since the background is otherwise rendered as a small square at the origin.
- Always set your Viewer to a Top View (180 Azimuth and 90 Elevation) before writing the VRML file.
- Do not use any modules which display in the 2D overlay. The 2D overlay is analogous to drawing on the glass on a TV or monitor. Items in the 2D overlay do not move, rotate or scale when you manipulate your 3D model. Examples are add_logo, Titles, and Legend.
- Do not use volume rendering. These techniques are not supported.
- VRML does not support the full spectrum of data coloring supported in EVS.
    - Though both cell and nodal data coloring is supported, sometimes combinations of these cause problems.
    - Object colors (such as the red, blue, green grid lines of the axes module) often revert to white (uncolored). This can be problematic on a white background.
    - The texture_colors module is recommended for final output of most all colored objects to help avoid these issues.
- Trial and Error is often the only way to determine what combinations of rendering modes are supported, especially for 3D PDF and 3D printing. Remember these vendor's software all interpret the VRML files in slightly different ways. You will likely not be able to do everything you can do in a 4DIM or in EVS/MVS.
- VRML Viewers: There is a list of VRML viewing software published by National Institute of Standards and Technology here. We recommend Cosmo, though it is far from perfect. We have created VRML files which will not display correctly in any of the VRML Viewers that we have tested (including Cosmo), but which DO convert to 3D PDF perfectly. Conversely, there are occasions when something will look ok in VRML and not convert properly to 3D PDF.

Module Input Ports

write_VRML has one purple input port which connects to the viewer.

Module Output Ports

None



Module Control Panel

The write_VRML module captures the contents of the view and writes graphics primitives to a VRML-formatted file. The module contains a user interface shown above. A description of the various options follows:

The *Select VRML File* button allows you to enter the name of the output file to be created. The user may browse to the desired path and/or file via a browse button.

Output Optimization provides three options to improve output for the specified target use:

- Target 3D PDF is intended for creating 3D PDFs using a utility program that is separately licensed and sold by C Tech.
- Target 3D Printing is optimized for output to 3D printers such as http://www.3dsystems.com/
- Compressed VRML outputs generic VRML which is GNUzipped. For use with most VRML viewers.

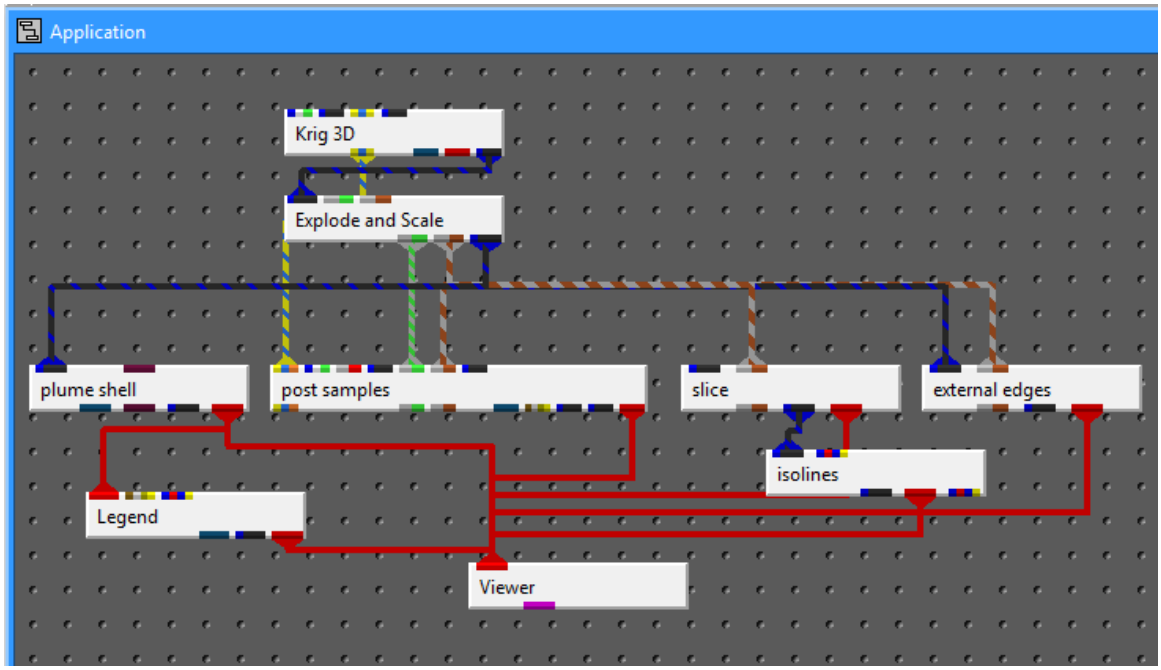
The *Accept Values* button outputs the VRML file.

Guidelines for 3D PDF Creation

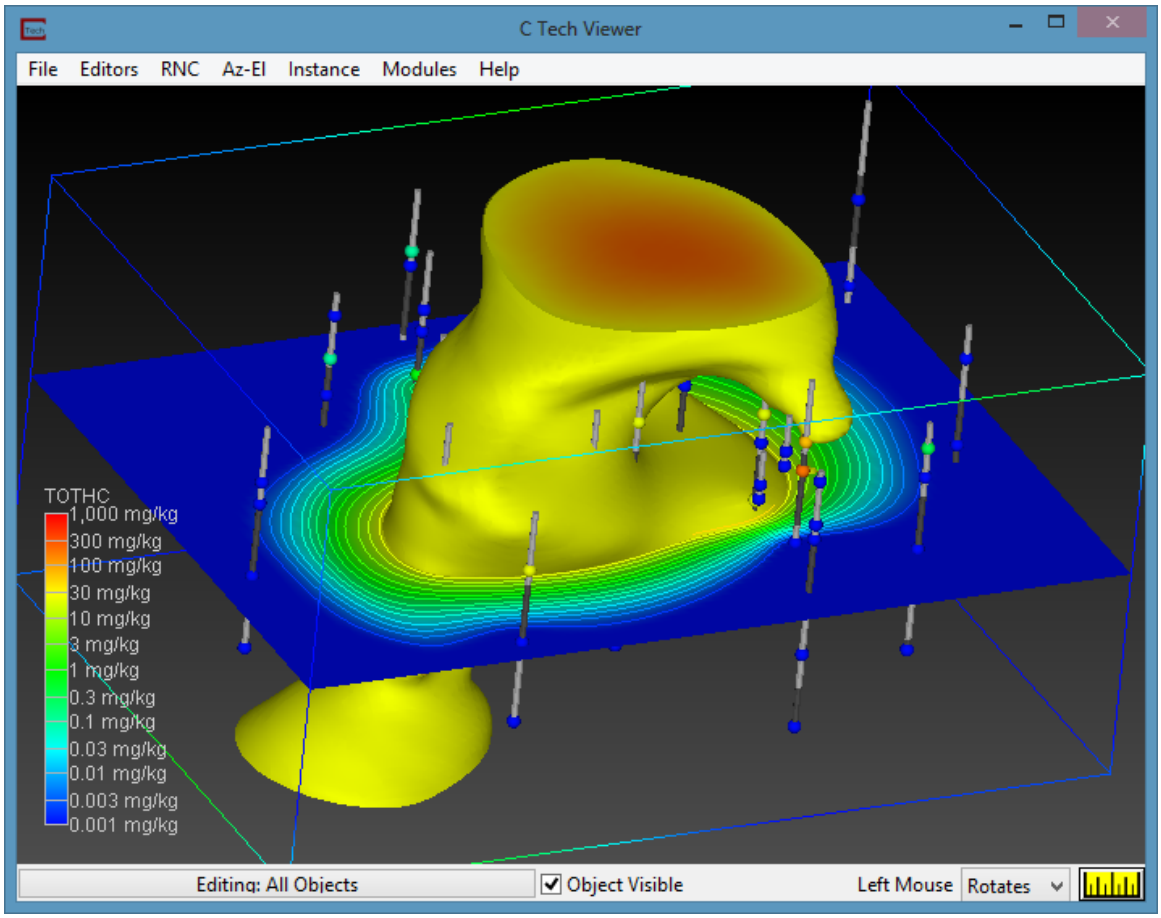
The following is a list of guidelines that must be considered when making MVS models that will be output as 3D PDF files using the *C Tech 3D PDF Converter.*

MVS output from write_VRML. You must follow the guidelines in write_VRML in addition to these additional guidelines.

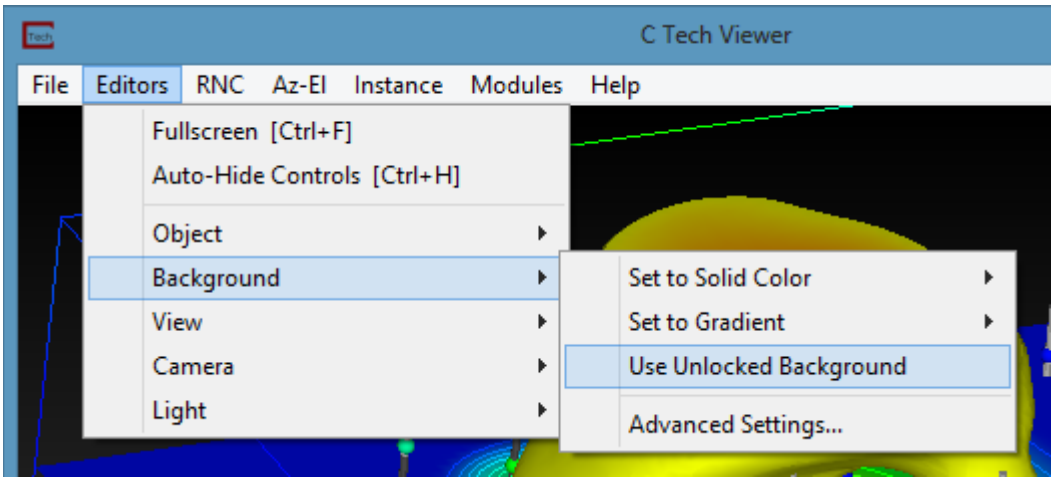
Let's begin by building a simple application



Whose output is:

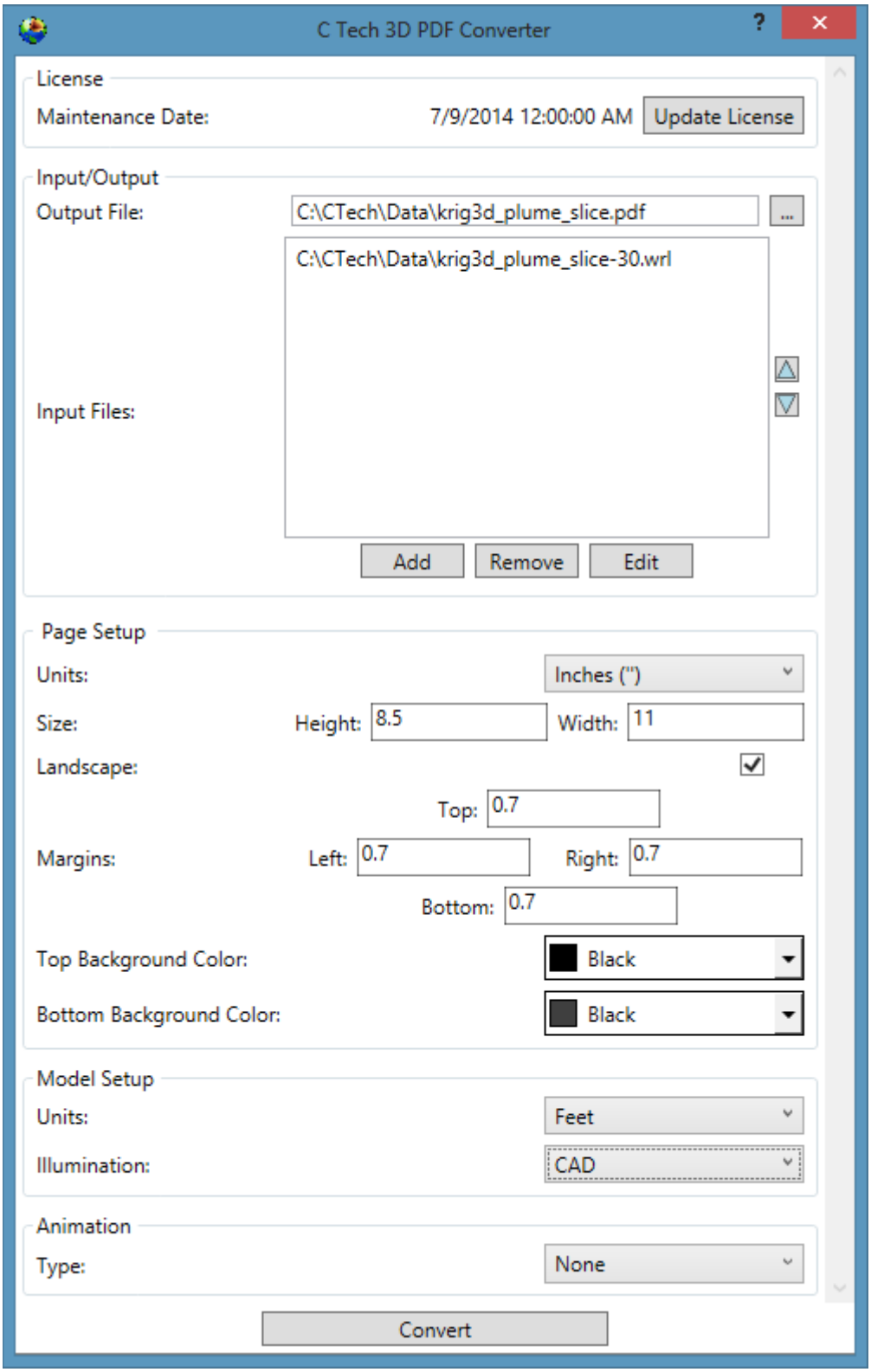


The first things we MUST do for VRML output are to remove the Legend and use an *Unlocked Background*. If you see a gradient background in your Viewer, you definitely aren't using an unlocked background. Once you use an unlocked background, you can still set a solid (single) background color.

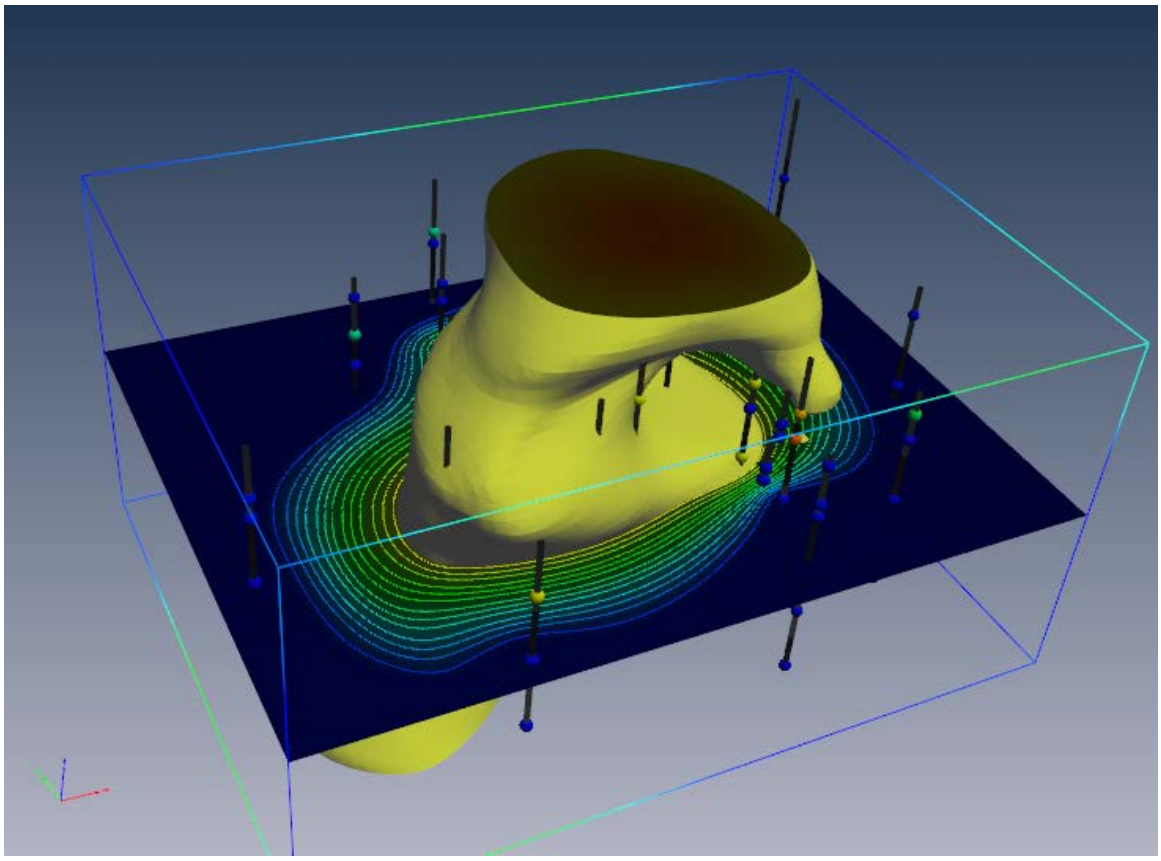


Always set your Viewer to a Top View (180 Azimuth and 90 Elevation) before writing the VRML file.

If we output this current model as VRML and convert to 3D PDF,



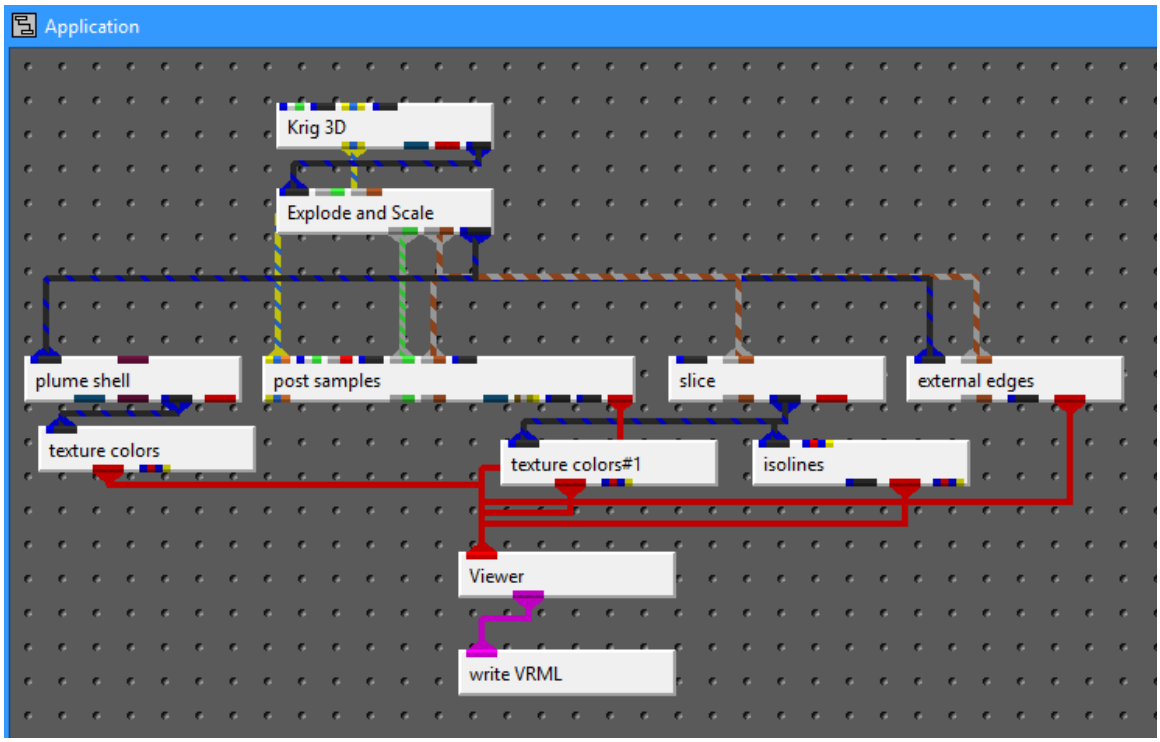
the results are less than wonderful:



The above 3D PDF has three obvious problems:

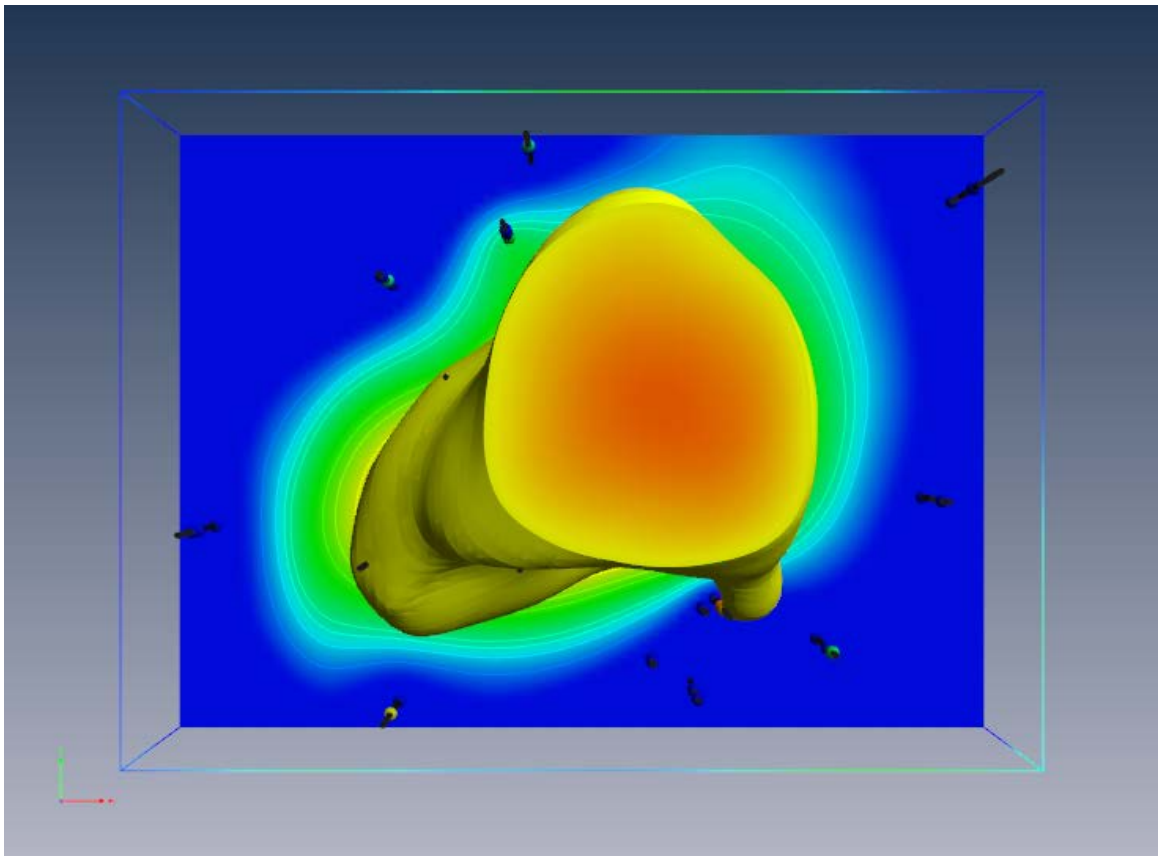
1. The top and bottom of the plume are very dark.
2. The slice is dark
3. post_sample's borings are dark.

We need to modify the application using two texture_colors modules as follows:

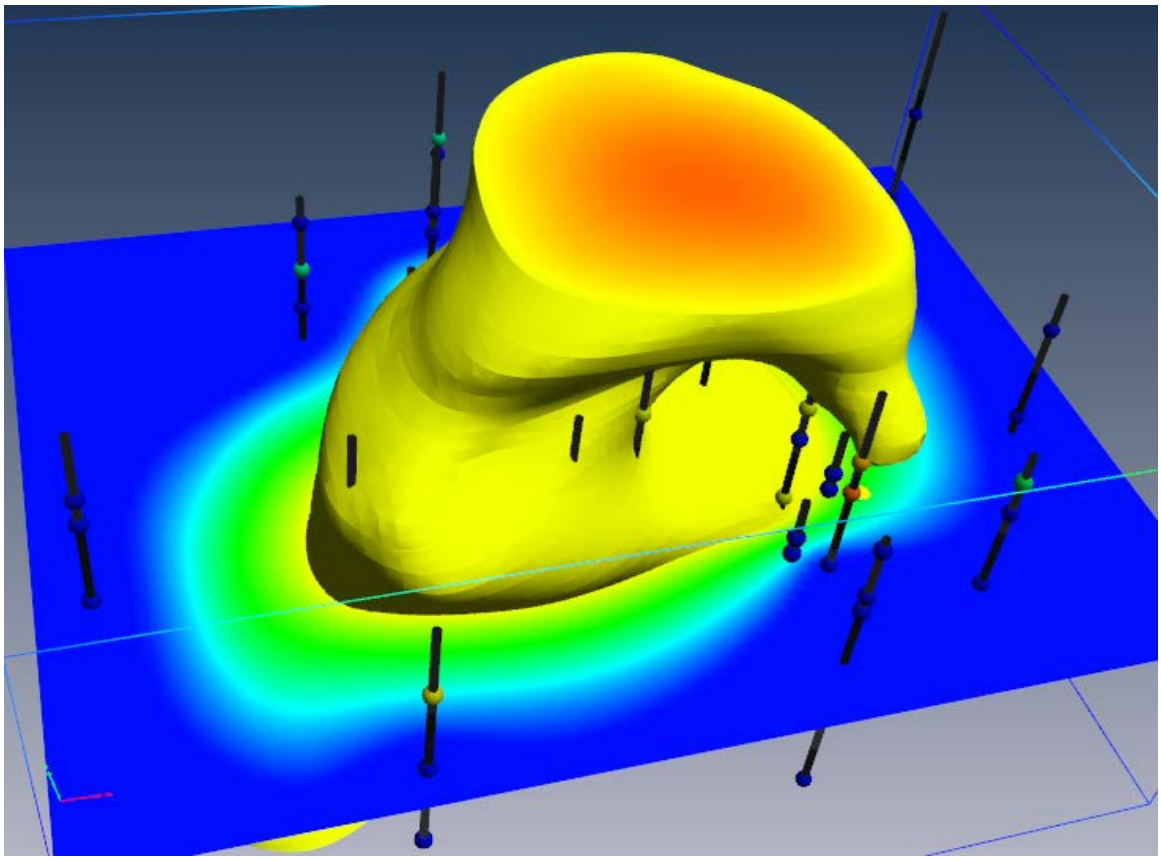


You'll notice that in the revised application, the output in the Viewer is virtually identical. This will address the first two problems, however we expect to resolve the dark borings in an upcoming release.

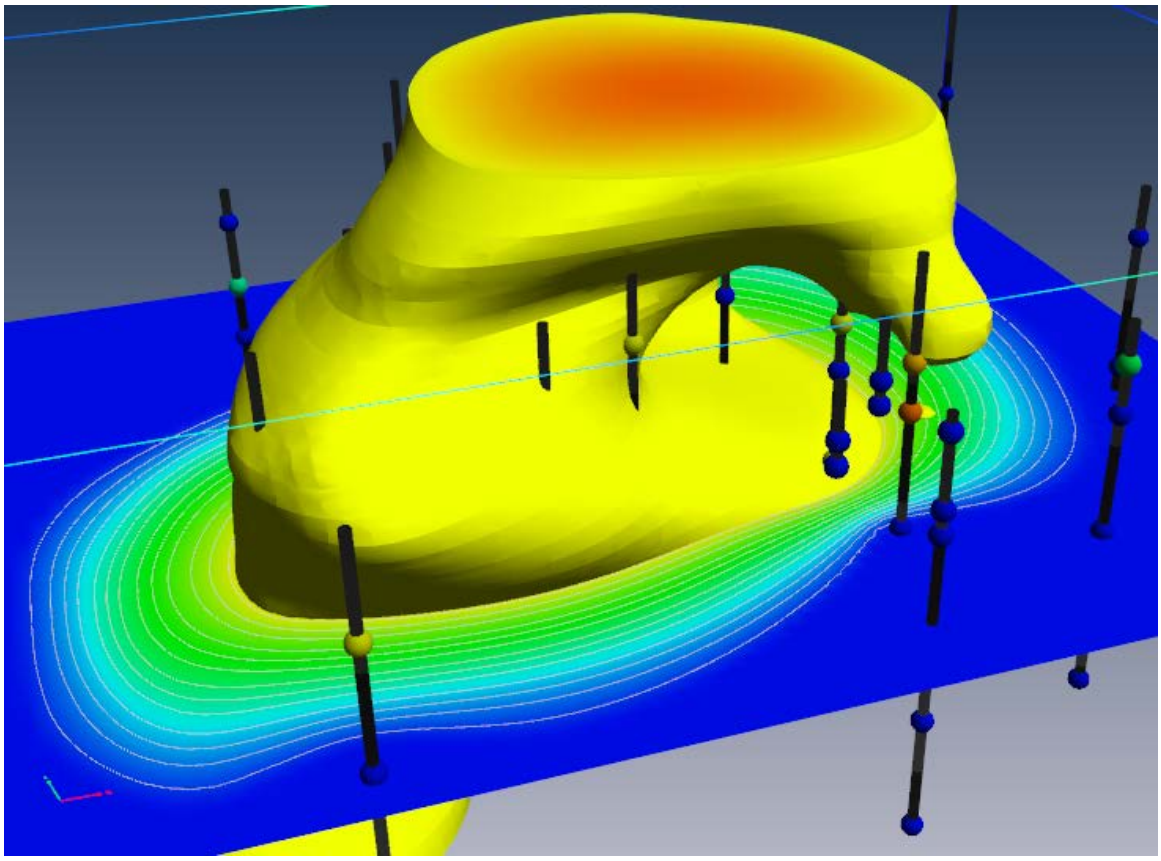
If we export this model to VRML and convert to PDF, the result is:



One other issue is that by default, we create isolines coincident with the surface(s) and resolve the coincidence in EVS using jitter. At some rotations you will notice that the isolines may disappear. This can be because jitter is not supported, but also because the underlying surface is so bright that the lines are not distinguishable.



This can be addressed using the surface_offset parameter in isolines. This will offset the lines from teh surface (in one direction) and eliminate the coincidence. However, this will also mean that the lines will not be visible from one side of the slice. Making the lines uncolored is another option.



Guidelines for 3D Printing

The following is a list of guidelines that must be considered when making visualizations that will be printed using 3D Systems (previously Zcorp) technology. As of this software release, no other full color 3D printer has been successfully tested with MVS output from write_VRML. You must follow the guidelines in write_VRML in addition to these additional guidelines.

These guidelines are provided to minimize printing problems. Users should fully understand the issues below or they will likely not create VRML files suitable for 3D printing. Given the cost of the raw material it is best to do it right the first time!

Many of these issues (if not heeded) will be obvious when the model is viewed in Z Corp's ZPrint software. Make sure the model is carefully examined in ZPrint before actual printing.

1. **Internal Faces:** You must avoid internal External faces. This naturally occurs when we cut a hexahedral volumetric model with plume_volume. The volumetric subset consists of hexahedron and tetrahedron cells. The external_edges module (and many others) creates the external faces of both sets of cells. This creates surfaces that are internal to the model even though they represent the external faces of each set of cells. The real problem here is that the mating surfaces of each cell set

are coincident (see 4 below). This major problem and many others are resolved by the sequential_subset module.

2. **Normals:** Must have all surface normals facing outward to define a solid volume for printing (handled by sequential_subset module)
3. **Coincident surfaces:** You **CANNOT HAVE** coincident surfaces. If two layers (or other objects) have coincident sufaces this will result in open parts and printing problems. You must separate the parts by a small amount (recommend 0.005 inches in final printed size) which should not be noticeable visually. Z-Print's process will fuse these parts together (because there isn't sufficient gap to keep them truly separate).
4. **Overlapping parts:** This is supported. It is possible to have two closed volumes overlap each other and Z-Print will sort it out so long as 1, 2 and 3 above are still valid.
5. **Surfaces**: *Must be extruded* or represented as a volumetric layer. Surfaces have no thickness and if placed coincident with the top of a volumetric object will result in leaving the volume OPEN (unclosed). This will cause serious problems.
6. **Cell Data:** Another limitation is the inability to mix nodal and cell data. Since we use nodal data for so many things you should always strip out the cell data and use nodal data exclusively. You must be aware of the following:
   a. Ensure that there are no modules connected to the Viewer that contain cell data. The safest way to ensure this is to pass questionable modules through extract_mesh with "Remove Cell Data" toggle ON. Normally you would want the "Remove Nodal Data" toggle OFF.
   b. If you want your cell data (colors) to be displayed, pass the cell data through the cell_to_node module. However be aware that you'll still need to use extract_mesh afterwards because cell_to_node doesn't remove the cell_data it just creates new nodal data from cell data.
   c. Typical modules that have cell data are Read_Shapefile, Indicator_Geology, Solid_3D_Set, Solid_contour_set, and most of the modules in the **Cell Data** library.
7. **Explode distance:** Need to ensure that there is sufficient gap between exploded layers (separate parts) so that they don't fuse together. Separation should be 1 mm (0.04 inches) minimum in the final print scale. Be aware that a 1 mm gap in the Z direction isn't equivalent to a 1 mm separation if the mating parts have high slopes. If yourmatingsurfaces have a 45 degree slope, the separation is reduced by cos(45) (~0.7). If you have higher slopes such as 80 degrees, the factor would be ~0.17. This would mean that you would need a Z gap of nearly 6 mmto ensure a 1 mm separation between parts.

8. **Disconnected pieces:** Although Z Print can print disconnected pieces, they won't retain their spatial position. Plumes that aren't connected to solid structure will just be loose pieces in the final print. This would also apply to post_samples' borings and spheres, unless they are connected by some common surface or geologic layer.
9. **Concepts that areNOTSupported**:
    a. **Pointsand Lines:** Points and Lines cannot be printed (except as elements of an image used in a texture map). Lines must be converted to some 3D solid structure (such as closed tubes) and they must be of sufficient thickness to have some strength AND must not be disconnected pieces. Points should be represented as glyphs of sufficient size and not be disconnected.
    b. **Transparency**: Transparency as an object property cannot be supported since Z Print's ink is printed onto opaque plaster or starch powder. The illusion of transparency could be achieved by creating a texture map that was a blend (using the image_transition module) between two different images.
    c. **Volume rendering**: This is a subset of Transparency and therefore is not supported at all.
    d. **Jitter:** First, you must make sure that coincident surfaces are avoided anyway. Jitter is designed into EVS/MVS to allow preferential visualization of coincident objects. With Z Printing we cannot have coincidence in the first place! Offset the desired **primary** object to ensure that it is visible. Remember no lines and no surfaces!
10. **Thin sections:** This is a somewhat subjective issue in that we really can't tell you the definition of *too fragile*?. We would recommend a minimum thickness of 0.5 mm, but depending on the width (total cross sectional area of the section) this may be too fragile or exhibit too much distortion during curing. We still want to have lenses pinch out, but if sections get very thin, the pieces may break.
11. **Top View:** You should write out the VRML file from a top view If there are any truly flat (horizontal) surfaces, this keeps them flatter and smoother. Also, it helps to keep the models with the largest dimensions in the x-y plane (rather than z). This speeds up printing.

symbols

General Module Function

Symbols creates symbolic representations of different borehole identifiers based on a set of user defined parameters. The symbols are displayed at the top of the each borehole based on its x,y & z coordinates. A sample file with 48 predefined symbols is included, but it can be customized to produce special symbols.

Each symbol is made up of three components. The first shape is a fixed polygon with an outline. The thickness of the outline is selectable (via the

control panel). A second polygon, which overlaps the first and has the same number of sides, has selectable minimum and maximum radial values (via the .SYM file). The third component is made up of a user defined set of lines (0 gives no lines). Each polygon has the same number of faces as defined in the #face parameter in the .SYM file. The area created by the difference between the Rmin value and the Rmax value is solid.

Module Input Ports

Symbols has two input ports:

1) The first connects to Explode_and_Scale to inherit the Z_Exaggeration factor.

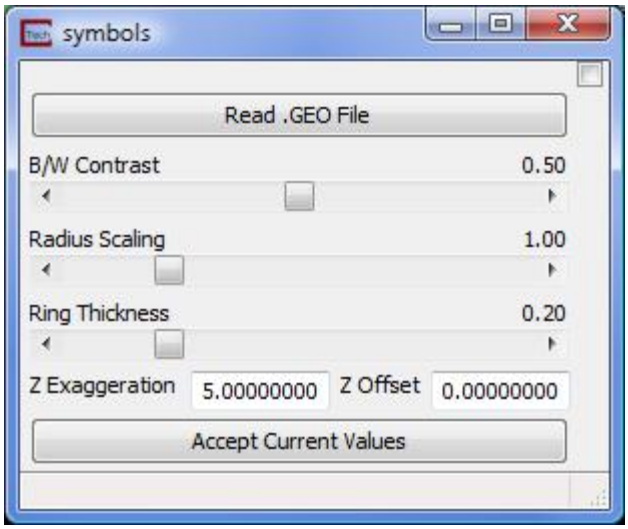
2) The second is the geology file name port.

Module Output Ports

Symbols has two output ports

1) The first is the geology file name port.

2) The second is a renderable geometry directly to the Viewer.



Parameters

Read .GEO File

Use to select the GEO formatted data file. An additional column must be added to the file after the borehole name column. Use the Symbol(sym #) number from user defined symbols definition file to assign each type of sample an appropriate symbol.

B/W Contrast

This value sets the contrast between light and dark colors produced. A value of .50 (default) makes the lightest color .75 gray and the darkest color .25 gray. A value of 1.0 would make white and black.

Radius Scaling

This value is used to scale the radius of the symbol based on user units. The symbols overall size is affected by this parameter.

Ring Thickness

This parameter sets the outline thickness of the first polygon.

Z_Exaggeration

This parameter correlates to the Z coordinate found in the APDV data file. A larger number than 1 will raise the symbol further above the actual Z coordinate.

Z offset

Determines the amount of Z axis offset each symbol will have.

EVS.SYM file:

The following is a listing of the file evs.sym in evs\data\special. This file can be customized to produce other symbols.

```
# rmin rmax lmin lmax #face #line bw rot lrot rvrs name
48
1 0. 1 1 1 12 0 1 0 0 0 solid fill circle
2 0. .7 .7 1.2 12 4 1 0 0 0 solid fill circle w/ line
3 .8 1 1 1 12 0 1 0 0 0 circle ring
4 .4 1 1 1 12 0 1 0 0 0 fat circle ring
5 .0 .4 1 1 12 4 1 0 0 0 circle ring w/lines
6 .8 .7 .7 1.2 12 4 1 0 0 0 circle ring w/lines
7 .4 1 1 1 4 0 1 0 0 0 fat square box
8 .8 1 1 1 4 0 1 45 0 0 thin square box
9 .0 1 1 1 4 0 1 45 0 0 solid square box
10  .0 .7 .7 1.2 12 4 2 30 -30 0 half moon bk top w/line
11  .0 .7 .7 1.2 12 4 2 300 -300 0 half moon bk rt w/line
12 .0 .7 .7 1.2 12 4 2 210 -210 0 half moon bk bot w/line
13 .0 .7 .7 1.2 12 4 4 30 -30 0 qrtr moon bk ul w/line
14 .0 .7 .7 1.2 12 4 4 120 -120 0 qrtr moon bk ur w/line
15 .8 .7 0 1.2 12 4 1 0 0 0 open bulls-eye
16 .0 .7 .7 1.2 12 4 2 120 -120 0 half moon bk lft w/line
17 .0 1 1 1. 3 0 1 30 0 0 solid black triangle
18  .8 .7 .7 1.2 3 3 1 90 0 0 hollow blk triangle w/line
19 .0 1 1 1. 3 0 1 90 0 0 solid black triangle
20  .8 .7 .7 1.2 4 4 1 0 0 0 diamond w/line
21 .8 1 1 1. 4 0 1 0 0 0 diamond
22 .0 .7 .7 1.2 4 4 1 0 0 0 solid diamond w/line
23  .0 .7 .7 1.2 6 6 4 0 0 0 hex moon bk ul w/line
24  .0 .7 .7 1.2 6 6 4 180 0 0 hex moon bk ul w/line
25 0. 1 1 1 12 0 1 0 0 1 solid fill circle
26 0.  .7 .7 1.2 12 4 1 0 0 1 solid fill circle w/ line
27  .8 1 1 1 12 0 1 0 0 1 circle ring
28  .4 1 1 1 12 0 1 0 0 1 fat circle ring
29  .0 .4 1 1 12 4 1 0 0 1 circle ring w/lines
30  .8 .7 .7 1.2 12 4 1 0 0 1 circle ring w/lines
31  .4 1 1 1 4 0 1 0 0 1 fat square box
32 .8 1 1 1 4 0 1 45 0 1 thin square box
33 .0 1 1 1 4 0 1 45 0 1 solid square box
34 .0 .7 .7 1.2 12 4 2 30 -30 1 half moon bk top w/line
35 .0 .7 .7 1.2 12 4 2 300 -300 1 half moon bk rt w/line
36 .0 .7 .7 1.2 12 4 2 210 -210 1 half moon bk bot w/line
37 .0 .7 .7 1.2 12 4 4 30 -30 1 qrtr moon bk ul w/line
```

```
38 .0 .7 .7 1.2 12 4 4 120 -120 1 qrtr moon bk ur w/line
39 .8 .7 0 1.2 12 4 1 0 0 1 open bulls-eye
40 .0 .7 .7 1.2 12 4 2 120 -120 1 half moon bk lft w/line
41 .0 1 1 1. 3 0 1 30 0 1 solid black triangle
42 .8 .7 .7 1.2 3 3 1 90 0 1 hollow blk triangle w/line
43  .0 1 1 1. 3 0 1 90 0 1 solid black triangle
44 .8 .7 .7 1.2 4 4 1 0 0 1 diamond w/line
45 .8 1 1 1. 4 0 1 0 0 1 diamond
46 .0 .7 .7 1.2 4 4 1 0 0 1 solid diamond w/line
47 .0 .7 .7 1.2 6 6 4 0 0 1 hex moon bk ul w/line
48 .0 .7 .7 1.2 6 6 4 180 0 1 hex moon bk ul w/line
```

sym #

Use to number(label) each symbols algorithm. This is the same

number used in the last column of the APDV data file.

Rmin, Rmax, Lmin, and Lmax

These values determine the size of the three possible shapes used to create each symbol. The center point is at 0.0 and the outer edge of the polygons is at 1.0. The x/y lines can start at the center(0.0) or at any other position within the polygon. They can also be extended beyond 1.0 to a position of 1.7.

Rmin

Sets the minimum radius of the inside of the second polygon. With a setting of 0.0 the inside is fully minimized thus creating a solid polygon from the center out to Rmax. A setting of 0.8 will create a solid polygon, with an empty center, out to Rmax.

Rmax

Sets the maximum radius of the outside of the second polygon. A setting of 1.0, places the outside edge directly over the outside edge of the first, fixed polygon. A setting of 0.2 and a Rmin setting of 0.0 creates a small solid polygon centered in the middle of the first polygon.

Lmin

Sets the starting point for the x/y lines. 0.0 starts the lines from the center of the polygons. 1.0 starts the lines at the outer edge of the polygons.

Lmax

Determines how far the lines will extend from Lmin. If Lmax and Lmin equal 1.0 then no lines will be displayed. If Lmin is 0.0 and Lmax is 1.7 the lines will extend from the center past the outer edge of the polygons.

#face

This value determines the number of faces both polygons will display. A value of 12 displays a convincing circle.

#line

This value determines the number of lines.

bw

This parameter allows you to divide the second polygon into alternating light/dark solids with a x/y axis.

Valid values are 1, 2 and 4.

1 = full solid

2 = half solid

3 = alternating quarter solids

rot

Sets the rotation of the symbol in degrees.

lrot

Sets the rotation of the lines relative to the symbol in degrees.

rvrs

Use this parameter to reverse the symbols colors. A value of 0 is normally used but a value of 1 will reverse the colors.

name

an optional description of each symbol. This is only used for reference within the SYM file.

Sample Module Networks

The sample network shown below reads a GEO formatted data file, and a SYM formatted algorithm file. The output is displayed by the geometry viewer.

Symbols

|

|

EVS Viewer

A test geology file is included in the evs\special directory called TEST_SYM.GEO. It displays all 48 of the default symobls defined in the file shown above. The symbols are oriented starting at the lower left hand corner and going left to right and bottom to top.

Fence_Geology

General Module Function

The Fence Geology module uses data in specially formatted .geo files to model the surfaces of geologic layers in vertical planes, or cross sections. Fence Geology essentially creates layers of quadrilateral (4 node) elements (in a vertical plane) in which each node (and element) is assigned to an individual geologic layer. The output of Fence_Geology is a data field, consisting of a 2D line with each layers elevation as nodal data elements, that can be sent to the Krig Fence and 3D Geology Map modules where the quadrilateral elements are connected to the element nodes in adjacent geologic surfaces to create layers along the fence.

Module Input Ports

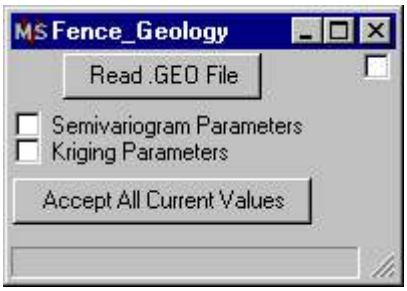
The Fence Geology module (shown above) has two input ports:

1. The first input port (leftmost) is the input .geo file.
2. The second input port (Available only in EVS-Pro) is for a data field consisting of line data. For example the output of Click Sketch or Read_Lines. This allows the user to define a Fence that is not drawn from boring to boring.

Module Output Ports

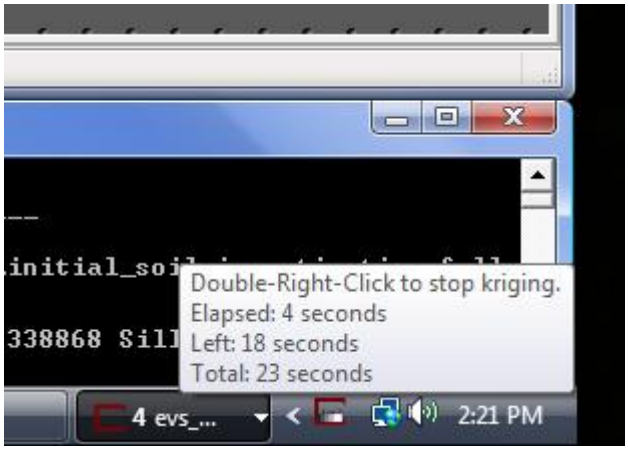
Fence Geology has three output ports.

1. The left port is the filename.
2. (brown-grey-green/brown-yellow/brown) : Provides geologic material information for the Legend module.
3. The right port is a data field consisting of a 2D line with layer elevation at each node.



***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.

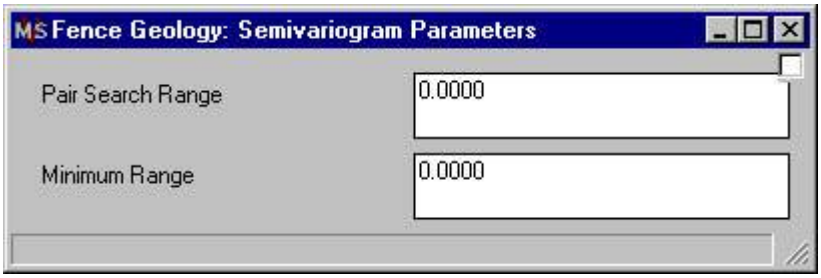


Module Control Panel

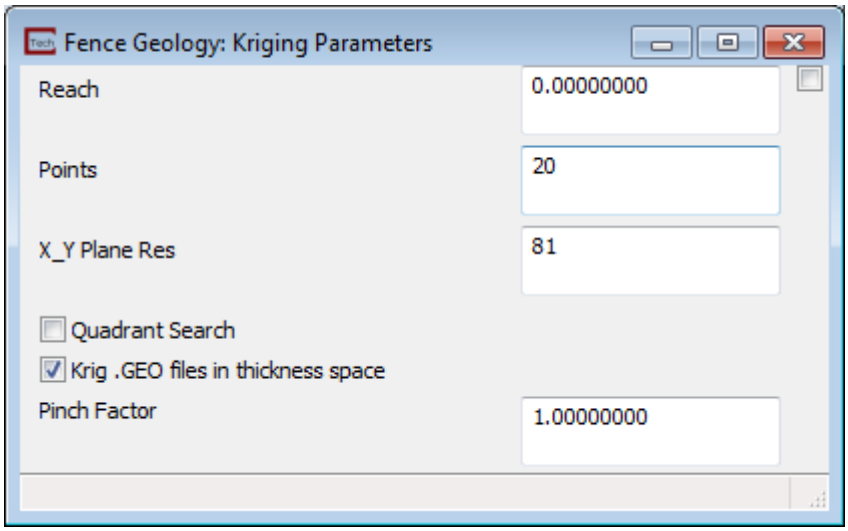
The control panel for Fence_Geology is shown above. Input files are read into the module using the Fence .GEO File button, which presents a standard windows style file browser. The general format of .geo files is discussed in detail in the geo file format section of the help system. Fence Geology input files contain only those borings that the user wishes to include for an individual cross section of the fence, in the order that they will be connected along the section. Each instance of Fence Geology produces one cross section, so that building a fence diagram requires instancing as many Fence Geology (and Krig Fence) modules as there are individual sections in the diagram. The order of the boring listings determines the connectivity of the fence diagram, and must match the order of the borings in the associated analyte (e.g. chemistry) file used in Krig Fence. The data for the boring(s) at which individual sections will be joined to produce the fence diagram are included in each of the .geo files that will intersect. The Fence Geology module begins running when the user clicks on the Accept All Current Values button.

Module Parameter Subpanels

Fence Geology has two subpanels, which allow the user to set the parameters used for the semivariogram production and execution of kriging. Clicking on either the check boxes next to the subpanel names or on the names themselves will bring up the subpanel data entry screens. Note that the subpanels can only be closed by clicking on the box in the **Module Control Panel**, (they cannot be closed by clicking on the subpanel's window control icon).



The *Semivariogram Parameters* subpanel is shown in the above figure. . The pair search range specifies the radial distance from any input data point that will be searched to assemble the data pairs that are used in the variance analysis. The values in the data windows are changed by clicking in the window and using standard windows style editing procedures. The default value for the pair search range is set to 0, which if left alone, results in the value being set to approximately 2/3 of the largest distance between data points in the data set. The user must consider the spatial characteristics of the data set when setting or revising the default calculated Pair Search Range. If large areas exist in the data domain that do not have data points within them, the user must set the Pair Search Range to a value that will allow a pair of data points to be identified, if these outlying data are going to affect the characteristics of the semivariogram. Data sets with large variations over short distances can be modeled most accurately using smaller pair search ranges, as this effectively limits the distance over which the semivariogram will search for and include data points.



The Kriging Parameters subpanel is shown in the above figure. The *Reach* input field defines the radial distance in the plane defined by the cross section (in user units) from any given model node that the kriging module will look for data points to be included in the estimation of the model parameter at that node. The default value of reach is 0, which results in the module calculating a reach value which is approximately two-thirds of the longest distance between any two data points in the data set. Note that the reach must be set to a value which allows the kriging algorithm to find at least one point, or else kriging of the property value cannot be completed at that model node.

The ***Points*** parameter defines the maximum number of data points (within the specified reach) that will be considered for the parameter estimation at a model node. The default value for points is 20, which generally provides reasonably smooth modeled parameter distributions. The effects of decreasing and increasing the values for reach and points on the model output are somewhat similar, but for different reasons. If the data have a fairly even spatial distribution throughout the domain, then increasing these values will generally include more of the input data points that will be used to krige the value for a given model node, and thus will result in smoother modeled data distributions. Decreasing the values of reach and points (in an evenly distributed data set) results in fewer input data points being used to calculate the parameter estimates at a given model node, and result in modeled distributions with greater variations across smaller areas.

The user should consider both the spatial distribution and the range of values in the input data set when deciding upon values for the reach and points parameters. If the specified kriging reach is too small to allow the kriging module to locate at least one point within the search area, then kriging will not be completed for that node, the property value will be set to 0, and the confidence value will be set to 0.1%. Note that this predicted property value could represent an inappropriate value. Therefore, the user should **always** examine the kriging confidence distribution to evaluate whether the kriging is producing reliable estimates in all areas of the model domain.

If the user specifies a large number of points (that are within the specified reach), then the output will be smoother, but the execution time for the kriging can increase significantly. By posting the input data using the Map Spheres module, and looking at the characteristics of the resulting kriged data using the plume_shell and Statistics modules, the user can quickly analyze the characteristics and distribution of the kriging output for a given set of parameters, and test the effects of changing the kriging parameter values.

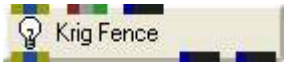
The ***X_Y Plane Res*** parameters specify the number of grid nodes that will be included along the arc length of the cross section (or fence). The number of grid elements along the arc length of the model is simply the X _Y Plane Res value minus one, as every element has two bounding nodes. The default value for this parameter is 51, but the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user. The robust kriging algorithms in EVS generally produce reasonable modeled distributions with a fewer number of grid nodes than the user may be used to, so the recommended procedure for setting the X_Y Plane Res parameter is to start with less, and then increase the value until an acceptable model is obtained.

The ***Quadrant Search*** toggle changes the method by which data sample points are selected for inclusion in the kriging matrix. If this is on, the "Points" parameter switches to "Max Points in Quadrant". Searching is performed for each of the four quadrants surrounding the point to be kriged. Within each quadrant a maximum number of points (up to one-half of the total points) are selected. Then, points are taken sequentially from each quadrant up to the maximum number of total points or until all quadrant's points have been used. The panel display changes when this option is selected as shown above.

The ***Krig .GEO files in thickness space*** toggle off causes GEO files to be kriged like GMF files. Each surface get kriged independently of the other surface instead of being kriged in thickness space. This only applies to GEO files without the $W/$G flags.

The ***Pinch Factor*** parameter provides the ability to control where pinching occurs between positive thicknesses and borings having the pinch flag. It defaults to 1.0 which causes pinching to occur approximately half-way between positive thicknesses and borings having the pinch flag. When older applications created before version 9.5 are loaded they will have a value of 0.0 for backwards compatibility.

Krig_Fence



General Module Function

Krig Fence models parameter distributions within domains defined by the boundaries of the input data in 3D Fence sections which can "snake" around in the x-y plane and are parallel to the z-axis. Krig_Fence can also receive the geologic system modeled by Fence Geology. It creates a quadrilateral

finite-element grid with kriged nodal values of any scalar property and its kriged confidence level, and outputs a geometry whose elements can be rendered to view the color scaled parameter distribution on the element surfaces. Krig Fence provides several convenient options for pre- and post-processing the input parameter values, and allows the user to consider anisotropy in the medium containing the property.

Module Input Ports

1. Read__data_file (Yellow/Blue/Yellow): This port is the filename (see above).
2. fld_in (Brown/Grey/Green): This port can only accept output **only** from Fence Geology.
3. external_data (Blue/Black): (Available only in MVS)This port allows external data to be imported and kriged to the created fence grid.

Module Output Ports

1. Read__data_file (Yellow/Blue/Yellow): This port is the filename.
2. ucd_out (Blue/Black): This port outputs a data field for a cross section, which can be input to the Fence Merge module. The Fence Merge module accepts data fields from multiple Krig Fence modules, and merges them to produce a fence diagram.
3. out1 (Blue/Black): This port outputs a typical data field which can be input to external faces (and/or any of the Subsetting and Processing modules which have the same color port) to produce a geometry of a single cross section.

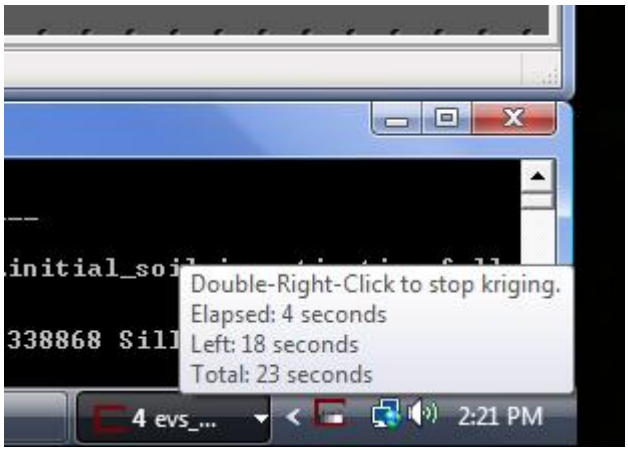
Selecting Semivariogram Parameters in the Kriging Paramaters window, and then selecting Plot Semivariogram will cause an additional three ports to appear as shown below.

1. out2 (Red/Grey): This port can be connected to the viewer, and will output a renderable geometry of the calculated semivariogram.
2. VG_fld_out (Blue/Black): This port will output a field consisting of the 3D lines that make up the variogram, this is generally passed to the Generate Axes module.
3. VG_scale_fact (Grey/Brown): This port will output the variogram scale. This is also generally used in the Generate Axes module.

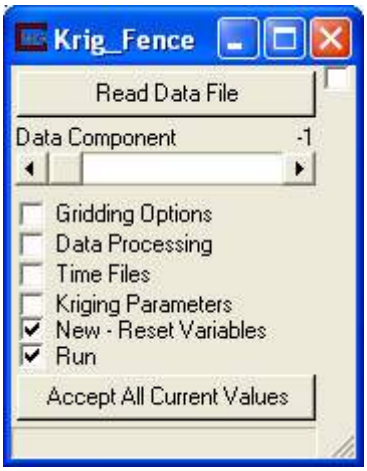
***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-

Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel



The control panel of Krig Fence is shown in the figure above. The **Read Data File** button opens a *File Browser* as described for Krig_3D_Geology, but which lists the *.apdv, *.aidv, and *.vdf files that are present in the current directory shown in the directory window. The format of .apdv files is described in the apdv file format help topic.

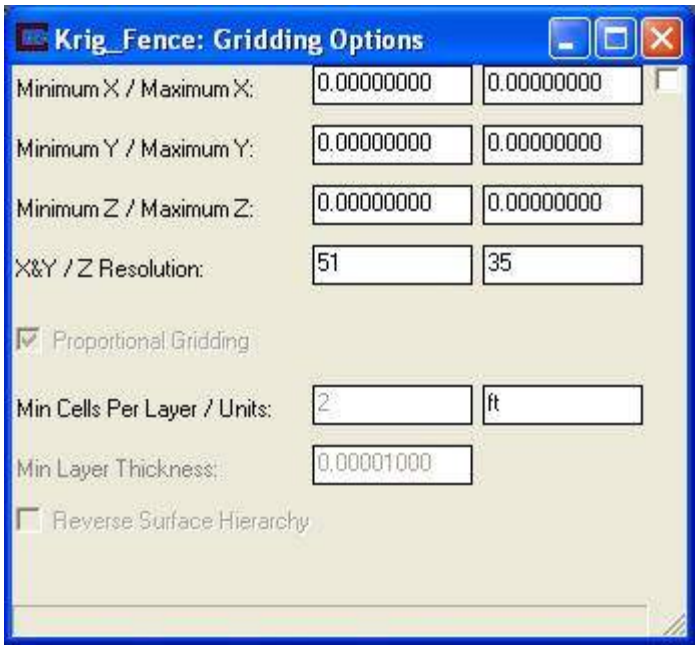
One instance of Krig Fence reading one .apdv file is required for each cross section that will be included in the fence diagram. The data for borings at which the fences will intersect are included in each of the intersecting cross section files

The **Data Component** slider allows the user to select which of the property values in the .apdv file will be kriged by execution of the module. The default value is 0, which corresponds to the first property included in the .apdv file.

Krig Fence will not begin running until a valid file has been selected and the **Accept All Current Values** button is pushed.

Module Parameter Subpanels

Krig Fence has four subpanels, which allow the user to set the parameters used for processing the input data, producing the semivariogram, and executing kriging. Clicking on either the check boxes next to the subpanel names, or on the names themselves, will bring up the subpanel parameter screens. Note that the subpanels can be closed only by clicking on the box in the **Module Control Panel**, and **not** by double clicking on the subpanel's window control icon (the small horizontal bar in the upper left corner of the window).



The ***Gridding Options*** subpanel is shown in the figure above. It is used to specify all parameters that affect the grid exported from Krig_Fence. The availability of many options depends on whether geologic input is available or if the grid is created totally within Krig_Fence. The window above shows the options with no geologic input.

The ***Minimum X, Maximum X, Minimum Y, Maximum Y, Minimum Z and Maximum Z*** parameters allow the user to define the horizontal and vertical domain within the data set in which kriging of the parameter distribution will be completed. A value of 0 is the default for these parameters, which results in a model domain that is defined by the rectilinear bounds of the entire data set when the module is run. Krig_Fence utilizes a model domain that is bounded by the limits of the data set, unless the user specifies a different domain by setting the Min and Max Values for X, Y, and Z, or it is passed a model (or finite-difference) domain from Fence_Geology. Utilizing the default extents effectively minimizes the extrapolation of parameters within the model to that area which is enclosed by the measured data points. Note that when a geologic model domain is passed to Krig_Fence, the kriging domain is restricted to that domain regardless of what X, Y, or Z values are set in the Krig_Fence inputs. If the user is uncertain of the X, Y and Z limits of the data domain, the module should be run with the default 0 values, and upon completion of execution, the values in the X, Y, and Z input fields will be the

min and max values of these parameters in the data set. The File_Statistics module can also be used to investigate the limits and distribution of values in the input data set, and the statistics module can be used to output the distribution of values in the kriged model.

The ***X & Y / Z Resolution*** parameters specify the number of grid nodes that will be included within the model domain. The number of grid elements along any axis of the model is simply the axis Res value minus one, as every element has two bounding nodes along an axis. The default value for the X and Y Res parameters is 51, and the default value for the Z Res is 35. However, the user can specify any number desired, up to the limit of available memory resources in the computer and run time limitations imposed by the patience of the user.

If geologic layers are being passed to Krig_Fence, then the number of nodes specified in Z Res will be distributed over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain.

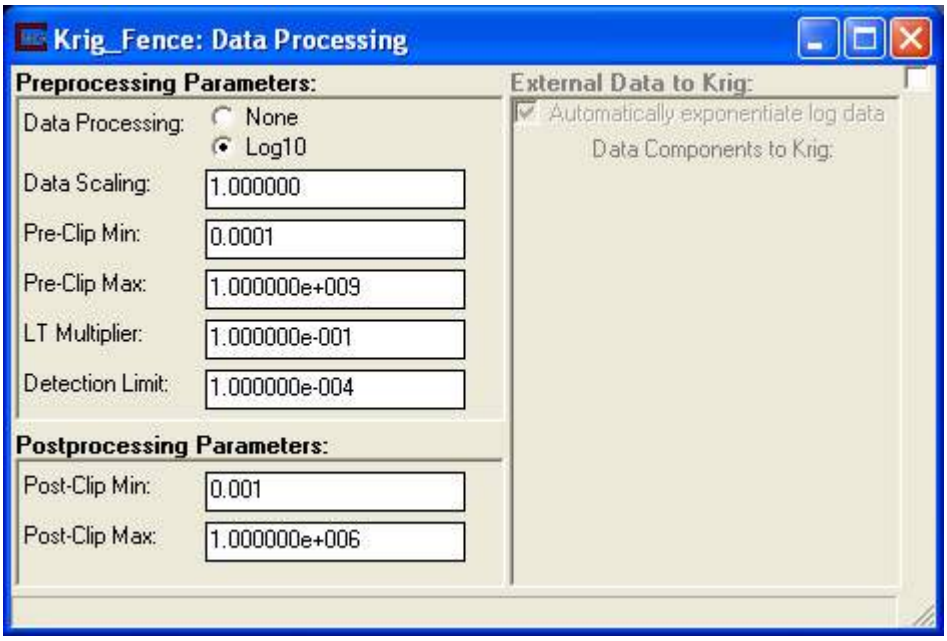
NOTE: If Proportional Gridding is toggled off then only Min Cells per Layer is enabled. In this case, only the Min Cells per Layer input will be used for each layer of the model.

Proportional Gridding - A toggle for activating the same algorithm for apportioning nodes in the Z direction as used in Krig_3D. The number of nodes specified for the Z Resolution will be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) of the Krig_3D domain will be placed in each geologic layer.

Min Cells per layer - A Type in for establishing a minimum number of cells (in the Z direction) per layer. The default is 2 cells which results in 3 nodes.

Reverse Surface Hierarchy – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

The ***Rectilinear Offset*** parameter is used to create an automatically generated rectilinear domain which is larger than the input data set by a user input percentage along all three axes. If 10% (0.1 default value) is used, the limits will be offset by 5% on all 6 sides of the rectilinear volume.



The ***Data Preprocessing*** parameters subpanel is shown in the figure above. It is divided into two groups which are preprocessing and postprocessing.

It is important to note that all preprocessing actions are applied directly to the data in memory, and that the original data file is not altered. However, all of the functions within Krig_Fence and the modules downstream of Krig_Fence will be using the preprocessed (and kriged) parameter distribution. The user should refer back to the preprocessing subpanel when setting values for the filtering and display of the kriged data (i.e., to correctly specify whether the data has been log transformed, scaled, and/or clipped).

The ***Data Processing*** radio buttons allow the user to specify whether the data will be used as is, or will be processed to compute the log (base 10) of the parameter value before kriging. Note that if the log10 of the data is taken, a Clip Min value (which must be greater than 0) must be used to replace values in the data below the specified minimum value because the log function is undefined for values equal or less than zero.

The ***Data Scaling*** input field is used to specify a value by which all nodal data values will be multiplied before kriging. The default value is 1, but the user can specify any negative or positive value. This option is most commonly used to convert the units of the property being kriged, such as to convert concentrations in ppb to ppm or visa-versa. The user should bear in mind that any scaling of the data that is completed in Krig_Fence will affect all downstream modules.

The number entered into the ***Pre-Clip Min*** input field will be used during preprocessing to replace any nodal property value that is less than the specified number. When log processing is being used, the value of Clip Min must be a positive, non-zero value. Generally, Clip Min should be set to a value that is one-half to one-tenth of the lowest detection limit in the data set, unless the user wishes to make the influence of not detected values stronger. As an example, if the lowest detection limit is 0.1 (which is present

in the data set as a 0), and the user sets Clip Min to 0.0001, the clipped non-detected values forces three orders of magnitude to be present between any detected value and the non-detected values.

The number entered into the ***Pre-Clip Max*** input field will be used during preprocessing to replace any nodal property value that is greater than the specified number. The clipping values can be used to lessen the importance placed on extremes of the data, or outlier data values, before kriging. The preprocessing functions can be used in various ways to investigate the kriging confidence levels within specified ranges or limits of the data sets. An example of this type of analysis is provided in the Sample Networks section.

The ***LT Multiplier*** value affects any file value with a preceeding "<" character. It will multiply these values by the set value.

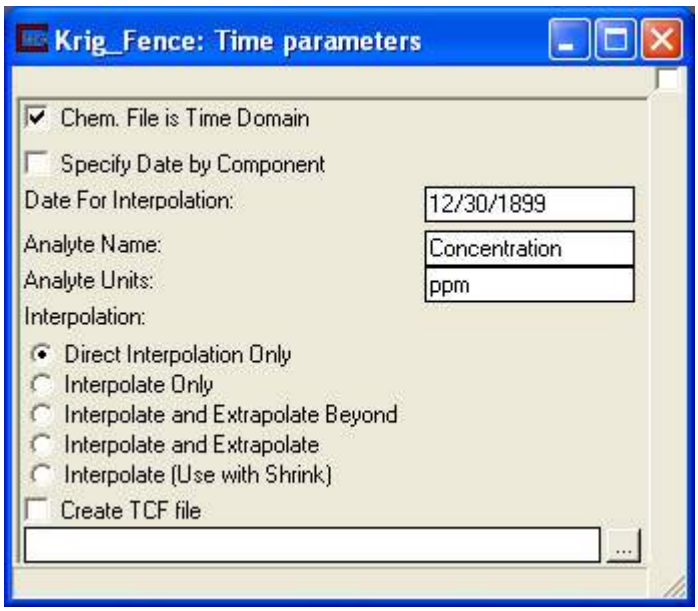
The ***Detection Limit*** value affects any file values set with the "ND" or other non-detect flags (for a list of these flags open the help for the APDV file format). When the module encounters this flag in the file it will insert the a value equal to (Detection Limit * LT Multiplier).

Postprocessing of the data from Krig_Fence affects the calculation of the kriging confidence and uncertainty estimates that are produced by the model, and allows the user to apply a filter for the data passed to all modules downstream of Krig_Fence. Note that the postprocessing **does not** affect any of the other Semivariogram or kriging algorithms that execute in Krig_Fence, so the user can experiment with different values of postprocessing parameters to obtain the type of display desired, independent of the internal kriging process.

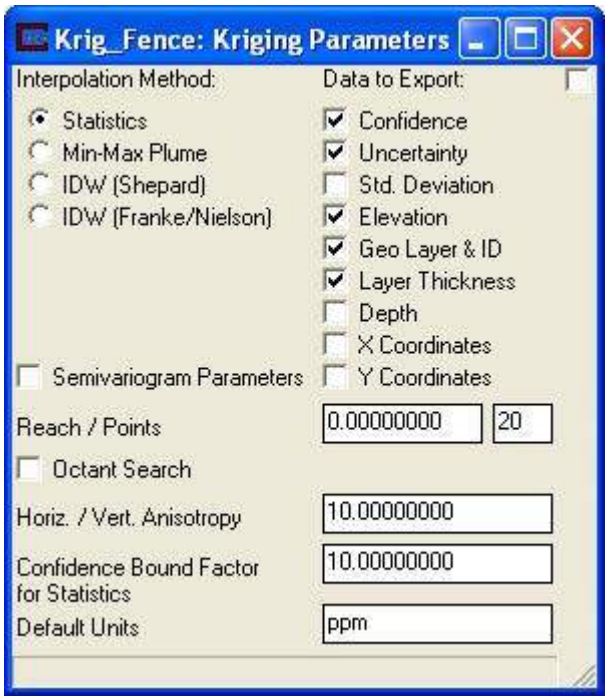
The ***Post-Clip Min*** parameter specifies the smallest nodal value that will be present in the data field output by Krig_Fence. This parameter is useful for limiting or enhancing the effects of not detected values or outliers in a data set, and for optimizing the use of the dynamic color range used to represent the property distribution. Clip Min has a default value of 0.001, but can be set to any negative or positive value with magnitude from -1.0 E09 to 1.0 E09. In general, good results are obtained by setting this value to the lowest property value or detection limit in the input data set. It is important to note that because not detected values are represented in the .apdv file as zeros, the Clip Min value can be used to strengthen the influence of small values or non detects that are present in the data set. As an example, if the detection limit for a certain chemical analysis is 0.1 (which is entered into the .apdv file as 0), and the Clip Min is set to 0.0001, then this not detected value will have an effective influence on the kriged distribution near this data point that is three orders of magnitude stronger than the actual detection limit. The user should bear in mind that if mass or volume estimates are being made in the analysis, the Clip Min value could affect the estimates if considerable data are present that are being clipped. For volume estimates, if the specified lower bound of the concentrations of interest is well above the Clip Min value, then their will be no effects on the results. If the specified lower bound of concentrations for the volume estimate is lower than Clip Min, then the clipping will truncate the volume at the Clip Min value, and the estimated volume will be from the entire model domain.

The ***Post-Clip Max*** parameter specifies the largest nodal value that will be present in the data field output by Krig_Fence. Again this parameter can be used to optimize the use of the dynamic color range when a data set has a few extremely high values, but the user is most interested in seeing the detailed changes in the distribution in some lower range of the data. It can also be used to investigate the kriging confidence or uncertainty levels near certain threshold values of a property. As an example, if the regulatory threshold for a parameter is 10 units, then the Clip Max parameter can be set to 10, and all kriged values greater than 10 will have the same strength in the calculation of the uncertainty levels in the kriged distribution. Clip Max has a default value of 1,000,000, but can be set to any negative or positive value with magnitude form -1.00E09 to 1.00E09.

When external data is being imported into the module the External Data to Krig window becomes active. This window allows the user to select which data components to krige by switching on or off the toggles associated with that data component. The ***Automatically exponentiate log data*** toggle will detect if the data being imported has been log processed and if it has exponentiate it. This avoids double log processing the data with the Preprocessing option.



The Krig_Fence Time Options window is the same as that of Krig_3D



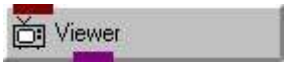
The Krig_Fence Kriging Parameters window is the same as that of Krig_3D

Further explanations of the use of Krig Fence are presented in Workbook5 .

Display Modules

Display Modules

Viewer



- File Pull-Down Menu
- Editors Pull-Down Menu
- RNC Pull-Down Menu
- Az-El Pull-Down Menu
- Instance Pull-Down Menu
- ModulesPull-Down Menu

General Module Function

The Viewer Module accepts renderable geometries from a broad variety of other EVS modules, renders the objects, and then outputs the display into a window that allows transformations and scaling of the objects using the mouse. The viewer tracks each object that gets sent to it individually or by groups of similar entities within an object, and organizes them into a parent\child hierarchy that allows direct assignment of properties to each object individually, or by inheritance from parent objects. There are also multi-window and single window viewers available. These viewers are not supplied by default with the EVS Viewer. This help section will describe the

EVS viewer in detail. There are many options available to control the rendering and manipulation of objects in the viewer. An introduction to using the viewer is presented in some sections of Workbook 1 Mouse Transformation

Module Input Ports

The Viewer module is shown in the figure above, and has only one input port. This red port accepts input from all EVS modules that output renderable geometry.

Module Output Ports

The Viewer has one (magenta or purple) output port, which can be connected to many modules which process the full content of the Viewer. These include Output_Images, write_VRML, Record_4DIM, axes, etc. This port establishes a special type of connection between some modules like Output_Images and the Viewer which allows a virtual viewer to be established that can output an image at any user specified resolution.

Shortcuts

There are a few keyboard shortcuts worth noting. These will give you quick control over the player.

- CTRL-F sets the player to FULL SCREEN mode. This is not equivalent to the maximize button in the upper right corner since this removes the normal borders.
- With the player as the active window, ESC(ape) exits Full Screen mode
- CTRL-H toggles the Auto-Hide mode.

Object Manipulation in the Viewer

When the Viewer is instanced, it opens a window in which objects connected to the viewer are rendered and can be manipulated. Objects can be transformed and scaled in the Viewer window by using combinations of mouse actions and various keys on the keyboard.

- Rotation of objects in the viewer is accomplished by either:
  A. Clicking and dragging on any portion of the Viewer window with the left mouse button.
  B. *While the Viewer is the active window:* After hitting the "R" key on the keyboard (not needed the first time, since this mode is default) the arrow keys ???? or the 4, 8, 6 & 2 keys on the NUM pad allow you to perform rotations about the Viewer's x & y axes similar to fundamental mouse rotations. The NUM pad keys 1, 7, 9 & 3 perform rotations about the diagonal axes.
- Translation of an object in the x/y plane can be accomplished by:
  A. Clicking and dragging on any portion of the Viewer window with the right mouse button.
  B. *While the Viewer is the active window:* After hitting the "T" key on the keyboard the arrow keys ???? or the 4, 8, 6 & 2 keys on the

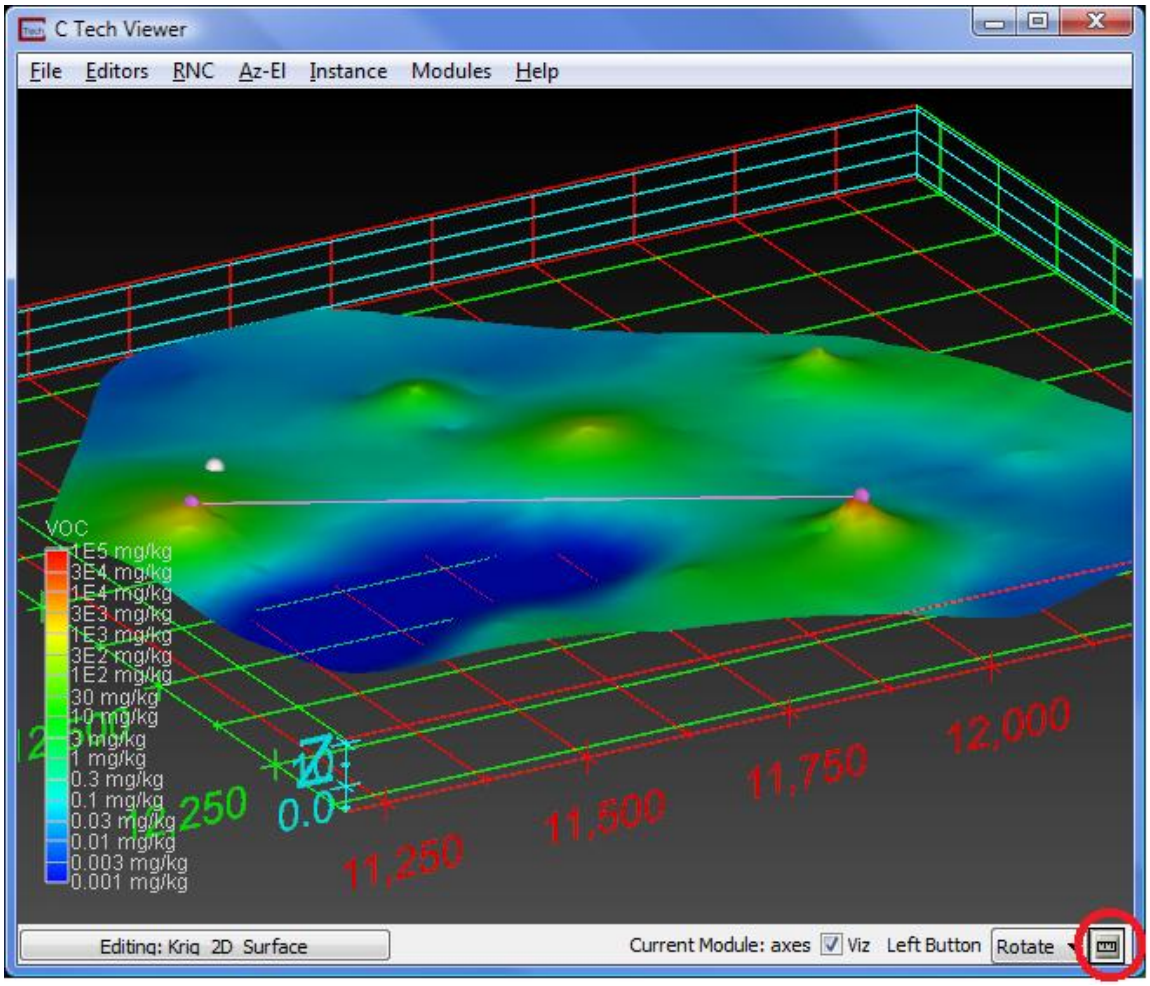
NUM pad allow you to perform translations of your model in the Viewer's x & y axes similar to fundamental mouse translations. NUM pad keys 1, 7, 9 & 3 perform diagonal translations.

- Zooming of an object in the viewer is accomplished by:
  A. Depressing the Shift button while clicking and dragging the left-mouse towards the upper right corner of the viewer window (zoom in), or towards the lower left corner of the viewer (zoom-out).
  B. *While the Viewer is the active window:* On mice with a wheel, depressing the wheel as a button and dragging towards the upper right corner of the viewer window (zoom in), or towards the lower left corner of the viewer (zoom-out).
  C. *While the Viewer is the active window:* On mice with a wheel, rotation of the wheel in a forward direction zooms in and backwards zooms out.
  D. *While the Viewer is the active window:* On the NUM pad, the "+" (plus) key zooms in and the "-" (minus) key zooms out. The "+/=" (plus-equals) and "_/-" (underscore-minus) keys on the main keyboard also zoom.

The user should work with the exercises in Workbook 1 to become comfortable with manipulating objects in the Viewer with mouse and key actions.

Distance Computations in the Viewer

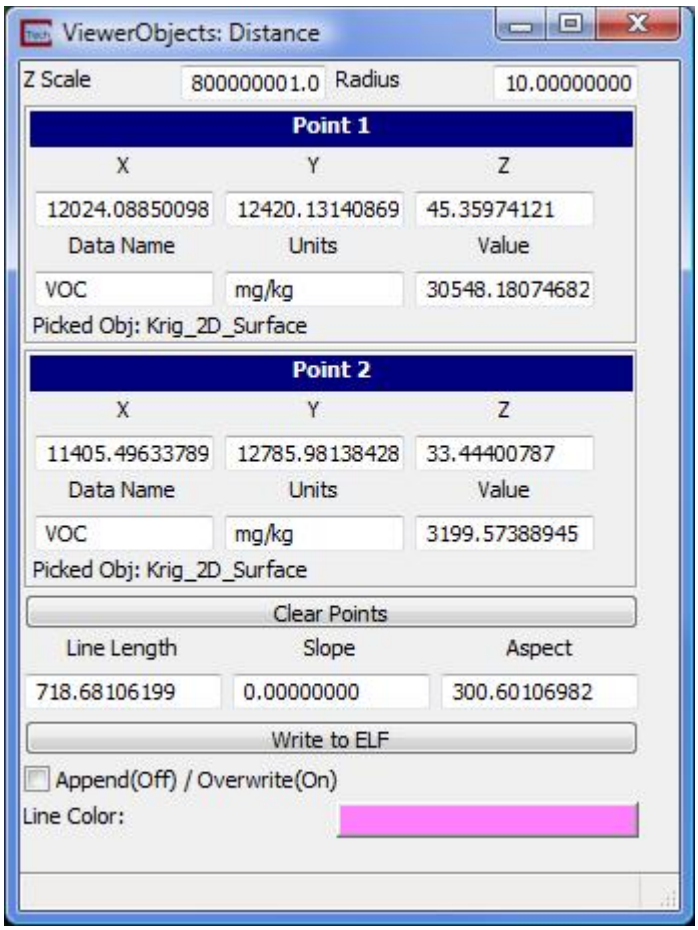
The Euclidean distance between two points in the Viewer can easily be determined. First press the button in the lower right corner of the Viewer (shown below circled in **RED**).



This will open a new panel that allows you to specify the Z Scale of the model in the viewer (this is not automatically set for you) and the radius of spheres that will be placed at each probed location. Distances are determined by using ALT+Left Mouse probing. The first point probed is the master point and each additional point probed after that will become the secondary point for distance computation.

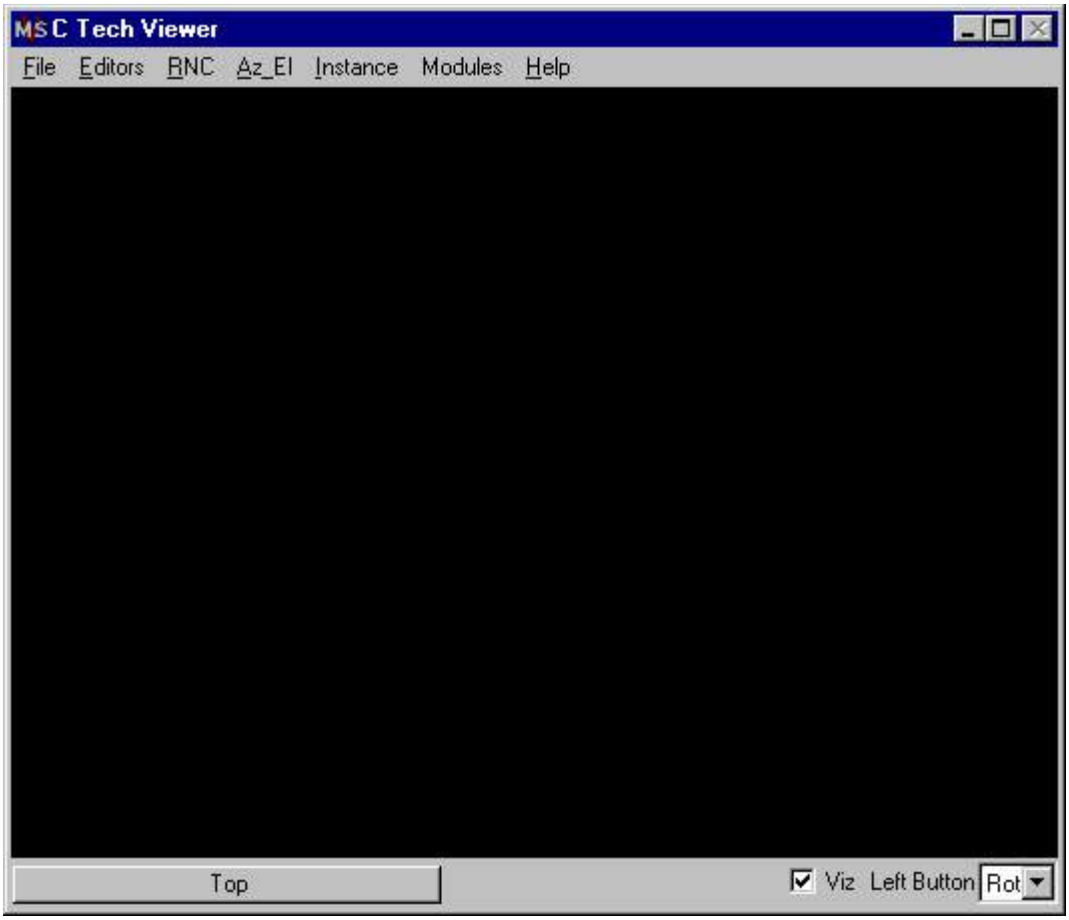
To begin again with a new master point, use the "Clear Points" button.

When two points are probed, spheres will be place at each point and a purple (you can change this color) line will connect the points. The **Distance** panel displays the coordinates and data for each point and the **Line Length** (distance) between the points as well as the **Slope** and **Aspect** of the line.



The line segment can be written to an ELF file and you may append or overwrite this file (append allows you to create a file with many line segments).

Viewer Module Control Menus



Above is an empty Viewer. The Modules pull-down menu is located on the Viewer and the main window of EVS or MVS.

Important Knowledge!: Many of the Viewer settings involve selecting a specific object in the viewer (Examples: making isolines thicker, coloring map lines, changing datamapping for plume_shell, etc.). This requires following the steps below for selecting objects in the Viewer.

Select Object

Note that most of the operations listed below will effect only the selected object and those objects inheriting the settings. This feature is NOT under the editors pull-down, but instead is located in the bottom lefthand corner of the viewer. The button in the bottom left corner of the Viewer interface is named "Top" by default, which reflects the selected parant object (All Objects). Clicking this All Objects button (All Objects) activates the dialog box for selecting the object of interest from a list of objects in the view. The process for selecting objects in the Viewer generally requires:

Click the All Objects button.

Click on the desired object in the selector list of Viewer objects, then click Apply. The selected object's name should appear within the button, thus replacing the word "Top".

Adjust the attributes of the selected object (for example, change it's opacity setting).

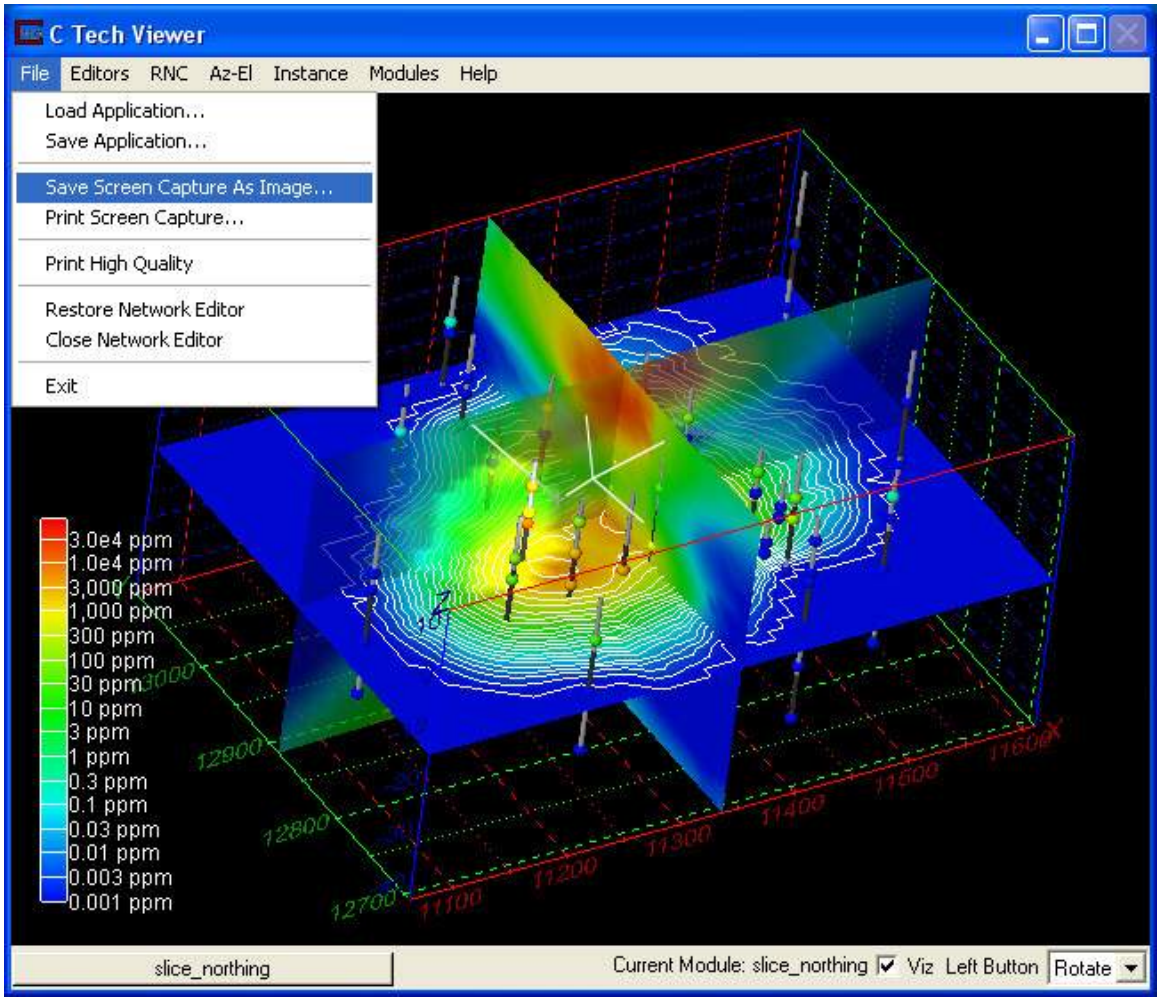
Go back to the Object Selector list and click on Top, then click OK. Now, the word "Top" should appear on the face of the button again.

Note Selecting ojects can also be performed with the mouse by pressing the Alt key, then pointing to the object of interest in the Viewer window and clicking the left mouse. This selects the object, but to de-select the object you press the Alt key and click anywhere outside the rendered objects, in the "blank" area of your Viewer window (this area is usually black assuming a black background). Hint: You'll know you have properly selected or de-selected your object by seeing the word "Top" (on the button face) switch to your object's name, and vice-versa.

The Viewer has four control menus which are located along the top of the viewer window as shown in the figure above. The menus can be accessed by either clicking on them with a mouse, or by depressing the alt key and the underlined first letter of each menu title.

File Menu

The first pull-down menu is "File" which includes the following items:



The first, second and last items are the same functionality as in the Network Editor (main window of EVS/MVS). These give you the ability to load and save applications, and exit the program. The "Restore and Close Network Editor" option provides a new ability to manage the windows on your desktop. With these, you can close the network editor and restore it.

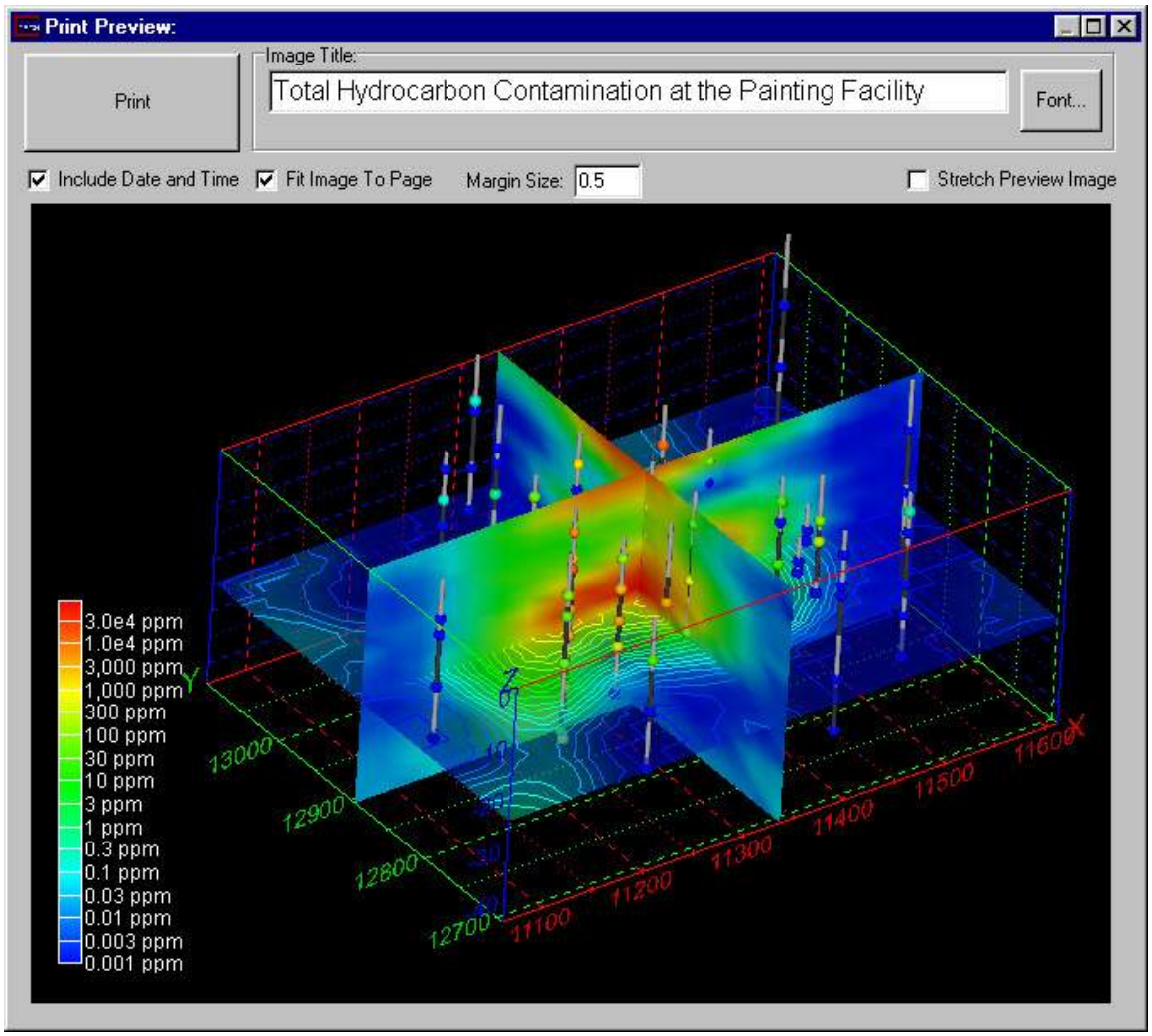
This capability was also added to accommodate use with ArcView (copyright ESRI) extensions which launch EVS/MVS. The default mode will be to automatically load an application with data queried from ArcView and the network editor will normally be hidden. Users will be able to load and save their applications (parameter settings, etc.) without the Network Editor, but will be able to access it to customize their applications (add or delete modules).

Save Screen Capture as Image provides a means to save the image in the viewer window (not the border) at whatever resolution (size) your viewer is at the time. Though you can always create high resolution images with Output_Images, this provides a very quick and easy way to save images that are often adequate resolution. When this option is selected you will be prompted for a file name. Supported file types are Windows .BMP (the most common uncompressed file format) and PNG, the highest compression *lossless* file format. A confirmation window will pop-up when the file writing is complete.

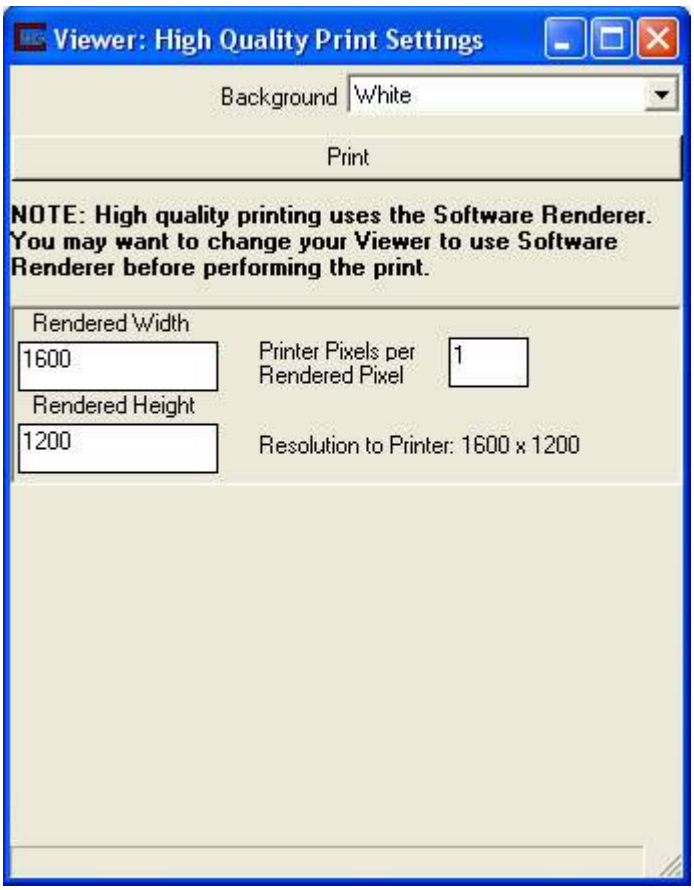
Print Screen Capture provides a means to print the image in the viewer window (not the border) at whatever resolution (size) your viewer is at the time. Though you can always print high resolution images with the Print Editor, this provides a very quick and easy way to save print the contents of the viewer **exactly as it appears** at a resolution that is often adequate. When this option is chosen, a window (as shown below) will appear.

You have the following options:

1. Adding an *Image Title* and specifying the font type, size and color.
2. Including the date and time
3. Fitting the Image to the Page, in order to have the image scaled to fill out to the margins (and additional buffer for the title).
4. Specifying Margin Size in inches.
5. *Stretch Preview Image*. This option **stretches** the preview image to fit the preview window. It is most useful for the preview in the Print Editor where the output resolution may be so large that you cannot see the entire image. It may (probably will) cause distortion of the image aspect ratio during preview, but does not affect the output.
6. The Print button opens a dialog allowing you to specify the printer that you wish to use. You may set additional printer settings such as "Landscape" that will affect the final size and orientation.



Print High Quality provides the ability to create a high quality print of your image at any desired resolution.



Print Quality Tips

The following provides hints and tips for obtaining the optimal quality from your printing device. This assumes you are using a color printer, but it is important to note that the user may print grayscale images with a black and white printer if desired. This would of course be best implemented by creating grayscale colormaps for the objects of interest (otherwise there will be ambiguous shades of gray that represent more than one color).

Printing: The following parameters and features are available:

Rendered Width and *Rendered Height* specify the resolution of the image to be printed and determine the image's aspect ratio (width divided by height). The width and height can be set to any (reasonable) value and may be set higher than the resolution of the current viewer window. Normally you will want to specify an aspect ratio that matches your current viewer.

Printer Pixels per Rendered Pixel is an integer type in the scales the image that is sent from the viewer to the printer. It doesn't change the resolution that the image is rendered to, but does scale up the image that the printer prints. This allows you to control the number of printer pixels (dots) that will represent each rendered image pixel. This is important if you want to have a specific printed image size and desire to achieve the highest level of print quality.

Note: If you use this, you should not use the *Fit Image to Page* option in the Print Preview window.

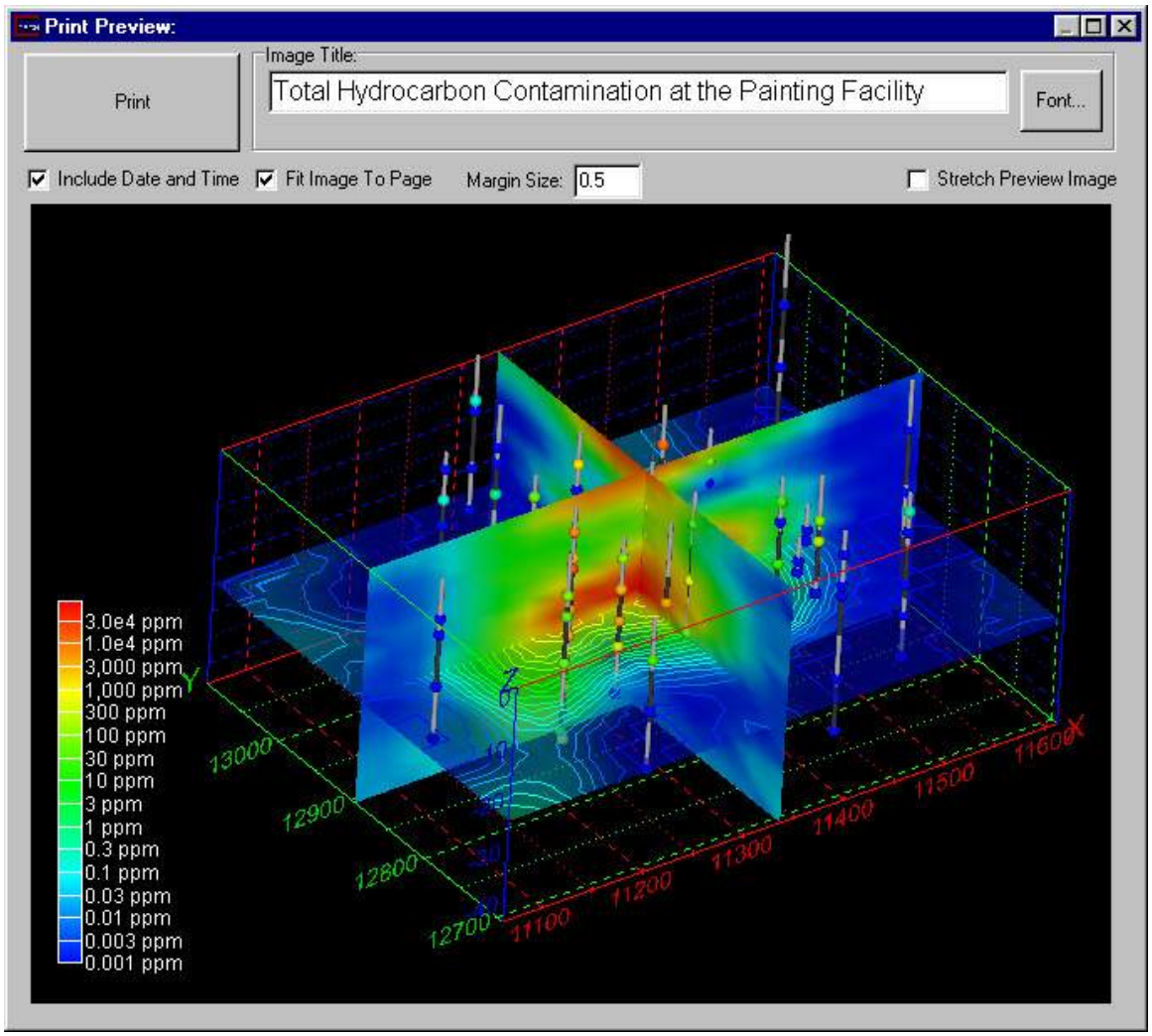
A detailed discussion of printing technology and specifications is available at the end of this topic.

The *Background* selector provides a choice of white, black or *keep* (to keep the current viewers background).

The *Print* button launches a sequence of processes that begins with rendering the current viewer objects at the specified resolution. When rendering is complete, the *Print Preview* window (as shown below) will appear.

In the *Print Preview* window the following options are available:

1. Adding an *Image Title* and specifying the font type, size and color.
2. Including the date and time as a label at the bottom of the page in the margin area
3. *Fit Image to Page* Fitting the Image to the Page, in order to have the image scaled to fill out to the margins (and additional buffer for the title).
4. Specifying Margin Size in inches.
5. *Stretch Preview Image*. This option **stretches** the preview image to fit the preview window. It is most useful for the preview in the Print Editor where the output resolution may be so large that you cannot see the entire image. It may (probably will) cause distortion of the image aspect ratio during preview, but does not affect the output.
6. The Print button opens a dialog allowing you to specify the printer that you wish to use. You may set additional printer settings such as "Landscape" that will affect the final size and orientation.



Detailed Discussion of Printing Technology

Color printers fall into three primary categories. They are: inkjet; color laser; and dye sublimation. EVS produces bitmaps which are continuous tone with 256 shades each of red, green and blue for a total of 16.7 million possible colors (256*256*256). Color printers must either produce continuous tones or approximate them using a pattern of primary colored pixels in an n-by-n grid.

Among these three printer categories there is considerable variation. Inkjet printers are (with a few exceptions) capable of producing one of only eight primary colors for each printer pixel (or dot). These colors are white, black, cyan, magenta, yellow, red, green and blue. Inkjets must therefore use a grid of primary colored pixels to approximate continuous tones. The larger the grid (4 by 4 vs. 2 by 2) the better the approximation. However, larger grids tend to create artifacts called jaggies, which are undesirable visually. The challenge is to balance the need for smoother color rendition with the desire to have higher resolutions.

Dye sublimation printers are at the other end of the spectrum. Their ability to reproduce continuous tones makes the task of choosing a resolution easy. A

typical dye-sub printer has a resolution of 300 dots per inch (dpi). If the intended size of the final printed image is 10 inches wide by 7 inches tall, then the optimal image size is 10*300 by 7*300 or 3000 x 2100 pixels. If quicker image creation and print times are desired, a compromise resolution would be exactly half or 1500 wide by 1050 high.

IMPORTANT NOTE: We always want to have an integer number of printer pixels for each "rendered" image pixel. When we reduce the image size by half, we get a 2-by-2 grid. The n-by-n grid concept applies to ALL TYPES OF PRINTERS. This "rule" is actually a guideline for best results. Other resolutions (non-integer ratios) create banding artifacts that are usually objectionable.

For inkjet printers you should always allow for at least a 2x2 grid and usually 3x3 to 5x5 gives the best results. For an EPSON printer with 720x1440-dpi resolution you should use the smaller resolution number (720) for your calculations. The printer will use the additional resolution to better approximate the colors.

Example: For an Epson with 720 dpi, to print an image 9 by 7.5 inches (landscape) we recommend that you start at a 4x4 grid which gives an effective printed resolution of 180 dpi. Your image width and height would therefore be:

width = 9.0 * 180 = 9.0 * (720/4) = 1620

height = 7.5 * 180 = 9.5 * (720/4) = 1350

Finally, color laser printers vary in their abilities to approximate continuous tones. This means that the rules to apply will be somewhere between dye-sub and inkjet properties.

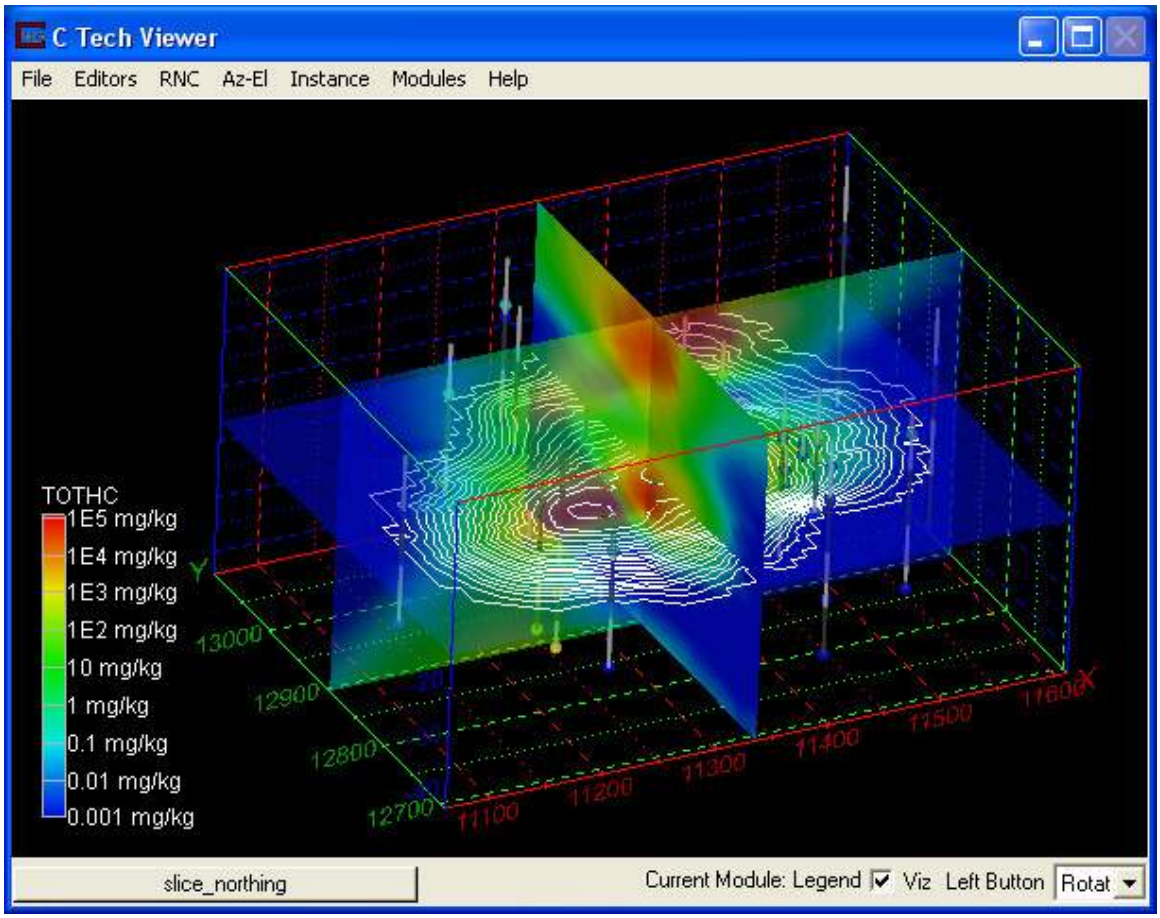
Editors Menu

The first two options in this menu are:

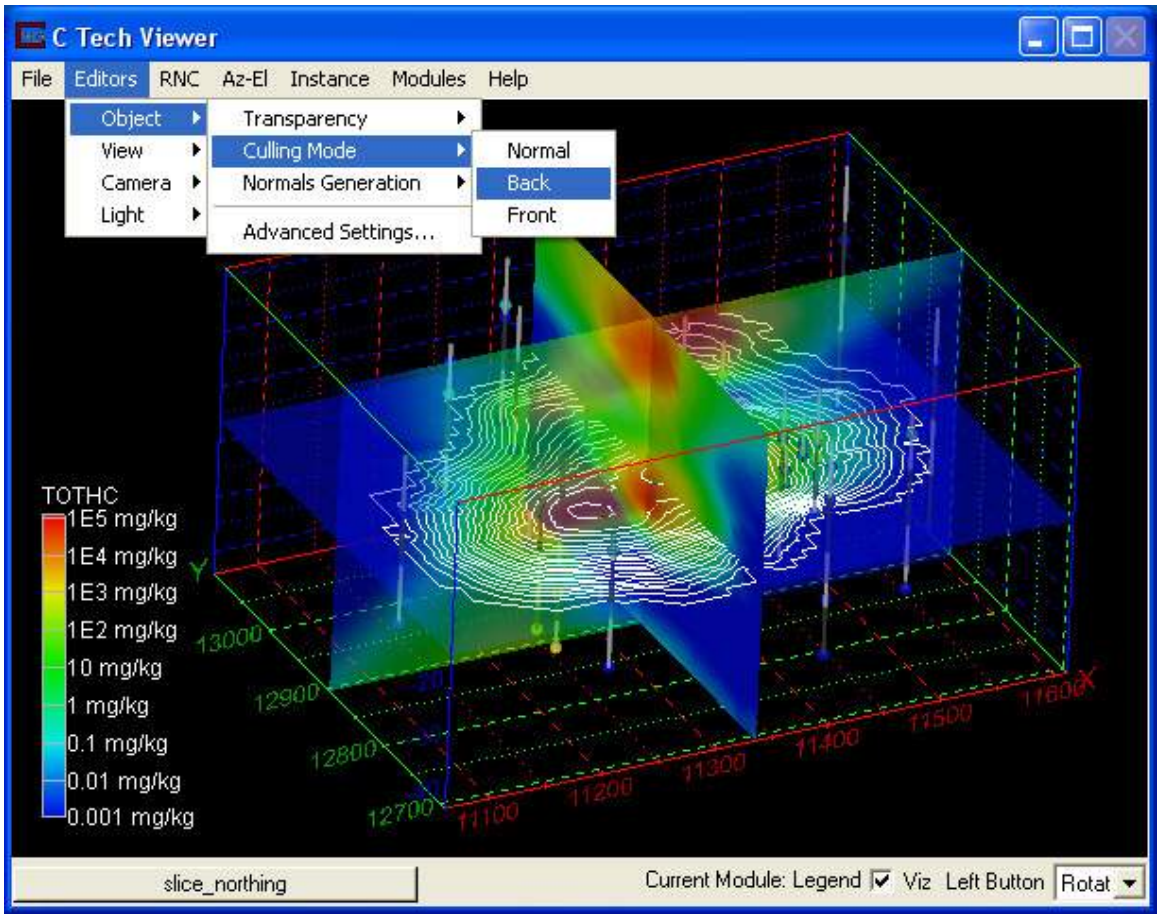
1. Full Screen: Creates a Viewer which has minimal borders and is the full size of your graphics screen. You can turn this on/off with CTRL+F or using the Editors Pull-down menu.
2. Auto-Hide Controls: Further increases the effective area of the viewer by suppressing all borders and titles until the mouse approaches the top or bottom of the screen. This is great for presentations.

Object Editor Functions
BackgroundEditor Functions
View Editor Functions
Camera EditorFunctions
Light EditorFunctions

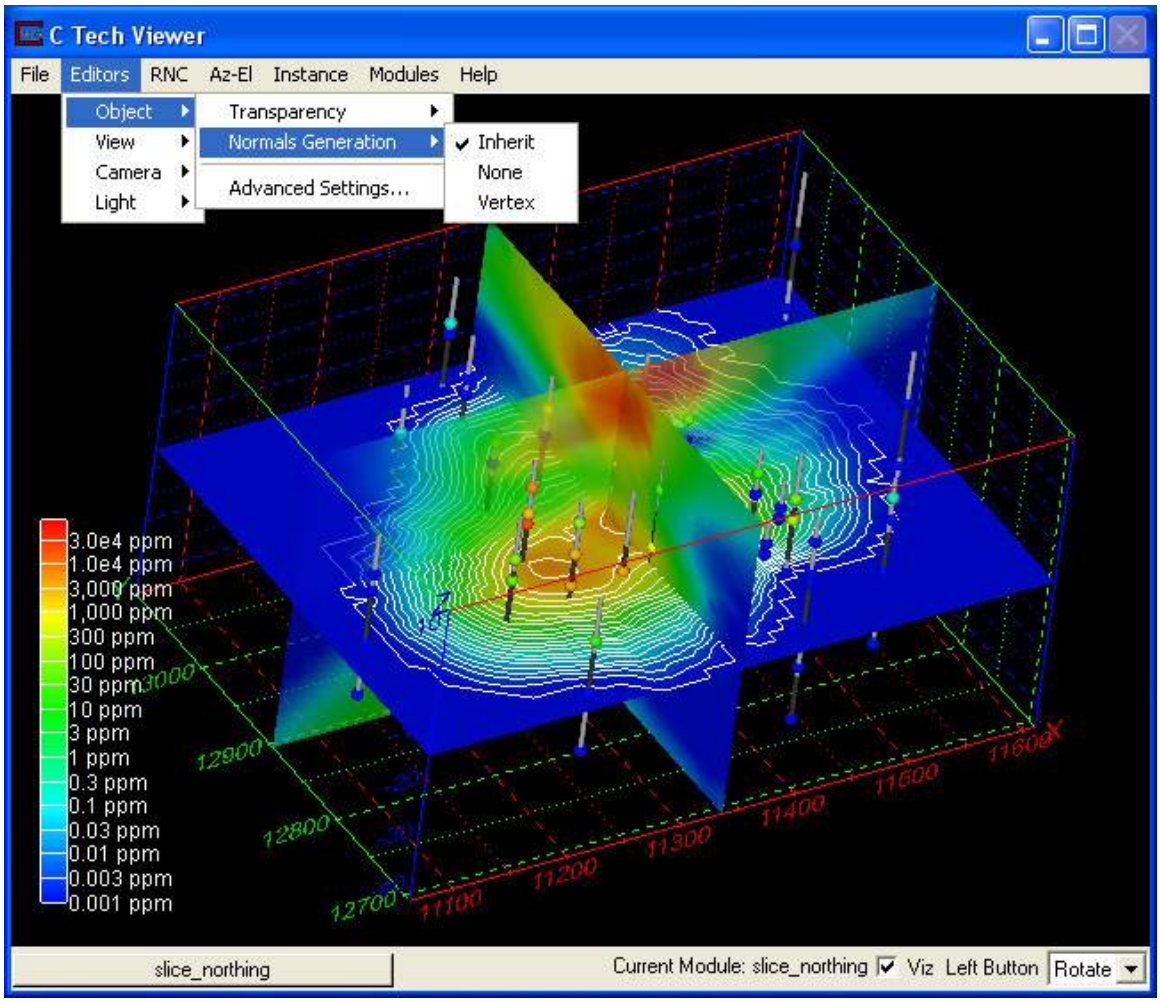
Object Editor



One of the most common object properties that we change is the transparency (or opacity) of objects. This can be done within the Object editor menu as shown above.

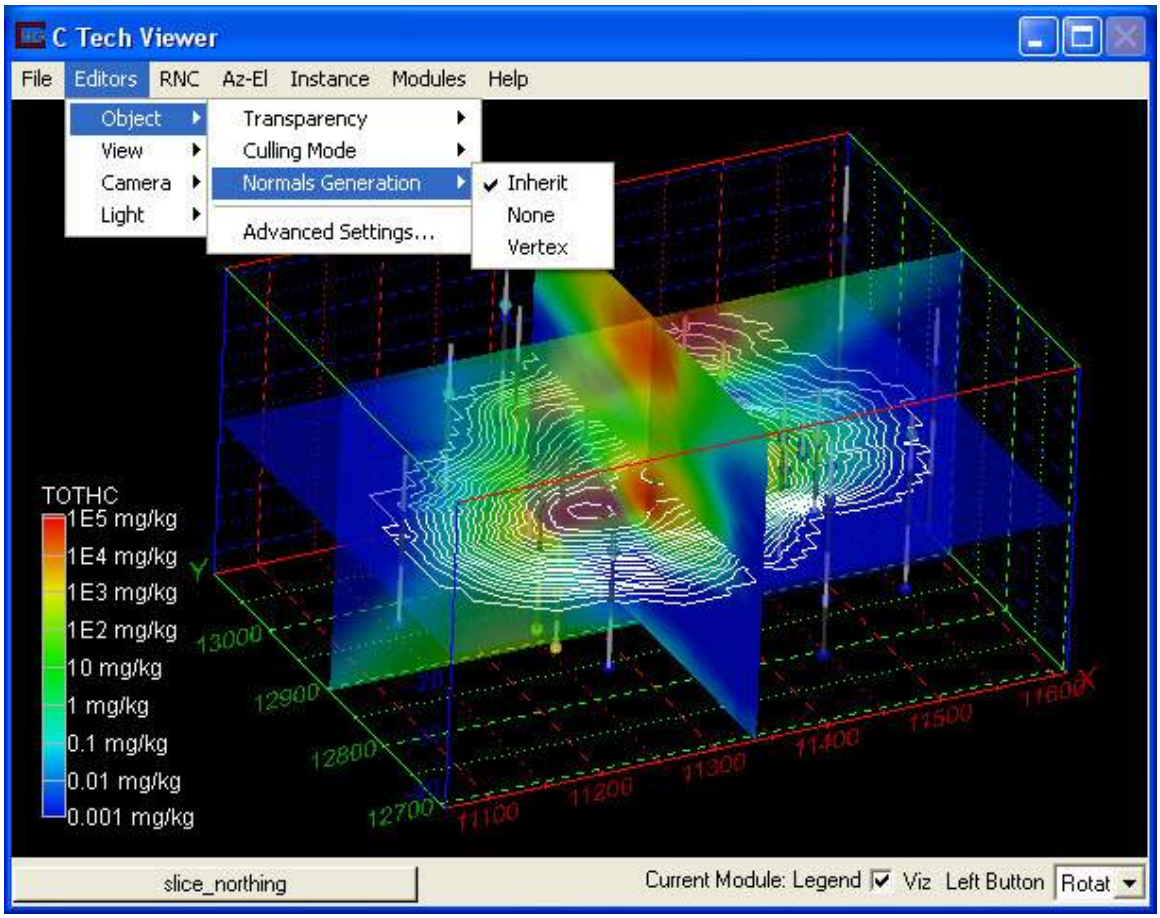


Related to transparency is whether to cull (not display) surfaces whose normal vectors point away from you. For closed objects (like plumes) this generally makes display of transparency superior and less ambiguous. For unclosed objects (like a Krig_2D surface or topography) it can create situations where part of the surface when viewed from underneath.



Another important parameter is whether to compute object vertex normal vectors. When vertices (nodes) have computed normals, the surfaces are rendered with Phong shading making the surfaces much more smooth. However this also tends to round the edges of objects which can be objectionable. With Object.Normals Generation set to None, the faces of each cell are distinct and the edges are sharp.

The *ObjectAdvanced Settings* submenu provides several advanced options for altering the characteristics of parent or child objects. The user should not change most of the settings in the General menu unless they are very familiar with the rendering processes that control the Viewer module.

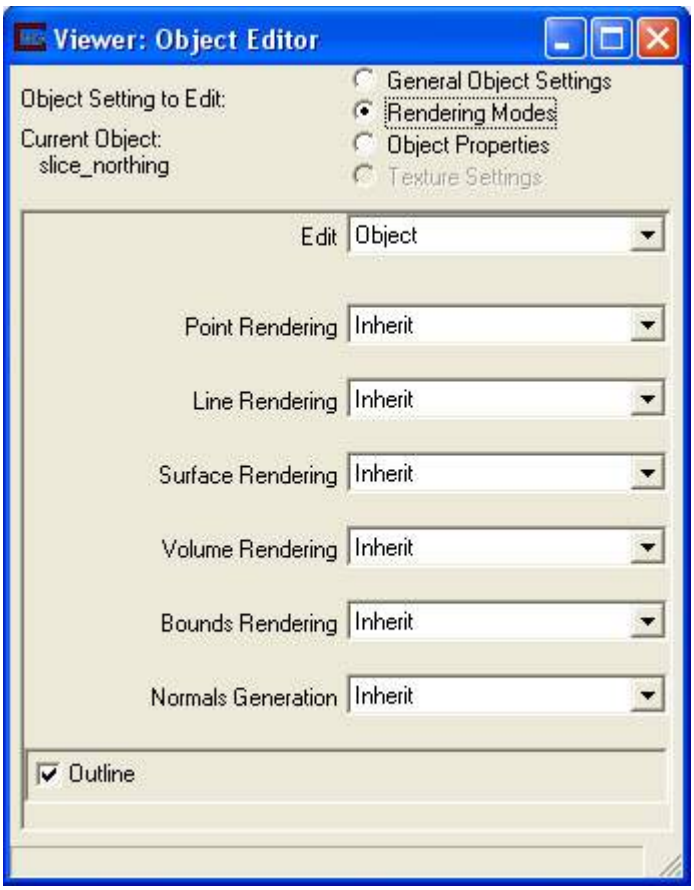


General Object Settings

The most important feature on the panel labeled **General** is the visible toggle which is also accessible on the bottom border of the Viewer itself. It provides a means to make any (selected) parent or child object invisible without having to disconnect that object from the Viewer.

On this panel you can also set whether an object is pickable (with Alt+Left Mouse), the Transform Mode (don't change this unless you understand the consequences), Cache Size, and VRML URL and Label.

RenderingModes



The Object Modes subpanel provides rendering options for points, lines, surfaces and model bounds. Settings for volume rendering are also supplied although seldom necessary due to easier interfaces in the volume_render module. Normally, objects are best left to Inherit the modes from the parent object, which has values that are most commonly desired. However, there are many occasions where modifying the modes of objects in the display is desired. Modes depend on the type of object. We will consider three primary types of objects Points, Lines and Surfaces. (The fifth mode, Bounds applies to all objects and consists of a box which surrounds the objects. This is used in EVS during rotation for certain objects.) Points do not have line or surface modes and lines do not have surface modes. Surfaces however, have all three possible modes.

The following options are listed along with the settings of most commonly used in EVS:

Point Rendering -- Inherit provides a default setting of NO point rendering. Pixel provides a single white point at each grid node of a surface or grid. Directed puts normal vector (porcupine quills) at each node. Tangent cross puts a "+" tangent to the surface at each node (only in OpenGL). Smooth Dot draws an anti-aliased fuzzy dot at each node (only in OpenGL).

Line Rendering -- Inherit provides a default setting of NO line rendering. Regular and Ribbon provide similar rendering of lines around each polygon of the selected surface(s).

Surface Rendering -- Inherit provides a default setting of Gouraud Shading resulting in a smooth (gradational) linear interpolation across any selected color range. No Light colors the objects a solid bright color (no lighting effects). Flat applies lighting effects, but colors triangular regions a solid color (no gradient shading), thus giving a faceted appearance to surfaces. Background makes all surfaces the color of the background. This may seem like no shading until you combine this with regular line mode.

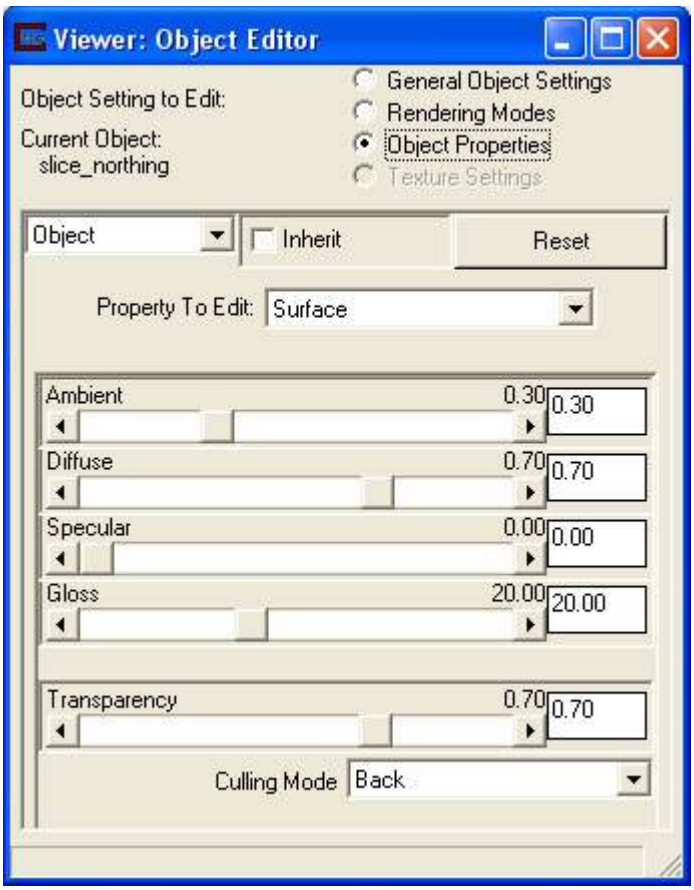
Volume Rendering -- This rendering type is only supported with uniform fields with two byte integer input. Choosing software renderer allows rendering via a ray tracing technique, while choosing OpenGL allows Back To Front (BTF) rendering. These settings are best adjusted from within the volume_render module.

Bounds Rendering -- Inherit provides a default setting of NO bounds rendering. Selecting Bounds results in a wire rectangular box around the current x, y, z extents of the selected object.

Normals Generation -- Inherit provides a default setting of Vertex normals which allows for surface interpolation between neighboring polygons. Selecting None results in flat surface polygons and thus creates a faceted appearance on surfaces. Sometimes a "None" setting is desirable when rendering cut or external faces.

Object Properties

The Object Properties submenu allows the user to change the colors, material characteristics, and line characteristics used to display objects in the Viewer. There are three radio button selectable items in Object Properties, and each brings up a window with controls for changing these characteristics.



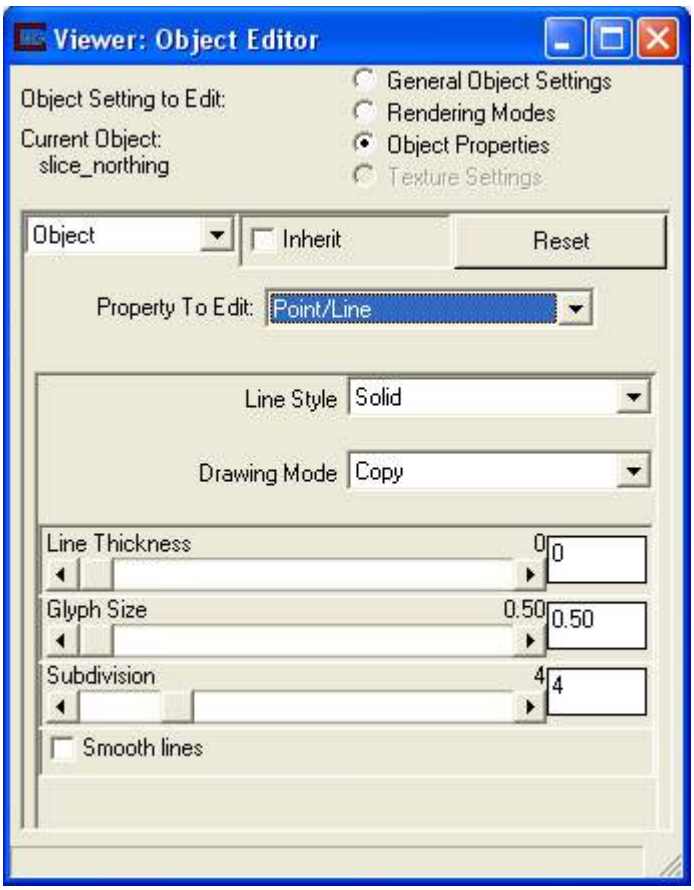
The ***Surface*** Type submenu presents sliders for surface lighting properties, surface reflection (shininess) and opacity.

The ambient and diffuse sliders affect the brightness of objects with respect to ambient (non-directional or lambertian reflection) and diffuse (directional or specular reflection).

The Specular and Gloss sliders determine the brightness and focus of white specular highlights on objects whose surface normal reflects the directional light source back to the viewer. This could also be considered shininess.

Of particular importance in this window is the ***Transparency*** slider, which can be used to set the transparency of objects in the viewer. A transparency value of 1 sets the object to be opaque, and a value of 0 makes the object completely invisible.

Culling Mode may be a useful adjustment if visibility problems arise using transparency. Try front culling on the object of interest if the rendering is not satisfactory with your opacity setting.



The ***Point/Line*** Type submenu presents line style choices and sliders for line and point display.

Line Style can be solid, dashed, dotted, and dashed-dotted.

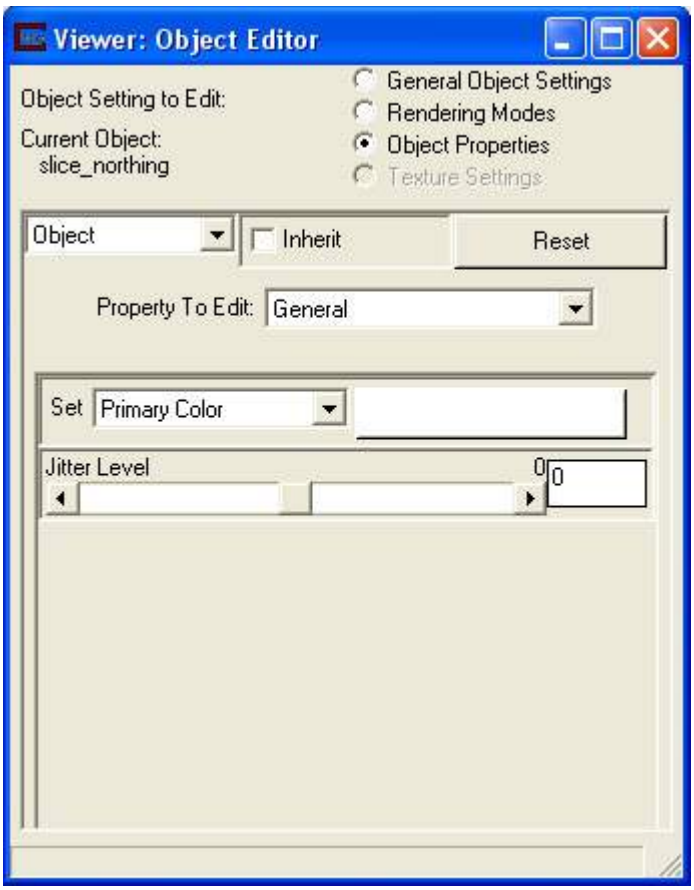
A choice of Copy and Xor Drawing Mode is presented, but EVS users should always use Copy (default).

The ***Line Thickness*** slider default is '0'. The Line Thickness slider sets the width of lines in pixels, and can be set to values ranging from 0 to 25. Zero is a special fast version of one pixel wide. The user will need to experiment with line widths to achieve the desired display.

Glyph Size applies only to special geometries.

The ***Subdivision*** slider controls the number of element subdivisions to be performed before drawing of lines along surfaces (grids). The higher the number, the more accurately the lines will be placed through the data. Higher accuracy does not always cause the lines to become smoother, depending on the data distribution. The default subdivision levels is four, and the maximum is 20.

The ***Smooth Lines*** toggle creates a (sometimes) -pleasing display of lines. The jagged display along pixel divisions is darkened and blurred to create a somewhat smoother (softer) appearance. Smooth lines produces thicker anti-aliased lines.



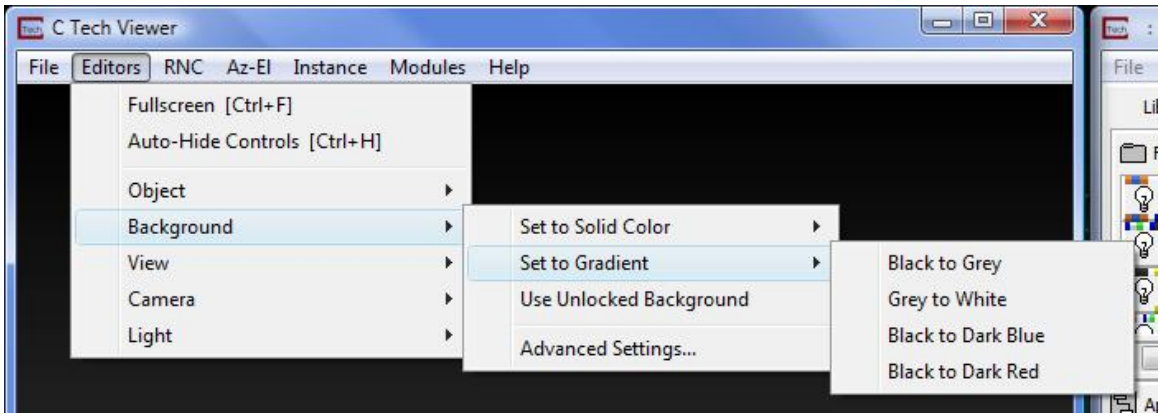
The ***General*** Type submenu presents a set of RGB value dials that will change the color of the selected object. The primary color can only be changed on objects which do not have colors already assigned to them. The extract_mesh module provides a means to strip off nodal data (and hence surface colorization) from objects.

The Primary Color submenus are used to change the color of highlight reflections on rendered objects with specular (or metallic) surface characteristics. These are not widely used in scientific visualization.

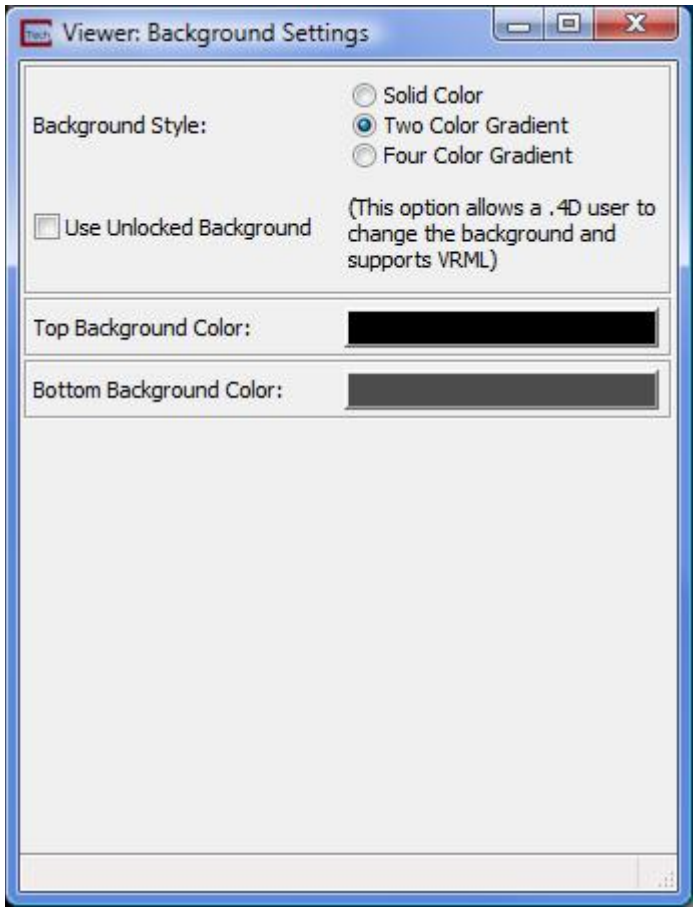
The Secondary Color submenus can used to change the color of lines in DXF files.

Background Editor

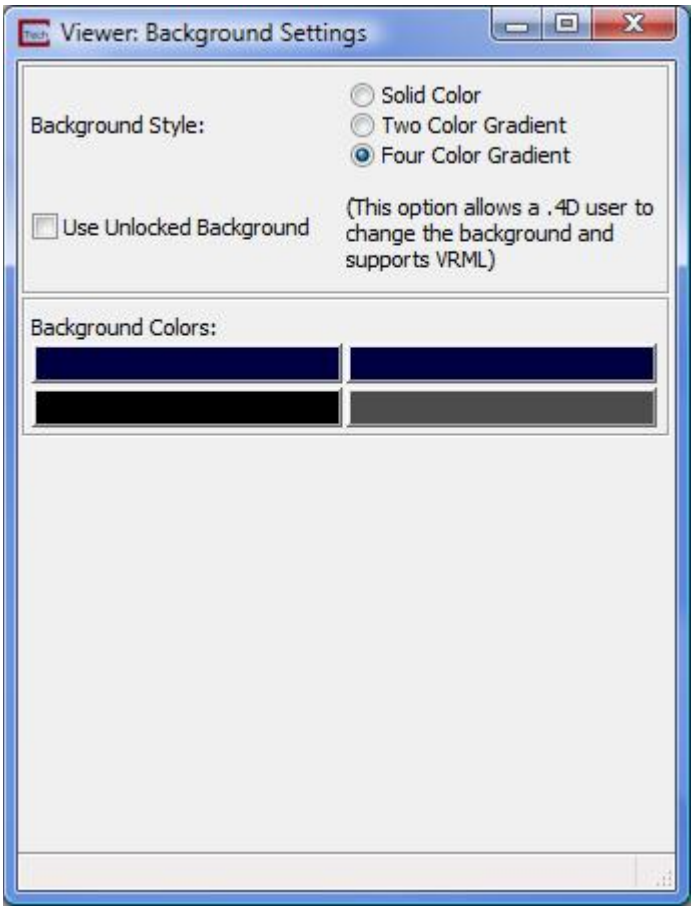
The background editor options are shown below.



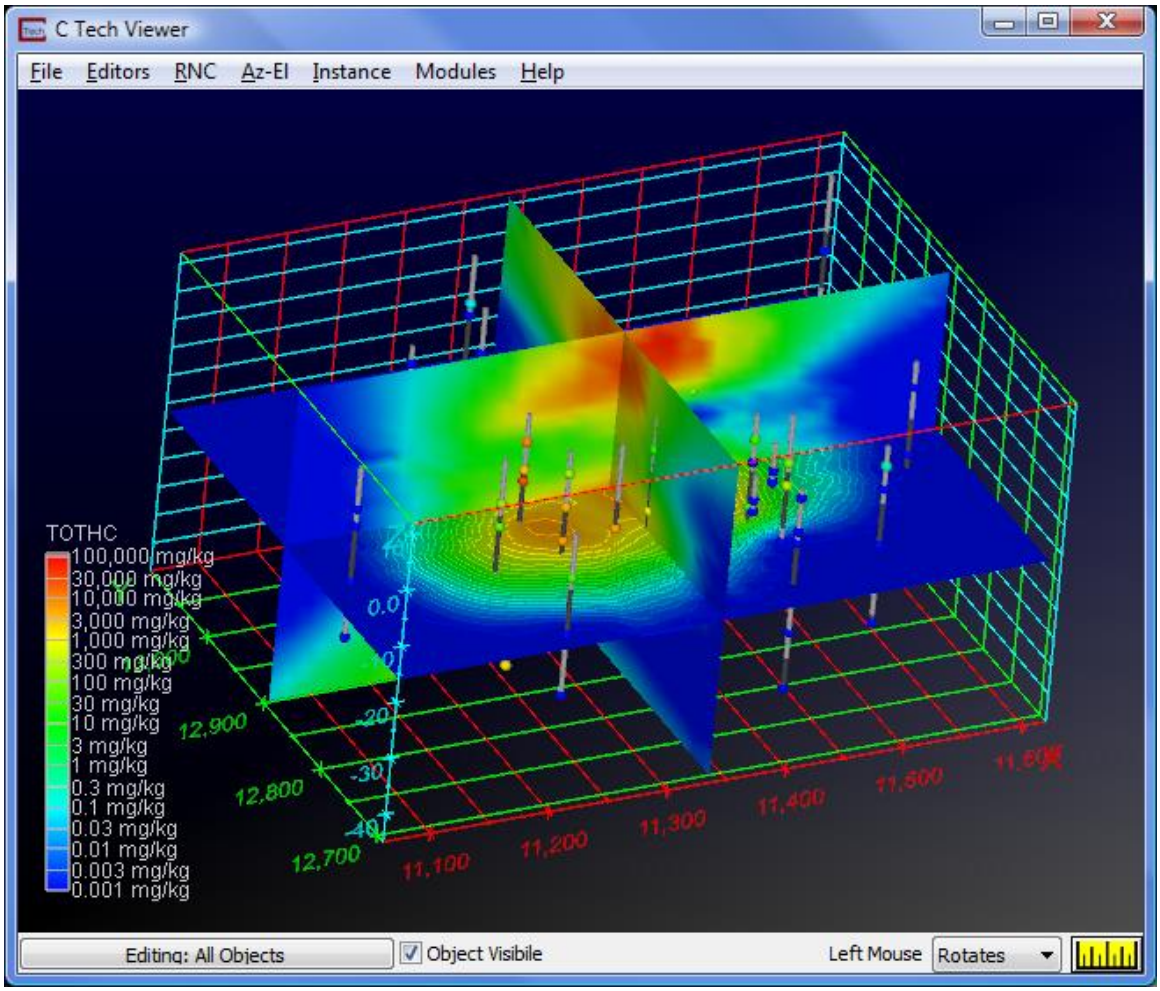
The advanced settings allow you to specify either a two color or four color gradient background. These "Locked" backgrounds cannot be changed in a 4DIM file. The two color style is shown below:



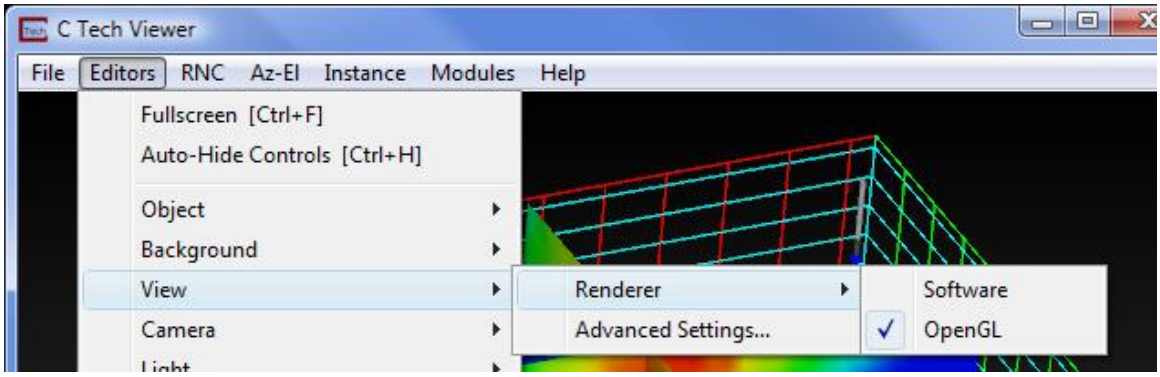
And the Four Color Gradient options are:



The background settings above result in the following Viewer appearance:

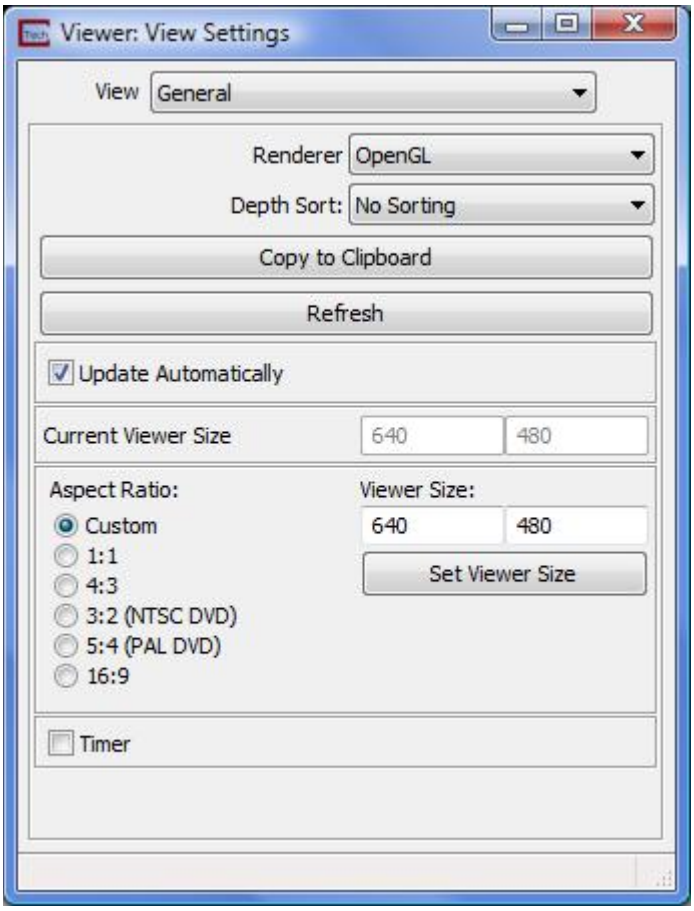


View Editor



The menu above provides a quick way to switch between OpenGL and Software Renderer.

The *View Advanced Settings* provides the user with additional feature rich submenus for changing viewer properties such as renderer selection, depth sorting, viewer window size, and even support for Stereo displays. Each of the submenus invokes a dialogue window with type-in options and/or toggles.



The ***General*** submenu contains most of the EVS user functionality and is shown in the figure above.

The ***Renderer*** pulldown list provides the option for either **OpenGL** or **Software** rendering.

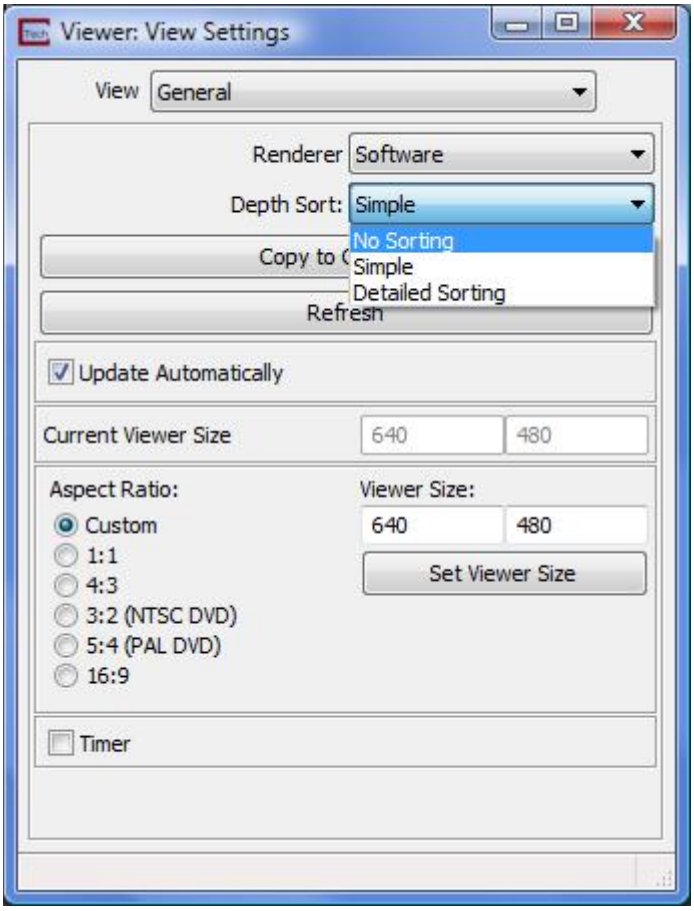
The OpenGL renderer is a high performance renderer which is native to the Windows operating system, and is the default option. For most objects, OpenGL rendering provides much faster rendering performance than the software renderer, and is the default setting.

However, the Software renderer provides some functionality that the OpenGL renderer does not, and therefore is the desired option for some operations. Specifically, spheres are calculated in the software renderer much more quickly and efficiently then OpenGL which renders a many faceted object. If you are visualizing a large number of spheres you should choose the software rendering option. The other important difference is that the Software Renderer support Depth Sorting when your view has multiple transparent objects.

When there are two or more transparent objects in your view, without depth sorting the appearance of which object is in front or behind can be improperly displayed. Properly displaying multiple transparent objects requires sorting the object's order of display based on their apparent position relative to the observer. There are two different depth sorting options, each

requires progressively more computation and is therefore increasingly slower than normal rendering.

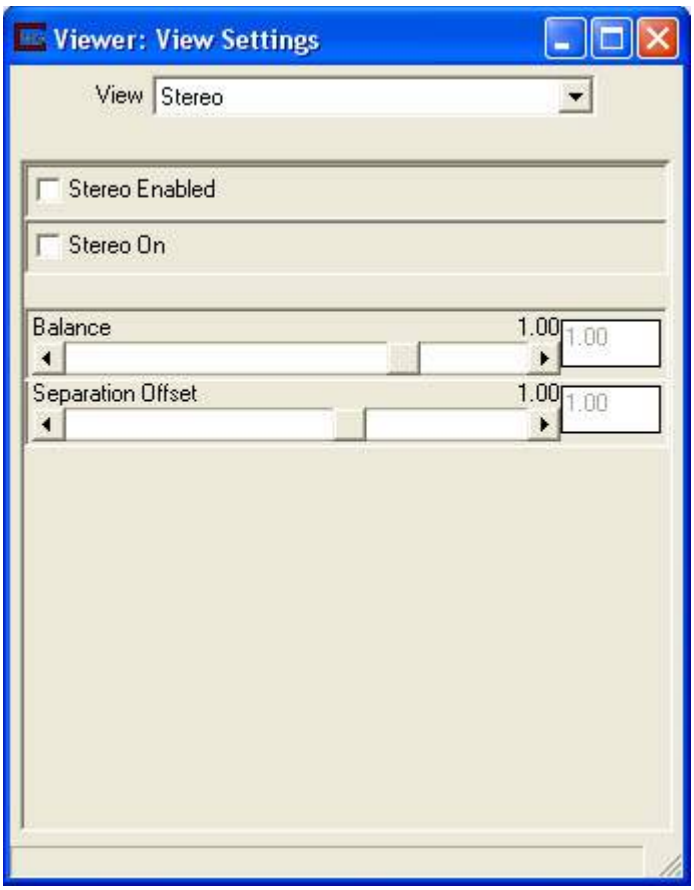
Remember that these options are not available in OpenGL since none of the graphics card manufacturers support this level of sophistication.



The **Viewer Size** type-ins control the **Width** and **Height** of the viewer window. The default is sized automatically by EVS depending on your computers display resolution. You may size the viewer window with your mouse or by using these type-in boxes.

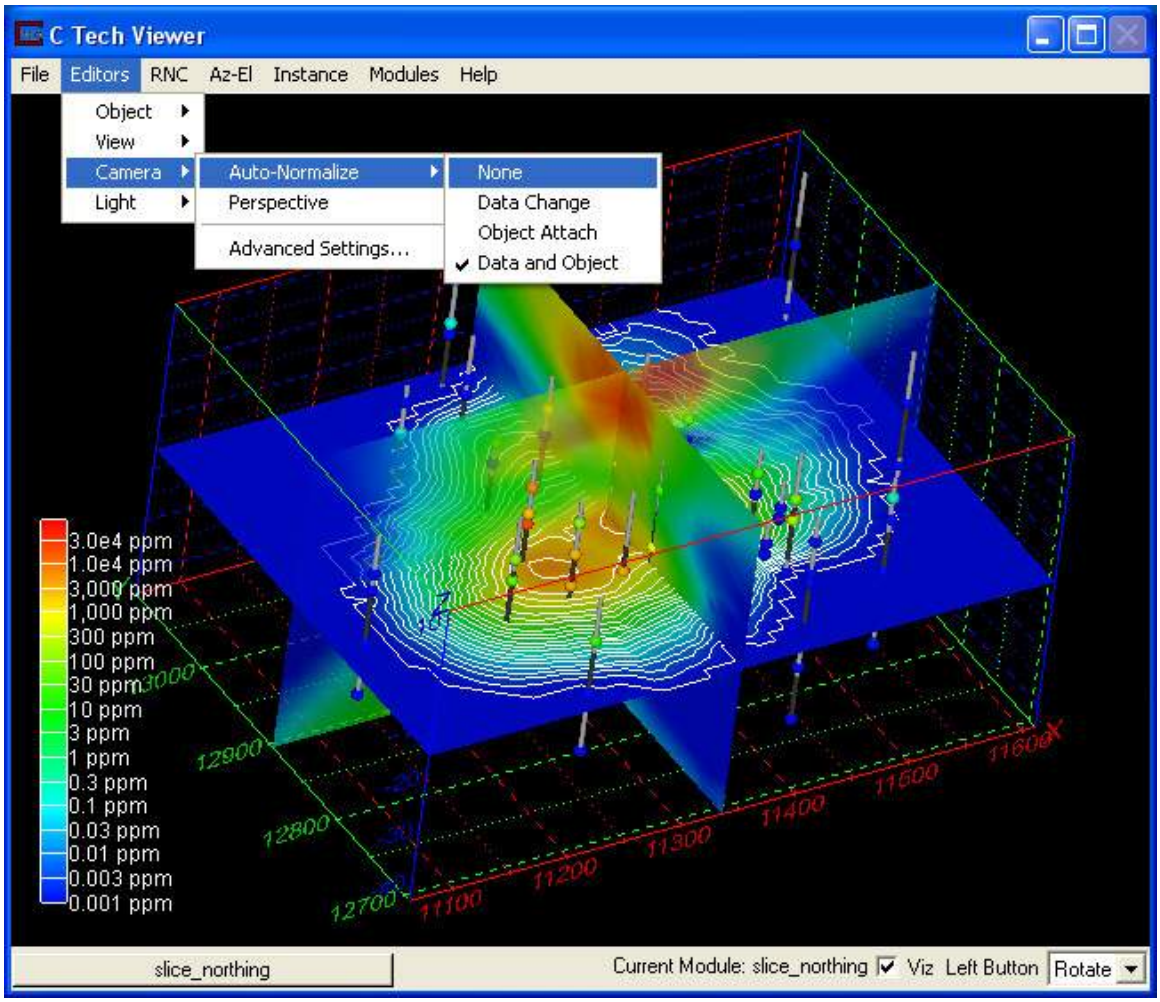
The **Aspect Ratio**selectors let you control the aspect ration based on the current width. This is a quick way to set a viewer shape that matches many standard output formats.

The ***Timer*** checkbox activates an internal clock which allows benchmarking of system hardware performance in terms of rendered screens per second. To use the timer, select this toggle and "throw" the object in the viewer. A throw is a method of rotating the objects in the view in a continuous fashion. To throw the view, start a rotation (left mouse button click...hold...and move) and release the button while the cursor is in motion. The next time you click (left mouse button) in the viewer it will stop the throw and performance stats will be printed to the right of the Timer toggle.



The ***Stereo*** submenu allows for Stereo viewing of objects and gives slider choices for Balance and Separation Offset. This functionality must be used in conjunction with special hardware. Please also note that in order to activate Stereo viewing, you must turn on **Stereo Enabled**, save your application and shut down EVS. Upon restarting and loading your application, the **Stereo On** toggle turns this on and off. Since Stereo Enabled uses so many resources, this roundabout method is required.

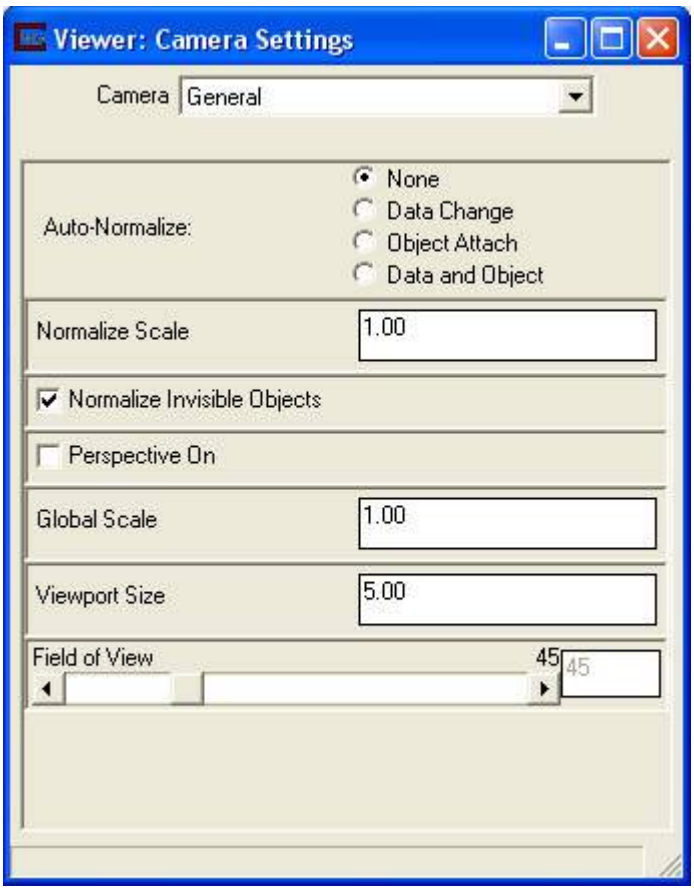
Camera Editor



One of the most common camera features to change is Autonormalize. This function causes the view to automatically scale and center about all of the objects in your view. Though incredibly useful when you are building your application and loading data, this feature can be very annoying when you are making small changes to your application.

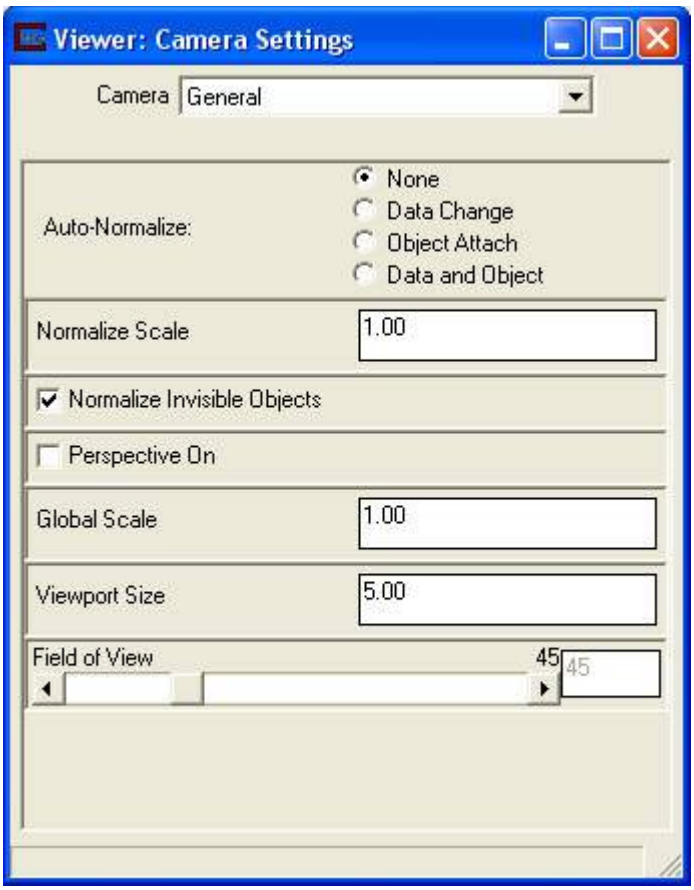
Another quick feature is whether to use Perspective. Remember this function is also on the Az-El panel.

The ***CameraAdvanced Settings*** menu provides the user with 5 submenus for changing camera properties such as the camera positioning, perspective and auto-normalization options. Each of the submenus invokes a dialogue window with type-in options and/or toggles.

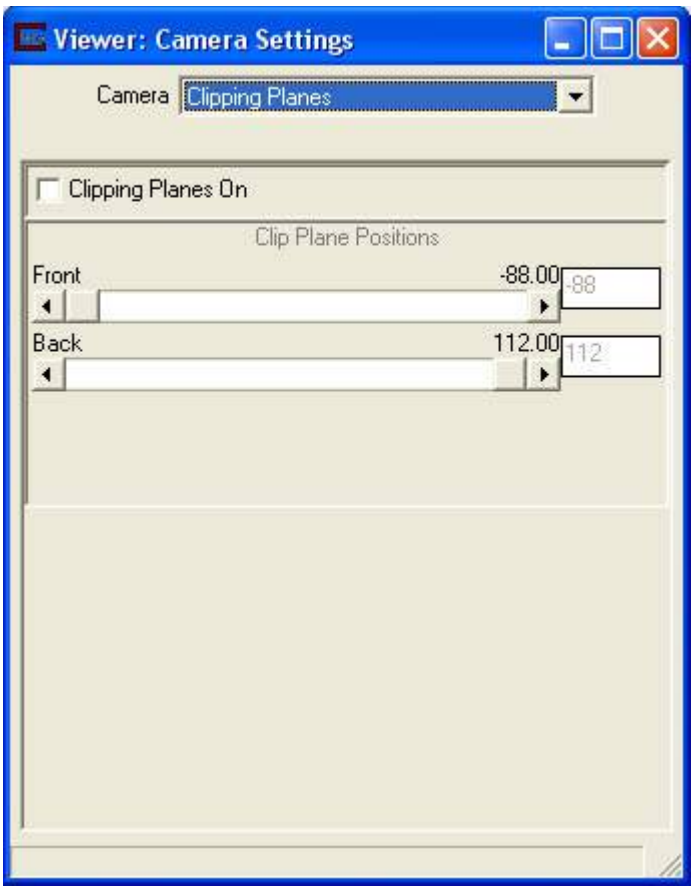


The ***General*** submenu contains the Auto Normalize option which is *Data and Object* by default which results in renormalization of the viewer whenever a new module is connected to the viewer or whenever an object's mesh is changed in any way. Any combination of the above is possible such as Auto Normalize on data (mesh) change only or object attach only, or none (Auto Normalize Off). All other settings in the General submenu should be left to the defaults by EVS users.

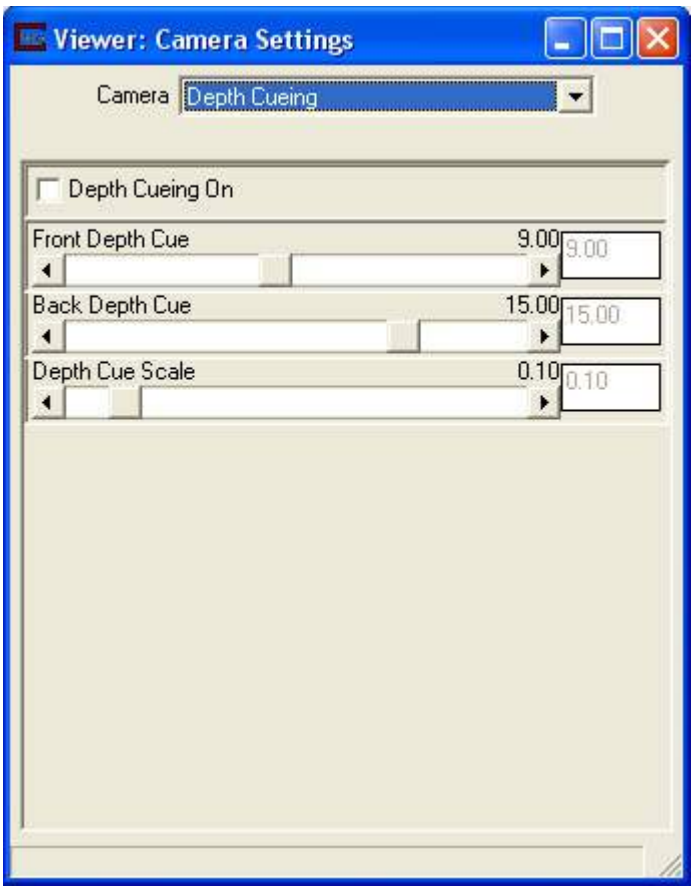
The lower half of this panel provides the main controls for perspective viewing. Perspective is an important feature, which is off by default. Normally we use a non-perspective view, which displays parallel lines as parallel. Objects are the same size regardless of their distance from the viewer. Perspective gives a view that can be likened to a camera view through a wide-angle lens. It sometimes seems more realistic, but perhaps less scientific (quantitative). The perspective option is used in conjunction with the Field of View slider at the bottom of the panel. Generally, objects will appear the same size (but distorted) with a 45 degree perspective view and the default view. A smaller Field of View is like zooming in on the display and reduces perspective effects. You can compensate for changing Field of View by adjusting the Viewport Size, changing the Scale Factor (on the Az-El panel), zooming in/out, or translating the objects in the "z" direction (shift right mouse button).



The ***Tripod*** submenu provides means of positioning the camera from a specified 'ground' to a tripod height and also through changing the actual 'ground' position and orientation of the tripod. This is difficult to set manually, but it is this functionality that is automatically driven by our fly-through and fly-over applications.



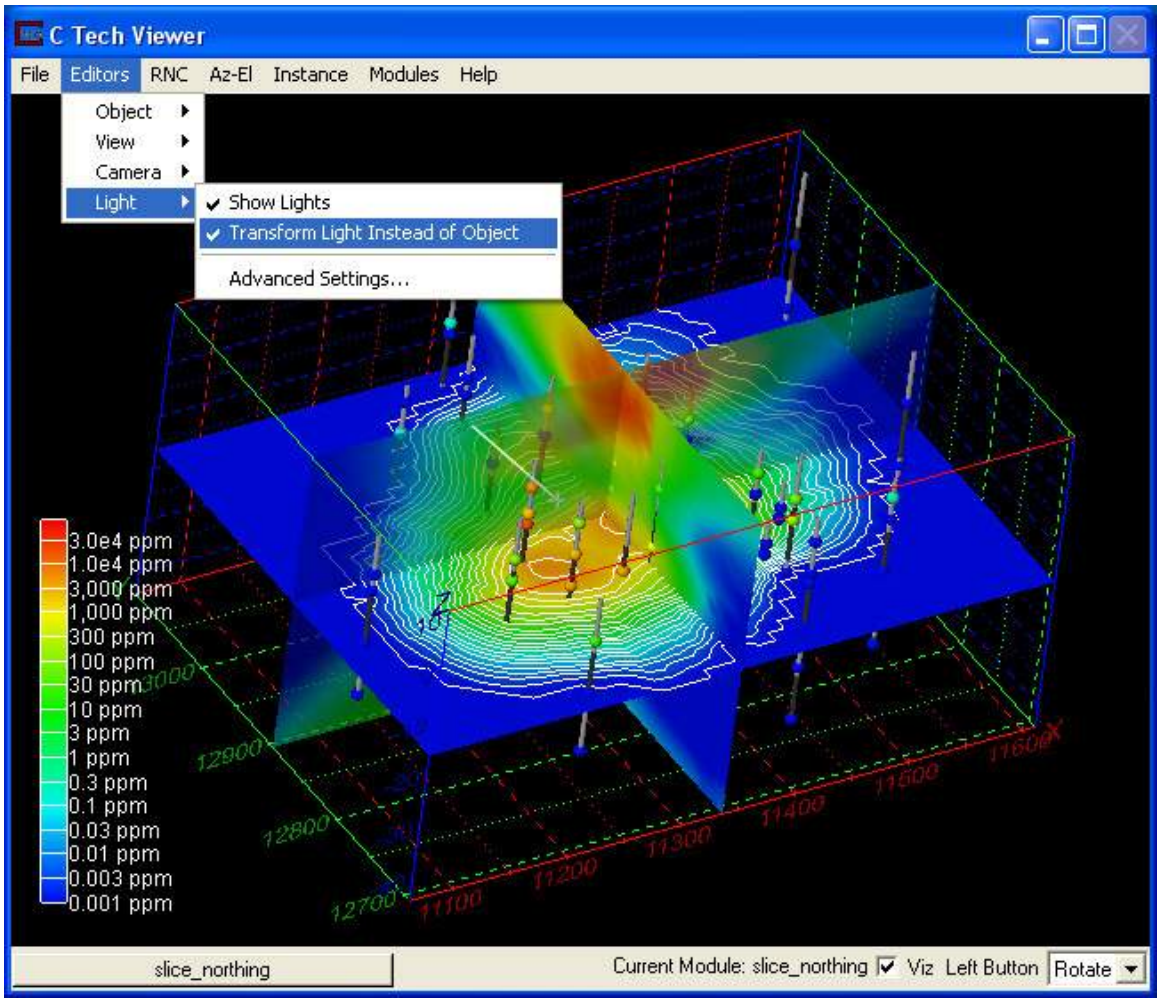
The ***Clipping Planes*** submenu provides tools for actually clipping away visibility of objects either near or far from the camera. Clipping can be turned on and off and the front or back planes can be moved with sliders to control clipping and z-buffering for the depth of view that is displayed in the window.



The ***Depth Cueing*** submenu provides an on/off toggle and three sliders for controlling front, back and scale. Depth Cueing is an advanced display option, which controls the fading of the light based on the distance objects are from the viewer.

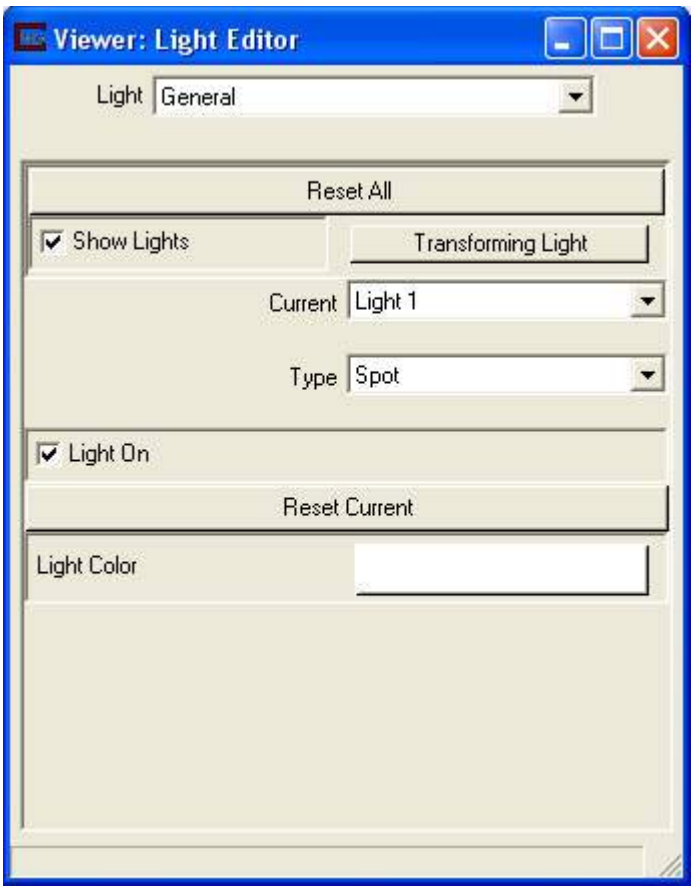
The user should experiment with different settings of the options in the Camera Editor Control Panel to understand the functionality of each. However, each time that a Viewer module is instanced, it is reset to have the default values, so no permanent changes to the camera are saved between sessions. However, camera settings are saved with Applications.

Light Editor

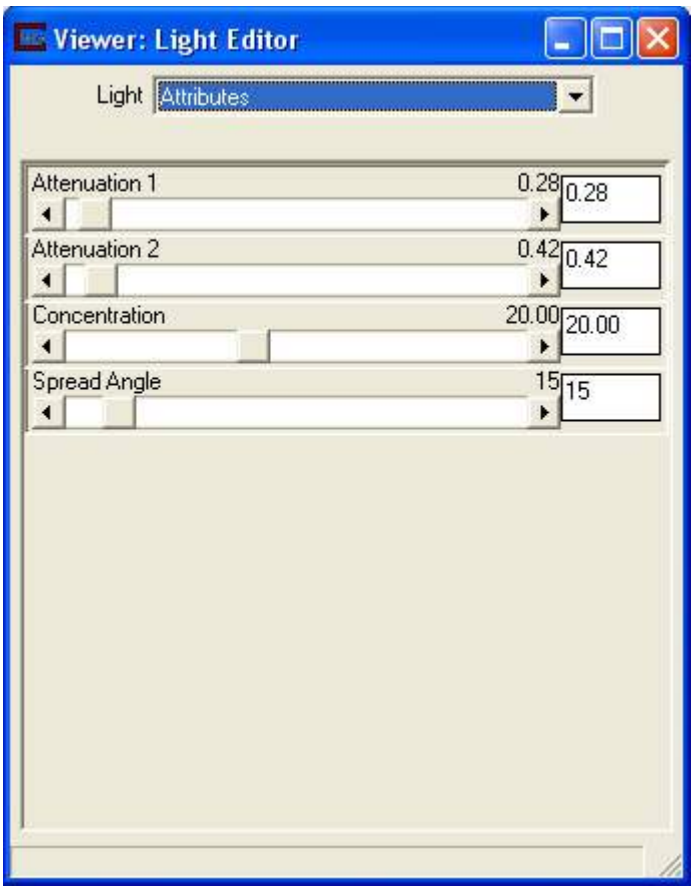


The two most commonly modified light features is whether to show the lights and to manipulate (transform) the light versus the objects in the Viewer. These two features can be controlled from the pull-down menu.

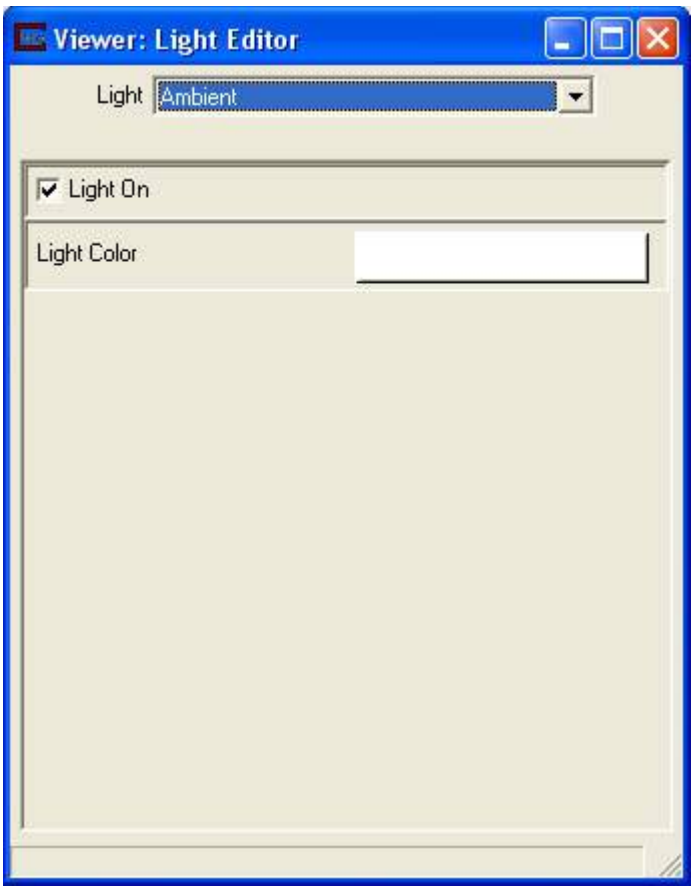
The *LightAdvanced Settings* menu provides the user with submenus for changing the characteristics of the ambient and directional lights that are being used to render the objects in the Viewer.



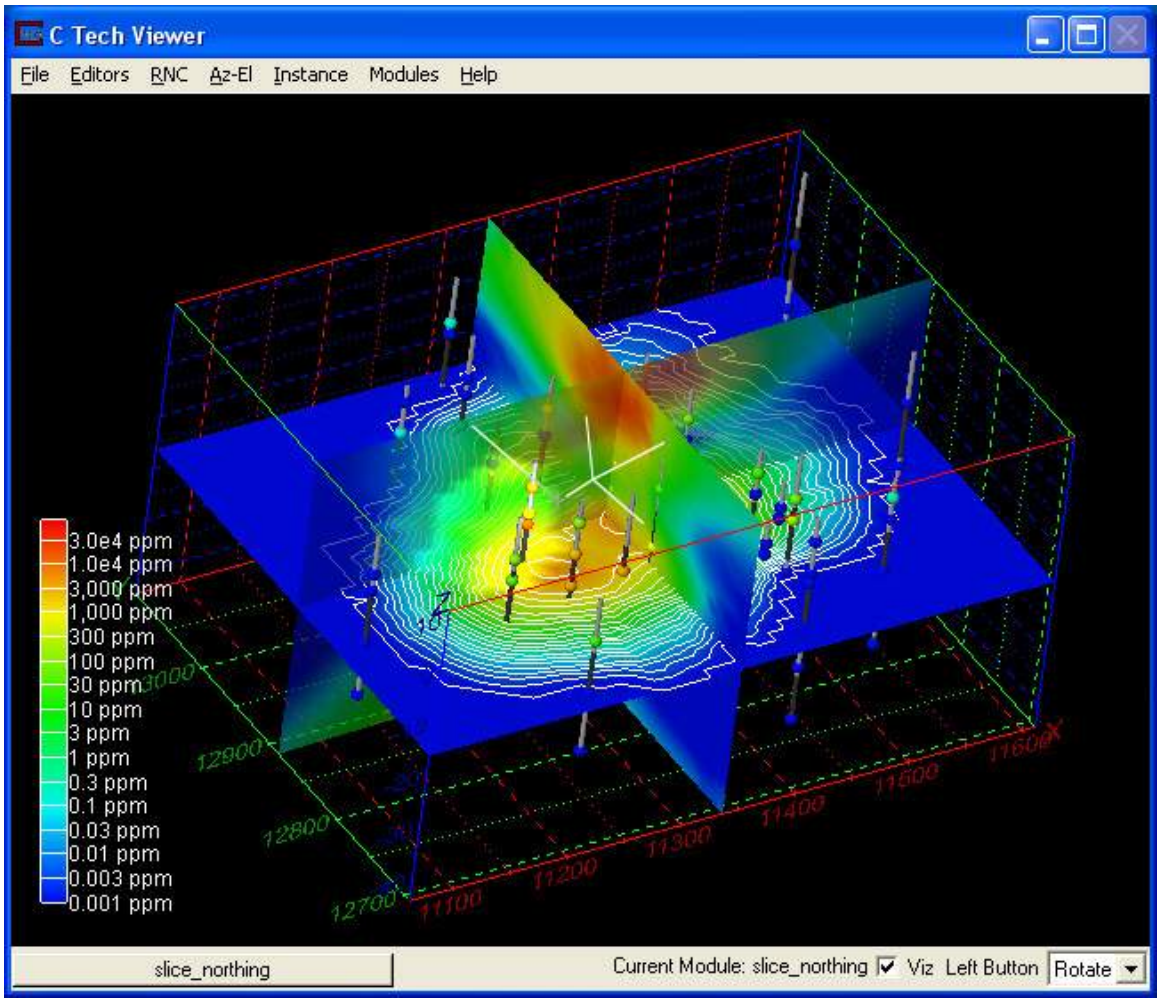
The ***General*** submenu allows for adjustment of four different directional lights (numbered 0 through 3). Each can be controlled using the Type selector (BiDirectional, Directional, Spot or Point).



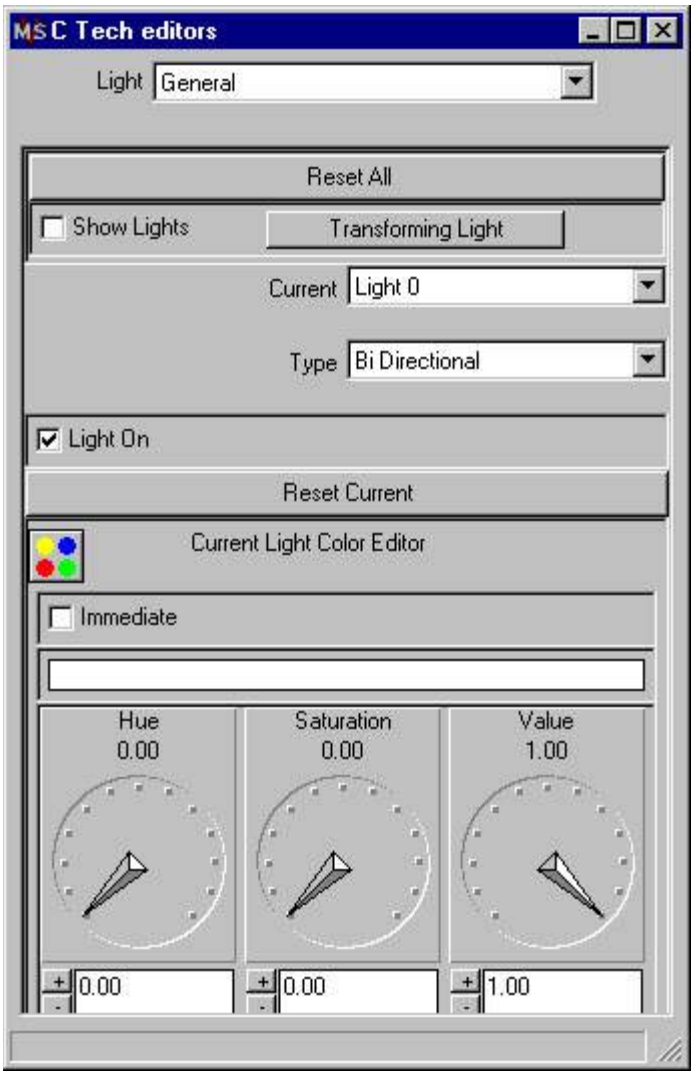
The Light On toggle controls which of the directional lights is being operated on with the color (dials or sliders) and ***Attributes*** submenu. Each light can be set to have different characteristics to achieve the type of lighting the user wishes for the display in the Viewer.

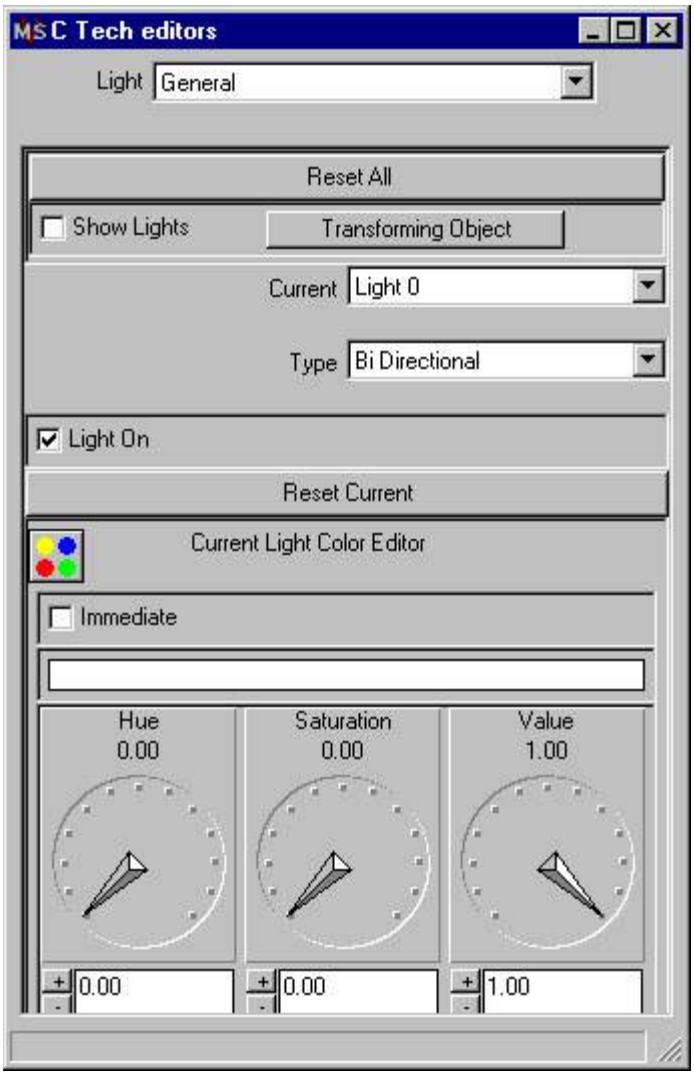


The ***Ambient*** submenu allows color adjustment of an ambient light source. The ambient light can be turned on or off by checking the Light On toggle, and its color can be modified using an RGB slider or dial control.



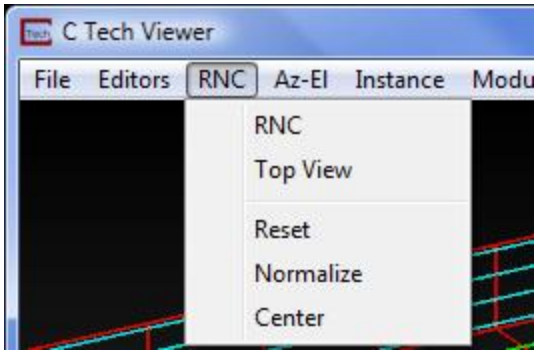
The picture above demonstrates adding a **Spot** light to highlight an area of a 3D model.


MS C Tech editors
Light
General
Reset All
Show Lights
Transforming Light
Current
Light 0
Type
Bi Directional
Light On
Reset Current
Current Light Color Editor
Immediate
Hue
0.00
Saturation
0.00
Value
1.00
0.00
0.00
1.00



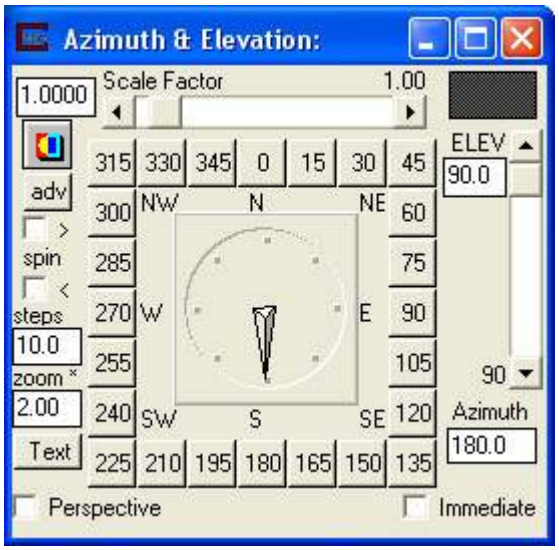
The Light_Editor has been enhanced to include a button which switches between "Transforming Object" and "Transforming Light". When "Transforming Light" is selected, your mouse (or the Transform Editor) will rotate the light. This is useful for enhancing surface topography and features. In many cases, you may want to switch to "Directional" vs. "Bi-directional" lighting.

RNC Menu



The RNC menu presents the user with buttons that will reset, normalize and center objects in the viewer. The Reset button resets the Viewer coordinate system to that of the objects so that the viewer space is consistent with a right handed system that has the +Z axis pointing toward the user. The Normalize button zooms the object so that it fills the width of the Viewer area. The Center button sets the center of rotation for the object to the center of the screen, so that rotating objects will remain on the screen. If the object is not centered in the viewer window, then rotations will occur around the origin at (0,0,0), which may move the object out of the viewer. Pressing the RNC buttons sequentially will always reset the center of rotation to have the geometric center of the object coincide with the center point of the viewer.

Azimuth & Elevation Panel

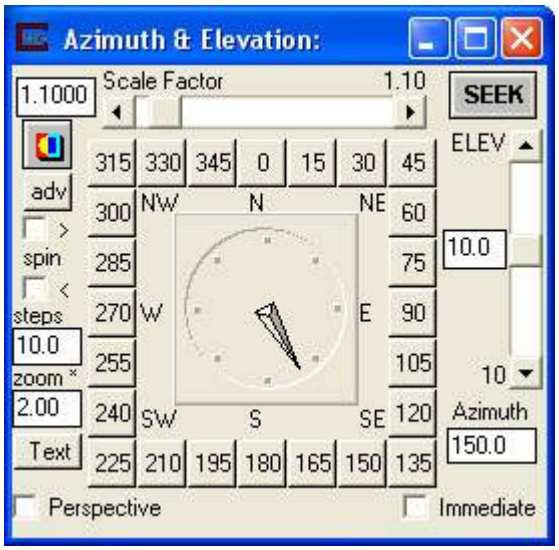


The Az-El panel provides the ability to set specific views using a variety of controls. These include:

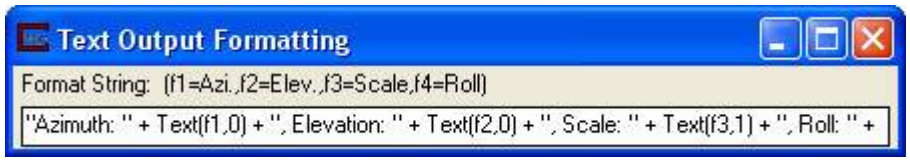
- A combined **Azimuth** dial or button array The panel is arranged with an array of azimuth selection buttons in 15 degree increments around the compass, and a dial for setting any azimuthal location. There is also a type-in box for setting a precise azimuth. An interesting feature is that if you move the dial with your mouse, as you cross North, it will add or subtract 360 degrees from the button values. This is important during animations since rotations crossing north must be properly specified to determine the direction and number of revolutions of your model.
- An **Elevation** slider (and type-in) sets the angle of the view from horizontal plane. An elevation value of +90 will result in a view from directly above the object (looking straight down the +Z axis), while an elevation value of -90 will result in a view from directly below the object (looking straight up the -Z axis). For example, a value of +20, provides a view looking at the object from 20 degrees above the horizontal

plane. Note that changing any of the settings on the Az-El panel updates the current view immediately.

- A **Scale Factor** slider (and accompanying type-in) allows the user to specify how much of the Viewer width the object(s) will occupy in the specified perspective view. A Scale Factor of 0.7 will therefore produce a display with the object occupying approximately 70% of the Viewer. Values above 1.0 will generally cause some of the model to be outside of the viewer window.
- The **RNC** (Reset-Normalize-Center) button with the tri-color icon in the upper left corner of the panel forces a recentering of the model and sets the view to the current Az-El settings. Furthermore, this button causes the Viewer to RENORMALIZE, which means that it recomputes the internal system wide scale factor required to fit all objects connected to the Viewer into the current view. Please note that the RNC pull-down menu on the Viewer includes a "Top View" option that is equivalent to this button PLUS resetting all Az-El parameters to their default settings.
- An important feature for creating simple animations and automating viewer manipulations is the **Spin** toggles and it's associated '**steps**' type-in box. The type-in specifies the number of degrees to spin the viewer azimuth and the '>' and '<' toggles determine the direction of spin. Use of this functionally essentially activates a loop whereby checking the '>' or '<' toggle increments the viewer azimuth according to the prescribed 'steps' type-in. For example, if you wish to automate views incrementing every 10 degrees with a clockwise rotation, you should type-in 10.0 (steps) and check the '>' toggle. The viewer will automatically step through from the beginning azimuth in 10.0 degree increments, in a clockwise direction. This spin can be stopped (interrupted) at any time by unchecking the '>' toggle. HINT: You may connect an output_images module to the viewer, then check the dynamic toggle to save a frame of each viewer position for use in building an animation.
- The **Perspective** toggle turns on Perspective mode. When it is turned on, an additional type-in for FOV appears. Additional settings for perspective are under the Camera editor.
- FOV (Field of View) sets the camera's included angle when in Perspective mode. The default value of 45 results in a camera with a 45 degree included field of view.



- The **Seek** button works in conjunction with the **zoom** type-in. After first renormalizing (hitting RNC), if you use the Alt+Left mouse button to select a location on any pickable object. At this point you should see the dark gray corner of the Az_El panel display the Seek button. Pressing this will cause the view to center on your selected point and zoom in by the zoom factor (default is 2). Subsequent pressing of Seek will continue to zoom in on the selected point.
- The **Immediate** toggle turns on immediate mode for scale and elevation sliders and the Azimuth dial. In immediate mode the view updates in real time as you move these controls.



- The **Text** button opens a *Text Output Formatting* window which allows you to create a text string for titling that contains view transformation (Azimuth, Scale, Elevation, and/or Roll).

The formatting options are identical to those used by the string_format module. Some examples are:

"Azimuth: " + Text(f1,0) + ", Elevation: " + Text(f2,0) + ", Scale: " + Text(f3,1) + ", Roll: " + Text(f4,0)

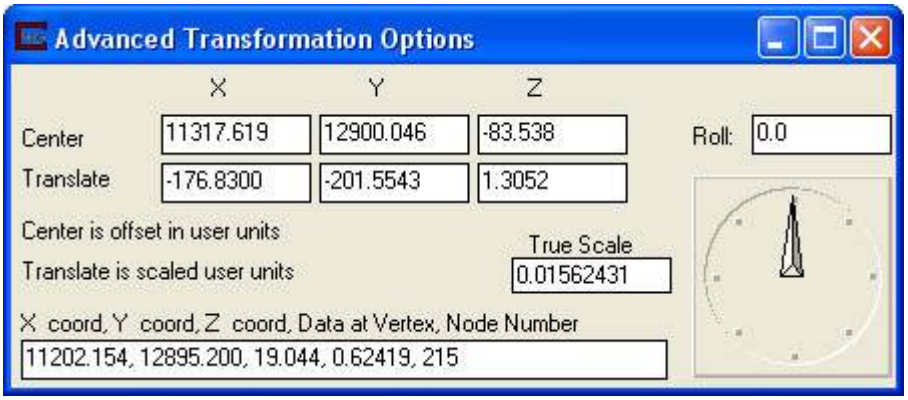
Displays all parameters with zero decimal points for Azimuth, Elevation and Roll and one decimal point for scale.

"Azimuth: " + Text(f1,1) + ", Elevation: " + Text(f2,1) + ", Scale: " + Text(f3,2)

Displays Az, El, and Scale only with one decimal points for Azimuth, Elevation and two decimal points for scale.

Advanced Transformation Options

Options for advanced transformations are available by clicking the small '**adv**' button located beneath the + button on the Az-El panel. Selecting this button provides XYZ dialogue boxes for changing the center of rotation for the object(s) in the viewer, and for translating the object's position within the viewer screen.



The values for the **XYZCenter and Translate** type-ins are automatically set when you renormalize or Seek. Normally the user will not change these values manually.

The **Roll** type-in and dial allows you to control the Roll of objects. This is a rotation that is not normally used since it causes vertical objects to not be vertical. It can be interesting for some fly-through animations or for unusual circumstances.

The type-in along the bottom of the window provides probing information about a point selected with the Alt+Left mouse button on any object. This includes coordinates, the nodal data (first data component) and the node number.

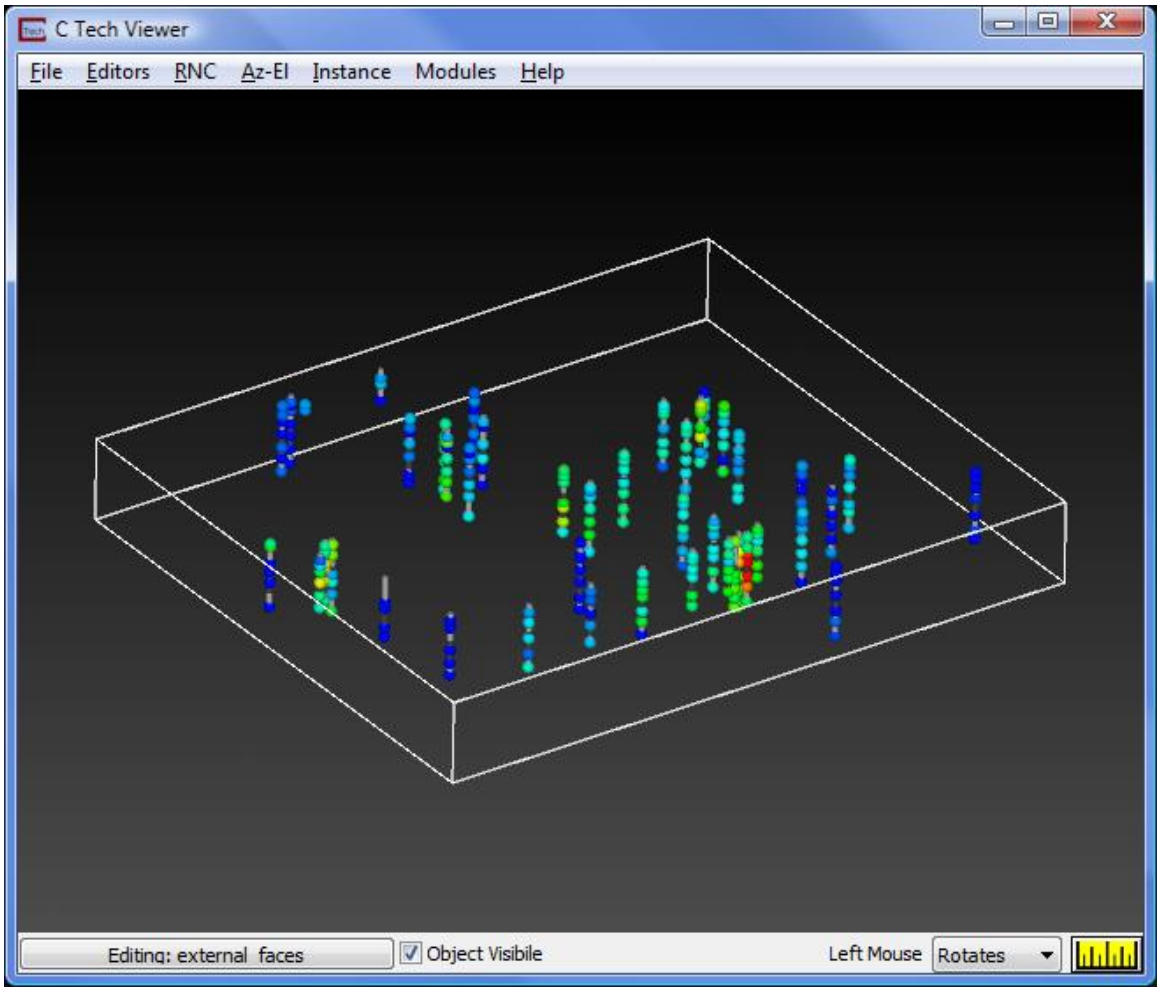
Instance_pull-down

Nearly all of the **Sample Applications** include the Viewer module, and the user can work through the tutorial in WorkBook 1 to gain an understanding of its operation.

Object Cache

One of the common problems that occurs with very large grids is that some objects (modules) connected to the Viewer will require more memory than is normally allocated. When this occurs you can recognize that this has happened when you see an object in your viewer that is only the white bounds of what SHOULD be displayed.

Such as:



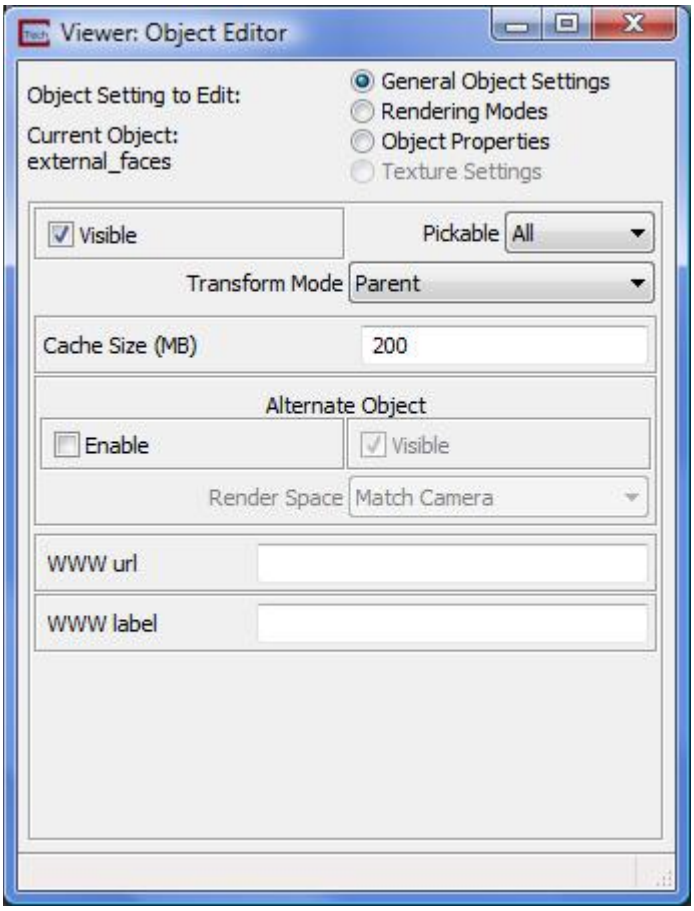
The Status Window also provides a warning message such as the one below that occurred when a 2 million+ node grid was connected to the external_faces module:

```
--- Warning from: module: external_faces ---
Field is too big (140 MB) to be put into GDobject's cache (128
MB). Drawing the
bounds only. Consider increasing the cache size or reducing the
field's
complexity.
-------------------------------------------------
```

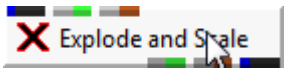
When this occurs, the procedure to fix it is:

1. Select the object using the "object selector" button (All Objects) in the lower left corner of the Viewer.

2. Under Object_Editor on the Viewer, under the "General" category, increase the cache size from the default value of 128 (Mb) to a larger value. In the example figure below we increased it to 200 Mb.



Explode_and_Scale



General Module Function

The Explode and Scale module is used to separate (or explode) and apply a scaling factor to the vertical dimension (z-coordinate) of geologic layers in a model. Explode and Scale can also translate the geologic layers in the z-coordinate direction, and can control the visibility of individual geologic layers.

Module Input Ports

Explode and Scale has three input ports.

1. The first (leftmost) accepts mesh and nodal data from Krig_3D, Read_UCD and 3D_Geology_Map. Input to this port must contain a UCD mesh and specific nodal data components.

2. The second input port accepts an explode factor, which specifies the distance that the geologic layer bottoms and tops will be separated.
3. The third input port (furthest right) accepts a scaling factor, which specifies the multiplication factor that the z-coordinates of the mesh nodes will be scaled by.

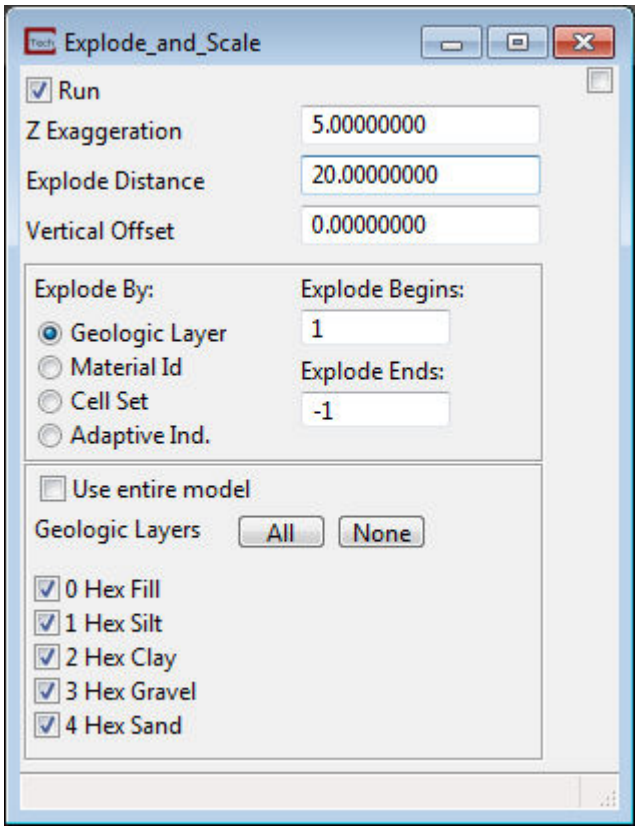
Note that the explode and scale factors can be passed to Explode_and_Scale, (from another Explode_and_Scale nodule) or the user can input them in the control panel if no factors are being applied to other modules in the network. Explode_and_Scale will report errors if the input data does not have geologic layer data for the fourth component (#3, #0 is first) and elevation data for the fifth component.

Module Output Ports

Explode_and_Scale has three output ports.

1. The first (closest to the left) outputs the explode factor that was input or passed to the module
2. The second (middle) port outputs the scaling factor. The first two ports can be connected to the post_samples and other Explode_and_Scale modules, to provide the factors being used by Explode_and_Scale.
3. The third output port passes the exploded and scaled data field, for input to other modules such as Generate Axes and Geologic Surface.

Remember that the values of the nodal data components are not affected by Explode_and_Scale, only the z-coordinates of the node positions are.



Module Control Panel

The control panel of Explode and Scale is shown in the figure above. There are five fields available for type in input:

- **Z Exaggeration** is the scaling factor for vertical (z) coordinates. This value reflects a multiplication factor of the layer thickness. For example, a layer that is 10 feet thick that is subjected to a z- exaggeration value of 4, will appear 40 feet thick in the viewer.
- **Explode Distance** is the distance between exploded layers, cell sets or materials. For example, an explode distance of 20 will separate each geologic layer by 20 feet from the layer above it and the layer below it. Note that the explode distance incorporates the scale factor, so that if a scaling factor of 5 is specified for a 10 foot thick layer to which an explode factor of 20 is applied, the geologic layers will be separated by 100 feet.
- **Vertical Offset** translates your entire model in Z by the value input multiplied by the Z Exaggeration. This is useful if you want to move your model to avoid having coincident objects.
- **Explode Begins** determines the first layer or material that will be exploded.
- **Explode Ends** sets the last layer/material which will be exploded. This value defaults to -1 which will explode all remaining layers or materials.

Explode By: determines what information is used to explode the layers or materials.

- The default is **Geologic_Layer** which is a nodal data component that should be present with any hierarchical geologic model created in EVS/MVS. For this type of model Geologic_Layer is always ZERO (0) for the uppermost layer and counts down to your lowest layer. If you have 5 geologic layers, the Geologic_Layer values will range from 0 to 4.
- **Material_ID** is a nodal data component that should be present with any hierarchical geologic model created in EVS/MVS. For this type of model Material_IDs are values which you have specified in your .GEO or .GMF file. Exploding by Material_ID allows you to keep geologic layers which represent the same material grouped together.
- The **Cell Set** option allows you to explode by the Cell Set number. For typical hierarchical geologic models created in EVS/MVS, exploding by Cell Set should be identical to exploding by Geologic_Layer. However, there are many occasions where the cell sets are quite different than the Geologic_Layer values.
- The **Adaptive Ind.** option is to be used only with Adaptive Indicator Kriged geologic models. These models have a cell data component called Indicator which corresponds to integer values assigned to material regions. Unlike models created with Indicator_Geology (which have only one cell set) models created with adaptive_indicator_krig have multiple cell sets corresponding to the Indicator values assigned to material regions in the PGF input file. For these models you should not try to explode by Cell Set since each material region will be represented by two cell sets; one of HEX cells and one of TET cells.

Under the **Geologic Layers** heading:

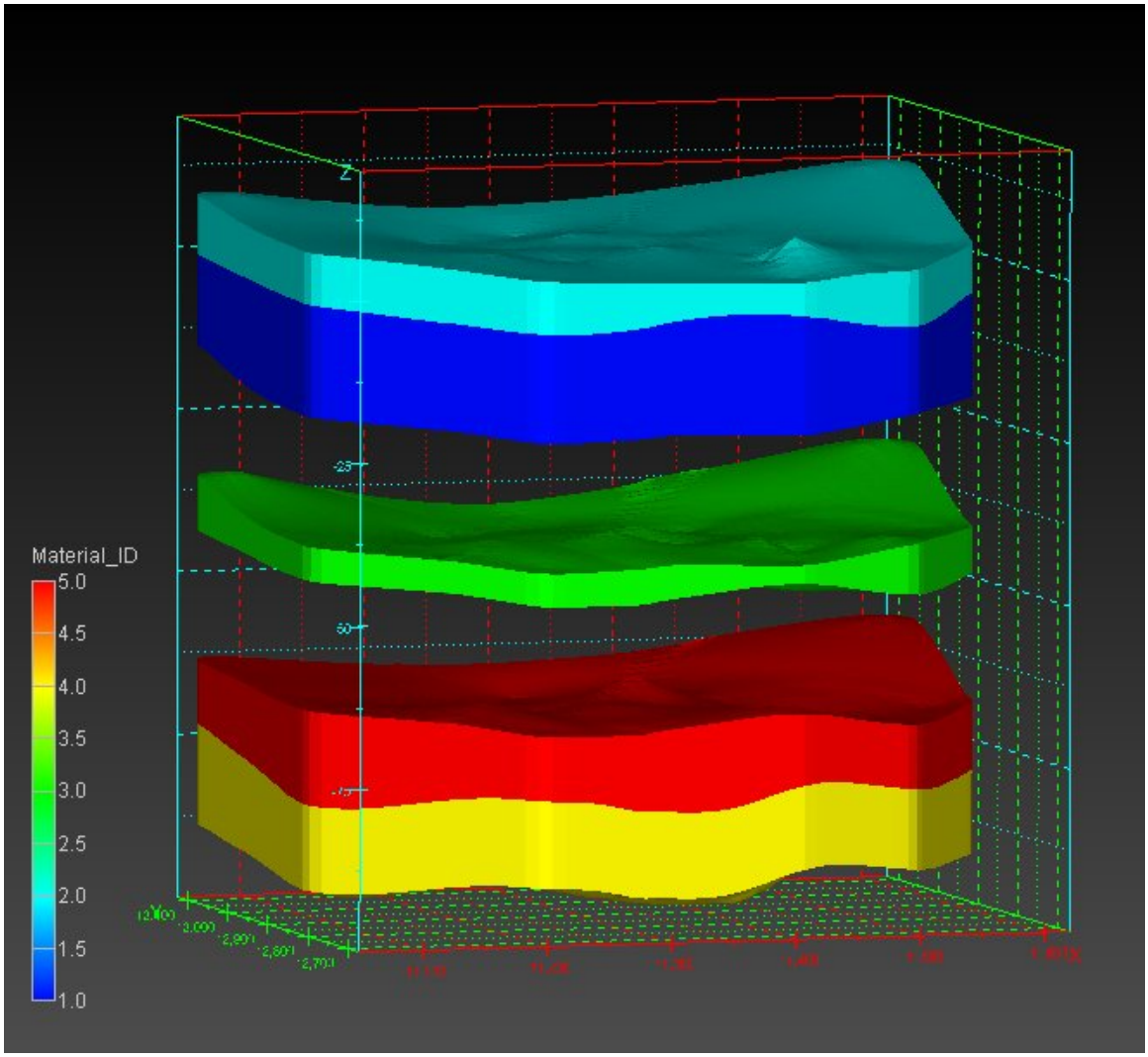
- The**Use entire model**toggle insures that all layers will be visible. If any upstream module drops the layers and then recreates them, without this toggle on only the first layer of the model may be visible. You must turn this toggle off in order to use the layer check boxes to subset layers.
- A series of check boxes will appear, one for each geologic layer (actually one for each cell set) in the model, which are used to select which layers (cell sets) will be output. This gives you control over the visibility of layers and materials.

Issues and Notes:

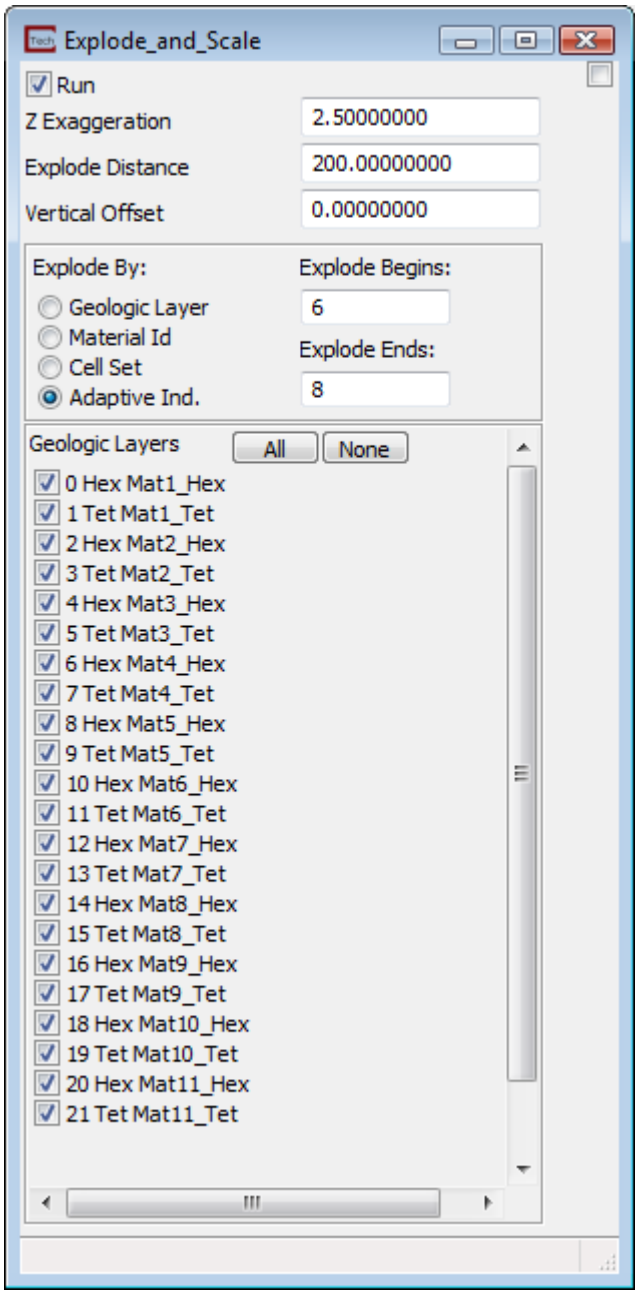
- Z Exaggeration and Explode Distance are values for the entire data set. Every geologic layer in the data will be exaggerated and exploded the same amount.
- Indicator Geology models cannot be exploded since there is only one cell set with shared nodes between different materials.

- Adaptive Indicator Geology models generally must have an Explode Distance which is greater than the total vertical extent of the unexploded model.

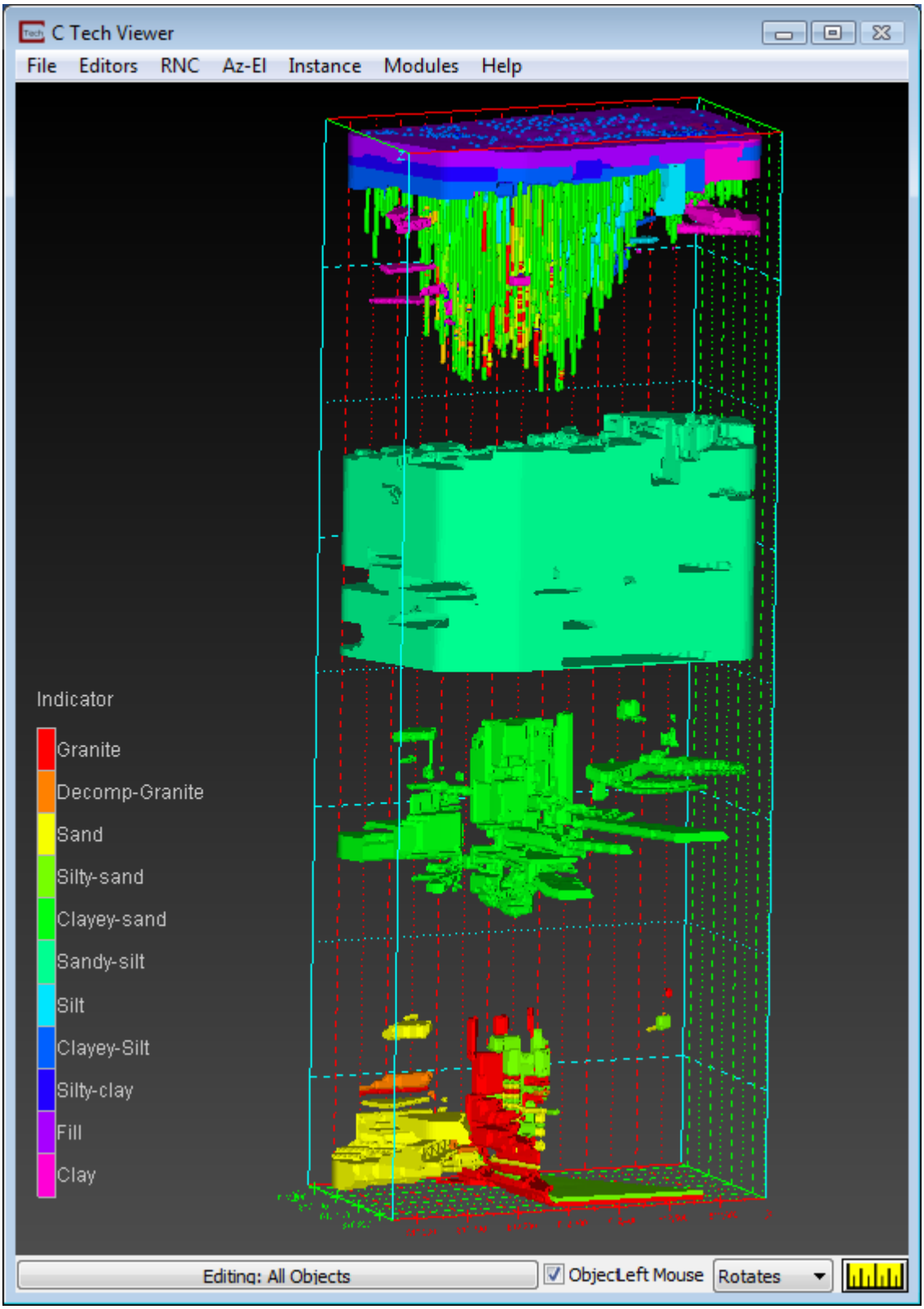
The example below shows the Viewer for the Explode_and_Scale settings show above with the data set used in Workbook 4.



The images below show the Explode_and_Scale window and resulting model for a 10 material (material IDs 0 to 9) Adaptive Indicator Geology model.



Since we are exploding from Indicator 6 through 8, the materials 0 through 5 are not exploded, materials 6 & 7 are separate and materials 8 through 11 are grouped together at the bottom. The model is in 4 groups, two more groups than the *Explode Ends minus Explode Begins*.


C Tech Viewer
File
Editors
RNC
Az-El
Instance
Modules
Help
Indicator
Granite
Decomp-Granite
Sand
Silty-sand
Clayey-sand
Sandy-silt
Silt
Clayey-Silt
Silty-clay
Fill
Clay
Editing: All Objects
Object
Left Mouse
Rotates

plume_shell

General Module Function

The plume_shell module creates the external faces of a volumetric subset of a 3D input. The resulting closed volume "shell" generally is used only as a visualization of a plume and would not be used as input for further subsetting or volumetric computations since it is hollow (empty). This module creates a superior visualization of a plume as compared with other modules such as plume_volume passing to external_faces.

Module Input Ports

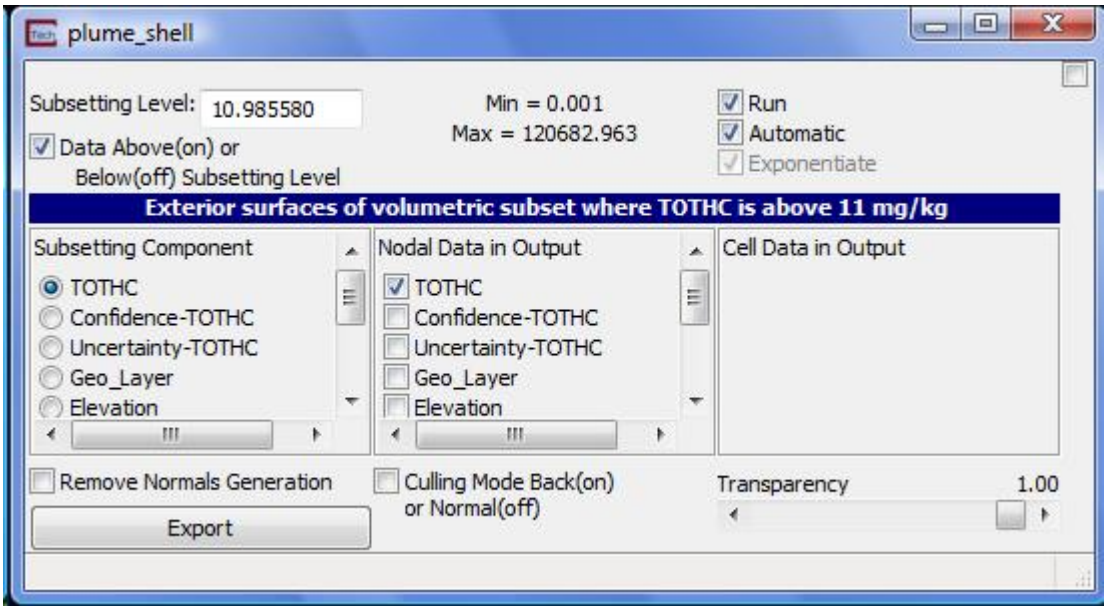
plume_shell has two input ports.

1. The leftmost (blue-black) port accepts unstructured mesh data.
2. The second (maroon) port provides a means to share the subsetting level with other modules.

Module Output Ports

plume_shell has four output ports.

1. The first output port (leftmost) provides an "info" string which can be connected to the Titles module for a quick simple title associated with the settings of this module. For the settings shown below, the Title displayed would be: "TOTHC above 11 mg/kg". This is constructed using the "Subsetting Component" name followed by "above" or "below", a rounded representation of the subsetting level and the units (if specified in the input).
2. The second (maroon) port provides a means to share the subsetting level with other modules.
3. The third (blue-black) port outputs a new unstructured mesh which contains the surfaces which are the external faces of a volumetric subset of a 3D input. Nodal data sent to this output port consists of the nodal data in output.
4. The fourth (red) port outputs a rendered geometry directly to the Viewer.



Module Control Panel

The control panel for plume_shell is shown in the figure above.

The ***SubsettingLevel*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default level is the arithmetic average of the minimum and maximum values in the subsetting component. If your input data has been kriged with log processing, the values here will be exponentiated already (provided that the Automatic toggle is on). You will not enter the Log of your desired subsetting value. Also, when the data is log processed, the default value is the geometric mean (vs. arithmetic mean) of your data extremes.

The ***Run*** toggle when not ON will prevent the module from running.

The ***Automatic*** toggle when ON determines "automatically" if the subsetting level values will be exponentiated for you vs. having to input value which are the LOG of your desired level. When this is on the Exponentiated toggle is inactive.

The ***Exponentiated*** toggle is inactive if the Automatic toggle is ON. Otherwise, it determines whether to real units to logarithmic units..

The ***Data Above (on) or Below (off) Subsetting Level*** check box (Above Box) is used to display data above the subsetting level or below the subsetting level. For example, to display a volumetric subset of all concentrations greater than or equal to 1 ppm, set subsetting level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the plume_shell of 1 ppm and below, simply turn the Above Box off (no check in box).

Subsetting Component refers to the nodal data component used to create the subset of the original input field. When an component is selected, the min

and max values of the variable are displayed in the upper center of the dialog box. The default component is the first (0th) component in the column.

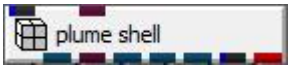
Nodal Data in Output determines which nodal data components will be included in the output. The first one in the list will determine the data used for coloring in the (red) output to the Viewer. For example, choosing subsetting component concentration and uncertainty for the nodal data in the output will create a volumetric SHELL subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The ***CellData in Output*** option box selector lets you map cell data (if any) to the surfaces output by plume_shell. The cell data will only be visible if all nodal data is unselected (under *nodal data in output*).

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Culling Mode Back (on) or Normal (off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

The ***Transparency*** slider changes the transparency of the output to the red port to the Viewer.

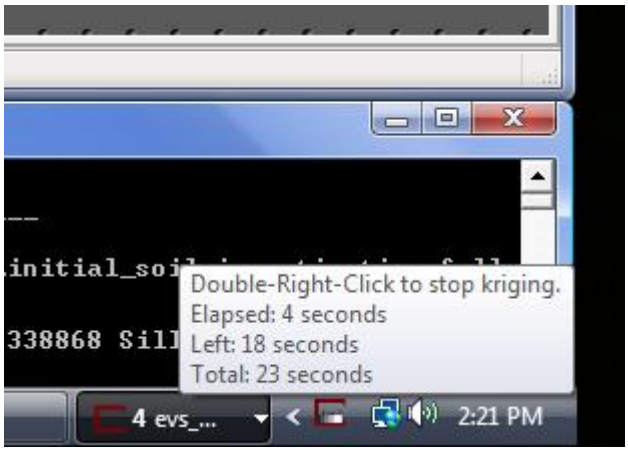


The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. Once this button is pushed, plume_shell will have has seven output ports.

1. The info string
2. The subsetting level
3. The string representing the selected subsetting component
4. The string representing "above" or "below"
5. The units as a string
6. The (blue-black) output
7. The (red) output to the Viewer.

Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Related Modules

-> plume_volume

-> isolines

sequential_subset

(This module is available only in MVS)

General Module Function

The sequential_subset is a powerful new module that incorporates some of the characteristics of plume_shell, yet allows for a large number of sequential (serial) subsetting operations.

This module creates a superior visualization of a plume that can be sent directly to the viewer for rendering. This is not a true volumetric subsetting module (as is plume_volume or contour) since it does not export a volumetric representation of the plume. It is used exclusively for plume visualization of 3D fields and for export as VRML. Sequential_subset outputs a specialized version of a sequentially subset plume that is suitable for VRML export for 3D printing to create full color physical models using Z Corp'stechnology.

For output to Z Corp. 3D printing, pleasejump tothe Issues for Z Printtopic.

Without sequential_subset it is very difficult if not impossible to create a VRML file suitable for printing, especially with complex models.

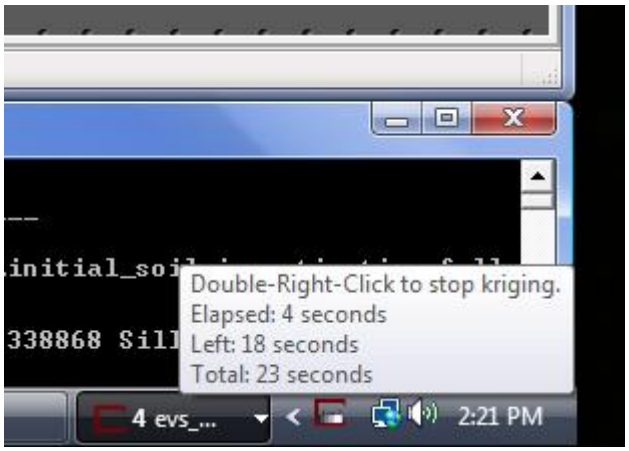
Module Input Ports

sequential_subset has one input port that accepts unstructured mesh data.

***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the

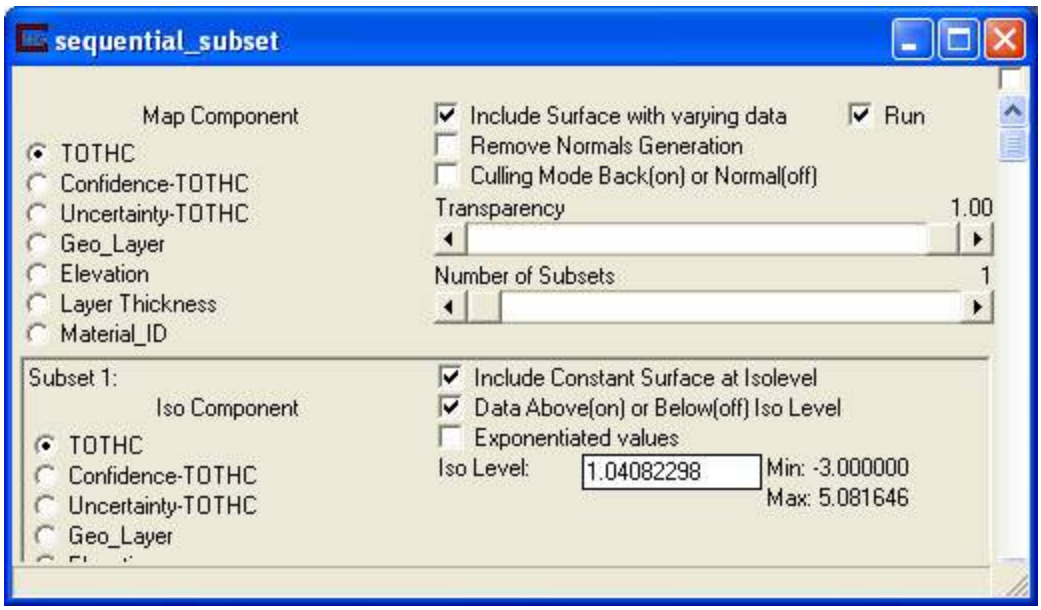
output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Output Ports

sequential_subset has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains a merged field containing all of the exterior faces of the plume resulting from the sequential subsetting operations.

The second port outputs a rendered geometry directly to the Viewer.



Module Control Panel

The control panel for sequential_subset is shown in the figure above. sequential_subset's user interfaces changes automatically depending on the selected ***Number of Subsets*** to be performed. The **TopPanel** allows you to select the following parameters:

Map Component determines which model data component will be sent to the output ports. The map component selected will be used to color the output. For example, choosing iso component concentration and map

component uncertainty will create a volumetric subset of concentration colored by uncertainty.

The ***Transparency*** slider changes the transparency of the entire plume.

The ***Include Surface with varying data*** toggle controls the visibility of the surface(s) having varying data (based on the map_component. The ability to control visibilty of portions of the output allows you to use multiple sequential subset modules and color or texture_map portions of each plume differently.

The ***Run*** toggle will keep the subsetting algorithms for running. If there is a large model being subset it is usually better to turn Run off and then make all of your changes. This keeps the module from running after every change and will speed the application construction process.

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Culling Mode Back(on) or Normal(off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

The remaining panels will be named **Subset 1**:, **Subset 2**: and so on. Each of these is identical in their options having the following parameters:

Iso Component refers to the model data component used to create the subset. When an iso component is selected, the min and max values of the variable are displayed in the right side of the panel.

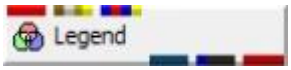
The ***Include Constant Surface at subsetting level*** toggle controls the visibility of the surface(s) having a constant value (based on the iso_component. The ability to control visibilty of portions of the output allows you to use multiple sequential subset modules and color or texture_map portions of each plume differently.

The ***Data Above(on) or Below(off) subsetting level*** check box (Above Box) is used to display data above the subsetting level or below the subsetting level. For example, to display a volumetric subset of all concentrations greater than or equal to 1 ppm, set subsetting level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the sequential_subset of 1 ppm and below, simply turn the Above Box off (no check in box).

The ***Exponentiated Values*** toggle makes another type-in field visible which convert real units to logarithmic units for you automatically.

The ***subsetting level*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default subsetting level value is the arithmetic average of the minimum and maximum values in the iso component. If your input data has been kriged with log processing, the values here will be the Log of your input data.

Legend



General Module Function

The Legend module is used to place a color scale bar in the viewer window. The Legend shows the relationship between the selected data component for a particular module and the colors shown in the Viewer. For this reason, the Legend's RED input port must be connected to the RED output port of a module which is connected to the Viewer and is generally the dominant colored object in view.

Many modules with red output ports have a selector to choose which ONE of the nodal or cell data components are to be used for coloring. The name of the selected data component will be displayed as the Title of the Legend if the Label Options are set to Automatic (default).

If the data component to be viewed is either Geo_Layer or Material_ID (for models where the grid is based upon geology), the Brown-Grey-Light Brown-Beige output port from Krig_3D_Geology must also be connected to Legend to provide the Geologic Layer (or material) names for automatic labeling. When this port is connected it will have no affect if any other data component is selected.

The minimum and maximum values are taken from the data input as defined in the datamap. Labels can be placed at user defined intervals along the color scale bar. Labels can consist of user input alphanumerical values or automatically determined numerical values. Legend has an output port which connects to the Viewer module.

Module Input Ports

Legend has three input ports.

1. (Red) : This port receives range and datamap information from any single module. The module connected to this port should contain the data that the Legend references. All modules that connect to the Viewer have a compatible output port.
2. (Brown-Grey-Light Brown-Beige) : This port receives geology info from those modules that read geologic data.
3. (Blue-Red-Blue-Beige) : This port is found on only a few modules in EVS/MVS. The modules with this port are contour_data, isolines, Datamap_Editor, and Legend.

These modules pass data corresponding to one or more subsetting levels (or solid contour levels). When this data is passed to Legend, it is used to set the precise break points in the legend where labeling will occur.

Module Output Ports

Legend has three output ports.

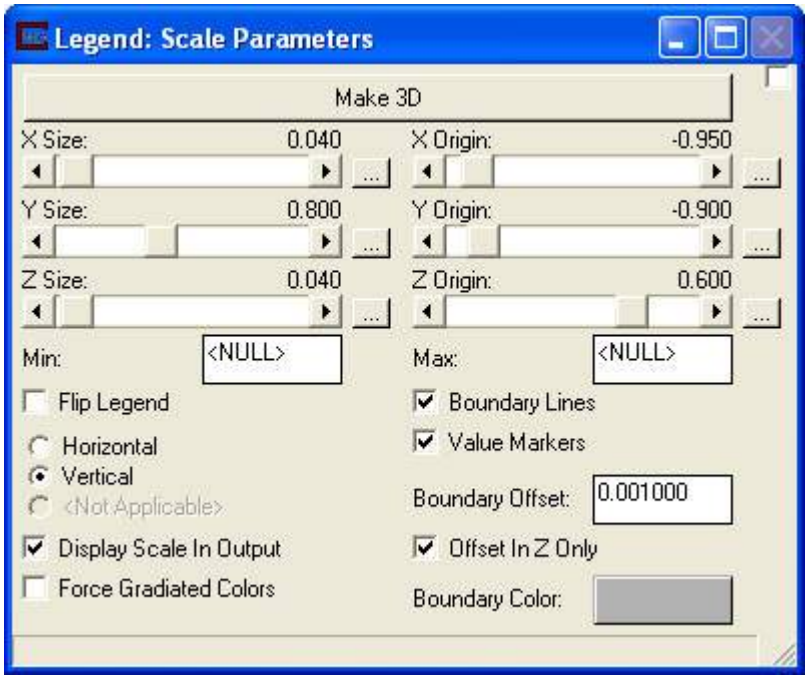
1. The bluish-grey port is the Legend title. Usually this is the analyte name.

2. The blue-black (field) port allows the colored legend portion to be texture mapped with modules such as contour_texture.
3. The red port connects directly to the viewer.

Module Control Panel

The primary control panel for Legend is shown in the figure above. This panel is used to spawn two control subpanels. The *Accept All Current Values* push button is used to apply all the currently selected scale and label settings.

The "Run" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed. When it is off, the module will not run.



The ***Scale Parameters*** subpanel is shown above. The *Make 3D* button at the top changes the legend from non-transformable 2D to a fully 3D legend in the space of your model.

The X, Y, & Z *Size* and *Origin* sliders (and their associated type-ins accesible with the "" buttons), allow you to position the legend in 2D (fixed) or 3D transformable space.

The Min / Max type-in boxes are used to override the default range in the color scale. Caution should be used when changing these values because the

Legend module will allow any numbers to be entered into these boxes. The minimum value will be placed at the bottom of the scale bar,the maximum at the top and the scale bar will be divided into equal pieces and labeled based on other settings in the module. Therefore, it is possible to have a data range from 0 to 10 and set the Legend to go from 100 to 500. In order to have the color scale bar labels exactly match the data, verify that the values type into the min and max values match the limits displayed below. If a more asthetically pleasing color scale is desired, either use an upstream clamp on the data before the module feeding Legend, or change the min and max value slightly to get even numbered increments. In the latter case, the color scale bar will not exactly match the data. However, if the data range is large and the changes to min and max are small, the differences should be negligible. The min and max of the input data range are displayed at the bottom of the control subpanel.

The *Boundary Lines* toggle specifies whether or not a boundary box is to be drawn around the Legend scale bar. A check in the box, the default, indicates that the boundary is to be displayed.

The *Value Markers* toggle specifies whether or not a tick mark is to be drawn at each label location.

The *BoundaryOffset* type-in offsets the labels from the Legend.

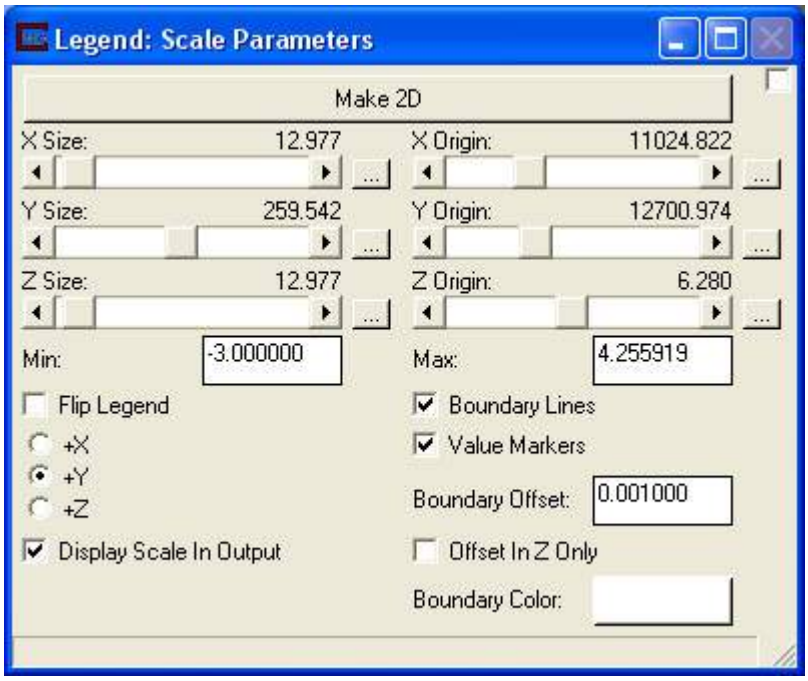
The *Offset in Z Only* toggle specifies whether (when on) to offset the boundary only in the z direction.

The *Boundary Color* button is used to set the color of the Legend boundary.

The *Value Markers* toggle specifies whether or not a tick mark is to be drawn at each label location.

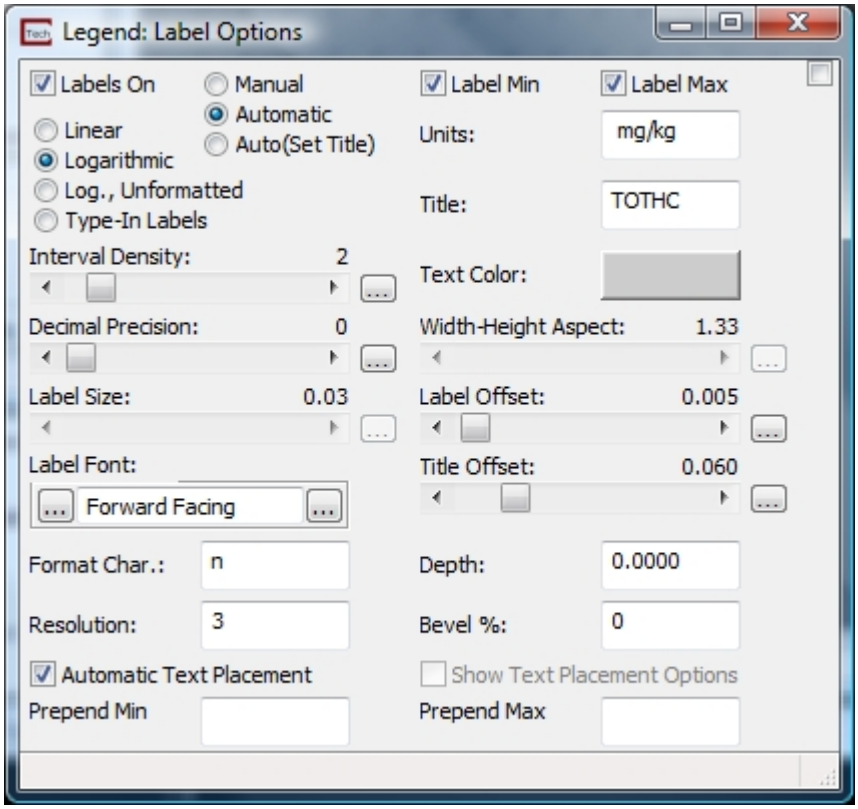
The *Flip Legend* toggle inverts the legend putting high values at the bottom.

The *Display Scale in Output* toggle specifies whether or not to output the colored portion of the legend in the Legend's red port. This should be on when you plan to texture map the legend.



The *Horizontal - Vertical* radio buttons determine the orientation of the legend. When the legend is 3D, this panel changes (as shown above) and you can choose whether the lengend is aligned with the X, Y, or Z axes.

The *Force Gradiated Colors* toggle will cause the Legend to display gradiated colors instead of solid color breaks when contour data is passed in.



The ***LabelOptions*** subpanel is shown above.

The **Labels On** toggle is used to specify whether or not the data labels are to be displayed with the Legend. A check in the box, the default, indicates that they are to be displayed.

The ***Manual*** radio selection specifies that the user will:

- set how the data was processed,
- set the title, and
- set the units.

The ***Automatic*** radio selection specifies whether or not the legend should automatically determine log vs. linear, apply the data component name as a title, and apply the units specified in the input field.
The ***Auto(Set Title)*** radio selection behaves the same as the Automatic radio selection but will use a user specified title.

The ***Label Min*** toggle places a label at the bottom (min) of the data. This only applies to linear or log unformatted.

The ***Label Max*** toggle places a label at the top (max) of the data. This only applies to linear or log unformatted.

The **Interval Density** slider is used to specify the number of label increments to display in the color scale bar. The interval density has different meanings depending on whether logarithmic or linear is chosen in the Main Legend Parameters control subpanel. If linear is chosen, the label density is the number of increments between the min value and the max value. The maximum label density is 31 and the number of labels will be the label interval plus 1. If logrithmic is chosen, the label density is the number of increments per decade. The maximum label density value allowed is 3 (however, any number larger than 3, up to 31, can be chosen and the display will display the same as if 3 was chosen) and the number of data labels displayed will be the number of decades times label density plus 1.

The **Resolution** type-in specifies the number of decimal places for the numbers. Together with the Format Char you can create a wide variety of legend labels.

The **Format Char***.:* type-in specifies how to format the numbers on the legend. The following values can be used:

> 'r;n' (default): Number format: The value is converted to a string of the form "-d,ddd,ddd.ddd..." The symbols used for thousands and decimal separators are determined in the Regional Settings of the Windows Control Panel.
>
> 'r;f': Fixed format: The value is converted to a string of the form "-ddddddd.ddd"
>
> 'r;e': Scientific Notation: The value is converted to a string of the form "-d.ddd...E+ddd". The resulting string starts with a minus sign if the number is negative. One digit always precedes the decimal point.
>
> 'r;g': General format: The value is converted to the shortest possible decimal string using fixed or scientific format.

'r;m': Money format: The value is converted to a string that represents a currency amount. The format is controlled by the Regional Settings in the Windows Control Panel, and will include the currency symbol, thousands separator, and decimal separator as specified in the Regional Settings.

The **Legend Type** radio buttons specify if the data is linear, logarithmically transformed (formatted or unformatted) or has Type-In Labels. Logarithmic is the default. Log Unformatted is similar to linear, but with log values. Type-in lables opens a new window where you can type in any label for each interval.

The **Label Size,*Label Offset*** and ***Title Offset*** sliders sets the label and title height and position relative to the legend length.

The **Label*Font*** selector sets the font used for all labeling. For more information on Font Selection Click Here.

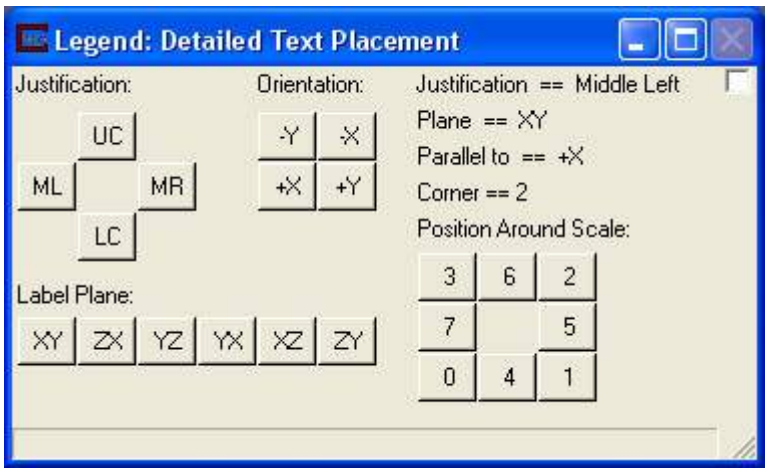
Depth and Bevel are options for using true 3D beveled text. See Text_3D for a complete description of these.

The **Width-Height** *Aspect* slider determines the aspect ratio of the label text. A small width to height ratio creates narrow text and a large ratio creates wide text. The default is 0.90 and the range is from 0.20 to 1.50. The decimal Prec. slider is used to indicate the number of decimal places to display in the labels. The default in 1 and the range is from 0 to 6. Precision only applies to linear ranges. The Label font slider is used to determine which font style is used for the labels. The default is font 0 and the range is from 0 to 5. The font styles are not currently documented.

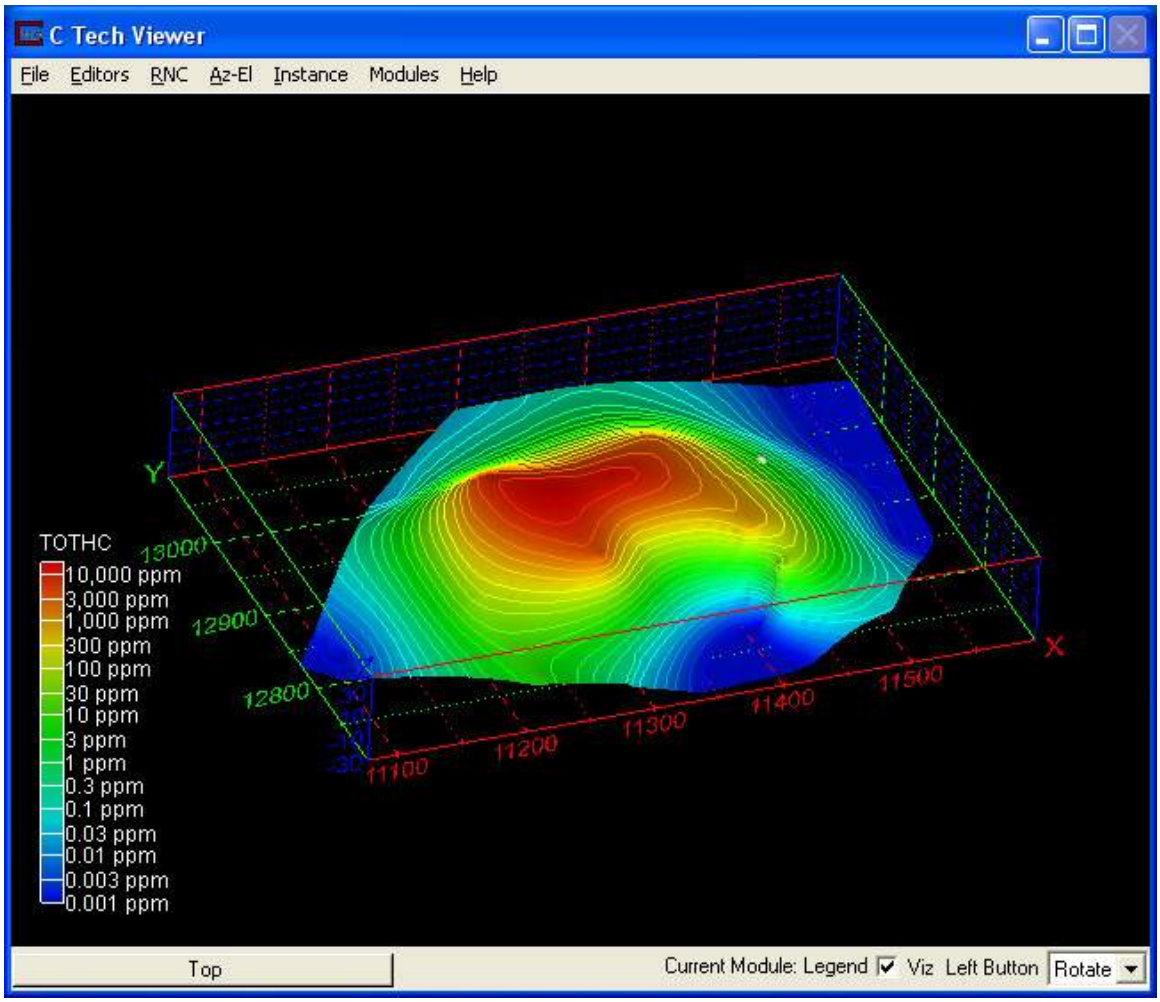
The ***Text Color*** button is used to set the color of the Legend labels.

If you turn off the *Automatic Text Placement* toggle and turn on the *Show Text Placement Options* toggle, another window (shown below) appears with options for more specific text justification.

The ***Prepend Min*** and ***Prepend Max*** will prepend the selected text to either the min value or the max value label in the legend.



Below is an example of a color scale bar produced by the Legend module.



Related Modules

-> Viewer

axes

General Module Function

The axes module is used to place 3D axes in the viewer scaled by the model data and/or user defined limits. Axes accepts data from many of the Subsetting and Processing modules and outputs directly to the viewer. Data passed to Axes should come from modules which have scaled or transformed the mesh data, for example Explode_and_Scale. Axes generated by axes and displayed in the viewer are transformable with other objects in the viewer.

The User interface to axes is very comprehensive. Each coordinate direction axis can be individually controlled. Axis labels and tick marks for each axes can be specified. The label font, label precision, label orientation, and other label parameters are all user specified. Many of the parameters do not have default values that will produce the desired results because many variables control how the axes should be defined.

axes requires a field input to position and size the axes. If you disconnect the (blue/black) field input port, you no longer lose the axes bounds values and your axes remain in place. This is useful when field data changes in an animation so that you don't constantly recreate the axes.

Also, the size of text and tick marks is based on a percentage of the x-y-z extent of the input field. This now allows you to set the extent of one or more axes to zero so you can have a scale of only one or two dimensions.

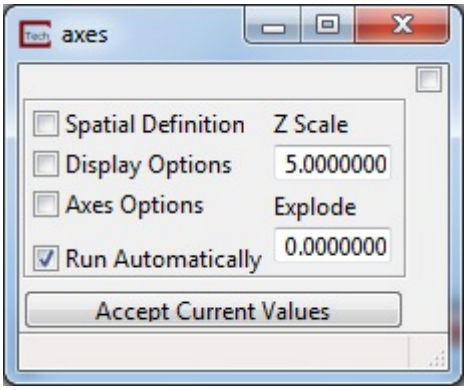
Module Input Ports

The axes module is shown above. The module has five input ports:

1. in_view (Purple) : This port accepts the output of the viewer directly. It will draw the axes around everything displayed in the viewer. This port will only cause the module to run when the port is connected or when the "Accept Current Values" button is pressed. If the models coordinate extents are going to change often then another input port should be used.
2. objects_in (Red) : This port accepts any number of (Red) output ports from other modules. When any of those modules are run the axes module will run as well.
3. meshs_in (Blue/Black) : This port accepts any number of (Blue/Black) output ports from other modules. When any of those modules are run the axes module will run as well.
4. explode (Grey/Green) : This port accepts a float value representing the explode distance from Explode_and_Scale. If you have an explode distance set to anything but 0, the Z axis tick labels are not printed.
5. z_scale (Grey/Brown) : This port accepts a float value representing Z exaggeration of the model from modules like Explode_and_Scale to ensure that the Z axis is correctly labeled.

Module Output Ports

axes has one output port which sends the axes to the viewer for display.



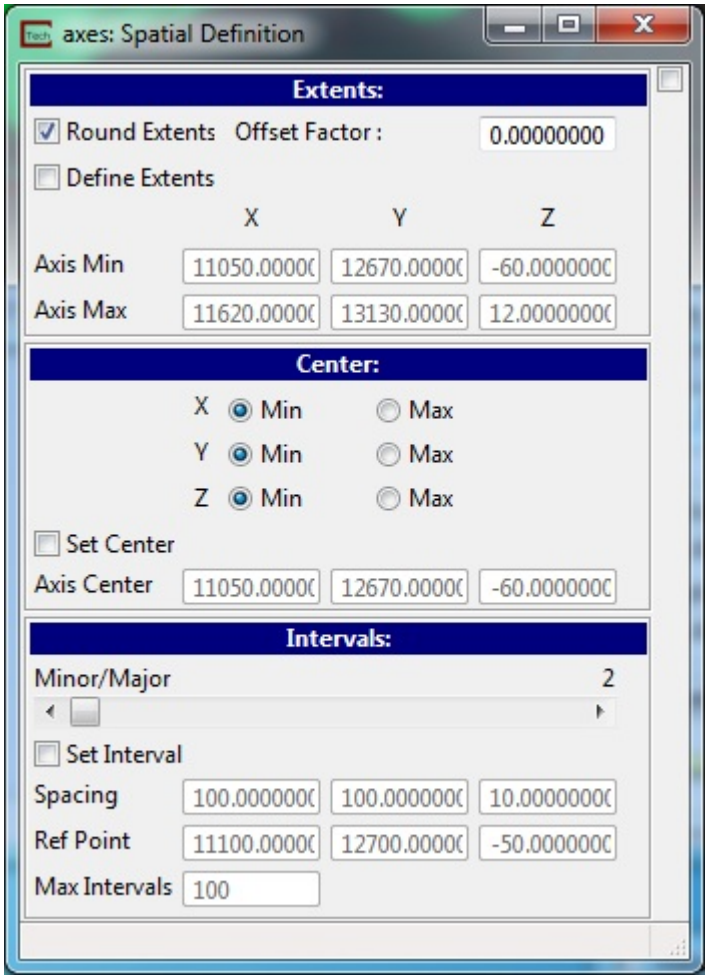
Module Control Panel

The main control panel for axes is shown above. The control parameters are divided into three panels, each made visible by toggles on this main panel.

1. Spatial Definition
2. Display Options
3. Axes Options

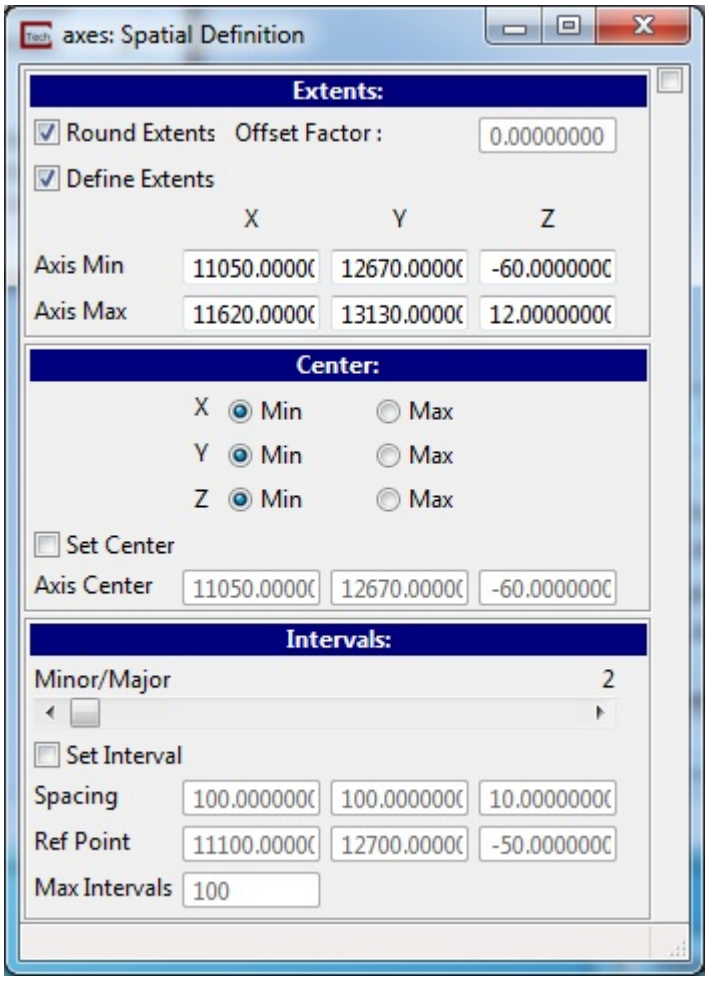
The sections below discuss each of these panels.

Spatial Definition

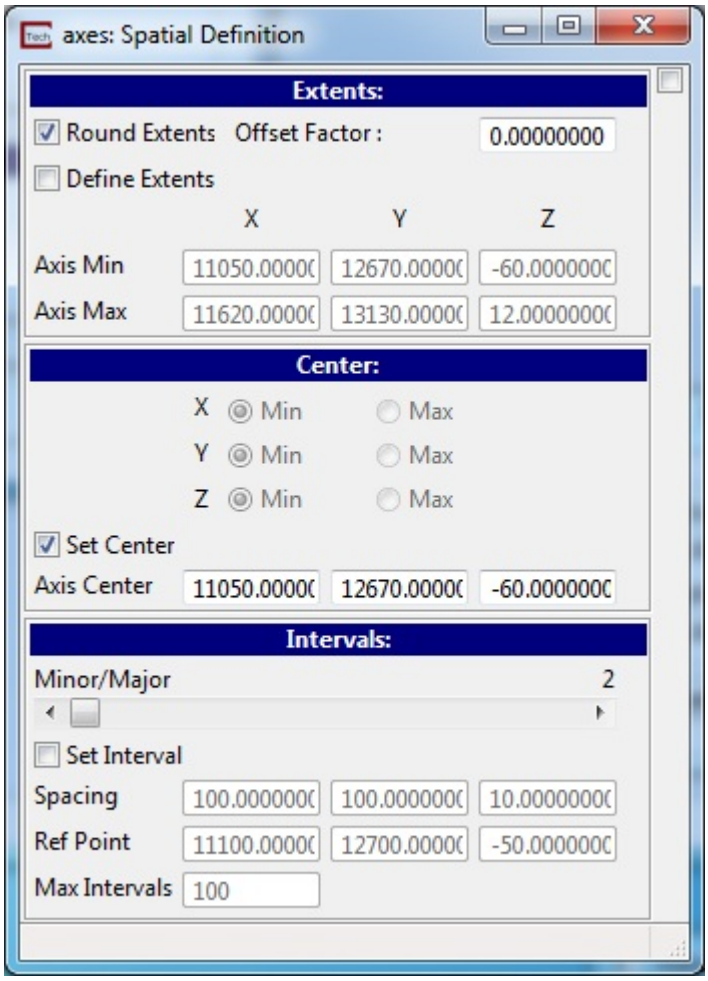


Normally the extents of your axes are determined by one or more modules connected to axes' blue-black and/or red input ports. However there are options that allow you to modify this behaviour. The options include:

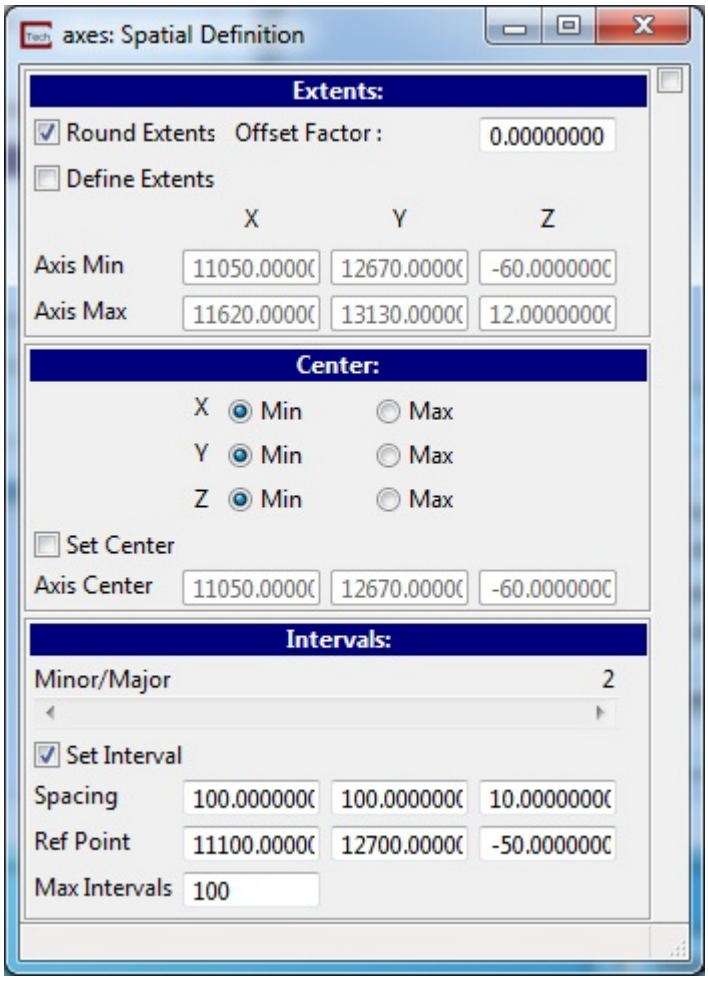
1. *Round Extents* increases the extents of the model to have the labeling start at logical coordinates rounded down at the minimum of x, y, & z and rounded up at the maximum.



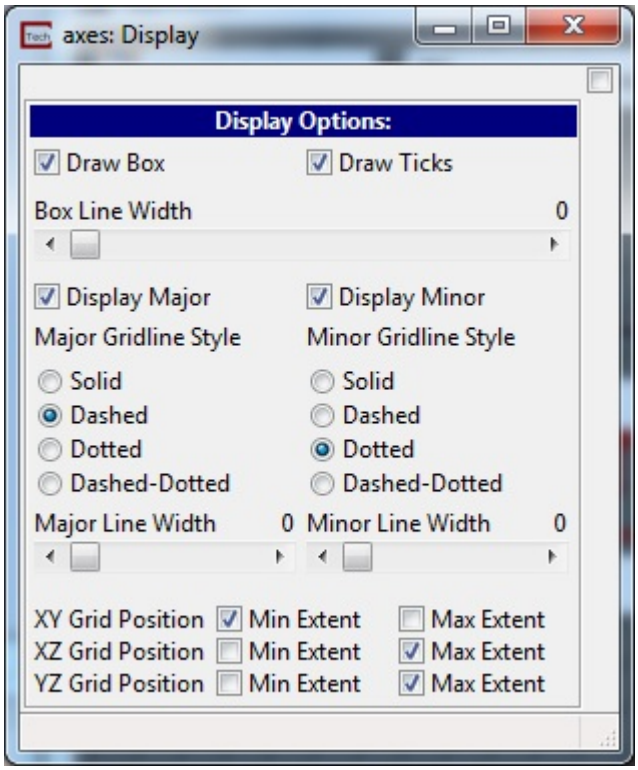
2. *Define Extents* allows you to specify any extents you wish instead of having the extents determined by module inputs.
3. *Offset Factor* is a fraction (of your total extents) to offset the extents beyond the data extents. At 0.1 it would make axes 10% bigger in all directions.



4. *Set Center* allows you to specify any center you wish instead of having the center determined by module inputs and Min/Max radio selectors.



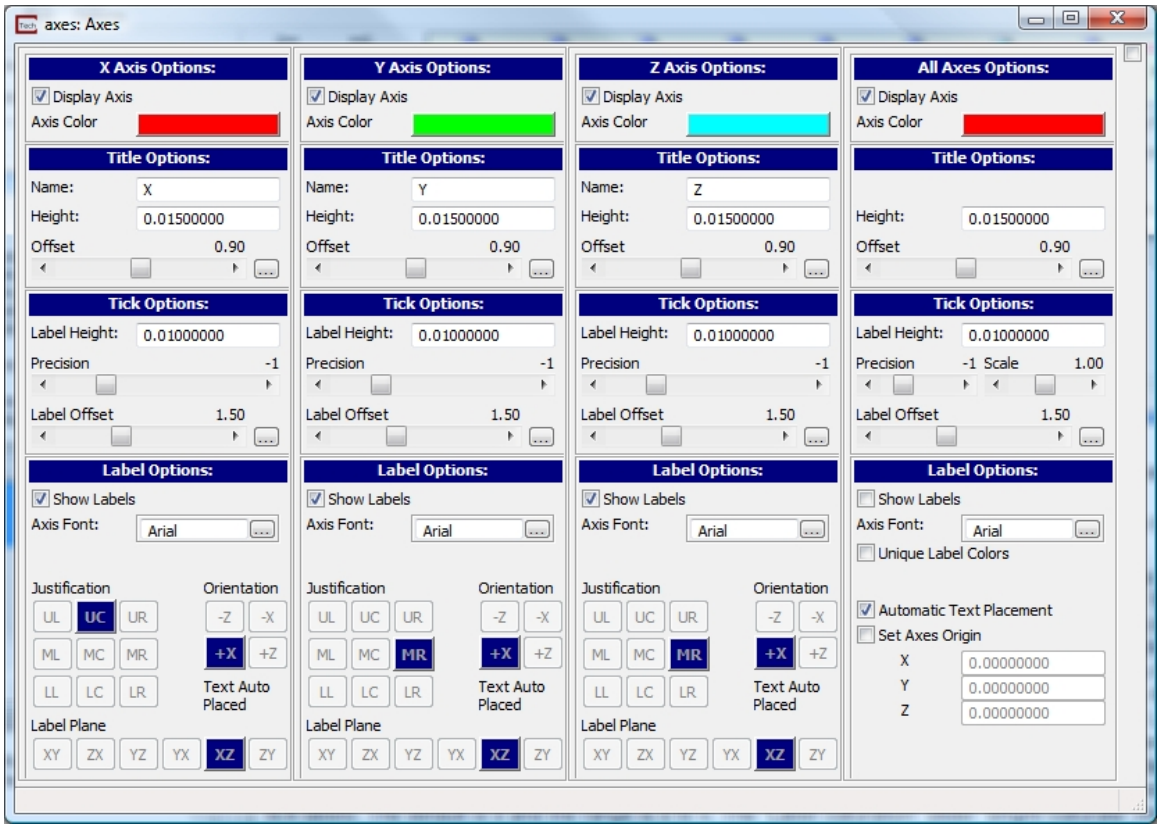
5. *Set Interval* allows you to override the expert system's tick spacing, reference point, and the maximum number of intervals.
6. *Minor/Major* is a slider to determine how many minor intervals for every major interval.



Display Options

The display options control several aspects of the appearance of the axes, tick marks, lines, boxes and more.

1. The *DrawBox* check box is used to specify if the axes should be drawn as a box around the data domain or as single X, Y, and Z lines.
2. *Box Line Width* determines the thickness of the box lines. 0 (and 1) give a single pixel wide line and larger numbers give wider lines. Note that only odd numbers 1, 3, 5, etc. are supported by the Software Renderer.
3. The *Display Major* and *Display Minor* check boxes are used to specify whether the tick marks and labels are displayed (on) or not (off). Both are on by default.
4. *Major (and Minor) Gridline Style* allow you to specify the line styles for the gridlines.
5. *Major (and Minor) Line Width* determines the thickness of the gridlines. 0 (and 1) give a single pixel wide line and larger numbers give wider lines. Note that only odd numbers 1, 3, 5, etc. are supported by the Software Renderer.
6. The remaining 6 toggles allow you to specify which planes of the axes should have gridlines.



AxesOptions

This panel has 4 subpanels that provide control over X, Y, & Z axes and an option for ALL. When parameters are modified in the All subpanel, the corresponding parameter in the X, Y, & Z axes are all modified.

These axis variable control panels consist of check boxes, edit fields and sliders all used to control the display of axes and labels

Under *X (Y, Z or ALL) Axis Options:*

1. The Display Axis toggle determines if that axis is displayed.
2. The Axis Color button lets you set the color for that axis. To have uncolored axes, merely change the All color button to white, black or any other color.

Under *Title Options:*

1. the *Name* edit field is used to specify the text that will appear as the axis label for the axis being edited. The default is the Cartesian axis specifier for the axis being edited (For example, X for the X axis). This label can be any text string and can contain spaces and special characters.
2. *Height* is the height of the Name field as a percentage of the total axes extent.

3. *Offset* is a distance to offset the Name from the corner of the axes box.

Under *Tick Options:*

1. Label*Height* is the height of the Name field as a percentage of the total axes extent.
2. *Precision* is an integer slider that determines how the tick positions (numeric values) will be determined. For values of zero and larger it determines the number of decimal points. If this number is negative, the function attempts to automatically determine the optimal number of decimal points to use based on the value. By changing this negative value, you can control the number of decimal points used in the automatic conversion.
3. *Scale* affects the relative size of the Tick marks and the label offsetting.
4. *Label Offset* is a distance to offset the Tick label from the end of the tick mark.

Under *Label Options:*

1) The *Justification* push buttons are used to set the location of the label relative to the end of the tick mark to be labeled. The layout of these push buttons corresponds to the location of the tick mark relative to the labels location. The label can exist in any of nine locations relative to the tick mark. For example:

- Selecting MC (middle center) the center of the label is placed directly over the end of the tick mark.

- Selecting UC (upper center) places the center of the label directly below the end of the tick mark (tick mark is above the center of the label).

- Selecting LL (lower left) places the left end of the label just above the end of the tick mark.

- Selecting MR (middle right) places the right end of the label directly to the left of the tick mark.

The other justification push buttons act in a similar manner.

2)The Orientation push buttons are used to specify which axis the labels should parallel. These push buttons are used in conjunction with the Label Plane push buttons. The Orientation push buttons make more sense once the Label Plane push buttons have been explained.

3) The *Label Plane* push buttons define which plane the label is to reside in. There are 6 possible planes corresponding to the six faces of a cube. The planes are defined as parallel to the axes in the push button name with the order of the plane names defining the normal vector or positive side of the plane (for example, the XY label plane is parallel to the X and Y axes and the normal vector is in the positive Z direction). The normal vector or positive side of the plane determines whether a label is forward and right side up when viewed from above or below. Once the plane for the label is chosen, the orientation of the label can be defined. In any given label plane, there

are four possible label orientations. For example, in the XZ plane the labels can be parallel to the +X, +Z, -X or -Z axis. +X will cause the labels to parallel the X axis in the XZ plane and read from left to right, right side up. -X create labels parallel to the -X axis which, when viewed from the Azimuth 180 - elevation 0, will appear upside down.

The orientation of the label also affects how the justification push buttons affect the label. The justification push buttons move the text around a label box relative to the end of the tick mark. The label box is oriented parallel to the orientation push button selection. Therefore, if +Z orientation is chosen, the UC justification location appear to move the label to the middle left of the tick mark in Cartesian coordinates, when viewed normal to the XZ plane. The LL will appear to move the label to the lower right of the tick mark when viewed normal to the XZ plane.

Under *All Axes Options*

There are four special options:

1. *Unique Label Colors* allows you to have labels with different colors than the axes
2. *Automatic Text Placement* automatically handles the lable options based on centering options selected.
3. *Set Axes Origin* lets you change the labeling (e.g. to make axes with labels starting at zero in x, y and z)

north

General Module Function

The north module is used to place a 3D North Arrow or Rose Compass in the 3D viewer scaled by the model data and/or user defined parameters.

Module Input Ports

The north module has four input ports.

1) in_view (Purple) : This port takes Viewer output and will use the extents of all objects connected to the viewer to place and scale the North arrow or rose.

2) objects_in (Red) : This port takes the renderable output from any module with a red output port and uses the exent of that object to scale and place the North arrow or rose.

3) in_field (Blue-Black) : Accepts mesh data information specifying the model extents.

4) z_scale (Grey-Brown) : This port accepts scaling data from Explode_and_Scale which contains information on the z exaggeration variable.

Module Output Ports

north has one output port which sends its output to the viewer for display.

Module Control Panel

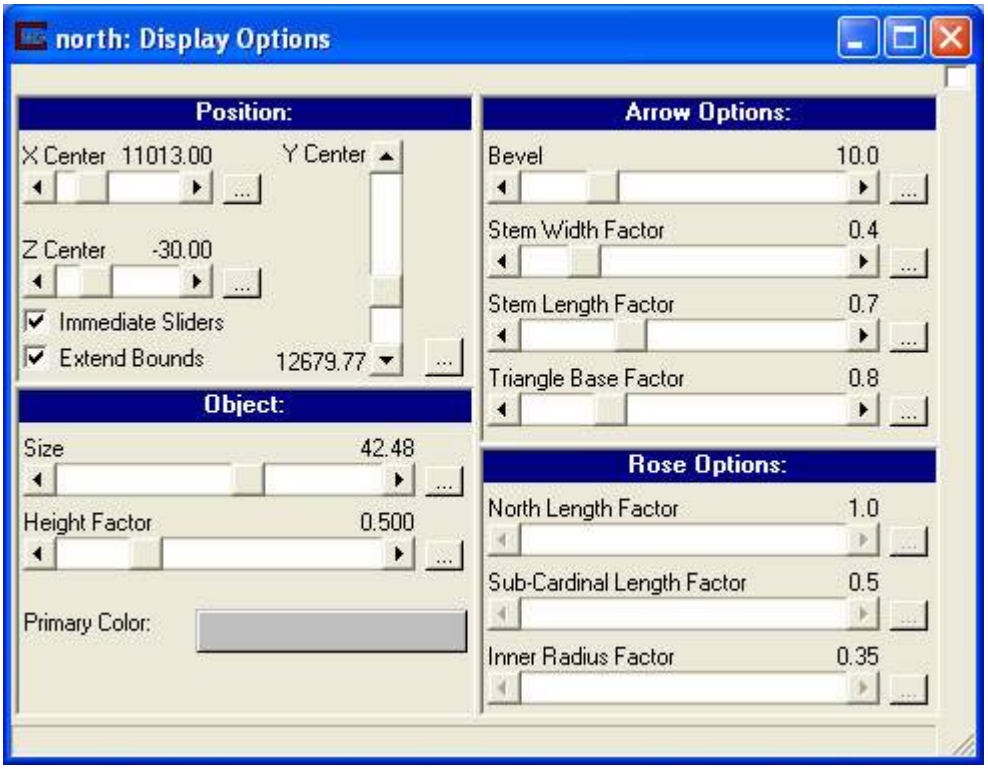
The main control panel for north is shown above.

a. A radio selector lets you choose between "**Arrow**" and "**Compass Rose**" outputs. This selection affects the appearance of other subpanels.

b. **Display Options** lets you adjust scale, positioning and various aspects of the two output options

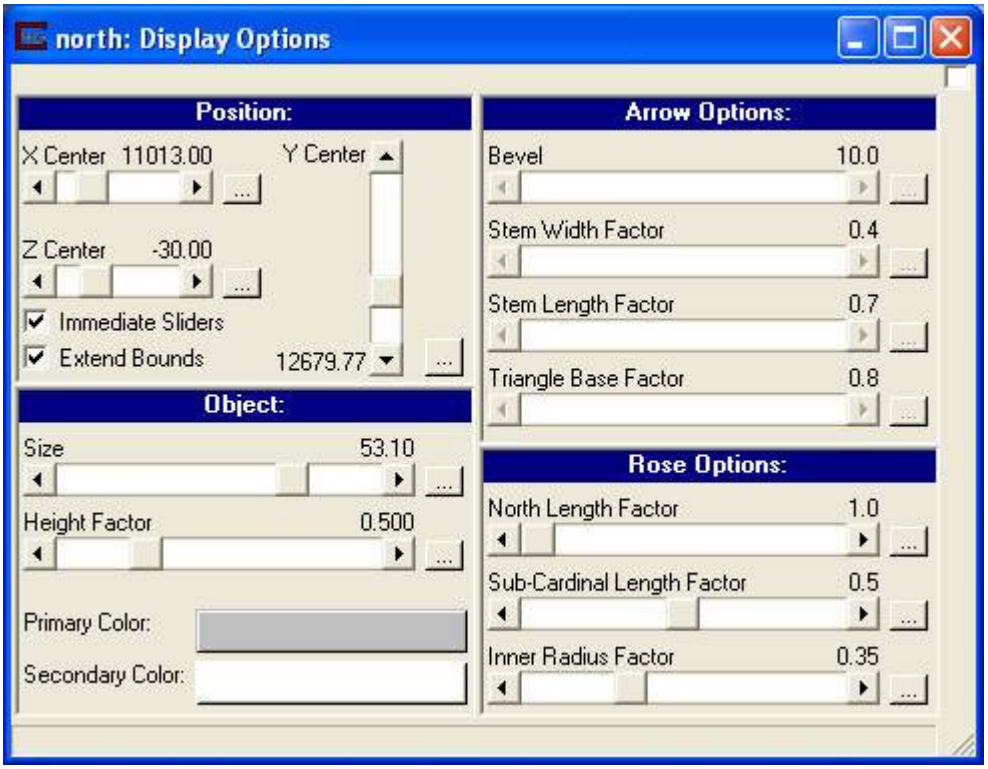
c. **Label Options** lets you adjust labeling scale, positioning and various aspects of the two output options

d. **Z Scale** is a type-in linked to the Z Scale input port. This isn't a critical input, however by having the proper Z Scale of your model you can place north's ouput in true z coordinates.



The **Display Options** panel is shown above for the default *Arrow* option.

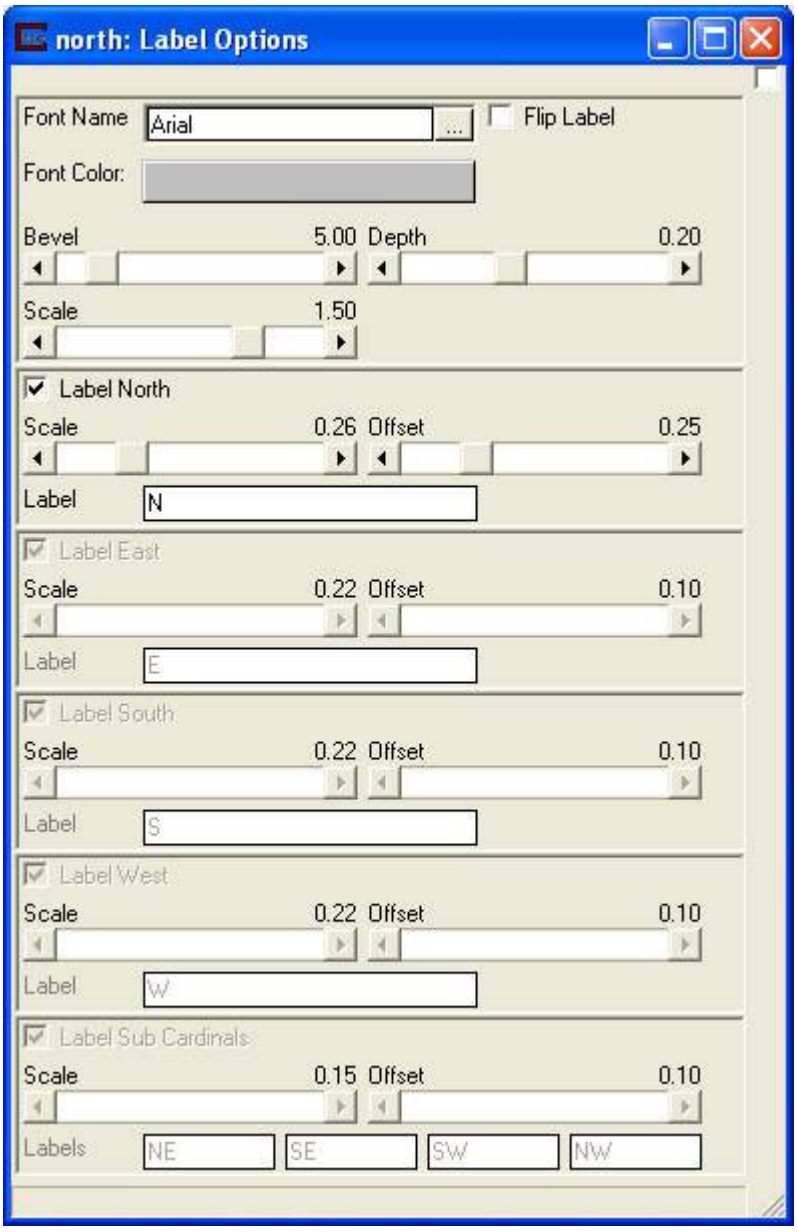
a. **Position** Panel
    a. Three sliders are provide for x, y, and z center of the output
    b. Toggles for *Immediate Sliders* and *Extend Bounds* allow you to place the ouput in real time and to position the annotation outside of the model's extents.
b. **Object** Panel
    a. Size is a scale factor for the entire arrow or "rose compass" object
    b. Height Factor affects the z-height of the arrow or rose.
    c. Primary Color is a button to set the color of most of the objects comprising the output
    d. Secondary Color is a button to set the color of the alternating faces of the compass' rose. To understand this completely make this color red to see what happens.
c. **Arrow Options** Panel
    a. Bevel is a percentage that affects the degree of beveling on the north arrow
    b. Stem Width Factor affects the width of the arrow's stem (shaft).
    c. Stem Length Factor affects the lenght of the arrow's stem (shaft).
    d. Triangle Base Factor affects the width of the arrow's head.



The **Display Options** panel is shown above for the *Compass Rose* option.

n. **Rose Options** Panel

a. North Length Factor allows you to exaggerate the length of the north pointing shaft.
b. Sub-Cardinal Length Factor controls the length of the 45 degree shafts relative to the primary shafts.
c. Inner Radius Factor controls the radius of the solid core section

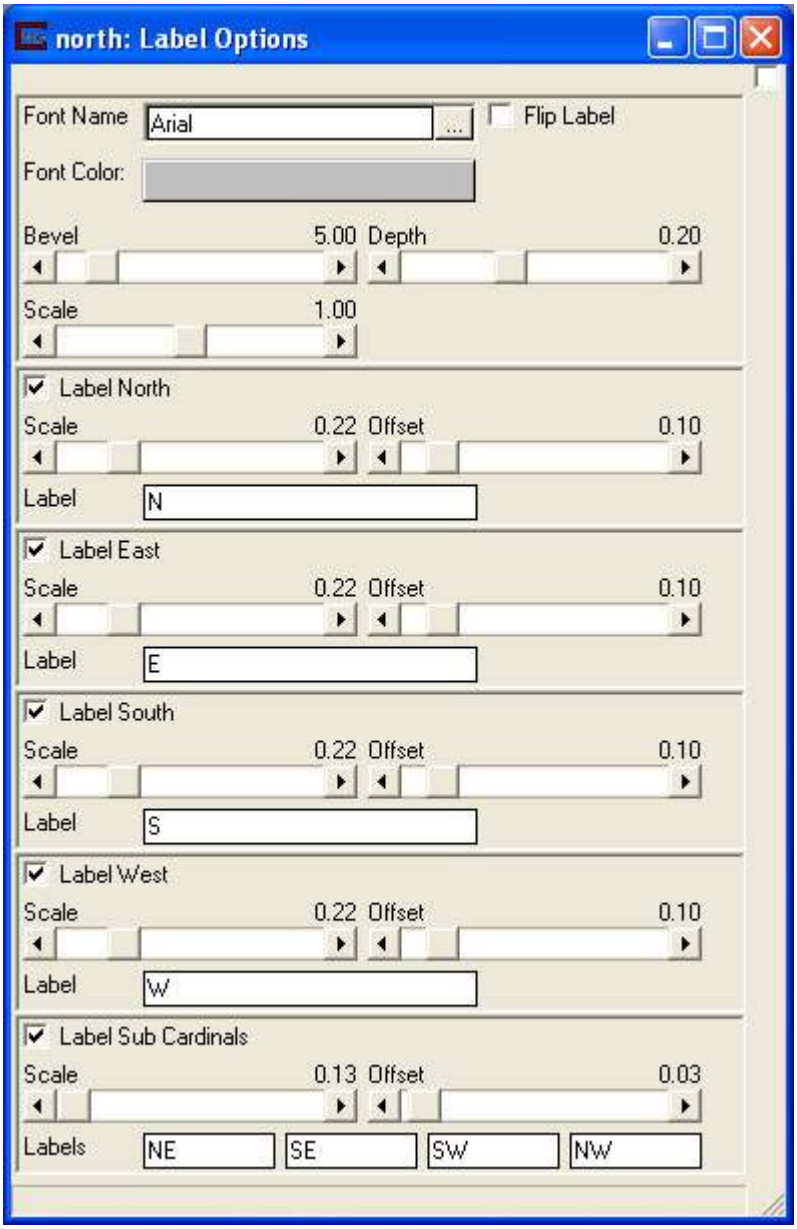


The **LabelOptions** panel is shown above for the default *Arrow* option.

a. **Upper Panel**
   a. Font Name and Color can be specified. Since this module creates 3D objects, forward facing text fonts are not available.
   b. Flip Label makes the text readable from the underside

c. Bevel is a percentage that affects the degree of beveling on the text objects
d. Depth is the height as a percentage of text width
e. Scale affects the size of all text objects.

b. **Label North** Panel (identical to Label East, South and West)
   a. Size sets the size of the North label (N by default) as a percentage of the entire North Arrow (or rose compass) object.
   b. Offset specifies the distance away from the arrow or rose shafts
   c. Label allows you to change the label text (N,E,W,S by default)



The **LabelOptions** panel is shown above for the *Compass Rose* option.

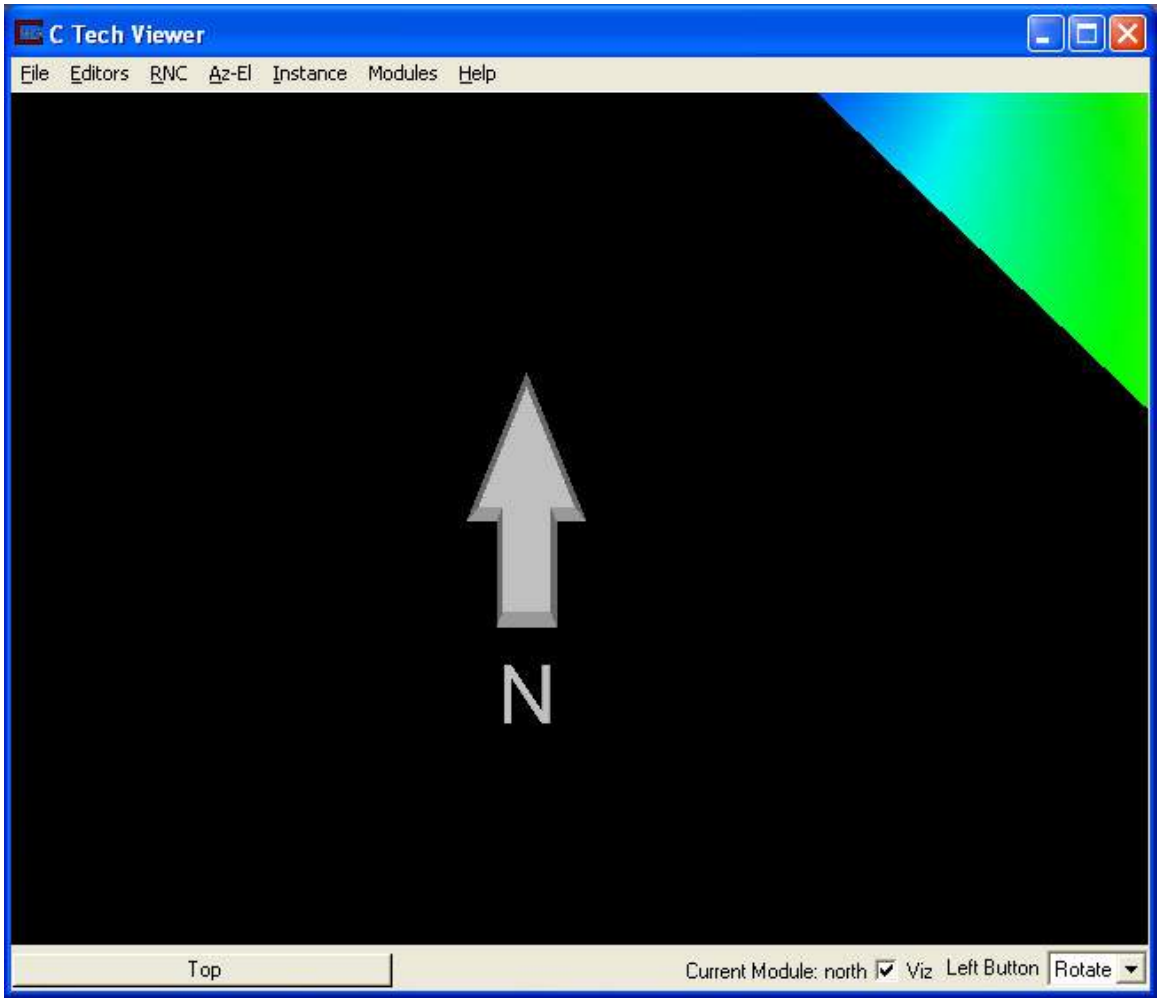
k. **Label Sub Cardinals** Panel

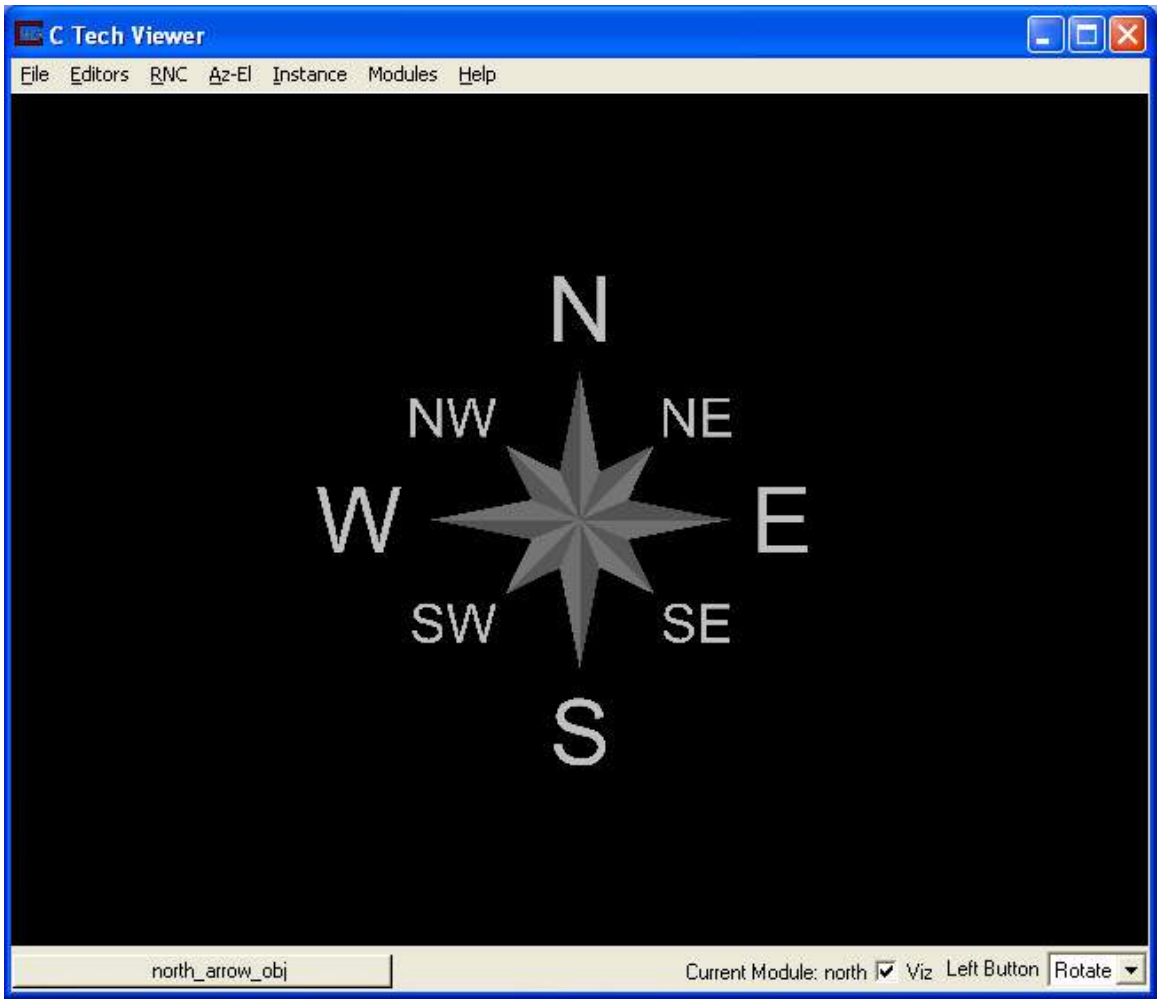
a. Size sets the size of the North label (N by default) as a percentage of the entire North Arrow (or rose compass) object.

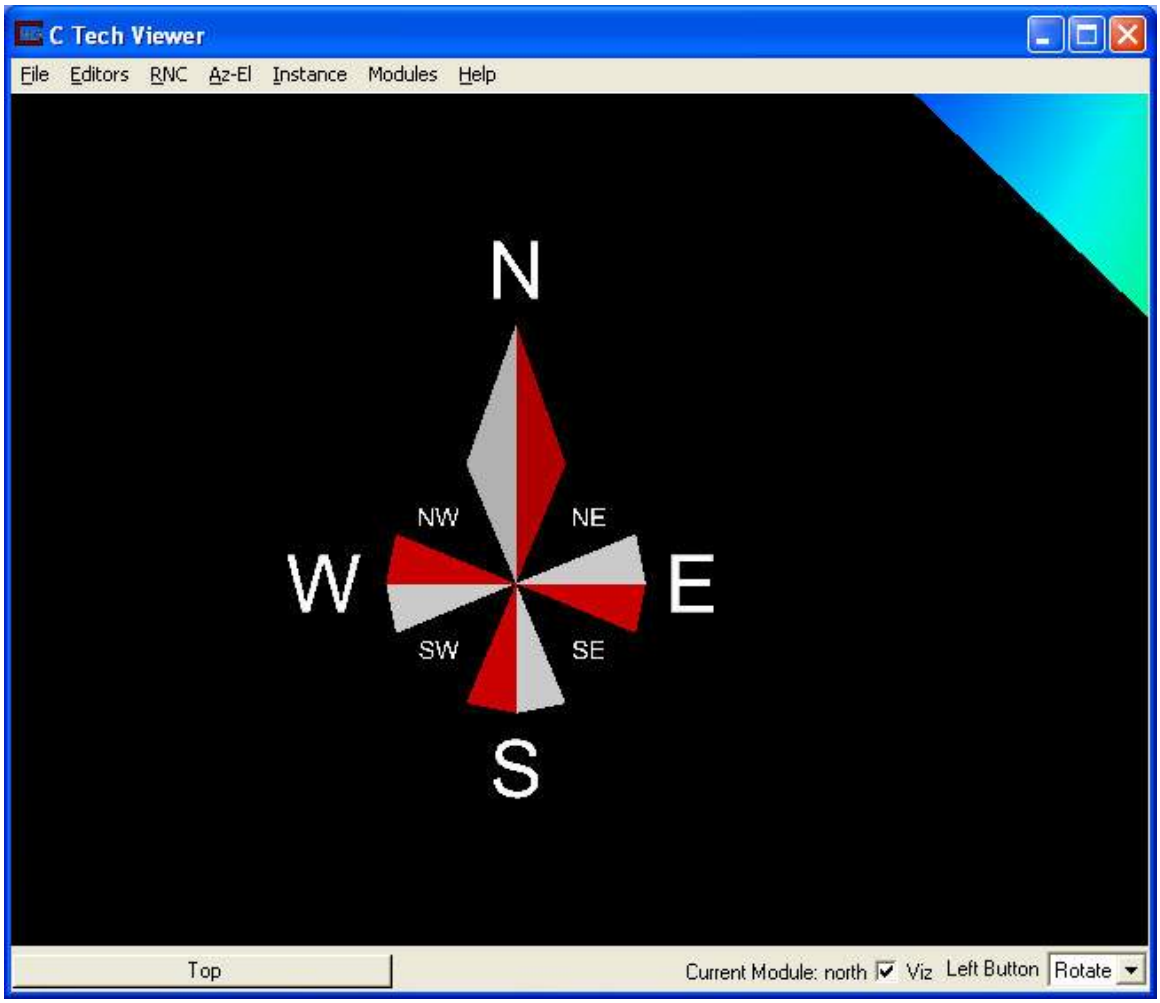
b. Offset specifies the distance away from the arrow or rose shafts

c. Four type-ins allows you to change the label text (NE, SE, SW, NW by default)

EXAMPLES: Below are several examples of output from the north module:




C Tech Viewer
File Editors RNC Az-El Instance Modules Help
N
NW
NE
W
E
SW
SE
S
north_arrow_obj
Current Module: north
Viz
Left Button
Rotate



add_logo

General Module Function

The add_logo module is used to place a logo or other graphic object in the Viewer's non-transformable overlay. It is extremely easy to use. There are sliders to adjust size and position and a button to select the image file to use as a logo.

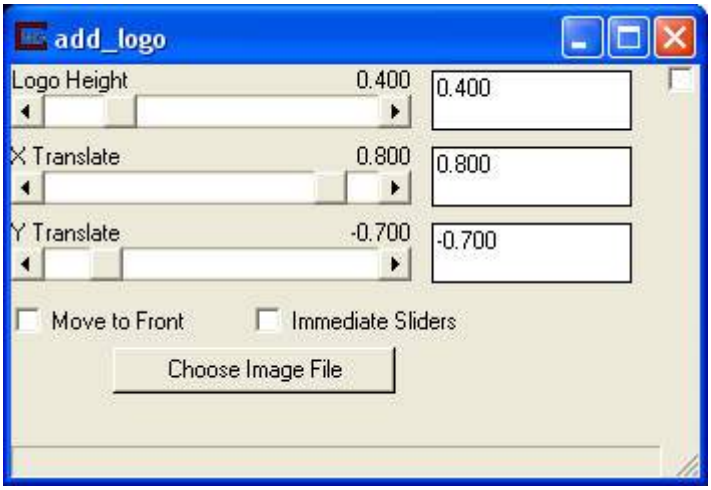
Module Input Ports

Add_logo has only one input port that **must be** connected in order to get any output. This input is the purple output port from the Viewer. This connection is used to maintain a proper aspect ratio for the logo when the Viewer is resized.

Note: You must use the same aspect ratio in Output_Images or a 4DIM file in order for the placement and height/width ration of the logo to be maintained.

Module Output Ports

Add_logo has only one output port which is a non-transformable renderable representation of the logo.



Module Control Panel

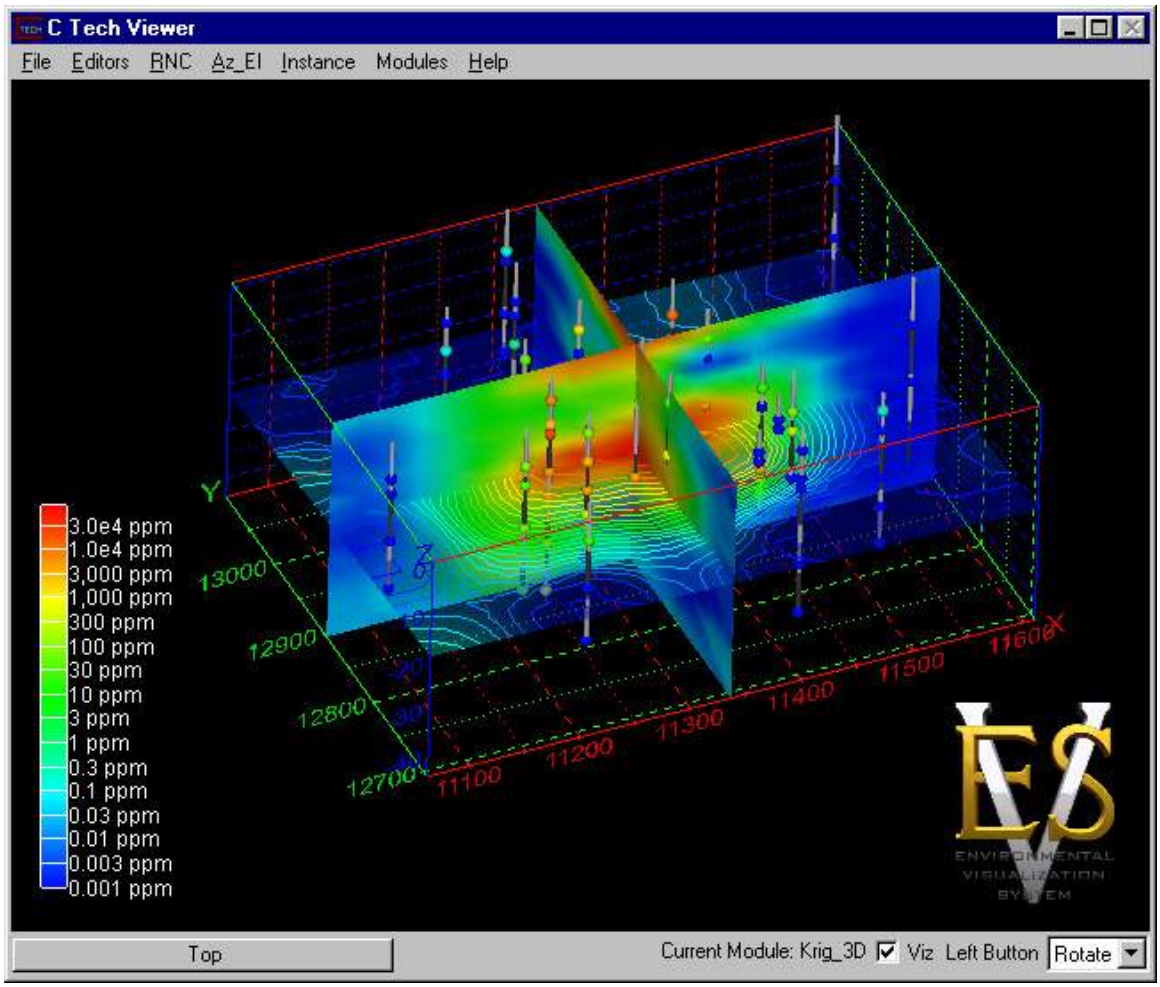
The control panel for add_logo is shown in the figure above. The three sliders allow you to place and scale the logo.

The "Move to Front" toggle places the logo on top of all 3D objects in the view, as opposed to the default position which is behind all objects.

The immediate toggle makes the sliders function in real-time.

The "Choose Image File" button allows you to specify the logo image file.

The image below is a demonstration of the add_logo module reading the image file evs_443_black_logo.png.



Titles

Attention:Click here for View_Title help .

General Module Function

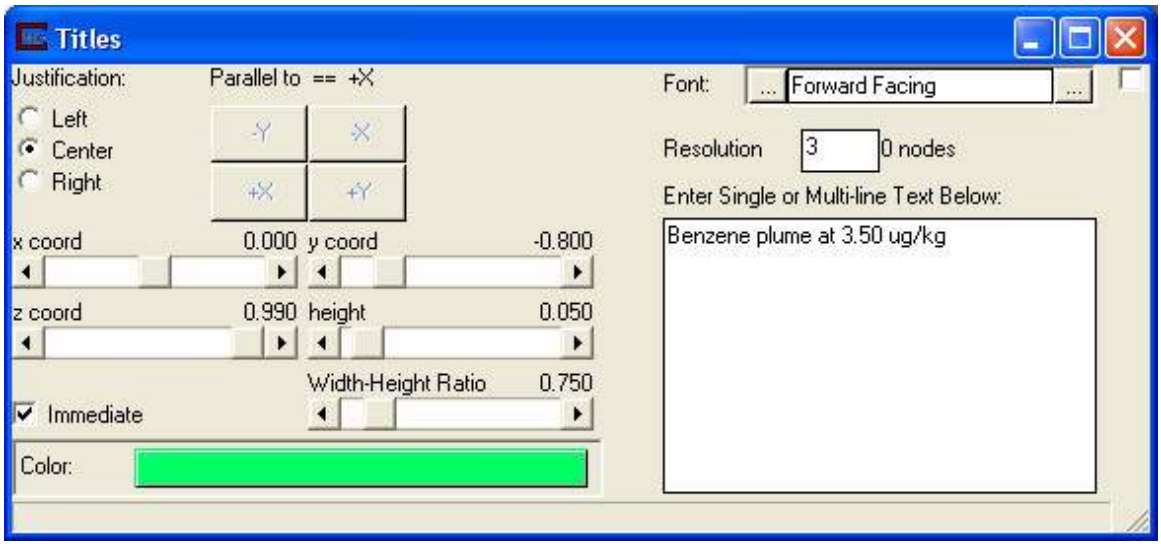
Titles connects to the red port on the Viewer (as does Color_Legend) and provides a straightforward means to place text in the Viewer. By using the red port, the text is not transformed by Viewer transformations and is positioned using sliders in the Titles user interface.

Module Input Ports

The string to be rendered.

Module Output Ports

Titles has only one output port. This port outputs labels to the Viewer.



Module Control Panel

The control panel for Titles is shown in the figure above.

X_coord and Y_coord position the labels in a window that is +/- 1.2 in both x and y (for a square viewer).

The Left, Center, and Right radio buttons determine the justification of the text.

Height is in the same window units.

Width-Height Ratio allows you to stretch or compress the font to adjust for non-square Viewer induced distortions.

The z_coord allows you to place text in front or behind other 3D objects in the viewer.

The Color button (default green) determine the title color.

Font: Titles supports all Windows True-type fonts, special forward facing text fonts and EVS vector stroke fonts. For more information on Font Selection Click Here.

Resolution: is the smoothness of curves used to form the letters. Higher resolution requires more creation time, more memory and slower rendering, however the letters look better.

The +X, -X, +Y, -Y buttons allow you to control the rotation of the text.

place_text

General Module Function

place_text replaces both Text3D and MultiText3D and provides a means to interactively place 2D and 3D renderable text strings or to read an EMT file to place the text.

Module Input Ports

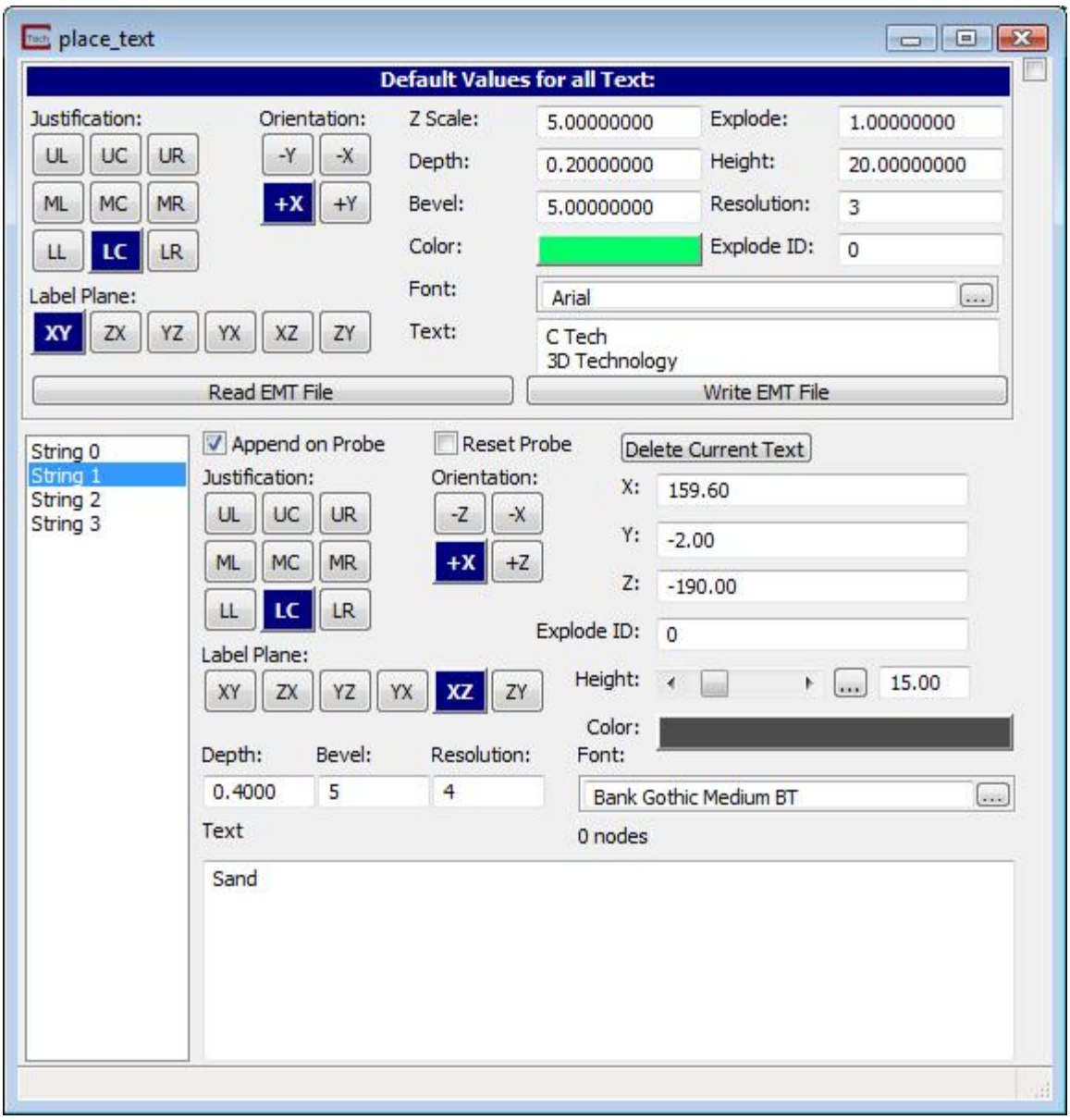
place_text has tthree input ports

1. The first (purple port) connects to the Viewers only output port.
2. The second (grey-brown) connects to Explode_and_Scale to inherit the Z_Exaggeration factor.
3. The third (grey-green) connects to Explode_and_Scale to inherit the explode distance

Module Output Ports

place_text has three output ports.

1. The first is a pass-through for the Z-Exaggeration
2. The second is a pass-through for explode distance
3. The third is a red port which sends the renderable text strings to the EVS viewer.



Module Control Panel

The control panel for place_text is shown above. The panel is in two sections. The first has the default values for all new text strings that you create by probing. The lower section has many of the same parameters, but allows you to edit the values for these parameters for each string whether the string was created by probing or input from an EMT file.

- **Justification** is identical to the controls in post_samples labeling.
- **Orientation** is identical to the controls in post_samples labeling.
- **Label Plane** Used in conjunction with the orient and angle subobjects, specifies the transformation for the character string: the choices are XY (analogous to horizontal/ground surface; etc. These are identical to the controls in post_samples Labeling.

- **X_coord, y_coord and z_coord** are type-ins that are populated by probing (Alt+Left_Mouse) on any object in your Viewer. You can also modify the 3D location of the strings using these.
- The **Color** button brings up a color selector window that allows you to specify any color for the text.
- **Z Scale: sets the z exaggeration factor for the z coordinates to place the text**
- **Explode:**allows z coordinates to be exploded with the same logic used to explode geologic layers. This makes it much easier to have labels on layers which move as you explode.
- **Depth**: causes the text to be extruded as a truly 3D object. If set to zero (0.00) the text is flat.
- **Bevel**: causes the 3D text to have beveled edges. Acceptable values are font dependent. If the value is too large the faces of some of the letters will appear to have holes and connectivity problems.
- **Resolution**: is the smoothness of curves used to form the letters. Higher resolution requires more creation time, more memory and slower rendering, however the letters look better.
- **Orient** Used in conjunction with the plane and angle subobjects, specifies the transformation for the character string: choices are left to right; right to left; down; and arbitrary (which requires using the angle slider).
- The **Height** slider specifies the height of the character; the default character height is set based on the extents of the input field. It is used as a scale factor for both the height and width of the character.
- The angle and path are adjusted by changing the orientation and plane. Spacing is determined by the font you choose. Each font's specific spacing is taken into account (this is actually part of the font itself).

EMT FILE FORMAT:

Lines beginning with "#" are comments

Lines beginning with "FONT" are font specification lines (more later)

Lines beginning with "END" specify the end of the file (this is optional, but if you want to have anything after the last command or data line, precede it with an "END" statement.

All other lines are DATA lines specifying the x-y-z coordinates of a string and the text for that string.

Blank lines are ignored.

The FONT specification lines contain the following information:

Size: The font size is the height of a typical Capitol letter in true user units

Justification: The justification options are the same as in post_samples labeling.

Plane: The plane options are the same as in post_samples labeling.

Orientation: The orientation options are the same as in post_samples labeling.

Red, Green, Blue: These 3 numbers determine the font color.

Resolution: The resolution parameter is the same as in post_samples labeling.

Depth: The parameter is the same as in post_samples labeling.

Bevel%: The Bevel percentage is the same as in post_samples labeling.

Font Face: The Font Face options are the same as in post_samples labeling.

The DATA lines contain four columns of information:

X coordinate

Y coordinate

Z coordinate

Text: Everything on the line after the z coordinate (and trailing spaces) is the text to be placed at the above coordinate.

```
# FONT Size Just. Plane Orient R G B Resolution Depth Bevel% Font Face
FONT, 4, MC, XZ, +X, 0.8, 0.8, 0.8, 3, 0, 0, Arial
# X, Y, Z, Bore
11566.34, 12850.59, 8.5, B-30
11586.34, 13050.59, 12.5, B-31
11381.7, 12747.5, 2.5, B-33
11414.4, 12781.1, 3, B-34
11410.29, 12724.69, 4.5, B-4
11427, 12780.9, 7.5, B-42
11086.52, 12830.67, 5.5, B-49
11211.87, 12710.75, 3, B-50
11199.04, 12810.16, 5, B-51
11496.34, 12753.59, 2.5, B-53
11209.35, 12993.94, 3, B-57
11301.97, 13079.66, 5.5, B-58
11286.77, 13026.7, 3, B-59

# FONT Size Just. Plane Orient R G B Resolution Depth Bevel% Font Face
FONT, 6, MC, XZ, +X, 1, 0.5, 0.5, 3, 0.1, 0, Arial
11393.47, 12948.9, 4.5, B-60
11309.03, 12948.99, 5, B-56
11248.75, 12870.91, 4, B-48
```

11259.67, 12819.29, 3, B-46
11298, 12808.63, 4, B-52
11338, 12830.8, 5, B-38
11401.73, 12897.77, 5, B-45
11416.9, 12819.45, 3.5, B-44

FONT, Size, Justification, Plane, Orientation, Red, Green, Blue, Resolution, Depth, Bevel%, Font Face
FONT, 8, MC, XZ, +X, 1, 0, 0, 3, .3, 0, Arial Bold
11340.49, 12892.61, 3.5, B-47
11251.3, 12929.27, 3, B-75
END

interactive_labels

General Module Function

The incteractive_labels module allows the user to place formatted labels at probed locations within the Viewer. The data displayed is whatever data is visible at the

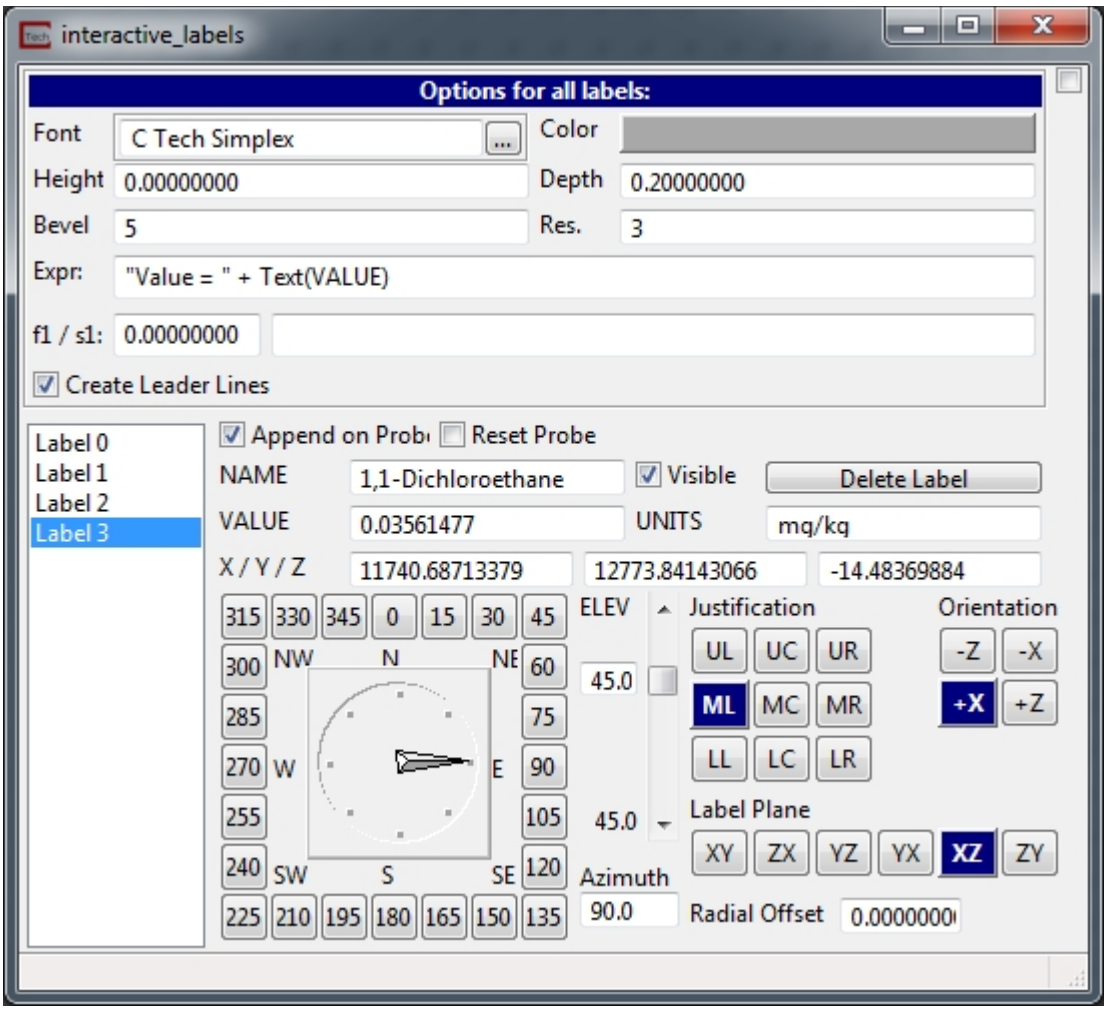
Module Input Ports

1) in_view (Purple) : This port takes input from the Viewer.
2) z_scale (Grey-Brown) : This port takes the z exaggeration factor.
3) f1 (Dark Brown) : This port takes a float value that can be used in the formatted label.
4) s1 (Greyish Blue) : This port takes a string value that can be used in the formatted label.

Module Output Ports

1) z_scale (Grey-Brown) : This port outputs the z exaggeration factor.
2) f1 (Dark Brown) : This port outputs the float value passed in via the input port.
3) s1 (Greyish Blue) : This port outputs the string value passed in via the input port.
4) out_obj (Red) : The port outputs a group renderable labels to the Viewer.

Module Control Panel



The Options for all labels window defines a group of parameters that are applied to all labels. These parameters cannot be set differently for individual labels.

The *Font, Color, Height, Bevel*, and *Res.* fields are the common font options for all labels.

The *Exrpr:* is the format string for the labels. The format string allows the same input as the string_format module with the additional variables of: NAME; VALUE; UNITS; X; Y; and Z. These additional variables correspond to the data values at the probed location.

The *f1* field is a float field that can be passed in and allows for the f1 variable to be used in the expression.

The *s1* field is a string field that can be passed in and allows for the s1 variable to be used in the expression.

There are many options that can be set for individual labels. The list box to the far left lists all of the labels created and the current label being edited.

The *Create Leader Lines* toggle will a leader line from the probed point to the label to be created.

The *Append on Probe* toggle will cause an additional label to be created when the Viewer has been probed in using the Alt + Left Mouse button.

The *Reset Probe* toggle when selected will allow the user to change the probe location of the label currently being edited.

The *NAME, VALUE, UNITS, X, Y*, and *Z* fields are populated based upon the displayed data at the probed location. Changing these values will not affect the data at the probed location.

The *Visible* toggle will turn on or off the currently selected label.

The *Delete Label* button will remove the currently selected label.

The *Azimuth* and *Elev* fields will rotate the label around the probed location.

The *Radial Offset* field allows the user to change the distance from the probed location to the label. There will be a line connecting the label to the probed location.

Justification, Orientation and the *Label Plane* are all label alignment options.

change_minmax

General Module Function

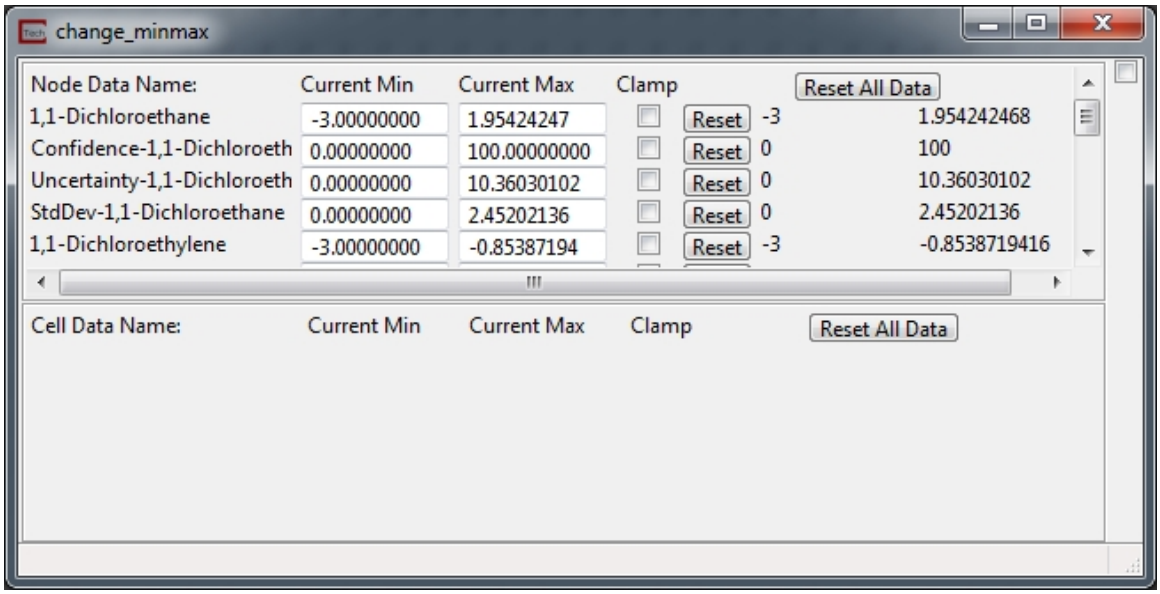
The change_minmax module extends the capabilities of the now deprecated set_minmax module by allowing setting of max values above the true maximum data range and min values below the true minimum data range. This functionality is commonly needed for color mapping of time-series data. For example, the user can set the minmax values to bracket the widest range achieved for many datasets thus allowing consistent mapping from dataset to dataset during a time-series animation. This way 100 ppm would always be red throughout the animation, and if one dataset did not reach a maximum of 100 ppm, there would be no red color mapping for those time-steps.

Module Input Ports

change_minmax has only one input port. Input piped to this port must contain nodal or cell data. If mesh data is piped to this port it will pass through change_minmax unchanged. Vector data should not be modified.

Module Output Ports

1) out_field (Blue-Black): This port outputs a new nodal data set containing the modified data components subjected to the change_minmax criteria.
2) output_data (Red): This port outputs a renderable geometry of the output nodal data set.



Module Control Panel

The control panel for change_minmax is shown in the figure above.

The **Current Min** field allows you to override (change) the minimum value used for coloring (datamaps) for the selected data component.

The **Current Max** field allows you to override (change) the maximum value used for coloring (datamaps) for the selected data component.

Each **Clamp** toggle causes the module to actually change the output values so that no value is greater than Current Max nor less than Current Min. This is functionality that was previously only in the now deprecated clamp module.

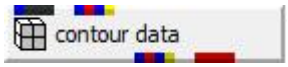
Each **Reset**button allows you to return to the original min/max for that data component.

The **Reset All Data**button allows you to return to the original min/max for all data components.

Related Modules

->clamp

contour_data



General Module Function

This module has many features which are available only in EVS-PRO and MVS.

contour_data provides a means to color surface and volumetric objects in solid colored bands vs. the default Gouraud shading (smoothly changing colors).

This module does not do subsetting like plume_shell (plume_shell), plume_volume, plume_area. It is used in conjunction with these modules to change the way their output is colored.

Module Input Ports

This module has 2 import ports:

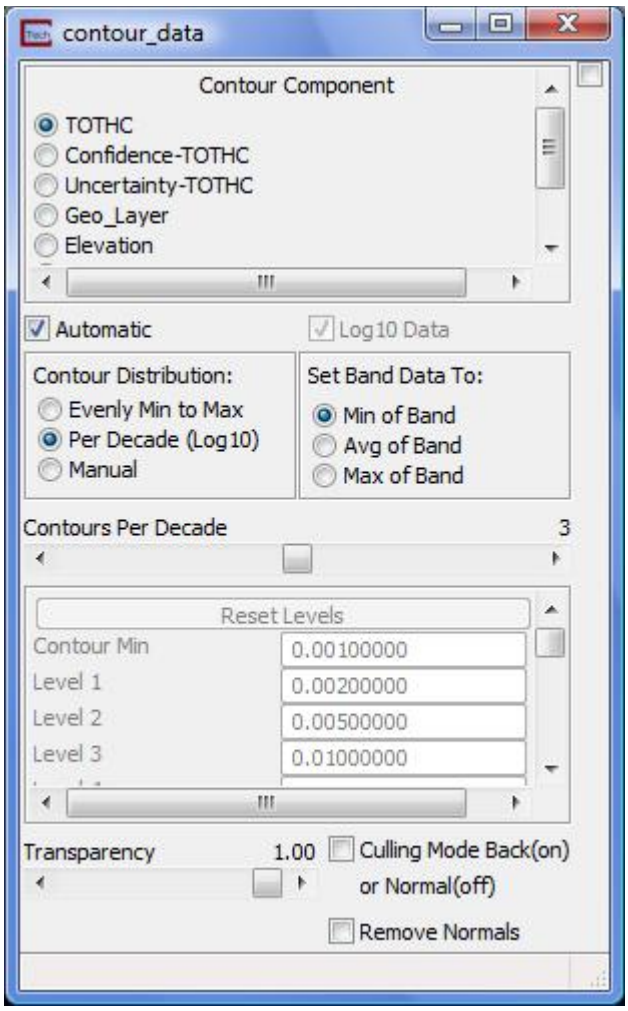
1. The leftmost port accepts an input field
2. The levels information which can be passed between modules. The modules with this port are contour_data, isolines, Datamap_Editor, and Legend.

Module Output Ports

This module has 2 output ports:

1. The leftmost port is the levels information which can be passed between modules. The modules with this port are contour_data, isolines, Datamap_Editor, and Legend.
2. The (red) port for connection to the Viewer.

Module Control Panel



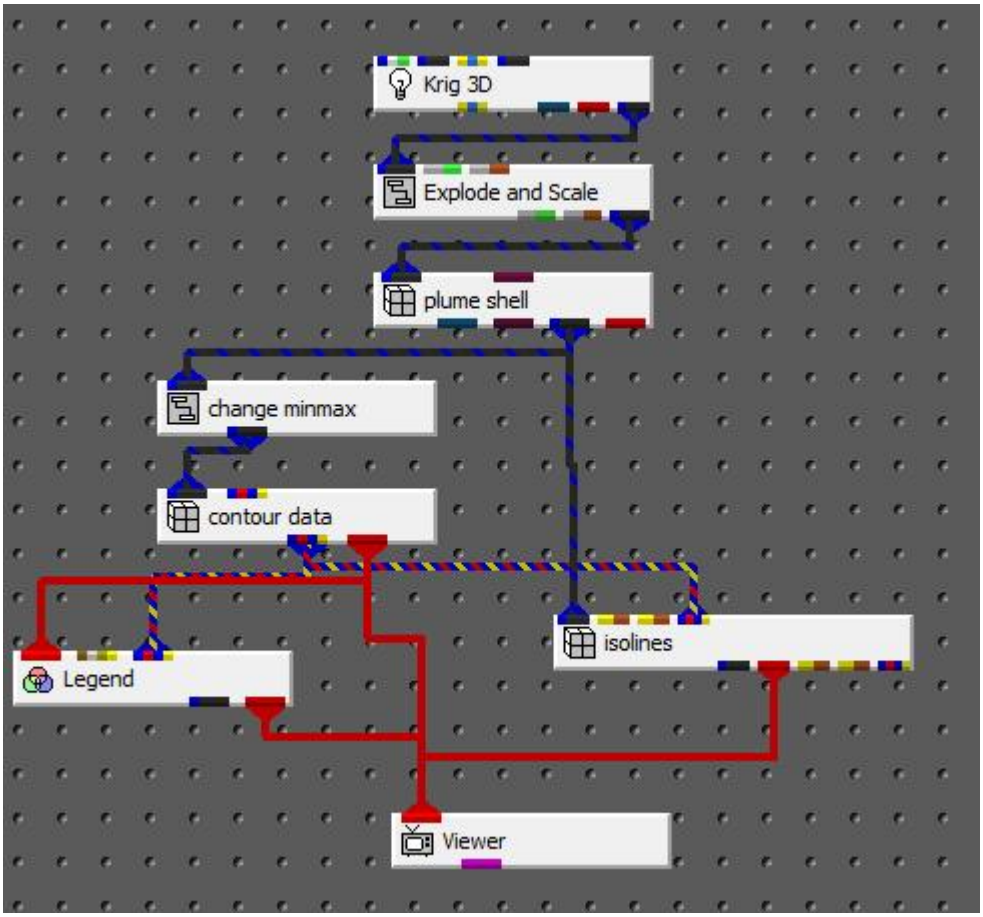
The parameters which you may control are:

1. ***Contour Component*** is the data component which will determine the solid colored bands
2. The ***Automatic*** toggle selects whether your data is Log processed for you
3. The ***Log10 Data*** toggle is inactive if Automatic is ON.
4. ***Contour Distribution*** has 3 options
    1. Evenly Min to Max is the only option available to EVS-Standard. It sets N bands evenly spaced from your data Min to Max.
    2. Per Decade (Log10) allows you to set bands which are similar to the intervals in the Legend module.
    3. Manual allows you to set any intervals you wish.
5. ***Set Band Data To***: has three options which determine what color is assigned to each solid colored band.
    1. Min of Band chooses the color (according to your Datamap) which would be determined by the minimum value of each band

2. Avg of Band chooses the color (according to your Datamap) which would be determined by the average value of each band
3. Max of Band chooses the color (according to your Datamap) which would be determined by the maximum value of each band

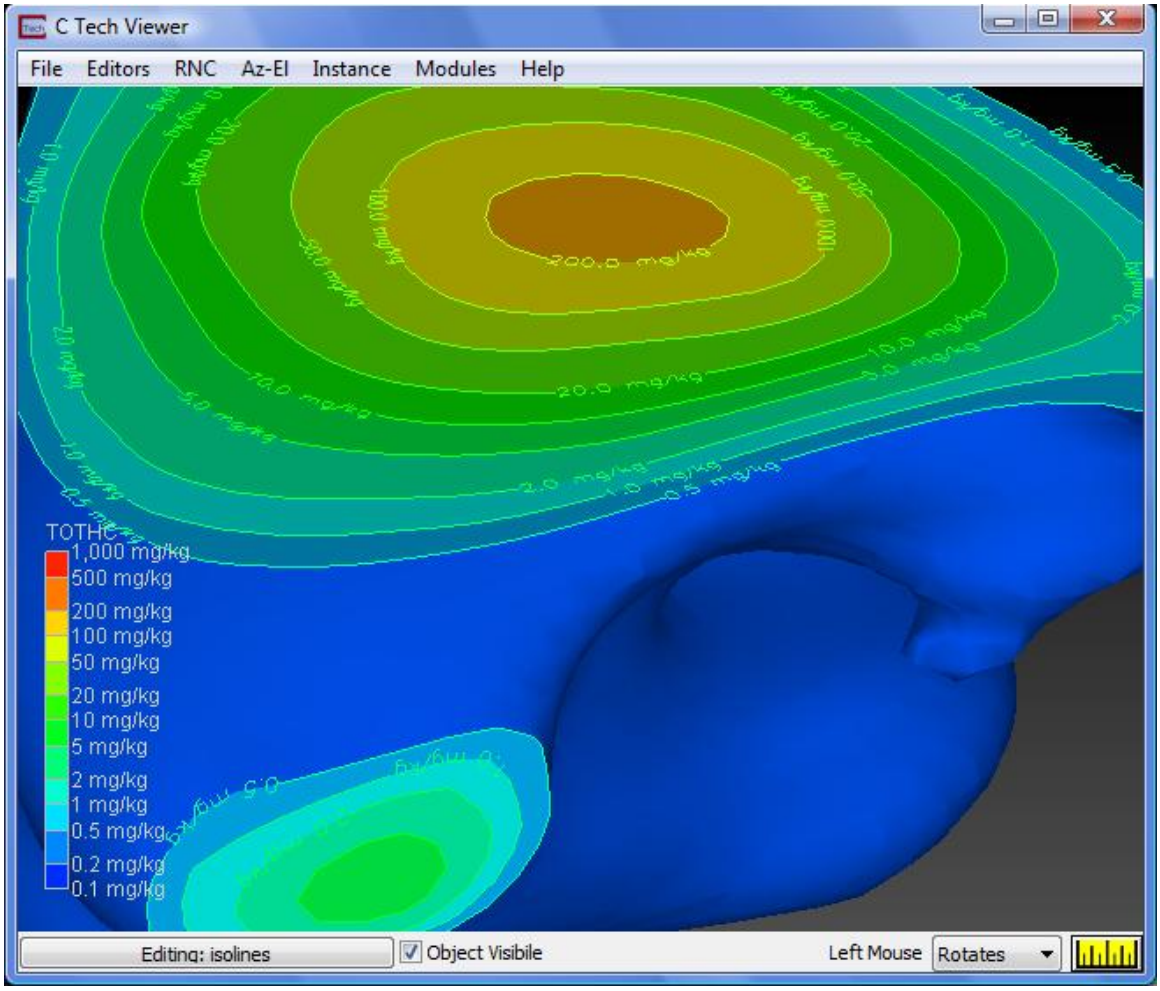
6. The ***Number of Contours*** slider changes its name depending on the Contour Distribution option chosen.
   1. ***Evenly Min to Max*** is labeled Number of Contours
   2. ***Per Decade (Log10)*** is labeled Contours Per Decade
   3. ***Manual*** is labeled Number of Contours. When Manual is chosen, the values for each band can be manually set.
7. The ***Transparency*** slider controls the transparency of the output
8. The ***Remove Normals*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.
9. The ***Culling Mode Back (on) or Normal (off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

An application showing the proper use of contour_data is shown below



The resulting output is below. Please note that the Legend and isolines automatically share the contour levels set in contour_data.

change_minmax is used in the application to cause the resulting colors to have more dynamic range for the plume level chosen (in plume_shell). Without this module, the colors would only be light blues to yellow.



volume_renderer

General Module Function

This module is available in EVS-PRO and MVS only.

Volume_renderer directly renders a 3D uniform field using either the Back-to-Front (BTF) or Ray-tracing volume rendering techniques. The Ray-tracing mode is available to both OpenGL and the software renderer. The BTF

renderer, which is configured as the default, is available only in the OpenGL renderer.

The basic concept of volume rendering is quite different than anything other rendering technique in EVS or MVS. Volume_renderer converts data into a fuzzy transparent cloud where data values at each point in a 3D grid are represented by a particular color and opacity.

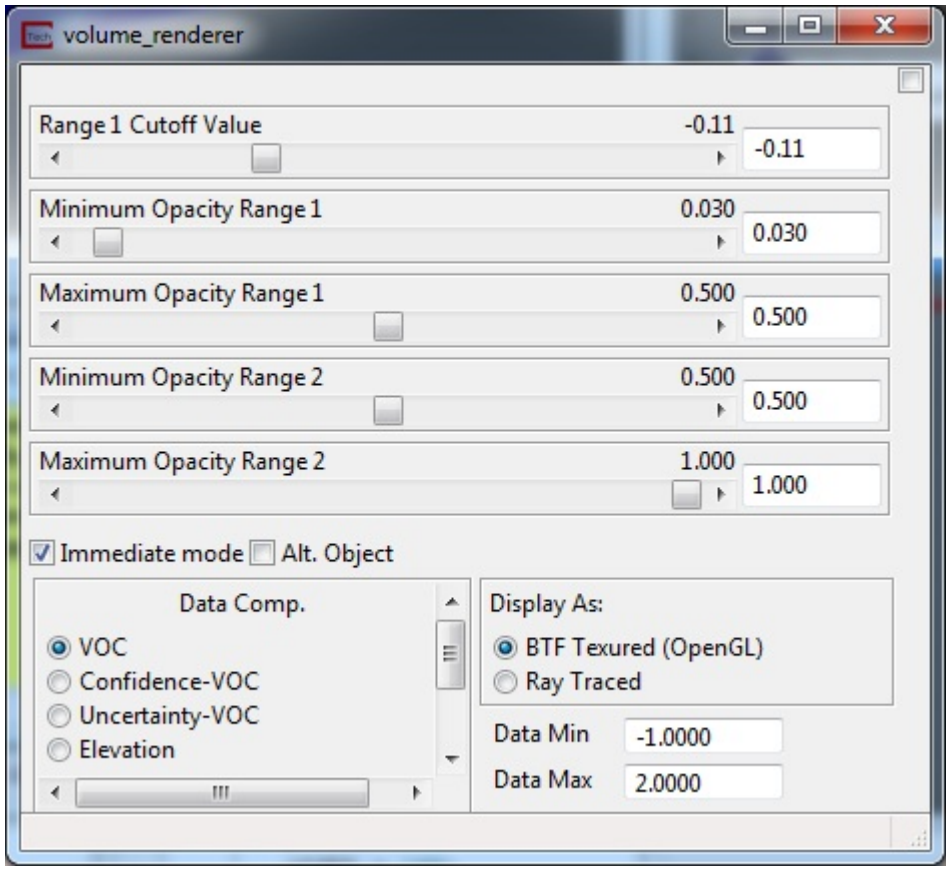
Module Input Ports

This module has one import port accepting an input field which must 3D uniform field with nodal data in order for it to be volume rendered.

Module Output Ports

This module has one output port which sends a renderable object to viewer.

Module Control Panel



Range 1 Cutoff Value slider allows positioning the breakpoint of the two data ranges to be adjusted. By default, it is set to the midpoint of the input nodal data range. The first range in the datamap spans from the minimum data value to the breakpoint. The second range in the datamap spans from the breakpoint to the maximum data value. The range of this slider matches the range of your data or the Data Min and Data Max values set in the fields described below.

The remaining four sliders control the opacity (opposite of transparency) for the minimum and maximum values for each data range. These slider operate in the range of 0.0 (zero) to 1.0.

Generally we recommend that the value for "*Maximum Opacity Range 1*" should *equal "Minimum Opacity Range 2*", otherwise a discontinuity in the opacity levels applied will occur. However, this may sometimes be desirable.

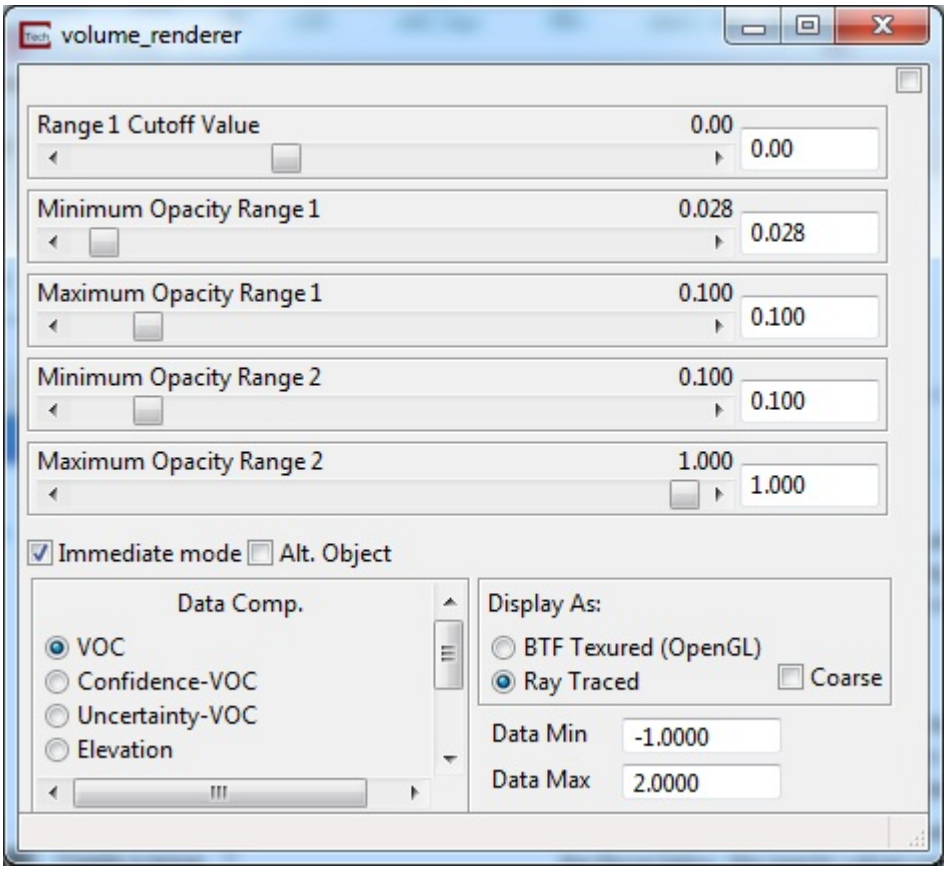
The "*Immediate mode*" toggle causes the sliders to function and update as they are moved. This is on by default, but is best turned off if you have slower hardware or are using the Software Renderer.

The volume_renderer module has an *Alt.* (alternate) *Object* toggle which is off by default. When this is on the volume will not be rendered during any viewer manipulations such as rotation, zooming and panning. The volume rendering will occur when the manipulation is complete and the mouse buttons are released.

Ray Tracing provides the highest quality volume rendering, but it can be as much as 100 times slower than BTF depending on your hardware and viewer (output) resolution. The two modes will produce similar results but with a markedly different appearance.

When Ray Tracing is checked, an additional toggle appears as shown below. (in the figure below, the opacity values were adjusted to create the example output also shown at the end of this topic).

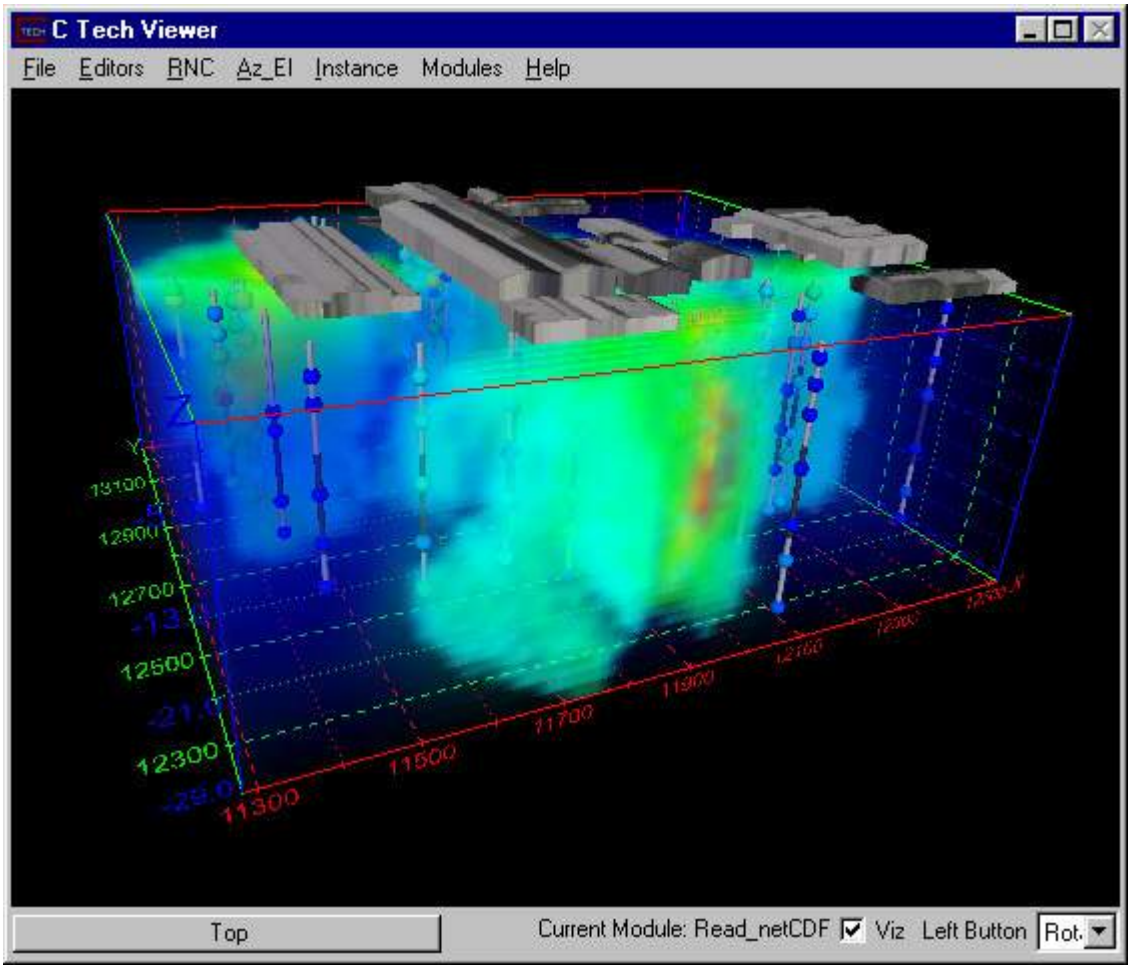
The "*Coarse*" toggle provides for a low resolution "quick" version of software rendering. This is useful for setting parameters, but is not recommended for final output.



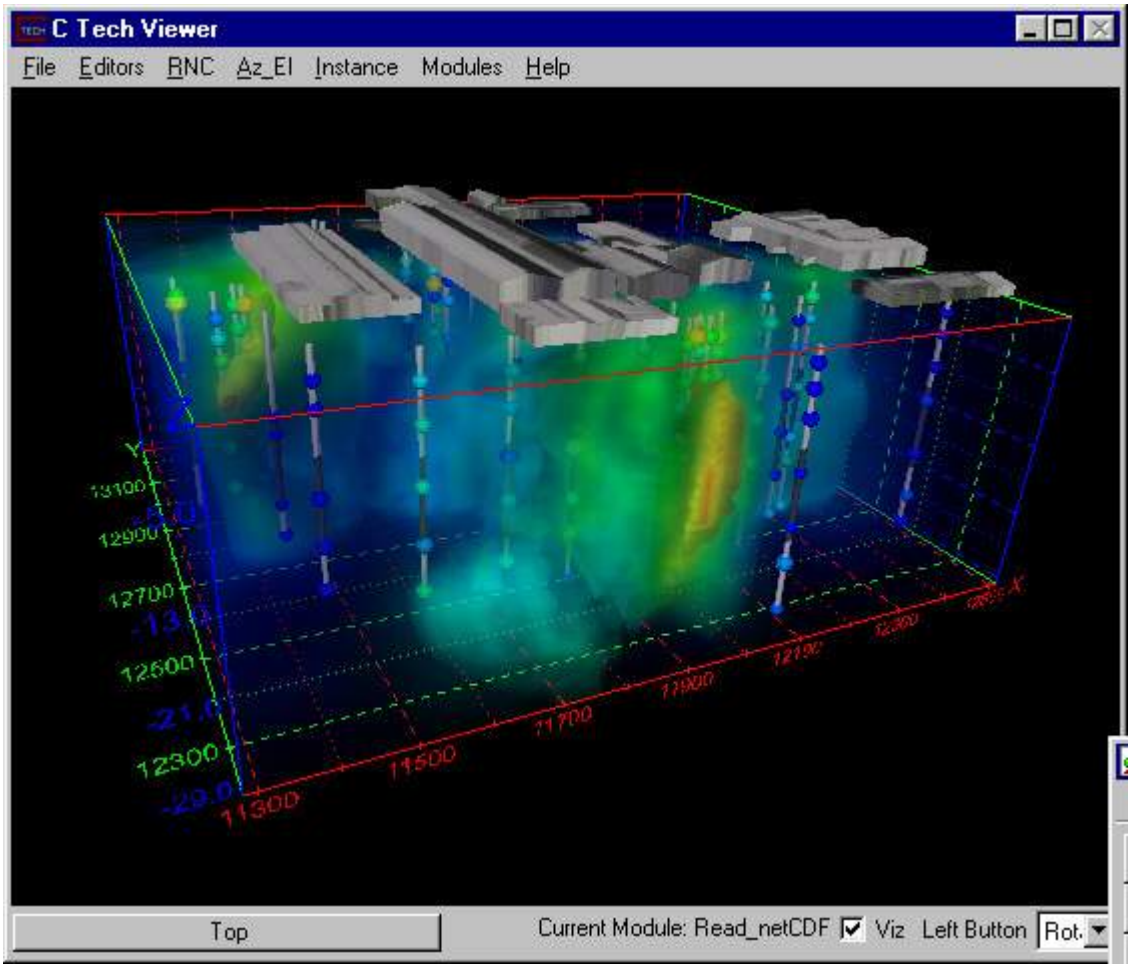
The *Data Comp.* radio selector allows you to choose the nodal data component to be volume rendered.

The *Data Min* and *Data Max* fields allow you to **reset** you actual data minimum and maximum values. Upon connecting the input port of the module, the values will represent your actual data min and max values. Changing these values is similar to using a clamp module in your network. However these values allow you to set the effective data min or max inside or **outside** of your actual data range. This is **critically** important when animating data over time using volume rendering.

An example of the output created using OpenGL (BTF) volume rendering is:



An example of the output created using software (ray traced) volume rendering is:



IMPORTANT NOTES:

Volume rendering is compatible with Perspective mode.

Output at any resolutions higher than desktop size require software rendering.

Volume rendering can only be export as an image or a sequence of images to be converted to an animation. This is accomplished with the Output_Images module.

Volume rendering is not supported by VRML or C Tech's 4DIM format, nor can it be export as a DXF, Shapefile, or any other vector formats.

adjust_transparency

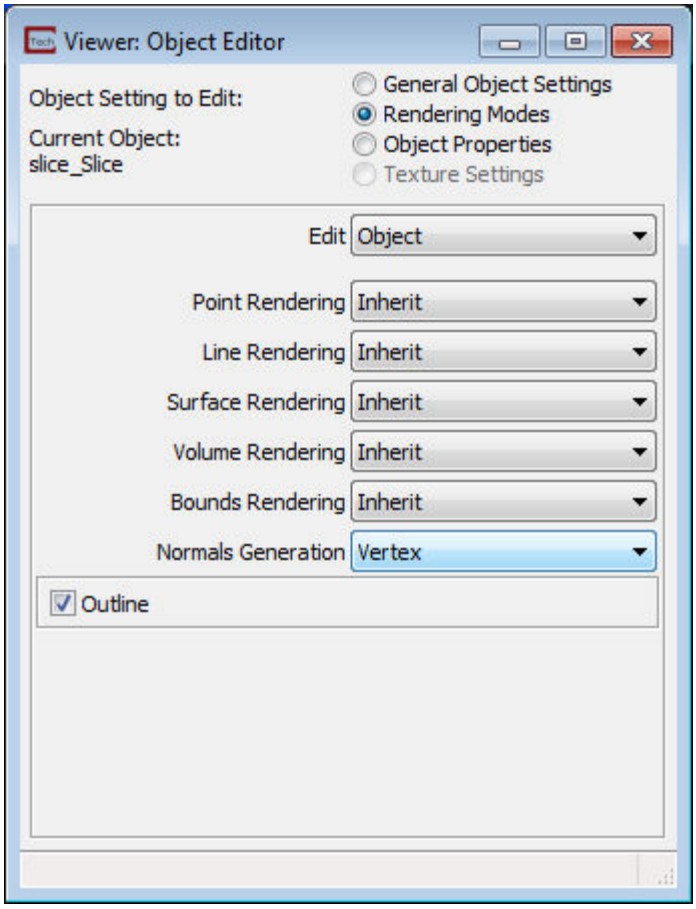
General Module Function

This module is available in EVS-PRO and MVS only.

adjust_transparency provides a means to adjust the transparency of any object based on its data values. The appearance of the resulting output is often similar in appearance to volume rendering though the approach is more like the Datamap_Editor. adjust_transparency converts data into

transparent surfaces where data values at each point in a grid are represented by a particular color and opacity.

NOTE: Any module connected after adjust_transparency MUST have Normals Generation set to Vertex (if there is a Normals Generation toggle on the module's panel, it must be OFF).



Module Input Ports

This module has 5 import ports:

1. The leftmost port accepts an input field
2. The real value for f1
3. The real value for f2
4. The real value for f3
5. The real value for f4

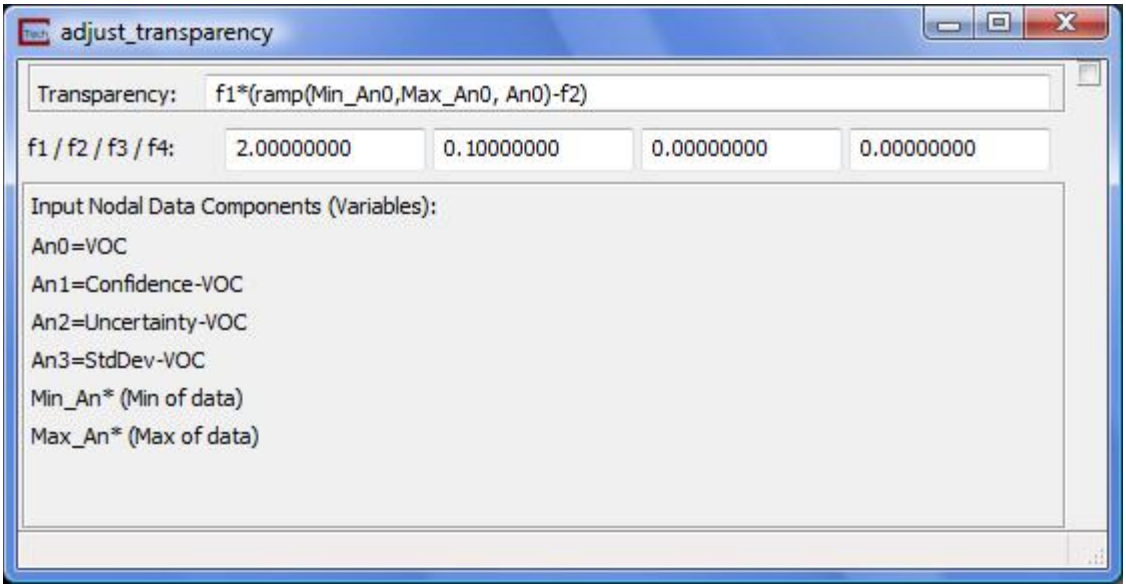
Module Output Ports

This module has 6 output ports:

1. The leftmost port is the real value for f1

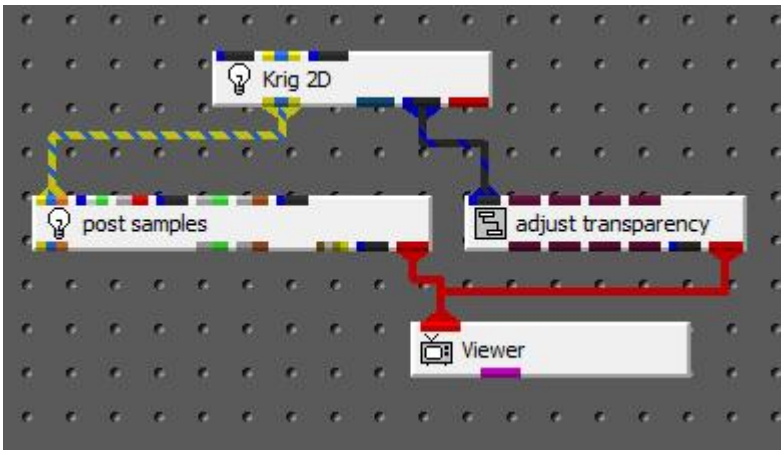
2. The real value for f2
3. The real value for f3
4. The real value for f4
5. The output field which passes the original data with a special new "transparency" data component for use with downstream modules (e.g. slice, plume_shell, etc.)
6. The (red) port for connection to the Viewer.

Module Control Panel

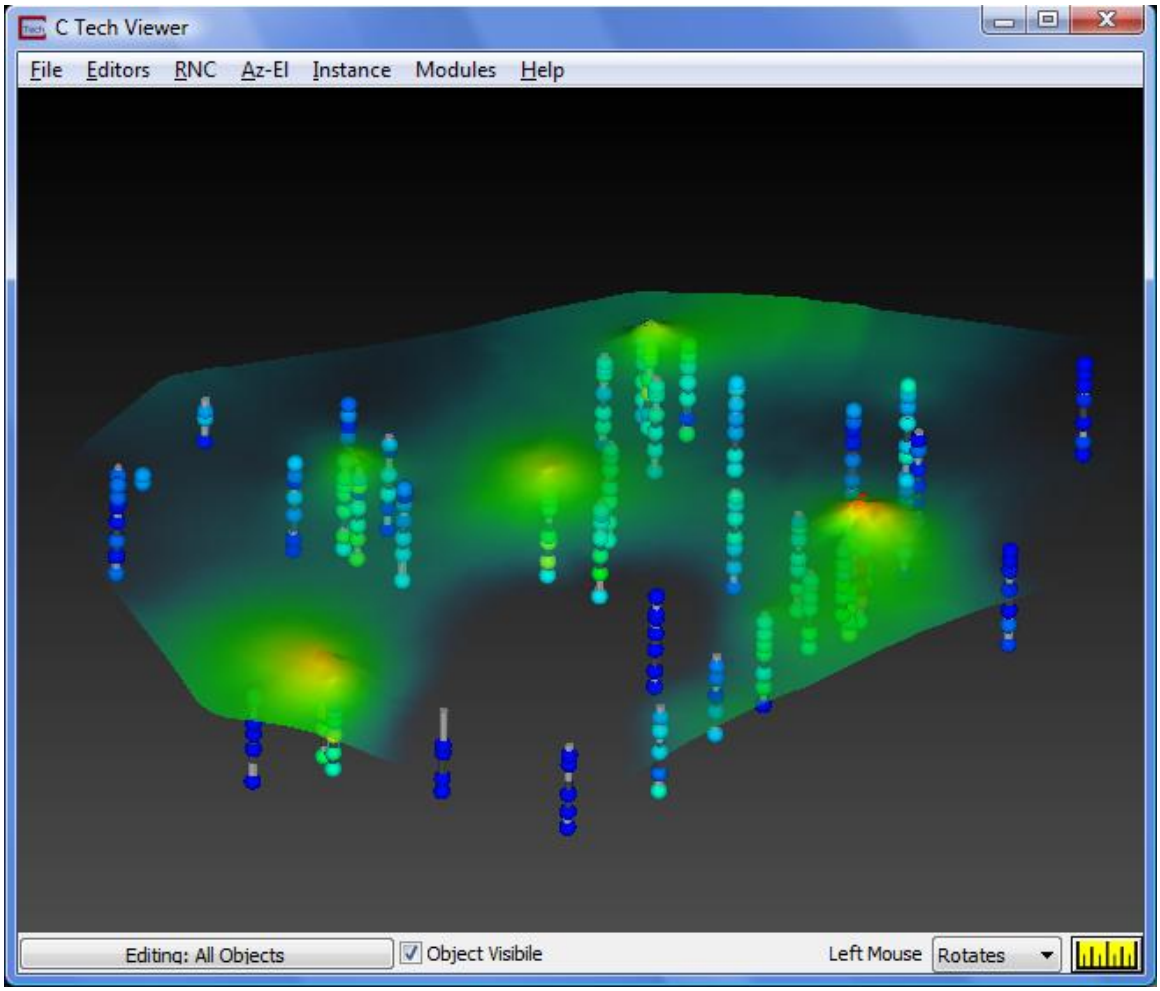


The control panel for adjust_transparency has one type-in "Transparency" equation and 4 numeric parameters which can be shared with other modules or animated. The equation determines the relationship between one or more data components and the resulting transparency which will be assigned to every node in the input grid. This provides a means to control the transparency of objects in ways that are not possible with any other module.

The default equation provides a ramped transparency based on the min and max values of the first data component (An0). With f1=1 and f2=0, this will make the transparency at the min value = 0.0 and 1.0 at the max value. Setting f1 to higher values causes the transparency to peak at 1 sooner (lower data values) and increasing f2 causes the minimum transparency to occur at higher values.

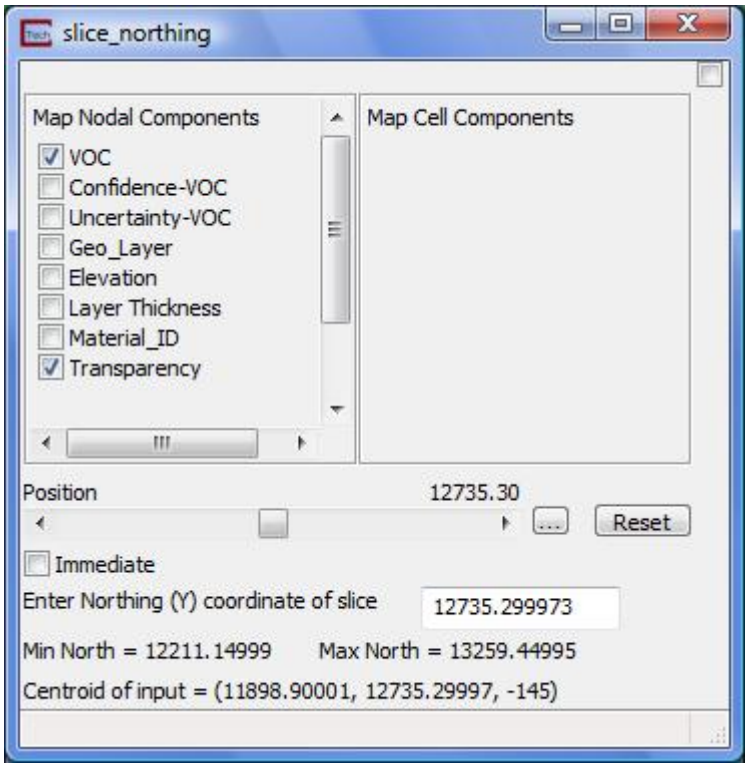


The above application is a very simple example of using this module and is shown using the default values (2.0 and 0.1).

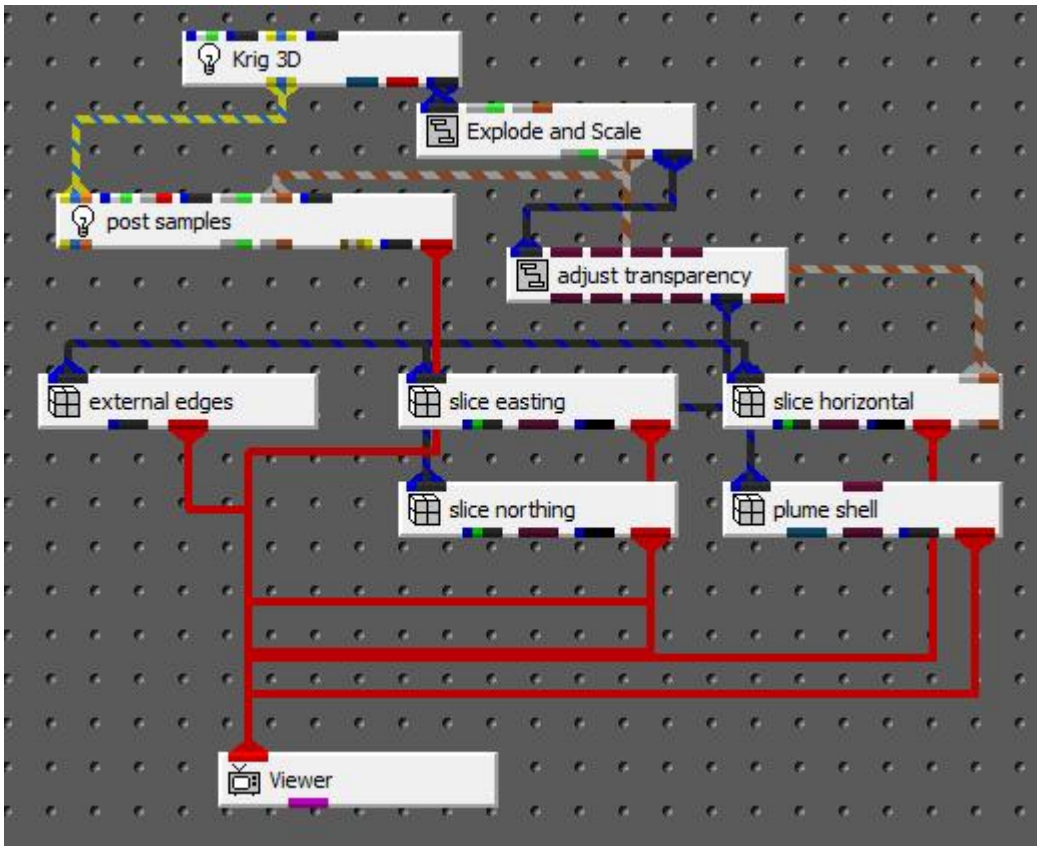


The application below shows much more sophisticated use of adjust_transparency. Note that adjust_transparency is NOT connected directly to the Viewer. The nodal transparencies of set by

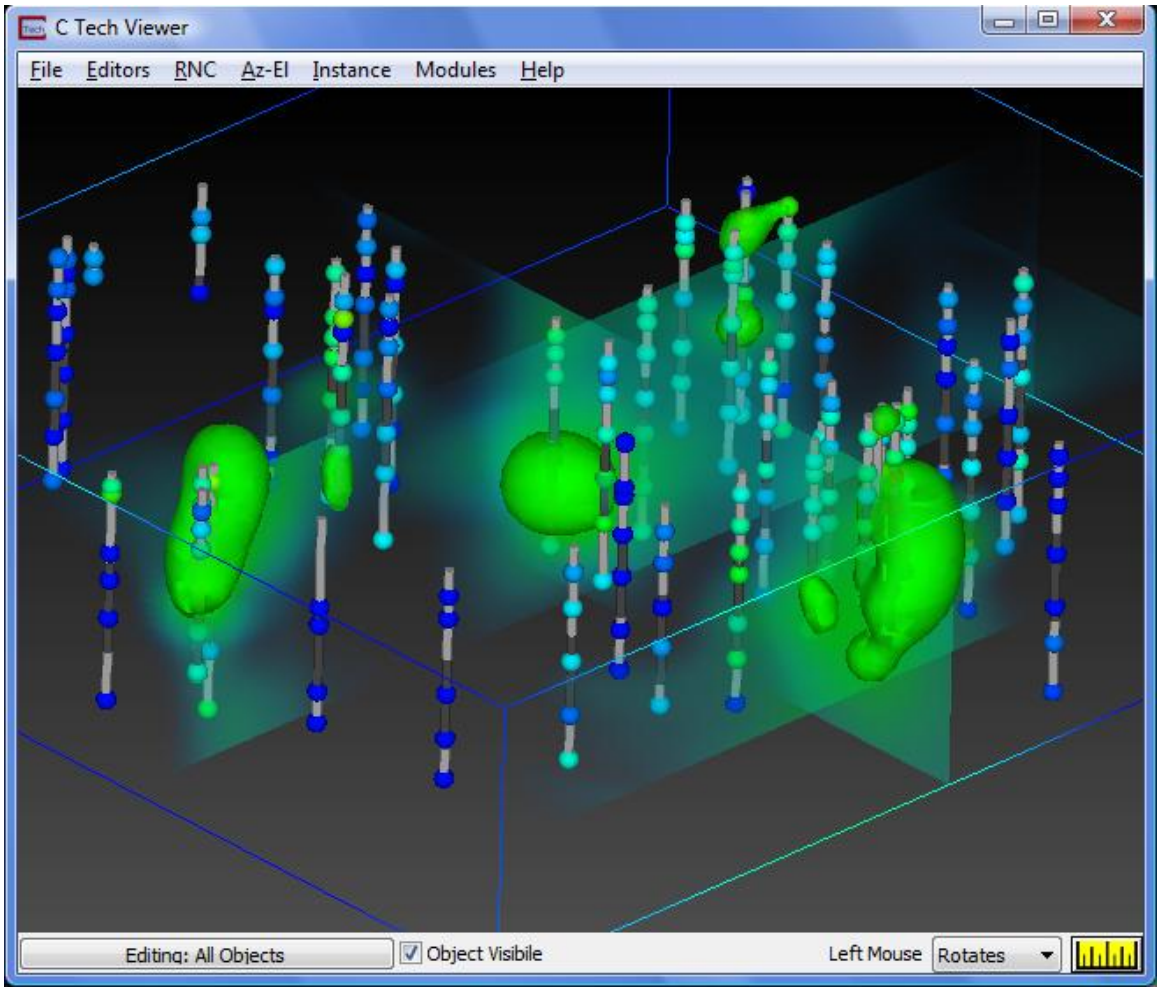
adjust_transparency, but are inherited in all downstream modules. To make the downstream modules you need to turn ON the "Transparency" data component in addition to the one to be used for coloring. Normally when more than one data component is ON, only the first one affects the output in the Viewer. Transparency is a different concept.



The application below:



Creates the following output.



texture_colors

General Module Function

The texture_colors module does high-quality coloring based on texturing. texture_colors allows you to change the way the display of colors is generated for fields with nodal data. Typically we assign a color directly to each node, which is the interpolated between the surfaces. In texture_colors internal datamap is used automatically to color, but instead of coloring by RGB values, an image is texture mapped onto the surface. Aberrant color interpolation (as described in Visualization Fundamentals)goes away with this technique. As users who have been to a class are aware, this can cause colors to appear which are not part of your datamap. As a solution, the texture_colors module will color an object by texture mapping an image onto your object directly instead of assigning colors. This allows the full range of colors to be displayed between two nodes, completely eliminating any aberrant colors. Many times this can lead to better quality output.

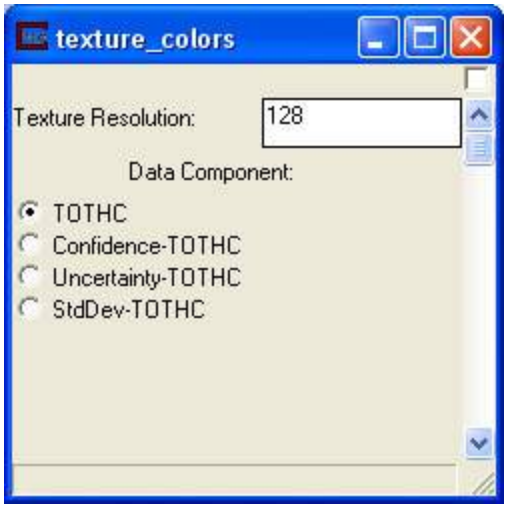
This is a MVS/PRO module only, since it uses texture mapping internally.

Module Input Ports

texture_colors has a single input port that accepts the grid with nodal data that you want to color with this technique.

Module Output Ports

texture_colors has two output ports. The red port can be connected to the Viewer and the multi-colored port is for input to the Legend. This should only be used with reasonably small texture resolution (less than 20).



Module Control Panel

The control panel for texture_colors is shown in the figure above.

Texture Resolution: determines the number of colors utilized in the texture map for coloring. Higher resolutions create smoother coloring, but at the expense of memory and compute/rendering time.

Data Component: selects the nodal data component for coloring.

tubes

General Module Function

The tubes module is used to produce open or closed tubes of constant or data dependent radius using 3D lines or polylines as input.

(This feature available only in EVS PRO and MVS)

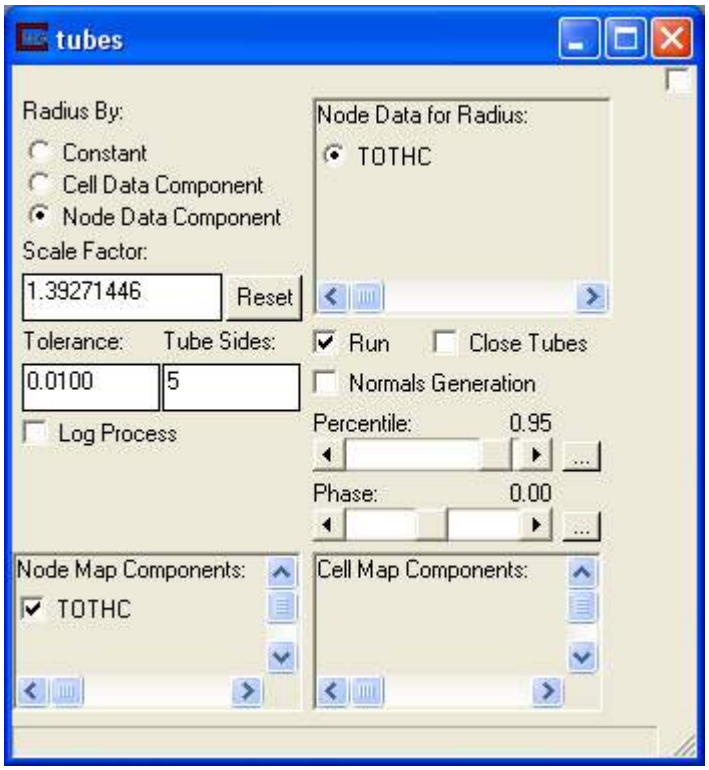
Module Input Ports

Tubes has one input port that accepts the 3D lines or polylines as input.

Module Output Ports

Tubes has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains the tubes. The second port outputs a renderable geometry, which can be connected directly to the Viewer.

Module Control Panel



The control panel for tubes is shown in the figure above.

Radius By: radio buttons allow you to choose constant or data dependent radius scaling.

Node Data for Radius: radio buttons allow you to choose the nodal data component for scaling.

Cell Data for Radius: radio buttons allow you to choose the cell data component for scaling.

The **Scale Factor:** parameter is multiplied by the nodal data to determine the radius.

The **Reset** button causes the automatic computation of scale factor to be performed. This is useful if you have changed the scale or the nodal/cell data used.

The **Percentile** slider allows you to control the automatic scaling of tubes based on the nth percentile value (versus the maximum 100th%). This addresses datasets where there are only a few nodes with extremely high values.

The **Tolerance:** parameter determines the maximum distance between nodes of adjacent line segments before those line's nodes should be merged to form a polyline. This also applies to closed polyline contours. Closed polylines will have their starting and ending nodes merged to form closed (torroidal) annuli.

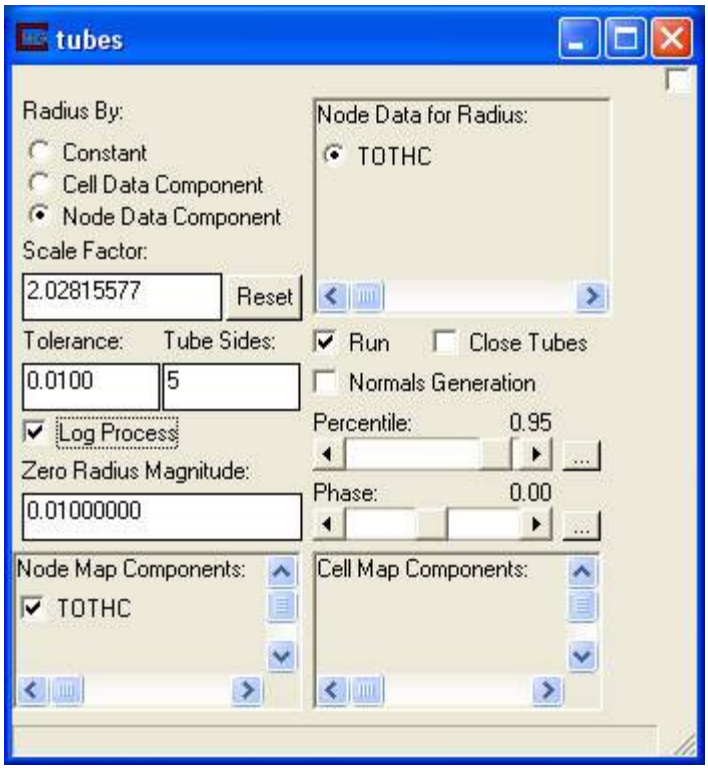
The **Tube Sides:** parameter is The number of faces on the tubes. (the default is 6)

The **Run** toggle causes the module to run whenever parameters or inputs change.

The **Close Tubes** toggle causes the tubes to have solid ends. However the tubes are not solid objects (they are hollow).

The **Normals Generation** toggle turns on Normals Generation for rendering the tubes. This makes their faces more distinct (less smoothed). If you want the tubes to represent cylinders, this should be off.

The **Log Process** toggle causes the data to be log scaled before it is used for radius scaling. This parameter causes the Zero Radius Magnitude parameter to become active also as shown in the figure below.



The **Zero Radius Magnitude** parameter is required when using log processing. Since you cannot take the log of zero (or negative values) you must set the value (from inside or outside of your range of data values) that will correspond to a zero radius.

The **Phase** slider allows you to control the apparent rotation of tubes. A phase of 0.5 rotates 180 degrees.

The**Node Map Components**check boxes allow you to select those nodal data components that are used for coloring or subsequent subsetting operations.

The**Cell Map Components**check boxes allow you to select those cell data components that are used for coloring or subsequent subsetting operations.

cross_section_tubes

General Module Function

The cross_section_tubes module is used to produce open or closed tubes of user defined cross-section and constant or data dependent radius using 3D lines or polylines as input for the centerline and a single 2D polyline as the cross-section of the tubes..

(This feature available only in MVS)

Module Input Ports

Tubes has two input ports.

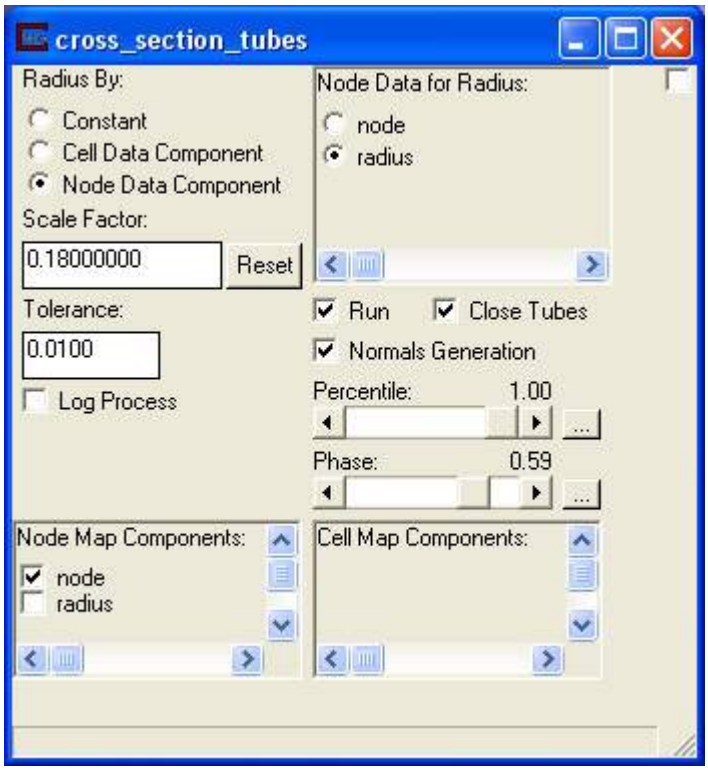
The left port accepts the 3D lines or polylines as input for the centerlines of the tubes

The right port accepts a single 2D polylines as input for the cross-section of the tubes.

Module Output Ports

cross_section_tubes has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains the tubes. The second port outputs a renderable geometry, which can be connected directly to the Viewer.

Module Control Panel



The control panel for cross_section_tubes is shown in the figure above.

Radius By: radio buttons allow you to choose constant or data dependent radius scaling.

Node Data for Radius: radio buttons allow you to choose the nodal data component for scaling.

Cell Data for Radius: radio buttons allow you to choose the cell data component for scaling.

The **Scale Factor:** parameter is multiplied by the nodal data to determine the radius.

The **Reset** button causes the automatic computation of scale factor to be performed. This is useful if you have changed the scale or the nodal/cell data used.

The **Percentile** slider allows you to control the automatic scaling of tubes based on the nth percentile value (versus the maximum 100th%). This addresses datasets where there are only a few nodes with extremely high values.

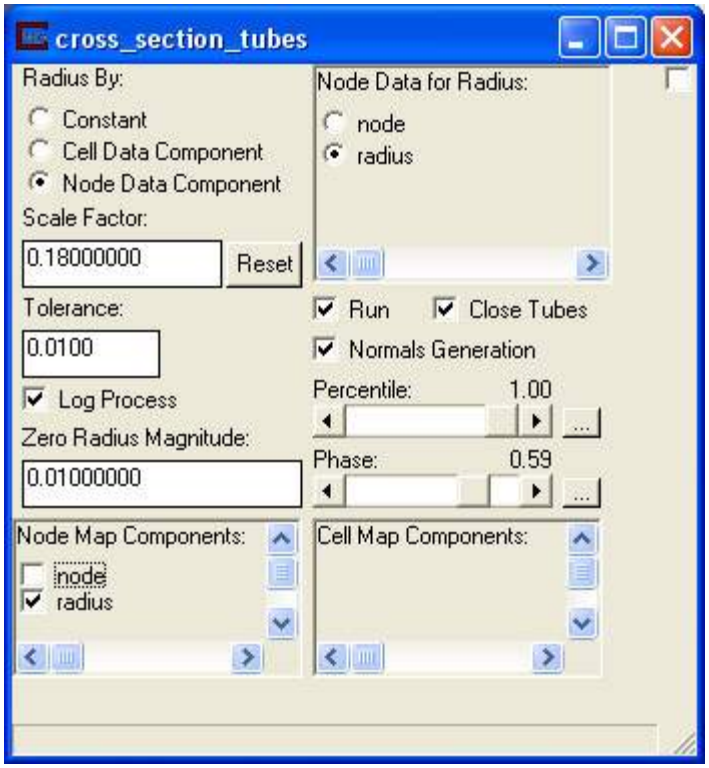
The **Tolerance:** parameter determines the maximum distance between nodes of adjacent line segments before those line's nodes should be merged to form a polyline. This also applies to closed polyline contours. Closed polylines will have their starting and ending nodes merged to form closed (torroidal) annuli.

The **Run** toggle causes the module to run whenever parameters or inputs change.

The **Close Tubes** toggle causes the tubes to have solid ends. However the tubes are not solid objects (they are hollow).

The **Normals Generation** toggle turns on Normals Generation for rendering the tubes. This makes their faces more distinct (less smoothed). If you want the tubes to represent cylinders, this should be off.

The **Log Process** toggle causes the data to be log scaled before it is used for radius scaling. This parameter causes the Zero Radius Magnitude parameter to become active also as shown in the figure below.



The **Zero Radius Magnitude** parameter is required when using log processing. Since you cannot take the log of zero (or negative values) you must set the value (from inside or outside of your range of data values) that will correspond to a zero radius.

The **Phase** slider allows you to control the apparent rotation of tubes. A phase of 0.5 rotates one-half of one face.

The**Node Map Components**check boxes allow you to select those nodal data components that are used for coloring or subsequent subsetting operations.

The**Cell Map Components**check boxes allow you to select those cell data components that are used for coloring or subsequent subsetting operations.

streamlines

General Module Function

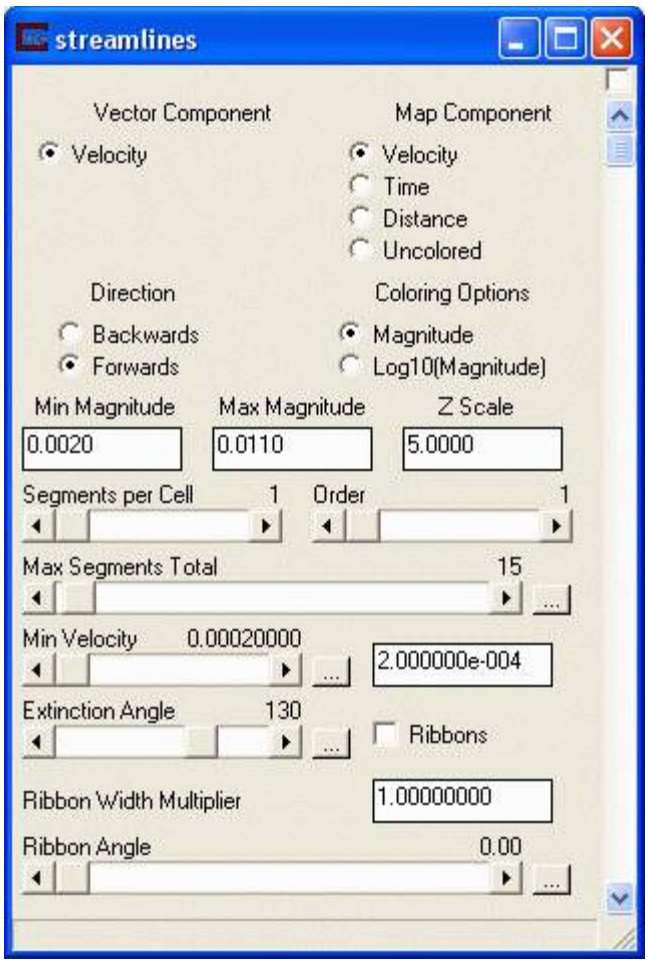
The streamlines module is used to produce streamlines or streamribbons of a field which is a 2 or 3 element vector data component on any type of mesh. Streamlines, which are simply 3D polylines, represent the pathways particles would travel based on the gradient of the vector field. Streamribbons are 3D streamlines which can be rendered. At least one of the nodal data components input to streamlines must be a vector. The direction of travel of streamlines can be specified to be forwards (toward high vector magnitudes) or backwards (toward low vector magnitudes) with respect to the vector field. Streamlines are produced by integrating a velocity field using the Runge-Kutte method of specified order with adaptive time steps.

Module Input Ports

Streamlines has three input ports. Data passed to the first port (the left port) can have a mesh of any type, and must have at least one vector nodal data component. To generate streamlines of scalar data, a module such as gradient must be used to calculate the gradient of a scalar field. The second port accepts data specifying the starting location of the streamlines. This data is typically generated by the slice or isolines or place_glyph modules. The last port is the z exaggeration factor. This input causes the stream path generation to be calculated in an unscaled system ensuring that velocities are not scaled and are accurate, then scales the position of the streamlines.

Module Output Ports

Streamlines has two output ports. The leftmost output port creates a new unstructured polyline mesh representing the streamlines. The output also contains a nodal data component referencing the velocity component. The right output port can send either streamline polylines or renderable streamribbons to the viewer.



Module Control Panel

The control panel for streamlines is shown in the figure above.

The **Vector*Component*** radio button displays all data components (vector and scalar) passed to streamlines. By default, the first (0th) component is selected.

The ***Map Component*** radio button allows you to choose which computed data component to use for coloring the lines. Choices are Velocity, Time, Distance and Uncolored.

The ***Direction*** radio buttons allow the user to specify forward or backward streamlines. Forward streamlines start from the specified starting points and travel to the maximum velocity location. Backward streamlines travel from the specified starting points to the minimum or zero velocity location.

The ***Coloring Options*** radio buttons allow the user to specify *Magnitude* (normal) or *Log10(Magnitude)* representation of the velocity data for coloring purposes. When velocities span several orders of magnitude working in log space is useful.

The ***Min Magnitude*** and ***Max Magnitude*** type-ins display the min/max velocities represented in the input field. Changing these values allows the user to reset the min-max values for coloring purposes. NOTE: once you change these values they will not automatically update.

The ***Z Scale*** type-in reflects the value of the z exaggeration port.

The **Segment per Cell** slider is used to set the number of integration steps to be used in each cell (i.e., the number of divisions of the cells) to calculate the streamline. The default is one and the range is 1 to 16.

The ***Max Segments Total*** slider is used to set the maximum allowable number of streamline segments that will be completed for each streamline. If the number of segments along a streamline exceeds the max number, the streamline is terminated at the end of the last (max) segment. The default is 15 and the range is 1 to 1000.

The **Order** slider is used to set the order of the integration. Higher order integration is more accurate, but executes much slower. The default is one and the range is 1 to 4.

The **Min Velocity** slider and type-in is used to specify the minimum velocity that will be considered in the integration. If the magnitude of the velocity field in a region is less than this minimum value, streamlines will end in that region (or will not be produced if the gradient at a starting point is less than the min). Setting this to a lower value will produce longer streamlines (and typically more). Higher values tend to produce less streamlines and shorter streamlines. If streamlines are not visible on the data set, setting this to a lower value may produce streamlines. The default value is 0.000001.

The ***Extinction Angle*** slider is used to specify the maximum allowable angle between successive line segments before integration (streamline generation) should be terminated. The default value is 130 degrees.

Streamribbons are essentially three dimensional streamlines which are renderable by the viewer.

The **Ribbons** check box is used to set the output to streamribbons (on) or streamlines (off). If the box is checked, then the ribbon width edit field can be used to set the width of the ribbons. The value entered here is a scale factor (relative to the divergence, or absolute value of the vector magnitude) used to set the width of the beginning point of the ribbon.

The **Ribbon Width Multiplier** scales the divergence (or magnitude) of the vector field. The default is 1.00 and the range is unbounded.

The ***Ribbon Angle*** slider, controls the initial angle at which the ribbon is drawn. The default is 0 and the range is 0 to 360. Note that if the ribbon angle is 0, the view may need to be rotated to see the ribbons.

The calculation of streamribbons is complex, and can be somewhat unstable if the parameters for calculating the streamlines are not set correctly. The user should experiment with different values of the streamline parameters if the module fails during production of stream ribbons. Setting order and nsegment to 2 or more will improve stability for ribbons.

Related Modules

->glyph

->gradient

streamline_surface

General Module Function

The streamline surface module is used to produce streamlines on any surface based on its slopes. Streamlines are 3D polylines representing the paths particles would travel based on the slopes of the input surface. The direction of travel of streamlines can be specified to be downhill or uphill for the slope case. A physics simulation option is also available which employs a full physics simulation including friction and gravity terms to compute streamlines on the surface.

Module Input Ports

Streamline surface has three input ports.

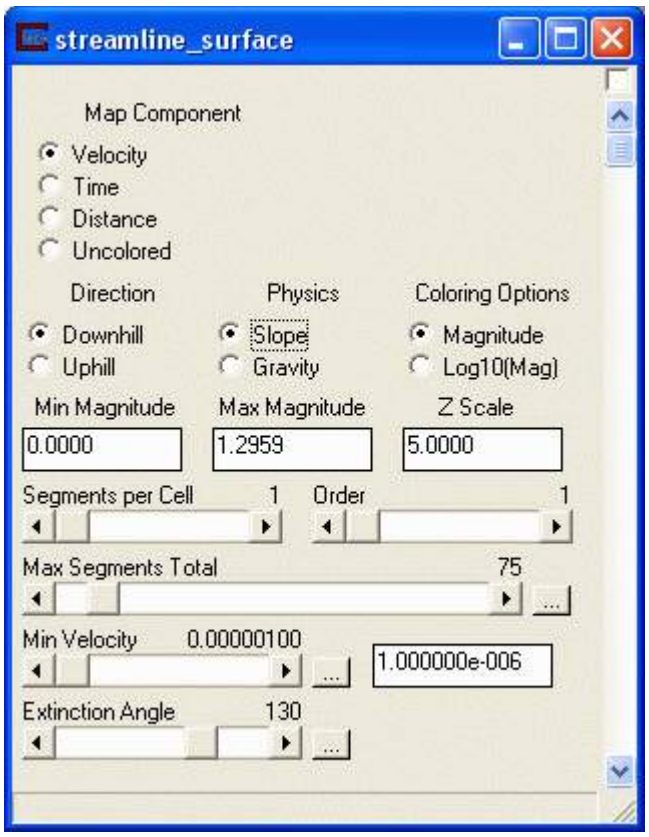
Data passed to the first port (the left port) must be a surface.

The second port accepts data specifying the starting location of the streamlines. This data is typically generated by the slice or isolines or place_glyph modules.

The last port is the z exaggeration factor. This input causes the stream path generation to be calculated in an unscaled system ensuring that velocities are not scaled and are accurate, then scales the position of the streamlines.

Module Output Ports

Streamline surface has two output ports. The leftmost output port creates a new unstructured polyline mesh representing the streamlines. The output also contains a nodal data component referencing the velocity component. The right output port can send either streamline polylines or renderable streamribbons to the viewer.



Module Control Panel

The control panel for streamline surface is shown in the figure above.

The *Map Component* radio button allows you to choose which computed data component to use for coloring the lines. Choices are Velocity, Time, Distance and Uncolored.

The *Direction* radio buttons allow the user to specify forward or backward streamlines. Forward streamlines start from the specified starting points and travel to the maximum velocity location. Backward streamlines travel from the specified starting points to the minimum or zero velocity location.

The *Coloring Options* radio buttons allow the user to specify Magnitude (normal) or Log10(Magnitude) representation of the velocity data for coloring purposes. When velocities span several orders of magnitude this is useful.

The *Physics* radio buttons allow the user to specify whether streamlines will be computed based on the slopes of the surface only or whether a full physics simulation including friction and gravity terms will be used to compute streamlines on the surface. When *Gravity* is selected *Segments perCell* and *Order* do not apply but additional parameters appear for the module. These are:

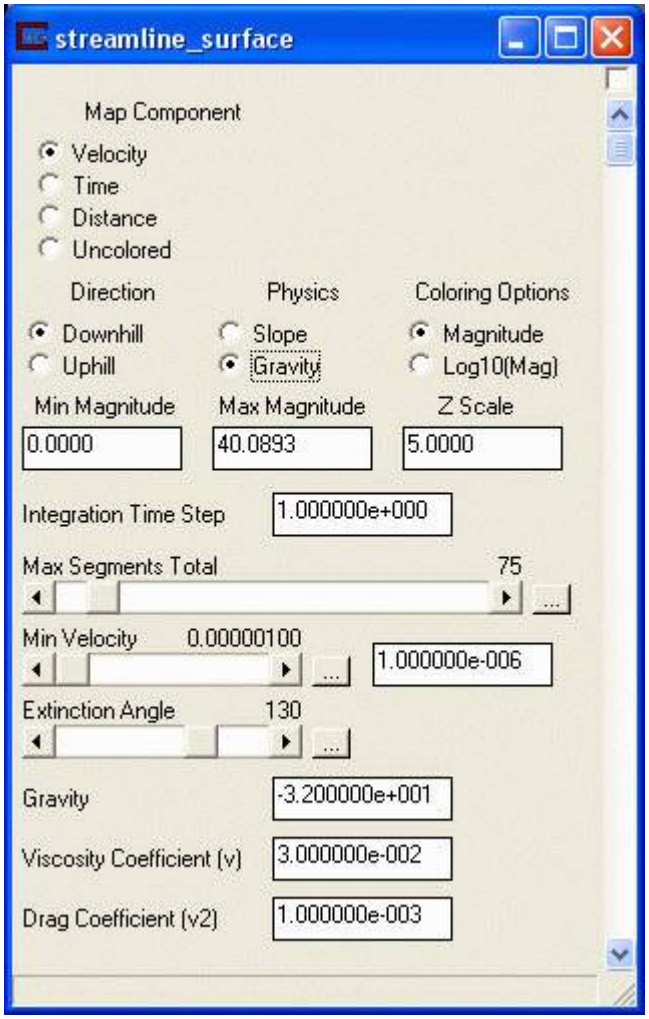
> *Integration Time Step* is the time step for the numerical integration of the paths. For typical gravity units (like 32 feet per second-squared) this value is in seconds.

Gravity is the coefficient of gravity for your units. If your coordinate units are feet, the appropriate (default) value would be 32 feet per second-squared.

Viscosity Coefficient (v) is the friction term that depends on velocity.

Drag Coefficient (v2) is the friction term that depends on velocity-squared.

The panel when Gravity is selected is show below.



The *Min Magnitude* and *Max Magnitude* type-ins display the min/max velocities represented in the input field. Changing these values allows the user to reset the min-max values for coloring purposes. NOTE: once you change these values they will not automatically update.

The *Z Scale* type-in reflects the value of the z exaggeration port.

The *Segment per Cell* slider is used to set the number of integration steps to be used in each cell (i.e., the number of divisions of the cells) to calculate the streamline. The default is one and the range is 1 to 16.

The *Max Segments Total* slider is used to set the maximum allowable number of streamline segments that will be completed for each streamline. If the number of segments along a streamline exceeds the max number, the

streamline is terminated at the end of the last (max) segment. The default is 15 and the range is 1 to 1000.

The *Order* slider is used to set the order of the integration. Higher order integration is more accurate, but executes much slower. The default is one and the range is 1 to 4.

The *Min Velocity* slider and type-in is used to specify the minimum velocity that will be considered in the integration. If the magnitude of the velocity field in a region is less than this minimum value, streamlines will end in that region (or will not be produced if the gradient at a starting point is less than the min). Setting this to a lower value will produce longer streamlines (and typically more). Higher values tend to produce less streamlines and shorter streamlines. If streamlines are not visible on the data set, setting this to a lower value may produce streamlines. The default value is 0.000001.

The *Extinction Angle* slider is used to specify the maximum allowable angle between successive line segments before integration (streamline generation) should be terminated. The default value is 130 degrees.

modpath

General Module Function

The modpath module uses the cell by cell flow values generated from a MODFLOW project along with head values and other MODFLOW parameters to trace the path of a particle of water as it moves through the ground. The paths are calculated using the same algorithms used by U.S. Geological Survey MODPATH and the results should be similar.

The modpath module at this point does not handle transient simulations the same way that the U.S.G.S. MODPATH does. It treats each time step as a steady state model, and uses the parameters from the .dwr/.dwz file based on the starting time.

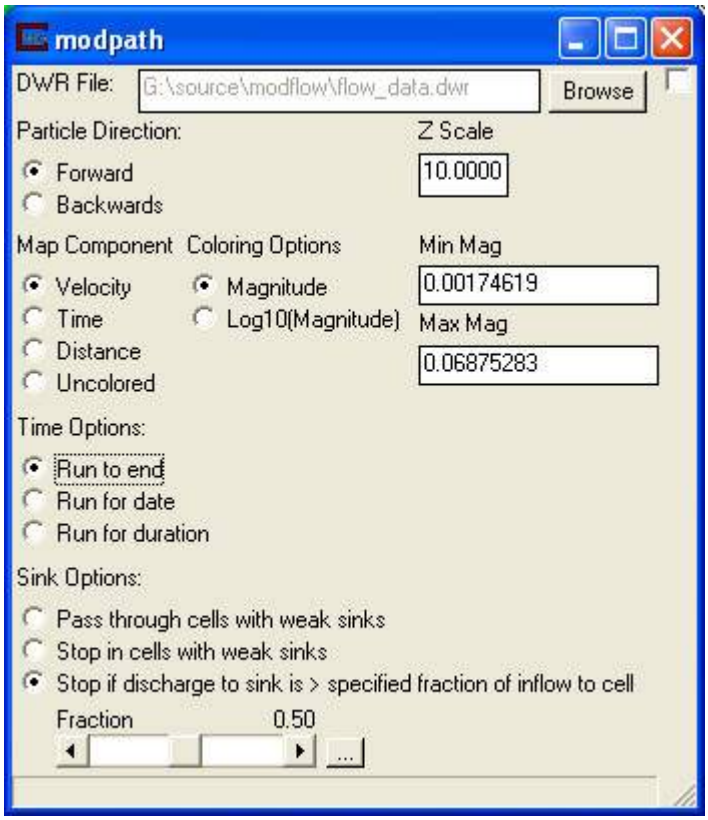
A valid modpath field file (.eff/.efz) should contain the following as cell data components: Head; CCF; ELEV_TOP; ELEV_BOT; and POROSITY. The Head component should contain the head value for each cell, the ELEV_TOP and ELEV_BOT should components should contain the elevation of the top of the cell, and the elevation of the bottom of the cell respectively, and the POROSITY should contain the flow due to porosity for that each cell. All other MODFLOW parameters (drains, wells, recharge, etc..) should be written into a .dwr/.dwz file.

Module Input Ports

The modpath module has five input ports. The first (leftmost) accepts field data usually from Load_EVS_Field or Read_TCF. The second port accepts data specifying the starting location of the particle paths, this data is typically generated by a slice, create_grid, isolines or place_glyph module. The third and fourth ports accept time values, this data can be generated with time_loop and is used to control the starting and ending values for transient models. The last port is the z exaggeration factor. This input causes the path generation to be calculated in an unscaled system ensuring that velocities are not scaled and are accurate, then scales the position of the paths.

Module Output Ports

modpath has five output ports. The first and second output ports (leftmost) pass the staring and ending times through. The third output port passes the z exaggeration factor. The fourth output port creates a new unstructured polyline mesh representing the particle paths. The output also contains a nodal data component referencing the velocity component. The rightmost and last output port can send modpath polylines to the viewer.



Module Control Panel

The control panel for modpath is shown in the figure above.

The*DWR File:*shows which .dwr/.dwz file has been selected, this file contains MODFLOW package data such as drains, wells, recharge, etc...For details on the DWR file format, see here.

The*Direction*radio buttons allow the user to specify forward or backward path projection. Both paths start from the specified starting points and travel until the selected option in either the "Time Options" category or the "Sink Options" category are satisfied.

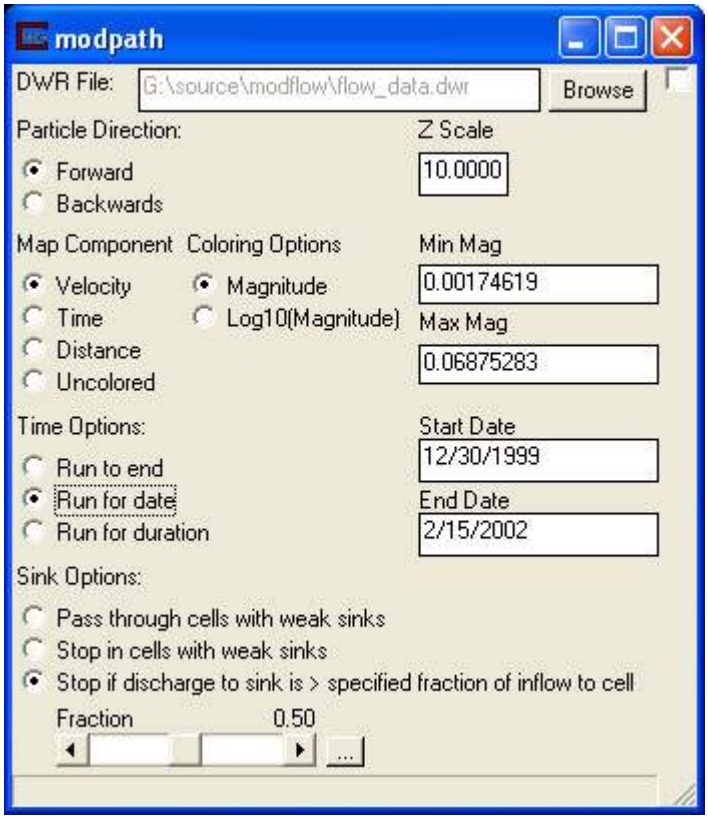
The*Z Scale*type-in reflects the value of the z exaggeration port.

The*Map Component*radio buttons allow you to choose which computed data component to use for coloring the lines. Choices are Velocity, Time, Distance and Uncolored.

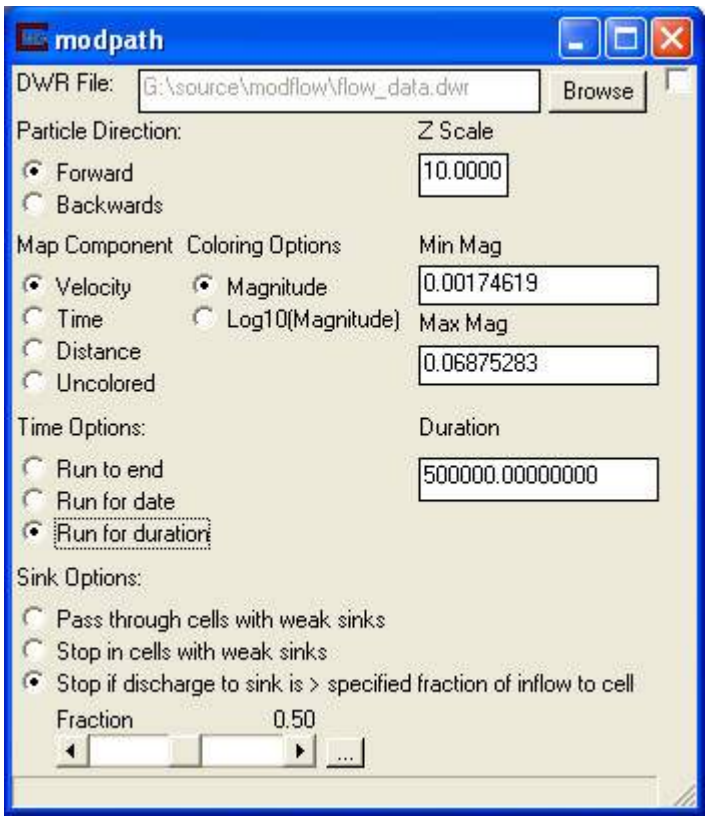
The*Coloring Options*radio buttons allow the user to specify*Magnitude*(normal) or*Log10(Magnitude)*representation of the velocity data for coloring purposes. When velocities span several orders of magnitude working in log space is useful.

The*Min Magnitude*and*Max Magnitude*type-ins display the min/max velocities represented in the input field. Changing these values allows the user to reset the min-max values for coloring purposes. NOTE: once you change these values they will not automatically update.

The*Time Options:*radio buttons indicate a time based stopping condition for the particles. Run to end indicates that the particle will run until it cannot escape a cell due to lack of flow.



Run for date allows a particle to travel from a start date to an end date.



The duration option allows the particles to run for a duration of time in days.

The*Sink Options*radio buttons indicate stopping conditions for the particles based on the amount of flow through a cell. The Pass through cells with weak sinks option allows all particles to pass through a cell, unless that cell has a strong sink. The Stop in cells with weak sinks button does not allow a particle to pass unless the flow out of a cell is greater than or equal the flow into it. The last option, Stop if discharge to sink is > specified fraction of inflow to cell, allows the user to specify what fraction of inflow will stop particles in that cell. This fraction value is set using the Fraction slider which is visible when the last option is selected. Only values between zero and one are useful.

advector

General Module Function

The advector module combines streamlines capability and a tool for sequential positioning of glyphs along the streamlines trajectory to simulate advection of weightless particles through a vector field (for example, a fluid flow simulation such as modflow). The result is an animation of particle motion, with the particles represented as any EVS geometry (such as a jet or a sphere). The glyphs can scale, deflect or deform according to the velocity vector it passes. At least one of the nodal data components input to advector must be a vector. The direction of travel of streamlines can be specified to be forwards (toward high vector magnitudes) or backwards (toward low vector

magnitudes) with respect to the vector field. The input glyphs travel along streamlines (not necessarily visible in the viewer) which are produced by integrating a velocity field using the Runge-Kutte method of specified order with adaptive time steps.

Module Input Ports

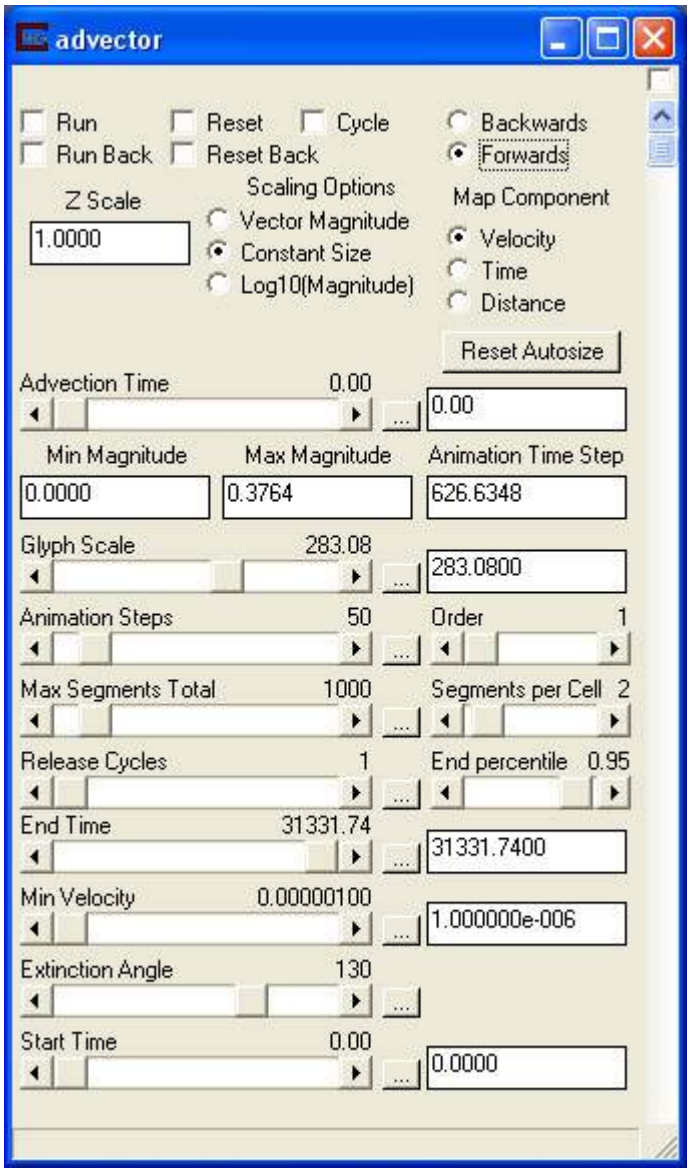
advector has three input ports.

- Data passed to the first port (the left port) can have a mesh of any type, and must have at least one vector nodal data component. To generate streamlines of scalar data, a module such as gradient must be used to calculate the gradient of a scalar field.
- The second port accepts data specifying the starting location of the streamlines. This data is typically generated by the create_grid, slice or isolines modules.
- The third port accepts any geometry mesh (glyph, usually read in with the load_glyph module) which will be rendered at each starting point and moved sequentially through the vector field during running of the advector module.

Module Output Ports

advector contains four output ports.

- The leftmost port outputs a mesh composed of the original mesh plus the meshes representing the particles.
- The second output port creates a new unstructured polyline mesh representing the streamlines, and also contains a nodal data component referencing the velocity component.
- The third output creates a renderable version of the glyphs.
- The fourth output port sends streamline polylines to the viewer.



Module Control Panel

The control panel for advector is shown in the figure above. Many of the parameters are related to streamline interpolation which is discussed in the streamlines help. Below is a discussion of those parameters unique to advector.

The ***Run*** toggle starts or stops advection.

The ***Cycle*** toggle starts the advection again at Start Time when End Time is reached.

The ***Reset*** Time toggle resets Time to the value of Start Time.

The ***Direction*** radio buttons allow the user to specify forward or backward streamlines. Forward streamlines start from the specified starting points and travel to the maximum velocity location. Backward streamlines travel from the specified starting points to the minimum or zero velocity location.

The ***Scaling Options*** radio buttons establishes how glyphs are sized based on the data values.

- ***Vector Magnitude*** sizes the glyphs proportional to the velocity at each node.
- ***Constant Size*** makes all glyphs the same size.
- ***Log10(Magnitude)*** scales by the log of the velocity. When velocities span several orders of magnitude working in log space is useful. When selected, an additional parameter is visible:
  - ***Magnitude for Zero Size*** defines the lower clamping velocity that will correspond to a ZERO (0.0) size glyph regardless of "Glyph Scale".

The ***Advection Time*** slider and type-in allow you to control the particle animation and also display the progress when Run is selected. The time runs from Start Time to End Time during running of advector.

The ***Advection Time Step*** type-in allows you to set the duration of each animation step.

The ***Glyph Scale*** slider adjusts the sizes of the glyphs. Default range is 0.0 to 100.00, but the type-in box allow for increasing the range to any value. The default scale of 1.00 is approximately 1/100th the x,y extents of the model.

The ***End percentile*** slider allows you to control the automatic scaling of glyphs based on the nth percentile value (versus the maximum 100th%). This addresses datasets where there are only a few nodes with extremely high values (like wells).

The ***Map Component*** radio buttons allow the user to color by Velocity, Time or Distance.

The ***Min Magnitude*** and ***Max Magnitude*** type-ins display the min/max velocities represented in the input field. Changing these values allows the user to reset the min-max values for coloring purposes. NOTE: once you change these values they will not automatically update.

The ***Z Scale*** type-in reflects the value of the z exaggeration port.

The **Segment per Cell** slider is used to set the number of integration steps to be used in each cell (i.e., the number of divisions of the cells) to calculate the streamline. The default is one and the range is 1 to 16.

The ***Max Segments Total*** slider is used to set the maximum allowable number of streamline segments that will be completed for each streamline. If the number of segments along a streamline exceeds the max number, the streamline is terminated at the end of the last (max) segment. The default is 15 and the range is 1 to 1000.

The **Order** slider is used to set the order of the integration. Higher order integration is more accurate, but executes much slower. The default is one and the range is 1 to 4.

The **Min Velocity** slider and type-in is used to specify the minimum velocity that will be considered in the integration. If the magnitude of the velocity field in a region is less than this minimum value, streamlines will end in that region (or will not be produced if the gradient at a starting point is less than the min). Setting this to a lower value will produce longer streamlines (and typically more). Higher values tend to produce less streamlines and shorter streamlines. If streamlines are not visible on the data set, setting this to a lower value may produce streamlines. The default value is 0.000001.

The ***Extinction Angle*** slider is used to specify the maximum allowable angle between successive line segments before integration (streamline generation) should be terminated. The default value is 130 degrees.

The ***Animation Steps*** slider controls the incrementation of the glyph stops between the Start Time and the time along the original streamline continuum for each advection step. For example, if the End Time is 10000 and the Animation Steps slider is 10 then there will be ten steps (frames) when the animation is run. The Animation Time Step Type-in box will reflect the time increment based on the slider choice. For the above example (10 steps) the type-in would read 1000.

The ***Release Cycles*** slider determines the number of times the particle glyphs will be sent from their release points. This could be thought of as the number of pulses of particle releases. For example, if the End Time is 10000, then a pulse of particles is released at the Start Time and again at 5000.

The ***End Time*** slider and type-in reflects the time value along the original streamline at which to halt advection of all particles. The default is 1000.0, but the type-in box allows for increasing the range to any value. Typically with groundwater flow problems (very low velocities) this number needs to be very large to achieve ample travel distance of particles.

Note that adjustments to either the End Time, Animation Steps, or Animation Time Steps will directly affect one or both of the remaining parameters. For example, adjusting the Animation Time Steps in the above example from 1000 to 2000 would increase the End Time from 10000 to 20000.

The ***Start Time*** slider reflects the time value along the original streamline continuum at which to start advection. The default is 0.0.

Related Modules

->glyph

->streamlines

advect_surface

General Module Function

The advect_surface module combines streamline_surface capability and a tool for sequential positioning of glyphs along the streamlines trajectory to simulate advection of particles down a surface. The result is an animation of particle motion, with the particles represented as any EVS geometry (such as

a jet or a sphere). The glyphs can scale, deflect or deform according to the velocity vector. The direction of travel of streamlines can be specified to be downhill or uphill (for the slope case). The input glyphs travel along streamlines (not necessarily visible in the viewer) which are produced by integrating a velocity field using the Runge-Kutte method of specified order with adaptive time steps.

The advect_surface module is used to produce streamlines and particle animations on any surface based on its slopes. The direction of travel of streamlines can be specified to be downhill or uphill for the slope case. A physics simulation option is also available which employs a full physics simulation including friction and gravity terms to compute streamlines on the surface.

Module Input Ports

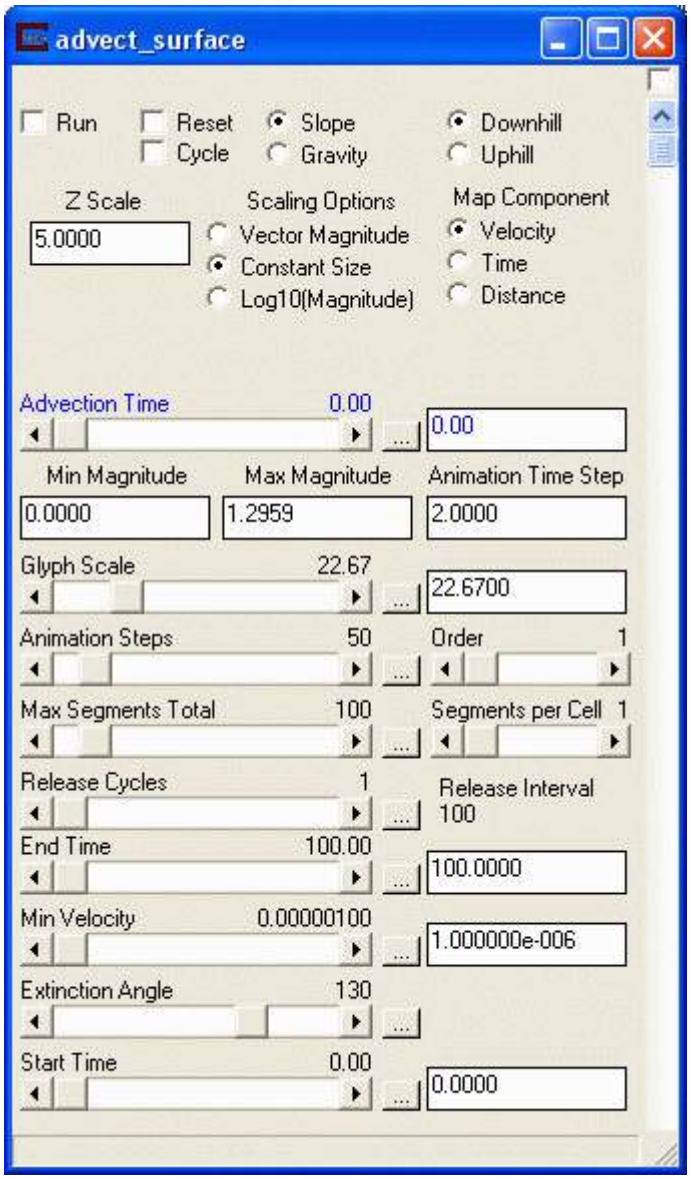
Advect surface has four input ports.

- Data passed to the first port (the left port) must be a surface.
- The second port accepts data specifying the starting location of the streamlines. This data is typically generated by the slice or isolines or place_glyph modules.
- The third port accepts the glyph, usually read in with the load_glyph module, to be used as the advector particles.
- The last port is the z exaggeration factor. This input causes the stream path generation to be calculated in an unscaled system ensuring that velocities are not scaled and are accurate, then scales the position of the streamlines.

Module Output Ports

advect_surface has two output ports.

- The leftmost output port creates a new unstructured polyline mesh representing the streamlines. The output also contains a nodal data component referencing the velocity component.
- The right output port can send either streamline polylines or renderable streamribbons to the viewer.



Module Control Panel

The control panel for advect_surface is shown in the figure above. Many of the parameters are related to streamline interpolation which is discussed in the streamline_surface help. Below is a discussion of those parameters unique to advect_surface.

The *Map Component* radio button allows you to choose which computed data component to use for coloring the lines. Choices are Velocity, Time, Distance and Uncolored.

The *Direction* radio buttons allow the user to specify forward or backward streamlines. Forward streamlines start from the specified starting points and travel to the maximum velocity location. Backward streamlines travel from the specified starting points to the minimum or zero velocity location.

The *Coloring Options* radio buttons allow the user to specify Magnitude (normal) or Log10(Magnitude) representation of the velocity data for coloring purposes. When velocities span several orders of magnitude this is useful.

The *Physics* radio buttons allow the user to specify whether streamlines will be computed based on the slopes of the surface only or whether a full physics simulation including friction and gravity terms will be used to compute streamlines on the surface. When *Gravity* is selected *Segments perCell* and *Order* do not apply but additional parameters appear for the module. These are:

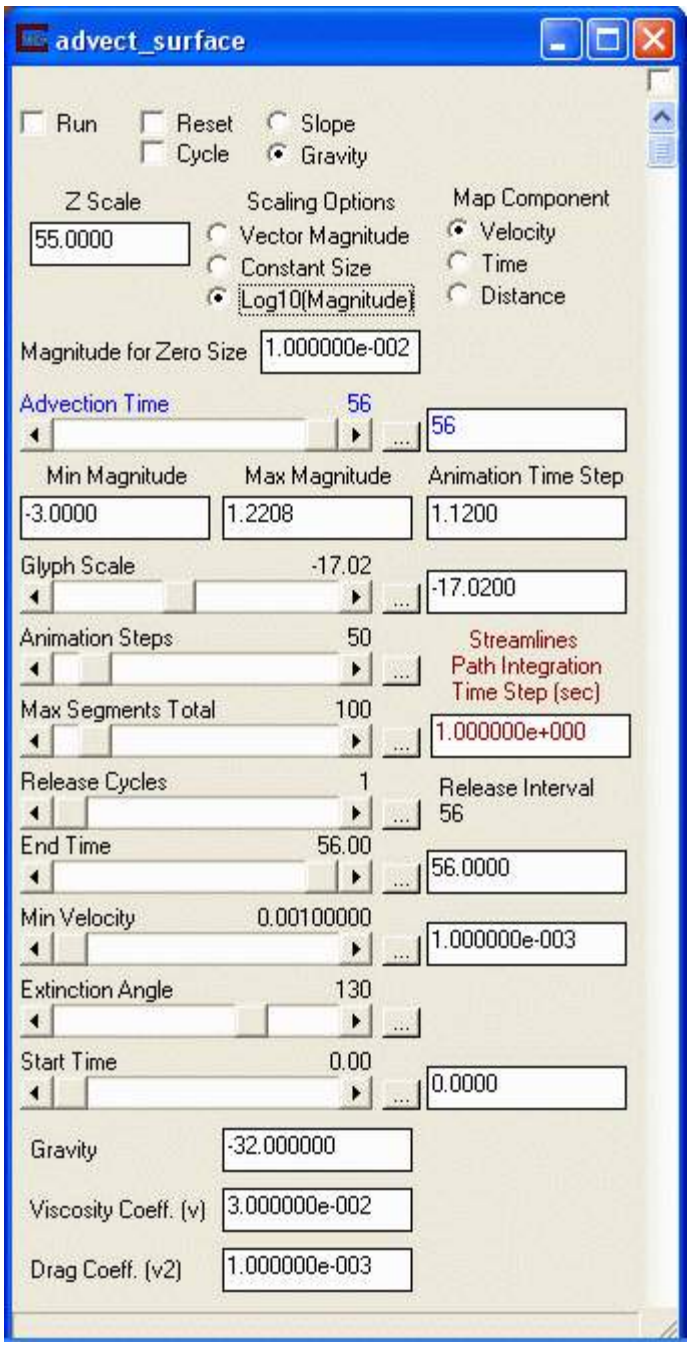
Integration Time Step is the time step for the numerical integration of the paths. For typical gravity units (like 32 feet per second-squared) this value is in seconds.

Gravity is the coefficient of gravity for your units. If your coordinate units are feet, the appropriate (default) value would be 32 feet per second-squared.

Viscosity Coefficient (v) is the friction term that depends on velocity.

Drag Coefficient (v2) is the friction term that depends on velocity-squared.

The panel when Gravity is selected is show below.



The **Min Magnitudeand *Max Magnitude*** type-ins display the min/max velocities represented in the input field. Changing these values allows the user to reset the min-max values for coloring purposes. NOTE: once you change these values they will not automatically update.

The ***Z Scale*** type-in reflects the value of the z exaggeration port.

The **Segment per Cell** slider is used to set the number of integration steps to be used in each cell (i.e., the number of divisions of the cells) to calculate the streamline. The default is one and the range is 1 to 16.

The **Max Segments** *Total* slider is used to set the maximum allowable number of streamline segments that will be completed for each streamline. If

the number of segments along a streamline exceeds the max number, the streamline is terminated at the end of the last (max) segment. The default is 15 and the range is 1 to 1000.

The ***Release Cycles*** slider determines the number of times the particle glyphs will be sent from their release points. This could be thought of as the number of pulses of particle releases. For example, if the End Time is 10000, then a pulse of particles is released at the Start Time and again at 5000.

The **Order** slider is used to set the order of the integration. Higher order integration is more accurate, but executes much slower. The default is one and the range is 1 to 4.

The **Min Velocity** slider and type-in is used to specify the minimum velocity that will be considered in the integration. If the magnitude of the velocity field in a region is less than this minimum value, streamlines will end in that region (or will not be produced if the gradient at a starting point is less than the min). Setting this to a lower value will produce longer streamlines (and typically more). Higher values tend to produce less streamlines and shorter streamlines. If streamlines are not visible on the data set, setting this to a lower value may produce streamlines. The default value is 0.000001.

The ***Extinction Angle*** slider is used to specify the maximum allowable angle between successive line segments before integration (streamline generation) should be terminated. The default value is 130 degrees.

modpath_advector

General Module Function

The modpath_advector module combines MODPATH capability and a tool for sequential positioning of glyphs along the MODPATH lines trajectory to simulate advection of weightless particles through a vector field. The result is an animation of particle motion, with the particles represented as any EVS geometry (such as a jet or a sphere). The glyphs can scale, deflect or deform according to the velocity vector it passes. The direction of travel of streamlines can be specified to be forwards (toward high vector magnitudes) or backwards (toward low vector magnitudes) with respect to the vector field. The input glyphs travel along streamlines (not necessarily visible in the viewer) which are produced by integrating a velocity field using the Runge-Kutte method of specified order with adaptive time steps.

Module Input Ports

The modpath_advector module has four input ports.

- The first (leftmost) accepts field data usually from Load_EVS_Field or Read_TCF.
- The second port accepts data specifying the starting location of the particle paths, this data is typically generated by a slice, create_grid, isolines or place_glyph module.
- The third port accepts any geometry mesh (glyph, usually read in with the load_glyph module) which will be rendered at each starting point

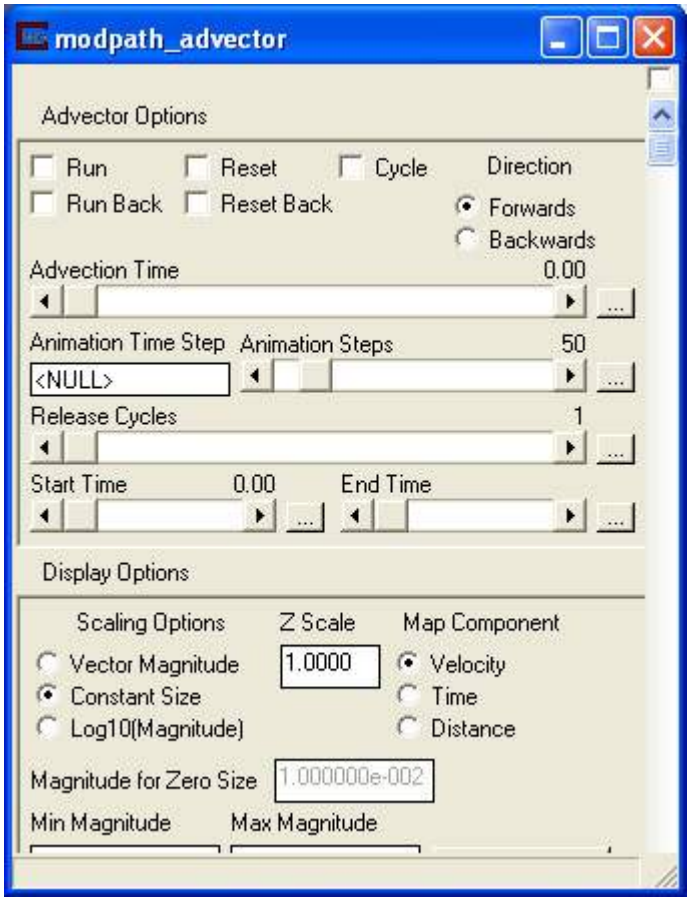
and moved sequentially through the vector field during running of the advector module.

- The last port is the z exaggeration factor. This input causes the path generation to be calculated in an unscaled system ensuring that velocities are not scaled and are accurate, then scales the position of the paths.

Module Output Ports

Modpath_advector contains four output ports.

- The leftmost port outputs a mesh composed of the original mesh plus the meshes representing the particles.
- The second output port creates a new unstructured polyline mesh representing the MODPATH lines, and also contains a nodal data component referencing the velocity component.
- The third output creates a renderable version of the glyphs.
- The fourth output port sends MODPATH polylines to the viewer.



Module Control Panel

The control panel for modpath_advector is shown in the figure above.

Advector options :

The ***Run*** toggle starts or stops advection.

The ***Cycle*** toggle starts the advection again at Start Time when End Time is reached.

The ***Reset*** Time toggle resets Time to the value of Start Time.

The ***Direction*** radio buttons control the direction of travel of the glyphs.

The ***Advection Time*** slider and type-in allow you to control the particle animation and also display the progress when Run is selected. The time runs from Start Time to End Time during running of advector.

The ***Advection Time Step*** type-in allows you to set the duration of each animation step.

The ***Animation Steps*** slider controls the incrementation of the glyph stops between the Start Time and the time along the original streamline continuum for each advection step. For example, if the End Time is 10000 and the Animation Steps slider is 10 then there will be ten steps (frames) when the animation is run. The Animation Time Step Type-in box will reflect the time increment based on the slider choice. For the above example (10 steps) the type-in would read 1000.

The ***Release Cycles*** slider determines the number of times the particle glyphs will be sent from their release points. This could be thought of as the number of pulses of particle releases. For example, if the End Time is 10000, then a pulse of particles is released at the Start Time and again at 5000.

The ***End Time*** slider reflects the time value along the original streamline at which to halt advection of all particles. The default is 1000.0, but the range may be increased to any value. Typically with groundwater flow problems (very low velocities) this number needs to be very large to achieve ample travel distance of particles.

Note that adjustments to either the End Time, Animation Steps, or Animation Time Steps will directly affect one or both of the remaining parameters. For example, adjusting the Animation Time Steps in the above example from 1000 to 2000 would increase the End Time from 10000 to 20000.

The ***Start Time*** slider reflects the time value along the original streamline continuum at which to start advection. The default is 0.0

Display Options:

The ***Scaling Options*** radio buttons establishes how glyphs are sized based on the data values.

Vector Magnitude sizes the glyphs proportional to the velocity at each node.

Constant Size makes all glyphs the same size.

Log10(Magnitude) scales by the log of the velocity. When velocities span several orders of magnitude working in log space is useful. When selected, an additional parameter is visible:

Magnitude for Zero Size defines the lower clamping velocity that will correspond to a ZERO (0.0) size glyph regardless of "Glyph Scale".

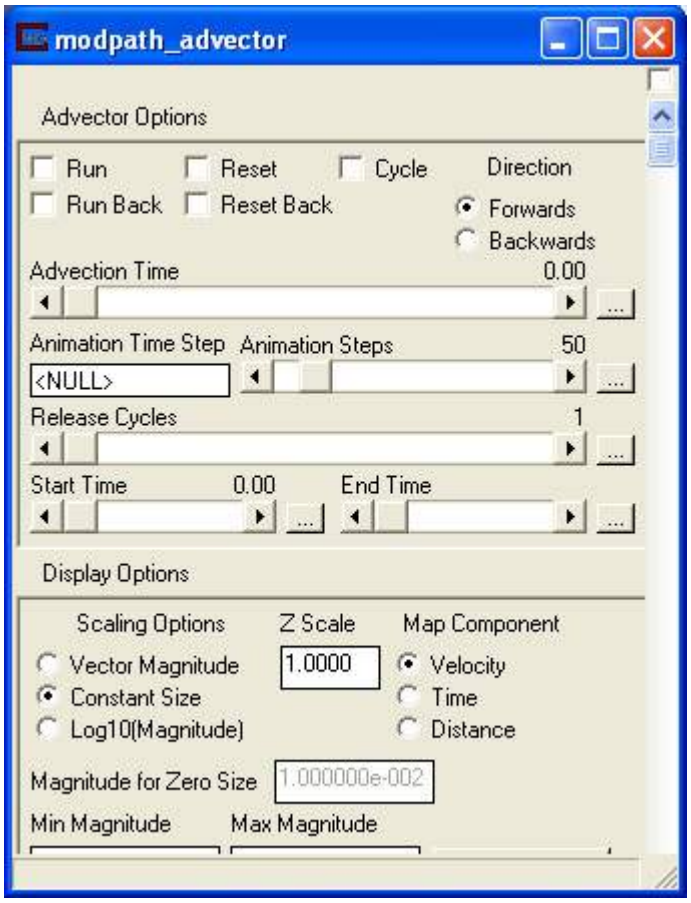
Min / Max Magnitude sets limits on the glyph scale.

The ***Glyph Scale*** slider adjusts the sizes of the glyphs. Default range is 0.0 to 100.00, but the type-in box allow for increasing the range to any value. The default scale of 1.00 is approximately 1/100th the x,y extents of the model.

The ***End percentile*** slider allows you to control the automatic scaling of glyphs based on the nth percentile value (versus the maximum 100th%). This addresses datasets where there are only a few nodes with extremely high values (like wells).

Z Scalecontrols the z exaggeration factor.

Map Componentselects which MODPATH data component to color the out put lines by.



Model Options:

The ***DWR File****:* shows which .dwr/.dwz file has been selected, this file contains MODFLOW package data such as drains, wells, recharge, etc... For details on the DWR file format, see here.

The ***Time Options*** radio buttons indicate a time based stopping condition for the particles.

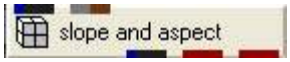
Run to end indicates that the particle will run until it cannot escape a cell due to lack of flow.

Run for date allows a particle to travel from a start date to an end date.

The ***duration*** option allows the particles to run for a duration of time in days.

The ***Sink Options*** radio buttons indicate stopping conditions for the particles based on the amount of flow through a cell. The Pass through cells with weak sinks option allows all particles to pass through a cell, unless that cell has a strong sink. The Stop in cells with weak sinks button does not allow a particle to pass unless the flow out of a cell is greater than or equal the flow into it. The last option, Stop if discharge to sink is > specified fraction of inflow to cell, allows the user to specify what fraction of inflow will stop particles in that cell. This fraction value is set using the Fraction slider which is visible when the last option is selected. Only values between zero and one are useful.

slope_and_aspect



General Module Function

The slope_and_aspect module determines the slope and aspect of a surface. The slope is the angle between the surface and the horizon. The aspect is the cardinal direction in degrees (rotating clockwise with 0° being North) that the slope is facing.

Module Input Ports

slope_and_aspect surface has two input ports.

Data passed to the first port (the left port) must be a surface.

The second port is the z exaggeration factor.

Module Output Ports

slope_and_aspect has three output ports.

The first port(the left port) outputs the surface with both slope and aspects as its cell data components.

The second port sends the surface with cells colored by slope to the viewer.

The third port sends the surface with cells colored by aspect to the viewer.

Module Control Panel

Flat Surface Aspect : The aspect for most cells is generated from the slope of the cells, however if you have a flat surface then the Flat Surface Aspect will set the aspect angle.

Z Scale: This sets the z exaggeration factor.

glyph

General Module Function

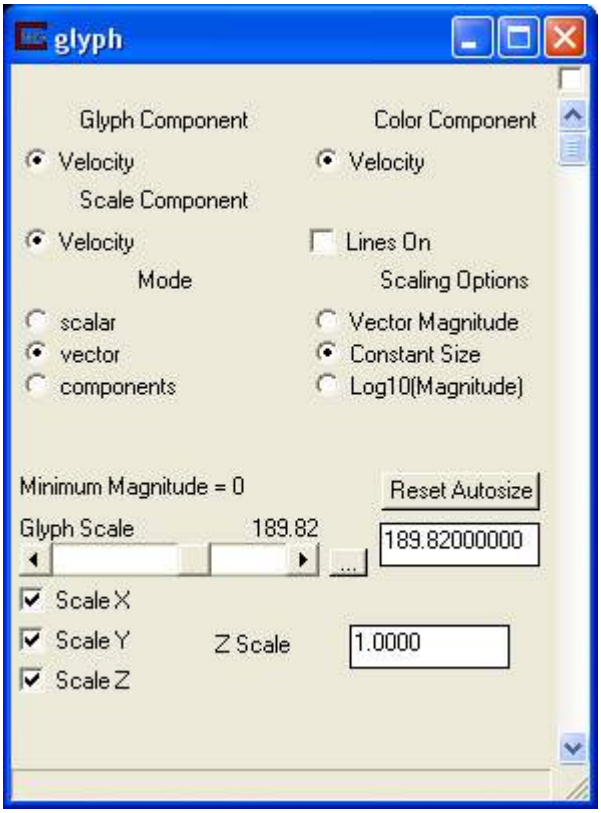
The glyph module is used to place geometric objects (glyphs) at nodal locations. The glyphs can be scaled, rotated and colored based on the input data. If the input data is a vector, the glyph can be scaled and rotated to represent the direction and absolute magnitude of the vector field. In a scalar data field, the objects can be scaled based on the magnitude of the scalar. The glyphs can represent the data field of one data component while being colored by another data component. Arrow glyphs are commonly used in vector fields to produce visualizations of the vector field.

Module Input Ports

Glyph has three input ports. Data passed to the first port (closest to the left) can have a mesh of any type and must have at least one nodal data component. The nodal data components can be vector or scalar. The second port accepts grid data specify the glyph object, usually read in with the load_glyph module. The third port is the z-exaggeration of the input data field.

Module Output Ports

Glyph has two output ports. The leftmost output port creates a new unstructured mesh containing the positions of all the glyphs. Nodal data sent to this port contains the magnitude of the vector or scalar data. The second output port sends the glyph geometry to the viewer.



Module Control Panel

The control panel for glyph is shown in the figure above.

The *Glyph component* radio button list displays all data components passed to glyph. Glyph component determines which data component is used to scale and rotate the glyphs. The default selection is the first (0th) data component.

The *Color Component* radio button list also displays all data components passed to glyph. Color component determines which data component is used to color the glyphs. By default, the first (0th) data component is selected.

The *Scale Component* radio button list also displays all data components passed to glyph. Scale component determines which data component is used to scale the glyphs. By default, the first (0th) data component is selected.

The *Mode* radio buttons are used to determine how the glyph component is to be handled. The default is vector. The scalar mode scales the glyph by the magnitude of the scalar data value at each node. If the data is scalar, this mode should be used and the nodal data should be greater than or equal to zero, since negative scales factors for object size don't make a lot of sense and will create a distribution of sizes that is not physical. Use field math to adjust scalar data if necessary. The vector mode scales the glyph by the magnitude of the vector, and also rotates the glyph in 2 or 3 dimensions based on the vector subcomponent values. The component mode scales the glyph in 2 or 3 dimension using the magnitudes of the vector subcomponents. For example, a Cross3D glyph in a 3D vector field would have each of its three lines (X, Y, and Z) scaled by different vector subcomponents.

The *Scaling Options* radio buttons determine how the glyphs will be sized.

Vector Magnitude specifies the size of the glyphs will be proportional to the relative magnitude of the data component values at each node.

Constant size specifies the size of the glyphs will be determined directly by Glyph Scale.

Log10(Magnitude) specifies the size of the glyphs will be proportional to the log10 of the magnitude of the data component values at each node. This requires another parameter which is the *Magnitude for Zero Length*.

Magnitude for Zero Length is the vector magnitude at which the glyphs will be zero size. This establishes the minimum level for taking log values since the log of zero is undefined (minus infinity).

The *Glyph Scale* slider and type-in is used to control the size of the glyphs. The default is automatically computed based on your input data. If you change the auto-computed value, you can use the *Reset* Autosize button to recalculate a new size.

The *Scale X, Scale Y, and Scale Z* toggles determine how the glyph will be scaled. The default is to have all three on and scale the glyph in all three directions.

The *Z Scale* type in is z-exaggeration of the input data field.

Related Modules

->**streamlines**

-> **gradient**

drive_glyph

General Module Function

The drive_glyph module provides a way to move any object (glyph or object from Read_DXF, etc.) along a path to create a "driving" animation.

Module Input Ports

drive_glyph has three input ports.

Data passed to the first port is the path to follow (normally from polyline_spline).

The second port accepts the glyph or vehicle to drive, usually read in with the load_glyph module.

The third port is a float parameter for the position of the glyph.

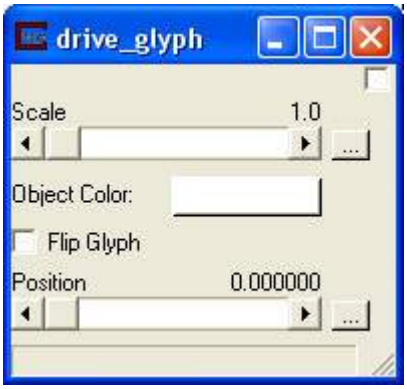
Module Output Ports

drive_glyph has three output ports.

The leftmost output port is a float parameter for the position of the glyph along the input path.

The center port is the animated glyph.

The right output port is the animated glyph in a renderable form for the viewer.



Module Control Panel

drive_glyph's control panel is shown above.

The Scale slider determines the size of the glyph.

The Object Color button sets the glyph color.

Flip Glyph toggle reverses the glyph direction.

Position is a slider for the position of the glyph along the input path.

drive_glyphs

General Module Function

The drive_glyph module provides a way to move any object (glyph or object from Read_DXF, etc.) along multiple paths to create a "driving" animation.

Module Input Ports

drive_glyphs has three input ports.

Data passed to the first port is the paths to follow (normally from read_lines).

The second port accepts the glyph or vehicle to drive, usually read in with the load_glyph module.

The third port is a float parameter for the position of the glyphs.

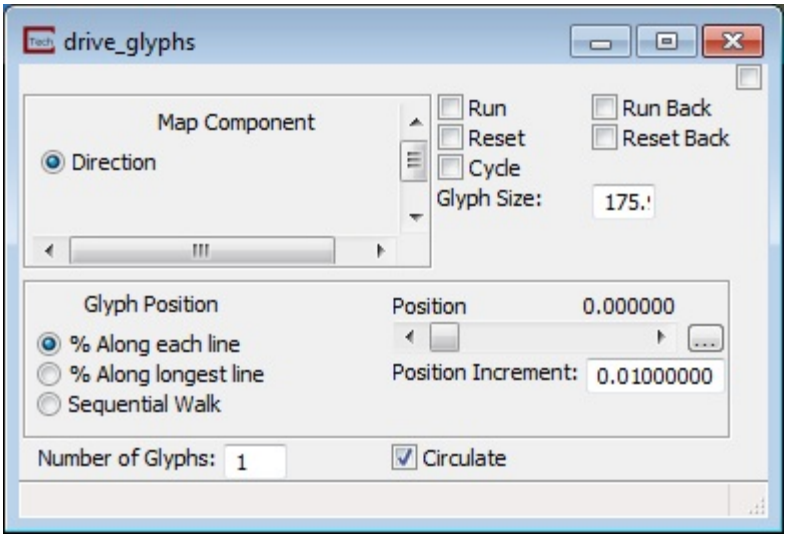
Module Output Ports

drive_glyph has three output ports.

The leftmost output port is a float parameter for the position of the glyphs along the input paths.

The center port is the animated glyphs.

The right output port is the animated glyphs in a renderable form for the viewer.



Module Control Panel

drive_glyph's control panel is shown above.

- The **Map Component** selector takes both the nodal data and the cell data from the input lines and allows the glyphs to be colored according to the selected component. For nodal data the color of the glyph is linearly interpolated between each node along the line, while with cell data the color of the glyph is constant throughout the cell.
- The **Run** toggle will increment the Position slider by the Position Increment until the value is 1.
- The **Run** Back toggle will decrement the Position slider by the Position Increment until the value reaches 0.
- The **Reset** toggle will set the Position to zero.

- The **ResetBack** toggle will set the Position to 1.
- The **Cycle** toggle will cause the Position slider to transition to the opposite end if the slider value reaches zero or one. This allows for a looping animation.
- The **Glyph Size** field sets the size of the glyph in real world units.
- **Glyph Position** indicates how each glyph will be placed along a line.
- If "***% Along each line*** is selected then the glyphs positions will be determined based upon each individual line length.
- If "***% Along longest line*** is selected then the glyphs position will be determined based on the longest line in the input.
- ***Sequential Walk*** will allow a glyph to walk along each line in succession.
- The **Position** slider gives the percentage along a line to use for glyph positioning.
- The **Position increment** changes the value the percentage is incremented by.
- **Number of glyphs** indicates how many glyphs at most will be present along the line.
- The **Circulate** toggle is used for looping or cycling animations. If a glyphs position is greater than the line length the glyph will be placed the appropriate distance from the start of the line. Or if the glyph position is before the end of the line it will be placed the appropriate distance from the end of the line. In example if you have selected 3 glyphs to be placed on a line (meaning that each glyph should be placed 33% of the line distance apart) and the Position is at zero. Then there will be one glyph at zero and another at 100 - 33 or 66% and a final glyph at 100 - 66 or 33% of the distance along the line length.

place_glyph

General Module Function

The place_glyph module is used to place a single scalable geometric objects (glyph) at an interactively determined location.

Module Input Ports

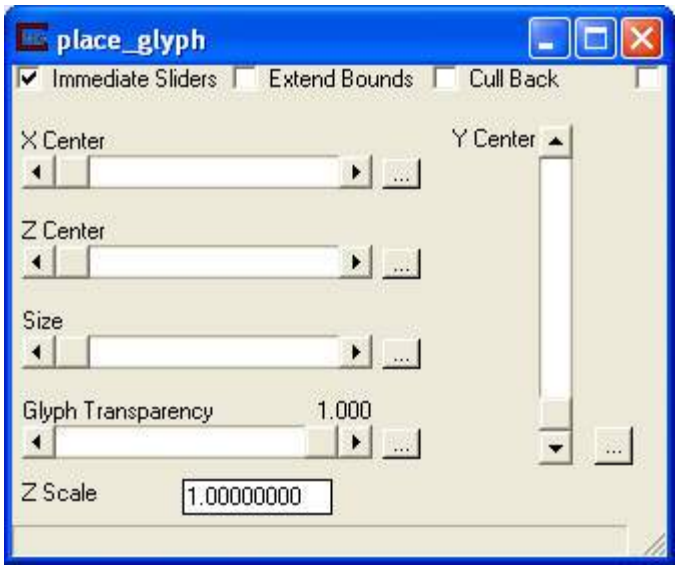
Place_glyph has three input ports.

1. The first port (closest to the left) can have a 3D mesh of any type and is used to set the bounds for the placement of the glyph.
2. The second (grey-brown) connects to Explode_and_Scale to inherit the Z_Exaggeration factor.
3. The third port accepts the glyph object, usually read in with the load_glyph module.

Module Output Ports

Place_glyph has two output ports. The leftmost output port creates the glyph.

The second output port sends the glyph geometry to the viewer.



Module Control Panel

The control panel for place_glyph is shown in the figure above.

The scale type-in is used to control the size of the glyphs. The default is 10 and the range is from -2000 to 2000. Negative scale numbers result in reversal of the direction in which the glyph is drawn. This is useful, for example, to draw arrows flowing in the backward gradient direction (high to low; see the gradient module discussion) as in ground water flow.

geo_glyph

General Module Function

The geo_glyph module is used to place differing types of geometric objects (glyphs) according to the attributes at nodal locations. This functionality could be used for example to place a different glyph type for each geologic layer type in a model. The unique nature of geo_glyph (as compared with glyph) is that it can be passed multiple glyphs to be displayed according to the range of values in the input field.

The glyphs can be scaled, rotated and colored based on the input data. In a scalar data field, the objects can be scaled based on the magnitude of the scalar. The glyphs can represent the data field of one data component while being colored by another data component.

Module Input Ports

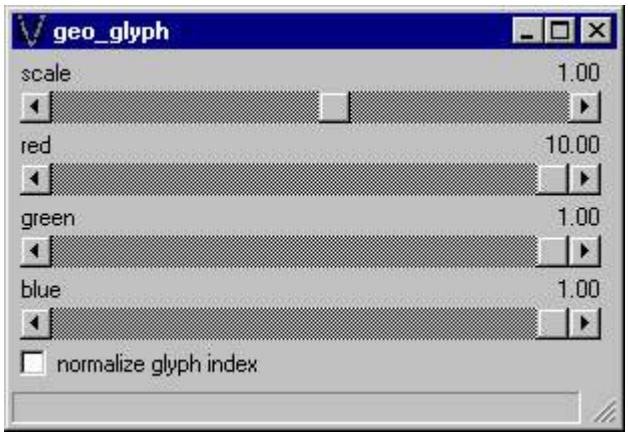
geo_glyph has two input ports.

- Data passed to the first port (closest to the left) can have a mesh of any type and must have at least one nodal data component. The nodal data components can be vector or scalar.
- The second port accepts the glyph object, usually read in with the load_glyph module.

Module Output Ports

geo_glyph has two output ports.

- The leftmost output port creates a new unstructured mesh containing the positions of all the glyphs. Nodal data sent to this port contains the magnitude of the vector or scalar data.
- The second output port sends the glyph geometry to the viewer.



Module Control Panel

The control panel for geo_glyph is shown in the figure above. The scale slider determines the relative size of the glyphs. The Red, Green and Blue sliders control coloring of the glyphs. The normalize of toggle disables scaling of glyphs according to magnitude of the scalar value.

Related Modules

->advector

-> glyphs

explode_fault_blocks

(This module is available only in MVS)

General Module Function

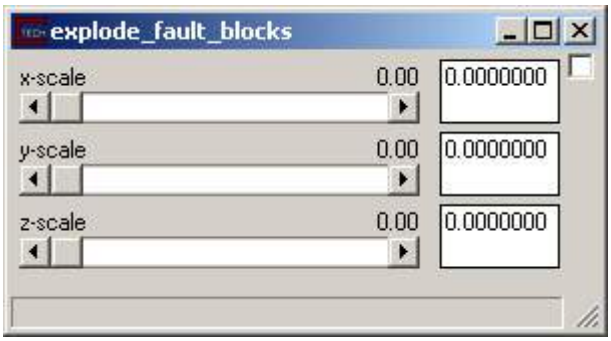
explode_fault_blocks takes each input field in the input array and translates it away from the common center of all the fields, which is computed by finding the midpoint of the bounding box. As the name implies, these fields are typically, separate fault blocks created by surf_cut, but they can actually be any type of mesh. You can set the amount of translation via parameters. The grid and data for the input fields are not modified, only their transformation matrices.

Module Input Ports

explode_fault_blocks has one input port which can accept multiple inputs of any type of EVS field including fields (any type) or UCD mesh input. There are also three parameter ports for hooking to loop modules, thereby automating animation of an exploding sequence.

Module Output Ports

explode_fault_blocks has two output ports. The first output port (closest to the left) outputs a mesh containing the transformed input fields. The second port outputs a renderable object.



Module Control Panel

The control panel for explode_fields is shown in the figure above. The x-scale, y-scale and z-scale sliders adjusts values specifying how much to scale the translation away from the center in the x,y or z directions. This value is in units of the bounding box of the original array of fields.

Related Modules

->Explode_and_Scale

render_field

General Module Function

render_field is a renderable object that contains the data to be rendered and other subobjects that contain the attributes that control how the data is rendered. In simple terms the render_field module enables rendering of non-renderable objects. This module can be connected directly to the EVS viewer or to GroupObject. The most common use for the module is for placing at the end of a stream of multiple non-renderable modules and then sending for rendering in the viewer.

Render_field essentially combines the following for use by the viewer module:

* DefaultMinMax to calculate the minimum and maximum values of any node or cell data.

* DefaultDatamap to convert scalar node or cell data to RGB color values. By default, the datamap is ranged to the values calculated by DefaultMinMax.

* DefaultProps to control color, material, line attributes, and geometrical attributes.

* DefaultModes to control point, line, surface, volume, and bounds rendering modes.
* DefaultPickInfo to contain information when this object is picked.
* DefaultTexture to control the texture mapping attributes.
* DefaultObject to control visibility, pickability, caching, transform mode, surface conversion, and image display attributes.

Module Input Ports

This module has two input ports.

The leftmost (blue-black) port accepts any mesh and nodal data. Using this port allows you to connect objects to the viewer from modules that do not have a Red output port.

The second (red) port can accept any number of Red input ports. This functionality is identical to the Group Object module.

NOTE: Do not use both the blue/black and red input ports simultaneously.

Module Output Ports

Render_field outputs a renderable object that can be connected directly to the EVS viewer.

Module Control Panel

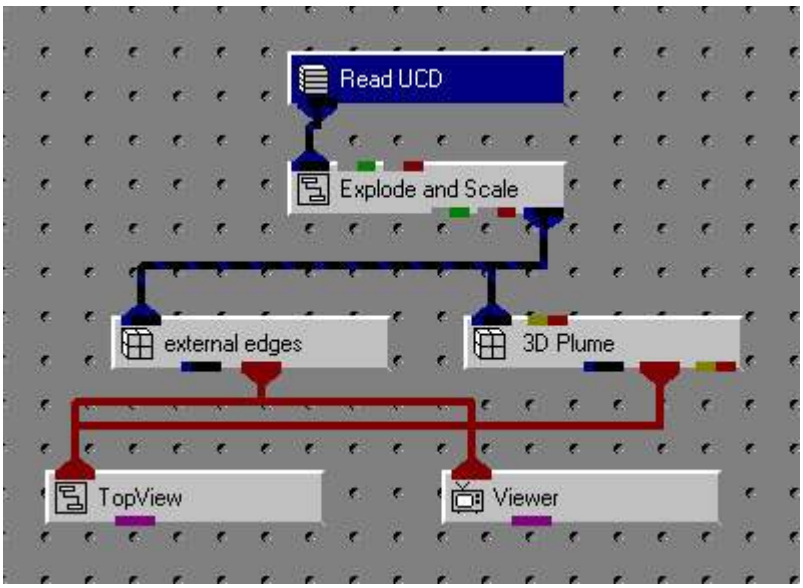
This module has no user interface.

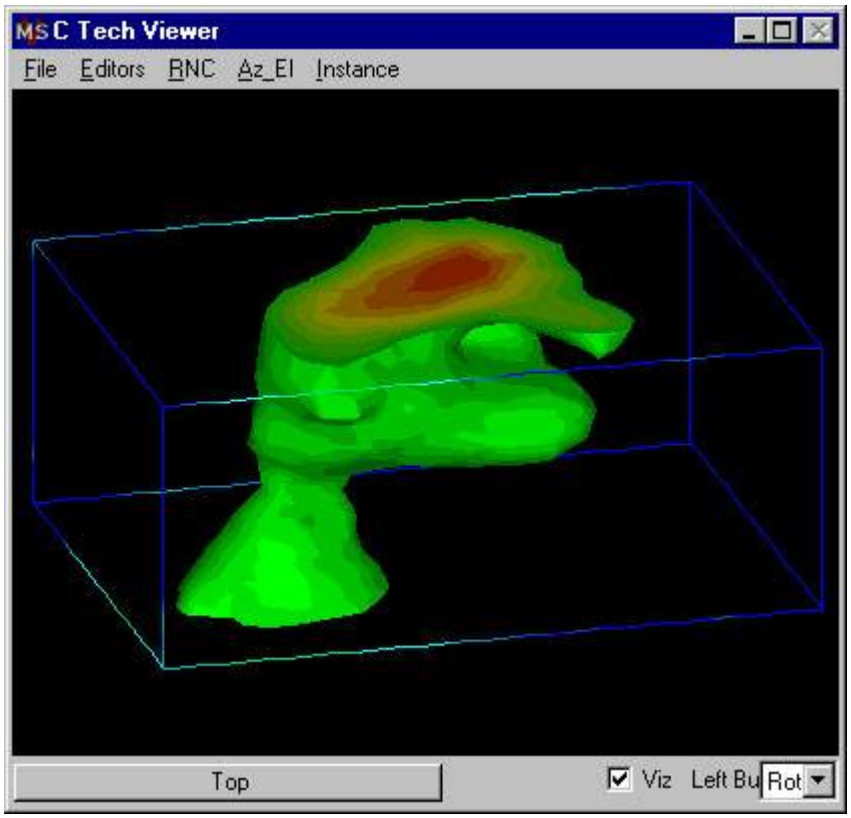
Related Modules

->GroupObject

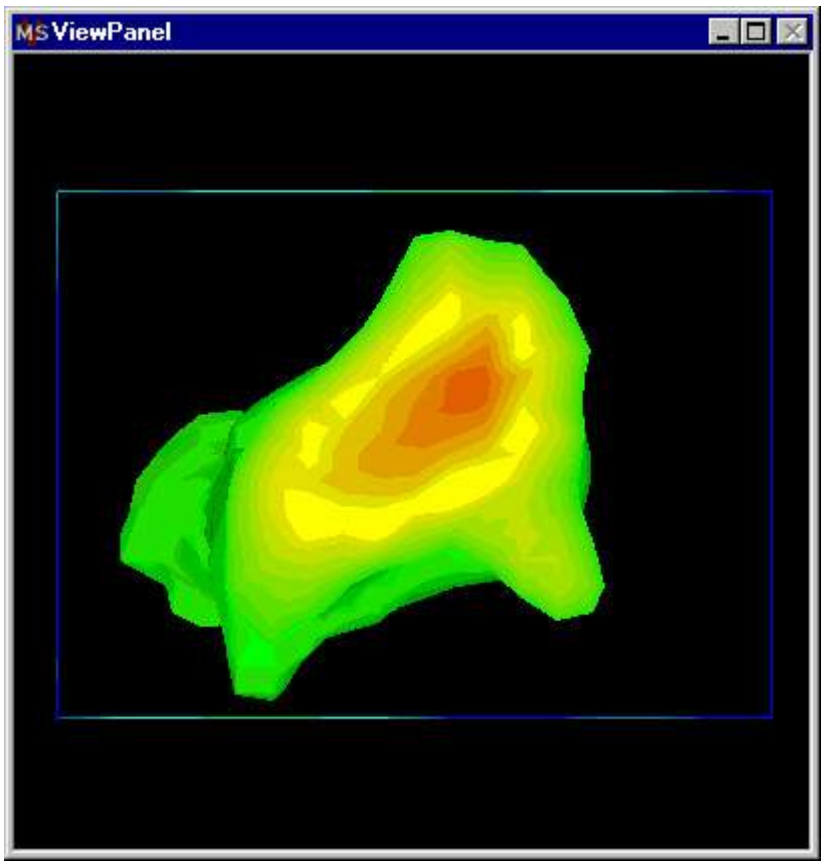
TopView

TopView: When objects are connected to this new module (like a Viewer) it renders a TOP view (cannot be zoomed, rotated or panned). It is a very efficient way to keep a Top view of your entire scene.





A view from the TopView's view panel



_2D_Overlay Group

General Module Function

2D_Overlay provides a module that applies any connected module's output to the Viewer's 2D overlay. Objects in the overlay are not transformed (rotated, zoomed, panned). These objects are locked in position. This provides a mechanism to apply graphics like drawing title blocks or company logos.

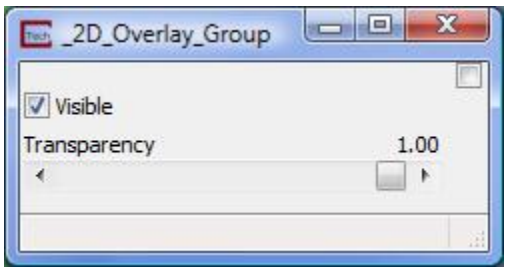
However, you must ensure that the object sent to the 2D overlay fits inside its limited spatial extent. The 2D overlay is a window with an x-extent from -1.0 to 1.0. The y-extent is dependent on the aspect ratio of the viewer. With a default Viewer having a 4:3 aspect ratio, it is three-quarters of the x-extent (e.g. -0.75 to 0.75).

Module Input Ports

Optionally one or more objects to be children of this object. Multilevel object hierarchies can be built in this manner.

Module Output Ports

The renderable NON-TRANSFORMABLE object that is connected directly to a viewer.



Module Control Panel

The user interface for this module is shown above.

Related Modules

-> GroupObject

Transform_Group

General Module Function

Transform_Group is a special GroupObject that allows all connected objects to be rotated (about a user defined center) and/or translated. This is useful if you wish to move objects that are complex, such as group objects like post_samples or axes and therefore cannot be contained in a single field (blue-black) port.

An example of this, would be the axes module. If you wanted an axes with an origin that did not match your data, it could be created separately and moved using the Transform_Group module.

Limitations

In some circumstances Transform_Group cannot be used with 4DIMs. It can cause the 4DIM extents to be different than they were in the EVS/MVS Viewer. This has been noted when doing rotations.

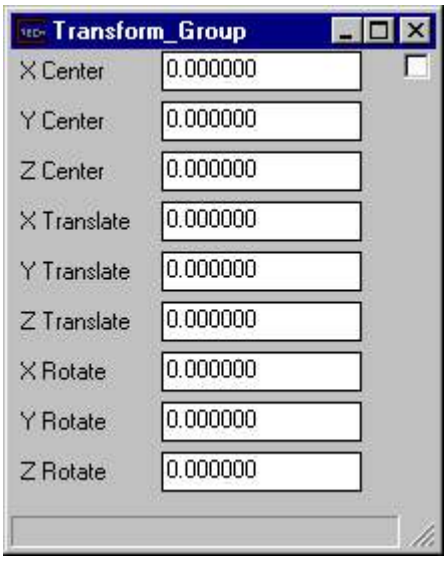
In most cases, the transform_field module can be used instead, however it does not allow for multiple objects to be connected to its input.

Module Input Ports

This module has a single red input port that can accept one or more objects (red ports) as inputs. Each input will become a child object of the output. Multilevel object hierarchies can be built in this manner.

Module Output Ports

The renderable object that can be connected directly to a viewer.



Module Control Panel

The user interface for Transform_Group consists of numerous type-in fields that control the center for rotation, the rotation angles (in degrees) and translations for each of the three coordinate axes.

Related Modules

Render_Field

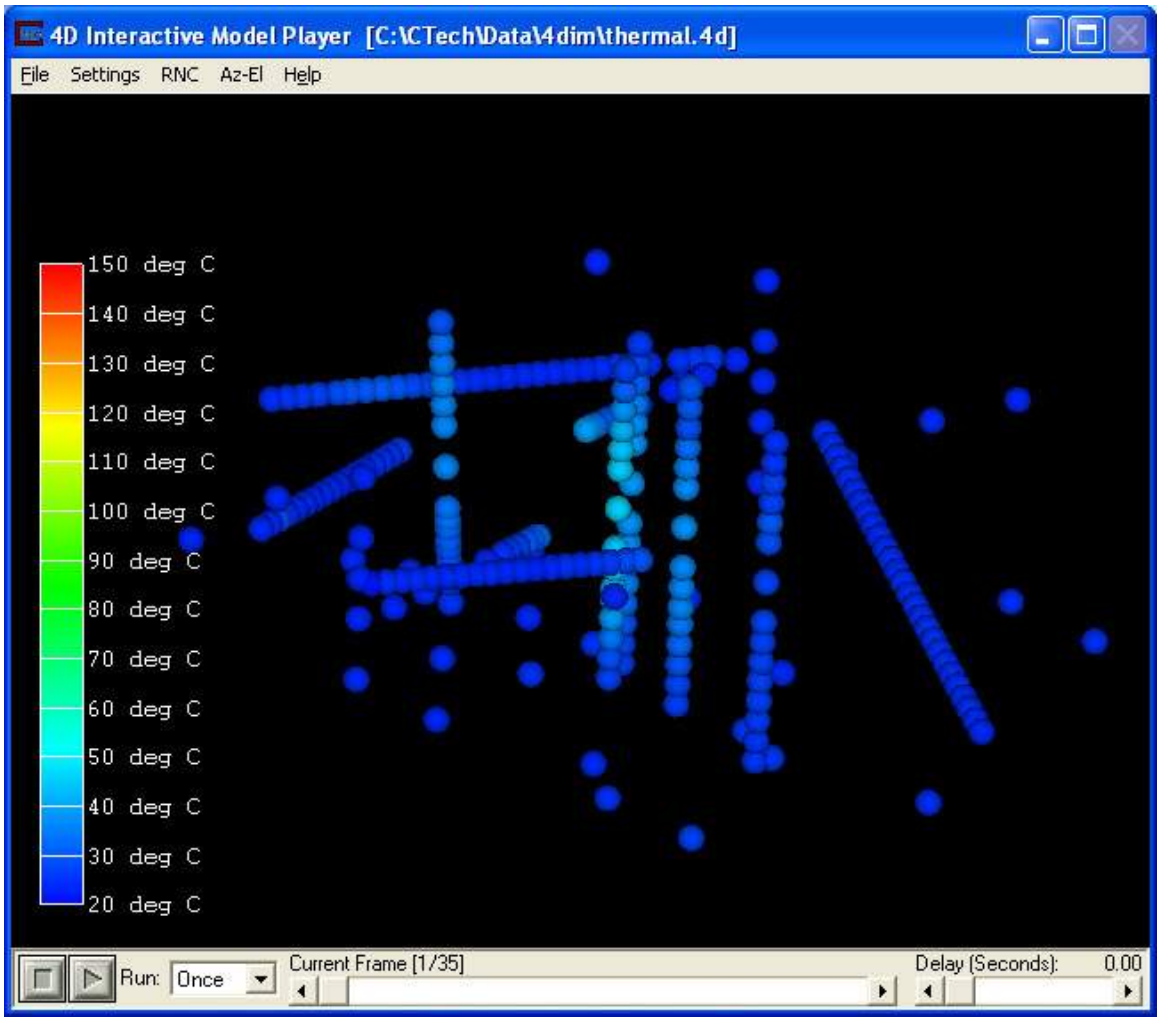
GroupObject

4D Interactive Model Player

The module for the player integrated in C Tech software is:

The Playback _4DIM module and the stand-alone 4DIM Player are identical in functionality with very minor exceptions. The players include an integrated C Tech Viewer with many of its features. Mouse interactions are nearly identical to the Viewer's. The bottom of the player window includes script controls and more.

After loading the file specified in the previous topic, the player's window should look like:



Shortcuts

There are a few keyboard shortcuts worth noting. These will give you quick control over the player.

- CTRL-F sets the player to FULL SCREEN mode. This is not equivalent to the maximize button in the upper right corner since this removes the normal borders.
- With the player as the active window, ESC(ape) exits Full Screen mode
- CTRL-H toggles the Auto-Hide mode.

The VCR type buttons on the player panel allows you to run or pause the selected scipt. The function of each button is:

Stop: Stop script from playing.

Pause: Pause script.

Play: The play button will open the Choose 4DIM View window. This window selects the current script to be played by the 4DIM player. There are

standard scripts that can be played with every 4DIM or complex scripts that can be created and bundled for specific 4DIM's. If a 4DIM has been paused while playing a script this button will instead resume the playing of the script.

Start by hitting the Play button.Select "Play Each Frame" from the Standard Script Library and then select the "Play Script" button.Notice that the animation plays from the beginning to the end only one time. There are options to change this behavior.

The **Run** option menu allows for specifying three different ways of playing the animation:

Once displays the script one scene at a time.

Cycle will display the script one scene at a time, when the script is complete it will repeat this process.

Bounce also displays the script one scene at a time, however, when the script is complete it will then play the scenes in the reverse order. It will continue to bounce until stopped.

Change to the **Cycle** option and Play again. Note now that the animation runs until you Pause or Stop. In this mode it runs from start to finish and then resets to the beginning and runs again.

Change to the **Bounce** option and Play again. In this mode it plays from start to finish and then from finish to start and so on.

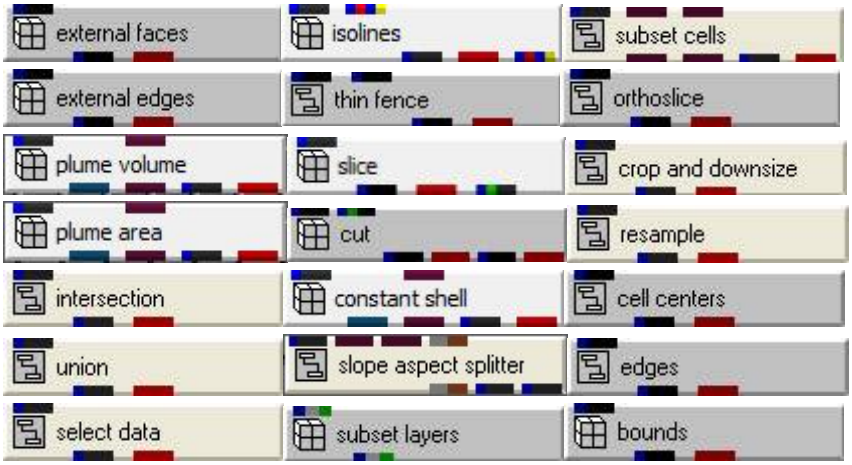
The **Current Frame** slider provides a visual display of the current frame, and more importantly provides a direct method of examining each frame. The slider button (moving square) can be dragged forward or backward to advance or retreat the current frame in real time.

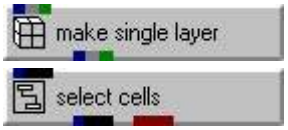
Now let's control the animation using the slider. You should have noticed that it moved when you play with any of the modes above. Press Stop (or Pause) and "click and hold" on the square moving button on the slider. As you move the slider, the animation responds by displaying the frame that you specify by your motion.

You can use the **Delay** slider to slow down the replay. The slider controls the amount of time between frames in seconds. You can set the amount of time from one-hundredth of a second to two seconds.

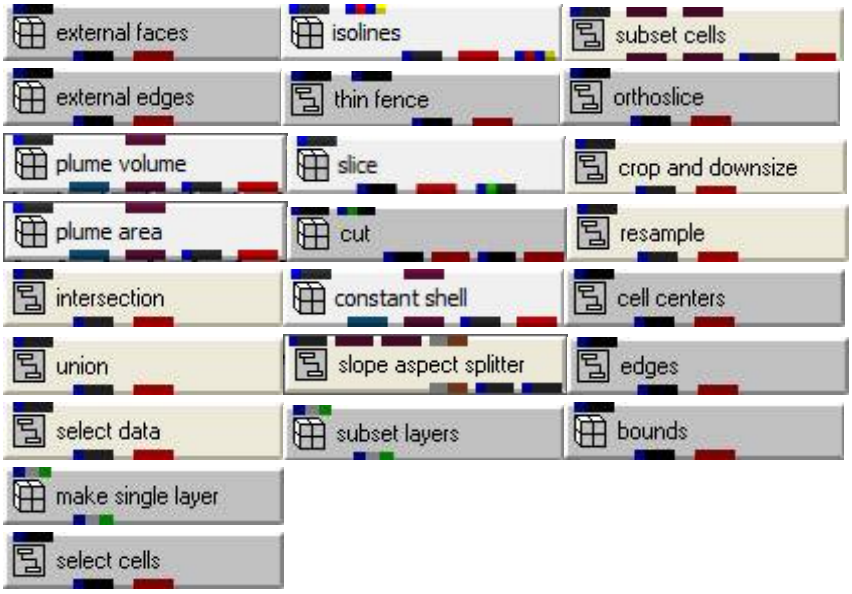
Now that you know how to play the animation, the real fun begins.

Subsetting Modules





Subsetting Modules



external_faces

General Module Function

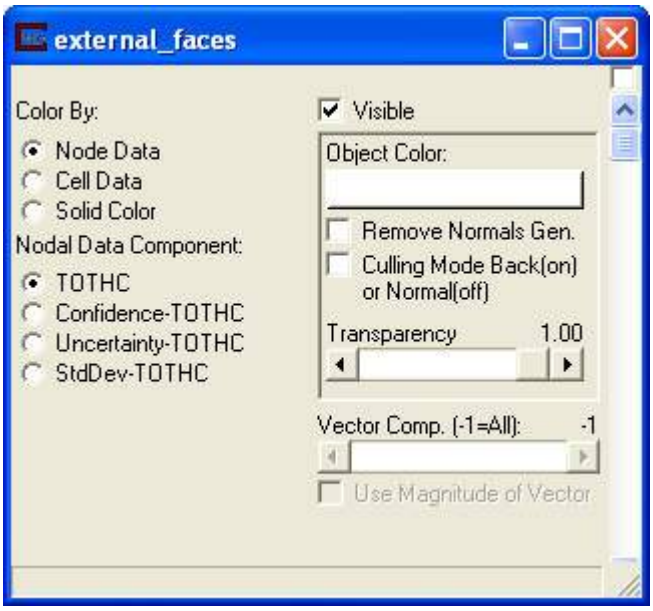
The external_faces module extracts external faces from a 2D or 3D field for rendering. external_faces produces a mesh of only the external faces of each cell set of a data set. Because each cell set's external faces are created there may be faces that are seemingly internal (vs. external). This is especially true when external faces is used subsequent to a plume_volume module on 3D (volumetric) input.

Module Input Ports

external_faces has one input port which can accept data from most other modules. plume_volume and Explode_and_Scale are commonly connected to the input port of external_faces.

Module Output Ports

external_faces has two output ports, the first of which (closest to the left) outputs a data field that can be input to other Subsetting and Processing modules. Data output from this port will contain a new unstructured cell data set representing the geometry of the external faces only. If nodal data was present in the data before entering external_faces, the first component will be preserved in the output field. The second port outputs a renderable geometry directly to the Viewer. This red port will be colored by the selected Data Component (see below).



Module Control Panel

The control panel for external_faces is shown in the figure above.

The ***Color By*** component radio button list contains all of the data components piped into the input port. Only one component can be selected at a time and the default selection is the first (0th) nodal data component. If Solid Color is selected the output field will have no data but the red output port object will be colored by the ***Object Color*** button.

You can also choose to extract a single scalar component of a vector data component.

The component selected affects only renderable geometry output from this module, the data field that is out will contain all of the data components.

The ***Remove Normals Gen***eration toggle controls how vertices and edges are rendered. When this is ON, it makes more distinct edges, but a more faceted overall surface.

The ***Culling Mode*** toggle controls whether back facing surface are visible. Generally you will want this ON when making the object(s) transparent.

The ***Transparency*** slider determines the opacity of the objects.

Related Modules

-> external_edges

external_edges

General Module Function

The external_edges module produces a wireframe representation of of an unstructured cell data mesh. This is generally used to visualize the skeletal shape of the data domain while viewing output from other modules, such as plume_volumes and isosurfaces, inside the unstructured mesh. external_edges produces a mesh of only the external edges which meet the

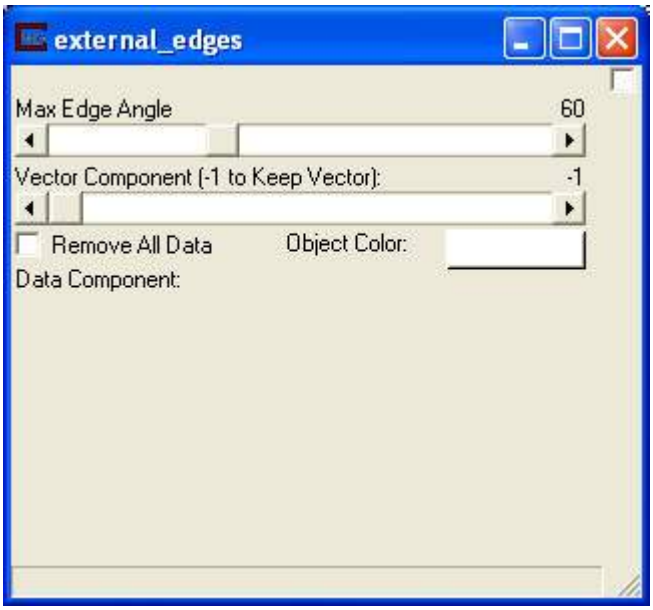
edge angle criteria below for each cell set of a data set. Because each cell set's external faces are used there may be edges that are seemingly internal (vs. external). This is especially true when external edges is used subsequent to a plume_volume module on 3D (volumetric) input.

Module Input Ports

external_edges has one input port, which can accept unstructured mesh data. The mesh data may contain nodal data.

Module Output Ports

external_edges has two output ports, the first of which (closest to the left) outputs a new unstructured mesh which contains line cell types, representing the external edges, and nodal data corresponding to the nodal data in the input unstructured data. The second port outputs a renderable geometry directly to the Viewer. This red port will be colored by the selected Data Component (see below).



Module Control Panel

The control panel for external_edges is shown in the figure above.

The ***Max Edge Angle*** available range is 0 to 180 degrees and the default is 25 degrees. Setting Max Edge Angle to 0 causes all external cell (element) edges in the model to be visible. This angle is defined as the angle between two faces or cells in the model. Intersections between faces or cells with angles greater than the Max Edge Angle are visible. Exterior model edges are always visible.

The ***Data Component*** radio button list contains all of the data components piped into the input port. Only one component can be selected at a time and the default selection is the first (0th) data component. This component selection is used only to select which data to display from this module. Any modules downstream of external_edges will receive all of the nodal data components. If the ***Remove All Data*** toggle is selected the resulting edges will be colored according to the ***Object Color***.

You can also choose to extract a single scalar component of a vector data component, with the ***Vector Component Slider***.

Related Modules

external_faces

plume_volume

General Module Function

The plume_volume module creates a true 3D volumetric subset of a 3D input. The resulting volume can be used as input for further subsetting or volumetric computations since it is volumetric. This module should not normally be used when you desire a visualization of a plume but rather when you wish to do subsequent operations such as analysis, slices, etc.

Module Input Ports

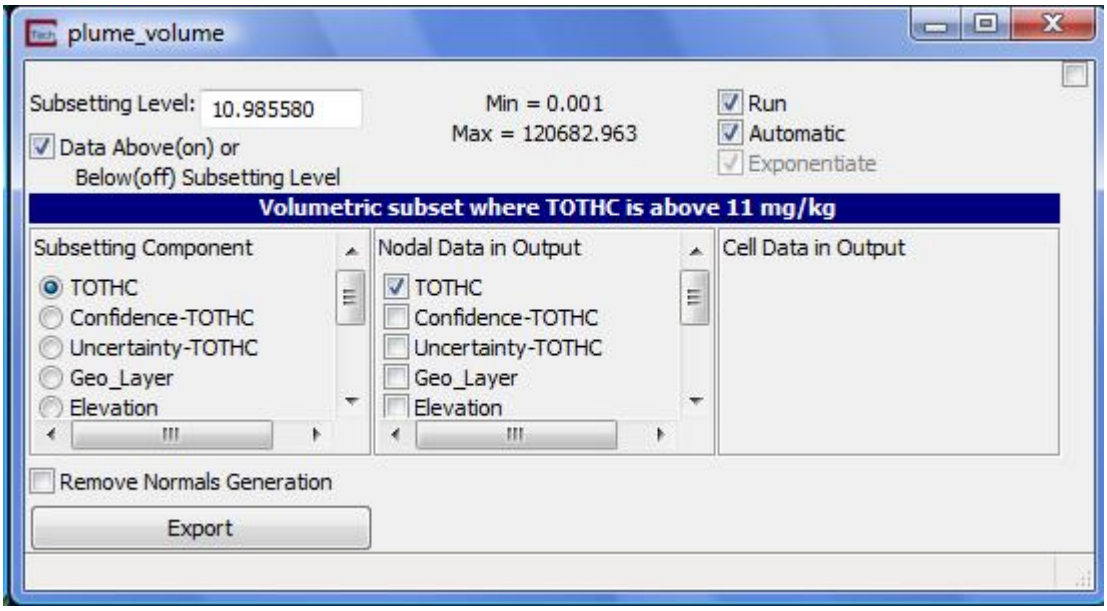
plume_volume has two input ports.

1. The leftmost (blue-black) port accepts unstructured mesh data.
2. The second (maroon) port provides a means to share the subsetting level with other modules.

Module Output Ports

plume_volume has four output ports.

1. The first output port (leftmost) provides an "info" string which can be connected to the Titles module for a quick simple title associated with the settings of this module. For the settings shown below, the Title displayed would be: "TOTHC above 11 mg/kg". This is constructed using the "Subsetting Component" name followed by "above" or "below", a rounded representation of the subsetting level and the units (if specified in the input).
2. The second (maroon) port provides a means to share the subsetting level with other modules.
3. The third (blue-black) port outputs a new unstructured mesh which contains the true 3D volumetric subset of the 3D input. Nodal data sent to this output port consists of the nodal data in output.
4. The fourth (red) port outputs a rendered geometry directly to the Viewer. However, normally plume_volume would not be connected to the Viewer.



Module Control Panel

The control panel for plume_volume is shown in the figure above.

The ***SubsettingLevel*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is used. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default level is the arithmetic average of the minimum and maximum values in the subsetting component. If your input data has been kriged with log processing, the values here will be exponentiated already (provided that the Automatic toggle is on). You will not enter the Log of your desired subsetting value. Also, when the data is log processed, the default value is the geometric mean (vs. arithmetic mean) of your data extremes.

The ***Run*** toggle when not ON will prevent the module from running.

The ***Automatic*** toggle when ON determines "automatically" if the subsetting level values will be exponentiated for you vs. having to input value which are the LOG of your desired level. When this is on the Exponentiated toggle is inactive.

The ***Exponentiated*** toggle is inactive if the Automatic toggle is ON. Otherwise, it determines whether to real units to logarithmic units..

The ***Data Above (on) or Below (off) Subsetting Level*** check box (Above Box) is used to display data above the subsetting level or below the subsetting level. For example, to display a volumetric subset of all concentrations greater than or equal to 1 ppm, set subsetting level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the plume_volume of 1 ppm and below, simply turn the Above Box off (no check in box).

Subsetting Component refers to the nodal data component used to create the subset of the original input field. When an component is selected, the min

and max values of the variable are displayed in the upper center of the dialog box. The default component is the first (0th) component in the column.

Nodal Data in Output determines which nodal data components will be included in the output. The first one in the list will determine the data used for coloring in the (red) output to the Viewer. For example, choosing subsetting component concentration and uncertainty for the nodal data in the output will create a volumetric SHELL subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The ***CellData in Output*** option box selector lets you map cell data (if any) to the surfaces output by plume_volume. The cell data will only be visible if all nodal data is unselected (under *nodal data in output*).

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Culling Mode Back (on) or Normal (off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

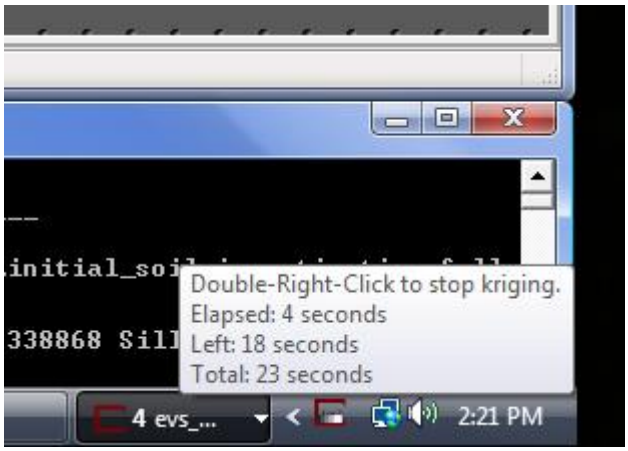
The ***Transparency*** slider changes the transparency of the output to the red port to the Viewer.

The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. Once this button is pushed, plume_volume will have has seven output ports.

1. The info string
2. The subsetting level
3. The string representing the selected subsetting component
4. The string representing "above" or "below"
5. The units as a string
6. The (blue-black) output
7. The (red) output to the Viewer.

Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Related Modules

-> plume_shell

-> plume_area

plume_area

General Module Function

The plume_area module creates a 2D subset of a 2D input. In other words, the input should be a surface and the output will be a subset of that surface. The resulting output can be used as input for further subsetting or output to the Viewer.

Module Input Ports

plume_area has two input ports.

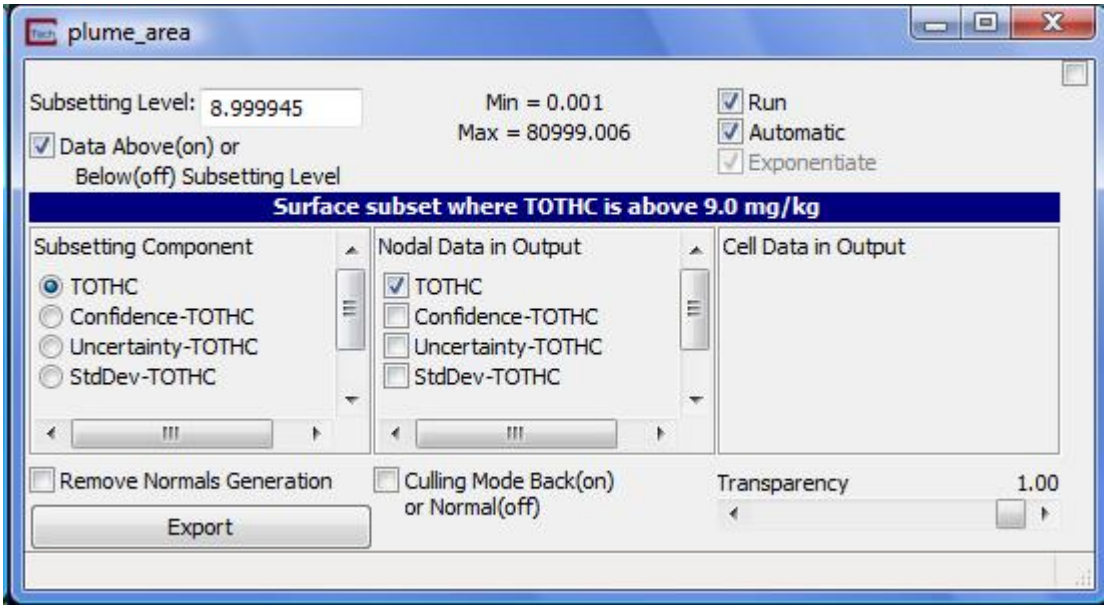
1. The leftmost (blue-black) port accepts unstructured mesh data which should represent a surface or surfaces..
2. The second (maroon) port provides a means to share the subsetting level with other modules.

Module Output Ports

plume_area has four output ports.

1. The first output port (leftmost) provides an "info" string which can be connected to the Titles module for a quick simple title associated with the settings of this module. For the settings shown below, the Title displayed would be: "TOTHC above 11 mg/kg". This is constructed using the "Subsetting Component" name followed by "above" or "below", a rounded representation of the subsetting level and the units (if specified in the input).
2. The second (maroon) port provides a means to share the subsetting level with other modules.

3. The third (blue-black) port outputs a new surface which contains the subset of the surface input. Nodal data sent to this output port consists of the nodal data in output.
4. The fourth (red) port outputs a rendered geometry directly to the Viewer. However, normally plume_area would not be connected to the Viewer.



Module Control Panel

The control panel for plume_area is shown in the figure above.

The ***SubsettingLevel*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is used. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default level is the arithmetic average of the minimum and maximum values in the subsetting component. If your input data has been kriged with log processing, the values here will be exponentiated already (provided that the Automatic toggle is on). You will not enter the Log of your desired subsetting value. Also, when the data is log processed, the default value is the geometric mean (vs. arithmetic mean) of your data extremes.

The ***Run*** toggle when not ON will prevent the module from running.

The ***Automatic*** toggle when ON determines "automatically" if the subsetting level values will be exponentiated for you vs. having to input value which are the LOG of your desired level. When this is on the Exponentiated toggle is inactive.

The ***Exponentiated*** toggle is inactive if the Automatic toggle is ON. Otherwise, it determines whether to real units to logarithmic units..

The ***Data Above (on) or Below (off) Subsetting Level*** check box (Above Box) is used to display data above the subsetting level or below the subsetting level. For example, to display a volumetric subset of all

concentrations greater than or equal to 1 ppm, set subsetting level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the plume_area of 1 ppm and below, simply turn the Above Box off (no check in box).

Subsetting Component refers to the nodal data component used to create the subset of the original input field. When an component is selected, the min and max values of the variable are displayed in the upper center of the dialog box. The default component is the first (0th) component in the column.

Nodal Data in Output determines which nodal data components will be included in the output. The first one in the list will determine the data used for coloring in the (red) output to the Viewer. For example, choosing subsetting component concentration and uncertainty for the nodal data in the output will create a volumetric SHELL subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The ***CellData in Output*** option box selector lets you map cell data (if any) to the surfaces output by plume_area. The cell data will only be visible if all nodal data is unselected (under *nodal data in output*).

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Culling Mode Back (on) or Normal (off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

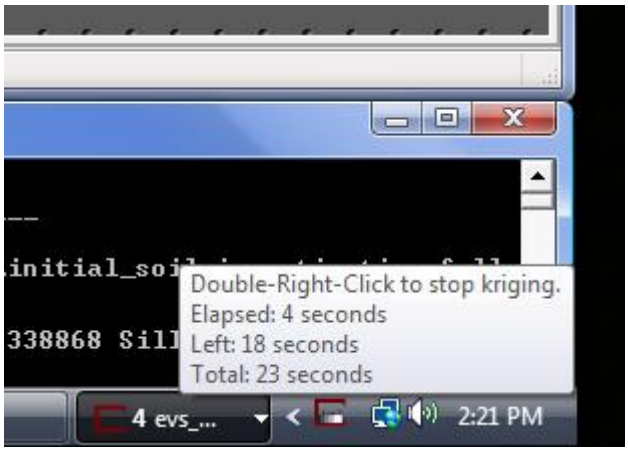
The ***Transparency*** slider changes the transparency of the output to the red port to the Viewer.

The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. Once this button is pushed, plume_area will have has seven output ports.

1. The info string
2. The subsetting level
3. The string representing the selected subsetting component
4. The string representing "above" or "below"
5. The units as a string
6. The (blue-black) output
7. The (red) output to the Viewer.

Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Related Modules

-> plume_shell

-> plume_volume

intersection

(This module is available only in MVS)

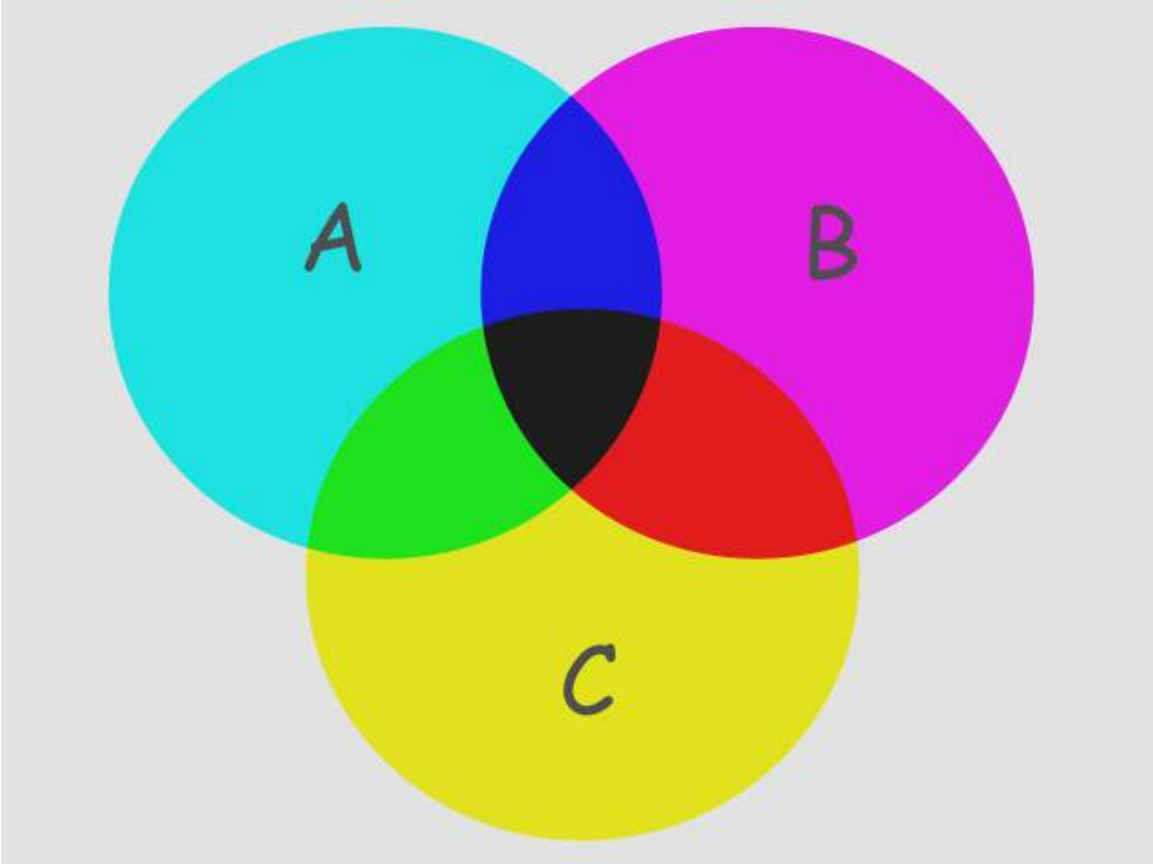
General Module Function

intersection is a powerful module that incorporates some of the characteristics of plume_volume, yet allows for any number of volumetric sequential (serial) subsetting operations.

The functionality of the intersection module can be obtained by creating a network of serial plume_volume modules. The number of analytes in the intersection is equal to the number of plume_volume modules required.

The intersection of multiple analytes and threshold levels can be equated to the answer to the following question (example assumes three analytes A, B & C with respective subsetting levels of a, b and c):

"What is the volume within my model where A is above a, **AND** B is above b, **AND** C is above c?"



The figure above is a Boolean representation of 3 analyte plumes (A, B & C). The intersection of all three is the black center portion of the figure. Think of the image boundaries as the complete extents of your models (grid). The "A" plume is the circle colored cyan and includes the green, black and blue areas. The intersection of just A & C would be both the green and black portions.

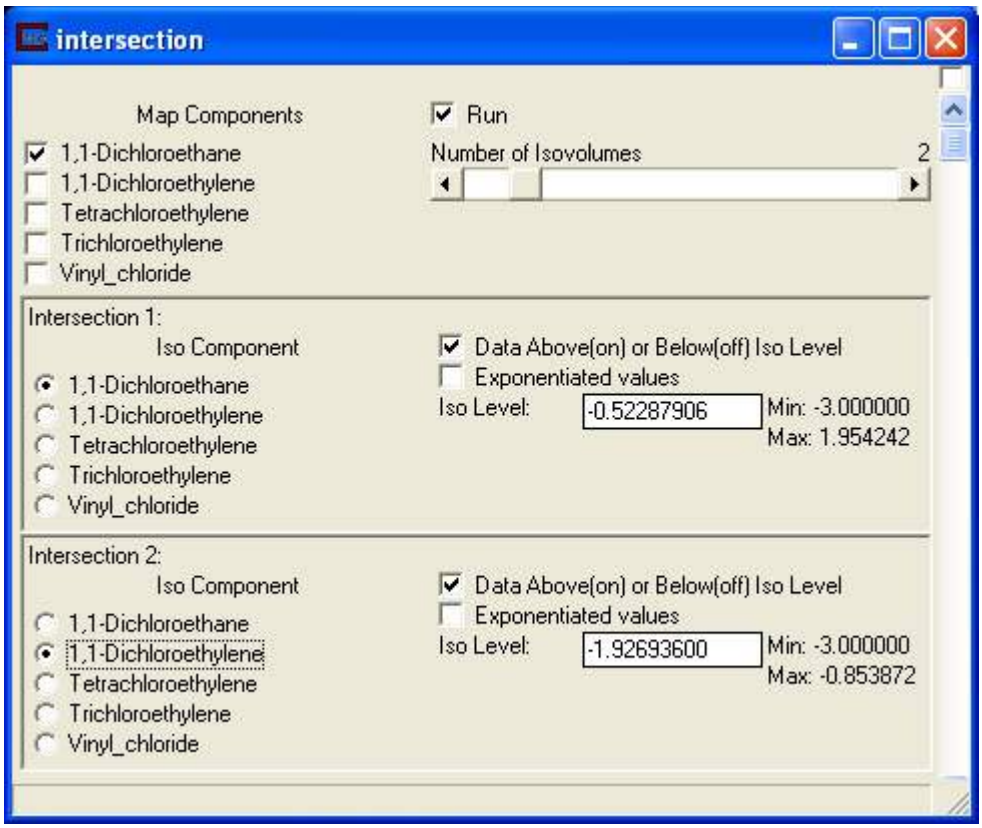
Module Input Ports

intersection has one input port that accepts unstructured mesh data.

Module Output Ports

intersection has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains a merged field containing all of the exterior faces of the plume resulting from the sequential subsetting operations.

The second port outputs a rendered geometry directly to the Viewer.



Module Control Panel

The control panel for intersection is shown in the figure above. intersection's user interfaces changes automatically depending on the selected ***Analytes in Intersection*** to be performed. The **Top Panel** allows you to select the following parameters:

Turning off the ***Run*** toggle allows you to make multiple changes without the module running automatically.

Map Components determines which model data components will be sent to the output ports. The first map component selected will be used to color the output. For example, choosing iso component *concentration* and map component *uncertainty* will create an intersection of concentration colored by uncertainty.

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The remaining panels will be named ***Analyte1***:, ***Analyte2***: and so on. Each of these is identical in their options having the following parameters:

Iso Component refers to the model data component used to create the subset. When an iso component is selected, the min and max values of the variable are displayed in the right side of the panel.

The ***Data Above(on) or Below(off) subsetting level*** check box (Above Box) is used to display data above the subsetting level or below the subsetting level. For example, to display a plume_volume of all

concentrations greater than or equal to 1 ppm, set subsetting level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the intersection of 1 ppm and below, simply turn the Above Box off (no check in box).

The ***Exponentiated Values*** toggle makes another type-in field visible which convert real units to logarithmic units for you automatically.

The ***subsetting level*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default subsetting level value is the arithmetic average of the minimum and maximum values in the iso component. If your input data has been kriged with log processing, the values here will be the Log of your input data.

union

(This module is available only in MVS)

General Module Function

union is a powerful module that allows for a large number of complex serial and parallel subsetting operations required to compute and visualize the union of multiple analytes and threshold levels. The functionality of the union module can be obtained by creating a network fragment composed of only plume_volume modules. However as the number of analytes in the union increases, the number of plume_volume modules increases very dramatically. The table below lists the number of plume_volume modules required for several cases:

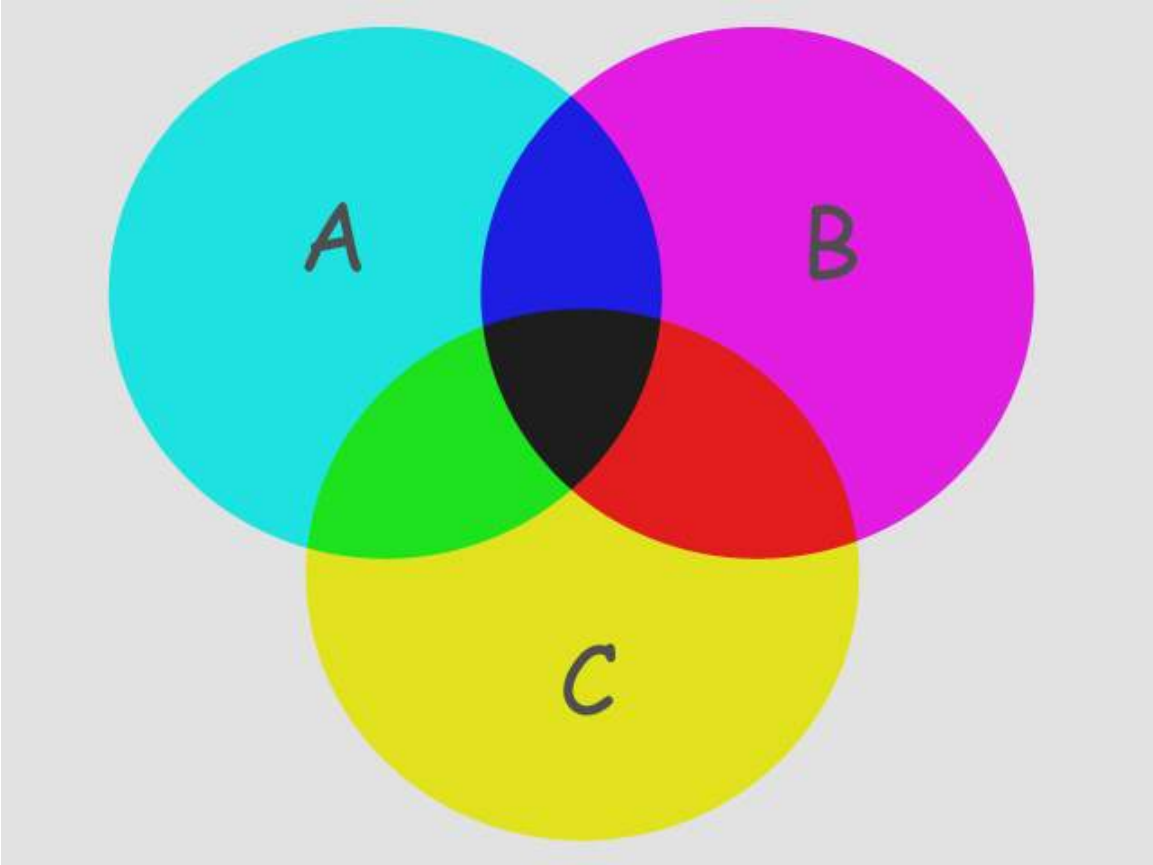
Number of Analytes	Number of plume_volume Modules
2	3
3	6
4	10
5	15
6	21
7	28
n	(n * (n+1)) / 2

From the above table, it should be evident that as the number of analytes in the union increases, the computation time will increase dramatically. Even

though union appears to be a single module, internally it grows more complex as the number of analytes increases.

The union of multiple analytes and threshold levels can be equated to the answer to the following question (example assumes three analytes A, B & C with respective subsetting levels of a, b and c):

"What is the volume within my model where A is above a, **OR** B is above b, **OR** C is above c?"



The figure above is a Boolean representation of 3 analyte plumes (A, B & C). The union of all three is the entire colored portion of the figure. Think of the image boundaries as the complete extents of your models (grid). The "A" plume is the circle colored cyan and includes the green, black and blue areas. The union of just A & C would be all colored regions EXCEPT the magenta portion of B.

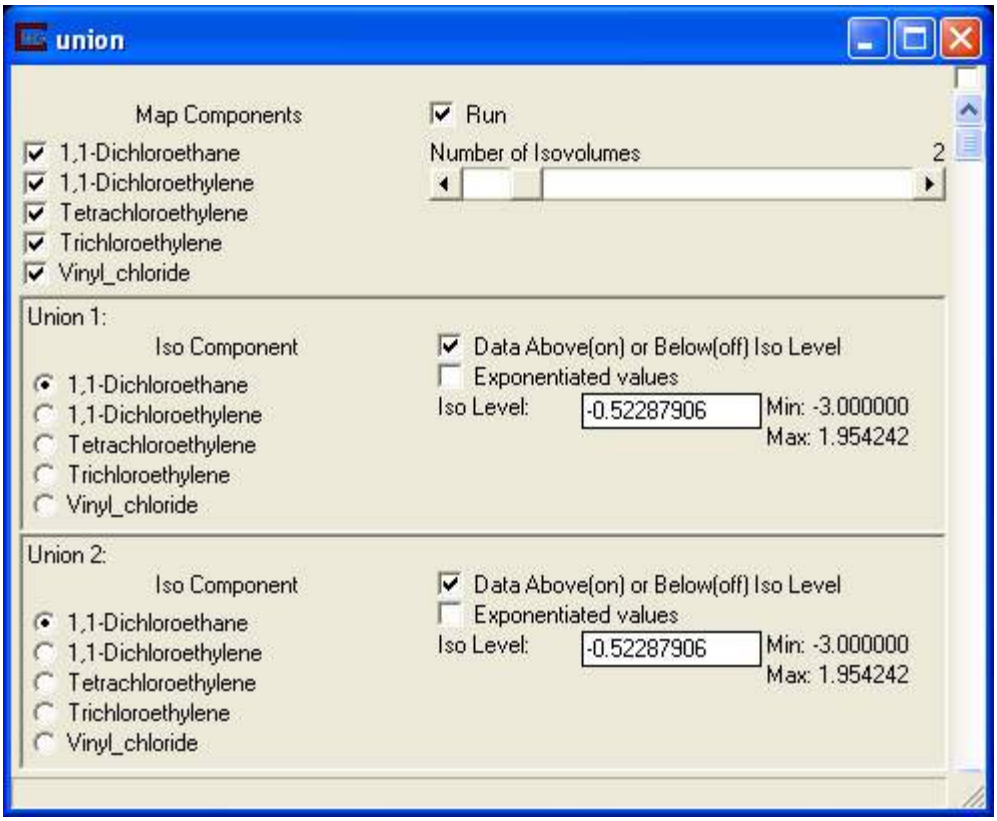
Module Input Ports

union has one input port that accepts unstructured mesh data.

Module Output Ports

union has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains a merged field containing the union of all analytes with the selected data components available for coloring or subsequent subsetting operations.

The second port outputs a rendered geometry directly to the Viewer.



Module Control Panel

The control panel for union is shown in the figure above. union's user interfaces changes automatically depending on the selected ***Analytes in Union*** to be performed. The **Top Panel** allows you to select the following parameters:

Turning off the ***Run*** toggle allows you to make multiple changes without the module running automatically.

Map Components determines which model data components will be sent to the output ports. The first map component selected will be used to color the output. For example, choosing iso component *concentration* and map component *uncertainty* will create a volumetric subset of concentration colored by uncertainty.

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The remaining panels will be named ***Analyte*1**:, ***Analyte*2**: and so on. Each of these is identical in their options having the following parameters:

Iso Component refers to the model data component used to create the subset. When an iso component is selected, the min and max values of the variable are displayed in the right side of the panel.

The ***Data Above(on) or Below(off) subsetting level*** check box (Above Box) is used to display data above the subsetting level or below the subsetting level. For example, to display a volumetric subset of all

concentrations greater than or equal to 1 ppm, set subsetting level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the union of 1 ppm and below, simply turn the Above Box off (no check in box).

The ***Exponentiated Values*** toggle makes another type-in field visible which convert real units to logarithmic units for you automatically.

The ***subsetting level*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default subsetting level value is the arithmetic average of the minimum and maximum values in the iso component. If your input data has been kriged with log processing, the values here will be the Log of your input data.

select_data

General Module Function

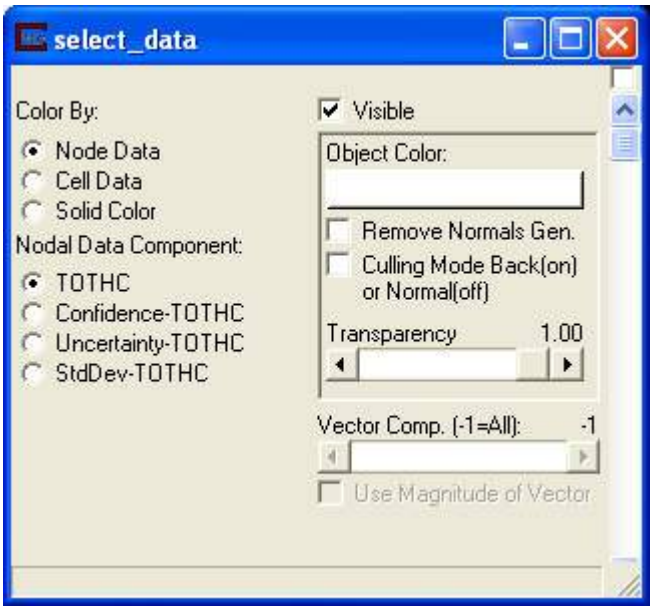
The select_data module extracts a single data component from a field. Select_data can extract scalar data components or vector components. Scalar components will be output as scalar components and vector components will be output as vector components.

Module Input Ports

Select_data has only one input port which accepts any nodal data (multiple scalar and/or vector).

Module Output Ports

Select_data has two output ports. The first port (closest to the left) outputs a new nodal data component containing only the extracted nodal data component. If mesh data was included in the input, it will be contained in the output. The second port will output a renderable object if the data are faces or lines.



Module Control Panel

The control panel for select_data is shown in the figure above.

The Color By radio button controls which type of data to export.

1)The Node Data component list contains all of the nodal data components piped into the input port. You can also choose to extract a single scalar component of a vector data component.

2) The Cell Data component list contains all of the cell data components piped into the input port. Only one component can be selected at a time and the default selection is the first (0th) data component. Any modules downstream of select_data will receive only the selected data component.

3) If Solid Color is selected the output field will have neither cell or nodal data but will be colored according to the Object Color button on the right panel.

The right side panel provides control over several commonly edited object properties including transparency, Normals generation, culling mode and visibility.

Related Modules

->extract_scalar

-> combine_comp

-> combine_vect

select_cells

General Module Function

select_cells provides the ability to select individual geologic layers for output. If connected to Explode_and_Scale multiple select_cells modules will allow selection of specific geologic layers for downstream processing. One example

would be to texture map the top layer with an aerial photo after one select_cells and to color the other layers by geologic layer with a parallel select_cells path. This can be accomplished by multiple Explode_and_Scale modules, but that would take much more memory.

Module Input Ports

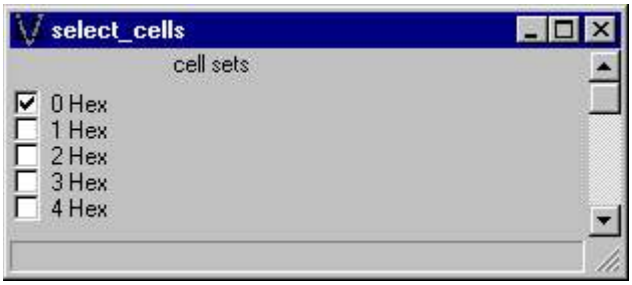
select_cells has only one input port which is an EVS field. Typically, this should be from Explode_and_Scale.

Module Output Ports

select_cells has two output ports. The first output port (closest to the left) outputs a subset of the input field based on which cell_sets (geologic layers) are chosen. The second port outputs a renderable geometry directly to the Viewer, but only if the input is 2D which is not typical.

Module Control Panel

The control panel of select_cells is shown in the figure below. The check boxes are used to select which geologic layers will be passed through the module.



isolines

General Module Function

The isolines module is used to produce lines of constant (iso)value which can be offset from a 2D surface (such as a slice plane), or the external faces of a 3D surface, such as the external faces of a plume_volume. The input data for isolines must be a 2D surface or faces, it cannot be a volumetric data field. Isolines will preserve any transform applied to the data upstream of the isolines module. If the input is the faces of a 3D surface, then the isolines will actually be 3D in nature. Isolines can automatically place labels (This feature available only in EVS PRO and MVS) in the 2D or 3D isolines.

Module Input Ports

Isolines has two input ports.

1. The first (left) port accepts unstructured mesh data and nodal data. The data piped into this port **must** be surfaces and contain nodal data. At least one component of the nodal must be scalar. If used with 3D data, a slice plane or other surface mapper such as external faces must be used before the data is sent to the isolines input port.

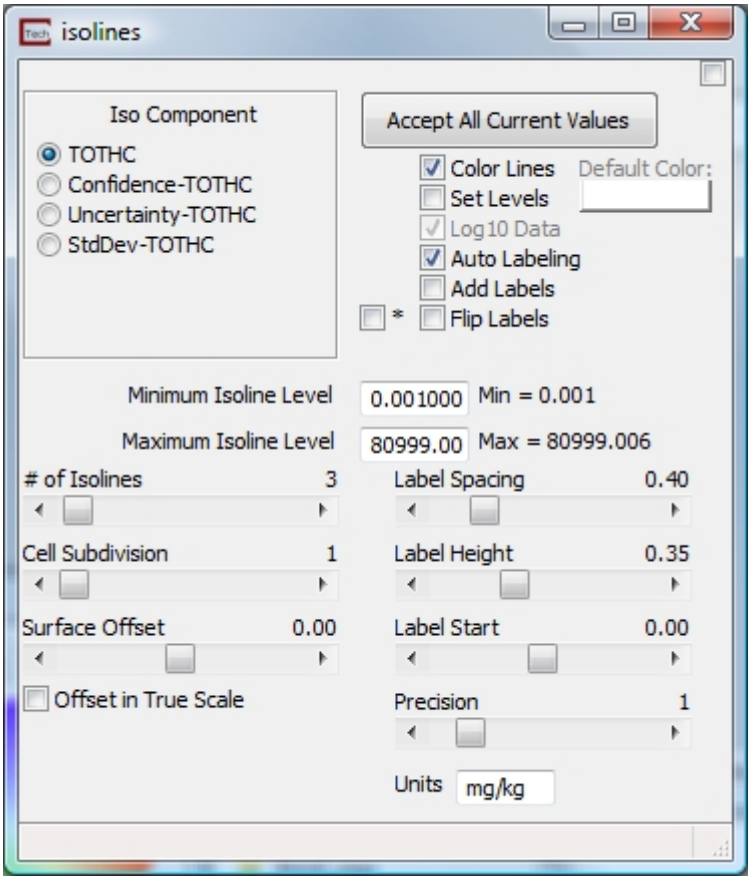
2. The second port (blue-red-blue-beige) is found on only a few modules in EVS/MVS. The modules with this port are contour_data, isolines, Datamap_Editor, and Legend.

These modules pass data corresponding to one or more subsetting levels (or solid contour levels). When this data is passed to isolines, it is used to set the precise break points where contour lines will occur. When this data is passed from isolines to other modules, the levels set in isolines will be reflected in those modules.

Module Output Ports

Isolines contains three output ports.

1. The first output port (closest to the left) outputs a new unstructured mesh which contains line cell types representing the isolines. Nodal data sent to this output port consists of the isoline level value.
2. The second port outputs a renderable geometry, which can be connected directly to the Viewer.
3. The third port (blue-red-blue-beige) is found on only a few modules in EVS/MVS. The modules with this port are contour_data, isolines, Datamap_Editor, and Legend.



Module Control Panel

The control panel for isolines is shown in the figure above. An Iso Component radio button will appear for every iso component piped into the input port. The Iso component refers to the nodal data component that will be used to create isolines. When a nodal data component variable is selected, the min (Min =) and max (Max =) values of the variable are displayed below the minimum and maximum subsetting level edit fields. The default iso component is the first (0th) component in the column.

The minimum and maximum isoline level input fields allow the user to constrain the isoline data range. The default values entered into the edit fields are the minimum and maximum values of the data component selected. If a value greater than the Max= value is entered into the maximum edit field, the Max = value will be automatically placed in the field. The converse is true for the minimum edit field. If the minimum value is greater than the maximum value, an isoline at the maximum level is the only isoline that will be displayed.

The *Auto Labeling* check box will automatically update the units label depending on the type of data passed into the module(linear, or log).

The *color lines* check box determines whether the isolines will be colored according to isoline values, or whether they are monochromatic and uncolored. The log10 data check box tells isolines if the data piped into the input port is log10 transformed or not. Further discussion of the use of these buttons is provided below.

The *minimum and maximum subsetting level* edit fields allow the user to subset the data used to create isolines. The default values entered into the edit fields are the minimum and maximum of the data component selected. If a value greater than the "Max =" value is entered into the maximum edit field, the "Max =" value will be placed in the field. The converse is true for the minimum edit field. If the minimum value is greater than the maximum value, an isoline at the maximum level is the only isoline that will be displayed.

The *Set Levels* toggle is a new functionality allowing setting of any number of isolines at any specified levels. For example, 5 contour lines between 0 and 100 would *NOT* have to be 0, 20,40, 60,etc, but instead could be 0.5, 5, 10, 25, 50. The subsetting level is set by a user type-in value and the desired number of contour levels is set with a slider.

The *number of isolines* slider is interpreted two different ways depending on whether or not the log10 check box is selected. If the log10 check box is selected, the number of isolines refers to the number of isolines per decade. The logarithmic option allows for non-integer data limits. The default number of lines is three and the maximum (for log option) is three. Isolines are placed on specific intervals within each decade. It will create isolines at suitable intervals dependent on number of isolines selected.

1 will give even decades.

2 will give lines at 1, 3, 10, 30, etc.

3 will give lines at 1, 2, 5, 10, 20, 50, etc.

If the log10 check box is not selected, then the number of isolines refers to the total number of isolines to be drawn on the surface. The default is three and the maximum allowable is 100. isolines are placed at equal intervals based on (max-min)/number of isolines.

Isolines has the ability to place labels on each isoline level on two-dimensional or three-dimensional isolines. Lines and labels can be colored or uncolored. Size, spacing, orientation and phasing of labels is user controllable.

The *Add_Labels* toggle specifies whether labels are added. Label_Spacing, Label_Height, and Label_Start determine the frequency of labels, their size and the starting position of labels within an isoline. All of these parameters are in units which are relative to the model extent. This allows the defaults to give reasonable starting results in most cases.

The *Flip_labels* toggle will flip all labels upside down. Depending on the viewpoint and the data, this may be useful for making the labels more readable.

The toggle before it "*" flips the isolines (and labels) to the other side of the surface. This is useful for putting isolines on the slice module or to put labeled lines on both sides (with two isolines modules) of a surface or fence diagram.

The *smoothing level* slider controls the number of element subdivisions to be performed before drawing the isolines. The higher the number, the more accurately the isolines will be placed through the data. Higher accuracy does not always cause the isolines to become smoother, depending on the data distribution. The default smoothing levels is one, and the maximum is eight.

The *surface offset* controls where the isolines will be placed in reference to the surface. A positive number places the isolines on the exterior of 3D surfaces, but below slice planes (and therefore the lines are only visible when looking from below). Conversely, a negative number places the isolines above slice planes which makes them visible when viewed from above. The numerical value of the offset distance determines how far above or below the reference surface to place the isolines.

The *Offset in True Scale* option (Available only in MVS) offsets in the model coordinates.

The *color lines* check box is used to control the display of the isolines. If the box is checked, then the isolines will be colored based on their subsetting level value. If the box is not selected, then the lines will be uncolored, and can be colored with a single color using the properties editor in the Viewer.

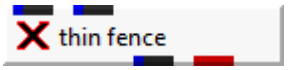
The *log10 data* check box is used to tell isolines whether or not the input data is log 10 transformed. If the data is log10 transformed and this box is selected, then up to three isolines per decade can be displayed (as described above). The color legend will reflect the untransformed values in the display. If the log10 data check box is not selected, and the data are log 10 transformed, then isolines thinks the data is not transformed and places the selected number of isolines (up to 99) between the minimum and maximum log 10 subsetting levels selected.

Related Modules

->solid_contour_set

-> isosurface

thin_fence



General Module Function

Thin_fence creates a fence diagram along a user defined (x, y) path. The fence cross-section has no thickness (because it is composed of areal elements such as triangles and quadrilaterals), but it is truly 3D. It receives a 3D field (with volumetric elements) into its left input port and it receives polylines (from click_sketch, Read_DXF, isolines, Read_UCD, or other sources) into its right input port. Its function is similar to fence_cut, however it actually creates a new grid and does not rely on any other modules (like plume_volume or plume_shell) to do the "cutting". Only the x and y coordinates of the input polylines are used because thin_fence cuts a projected slice that is z invariant. Thin_fence recalculates when either input field is changed (and Run Automatically is on) or when the "Accept" button is pressed.

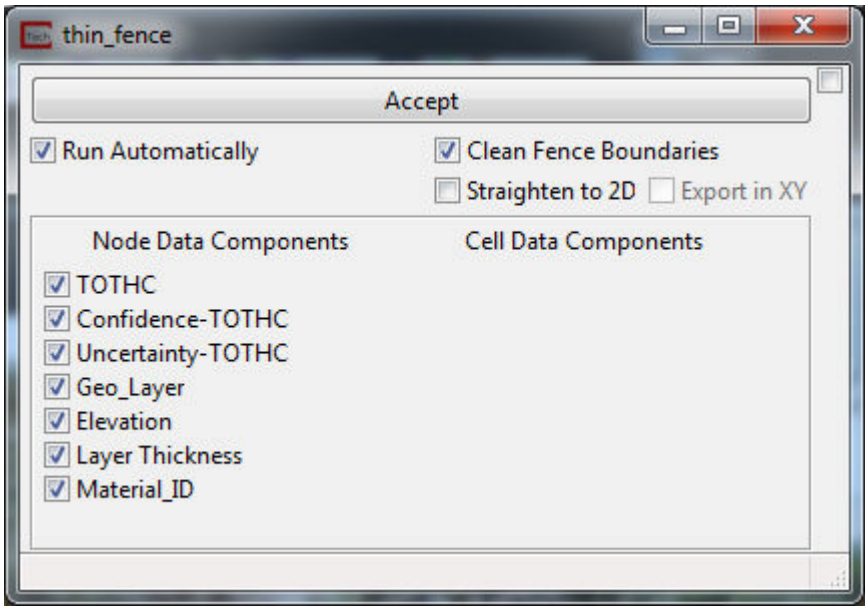
Fences produced with this module are planar objects (e.g. they have no volume). Unlike fence_cut, there is no limit to the size or detail of a thin_fence.

Module Input Ports

thin_fence has two input ports. Thin_fence receives any 3D field into its left input port that should be connected to or after Explode_and_Scale. It receives polylines (from Read_DXF, isolines, Read_UCD, or other sources) into its right input port.

Module Output Ports

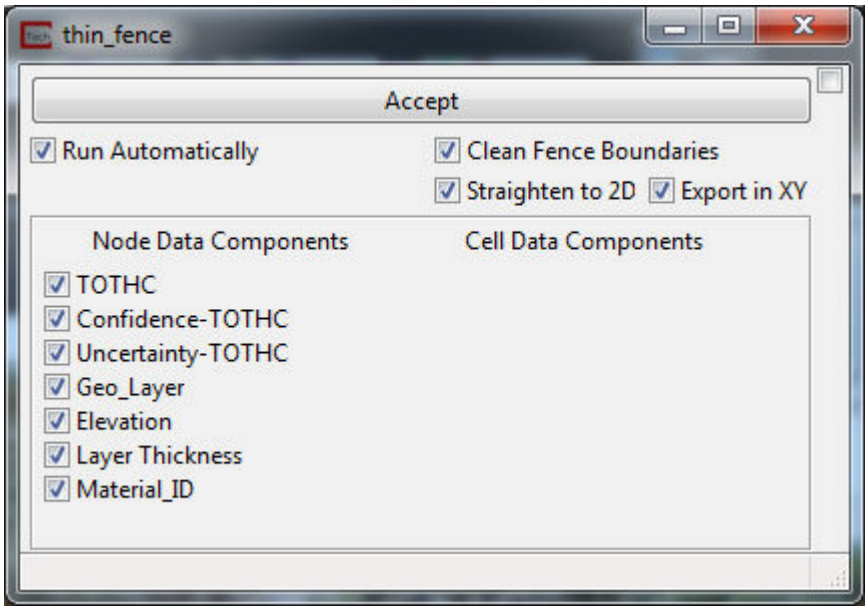
thin_fence has two output ports. The blue-black port outputs a field with the data components of the 3D input mapped to the fence. You can choose some or all of the data components. The red port is a renderable version of the other.



Module Control Panel

The module's control panel is shown above.

- The ***Accept*** button causes thin_fence to generate a fence cross-section if **Run Automatically** is not on.
- The ***Straighten to 2D****toggle* is used to project your 3D fence diagram to a 2D cross section whose X coordinate is the distance along the fence and the Y coordinate is always ZERO (0.0).
    - Please note that the coordinate extents of a straightened fence will be to the right of the origin (0,0). It is best to create 2D projections in a separate Viewer from any other objects in your application.
- The **Export in XY** toggle projects the data onto the XY plane vs. the XZ plane. This is the more useful option if the results are to be used in ESRI's ArcMAP.
- The **Clean Fence Boundaries** toggle (on by default) attempts to merge all cells between each fence segment (created by each line segment of your 2D input). Occasionally the output will look better with it off, but usually it is better left on.
- The **Node Data Components** toggles allow you to select which nodal data to include in the fence.
- The **Cell Data Components** toggles allow you to select which cell data to include in the fence.



slice

The slice module allows you to create a subset of your input which is of reduced dimensionality. This means that volumetric, surface and line inputs will result in surface, line and point outputs respectively. This is unlike cut which preserves dimensionality.

General Module Function

The slice module is used to slice through an input field using a slicing plane defined by one of four methods which are discussed below.

The slicing plane essentially slices through the data field showing the intersection of the slice plane with your input object.

Module Input Ports

slice has two input ports.

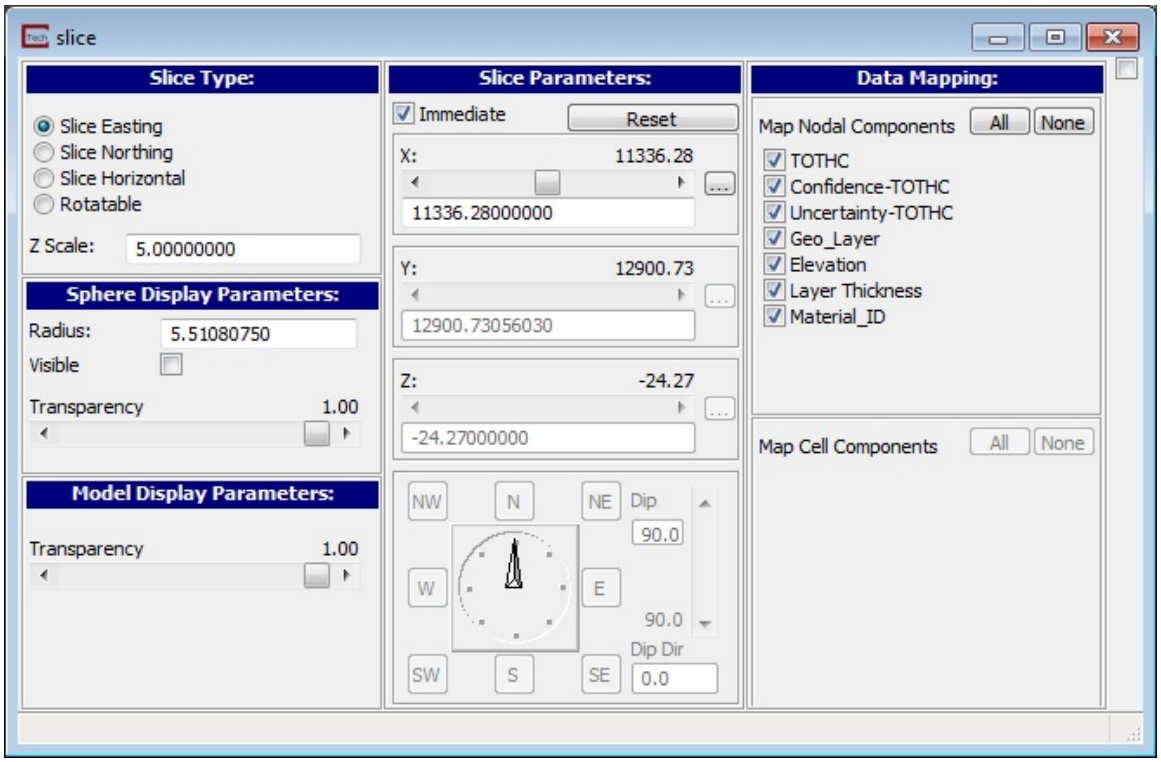
1. The leftmost port accepts any 1D, 2D or 3D mesh. The mesh can contain nodal data and/or cell data.
2. The right port is the Z-Scale and is required if Explode_and_Scale is upstream, in order to reference the correct z coordinates when options 3 or 4 are selected.

Module Output Ports

slice has three output ports.

1. The leftmost is the Z-Scale

2. The second (blue/black) outputs the slice. Data consists of nodal and/or cell data for all map components selected.
3. The red output port is a renderable version (for connection to the Viewer) of the second port.



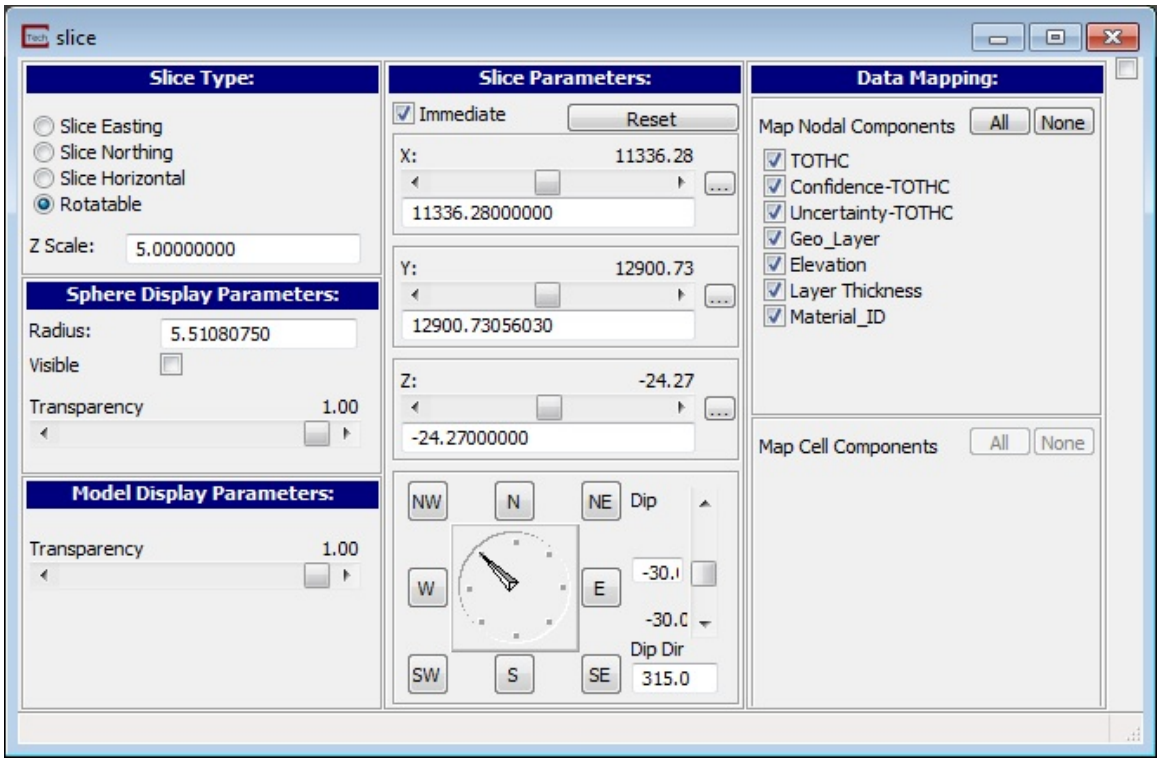
Module Control Panel

The control panel for slice is shown in the figure above. The above example shows "Slice Easting" as the selected slice type. For any of the first three slice types, the plane is defined by a single coordinate input (X, Y, or Z).

- The ***Slice Type*** is how the slicing plane is specified. It can be one of four methods:
    1. A vertical plane defined by an X or **Easting** coordinate
    2. A vertical plane defined by a Y or **Northing** coordinate
    3. A **Horizontal** plane defined by a Z coordinate
    4. An arbitrarily positioned **Rotatable** plane which requires:
        1. A 3D point through which the slicing plane passes. This point can be displayed using the *Reference Sphere*whose size, visibility and transparency can be controlled. Please note that the same slicing result can be achieved with an infinite number of 3D points, all of which would be on the same slicing plane.
        2. A ***Dip*** direction

3. A ***Strike*** direction

- ***Z Scale*** is required if Explode_and_Scale is upstream, in order to reference the correct z coordinates when options 3 or 4 are selected.
- ***Reference Sphere:*** The reference sphere is a pickable object which you can move using your mouse. To pick it, hold down the Alt key and click on the sphere with your left mouse button. Moving the sphere with the right mouse will move the resulting slice output.
  - Sphere ***Visibility*** is a toggle to turn on/off the sphere
  - Sphere ***Transparency*** is a slider to control the opacity of the sphere
  - Sphere ***Radius*** is a type-in to control the size of the sphere. The default value is computed for you based on the x-y extents of the model.
- The ***Slice Parameters*** are the coordinates and/or Dip/Strike angles required to specify the position and orientation of the slicing plane. These include:
  - X coordinate (used with Slice Easting and Rotatable)
  - Y coordinate (used with Slice Northing and Rotatable)
  - Z coordinate (used with Slice Horizontal and Rotatable)
  - Dip angle (used only with Rotatable)
  - Dip Direction (used only with Rotatable)
- ***Map Nodal Components*** determines which nodal components will be sent to the output ports. Also, the first map component selected will be the variable used to color the output.
- The ***Map Cell Components*** option box selector lets you map cell data (if any) to the output. The cell data will only be visible if all nodal data components are unselected (under *Map Nodal Components*).



Related Modules

-> cut

cut

The cut module allows you to create a subset of your input which is of the same dimensionality. This means that volumetric, surface, line and point inputs will have subsetted outputs of the same object type. This is unlike slice which decreases dimensionality.

General Module Function

The cut module is used to cut away part of the input field using a cutting plane defined by one of four methods which are discussed below.

The cutting plane essentially cuts the data field into two parts and sends only the part above or below the plane to the output ports (above and below are terms which are defined by the normal vector of the cutting plane). The output of cut is the subset of the model from the side of the cut plane specified.

Module Input Ports

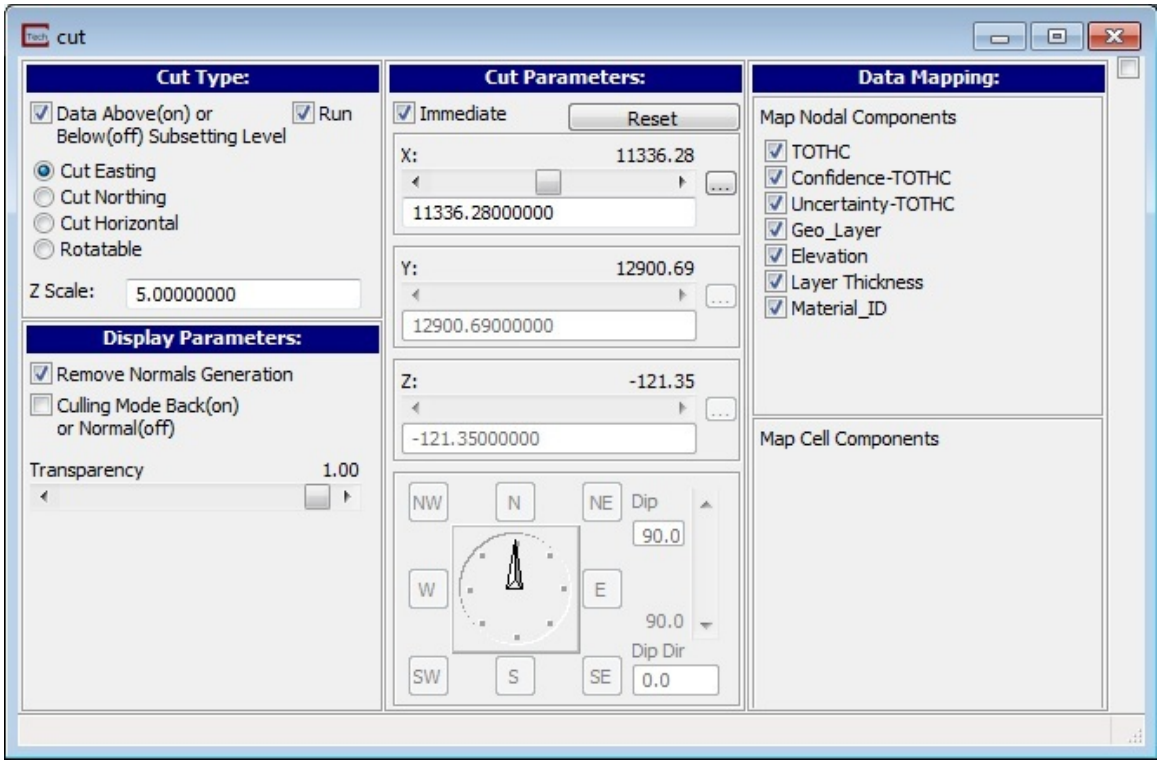
Cut has two input ports.

1. The leftmost port accepts any 1D, 2D or 3D mesh. The mesh can contain nodal data and/or cell data.
2. The right port is the Z-Scale and is required if Explode_and_Scale is upstream, in order to reference the correct z coordinates when options 3 or 4 are selected.

Module Output Ports

Cut has four output ports.

1. The leftmost is the Z-Scale
2. The second (blue/black) output port provides the cut (subsetted) output.
3. The third (blue/black) output port is the original grid (no subsetting performed) with an additional nodal data component which provides the distance to the cutting plane. This is useful if you plan to use a module like sequential subset to create a transparent output without internal faces. The name of the data component will be the name of the module.
4. The red output port is a renderable version (for connection to the Viewer) of the second port.

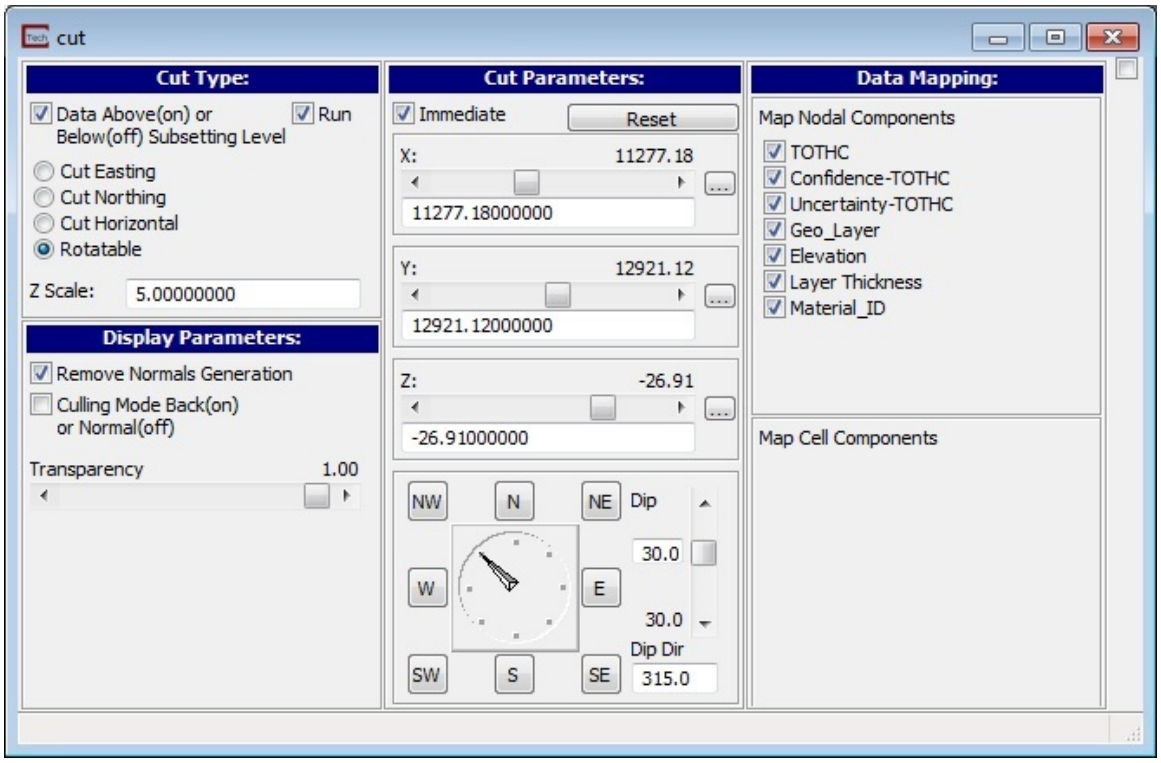


Module Control Panel

The control panel for Cut is shown in the figure above. The above example shows "Cut Easting" as the selected cut type. For any of the first three cut types, the plane is defined by a single coordinate input (X, Y, or Z).

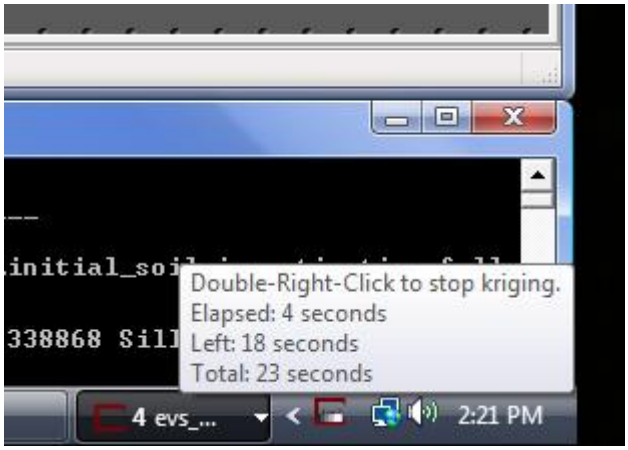
- The ***Above*** check box is used to determine whether the mesh above or below the cutting plane is sent to the output port.
- The ***Run*** toggle will either allow or prevent the module from running.
- The ***Cut Type*** is how the cutting plane is specified. It can be one of four methods:
    1. A vertical plane defined by an X or **Easting** coordinate
    2. A vertical plane defined by a Y or **Northing** coordinate
    3. A **Horizontal** plane defined by a Z coordinate
    4. An arbitrarily positioned **Rotatable** plane which requires:
        1. A 3D point through which the cutting plane passes. Please note that the same cutting result can be achieved with an infinite number of 3D points, all of which would be on the same cutting plane.
        2. A ***Dip*** direction
        3. A ***Strike*** direction

- The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and the default "On" state is generally preferable when cutting 3D objects.
- ***Z Scale*** is required if Explode_and_Scale is upstream, in order to reference the correct z coordinates when options 3 or 4 are selected.
- The ***Cut Parameters*** are the coordinates and/or Dip/Strike angles required to specify the position and orientation of the cutting plane. These include:
    - X coordinate (used with Cut Easting and Rotatable)
    - Y coordinate (used with Cut Northing and Rotatable)
    - Z coordinate (used with Cut Horizontal and Rotatable)
    - Dip angle (used only with Rotatable)
    - Dip Direction (used only with Rotatable)
- ***Map Nodal Components*** determines which nodal components will be sent to the output ports. Also, the first map component selected will be the variable used to color the output.
- The ***Map Cell Components*** option box selector lets you map cell data (if any) to the output. The cell data will only be visible if all nodal data components are unselected (under *Map Nodal Components*).



Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Related Modules

-> slice

constant_shell

General Module Function

The constant_shell module creates a 2D surface subset from a 3D input. The resulting surface represents all regions in your 3D input which are exactly equal to the selected Subsetting Level. This module replaces the isosurface module.

Module Input Ports

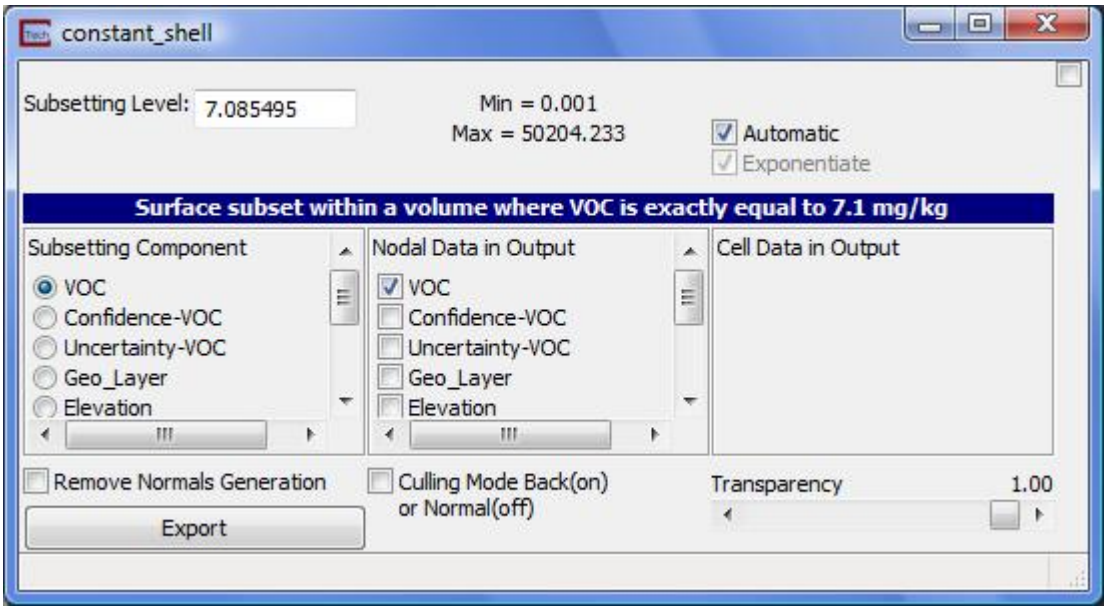
constant_shell has two input ports.

1. The leftmost (blue-black) port accepts unstructured mesh data.
2. The second (maroon) port provides a means to share the subsetting level with other modules.

Module Output Ports

constant_shell has four output ports.

1. The first output port (leftmost) provides an "info" string which can be connected to the Titles module for a quick simple title associated with the settings of this module. For the settings shown below, the Title displayed would be: "TOTHC above 11 mg/kg". This is constructed using the "Subsetting Component" name followed by "above" or "below", a rounded representation of the subsetting level and the units (if specified in the input).
2. The second (maroon) port provides a means to share the subsetting level with other modules.
3. The third (blue-black) port outputs a new unstructured mesh which contains the isosurface which is all regions in your 3D volumetric model which are EXACTLY equal to the Subsetting Level.
4. The fourth (red) port outputs a rendered geometry directly to the Viewer.



Module Control Panel

The control panel for constant_shell is shown in the figure above.

The ***SubsettingLevel*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is used. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default level is the arithmetic average of the minimum and maximum values in the subsetting component. If your input data has been kriged with log processing, the values here will be exponentiated already (provided that the Automatic toggle is on). You will not enter the Log of your desired subsetting value. Also, when the data is log processed, the default value is the geometric mean (vs. arithmetic mean) of your data extremes.

The ***Automatic*** toggle when ON determines "automatically" if the subsetting level values will be exponentiated for you vs. having to input value which are the LOG of your desired level. When this is on the Exponentiated toggle is inactive.

The ***Exponentiate*** toggle is inactive if the Automatic toggle is ON. Otherwise, it determines whether to real units to logarithmic units..

Subsetting Component refers to the nodal data component used to create the subset of the original input field. When an component is selected, the min and max values of the variable are displayed in the upper center of the dialog box. The default component is the first (0th) component in the column.

Nodal Data in Output determines which nodal data components will be included in the output. The first one in the list will determine the data used for coloring in the (red) output to the Viewer. For example, choosing subsetting component concentration and uncertainty for the nodal data in the output will create a volumetric SHELL subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The ***CellData in Output*** option box selector lets you map cell data (if any) to the surfaces output by constant_shell. The cell data will only be visible if all nodal data is unselected (under *nodal data in output*).

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Culling Mode Back (on) or Normal (off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

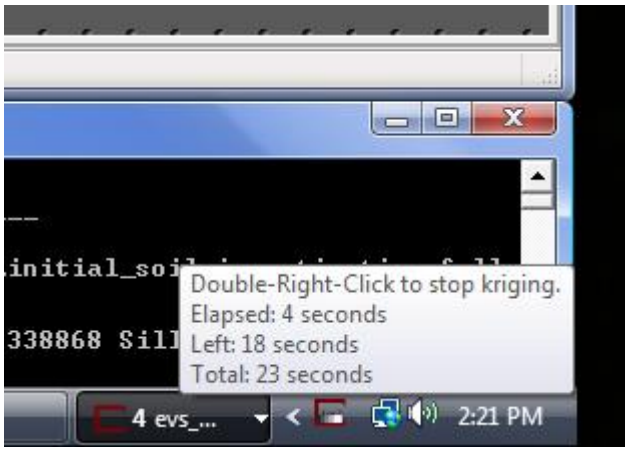
The ***Transparency*** slider changes the transparency of the output to the red port to the Viewer.

The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. Once this button is pushed, constant_shell will have has seven output ports.

1. The info string
2. The subsetting level
3. The string representing the selected subsetting component
4. The string representing "above" or "below"
5. The units as a string
6. The (blue-black) output
7. The (red) output to the Viewer.

Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Related Modules

-> plume_shell

-> plume_area

slope_aspect_splitter

General Module Function

The slope_aspect_splitter module will split an input field into two output fields based upon the slope and/or aspect of the external face of the cell and the subset expression used. The input field is split into two fields one for which all cells orientations are true for the subset expression, and another field for which all cells orientations are false for the subset expression.

All data from the original input is preserved in the output.

Module Input Ports

slope_aspect_splitter has four input ports:

1) in_field (Blue-Black): This port accepts the field to be split.

2) f1 (Dark Brown): This port accepts a double value that can be used as a variable in the subset expression.

3) f2 (Dark Brown): This port accepts a double value that can be used as a variable in the subset expression.

4) z_scale (Grey-Grown): This port passes the z-exaggeration factor. NOTE: The z_scale is important in this module because it affects the slope value calculated for the external face of each cell.

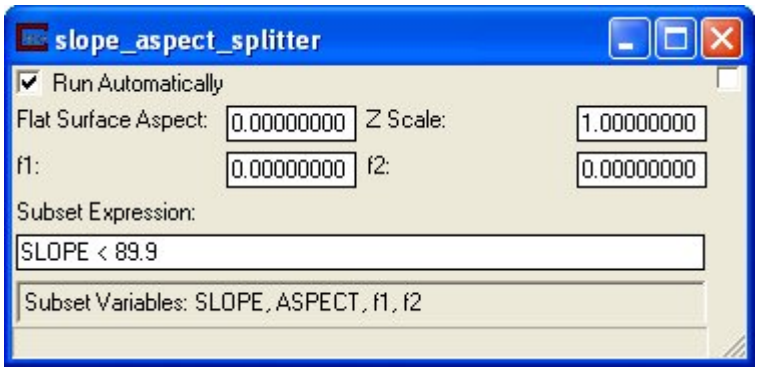
Module Output Ports

slope_aspect_splitter has three output ports

1) z_scale (Grey-Grown): This port passes the z-exaggeration factor.

2) output_true(Blue-Black): This port passes the created output field for those cells which met the criteria of the subset expression.

3) output_false(Blue-Black): This port passes the created output field for those cells which failed to meet the criteria of the subset expression.



Module Control Panel

The panel for slope_aspect_splitter is shown above.

Run Automatically: This toggle when turned on will cause the module to update every time the input or control parameters are changed.

Flat Surface Aspect: If you have a flat surface then a realistic aspect can not be generated. This field lets you set the value for those sells.

Z_Scale: This z-exaggeration factor. NOTE: The z_scale is important in this module because it affects the slope value calculated for the external face of each cell.

f1: This is a double value that can be used as a variable in the subset expression.

f2: This is a double value that can be used as a variable in the subset expression.

Subset Expression: This is the boolean expression used to divide the cells. Expression variables include: SLOPE; ASPECT; f1; and f2.

1) **To output just the top surface:** set the subset expression to be **SLOPE < 89.9**, since the sides of the model would be at slope of 90 degrees and the bottom would be 90+ degrees. (Notice there is potential for rounding errors use 89.9 instead of 90)

2) General expression (assuming a standard cubic building)

A) SLOPE > 0.01 (Removes the top of the building)

B) SLOPE > 0.01 and SLOPE < 179.9 (Removes the top and bottom of the building)

3) Since ASPECT is a variable it must be defined for each cell. In cells with a slope of 0 or 180 there would be no aspect without our defining it with the *flat surface aspect* field

4) Units are always degrees. You could change them to radians if you want inside the expression. (SLOPE * PI/180)

Related Modules

-> slope_aspect

subset_layers

General Module Function

The subset_layers module allows you to subset the output of Krig_3D_Geology so that downstream modules (Krig_3D, 3D_Geology_Map, Geologic Surface) act on only a portion of the layers kriged.

subset_layers is used to select a subset of the layers (and corresponding surfaces) export from Krig_3D_Geology. This is useful if you want (need) to krige parameter data in each geologic layer separately.

This is not normally needed with contaminant data, but when you are kriging data such as porosity that is inherently discontinuous across layer boundaries, it is essential that each layer be kriged with data collected ONLY within that layer.

Subset_layers eliminates the need for multiple Krig_3D_Geology modules reading data files that are subsets of a master geology. Inserting subset_layers between Krig_3D_Geology and Krig_3D allows you to select one or more layers from the geology.

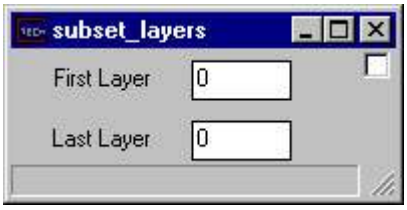
This functionality is very useful when you want to krige groundwater and soil data using a single master geology file that represents both the saturated and unsaturated zones.

Module Input Ports

subset_layers has one input port which only accepts input from Krig_3D_Geology, Spline_Geology or Load_EVS_Field (when the output of one of the other two modules was saved).

Module Output Ports

subset_layers has one output port that outputs a subset of the layers created by Krig_3D_Geology. It connects to modules such as Krig_3D, 3D_Geology_Map, or Geologic Surface.



Module Control Panel

The control panel for subset_layers is shown in the figure above. There are only two parameters. These are the "first layer" and "last layer" to be included in the subset. If inappropriate values are entered by the user, the module will revise the values to the nearest valid values.

Related Modules

Krig_3D_Geology

make_single_layer

General Module Function

The make_single_layer module allows you to subset the output of Krig_3D_Geology so that downstream modules (Krig_3D, 3D_Geology_Map, Geologic Surface) act on only a single merged layer.

make_single_layer is used to merge all layers (and corresponding surfaces) export from Krig_3D_Geology into a single layer (topmost and bottommost surfaces).

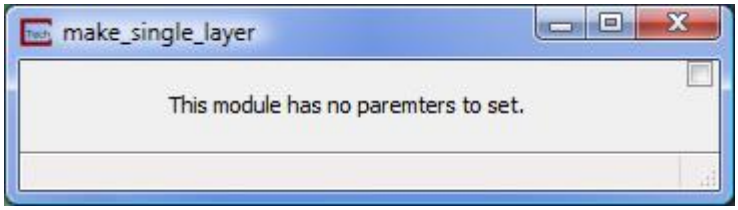
Make_single_layer eliminates the need for multiple Krig_3D_Geology modules reading data files that are single layer subset of a master geology. Inserting make_single_layer between Krig_3D_Geology and Krig_3D kriges all data into a single geologic layer. When used with subset_layers it allows for creating a single layer that represents a only a portion (subset) of the master geology file.

Module Input Ports

make_single_layer has one input port which only accepts input from Krig_3D_Geology or Read_EVS_Geology.

Module Output Ports

make_single_layer has one output port that outputs a subset of the layers created by Krig_3D_Geology. It connects to modules such as Krig_3D, 3D_Geology_Map, or Geologic Surface.



Module Control Panel

make_single_layer has no control panel. It always creates a single layer from geologic input containing one or more layers.

Related Modules

Krig_3D_Geology

subset_cells

General Module Function

The subset_cells module is a subsetting module used to eliminate whole cells from the input mesh based on the nodal data of one or more nodes that comprise the cells. It outputs a blocky mesh of cells bounded by the threshold value.

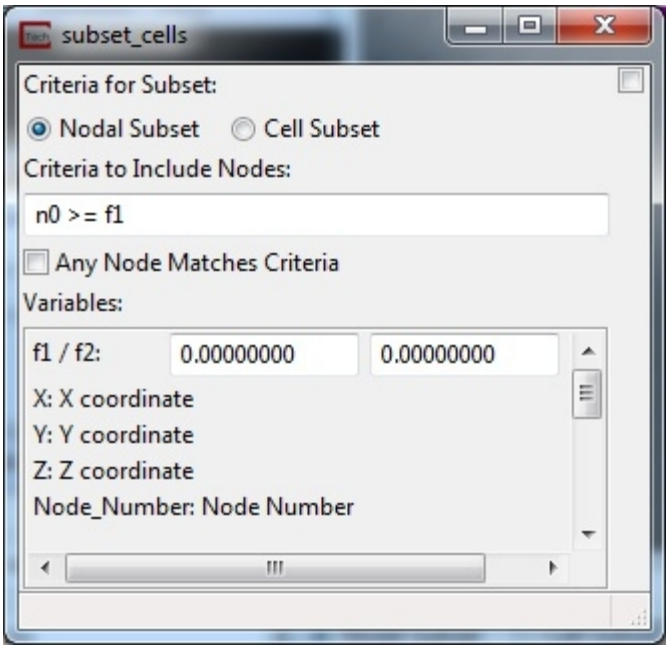
This subset_cells module replaces the older version of threshold which is deprecated. It works in a manner like data_math or coordinate_math. It allows you to do complex threshold calculations on coordinates and MULTIPLE data components with a single module, which can dramatically simplify your network and reduce memory usage. It has 2 floating point variables (f1,f2) which are setup with ports so they can be easily animated.

Module Input Ports

subset_cells has three input ports. Input sent to the left port must contain nodal data. The other two port are float numbers that can be used in the thresholding expression.

Module Output Ports

subset_cells has four output ports. The first two output ports (closest to the left) are the float values passed through. The third outputs only those cells where all the cell's nodes meet the inclusion criteria. The last (red) port outputs a renderable geometry.



Module Control Panel

The control panel for subset_cells is shown above with an input field having 6 nodal data components. This module is similar to plume_volume in its subsetting but more like data_math in it use.

The Criteria for Subset option allows for subsetting by either a nodal data component or a cell data component. The logical expression in the *Criteria to Include Nodes* type-in determines which nodes will be included in the output.

Example Expressions:

1) n0 >= f1 All nodes with "Concentration0" greater than or equal to f1 will be included.

2) (n0 < f1) or (n1 < f2) All nodes with "Concentration0" less than f1 OR "60PctMin-Conc0" less than f2 will be included.

3) (n0 >= f1) and (X < 11500.0) All nodes with "Concentration0" greater than or equal to f1 and X coordinates less than 11,500 will be included.

Jump to a list of available Mathematical Operators

orthoslice

General Module Function

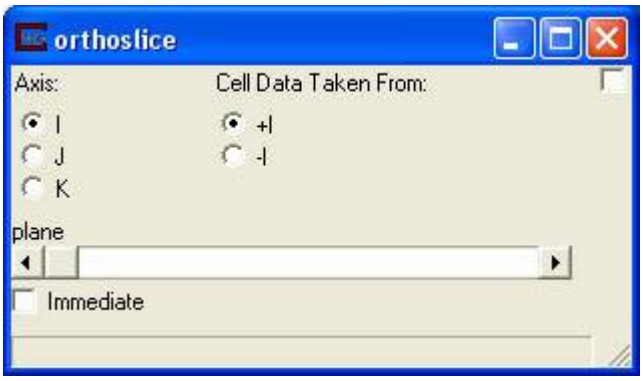
The orthoslice module is similar to the slice module, except limited to only displaying slice positions north-south (vertical), east-west (vertical) and horizontal. orthoslice subsets a structured field by extracting one slice plane and can only be orthogonal to the X, Y, or Z axis. Although less flexible in terms of capability, orthoslice is computationally more efficient.

Module Input Ports

orthoslice has one input port. Data passed to this port must contain a structured mesh (a field as opposed to a UCD) and Node_Data. The input can be 1D, 2D, or 3D.

Module Output Ports

orthoslice has two output ports. The leftmost output port creates a field with a new structured mesh of the same type as the input field. The mesh is the mesh of the slice. It contains the slice's Node_Data. The second output port sends a renderable version of the field (a slice) to the viewer.



Module Control Panel

The control panel for orthoslice is shown in the figure above:

The ***axis*** selector chooses which axis (I, J, K) the orthoslice is perpendicular to. The default is I. If the field is 1D or 2D, three values are still displayed. Select the values meaningful for the input data.

The ***plane*** slider selects which plane to extract from the input. This is similar to the position slider in slice but, since the input is a field, the selection is based on the nodal dimensions of the axis of interest. Therefore, the range is 0 to the maximum nodal dimension of the axis. For example, for an orthoslice through a grid with dimension 20 x 20 x 10, the range in the x and y directions would be 0 to 20.

The ***Cell Data Taken From*** selector determines which side of the nodal suface the cell data should taken from. Since orthoslice "slices" between layers of cells, the data must be selected from one side or the other.

The ***Immediate*** toggle causes the display to update **as you move the slider(s).**

Related Modules

->slice

crop_and_downsize

General Module Function

The crop_and_downsize module is used to subset an image, or structured 1D, 2D or 3D mesh (an EVS "field" data type with implicit connectivity). Similar to cropping and resizing a photograph, crop_and_downsize sets ranges of cells in the I, J and K directions which create a subset of the data. When used on an image (which only has two dimensions), crop removes pixels along any of the four edges of the image. Additionally, crop_and_downsize reduces the resolution of the image or grid by an integer downsize value. If the resolution divided by this factor yields a remainder, these cells are dropped.

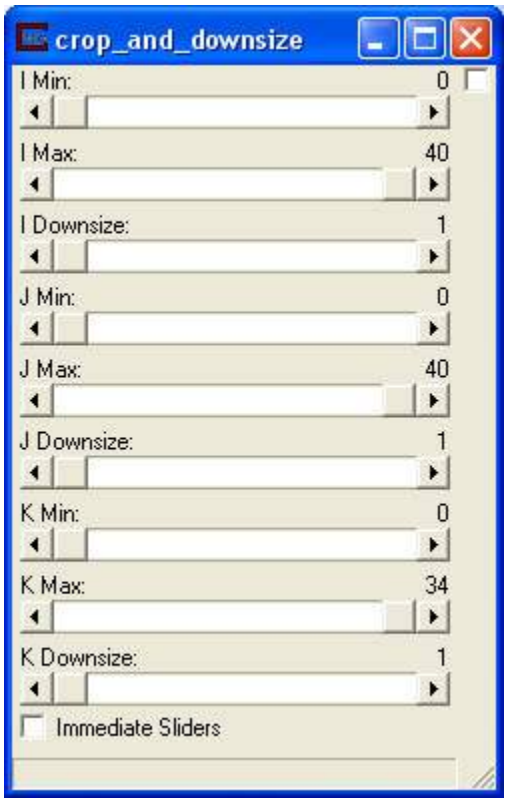
crop_and_downsize refers to I, J, and K dimensions instead of x-y-z. This is done because grids are not required to be parallel to the coordinate axes, nor must the grid rows, columns and layers correspond to x, y, or z. You may have to experiment with this module to determine which coordinate axes or model faces are being cropped or downsized.

Module Input Ports

crop_and_downsize has only one input port. Input to this port must contain structured mesh data and nodal data.

Module Output Ports

crop_and_downsize has two output ports. The first output port (closest to the left) outputs a structured mesh data and nodal data of the cropped region. The second port outputs a renderable geometry.



Module Control Panel

The control panel for crop_and_downsize is shown in the figure above. Sliders are used to select the I, J, and K dimension crop ranges. Each dimension range has sliders to select the min, max value and downsize factor. For example, to crop the structured mesh in the I coordinate direction from 10 to 22, set the I min slider to 10 and the I max slider to 22. The slider selections are inclusive. The default min and max for each coordinate direction reflect the min and max of the mesh data.

The ***ImmediateSliders*** toggle causes the display to update **as you move the slider(s).**

Related Modules

-> cut

resample

The resample module (previously named downsize) is generally superceded by the newer crop_and_downsize module.

General Module Function

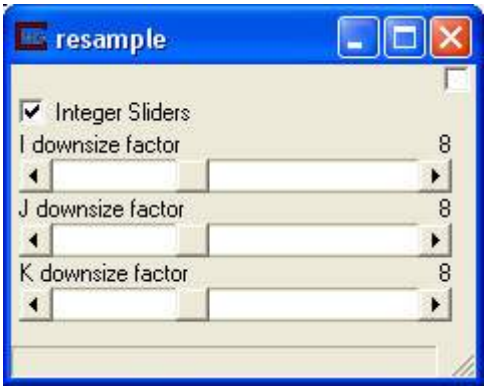
The resample module (previously named downsize) reduces or enlarges a 1D, 2D, or 3D structured mesh (an EVS field data type) by eliminating or adding nodes. Downsize factors greater than 1 create smaller data fields by thinning out or skipping data. Resample factors less than 1 create larger data files by duplicating (via interpolation) data.

Module Input Ports

Resample has only one input port. Input to this port must contain structured mesh data and nodal data.

Module Output Ports

Resample has two output ports. The first output port (closest to the left) outputs a new structured mesh of the same type as the input mesh and new nodal data containing data associated with the downsized region. The second output port is undocumented at this time



Module Control Panel

The control panel for resample is shown in the figure above. Three sliders are used to set the downsize factors for the I, J, and K dimensions. The default downsize factor for each dimension is 8 and the range is 0 to 20. The integer slider check box toggles between integer values only for the sliders (on) or float values for the sliders (off). Float sliders are most useful for enlarging the data field (downsize factor <1).

Related Modules

crop

cell_centers

General Module Function

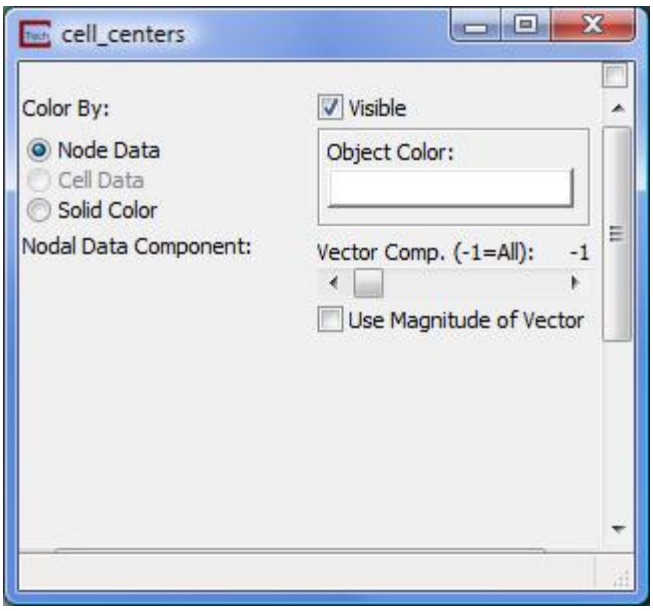
cell_centers module produces a mesh containing Point cell set, each point of which represents a geometrical center of a corresponding cell in the input mesh. The coordinates of cell centers are calculated by averaging coordinates of all the nodes of a cell. The number of nodes in the output mesh is equal to number of cells in the input mesh. If the input mesh contains Cell_Data it becomes a Node_Data in the output mesh with each node values equal to corresponding cell value. **Nodal data is not output directly.** You can use this module to create a position mesh for the glyph module. You may also use this module as mesh input to the interp_data module, then send the same nodal values as the input grid, to create interpolated nodal values at cell centroids.

Module Input Ports

cell_centers has one input ports which can contain any type of mesh.

Module Output Ports

cell_centers has two output ports. The first output port (closest to the left) outputs a new mesh that consists of points representing geometrical centers of the corresponding cells in the input mesh. It also may contain a Node_Data that corresponds to Cell_Data in the input mesh. The second port outputs a renderable version of the output field



Module Control Panel

The user interface for this module is shown above.

edges

General Module Function

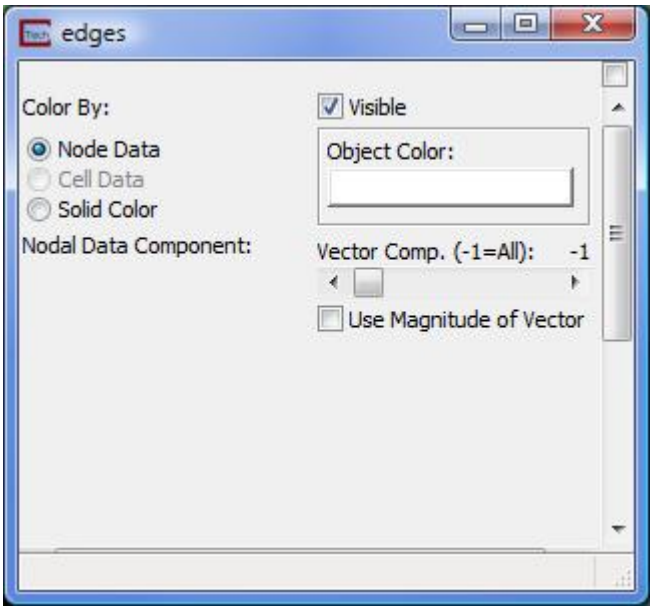
The edges module is similar to the External_Edges module in that it produces a wireframe representation of the nodal data making up an unstructured cell data mesh. There is however, no adjustment of edge angle and therefore only allows viewing of all grid boundaries (internal AND external) of the input mesh. The edges module is useful in that it is able to render lines around adaptive gridding locations whereas external_edges does NOT render lines around this portion of the grid.

Module Input Ports

edges has one input port which accepts unstructured mesh data and it's associated nodal data.

Module Output Ports

edges has two output ports. The leftmost port outputs a mesh composed of the original mesh and nodal data. The second output port sends renderable lines to the viewer.



Module Control Panel

The user interface for this module is shown above.

bounds

General Module Function

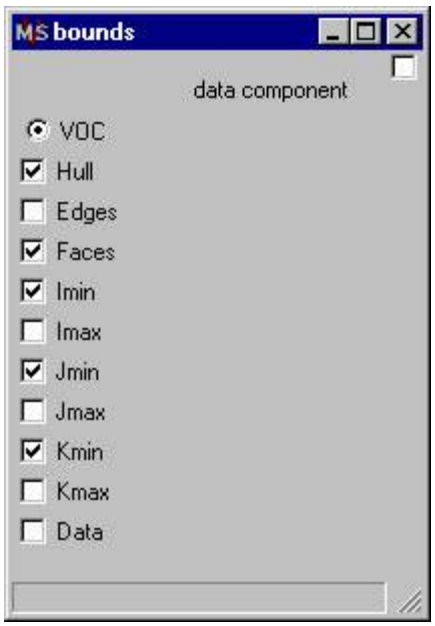
bounds generates lines and/or surfaces that indicate the bounding box of a 3D structured field. This is useful when you need to see the shape of an object and the structure of its mesh. This module is similar to external_edges (set to edge angle = 60), except, bounds allows for placing faces on the bounds of a model.

Module Input Ports

bounds has one input ports. Data passed to the first port (closest to the left) must contain any type of structured mesh (a field as opposed to a UCD). Node_Data can be present, but is only used if you switch on Data.

Module Output Ports

bounds has two output ports. The leftmost output port creates a new unstructured Mesh object with cell type Polyline (Edges selected) and/or Quad (Faces set) representing the bounds.. The second output port sends a renderable version of the output field to the viewer.



Module Control Panel

The control panel for bounds is shown in the figure above:

The data component radioBox selects which of the input field's components is sent to the output, if Data is also turned on. The default is the first (0th) component. If node data labels are present, they are displayed.

Hull, when on, draws a wireframe around the perimeter extents of the mesh. The default is on.

Edges, when on, causes the Imin/Imax, Jmin/Jmax, Kmin/Kmax controls to produce a wireframe representation of the mesh grid at that plane. The default is off.

Faces, when on, causes the Imin/Imax, Jmin/Jmax, Kmin/Kmax controls to produce a solid face representing the location of that plane extent. The default is off.

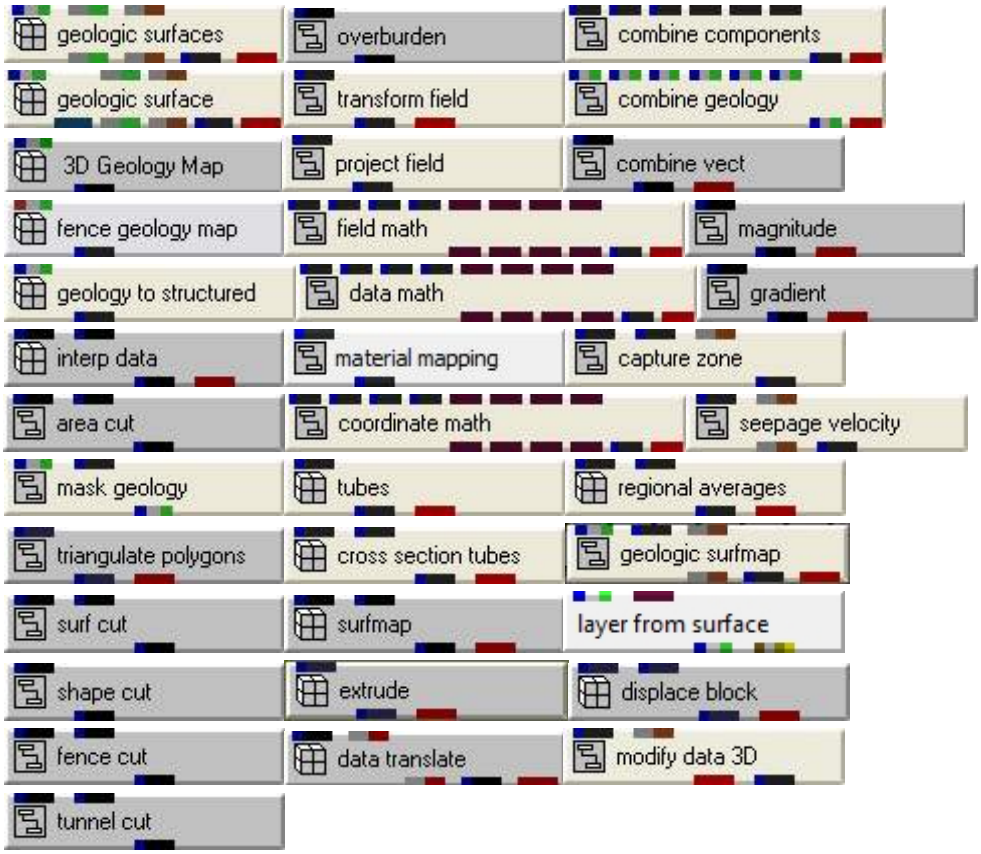
When on, each of these switches displays the grid (Edges turned on) or plane (Faces turned on) on one of the six faces of the hull. Imin/Imax draw a mesh or face showing the 2D slice of field objects with the minimum/maximum index value in the first dimension. Jmin/Jmax draw a mesh or face showing the 2D slice of field objects with the minimum/maximum index in the second dimension. Kmin/Kmax control the third dimension. The default for all is off.

data, when on, makes bounds copy the selected component's Node_Data values at node points along the output mesh to the output field. Because the data is present, the bounds lines can be colored by the interpolated data values of the selected Node_Data component. Otherwise, the lines display as white in the renderer. The default is off.

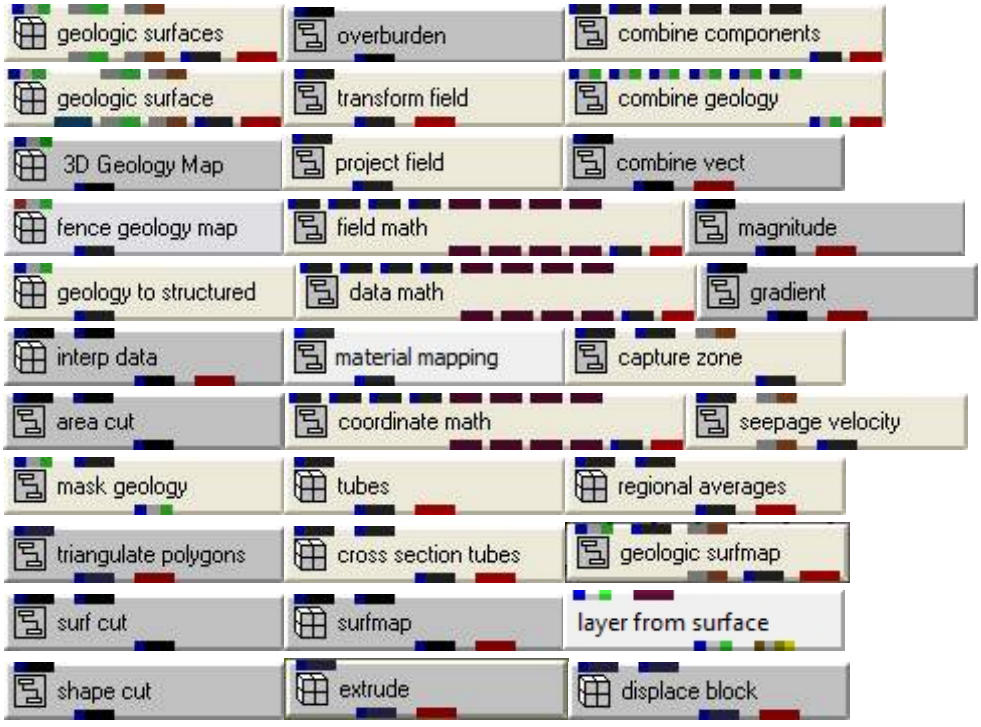
Related Modules

->external_edgesexternal_edges

Processing Modules



Processing Modules



geologic_surfaces

General Module Function

The geologic_surfaces module provides complete control of displaying, scaling and exploding one or more geologic surfaces from the set of surfaces output by Krig 3D Geology. This module allows visualization of the topology of any or all surfaces and\or the interaction of a set of individual surfaces.

geologic_surfaces can explode geologic surfaces analogous to how Explode_and_Scale explodes layers created by 3D_Geology_Map or Krig_3D. The ability to explode the surfaces is integral to this module.

geologic_surfaces also allows the user to either color the surface according to the surface Elevation or any other data component exported by Krig_3D_Geology.

Module Input Ports

geologic_surfaces has three input ports.

1. The leftmost port accepts the 2-D surface data fields from Krig_3D_Geology, Spline_Geology or Read_netCDF when it was used to save the output of either of these modules. Please note that if any portions of the input geology is NULL, these cells will be omitted from the surface that is created. This can save memory and provide a means to cut (in a Lego fashion) along boundaries.
2. The middle input port will accept the explode factor from the Explode and Scale module. Note using Explode and Scale with geologic_surfaces is not necessary,.as geologic_surfaces has its own explode factor parameter that can be used to set the surface separation directly.
3. The right input port will accept a scale factor from the Explode and Scale module or other modules with a Z scale port. Note using Explode and Scale with geologic_surfaces is not necessary,.as geologic_surfaces has its own scale factor parameter that can be used to set the surface scaling directly.

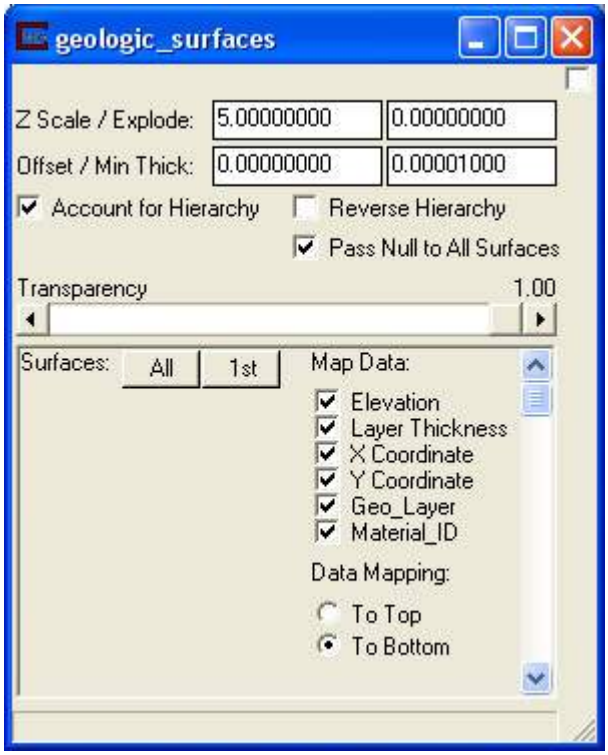
Module Output Ports

geologic_surfaces has four output ports.

1. The first port (grey-green) outputs the explode factor.
2. The second port outputs the Z-exaggeration.
3. The third port, from the left, outputs the data field containing all selected surfaces. This port can be connected to any module that can process data fields, such as the isoline module, which can use the data

field to display isolines on the geologic surface, the statistics module, which can report the characteristics of the suface data, or the generate axes module, which can generate a set of axes around the surface display.

4. The rightmost port outputs a renderable geometry of the surface that can be input to the viewer. This port can also be connected to the color legend module to set the limits of the displayed color scale.



Module Control Panel

The control panel for geologic_surfaces is shown above.

The ***Z Scale*** determines the amount of vertical (or z-coordinate direction) scaling that will be applied to each geologic layer.

Explode determines the distance by which each geologic layer will appear to be separated. In geologic_surfaces, explode causes the surfaces to stay aligned with exploded layers.

The **Surface Z Offset** parameter specifies the distance that the surface willl be translated in the Z axis before display.

The ***Min Thick****(ness)* parameter determines the minimum spacing between surfaces when hierarchy is imposed.

Note that these parameters affect the Z coordinates of the grid nodes, but not the values of the depth or elevation that are associated with those nodes by Krig 3D. Recall that in EVS each grid node has x,y,z, coordinates, and property values associated with the node. The property value assigned to the nodes by Krig_3D_Geology is the original (pre-scaling and translation) elevation or depth values. The Z

Exaggeration and Surface Z Offset parameters affect the Z coordinates of the nodes of the quadrilateral elements in the mesh, but do not affect the property values associated with the nodes. This distinction is important for the user to remeber when passing the field data from geologic_surfaces to other modules. As an example, if the field output of geologic_surfaces is connected to the Statistics module, the user will see that specifying different Z Exaggeration and Surface Z Offset parameters affects the *coordinate extents* reported by Statistics, but has no effect on the *parameter value* distribution reported (.ie., the min, max , mean, std deviation, and histogram bins).

The ***Account for Hierarcy*** toggle causes geologic surfaces that might otherwise intersect each other to be corrected to account for Geologic Hierarchy.

The ***Pass Null to All Surfaces*** toggle causes the Null nodes/cells defined in any surface to be passed to all of the surfaces.

The ***Reverse Hierarcy*** toggle causes hierarchy to be imposed from the bottom up versus the normal top down.

The ***Transparency*** slider affects the transparency (opacity) of all surfaces.

The ***Surfaces*** toggles allow you to control which surfaces are output for visualization or subsequent processing.

The ***All & 1st*** buttons respectively set (turn on) all of the surfaces toggles or only the first one in the list.

The ***Map Data*** toggles allow you to control which data components are included in the surfaces.

The ***Data Mapping*** radio-box toggles allow you to control to which surfaces the data components are mapped. For example, if Top is selected, layer thicknesses are mapped to the surface that corresponds to the top of each layer.

geologic_surface

This module allows visualization of the topology of any single surface.

geologic_surface can explode the geologic surface analogous to how Explode_and_Scale explodes layers created by 3D_Geology_Map or Krig_3D. The ability to explode the surface is integral to this module.

geologic_surface also allows the user to either color the surface according to the surface Elevation or any other data component exported by Krig_3D_Geology.

Module Input Ports

geologic_surface has three input ports.

The leftmost port accepts the 2-D surface data fields from Krig_3D_Geology, Spline_Geology or Read_netCDF when it was used to save the output of either of these modules. Please note that if any portions of the input geology is NULL, these cells will be omitted from the surface that is created. This can

save memory and provide a means to cut (in a Lego fashion) along boundaries.

The middle input port will accept the explode factor from the Explode and Scale module. Note using Explode and Scale with geologic_surface is not necessary,.as geologic_surface has its own explode factor parameter that can be used to set the surface separation directly.

The right input port will accept a scale factor from the Explode and Scale module or other modules with a Z scale port. Note using Explode and Scale with geologic_surface is not necessary,.as geologic_surface has its own scale factor parameter that can be used to set the surface scaling directly.

Module Output Ports

geologic_surface has five output ports

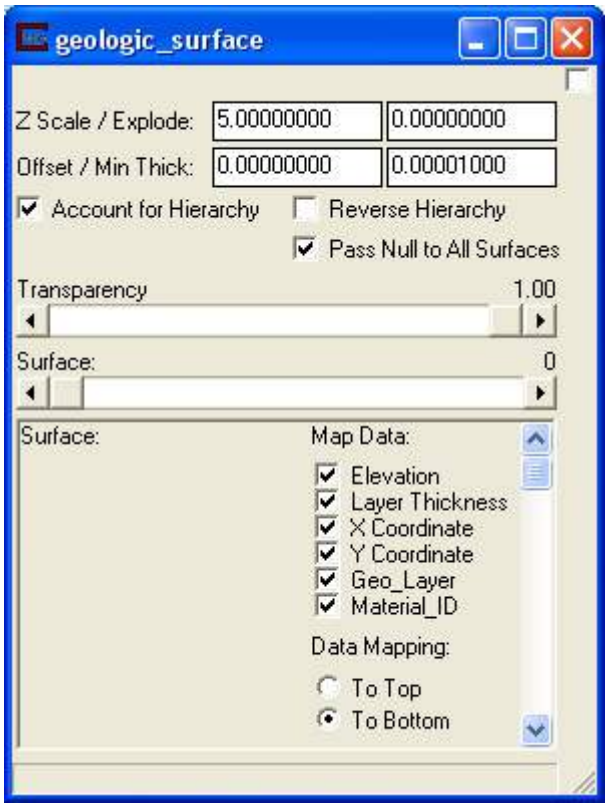
The first port outputs the surface name as a string;

The second port (grey-green) outputs the explode factor.

The third port outputs the Z-exaggeration.

The fourth port, from the left, outputs the data field containing all selected surfaces. This port can be connected to any module that can process data fields, such as the isoline module, which can use the data field to display isolines on the geologic surface, the statistics module, which can report the characteristics of the suface data, or the generate axes module, which can generate a set of axes around the surface display.

The rightmost port outputs a renderable geometry of the surface that can be input to the viewer. This port can also be connected to the color legend module to set the limits of the displayed color scale.



Module Control Panel

The control panel for geologic_surface is shown above.

The ***Z Scale*** determines the amount of vertical (or z-coordinate direction) scaling that will be applied to each geologic layer.

Explode determines the distance by which each geologic layer will appear to be separated. In geologic_surface, explode causes the surfaces to stay aligned with exploded layers.

The (Surface Z) **Offset** parameter specifies the distance that the surface willl be translated in the Z axis before display.

The ***Min Thick****(ness)* parameter determines the minimum spacing between surfaces when hierarchy is imposed.

Note that these parameters affect the Z coordinates of the grid nodes, but not the values of the depth or elevation that are associated with those nodes by Krig 3D. Recall that in EVS each grid node has x,y,z, coordinates, and property values associated with the node. The property value assigned to the nodes by Krig_3D_Geology is the original (pre-scaling and translation) elevation or depth values. The Z Exaggeration and Surface Z Offset parameters affect the Z coordinates of the nodes of the quadrilateral elements in the mesh, but do not affect the property values associated with the nodes. This distinction is important for the user to remeber when passing the field data from geologic_surface to other modules. As an example, if the field output of geologic_surface is connected to the Statistics module, the user will see that specifying different Z Exaggeration and Surface Z Offset parameters affects the *coordinate extents* reported by Statistics, but has no

effect on the *parameter value* distribution reported (.ie., the min, max , mean, std deviation, and histogram bins).

The ***Account for Hierarchy*** toggle causes geologic surfaces that might otherwise intersect each other to be corrected to account for Geologic Hierarchy.

The ***Reverse Hierarchy*** toggle causes hierarchy to be imposed from the bottom up versus the normal top down.

The ***Pass Null to All Surfaces*** toggle causes the Null nodes/cells defined in any surface to be passed to all of the surfaces.

The ***Transparency*** slider affects the transparency (opacity) of all surfaces.

The ***Surfaces*** toggles allow you to control which surfaces are output for visualization or subsequent processing.

The ***Surface*** slider chooses which single surface to display.

The ***Map Data*** toggles allow you to control which data components are included in the surfaces.

The ***Data Mapping*** radio-box toggles allow you to control to which surfaces the data components are mapped. For example, if Top is selected, layer thicknesses are mapped to the surface that corresponds to the top of each layer.

_3D_Geology_Map

General Module Function

The 3D Geology Map module creates 3-dimensional solid layers from the 2-dimensional surfaces produced by Krig_3D_Geology, to allow visualizations of the geologic layering of a system. It accomplishes this by creating a user specified distribution of nodes in the Z dimension between the top and bottom surfaces of each geologic layer.

The number of nodes specified for the Z Resolution may be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) will be placed in each geologic layer.

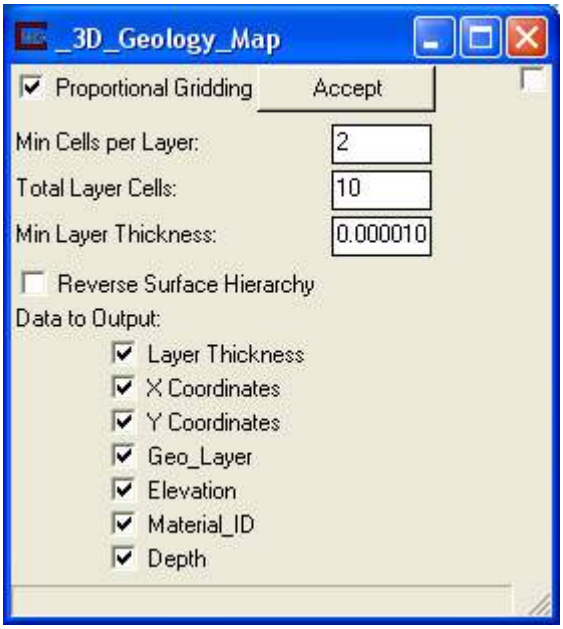
Module Input Ports

3D Geology map has only one input port, which typically receives its input from Krig_3D_Geology. However, it must be of the same type (blue-green-gray port), but may also come from Spline_Geology, or from multiple raster_to_geology modules passed through combine_geology.. Please note that if any portions of the input geology is NULL, these cells will be omitted from the grid that is created. This can save memory and provide a means to cut (in a Lego fashion) along boundaries.

Module Output Ports

3D Geology map has only one output port, which passes a three dimensional data field to modules downstream. In most typical networks, the output port is connected to Explode and Scale. Note that the scalar data components of

layer thickness, x coordinate, y coordinate, Geo_Layer, Elevation, Material_ID and Depth are output from 3D Geology Map, which can be filtered by the select_data module for coloring of the output.



Module Control Panel

The control panel for 3D_Geology_Map is shown above.

Proportional Gridding - A toggle for activating the same algorithm for apportioning nodes in the Z direction as used in Krig_3D. The number of nodes specified for the Z Resolution will be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) will be placed in each geologic layer.

Min Cells per layer - A Type in for establishing a minimum number of cells (in the Z direction) per layer. The default is 2 cells which results in 3 nodes.

Total Layer Cells - A guide for establishing the total number of cells in the model. The reason the term guide is used is due to the possibility that the module could potentially produce more than the total. Here is an example of how this could occur:

If the Min Cells per layer times the number of layers is greater than the Total Layer Cells value, then the input will be superceded. (Example: The Total Layer Cells was 13, but there were 3 layers with Min Cells per Layer of 5, resulting in 15 cells).

NOTE: If Proportional Gridding is toggled off then only Min Cells per Layer is enabled. In this case, only the Min Cells per Layer input will be used for each layer of the model.

Data to Output – These toggles allow you to specify which data components to include in your output. Note that depth allows you to cut geologic layers at a constant distance below ground surface.

Reverse Surface Hierarchy – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

Minimum Layer Thickness forces layers thinner than the specified value to have a minimum thickness and therefore not pinch-out completely.

Further explanations of the use of 3D Geology Map are presented in Workbook2.

fence_geology_map

General Module Function

The fence_geology_map module creates 3-dimensional fence diagram from the 1-dimensional line contours which follow your geology produced by fence_geology, to allow visualizations of the geologic layering of a system. It accomplishes this by creating a user specified distribution of nodes in the Z dimension between the top and bottom lines defining each geologic layer.

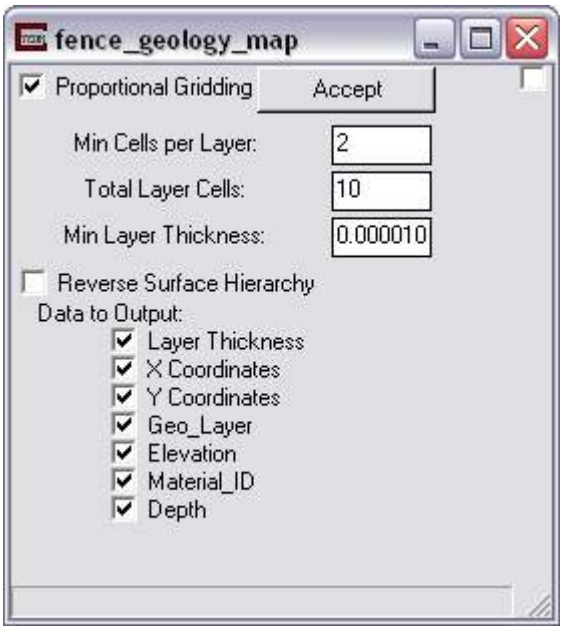
The number of nodes specified for the Z Resolution may be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) will be placed in each geologic layer.

Module Input Ports

fence_geology_map has only one input port, which accepts the 1-D surface data fields from fence_geology. It does not accept data from any other module.

Module Output Ports

fence_geology_map has only one output port, which passes a three dimensional data field (representing fence surfaces) to modules downstream. In most typical networks, the output port is connected to Explode and Scale. Note that the scalar data components of layer thickness, x coordinate, y coordinate, Geo_Layer, Elevation, Material_ID and Depth are selectively output from fence_geology_map, which can be filtered by the select_data module for coloring of the output.



Module Control Panel

The control panel for fence_geology_map is shown above.

Proportional Gridding - A toggle for activating the same algorithm for apportioning nodes in the Z direction as used in Krig_3D. The number of nodes specified for the Z Resolution will be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) will be placed in each geologic layer.

Min Cells per layer - A Type in for establishing a minimum number of cells (in the Z direction) per layer. The default is 2 cells which results in 3 nodes.

Total Layer Cells - A guide for establishing the total number of cells in the model. The reason the term guide is used is due to the possibility that the module could potentially produce more than the total. Here is an example of how this could occur:

If the Min Cells per layer times the number of layers is greater than the Total Layer Cells value, then the input will be superceded. (Example: The Total Layer Cells was 13, but there were 3 layers with Min Cells per Layer of 5, resulting in 15 cells).

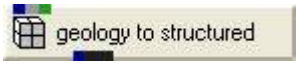
NOTE: If Proportional Gridding is toggled off then only Min Cells per Layer is enabled. In this case, only the Min Cells per Layer input will be used for each layer of the model.

Data to Output – These toggles allow you to specify which data components to include in your output. Note that depth allows you to cut geologic layers at a constant distance below ground surface.

Reverse Surface Hierarchy – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

Minimum Layer Thickness forces layers thinner than the specified value to have a minimum thickness and therefore not pinch-out completely.

geology_to_structured



General Module Function

The geology_to_structured module creates 3-dimensional solid layers from the 2-dimensional surfaces produced by Krig_3D_Geology, to allow visualizations of the geologic layering of a system. It accomplishes this by creating a user specified distribution of nodes in the Z dimension between the top and bottom surfaces of each geologic layer.

This module is similar to 3D_Geology_Map, but does not duplicate nodes at the layer boundaries and therefore the model it creates cannot be exploded into individual layers. However, this module has the advantage that its output is substantially more memory efficient and can be used with modules like crop_and_downsize or ortho_slice.

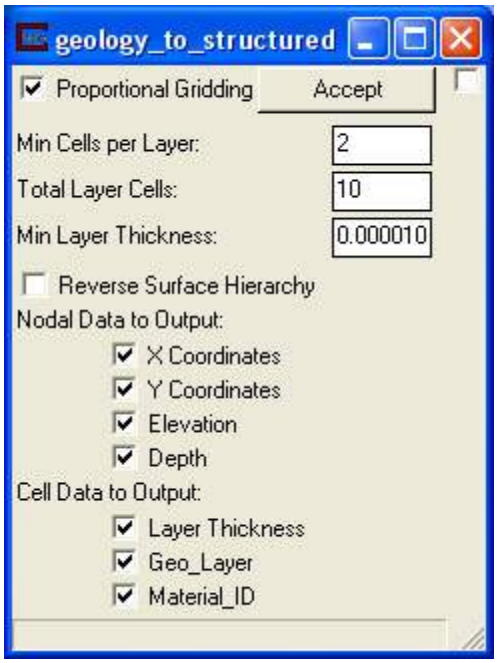
The number of nodes specified for the Z Resolution may be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain.

Module Input Ports

geology_to_structured has only one input port, which accepts the 2-D surface data fields from Krig_3D_Geology. It does not accept data from any other module. Please note that if any portions of the input geology is NULL, these cells will be omitted from the grid that is created. This can save memory and provide a means to cut (in a Lego fashion) along boundaries.

Module Output Ports

geology_to_structured has only one output port, which passes a three dimensional data field to modules downstream. In most typical networks, the output port is connected to Explode and Scale. Note that the scalar data components of layer thickness, x coordinate, y coordinate, Geo_Layer, and z coordinate are output from 3D Geology Map, which can be filtered by the extract scalar module for coloring of the geologic surfaces by these components.



Module Control Panel

The control panel for geology_to_structured is shown above.

Proportional Gridding - A toggle for activating the same algorithm for apportioning nodes in the Z direction as used in Krig_3D. The number of nodes specified for the Z Resolution will be distributed (proportionately) over the geologic layers in a manner that is approximately proportional to the fractional thickness of each layer relative to the total thickness of the geologic domain. In this case, at least three layers of nodes (2 layers of elements) of the Krig_3D domain will be placed in each geologic layer.

Min Cells per layer - A Type in for establishing a minimum number of cells (in the Z direction) per layer. The default is 2 cells which results in 3 nodes.

Total Layer Cells - A guide for establishing the total number of cells in the model. The reason the term guide is used is due to the possibility that the module could potentially produce more than the total. Here is an example of how this could occur:

If the Min Cells per layer times the number of layers is greater than the Total Layer Cells value, then the input will be superceded. (Example: The Total Layer Cells was 13, but there were 3 layers with Min Cells per Layer of 5, resulting in 15 cells).

NOTE: If Proportional Gridding is toggled off then only Min Cells per Layer is enabled. In this case, only the Min Cells per Layer input will be used for each layer of the model.

Reverse Surface Hierarchy – This toggle determines whether the surfaces will define layers in a normal (top to bottom) manner or reversed (if on). This topic is discussed in more detail in Workbook 12.

Minimum Layer Thickness forces layers thinner than the specified value to have a minimum thickness and therefore not pinch-out completely.

Nodal (or Cell) Data to Output – These toggles allow you to specify which data components to include in your output. Note that depth allows you to cut geologic layers at a constant distance below ground surface. Layer_Thickness, Geo_Layer and Material_ID must be included as cell data with this module because nodal layers are not duplicated at geologic unit boundaries.

interp_data

General Module Function

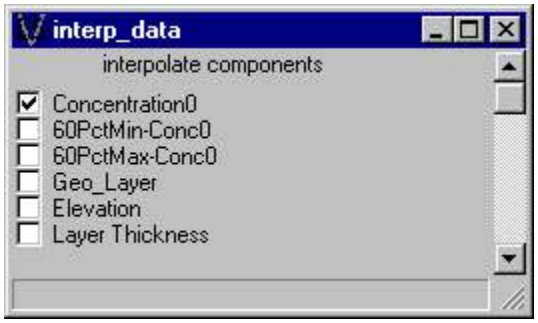
The Interp data module interpolates nodal data from a 3D or 2D field to either a 2D mesh or 1D line. Typical uses of this module are mapping of nodal data from a 3D mesh onto a geologic surface or a 2D fence section. In these applications the 2D surface(s) simply provide the new geometry (mesh) onto which the adjacent nodal values are interpolated. The primary requirement is that the nodal data be equal or higher dimensionality than the mesh to be interpolated onto. For instance, if the user has a 2D surface with nodal data (perhaps z values), then a 1D line may be input and the nearest nodal values from the 2D surface will be interpolated onto it.

Module Input Ports

Interp data has two input ports. The left input port will accept a 2D or 3D field (mesh AND nodal data) which contains the parameter to be interpolated. The right input port accepts a mesh of equal or higher dimensionality. Nodal values are not needed for data sent to the right input port, because the mesh from the right input port is used strictly to obtain coordinate locations where the left ports nodal data will be interpolated.

Module Output Ports

Interp data has two output ports. The left port outputs the data field with interpolated nodal data. This port can be connected to any module that can process data fields, such as the isolines module, which can use the data field to display isolines if the output were a surface. The right port outputs a renderable geometry of the surface that can be input to the viewer. This port can also be connected to the color legend module to set the limits of the displayed color scale.



Module Control Panel

The control panel for Interp data is shown in the figure above. The data component radio buttons are used to select which scalar data component is

to be interpolated onto the surface or line of interest. The default selection is the first (0th) data component. Any number of check boxes may be chosen to pass downstream. The 0th data component is selected by default.

Related Modules

->Geologic_Surface

area_cut

(This module is available only in PRO and MVS)

General Module Function

Area_cut receives any 3D field into its left input port and it receives triangulated polygons (from triangulate_polygon, or other sources) into its right input port. Its function is similar to fence_cut or shape_cut. It adds a data component to the input 3D field and using plume_shell, you can cut structures inside or outside of the input polygons. Only the x and y coordinates of the polygons are used because area_cut cuts a projected slice that is z invariant. Area_cut recalculates when either input field is changed or the "Accept" button is pressed.

Module Input Ports

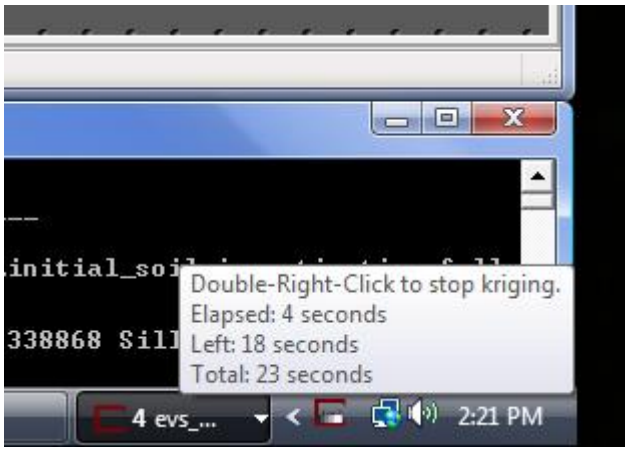
Area_cut has two input ports. Area_cut receives any 3D field into its left input port that should be connected to or after Explode_and_Scale. It receives triangulated polygons (from triangulate_polygon, or other sources) into its right input port.

Module Output Ports

area_cut has one output port which outputs a field with an additional "area_cut" data component which is negative inside the area and positive outside. Zero as an subsetting level cuts on the boundary of the polygon(s).

***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel

The module's control panel is shown above.

The ***Run Immediate*** toggle causes the module to run whenever your inputs change.

The ***Nested Surface Cutting*** toggle allows you to cut donut like shapes if you have polygons INSIDE of other polygons.

The ***Max Edge Angle:*** type-in is similar to the parameter in external_edges. This should not need to be modified unless your cutting area is highly non-planar (3D). If the cutting results seem aberrant, you should try adjusting this.

The ***Accept*** button must be pushed to create output unless the ***Run Immediate*** toggle is on.

Related Modules

->ClickSketch

mask_geology

General Module Function

mask_geology receives geologic input into its left input port and an optional input masking surface into its right port.

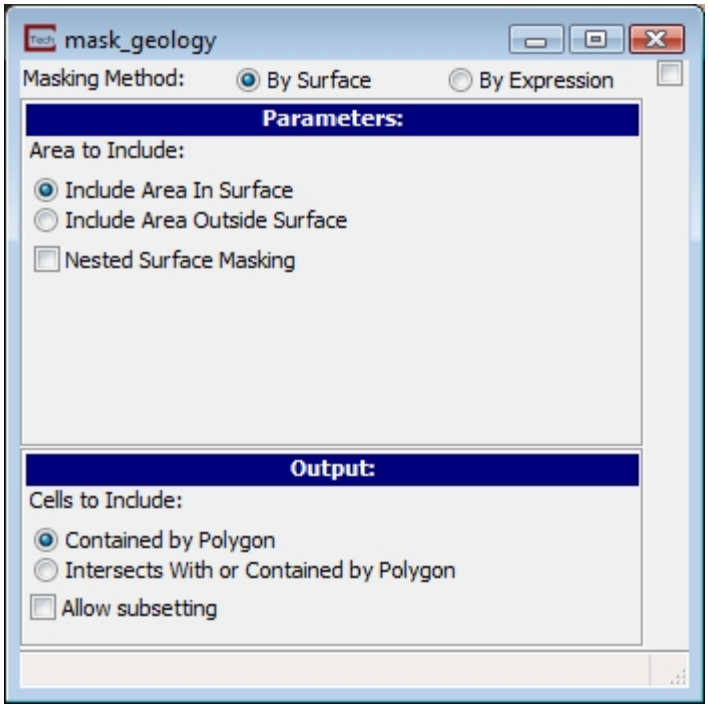
Module Input Ports

mask_geology has two input ports.

1. mask_geology receives geologic input into its left input port and
2. an optional input masking surface into its right port.

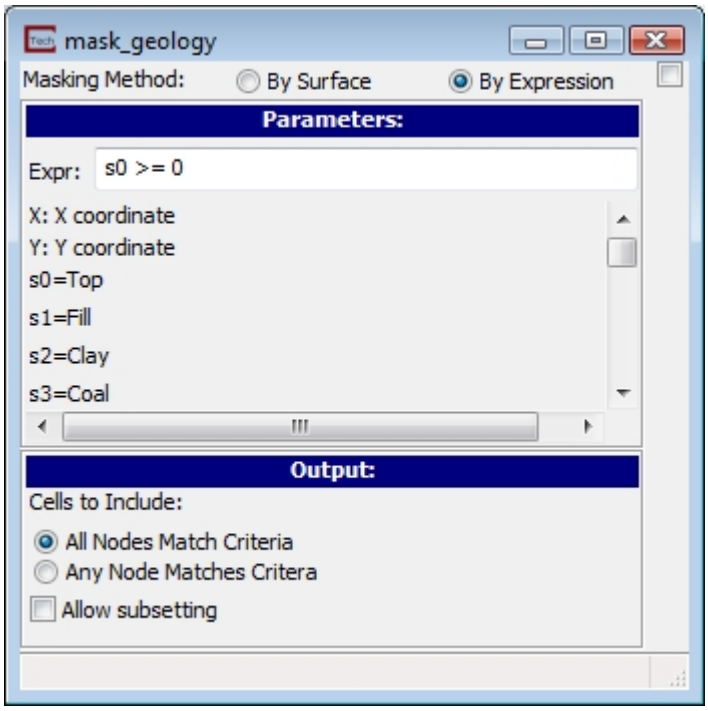
Module Output Ports

mask_geology has one output port which outputs the geologic input with the masked cells removed.



Module Control Panel

The module's control panel is shown above with the Masking Method set to "By Surface" With surface masking you can choose whether to include those areas inside or outside the cutting surface(s).



When using expressions, all nodes that meet the expression criteria are included. These options are shown above.

- The ***Masking Method*** selector chooses between surface based masking (second input port) or masking based on a mathematical expression using the coordinate values.
- The ***Nested SurfaceMasking*** toggle allows you to cut donut like shapes if you have polygons INSIDE of other polygons.
- The ***Expression*** type-in is active if "By Expression" is selected and is similar to subset_cells or data_math. An example would be: s0 >= 0
- The **Cells to Include** radio selector provides two options which depend on the Masking Method:
  - For "By Surface"
    1. *Contained by Polygon* outputs only cells which are fully inside the polygon
    2. *Intersect With or Contained by Polygon outputs cells which are fully inside the polygon and any cells which intersect the polygon (e.g. have even one node inside).*
  - For "By Expression"
    1. *All Nodes Match Criteria* outputs only cells where all nodes in the cell match the expression criteria.
    2. *Any Node Matches Criteria outputs all cells where any of the nodes in the cell match the expression criteria.*

triangulate_polygons

(This module is available only in PRO and MVS)

General Module Function

triangulate_polygons converts a closed polyline into a triangulated surface. This surface can be extruded or used by the area_cut module to perform areal subsetting of 3D models.

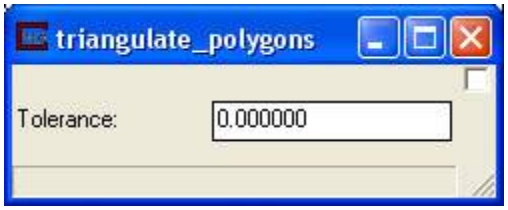
Polylines with WIDTH in AutoCAD DWG files are converted by Read_CAD into triangle strips of the specified width. As you zoom in on polylines with width, the apparent width will change, whereas the apparent width of lines DOES NOT change. However, once they are triangles, they DO NOT define a closed area and therefore would not work with triangulate_polygons.

Module Input Ports

triangulate_polygons has only one input ports which is one or more closed polylines that will be converted to triangles. triangulate_polygons does not use the convex hull of a closed polyline, it honors the full complexity.

Module Output Ports

The triangulate_polygons module has two output ports. The first output port (closest to the left) outputs an EVS field containing the triangulated surface. The second port outputs a renderable version which can connect directly to the Viewer.



Module Control Panel

The control panel has only one parameter which is the tolerance for closing polylines.

surf_cut

(This module is available only in MVS)

General Module Function

Surf_cut receives any 3D field into its left input port and it receives a surface (from scat_to_tin, geologic_surface, slice, etc.) into its right input port. Its function is similar to shape_cut. It adds a data component to the input 3D field referencing the cutting surface. With this new data component you can use a subsettting module like plume_volume to pass either side of the 3D field as defined by the cutting surface, thereby allowing cutting of structures along any surface. The surface can originate from a TIN surface, a slice plane or a geologic surface. The cutting surface can be multi-valued in Z, which means the surface can have instances where there are more one z value for

a single x, y coordinate. This might occur with a wavy fault surface that is nearly vertical, or a fault surface with recumbent folds.

Surf_cut recalculates when either input field is changed or the "Accept" button is pressed.

The general approach with surf_cut is:

Create a cutting surface representing either a fault plane, a scouring surface (unconformity), or an excavation.

Create a 3D model of the object you wish to cut.

Pass the 3D model into the left port of surf_cut, and the cutting surface to the right port of surf_cut and hit accept. Note that since this is for faulting, then the second surf_cut is needed with a translate module upstream for offsetting that fault block.

Here's a picture of the network fragment for this approach.

Use plume_volume modules to pass each side of the 3D model to downstream modules. In the case of the fault block note that one branch has been sent to a translate module for offsetting the fault block (this work MUST be done before surf_cut). The plume_volume modules subset on the surf_cut data component thus passing either the positive values or negative values (depending on whether you are sending the right or left block to be rendered).

An erosional surface can be achieved (or an excavation or unconformity), on an already cut block. In the same example, the translated fault block has another surf_cut performed, but this time using the ground surface from the original geologic input file.

Here's a picture of the same application but focused on a slightly different portion of the application for this approach. 

Now you have created a model with two blocks that are actually two versions of the same geologic model. One model is rendered with the surf_cut data component above "0.0". The other model is rendered with the surf_cut component below "0.0" AND it has been translated. But wait, we also used another surf_cut module to cut the up-thrown block with the original topographic surface.

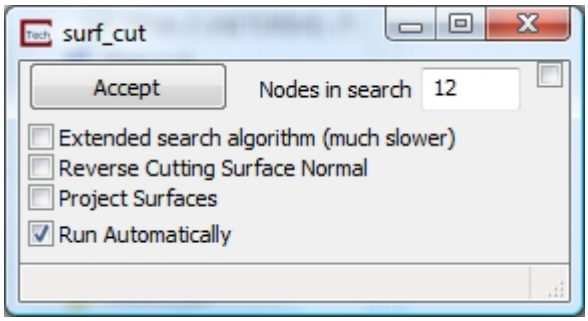
Here's a picture of model showing the two fault blocks, the fault plane and the eroded surface. 

Module Input Ports

surf_cut has two input ports. Surf_cut receives any 3D field into its left input port that should be connected to or after Explode_and_Scale. It receives a surface (from geologic_surface, scat_to_TIN or Read_UCD) into its right input port.

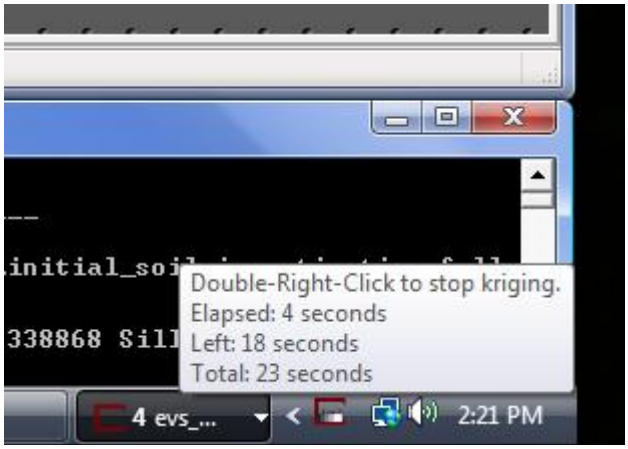
Module Output Ports

surf_cut has one output port which outputs a field with an additional "surf_cut" data component which is zero at the cutting surface, less than 0.0 on one side of the surface, and greater than 0.0 on the other side.



***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel

The module's control panel is shown above. The only type-in parameter is search level, which is set to 12.00 by default.

- The "Run" toggle controls whether the module will run when applications are loaded or data changes. When this is on, the module runs when applications are loaded or the "Accept" button is pushed. When it is off, the module will not run.
- The "Nodes in search" type_in determines the maximum number of nodes that are used in the search for the nearest node or edge when cutting through all nodes in the 3D volume. Higher values result in more accurate cutting, but longer compute times. Raise this number if any problems occur with the module's ability to cut along the surface you are using.

- The "Extended search algorithm" toggle is yet another level of accuracy and should only be chosen if the cutting process is not honoring the input surface well. It also increases compute times dramatically but should always result in equal or greater accuracy.
- The "Reverse Cutting Surface Normal" toggle flips the normal vector of the cutting surface. In other words, this changes the "sign" of the surf_cut data component. This is useful for certain applications where you want the distance from the surface to be positive on a particular side. This is most applicable when you intend to use merge_fields and downstream cut multiple surf_cut surfaces with a single plume_volume (or plume_shell) module.
- The "Project Surfaces" toggle will cause the cutting surface to be projected to the extents of the input model. This can be less accurate if there are sharp corners in the cutting surface.

NOTE: The subsetting level values are in model units. So if you type-in a value of 10.0 and your units are feet, the resulting surface will be rendered 10.0 feet away from the input surface. This may have functional use for visualizing the advancement of a pit excavation.

Related Modules

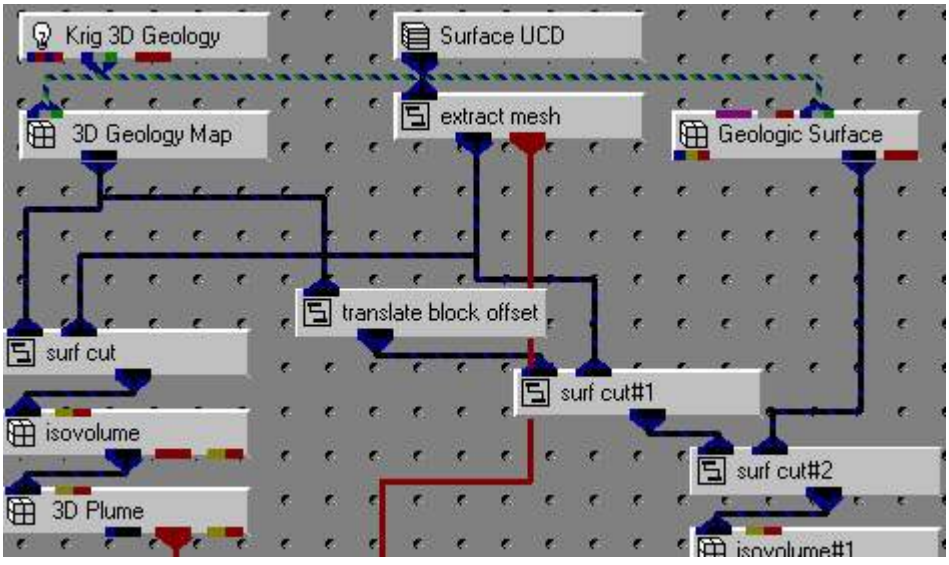
->Fence_Cut

->Tunnel_Cut

->Shape_Cut

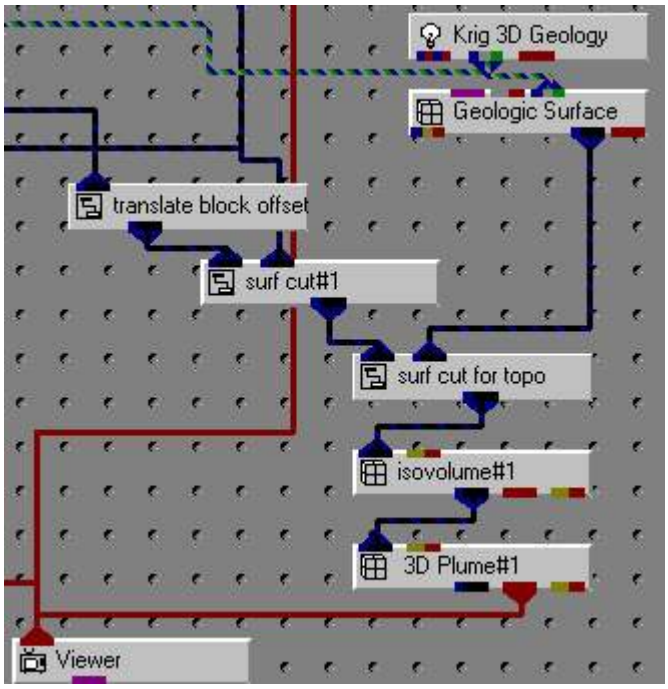
surf cut example images

Example of a network fragment for two fault blocks cut by a non-planar surface:



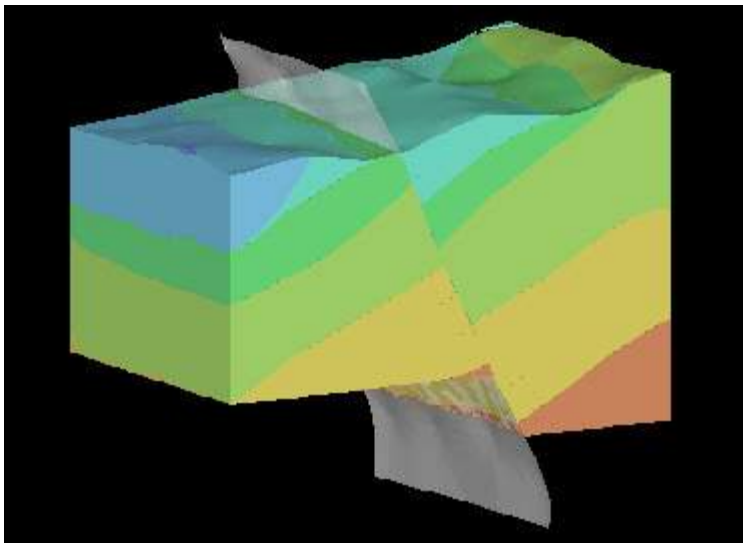
Notice that the translate module has been renamed 'translate_block_offset' for clarity.

Example of network fragment (from same application) for cutting the translated fault block with a geologic surface.



Notice that one of the surf_cut modules has been renamed 'surf_cut_for_topo' to show which surf_cut module is performing the erosional cutting (topography) vs. the surf_cutting for the fault block.

Resulting model after running the above described application.



shape_cut

General Module Function

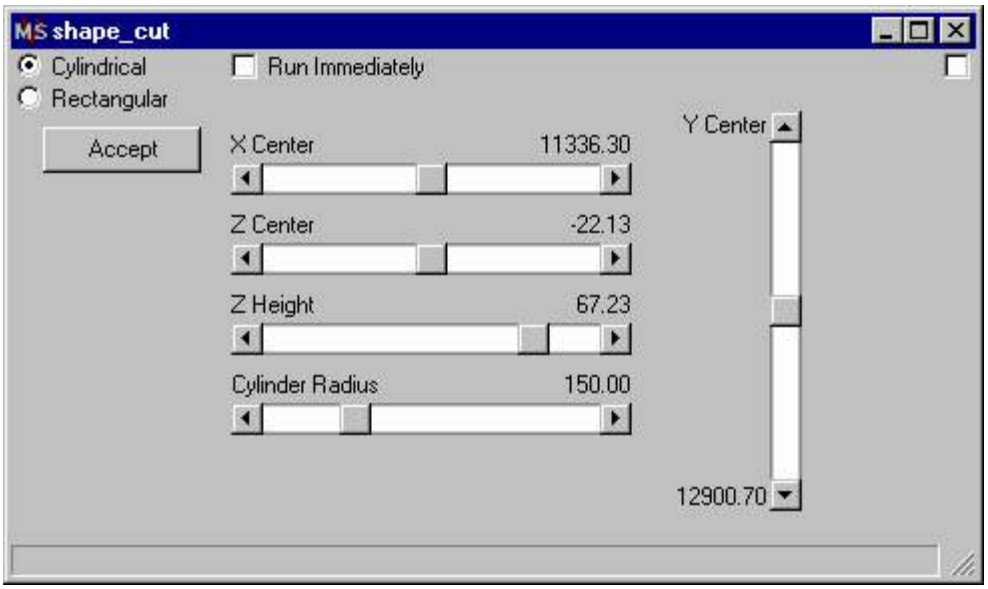
shape_cut receives any 3D field into its input port and outputs the same field with an additional data component. Using plume_shell, you can cut structures with either a cylinder or rotated rectangle. The cutting action is z invariant (like a cookie cutter). Depending on the resolution of the input field, rectangles may not have sharp corners. With rectilinear fields (and non-rotated rectangles), the threshold module can replace plume_shell to produce sharp corners (by removing whole cells). plume_volume can be used to output 3D fields for additional filtering or mapping.

Module Input Ports

shape_cut has one input port that should be connected to or after Explode_and_Scale.

Module Output Ports

shape_cut has one output port which outputs a field with an additional "shape_cut" data component which is one at the surface of the cylinder or rectangle, less than 1.0 inside and greater than 1.0 outside.



Module Control Panel

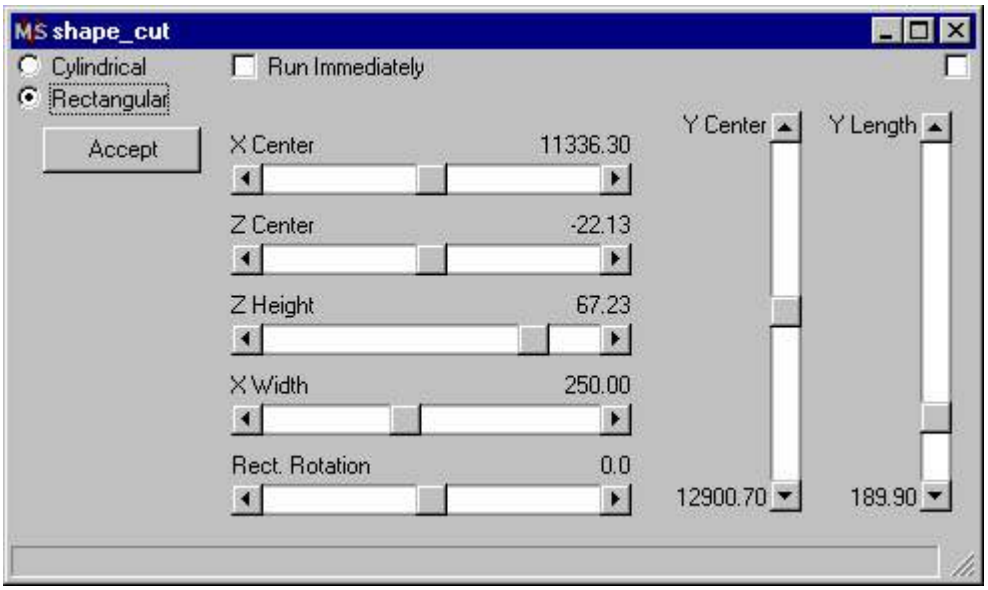
The module's control panel is shown above. Radio buttons in the upper right corner allow selection of Cylindrical or rectangular cutting. (cylindrical is shown for this example).

All parameters have been converted to sliders and there are options for immediate running and immediate mode sliders (change as you move them...not just when you release). If you have a fast computer, you can manipulate the shapes you are cutting in real time. Also, the default values for rectangle width/height and cylinder radius are based on your model (not fixed at 100.0).

The X_Center, Y_Center, Z_Center are set to the centroid of the input model the first time the module is connected to a valid field. After this initial setting, these values can be changed by the user, but will not be updated for new input data.

Both cylinders and rectangles have heights that allow cutting a rectangular or cylindrical "chunk" of a model.

If Cylindrical is chosen, a type-in field for cylinder radius is presented.



If Rectangular is chosen as shown above, type-in fields for rectangle X_Width, Y_Height and Rotation (angle measured counterclockwise in degrees from the x axis) are presented.

The Accept button must be pushed to create output.

All units are in the user's input units.

Related Modules

cut

fence_cut

General Module Function

Fence_cut receives any 3D field into its left input port and it receives polylines (from Read_DXF, isolines, Read_UCD, or other sources) into its right input port. Its function is similar to shape_cut. It adds a data component to the input 3D field and using plume_shell, you can cut structures along the path of the input polylines. Only the x and y coordinates of the polylines are used because fence_cut cuts a projected slice that is z invariant. Fence_cut recalculates when either input field is changed or the "Accept" button is pressed. The width parameter specifies the width of the fence. Fences produced with this module are not planar objects, they must be 3D. The width must be large enough to include nodes in the input 3D field. If set too narrow, the fence will be irregular or have gaps.

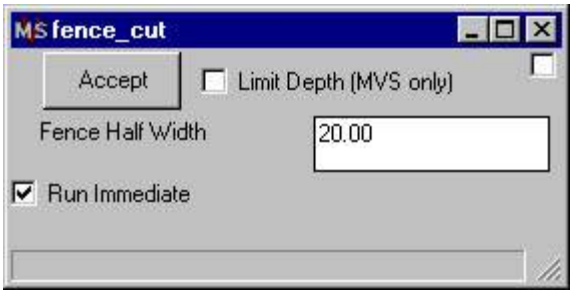
Module Input Ports

fence_cut has two input ports. Fence_cut receives any 3D field into its left input port that should be connected to or after Explode_and_Scale. It

receives polylines (from Read_DXF, isolines, Read_UCD, or other sources) into its right input port.

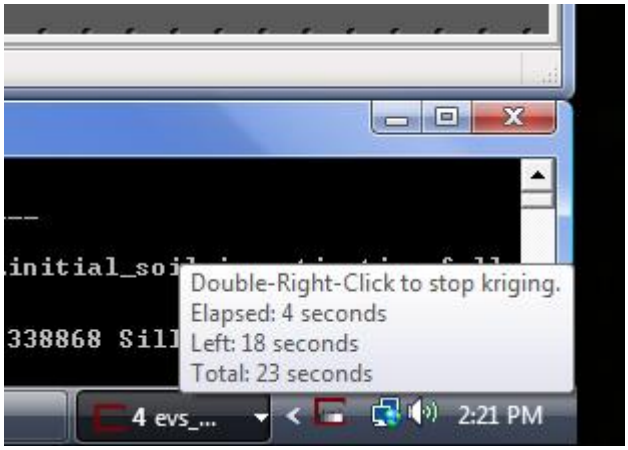
Module Output Ports

fence_cut has one output port which outputs a field with an additional "fence_cut" data component which is one at the surface of the cylinder or rectangle, less than 1.0 inside and greater than 1.0 outside.



***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel

The module's control panel is shown above. The first parameter is a toggle called Limit Depth (MVS only), which allows for setting a Zmin and Zmax thus creating (passing) a fence that only extends to those specified elevations. The Fence_Half-Width type-in controls the width of the fence_cut component as a distance from the input fence line. This parameter is set to 20.00 by default. Hint: If you desire a thinner fence but are getting breaks in the fence rendering you must krige the input mesh on a finer grid and try

again. All units are in the user's input units. The Accept button must be pushed to create output.

Related Modules

->ClickSketch

tunnel_cut

(This module is available only in MVS)

General Module Function

The tunnel_cut module is similar to the surf_cut module in that it receives any 3D field into its left input port, BUT instead of a surface, it receives a line (along the trajectory of a tunnel, boring or mineshaft) into its right input port. The tunnel_cut module then cuts a cylinder, of user defined radius, along the line trajectory. The algorithm is identical in concept to surf_cut in that it adds a data component to the input 3D field referencing the distance from the line (trajectory). With this new data component you can use a subsetting module like plume_volume to pass either portion of the 3D field (inside the cylinder or outside the cylinder), thereby allowing cutting tunnels along any trajectory. The trajectory line can originate from a DXF file or a NetCDF file.

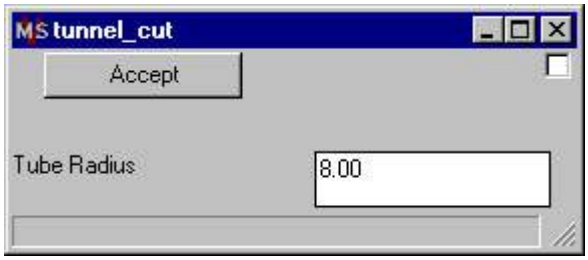
The general approach is to subset the tunnel_cut data component with either constant_shellor plume_volume. The choice of 1.0 for the subsetting level will result in cutting AT the user radius, while less than 1.0 is inside the cylinder wall and greater than 1.0 is outside the cylinder wall.

Module Input Ports

tunnel_cut has two input ports. It receives any 3D field into its left input port that should be connected to or after Explode_and_Scale. It receives polylines (from Read_DXF or Read_NetCDF) into its right input port.

Module Output Ports

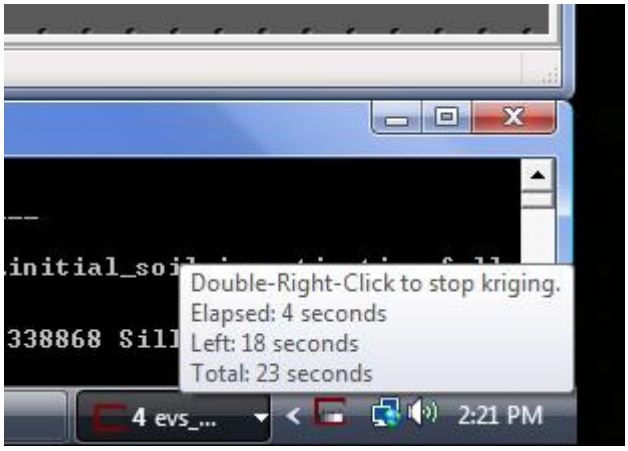
tunnel_cut has one output port which outputs a field with an additional "tunnel_cut" data component which equals 1.0 when AT the cylinder wall surface (defined by user defined radius), less than 1.0 on the inside of the cylinder, and greater than 1.0 outside the cylinder wall.



***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you

the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel

The module's control panel is shown above. The only parameter is Tube Radius, which is set to 8.00 by default. This radius is measured as a normal to the trajectory line in model units. Pressing the Accept button activates the tunnel_cut calculation and thus, passes the 3D field with the tunnel_cut data component for subsetting by modules such as plume_volume, isosurface, plume_shell, etc.

Related Modules

->fence_cut

->surf_Cut

->shape_cut

overburden

(This module is available only in MVS)

The overburden module computes the complete volume required to excavate a plume or ore body given the pit wall slope (measured from vertical) and the excavation digging accuracy (we refer to as buffer size).

overburden receives any 3D field into its input port and outputs the same field with an additional data component. Its function is similar to shape_cut, but instead involves computing a new data component based on the nodal values in the 3D field and two user defined parameter values called Wall Slope and buffer size (addressing excavation accuracy). The data component is subset according to a concentration input (based on the subsetting level you want excavated). For example, once overburden has been run for GOLD at a 45 degree pit wall slope, the user would select 45-deg:overburden_GOLD and subset all data below 1 ppm to render a 45 degree slope pit which would excavate everything higher than 1 ppm

concentration. A volumetrics calculation could be made on these criteria which would encompass the excavation and the ore body above 1 ppm.

NOTE: overburden must be placed before any scaling modules (such as Explode_and_Scale) to ensure an accurate slope angle during computations and subsequent visualizations.

Module Input Ports

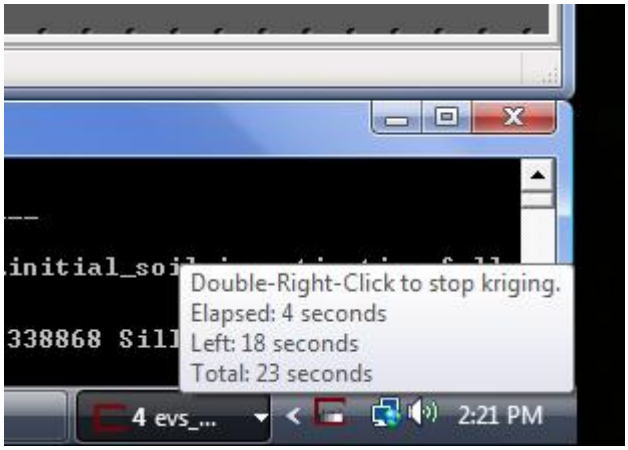
overburden has one input port which receives any 3D field.

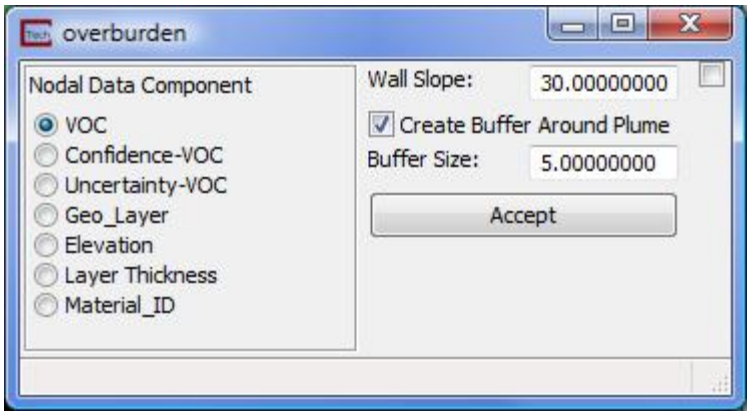
Module Output Ports

overburden has one output port which outputs a field with an additional "overburden" data component in the same units as the concentration data component, which you wish to subset on.

***Module Status*: Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.





Module Control Panel

The module's control panel is shown above. The two type-in parameters are:

Nodal Data Component -is the data on which the overburden is computed. This is the contaminant or ore that you wish to excavate.

Wall Slope - Angle measured from vertical whereby "0" is vertical and "90" is horizontal.

Create Buffer Around Plume - This toggle determines if the overburden computations are rigorous and determine the bugger on all side of the plume (ore body). If this is off, the module runs much quicker.

Buffer Size - An accuracy level resulting in the amount of excavation outside the subsetting level of interest. For example, a type-in of 10.0 would result in 10 feet of over-excavation from the subsetting level of interest.

Overburden creates a data component name that includes the wall slope, module name (including #1 or #2 if there are more than one copy in your application), and original data component (analyte) name. (i.e. 30-deg:overburden#1 of Benzene)

The overburden data component may be subset by modules such as plume_volume, isosurface, plume_shell, etc.

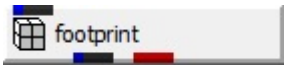
Related Modules

->fence_cut

->surf_Cut

->shape_cut

footprint



General Module Function

The footprint module is used to create the 2D footprint of a plume_shell. It creates a surface at the specified Z Position with an x-y extent that matches the 3D input. The footprint output does not contain data, but data can be mapped onto it with external kriging available in MVS.

NOTE: Do not use adaptive gridding when creating the 3D grid to be footprinted and mapping the maximum values with Krig_2D (as in the example shown below). Footprint will produce the correct area, but Krig_2D will map anomalous results when used with Krig_3D's adaptive gridding.

Module Input Ports

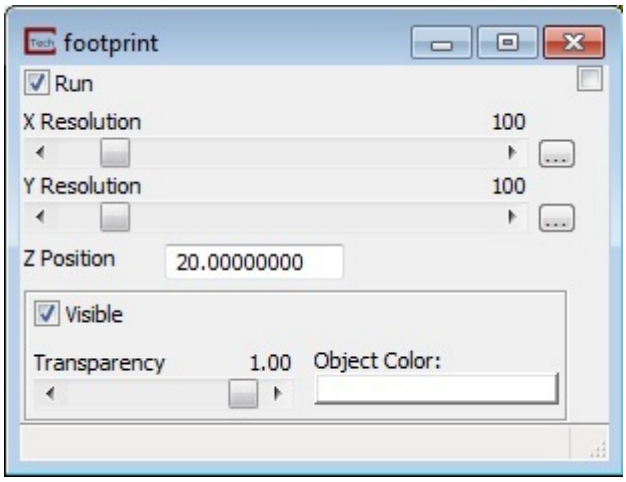
footprint has one input port that accepts a 3D mesh. Ideal input would be from the plume_shell or sequential_subset modules. The output is rasterized to the resolution specified.

Module Output Ports

footprint has two output ports.

1. The leftmost is a field with the 2D footprint.

2. The red output port is a renderable version (for connection to the Viewer) of the first port.

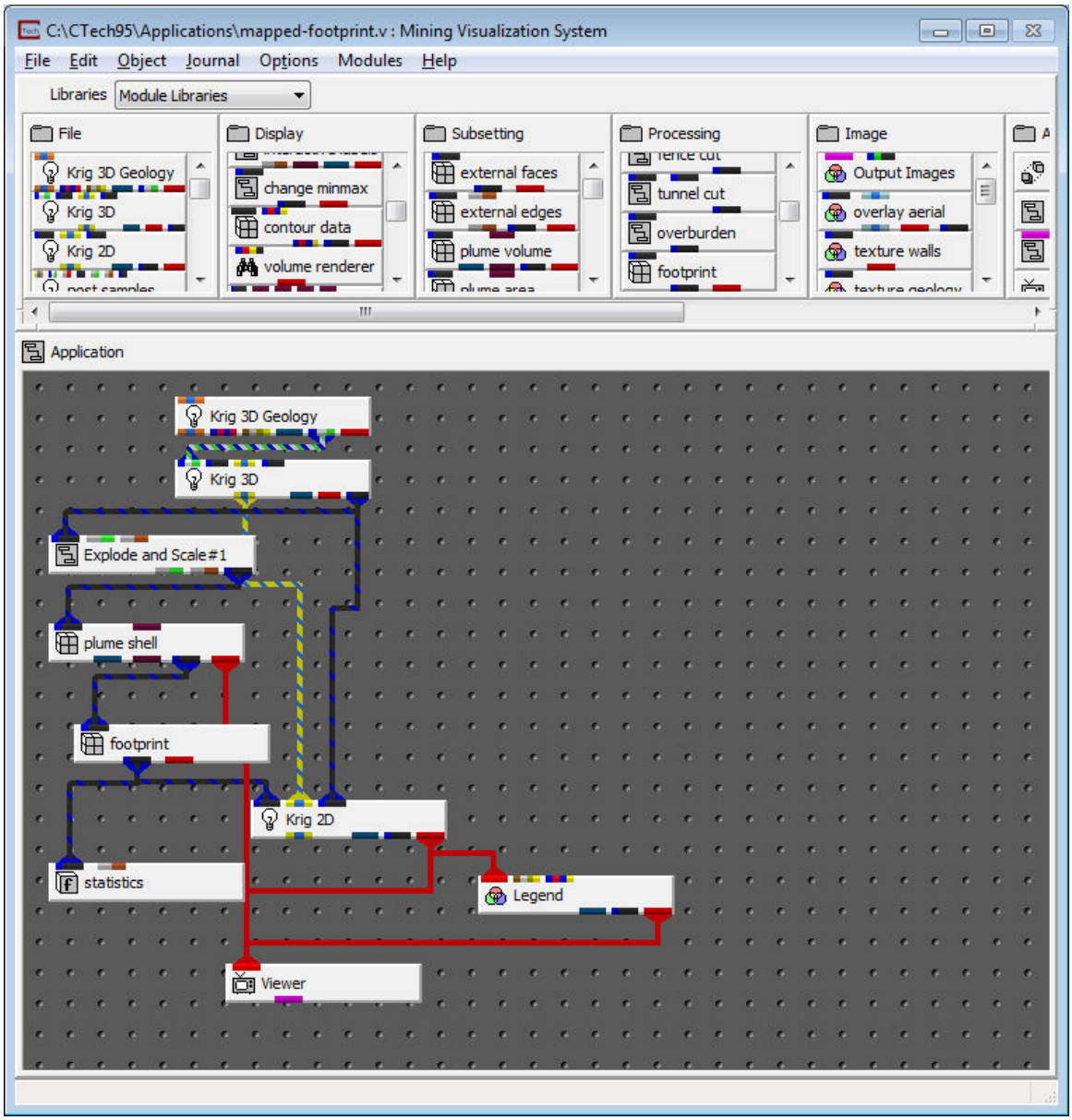


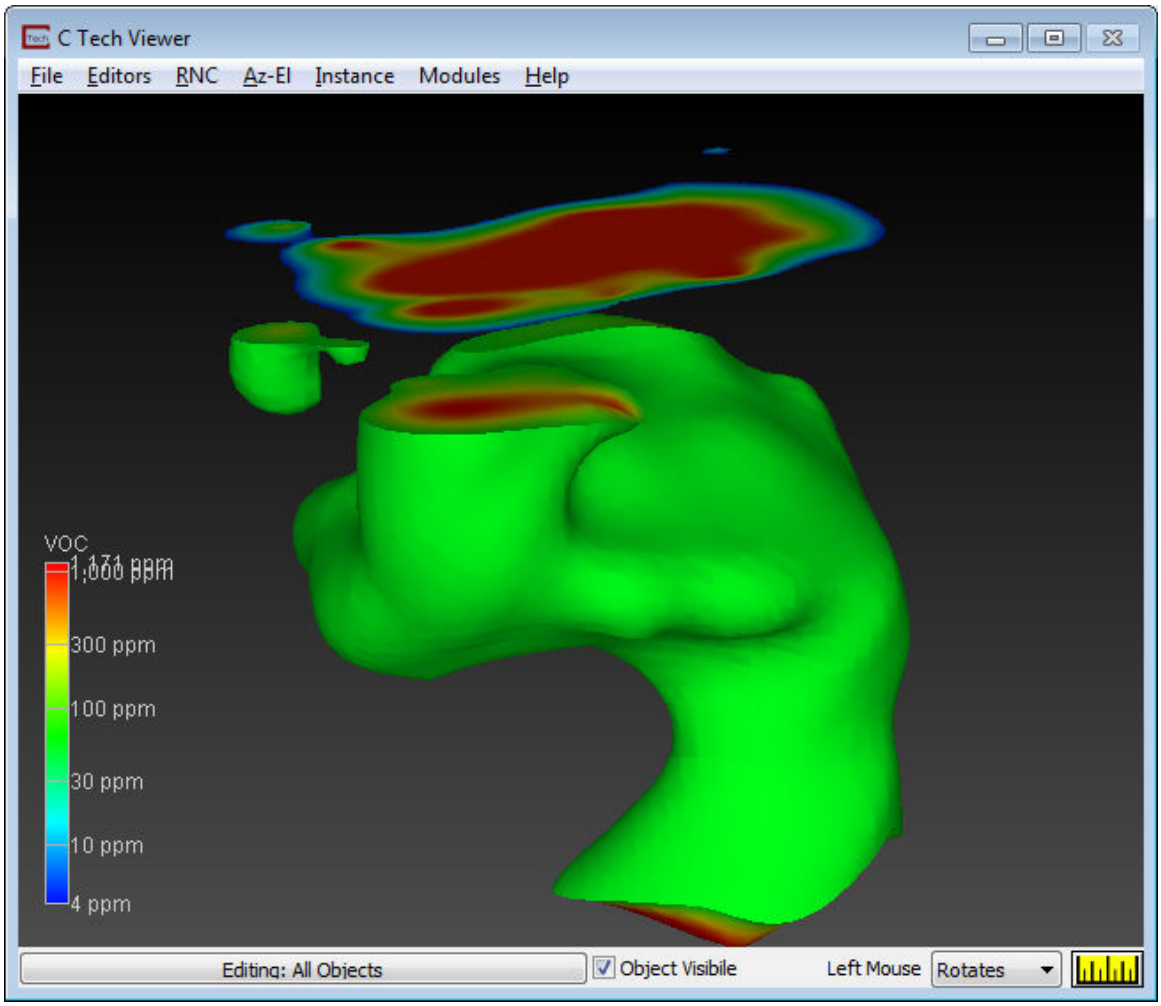
Module Control Panel

The control panel for footprint is shown in the figure above.

- The ***Run*** toggle will either allow or prevent the module from running.
- X Resolution determines the number of X cells in the output.
- Y Resolution determines the number of Y cells in the output.
- Z Position determines the Z location of the output. This allows you to position the output at the most convenient elevation.
- The ***Visible*** toggle controls whether the output is visible. Generally you will want this ON.
- The ***Transparency*** slider determines the opacity of the objects.
- ***Object_Color*** allows you to color the output any SINGLE color.

NOTE: Creating a 2D footprint with the maximum data within the plume volume mapped to each x-y location requires the external data and external gridding options in the MVS version of Krig_2D. A typical network and output is shown below.


C:\CTech95\Applications\mapped-footprint.v : Mining Visualization System
File Edit Object Journal Options Modules Help
Libraries
Module Libraries
File
Krig 3D Geology
Krig 3D
Krig 2D
Display
change minmax
contour data
volume renderer
Subsetting
external faces
external edges
plume volume
Processing
tunnel cut
overburden
footprint
Image
Output Images
overlay aerial
texture walls
Application
Krig 3D Geology
Krig 3D
Explode and Scale#1
plume shell
footprint
Krig 2D
statistics
Legend
Viewer



transform_field

General Module Function

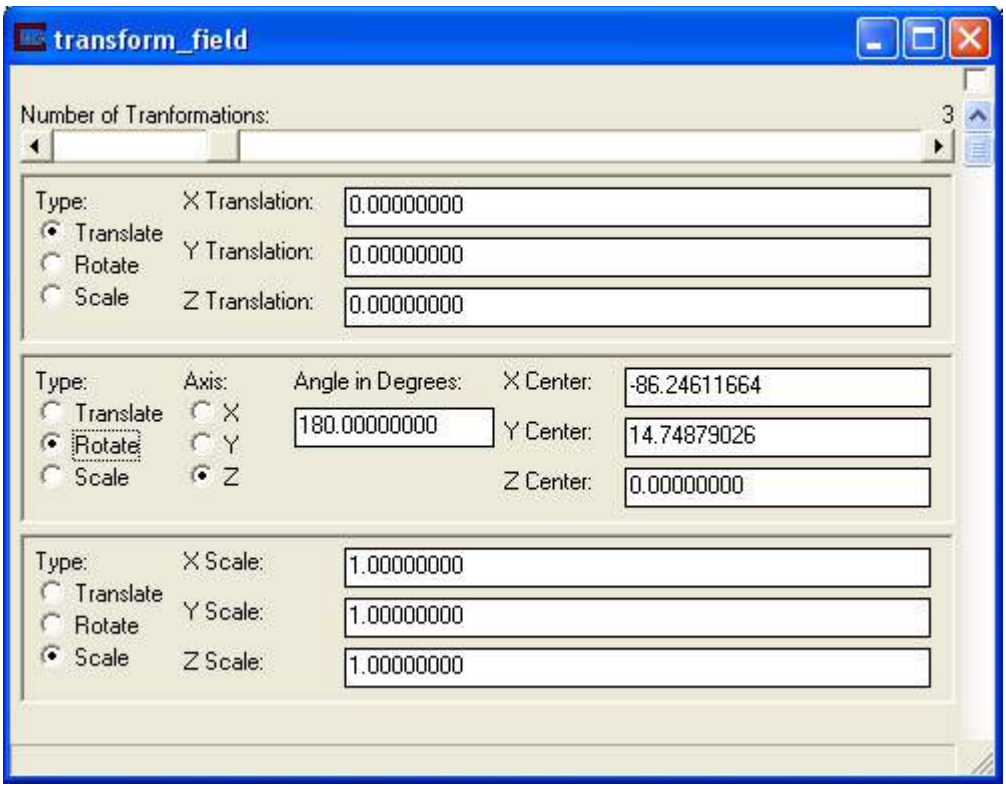
The transform_field module is used to translate, rotate or scale the coordinates any field. Uses for this module would be to rotate and translate a modflow or mt3d grid (having a grid origin of 0,0,0) to the actual coordinate system of the modeled area.

Module Input Ports

transform_field has only one input port which accepts a structured or unstructured mesh. There can be several data components, either vector or scalar, in the data field.

Module Output Ports

transform_field has two output ports. The first output port (closest to the left) outputs the same connectivity data as the input mesh with only the coordinate values adjusted to reflect the new translated coordinate information. The second output port sends a renderable object to the viewer.



Module Control Panel

The control panel for transform_field is shown in the figure above.

The first item is a slider for specifying the ***Number of Transformations*** to be performed. This can range from 1 to 12. A separate panel is provided for each transformation. The example above shows three. The content of each panel depends on the type of transformation selected. The above example shows all three types.

Translate:

Type-ins are available for adjusting the, Y, and Z translations. The default for each of these edit fields is 0.0 and the available range is unrestricted.

Rotate:

A radio-button selector allows you to choose the axis for the rotation.

A type-in allows you to specify the angle of rotation in degrees.

Type-ins are available for adjusting the, Y, and Z center for the rotation. The default for each of these edit fields is the centroid of the input and the available range is unrestricted.

Scale:

Type-ins are available for adjusting the, Y, and Z scaling factors. The default for each of these edit fields is 1.0 and the available range is unrestricted.

Related Modules

->Rotate

project_field

General Module Function

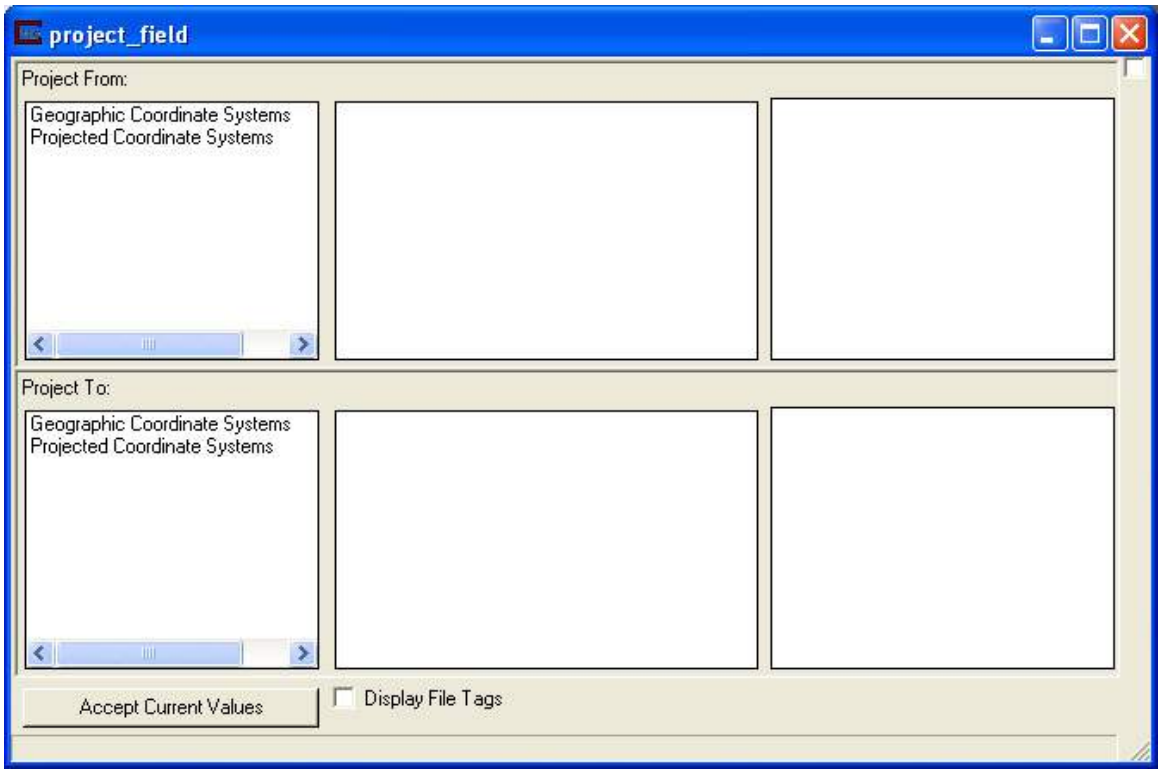
The project_field module is used to project the coordinates in any field, from one coordinate system to another.

Module Input Ports

1) in_field (Blue/Black): This port is used to import a previously created field to be projected.

Module Output Ports

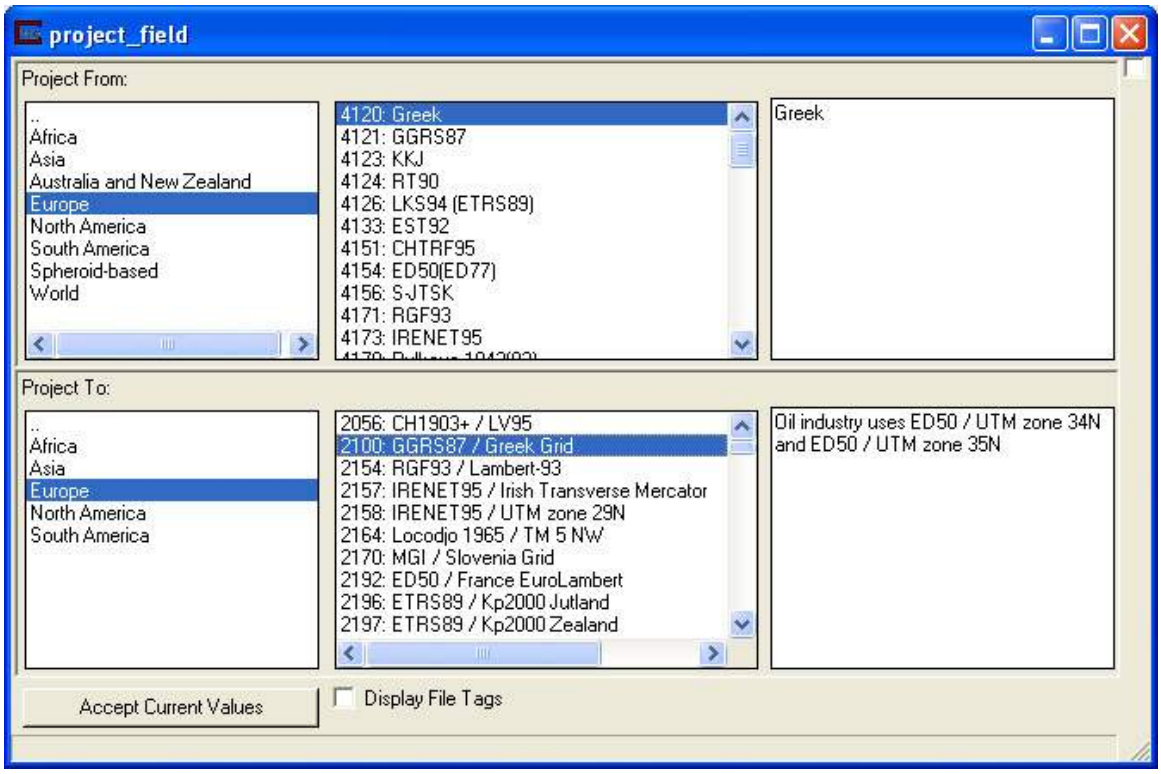
1) projected_fld (Blue/Black): This port exports the projected field.



Module Control Panel

The control panel for project_field is shown in the figure above.

Each coordinate system is divided into either Geographic or Projected coordinate systems. The coordinate system types are navigated by selecting the appropriate system type in the far left window. When a general coordinate system has been selected a specific coordinate system can be selected from the center window. If there are any details regarding the selected specific coordinate system, they will appear in the text window on the right. A specific coordinate system must be selected both to project from and to project to as in the picture below.

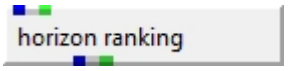


When both have been selected the Accept Current Values button must be pressed for the projection to occur.

The Display File Tags toggle is used to bring up the panel below.

This block can be copied and pasted into a file, along with the REPROJECT flag, so that the file is projected when read. The Coordinate Units text type in is used to set the UNITS flag in the projection block. Please see the help on Projecting File Coordinates for more details on how to incorporate this block in your files and for additional file flags.

horizon_ranking



General Module Function

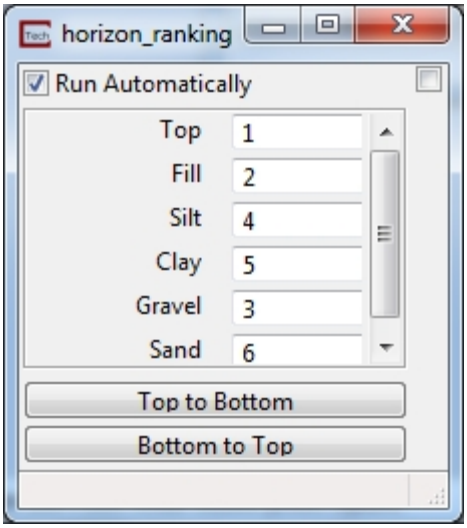
The horizon_ranking module is used to give the user control over individual surface priorities and rankings. This allows the user to fine tune their hierarchy in ways much more complex than a simple top-down or bottom-up approach.

Module Input Ports

horizon_ranking has one input port which receives geologic input from modules like Krig_3D_Geology

Module Output Ports

horizon_ranking has one output port which outputs the geologic input with re-prioritized hierarchy



Module Control Panel

The module's control panel is shown above. By default the ranking will start at 1, with increasing numbers (denoting lower ranking).

- The Top to Bottom button returns the rankings to their default values (1 on top)
- The Bottom to Top button reverses the ranking and is equivalent to reversed hierarchy

Detailed use of this module is covered in Workbook 12

material_mapping

General Module Function

This module can re-assign data corresponding to:

- Geologic Layer
- Material ID
- Indicator
- Adaptive Indicator

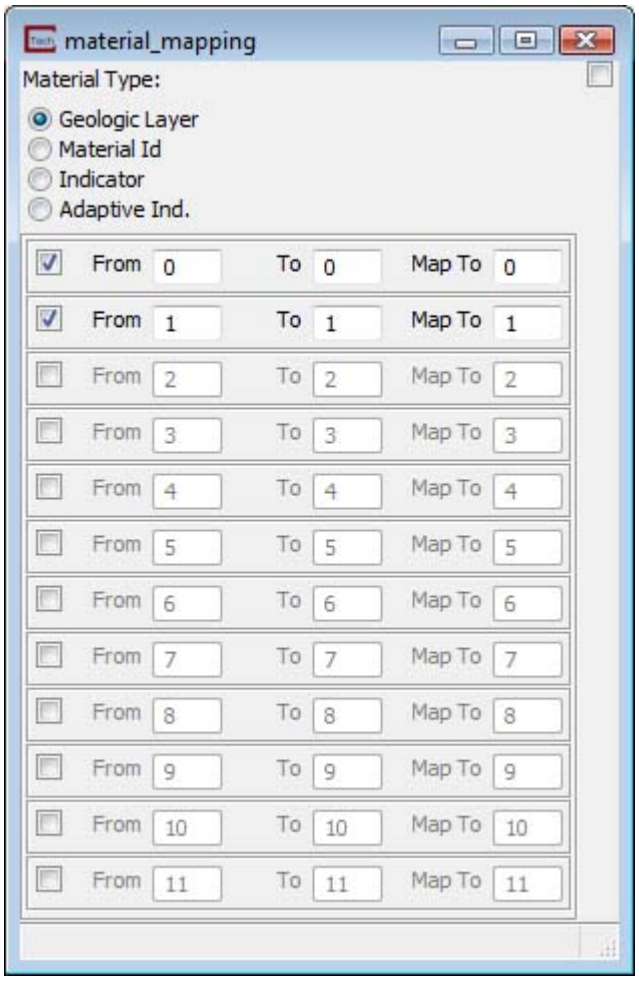
for the purpose of grouping. This provides great flexibility for exploding models or coloring.

Module Input Ports

material_mapping has one input port which accepts a field containing geologic layers, materials IDs, or other integer nodal or cell data.

Module Output Ports

material_mapping has one output port which contains the field with modified data



Module Control Panel

material_mapping has two types of controls.

1. The **Material Type** radio selector determines the type of data to be modified (remapped)
2. There are 12 groups of type-ins. Each group has:
    1. An "unnamed" ACTIVE toggle. Only groups that are toggled **on** will process your data.
    2. **From**: specifies the integer starting value for that group
    3. **To**: specifies the integer ending value for that group

4. **Map To**: specifies the integer that the range (from-to) maps to for that group

Groups are processed from Top to Bottom. You can have overlapping groups or groups whose range falls inside a previous group. In that event, the lower groups override the values mapped in a higher group.

For example, if you have ten material ids (0 through 9) and you want to have them all be 0 except for 5 & 6 which should be 1, this can be accomplished with two groups:

1. From 0 to 9 Map to 0
2. From 5 to 6 Map to 1

Please note that in the animator, you can animate these values. Each group has From, To and Map To values that are numbered zero through eleven (e.g. From0, MapTo5)

field_math

General Module Function

The field_math module (only in MVS) is used to perform mathematical operations on nodal data fields **and** coordinates. This module integrates the functionality of both data_math and coordinate_math in a single module. Furthermore, data values can be used to affect coordinates (x, y, or z) and coordinates can be used to affect data values. This makes this module **much more powerful** than data_math or coordinate_math even when both are used.

Up to four fields can be input to field_math. Mathematical expressions can involve any or all of these input fields.

Nodal data input to each of the four ports is normally scalar, however if a vector data component is used, the values in the expression are automatically the magnitude of the vector (which is a scalar). If you want a particular component of a vector, insert an extract_scalar module before connecting a vector data component to field_math. The output is always a scalar. If a data field contains more than one data component, you may select from any of them. If the ***Data*** toggle is selected your modified data will be prepended to the original data components input to the A port. If it is off, the original data components input to the A port will pass through unaffected.

Module Input Ports

field_math has eight input ports.

There are four identical input ports that can accept any type of mesh, but the meshes must have the same cell types and number of nodes (i.e., they must have identical geometries). At least one input port must be used and up to four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right.

Nodal data passed to ports one, two, three and four are referred to as Anx, Bnx, Cnx and Dnx (where x refers to the number of the data component) in the appropriate mathematical expression.

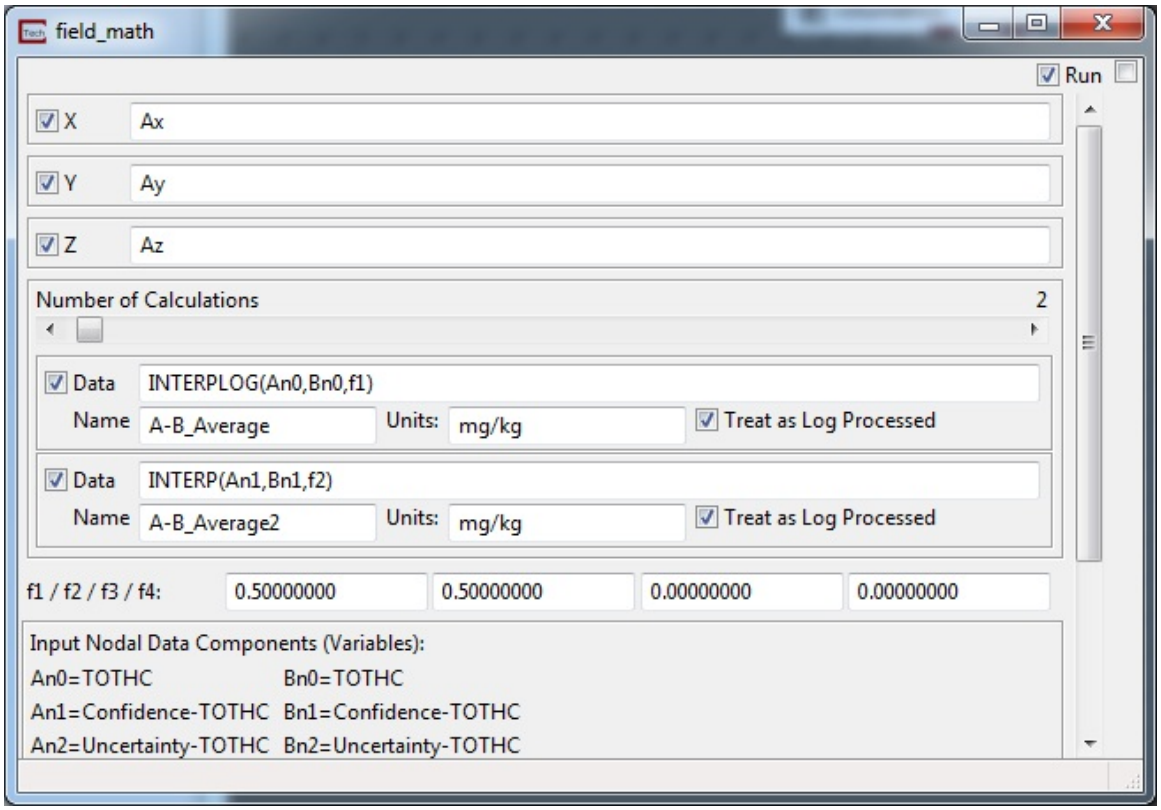
Coordinate data passed to ports one, two, three and four are referred to as Ax, Bx, Cx and Dx (y, or z) in the appropriate mathematical expression. For example if you want to refer to X coordinate of the first input you would use Ax in the expression, if you want to refer to Z coordinate of the second input you should use Bz in the expression.

For example if you want to refer to third nodal data component of the first input you would use An2 (numbers start at zero) in the expression, if you want to refer to first nodal data component of the second input you should use Bn0 in the expression.

There are also four identical input ports that accept floating point numbers (variables). None of these are required to be used, but up to four can be used. The first of these ports is the fifth from the left and the ports are numbered sequentially in ascending order to the right. These **variables** are referred to as f1, f2, f3 and f4 in your mathematical expressions.

Module Output Ports

Field_math has six output ports. The first four output port (from the left) pass the four variables so they can be shared with other modules. The next port is the output mesh containing revised data and coordinates that are a result of the computation. The last (red) output port is renderable version of the output mesh.



Module Control Panel

The control panel for field_math is shown in the figure above.

- The ***Run*** toggle will prevent the module from running when unchecked. This allows the user to make multiple changes to expressions without the module constantly running.
- The ***X, Y, & Z*** type-in boxes are for inputting mathematical expressions to modify the coordinates..

  For example, the following expressions (one for x and one for y) converts input coordinates specified in the Polar coordinate system (where the original x was radius and y was the angle in radians) into Cartesian coordinates:

  X Ax*cos(Ay)

  Y Ax*sin(Ay)

- The ***Number of Calculations*** slider lets you specify one or more new data components to create based on input mathematical expressions.
- The ***Name*** type-in box is used for defining the name of the new data being created data. It defaults to the name of the module, but allows you to give more descriptive names to the output data.
- The ***Units*** type-in box allow the units to be set for newly created data component.

- The ***Treat as Log Processed*** toggle will let modules downstream know to treat this data as log processed. This can make your network and application far more understandable.
- The ***Data*** type in boxes are for inputting mathematical expressions to create one or more new data components to create based on input mathematical expressions.

    An example of an equation to blend the values of (interpolate between) two different fields having nodal data with a logarithmic distributions as f1 ranges from 1 to 0.0 is:

    log10(pow(10,An0)*f1+pow(10,Bn0)*(1.0-f1))

    You can also use

    interplog(An0, Bn0,f1)

    To perform interpolation between non-log processed data use:

    An0 * f1 + Bn0 * (1.0-f1) OR interp(An0, Bn0,f1)

Pop-UpAvailable Mathematical Operators here.

Jump to a list of availableMathematical Operators

Related Modules

->coordinate_math

data_math

General Module Function

The data_math module is used to perform mathematical operations on nodal data fields **and** coordinates. Up to four fields can be input to data_math. Mathematical expressions can involve any or all of these input fields. If a data field contains more than one data component, you may select from any of them.

Nodal data input to each of the four ports is normally scalar, however if a vector data component is used, the values in the expression are automatically the magnitude of the vector (which is a scalar). If you want a particular component of a vector, insert an extract_scalar module before connecting a vector data component to data_math. The output is always a scalar. If a data field contains more than one data component, you may select from any of them. If the ***Data*** toggle is selected your modified data will be prepended to the original data components input to the A port. If it is off, the original data components input to the A port will pass through unaffected.

Module Input Ports

data_math has eight input ports.

There are four identical input ports that can accept any type of mesh, but the meshes must have the same cell types and number of nodes (i.e., they must have identical geometries). At least one input port must be used and up to

four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right.

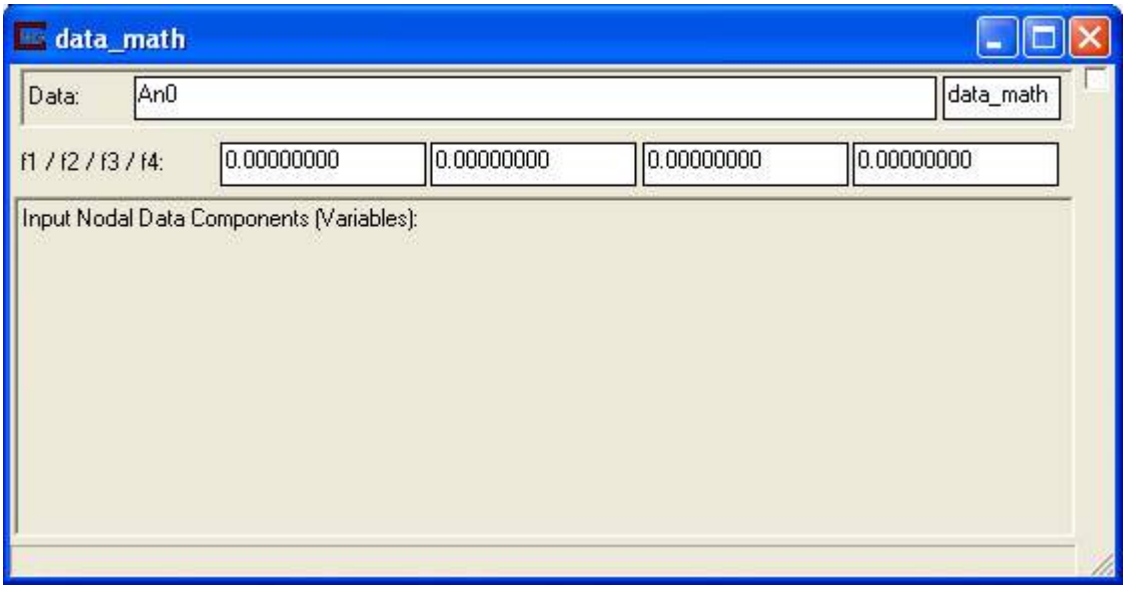
Nodal data passed to ports one, two, three and four are referred to as Anx, Bnx, Cnx and Dnx (where x refers to the number of the data component) in the appropriate mathematical expression.

For example if you want to refer to third nodal data component of the first input you would use An2 (numbers start at zero) in the expression, if you want to refer to first nodal data component of the second input you should use Bn0 in the expression.

There are also four identical input ports that accept floating point numbers (variables). None of these are required to be used, but up to four can be used. The first of these ports is the fifth from the left and the ports are numbered sequentially in ascending order to the right. These **variables** are referred to as f1, f2, f3 and f4 in your mathematical expressions.

Module Output Ports

Data_math has six output ports. The first four output port (from the left) pass the four variables so they can be shared with other modules. The next port is the output mesh containing revised data and coordinates that are a result of the computation. The last (red) output port is renderable version of the output mesh.



Module Control Panel

The control panel for data_math is shown in the figure above. The last type-in box in the Data row is for setting the data component name. This can make your network and application far more understandable, especially if you are dealing with multiple data_math modules and recombining them later. The other type-in boxes are for inputting mathematical expressions.

An example of an equation to blend the values of (interpolate between) two different fields having nodal data with a logarithmic distributions as f1 ranges from 1 to 0.0 is:

log10(pow(10,An0)*f1+pow(10,Bn0)*(1.0-f1))

You can also use

interplog(An0, Bn0,f1)

To perform interpolation between non-log processed data use:

An0 * f1 + Bn0 * (1.0-f1) OR interp(An0, Bn0,f1)

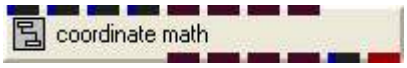
Pop-UpAvailable Mathematical Operators here.

Jump to a list of availableMathematical Operators

Related Modules

->coordinate_math

coordinate_math



General Module Function

The coordinate math module (This module is available only inEVS PRO and MVS) is used to perform mathematical operations on any EVS mesh surface or 3D mesh. Up to four data fields can be input to coordinate_math. Mathematical expressions can involve any or all of these input fields. Some valuable uses of coordinate_math are for applying transformations of the z coordinates of a grid (such as a water table surface elevation), or for applying a mathematical operation on a surface (such as a fault surface displacement , rotation, etc.).

Module Input Ports

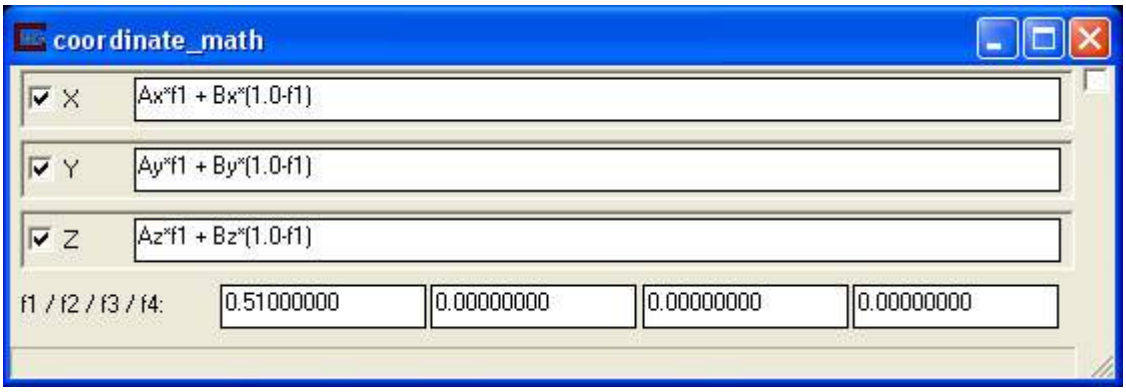
coordinate_math has eight input ports.

There are four identical input ports that can accept any type of mesh, but the meshes must have the same cell types and number of nodes (i.e., they must have identical geometries). At least one input port must be used and up to four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right. Coordinate data passed to ports one, two, three and four are referred to as Ax, Bx, Cx and Dx (y, or z) in the appropriate mathematical expression. For example if you want to refer to X coordinate of the first input you would use Ax in the expression, if you want to refer to Z coordinate of the second input you should use Bz in the expression.

There are also four identical input ports that accept floating point numbers (variables). None of these are required to be used, but up to four can be used. The first of these ports is the fifth from the left and the ports are numbered sequentially in ascending order to the right. These **variables** are referred to as f1, f2, f3 and f4 in your mathematical expressions.

Module Output Ports

coordinate_math has six output ports. The first four output port (from the left) pass the four variables so they can be shared with other modules. The next port is the output mesh containing revised coordinates that are a result of the computation. The last (red) output port is renderable version of the output mesh.



Module Control Panel

The control panel for field math is shown in the figure above. The type-in boxes are for inputting mathematical expressions. For example, the following expression converts input coordinates specified in the Polar coordinate system (where the original x was radius and y was the angle in radians) into Cartesian coordinates:

X Ax*cos(Ay)

Y Ax*sin(Ay)

Pop-UpAvailable Mathematical Operators here.

Jump to a list of availableMathematical Operators

Related Modules

->field_math

tubes

General Module Function

The tubes module is used to produce open or closed tubes of constant or data dependent radius using 3D lines or polylines as input.

(This feature available only in EVS PRO and MVS)

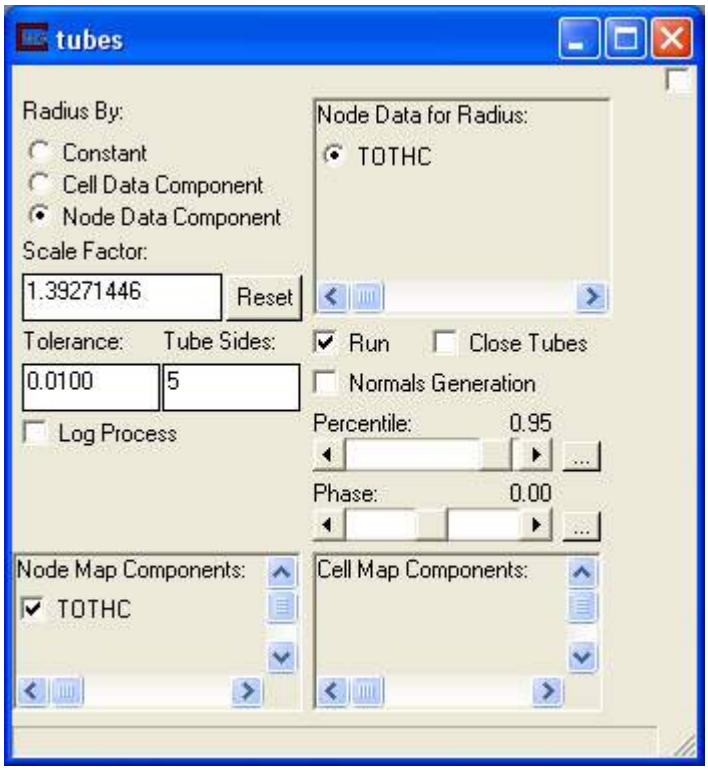
Module Input Ports

Tubes has one input port that accepts the 3D lines or polylines as input.

Module Output Ports

Tubes has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains the tubes. The second port outputs a renderable geometry, which can be connected directly to the Viewer.

Module Control Panel



The control panel for tubes is shown in the figure above.

Radius By: radio buttons allow you to choose constant or data dependent radius scaling.

Node Data for Radius: radio buttons allow you to choose the nodal data component for scaling.

Cell Data for Radius: radio buttons allow you to choose the cell data component for scaling.

The **Scale Factor:** parameter is multiplied by the nodal data to determine the radius.

The **Reset** button causes the automatic computation of scale factor to be performed. This is useful if you have changed the scale or the nodal/cell data used.

The **Percentile** slider allows you to control the automatic scaling of tubes based on the nth percentile value (versus the maximum 100th%). This addresses datasets where there are only a few nodes with extremely high values.

The **Tolerance:** parameter determines the maximum distance between nodes of adjacent line segments before those line's nodes should be merged to form a polyline. This also applies to closed polyline contours. Closed polylines will have their starting and ending nodes merged to form closed (torroidal) annuli.

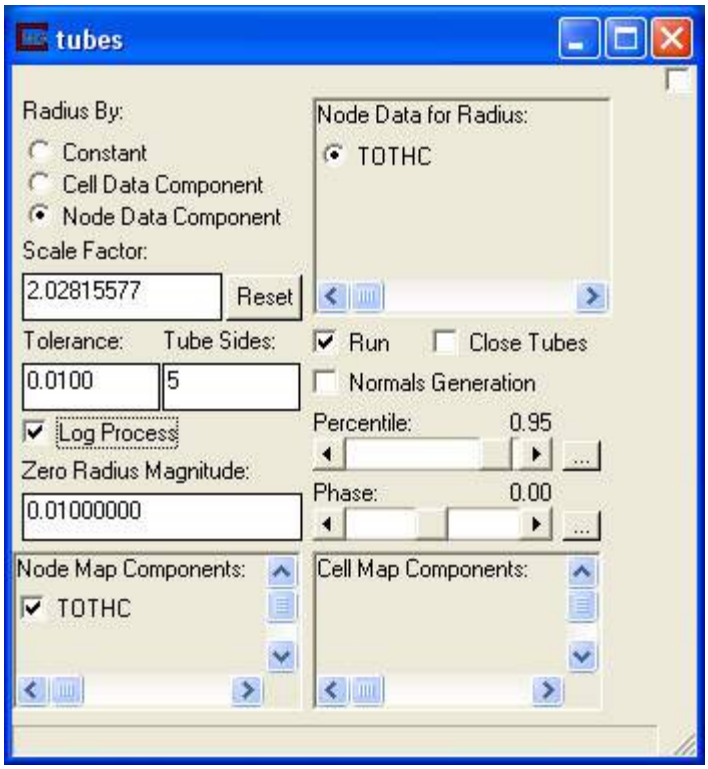
The **Tube Sides:** parameter is The number of faces on the tubes. (the default is 6)

The **Run** toggle causes the module to run whenever parameters or inputs change.

The **Close Tubes** toggle causes the tubes to have solid ends. However the tubes are not solid objects (they are hollow).

The **Normals Generation** toggle turns on Normals Generation for rendering the tubes. This makes their faces more distinct (less smoothed). If you want the tubes to represent cylinders, this should be off.

The **Log Process** toggle causes the data to be log scaled before it is used for radius scaling. This parameter causes the Zero Radius Magnitude parameter to become active also as shown in the figure below.



The **Zero Radius Magnitude** parameter is required when using log processing. Since you cannot take the log of zero (or negative values) you must set the value (from inside or outside of your range of data values) that will correspond to a zero radius.

The **Phase** slider allows you to control the apparent rotation of tubes. A phase of 0.5 rotates 180 degrees.

The**Node Map Components**check boxes allow you to select those nodal data components that are used for coloring or subsequent subsetting operations.

The**Cell Map Components**check boxes allow you to select those cell data components that are used for coloring or subsequent subsetting operations.

cross_section_tubes

General Module Function

The cross_section_tubes module is used to produce open or closed tubes of user defined cross-section and constant or data dependent radius using 3D lines or polylines as input for the centerline and a single 2D polyline as the cross-section of the tubes..

(This feature available only in MVS)

Module Input Ports

Tubes has two input ports.

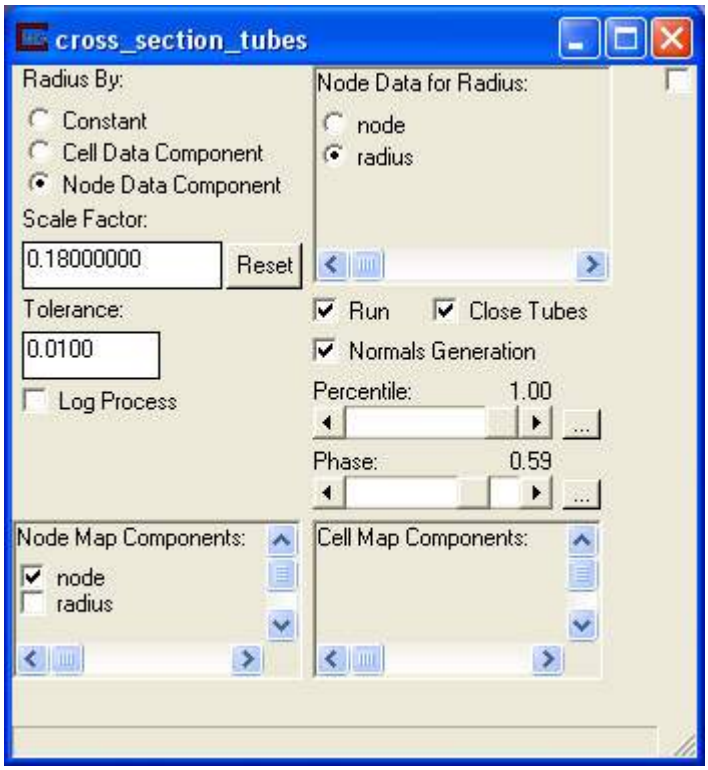
The left port accepts the 3D lines or polylines as input for the centerlines of the tubes

The right port accepts a single 2D polylines as input for the cross-section of the tubes.

Module Output Ports

cross_section_tubes has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains the tubes. The second port outputs a renderable geometry, which can be connected directly to the Viewer.

Module Control Panel



The control panel for cross_section_tubes is shown in the figure above.

Radius By: radio buttons allow you to choose constant or data dependent radius scaling.

Node Data for Radius: radio buttons allow you to choose the nodal data component for scaling.

Cell Data for Radius: radio buttons allow you to choose the cell data component for scaling.

The **Scale Factor:** parameter is multiplied by the nodal data to determine the radius.

The **Reset** button causes the automatic computation of scale factor to be performed. This is useful if you have changed the scale or the nodal/cell data used.

The **Percentile** slider allows you to control the automatic scaling of tubes based on the nth percentile value (versus the maximum 100th%). This addresses datasets where there are only a few nodes with extremely high values.

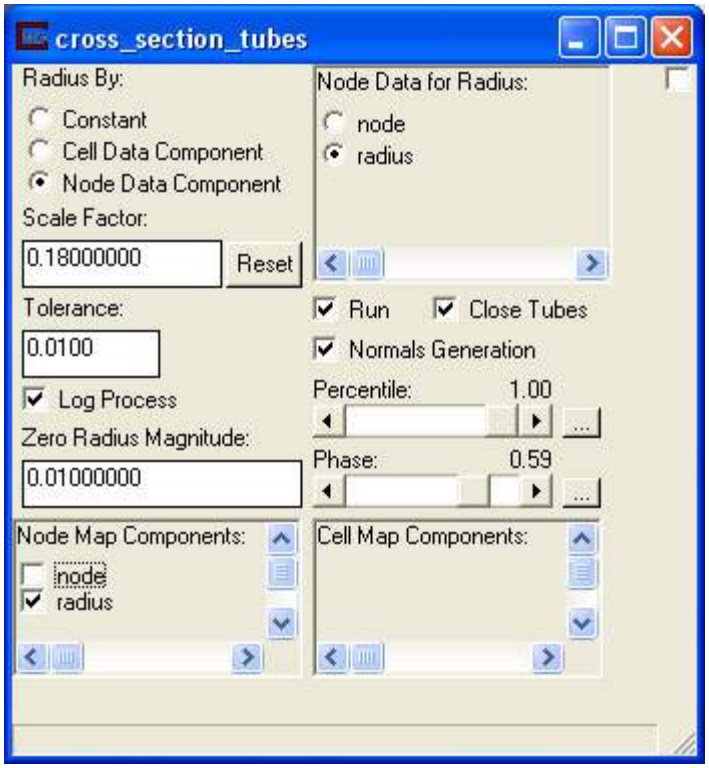
The **Tolerance:** parameter determines the maximum distance between nodes of adjacent line segments before those line's nodes should be merged to form a polyline. This also applies to closed polyline contours. Closed polylines will have their starting and ending nodes merged to form closed (torroidal) annuli.

The **Run** toggle causes the module to run whenever parameters or inputs change.

The **Close Tubes** toggle causes the tubes to have solid ends. However the tubes are not solid objects (they are hollow).

The **Normals Generation** toggle turns on Normals Generation for rendering the tubes. This makes their faces more distinct (less smoothed). If you want the tubes to represent cylinders, this should be off.

The **Log Process** toggle causes the data to be log scaled before it is used for radius scaling. This parameter causes the Zero Radius Magnitude parameter to become active also as shown in the figure below.



The **Zero Radius Magnitude** parameter is required when using log processing. Since you cannot take the log of zero (or negative values) you must set the value (from inside or outside of your range of data values) that will correspond to a zero radius.

The **Phase** slider allows you to control the apparent rotation of tubes. A phase of 0.5 rotates one-half of one face.

The**Node Map Components**check boxes allow you to select those nodal data components that are used for coloring or subsequent subsetting operations.

The**Cell Map Components**check boxes allow you to select those cell data components that are used for coloring or subsequent subsetting operations.

clamp

This module is deprecated and its functionality has been included in change_minmax

General Module Function

The clamp module is used to set the upper and/or lower bounds on a data component. Clamp modifies the nodal data in a mesh by outputting values ranging from the Min Value to the Max Value, with the Max Value assigned to all data values exceeding the Max Value, and the Min Value assigned to all data values below the Min Value. The range of the min and max values can also be reset by clamp, so that color scales used downstream of clamp will only reflect the clamped range. As of version 6.2, clamp includes all functionality that was previously in set_minmax also.

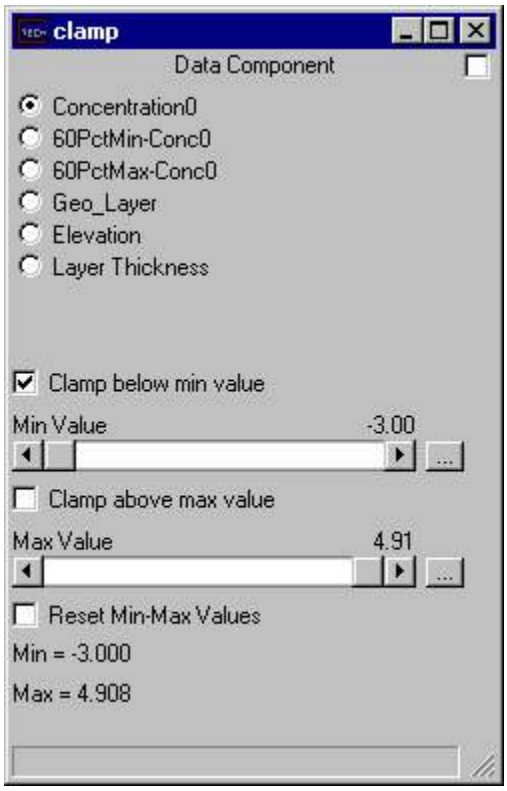
Clamp outputs all of the original data components preceded by a modified version of the selected component.

Module Input Ports

Clamp has only one input port. Input piped to this port must contain nodal data. If mesh data is piped to this port it will pass through clamp unchanged. The nodal data components can be scalar or vector.

Module Output Ports

Clamp has two output ports. The first output port (closest to the left) outputs a new nodal data set containing only the selected data component subjected to the clamping criteria. If a mesh is passed to the threshold input it will pass to the output port unchanged. The second port outputs a renderable geometry if a mesh is present.



Module Control Panel

The control panel for clamp is shown in the figure above.

The data component radio buttons determine which data component will be considered in clamp.

The *Clamp above max value* and *Clamp below min value* check boxes determine if the data will be clamped above the maximum value selected and/or below the minimum value selected.

The Reset min-max tells clamp to send the clamped values as the min and max values to modules downstream. If this box is not checked, the original min and max of the data will be passed downstream, but the actual nodal data will be within the clamp range specified.

The Min Value slider determines the minimum clamp value and the Max Value slider determines the maximum clamp value. By clicking on the small button with "" you can type in the value rather than use the sliders.

The data component value determines which data component is to be modified by clamp. By default, the first (0th) component is selected. When a data component is selected under data component, it's nodal data values are checked against the Min Value and Max Value. Then, if the Above Max Value check box is selected, any nodal data value above the Max Value will be assigned the Max Value. Similarly, if the Below Min Value check box is selected, any nodal data value below the Min value will be assigned the Min Value.

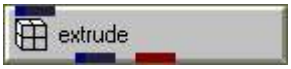
The Above Max Value and Below Min value check boxes determine whether the data is to be clamped at the selected Max Value and/or at the Min Value.

By default, the Above Max value is off (not checked) and the Below Min Value is on (checked). If both the Above Max Value and Below Min Value check boxes are selected, the resulting output is the data component subjected to both the Min Value and Max Value criteria. If neither of the check boxes are selected, the resulting output is the data component subjected to no clamping. The Reset min-max box is checked to redefine the total range of the data for all downstream modules to the min-max values specified,.

Related Modules

threshold

extrude



General Module Function

The extrude module accepts any mesh and adds one to the dimensionality of the input by extruding the mesh in the Z direction. The interface enables changing the height scale for extruded cells and extruding by a constant, any nodal or cell data component. This module is often used with the Read_Shapefile module to convert polygonal shapefiles into extruded volumetric cells.

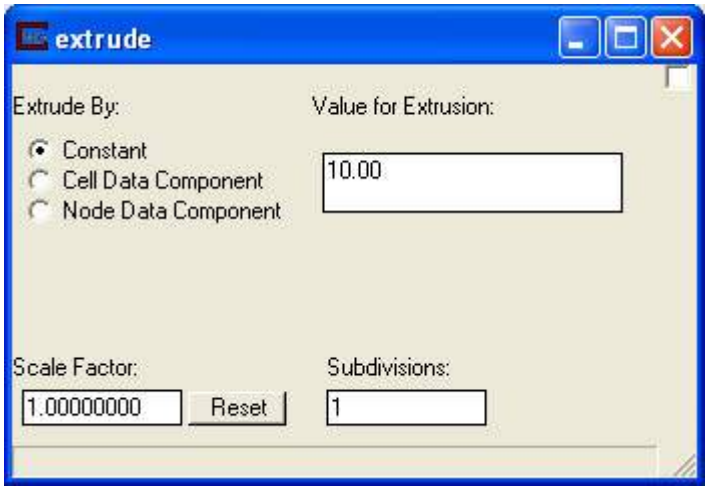
Module Input Ports

The extrude module has a single input port that accepts any non-volumetric mesh. Point, line, or polygonal input is accepted, as well as any number of mesh or cell data components.

Module Output Ports

The extrude module has two output ports. The blue/black output port is the new mesh which will have a dimensionality one greater than the input. The new mesh will include all of the nodal and cell data in the input field. The red port outputs a renderable object which will be colored by the first nodal data component in the input field, the first cell data component if the input field has no nodal data, or a constant color if the input mesh has no data components.

Module Control Panel

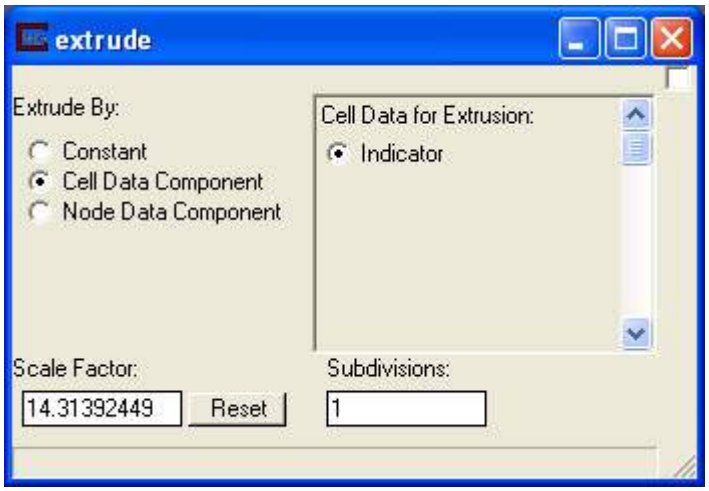


The control panel for extrude is shown above. The "Extrude By" radio buttons control the method of extrusion. The "Node/Cell Data for Extrusion" radio buttons determine which data component will be used. The output will contain all data components.

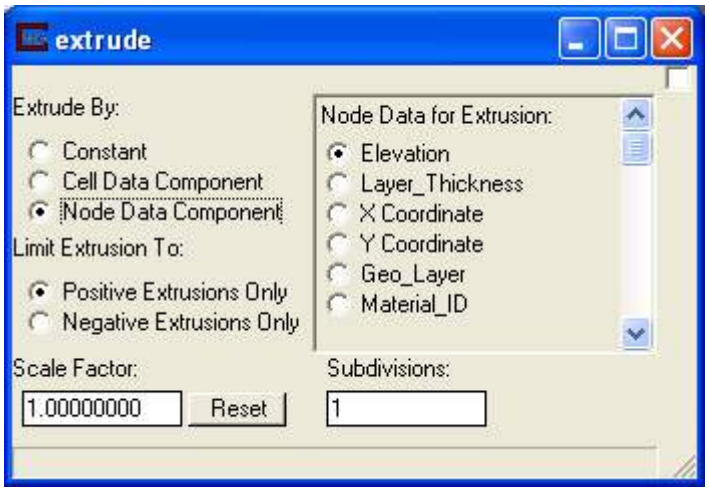
The Scale Factor is multiplied by the data value to determine the extrude height.

The Subdivisions field determines how many cells vertically to create.

When Constant is selected, the cells in the output mesh will be extruded by the "Value for Extrusion" times the "Scale Factor".



When Cell Data Component is chosen, the output cells will be extruded by the Scale Factor times the value of whichever cell data component is selected on the right.



When Node Data Component is chosen, the output cells will be extruded by the Scale Factor times the value of whichever nodal data component is selected on the right. With nodal data extrusion you must select "Positive Extrusions Only" or "Negative Extrusions Only". Since each node of a triangle or quadrilateral can have different values, it is possible for a single cell to have both positive and negative data values at its nodes. If this type of cell is extruded both directions, the cell topology can become tangled.

For this reason, nodal data extrusions must be limited to one direction. To extrude in both directions, merely use two extrude modules in parallel, one set to positive and the other to negative.

Related Modules

data_translate

data_translate

General Module Function

The data_translate module accepts **nearly** any mesh and translates the grid in x, y, or z based upon either a nodal or cell data component or a constant.

The interface enables changing the Scale Factor for z translates to accommodate an overall z exaggeration in your applications. This module is most useful when used with the Read_Shapefile module to properly place polygonal shapefile cells at the proper elevation.

Module Input Ports

The data_translate module has two input ports. The leftmost port accepts the Z Exaggeration factor from modules such as Explode_and_Scale.

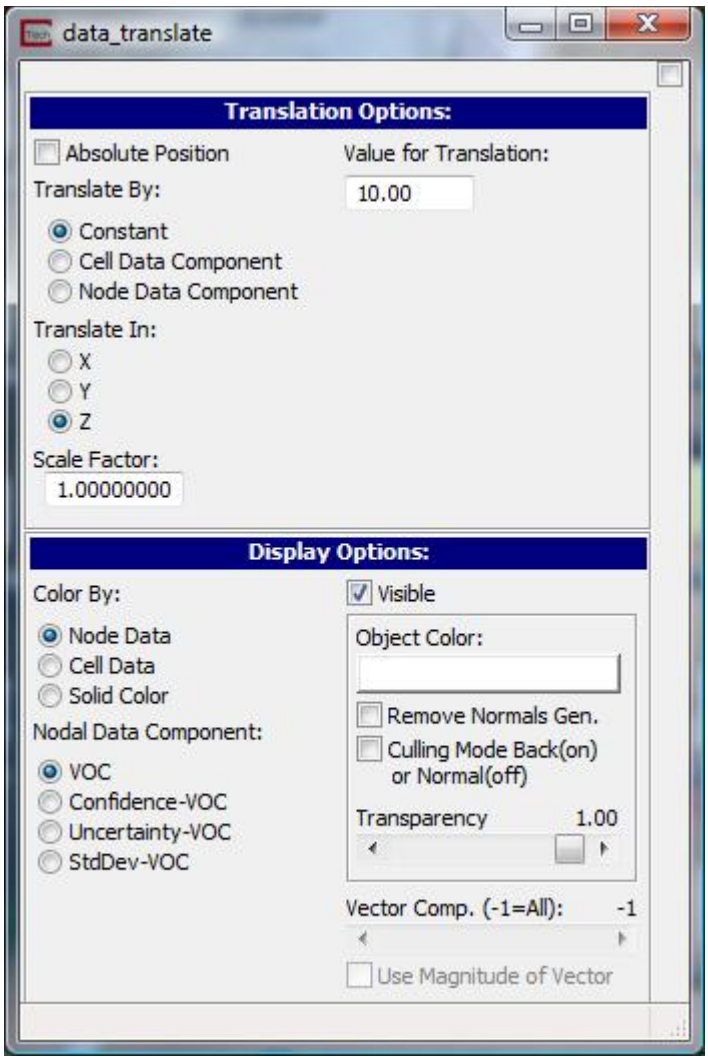
The right blue/black port accepts any mesh to be used with cell data or any non-volumetric mesh to be used with nodal data. Volumetric cells cannot be translated by their nodal data because cell topology can easily be corrupted.

Module Output Ports

The data_translate module has three output ports. The leftmost port outputs the Z Exaggeration factor to modules such as Explode_and_Scale.

The center blue/black output port is the new mesh which will have the translated version of the input. The new mesh will include all of the nodal and cell data in the input field. The red port outputs a renderable object which will be colored by the first nodal data component in the input field, the first cell data component if the input field has no nodal data, or a constant color if the input mesh has no data components.

Module Control Panel



The control panel for data_translate is shown above and below.

The **Translate By:** radio buttons control the method of translation. Depending on the method selected, the options in the module (and the appearance of the panel) vary.

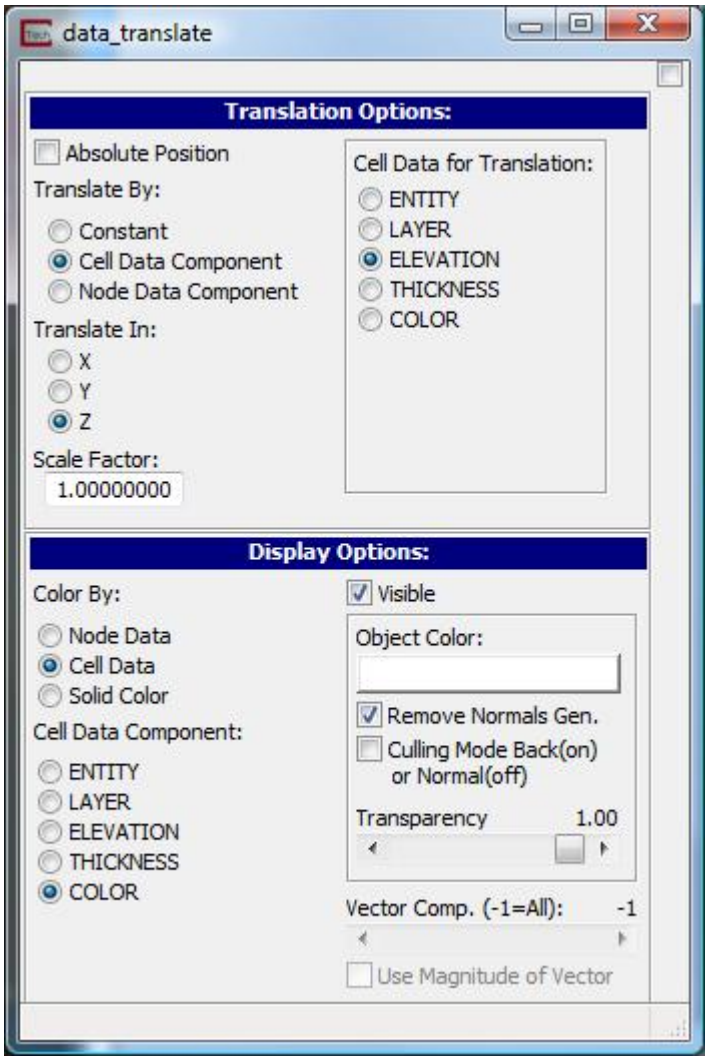
The **Translate In:** radio buttons determine the axis of translation.

Warning: The scale factor is always applied. If translating along any axis other than z, it is unlikely that you want to use the Z Exaggeration factor used elsewhere in your application.

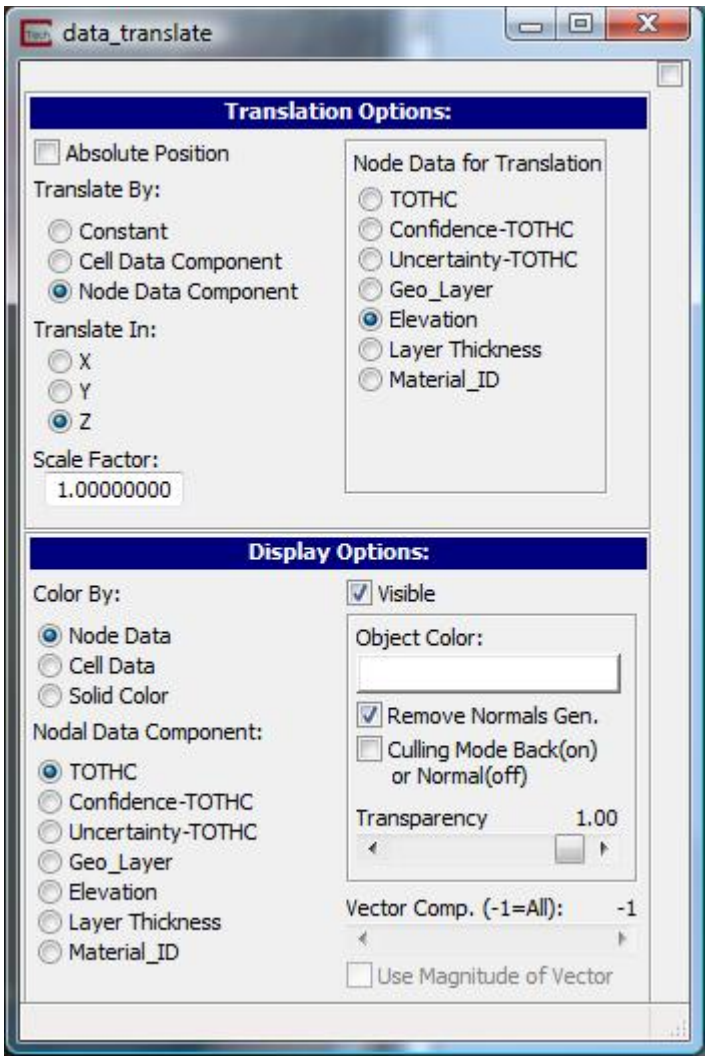
- When translating by a Constant, the amount is affected by the Z Scale Factor.
- When translating by Cell Data, a radio box appears to allow specification of the cell data component
- When translating by Node Data, a radio box appears to allow specification of the nodal data component

Under the **Display Options** subpanel you can specify the data for coloring as well as typical object properties.

The control panel shown below shows a cell data case that was used in the application shown at the bottom of this topic.



The control panel shown below shows a nodal data case.



When Node Data Component is chosen, the output cells will be translated by the Scale Factor times the value of whichever nodal data component is selected on the right. With nodal data you can only translate non-volumetric cells, otherwise the cell topology can become tangled.

Related Modules

extrude

combine_components

General Module Function

The combine_components module is used to create a new set of nodal data components by selecting components from up to six separate input data fields. The mesh (x-y-z coordinates) from the first input field, will be the mesh in the output. The input fields should have the same scale and origin, and number of nodes in order for the output data to have any meaning. This module is useful for combining data contained in multiple EVS Field (or netCDF, UCD, etc.) files, or from different Kriging modules.

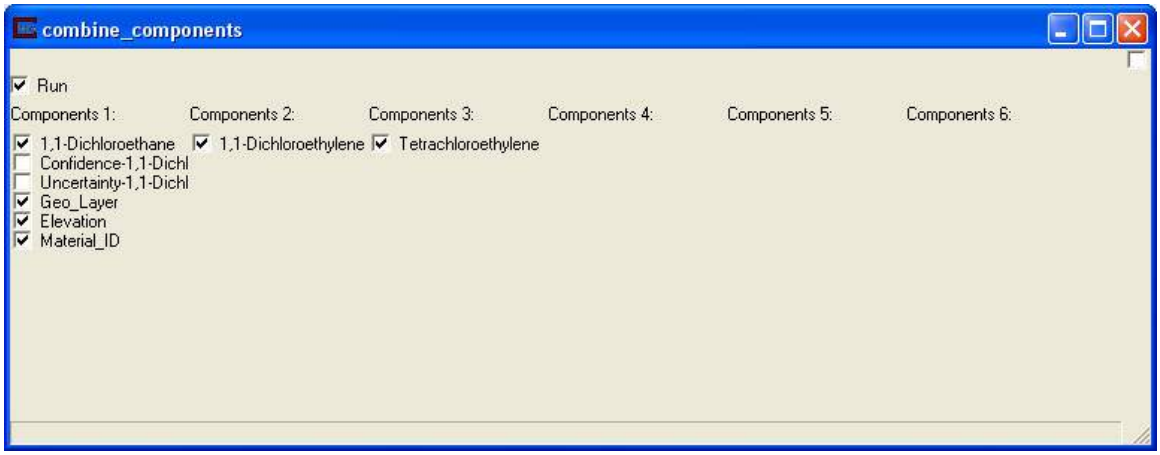
Module Input Ports

combine_components has six input ports. The left port must always be used and any number of the others (from zero to five). The mesh present in the first input port is the one that is exported and the number of nodes in each input port used must match the first or an error will be reported. You cannot combine data from fields with different numbers of nodes. Also, be aware that combine_components will not check to see that your coordinates or grids are identical, but it assumes that is the case.

Note: If you connect to an input port and then disconnect, the data components list will not be cleaned out.

Module Output Ports

combine_components has two output ports. The first output port (closest to the left) outputs a new set of nodal data components reflecting the nodal data components combined from any of the sixthe input data fields. If mesh data was supplied by the first input field, it will be preserved in the output. The second port (red) outputs a renderable object if the input is faces or lines.



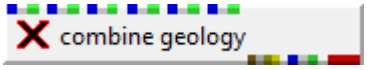
Module Control Panel

The control panel for combine_components is shown in the figure above. Up to six columns of check boxes are displayed. The left column contains all of the data component fields passed to the first input port. Likewise, the other columns contain all of the data components passed to their input ports. A check in the box next to any of these data components indicates a component to be included in the output. By default, the first (0th) data component in each column is selected and all other data components are not selected. Components having the same nodal component label will cause the output to have multiple nodal component labels that are the same. For example, if geo_layer is present and selected in both data component columns, the output will have two geo_layer data component.

It also has a Run toggle (to prevent downstream modules from firing during input setting changes).

Note that this module can be used similar to extract_component but allows you to choose one or more data components from a single field (must be input to the left port).

combine_geology



General Module Function

The combine_geology module is used to merge up to six geologic surface (per combine_geology module) to create a field representing multiple geologic layers.

The mesh (x-y coordinates) from the first input field, will be the mesh in the output. The input fields should have the same scale and origin, and number of nodes in order for the output data to have any meaning.

Module Input Ports

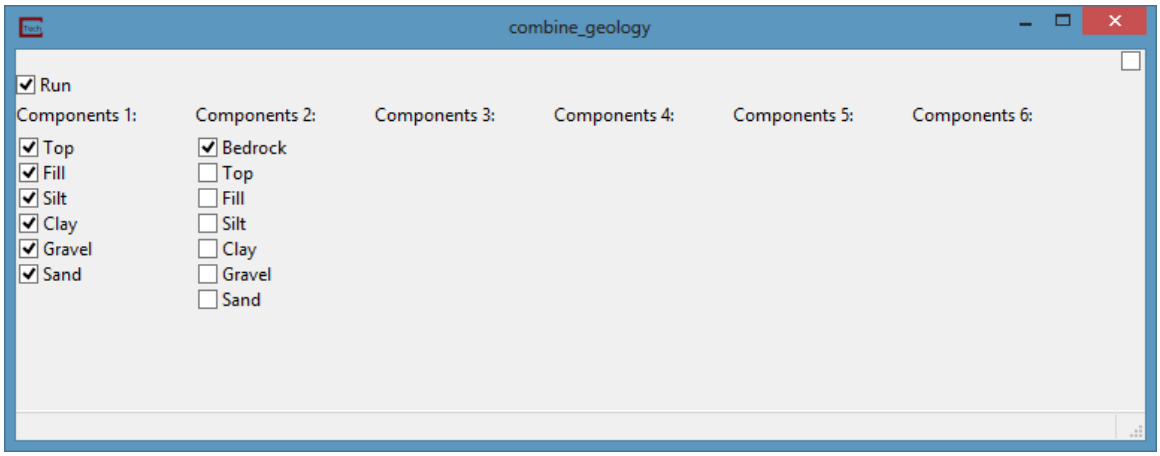
combine_geology has six input ports. The left port must always be used and any number of the others (from zero to five). The mesh present in the first input port is the one that is exported and the number of nodes in each input port used must match the first or an error will be reported. You cannot combine data from fields with different numbers of nodes. Also, be aware that combine_geology will not check to see that your coordinates or grids are identical, but it assumes that is the case.

Note: If you connect to an input port and then disconnect, the data components list will not be cleaned out. We recommend that you get a new module in this case.

Module Output Ports

combine_geology has three output ports.

1. The first output port (closest to the left) is Geology Material Names (Brown-Grey-Light Brown-Beige). It provides geologic material information for the Legend or Save_EVS_Field when saving the combined geologic 2D grid.
2. The second output port outputs a new set of nodal data components reflecting the nodal data components combined from any of the six input data fields. If mesh data was supplied by the first input field, it will be preserved in the output.
3. The third port (red) outputs a renderable object if the input is faces or lines.

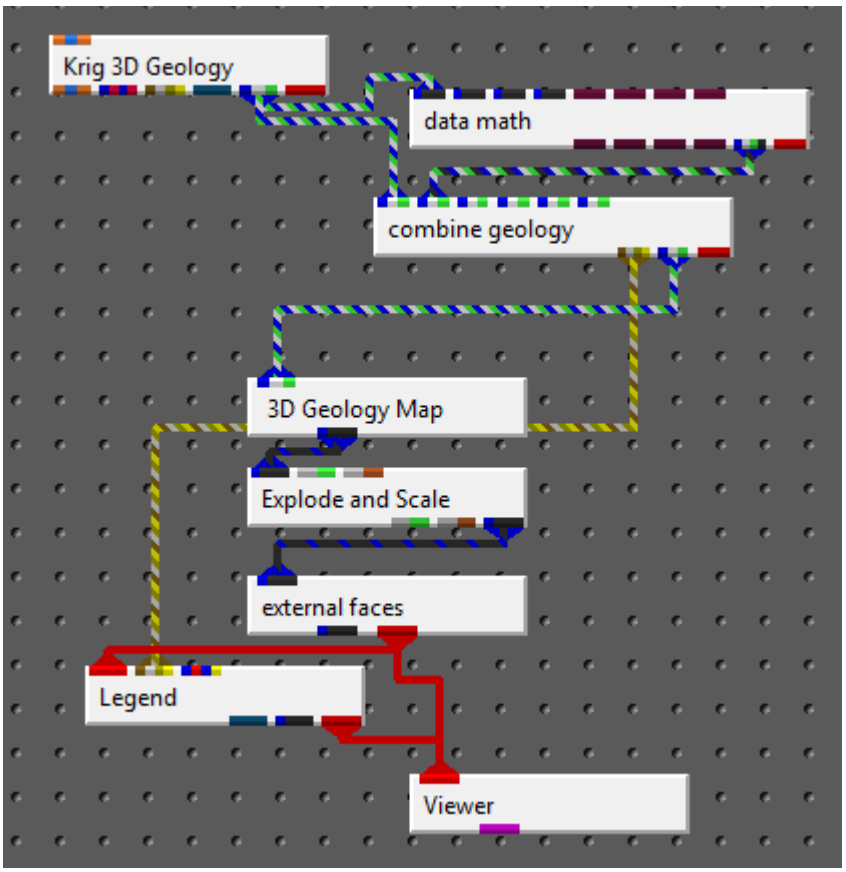


Module Control Panel

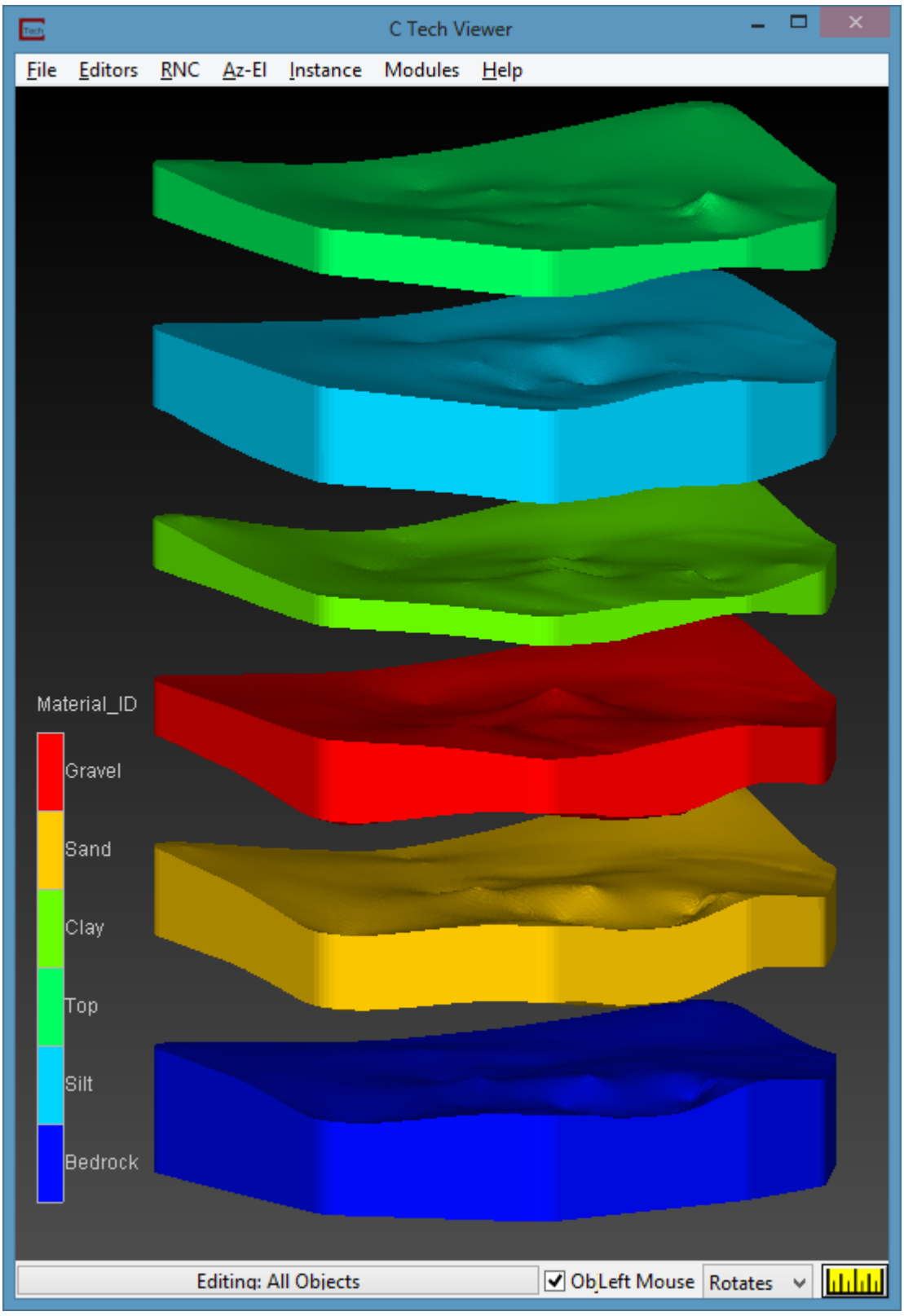
The control panel for combine_geology is shown in the figure above. Up to six columns of check boxes are displayed. The left column contains all of the data component fields passed to the first input port. Likewise, the other columns contain all of the data components passed to their input ports. A check in the box next to any of these data components indicates a component to be included in the output. By default, the first (0th) data component in each column is selected and all other data components are not selected. Components having the same nodal component label will cause the output to have multiple nodal component labels that are the same. For example, if geo_layer is present and selected in both data component columns, the output will have two geo_layer data component.

It also has a Run toggle (to prevent downstream modules from firing during input setting changes).

combine_geology provides an important ability to merge sets of surface or add additional surfaces to geologic models. It is important to understand the consequences of doing so and the steps that must be taken. The Brown-Grey-Light Brown-Beige port contains the material_ID numbers and names and it is important that the content of this port reflect the current set of surfaces/layers reflected in the geology. When Material_ID or Geo_Layer is presented in a Legend, this port is necessary to automatically provide the layer names. When combine_geology is used to construct modified geologic horizons, its Brown-Grey-Light Brown-Beige port MUST be used vs. the same port in Krig_3D_Geology such as in the application below:

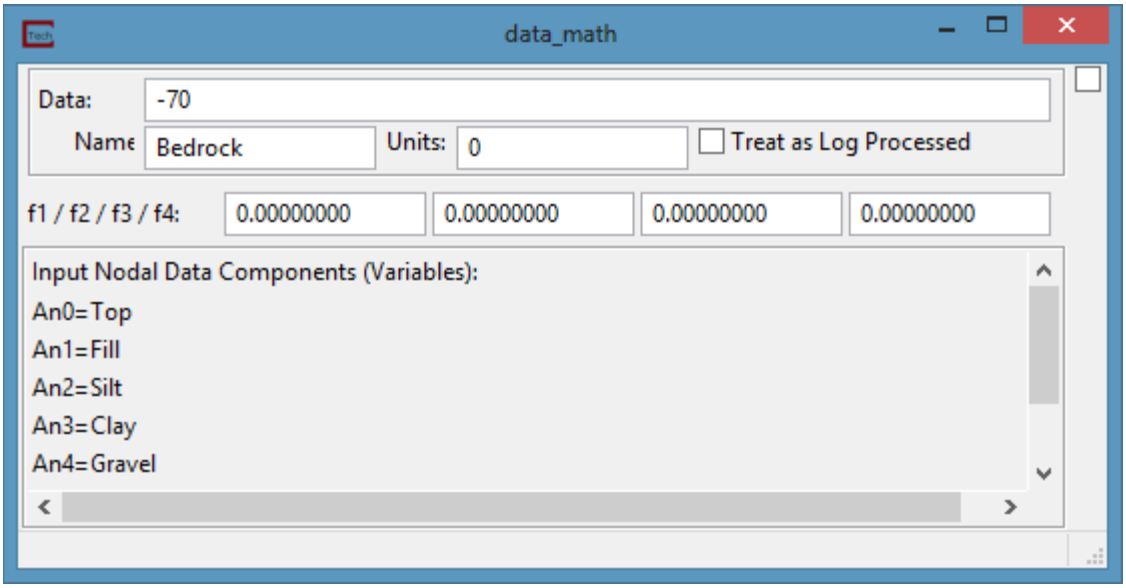


which creates the following output.

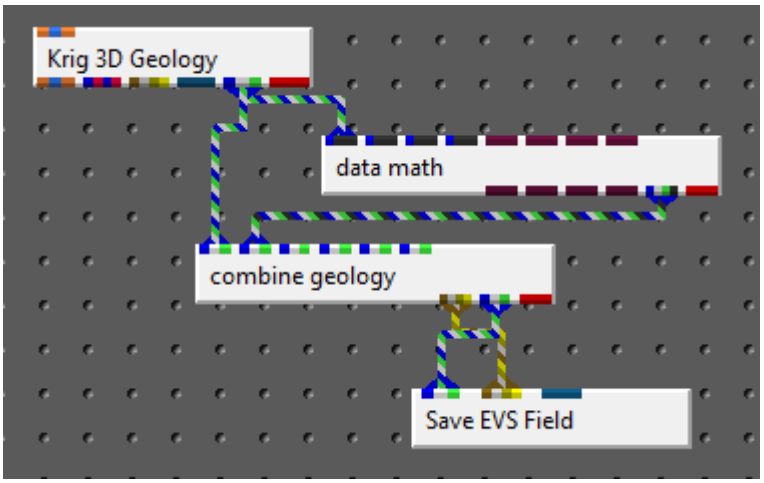


Please note that this is the standard dataset used throughout the workbooks, but now has a lower "Bedrock" layer which was created using data_math.

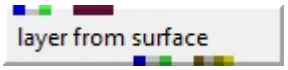
Please note that Units must contain a number which corresponds to the Material_ID for this new surface (geologic horizon).



When Save_EVS_Field is used to save combined geology, you must connect the ports as shown:



layer_from_surface



General Module Function

The layer_from_surface module will create a single geo layer based upon an existing surface and a constant elevation value.

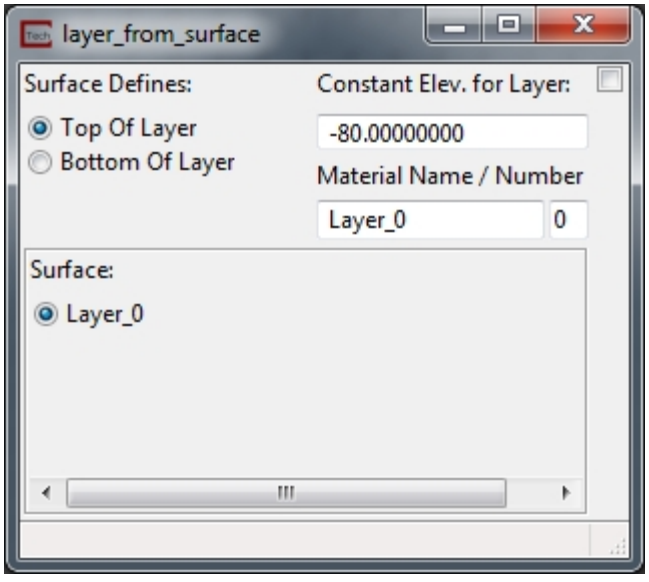
Module Input Ports:

- in_field (Blue-Grey-Green) : This input port accepts a geologic field containing the surface of interest for layer creation.
- ConstantValue (Maroon) : This input port accepts a float value used to define one of the surfaces for layer creation.

Module Output Ports:

- out_field (Blue-Grey-Green) : This output port contains the geology field consisting of the layer created by the module.
- geologyInfoOut (Brown-Grey-Green/Brown-Yellow/Brown) : Provides geologic material information for the Legend module.

Module Control Panel



The Surface Defines option will allow the user to set whether the selected surface defines the top or the bottom of the layer. For example if the Top Of Layer is chosen the selected surface will define the top, while the Constant Elevation for Layer will define the bottom of the layer. The 'Material Name / Number' will define the geologic layer name and number for the newly created layer. The Surface radio selector will allow the user to specify the layer of interest.

combine_vect

General Module Function

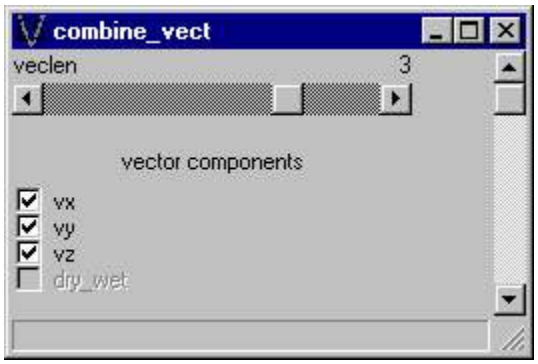
The combine vect (combine vector) module is used to create an n-length vector by combining n selected scalar data components. The vector length is determined by the veclen slider.

Module Input Ports

Combine vect has only one input port. The input field must contain nodal data components of which at least two must be scalar data components.

Module Output Ports

Combine vect has two output ports. The first output port (closest to the left) outputs a new nodal data component which is an n-length vector. If a mesh was present in the input field, it is preserved in the output. The second output port is undocumented at this time.



Module Control Panel

The control panel for combine vect is shown in the figure above. The veclen slider is used to specify the number of vector components to incorporate into the vector. Check boxes are used to specify which of the data components are to be included in the vector. Once the veclen number of components has been selected, the other data components are grayed out and not selectable. To change selections. first deselect one of the vector components and then select a new component. If no components are selected, then all components are selectable. The order in which the components are selected will determine in which order they occur in the vector.

Related Modules

combine_comp

extract_component

extract_scalar

magnitude

General Module Function

The magnitude module calculates the magnitude of a vector field data component at every node in a mesh. Input to magnitude must contain a

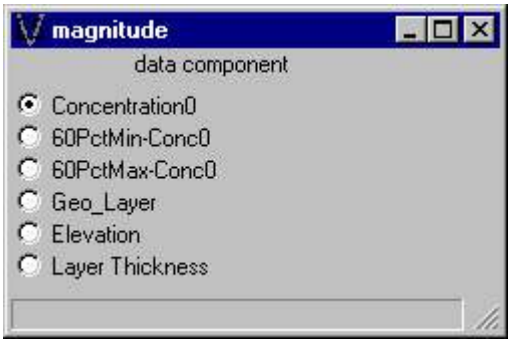
mesh of any type and nodal data. Nodal data components can be scalar or vector with up to 3 vector subcomponents.

Module Input Ports

Magnitude has only one input port. Input to this port must contain mesh data of any type mesh and nodal data.

Module Output Ports

Magnitude has two output ports. The first output port (closest to the left) outputs a new nodal data component consisting of a scalar data component representing the calculated magnitude at every node. Output mesh data is referenced to the input mesh data. The second port outputs a renderable geometry.



Module Control Panel

The control panel for magnitude is shown in the figure above. Data component radio button are used to select which data component is to be considered by magnitude. The default selection is the first (0th) data component. The data component selected can be a scalar or a vector with up to three vector subcomponents. Only one data component can be selected at a time. Therefore, any modules downstream of magnitude will receive only the data component selected in magnitude.

Related Modules

gradient

gradient

General Module Function

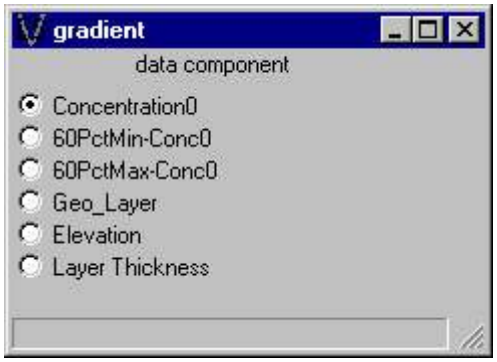
The gradient module calculates the vector gradient field of a scalar data component at every node in a mesh. Input to gradient must contain a mesh of any type and nodal data, with at least one scalar nodal data component. Gradient uses a finite-difference method based on central differencing to calculate the gradient on structured (rectilinear) meshes. Shape functions and their derivatives are used to calculate the gradient on unstructured meshes.

Module Input Ports

Gradient has only one input port. Input to this port must contain mesh data of any type mesh and nodal data. At least one of the nodal data components must be scalar.

Module Output Ports

Gradient has two output ports. The first output port (closest to the left) outputs a new nodal data component consisting of vectors with the X-Y-Z components representing the calculated gradient at every node. Output mesh data is referenced to the input mesh data. The second port outputs a renderable geometry if the input is not volumetric.



Module Control Panel

The control panel for gradient is shown in the figure above. The data component radio buttons are used to select which scalar data component is to be considered by gradient. The default selection is the first (0th) data component. Again, the data component selected must be scalar or an error will occur. Only one data component can be selected at a time. Therefore, any modules downstream of gradient will receive only the gradient of the data component selected.

Related Modules

->magnitude

capture_zone

General Module Function

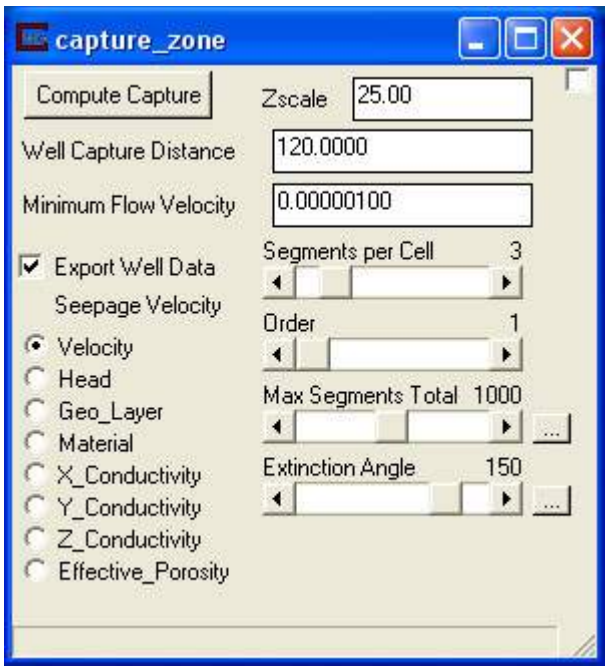
The capture_zone module utilizes streamlines technology to determine the volumetric regions of your model for which groundwater flow will be captured by one or more extraction wells.

Module Input Ports

capture_zone has three input ports. The selected nodal data component to the left input port must be a vector representing seepage velocity. The middle input port receives the nodal coordinates of the well(s) and does not require any data associated with the wells as only the coordinates are used. The right input port receives the z-scale (exaggeration) associated with BOTH input ports.

Module Output Ports

capture_zone has only one output port. The port outputs the original mesh with three new nodal data components.



Module Control Panel

The control panel for capture_zone is shown in the figure above. Many of the parameters are related to streamline interpolation which is discussed in the streamlines help.

The ***Seepage Velocity*** radio buttons display all data components (vector and scalar) passed to capture_zone. By default, the first (0th) component is selected.

The ***Zscale*** type-in reflects the value of the z exaggeration port.

The ***Well Capture Distance*** type-in is used to specify the **maximum** distance that a streamline must pass near an extraction well before that streamline's integration is terminated and it's original starting location is deemed to have been captured by a well. Setting this value too high may result in regions that are designated as being captured but are not. However, setting it too low may also result in incorrect determination of which (if any) well is responsible for capturing regions. **Fortunately**, the acceptable bounds for this parameter are rather wide and generally a value corresponding to one nominal cell width (those cells in the region of your wells) is appropriate.

The ***Minimum Flow Velocity*** type-in is used to specify the minimum velocity that will be considered in the integration. If the magnitude of the velocity field in a region is less than this minimum value, streamlines will end in that region (or will not be produced if the velocity at a starting point is less than the minimum). Setting this to a lower value will more accurately define the FULL extent of the capture zone region. However lower values may increase run time. The default value is 0.000001.

The ***Segments per Cell*** slider is used to set the number of integration steps to be used in each cell (i.e., the number of divisions of the cells) to calculate the streamline. The default is one and the range is 1 to 16.

The ***Order*** slider is used to set the order of the integration. Higher order integration is more accurate, but executes much slower. The default is one and the range is 1 to 4.

The ***Max Segments Total*** slider is used to set the maximum allowable number of streamline segments that will be completed for each streamline. If the number of segments along a streamline exceeds the max number, the streamline is terminated at the end of the last (max) segment. The default is 15 and the range is 1 to 1000.

The ***Extinction Angle*** slider is used to specify the maximum allowable angle between successive line segments before integration (streamline generation) should be terminated. The default value is 130 degrees.

seepage_velocity

General Module Function

The seepage_velocity module is used to compute the vector groundwater flow field visualizations of the vector field.

The input data requirements for the seepage_velocity module are:

1. A data component representing head (can have any name).
2. A Geo_Layer data component.
3. A Material_ID data component. If there is no Material_ID, we treat each layer as a separate material.
   Layer 0 becomes material -1
   Layer 1 becomes material -2
   Layer 2 becomes material -3, etc.

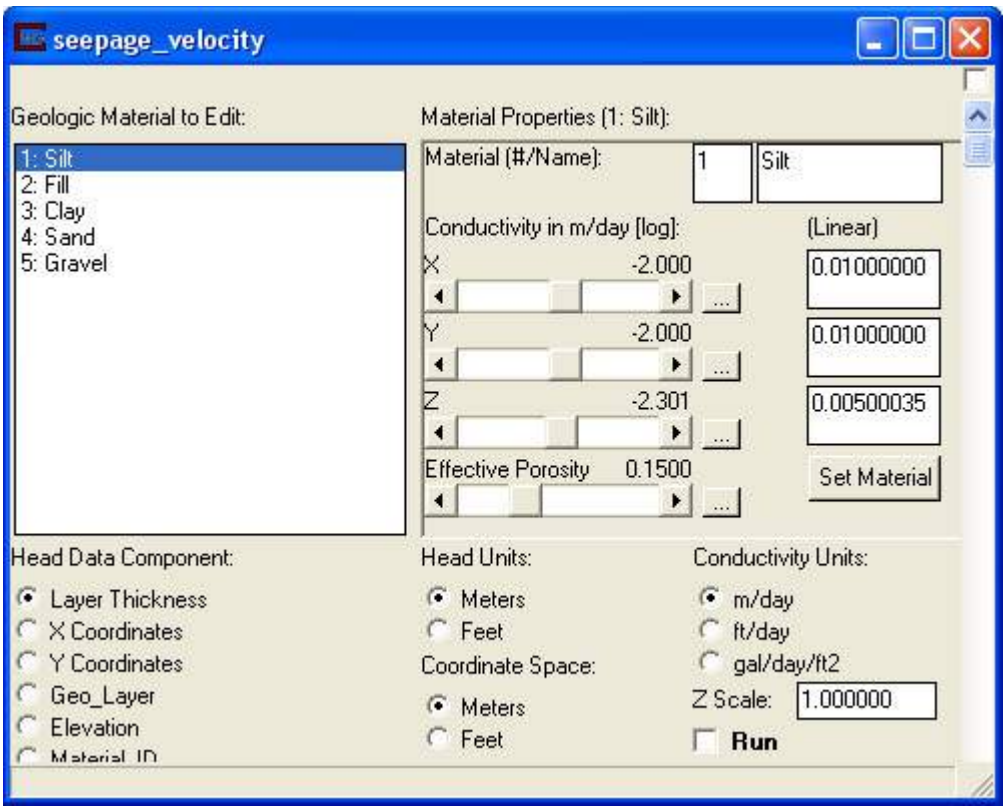
Note: If you use Krig_3D to krige head data with geologic input (in Version 6.0 or later) your output will meet these criteria (provided you toggle on these data components under *Kriging Parameters*). Data kriged in older versions and saved as a UCD or netCDF will lack the Material_ID data component.

Module Input Ports

Seepage_velocity has two input ports. Data passed to the first port must be a 3D mesh with data representing heads and *normally* multiple geologic layers. The second port is the Z-Scale (exaggeration) parameter.

Module Output Ports

Seepage_velocity has two output ports. The first is the Z-Scale (exaggeration) parameter . The second port contains vector data representing x, y, and z components of seepage velocity. This data can be processed by the glyph, streamlines, or advector modules.



Module Control Panel

The control panel for seepage_velocity is shown in the figure above.

The seepage_velocity

Geologic Layer to Edit: allows you to select the geologic layer to assign conductivity and porosity properties.

Head Data Component radio button list displays all data components passed to seepage_velocity.

HeadUnits radio button list allows you to specify the units of your head data.

Coordinate Space: radio button list allows you to specify the units of your coordinate data.

Conductivity Units: radio button list allows you to choose the units for specifying the conductivity in all three (x, y, z) directions for each geologic layer. You can choose any units (regardless of your head and coordinate units) and the appropriate conversions will be made for you.

The *Conductivity* sliders (with type-in buttons) allow you to change the log10 of the x, y, & z conductivity. These specify log values because conductivities vary over many orders of magnitude. These update when the (Linear) type-ins are changed.

The *Conductivity* type-ins allow you to change the x, y, & z conductivity. These are actual values and update when the sliders are changed.

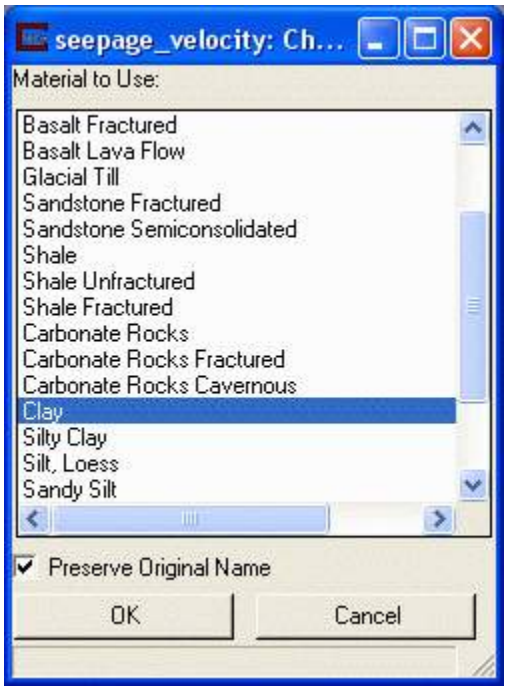
The *Effective Porosity* slider (with type-in buttons) allows you to change the value of effective porosity.

The *Set Material* button opens another window (shown below) where you can choose from many different geologic materials. When one of these materials is selected, all of the conductivity and the porosity values (and ranges in the sliders) are set for you. This allows you to quickly select nominal conductivity values based on the materials in your geologic layers.

The ***Run*** toggle determines if the module runs immediately when you change conductivity values. The default is **off.**

Material (#/Name): allows you to specify the material type if it is not specified in your geologic layers. This is only to help you assign proper conductivities.

Seepage_velocity component determines which data component is used to scale and rotate the seepage_velocity velocity vectors. The default selection is the first data component. The Map component radio button list also displays all data components passed to seepage_velocity. Map component determines which data component is used to color the seepage_velocity velocity vectors. By default, the first (0th) data component is selected.



Technical Details

Inherent in the solution of seepage velocity implemented in this module is the assumption that within each geologic layer/material the conductivities are uniform. Clearly, this will never be completely accurate, however we would contend that there is seldom if ever a better measure of the site conductivities (true conductivity tensor) than the site heads because head is far easier to measure. Furthermore, geologic materials can be deposited such that their conductivities are very complex and directional and most groundwater models (e.g. MODFLOW) do not provide a way to reflect this EVEN IF IT COULD BE MEASURED.

This approach allows users to quickly investigate the impact on flow paths due to changes in the conductivity assigned to each layer/material, BASED ON THE MEASURED/KRIGED HEAD DISTRIBUTION. Clearly, the more accurately the head is characterized the better.

At this point, we don't propose to provide a mechanism to account for conductivity variations within a geologic layer. We obviously cannot account for natural or artificial barriers (low conductivity regions) UNLESS they are represented by the geologic materials.

Our approach is:

Compute the true seepage velocity (Vx, Vy, Vz) at each node, by taking the gradient of (kriged) head (without any z-exaggeration) and multiplying each component of head gradient by the component of conductivity at that node (based on its material) (Kx, Ky, Kz) and dividing by the Effective Porosity for that material.

Vx = dH/dx * Kx / Ne

Vy = dH/dy * Ky / Ne

Vz = dH/dz * Kz / Ne

regional_averages

General Module Function

The regional_average module averages nodal data values from the input field that fall into the input polygon regions. It then outputs a point for each region that contains the average x, y coordinates and the average of each selected nodal data component.

Module Input Ports

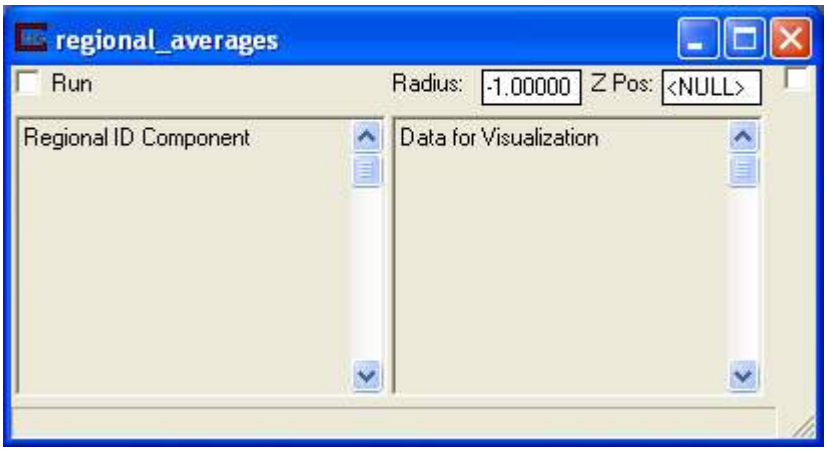
regional_averages has two input ports. The first port(the leftmost one) receives any field with nodal data. The second port receives triangulated polygons (from triangulate_polygon, or other sources) into its right input port. These polygons must contain at least 1 cell data component representing the regional ID.

Module Output Ports

regional_averages has two output ports.

The first port outputs a field of points with x and y coordinates located at the average x and y position for each region. The z coordinate of each point is located at the midpoint of the z extents of the input field. Each point contains the average of each nodal data component from the input field that falls within the selected polygon regions.

The second port contains a renderable sphere placed at each output point that is colored according to the selected data component.



Module Control Panel

The module's control panel is shown above.

The ***Run*** toggle causes the module to run whenever your inputs change.

The ***Radius*** field sets the radius of the renderable output spheres. Setting this value to -1.0 causes the radius of the spheres to be determined from the extents of the input field.

The ***Z Pos*** field allows the user to set the vertical position of the regional average output points.

The ***Regional ID Component*** selects which cell data component from the polygon regions input should be used to identify each region.

The ***Data for Visualization*** parameter selects which nodal data component from the input field to color the output spheres by.

geologic_surfmap

General Module Function

geologic_surfmap (only in EVS PRO and MVS) provides a mechanism to drape lines onto Geologic surfaces

Module Input Ports

geologic_surfmap has three input ports.

1) in_geology (Blue-White-Green) : This port should import a geologic surface.

2) in_field (Blue-Black) : This port takes a field of line cells that will be projected onto the geologic surface

3) z_scale (Grey-Brown) : This input port is the z-exaggeration factor (e.g. from Explode_and_Scale).

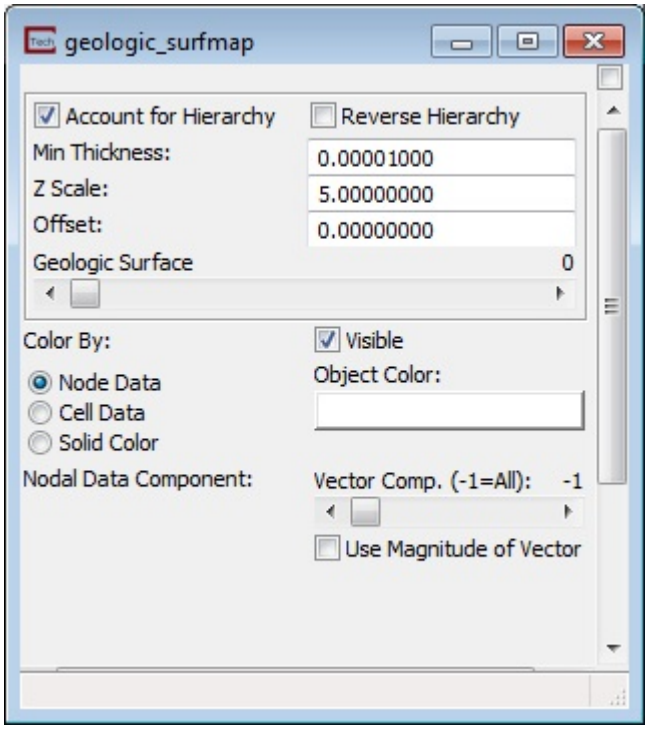
Module Output Ports

geologic_surfmap has three output ports.

1) z_scale (Grey-Brown) : This output port is the z-exaggeration factor.

2) out_fld (Blue-Black) : This port outputs an EVS field containing the draped line segments. surfmap will not preserve DXF entity colors. Rather, it colors the lines according to their elevations.

3) out_obj (Red) : This port outputs a renderable geometry directly to the Viewer.



Module Control Panel

The control panel for geologic_surfmap is shown in the figure above.

- The ***Account for Hierarcy*** toggle causes geologic surfaces that might otherwise intersect each other to be corrected to account for Geologic Hierarchy.
- The ***Reverse Hierarcy*** toggle causes hierarchy to be imposed from the bottom up versus the normal top down.
- The **Geologic Surface** slider selects which geologic surface to project the field onto.
- The ***Z_Scale*** should match that used for other objects in your application.
- The ***Offset*** parameter provides a means to offset the draped lines from the surface so that z-buffering will not make the lines ragged or partially invisible where they intersect the surface.
- The **Color By** options will color the output lines by either Node Data, Cell Data, or a Solid Color which can be selected using the Object Color button. If the data component is a vector the default is to color by the magnitude of the vector. The lines can be colored by a component of the vector by changing the Vector Comp slider.

surfmap

General Module Function

surfmap provides a mechanism to drape DXF files onto surfaces. You should also investigate the geologic_surfmap module.

Module Input Ports

surfmap has two input ports. The left input port should be geologic surface or other surface (such as the output of Krig_2D) and the right input port should be the output from Read_DXF.

Module Output Ports

surfmap has two output ports. The first output port (closest to the left) outputs an EVS field containing the draped line segments. surfmap will not preserve DXF entity colors in the current version of EVS. The second port outputs a renderable geometry directly to the Viewer.

Module Control Panel

The control panel for surfmap is shown in the figure above. The Z_Offset parameter provides a means to offset the draped lines from the surface so that z-buffering will not make the lines ragged or partially invisible where they intersect the surface.

If the *Run Automatically* toggle is not selected the module will not run unless the *Accept All Current Values* button is pressed. This keeps processing time to a minimum if there are multiple changes to be made upstream.

displace_block

(This module is available only in MVS)

General Module Function

displace_block receives any 3D field into its input port and outputs the same field translated in z according to a selected nodal data component of an input surface allowing for non-uniform fault block translation.

This module allows for the creation of tear faults and other complex geologic structures. Used in conjunction with surf_cut it makes it possible to easily model extremely complex deformations.

Module Input Ports

The displace_block module has two input ports. The leftmost port accepts input 3D grid and the right port, the incoming (2D) surface.

Module Output Ports

The displace_block module has two output ports.

The leftmost blue/black output port is the new mesh which will have the translated version of the input. The new mesh will include all of the nodal and cell data in the input field. The red port outputs a renderable object which will be colored by the first nodal data component in the input field.

Module Control Panel

The user interface for this module is shown above.

modify_data_3D

General Module Function

The modify_data_3D

Module Input Ports

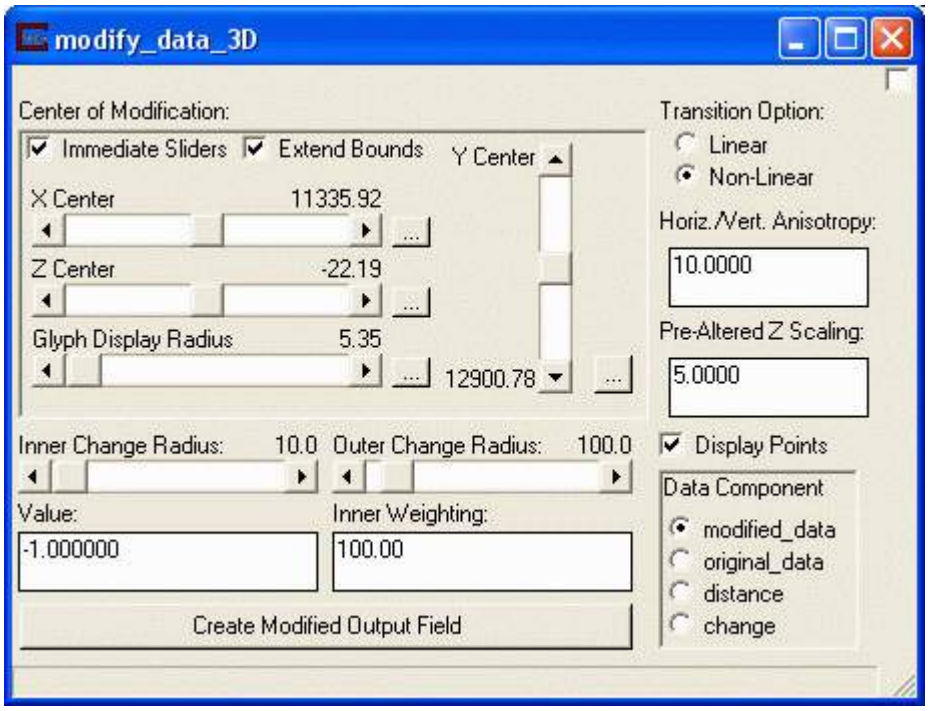
modify_data_3D has two input ports. The left input port will accept 3D field (mesh AND nodal data) which contains the data to be altered.

The right input port accepts the z scale (exaggeration) of the incoming coordinates.

Module Output Ports

modify_data_3D has two output ports.

The left port outputs the data field with modified nodal data. This port can be connected to any module that can process data fields, such as the isolines module, which can use the data field to display isolines if the output were a surface. The right port outputs a renderable version of the modified data.



Module Control Panel

The control panel for modify_data_3D is shown in the figure above.

The *Center of Modification* panel has sliders (with "" buttons to access type-ins) that control the center and radius of the

Glyph Display Radius is the size (in your units) of the spherical glyph that identifies the center modification location.

Outer Change Radius: is the radius in the x-y plane of the total region affected.

Inner Change Radius: is the radius in the x-y plane of the region affected at the constant *Inner Weighting*.

Modification Value is the value to change (clamp) the data set.

Inner Weighting is the constant value over the inner change radius.

Transition Options

Linear causes the weighting to be a straight line (linear interpolation) from the Inner Weighting value to zero at the outer change radius.

Non-Linear is a smooth transition (no discontinuities in slope) from the Inner Weighting value to zero at the outer change radius.

Horiz./Vert. Anisotropy is the ratio of the change radius in the x-y plane to the vertical (z) direction (for undistorted..not z-scaled coordinates).

Pre-Altered Z Scaling is the z-exaggeration you are using for display purposes.

Display Points toggle controls whether the affected nodes (cloud of points) will be displayed.

(Display) Data Component is the data component that is used for coloring the cloud of points.

modified_data will show the data as modified.

original_data will show the data as it is **unmodified**.

distance is the from the center.

change is the amount of change to the data.

Create Modified Output Field button causes the modified data to be output.

The figure below shows the cloud of points display from this module. Note the adaptively gridded regions with clusters of nodes!

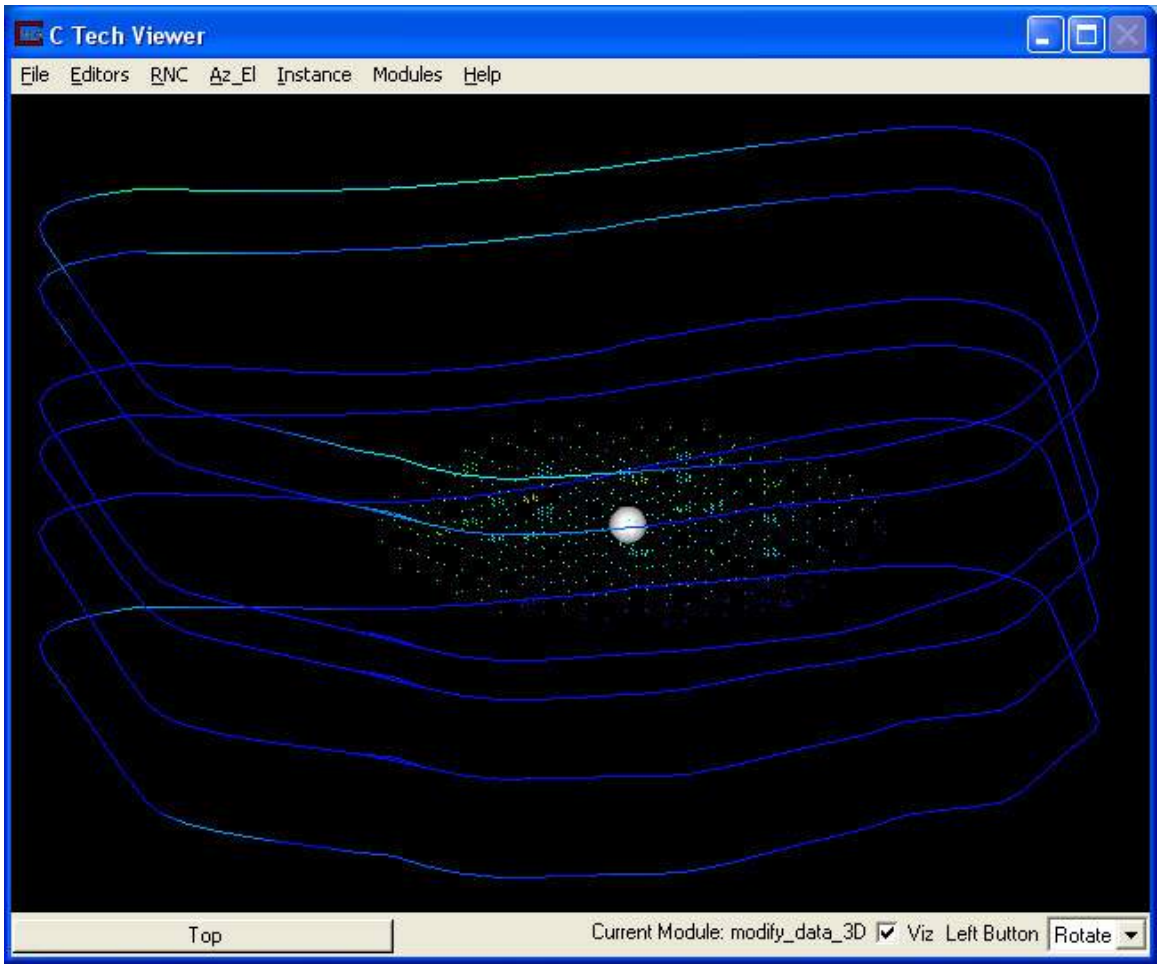


Image Modules

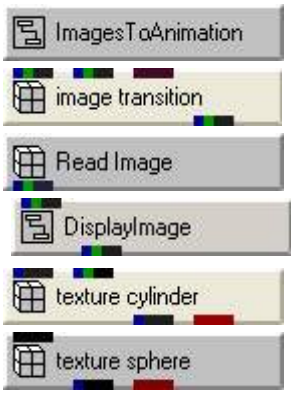


Image Modules

Output_Images

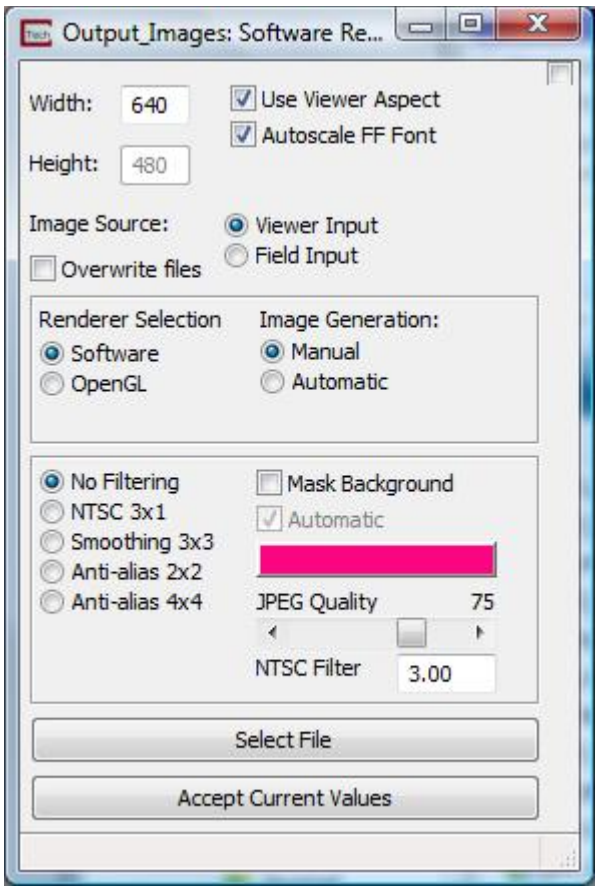
General Module Function

Output_Images will produce virtually all types of bitmap images supported by Windows. The most common types are .png;.bmp;.tga;.jpg; and .tif. PNG is the recommended format because it has high quality **lossless** compression.

Output_Images also allows automatic creation of images for using with modules such as advector or with fly-over loops.

Module Input Ports

Output_Images has only two input ports. The leftmost port only connects to the EVS Viewer and the right port connects to Read_Image and other modules that output images (e.g. image_transition).



Module Control Panel

The control panel for output_images is shown in the figure above. The user can specify the size of the virtual viewer using the Width and Height edit fields.

The **Width and Height** edit field are used to specify the number of pixels in the horizontal and vertical direction used to define the virtual viewer. These edit fields default to a width of 640 and a height of 480 (640x480) and their maximum is 4000. A Larger number of pixels creates better resolution, much larger files and longer creation times. See *Printing Quality Tips* at the end of this help topic for information on resolution, as it relates to printing devices and color reproduction.

If the **Use Viewer Aspect** toggle is selected, the height of the image will be automatically be determined based upone the aspect ratio of the Viewer and the selected image Width.

When the **Autoscale FF Font** toggle is selected all Forward Facing fonts in the image will be scaled depending upon the size of the output image.

The **suffix** specified for the output image file determines the type of output.

For PNG (portable network graphics), a compression slider is provided. The max value of 9 results in a very small increase in compute time for compressing the images. Since PNG is a LOSSLESS compression format, the quality of the image is not affected by this value.

For JPEG, a quality slider is provided. Higher qualities result in less LOSS to the image but create much larger files. We recommend using PNG instead of JPEG whenever possible. The PNG images are often smaller and are always higher quality than a JPEG image.

The **Image Generation** Manual and Automatic toggles determine whether the image will be created upon hitting Accept Current Values, or automatically create an image after each viewer change. For instance, the Automatic feature works well if you would like to save every advector step in an advector application run.

The **Renderer Selection** Software or OpenGL toggle determines the renderer to be used during image creation. The default is for Software rendering and will ignore the rendering currently chosen in the Viewer. Choosing *OpenGL* will utilize the *OpenGL* renderer in your graphics card. Depending on the capabilities of your card, you can write images up to 4096 x 4096 pixels (some professional cards may be able to go to 8192 x 8192).

Filtering: To enhance the quality of images (especially for video output to NTSC DVD) a number of filtering options have been added. The default is No Filtering and is most appropriate for output that will be printed to color printers. The other options should generally be used when creating .avi files or output to NTSC video where the resolution is limited by the media format.

- The NTSC (3x1) option is the quickest filter for NTSC (standard television) animation. NTSC video standard uses pairs of interlaced fields which contain odd and even pixel scan lines (respectively). Horizontal single pixel wide lines will flicker when viewed in NTSC. To minimize this effect, a blurring filter is used to blend lines to more than one horizontal pixel row. The value of this filter determines the level of blurring. A value of three is a reasonable starting point. A non-zero value, N results in a blurring which places N/(N+2) of the pixel's original color at the center and 1/(N+2) above and below the scan line. Small values of N therefore produce more blurring.
- Smoothing 3x3 is similar to NTSC, but blurs the image in the row direction and the column direction.
- The Anti-alias options are very different, and slow the image creation process. These options actually render an image that is twice as big as the specified Width and Height. This high resolution image is the filtered and subsetted to the specified size. This process reduces the brightness (contrast) of fine lines but it also smooths the lines and dramatically reduces jaggies. The 4x4 option with an NTSC Filter value of ~8.0 produces great results.

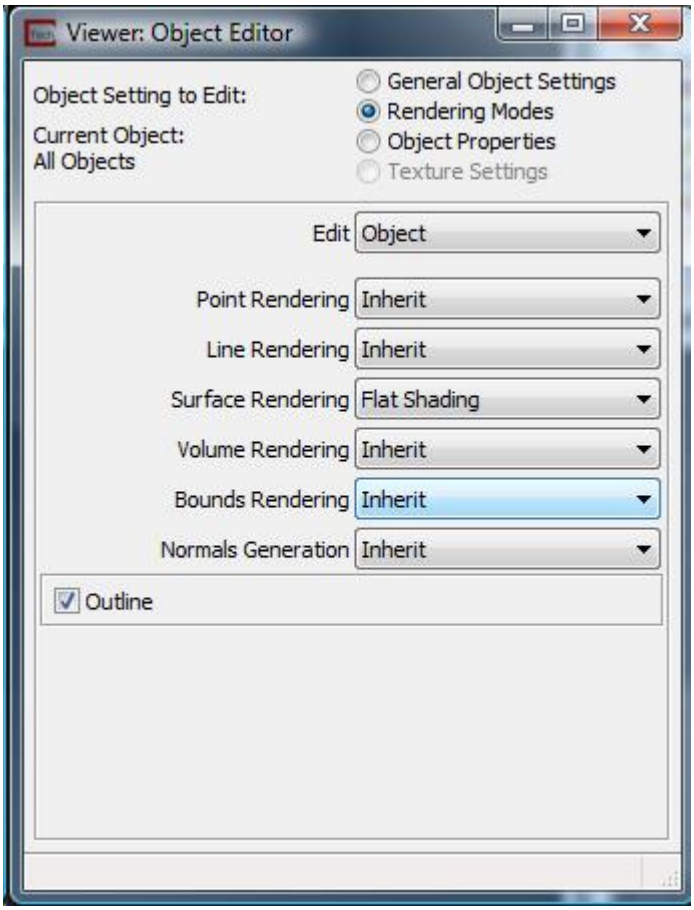
The **Mask Background** toggle allows you to create an image with a transparent background. In order to accomplish this, several things must be done:

- You must specify an image type that supports transparent backgrounds. PNG is recommended

- You must have a background color which is unique from any pixels in your objects which are rendered. This can be somewhat difficult if you have a rendered object with shading and specular highlights. Shading creates darker versions of the colors in your datamap and specular highlights creates less saturated (more white) versions of those colors. To avoid creating object colors that match your background, a masking background color should be selected which has a unique HUE not found in your datamap.
- When the **Mask Background** toggle is on, the **Automatic** toggle becomes active (selectable).
- With Automatic ON, upon rendering the image, the background color is temporarily changed to the color shown in the color button (default is a pink magenta), the image is rendered and your background is returned to normal.
- With Automatic OFF it still uses the color on the color button as the mask but expects you to set your background.
- Anti-Aliasing and filtering will intelligently detect the edges that are transparent and not mix in "pink" edges on your objects.

NOTE: There is no tolerance for matching the background color. The color must match the RGB value exactly.

TIP: The mask background function can be used to create transparent HOLES in your images. For example, a lake, which is rendered as a unique color could become a transparent hole in your rendered output. In order to accomplish this, the object which represents the lake must be colored to exactly match your mask color and it must have its surface rendering set to "Flat Shading".



The **Select File** button is used to bring up a standard windows file browser for choosing the name and location of the file to create. The Accept Current Values push button begins creation of the file.

When a file name has been selected using the Select File push button and all other parameters are set, the Accept Current values push button is used to create the file. When this button is selected, the EVS Status Window should indicate that the type of file being created. Nothing will happen when this button is pushed if 1.) No file has been selected, or 2.) no data is being sent to the input port of Output_Images.

Print Quality Tips:

The following provides hints and tips for obtaining the optimal quality from your printing device. This assumes you are using a color printer, but it is important to note that the user may print grayscale images with a black and white printer if desired. This would of course be best implemented by creating grayscale datamaps for the objects of interest.

The print features in EVS & MVS allow you to specify the resolution of the rendered bitmap image that will be used for output to a printer. The process for determining the optimal size (width and height) requires taking several factors into consideration.

First, you must know the characteristics of the printer and the size of the printed image. Printers vary considerably and no single recommendation can be appropriate. Color printers fall into three primary categories. They are: inkjet; color laser; and dye sublimation. EVS produces bitmaps which are continuous tone with 256 shades each of red, green and blue for a total of 16.7 million possible colors (256*256*256). Color printers must either produce continuous tones or approximate them using a pattern of primary colored pixels in an n-by-n grid.

Among these three printer categories there is considerable variation. Inkjet printers are (with very few exceptions) capable of producing one of only eight primary colors for each printer pixel (or dot). These colors are: white, black, cyan, magenta, yellow, red, green and blue. Inkjets

must therefore use a grid of primary colored pixels to approximate continuous tones. The larger the grid (4 by 4 vs. 2 by 2) the better the approximation. However, larger grids tend to create artifacts called jaggies which are undesirable visually. The challenge is to balance the need for smoother color rendition with the desire to have higher resolutions.

Dye sublimation printers are at the other end of the spectrum. Their ability to reproduce continuous tones makes the task of choosing a resolution easy. A typical dye-sub printer has a resolution of 300 dots per inch (dpi). If the intended size of the final printed image is 10 inches wide by 7 inches tall, then the optimal image size is 10*300 by 7*300 or 3000 x 2100 pixels. If quicker image creation and print times are desired, a compromise resolution would be exactly half or 1500 wide by 1050 high.

IMPORTANT NOTE: We always want to have an integer number of printer pixels for each "source" image pixel. When we reduce the image size by half, we get a 2-by-2 grid. The n-by-n grid concept applies to ALL TYPES OF PRINTERS. This "rule" is actually a guideline for best results. Other resolutions (non-integer ratios) create banding artifacts that are usually objectionable.

For inkjet printers you should always allow for at least a 2x2 grid and usually 3x3 to 5x5 gives the best results. For an EPSON printer with 720x1440 dpi resolution you should use the smaller resolution number (720) for your calculations. The printer will use the additional resolution to better approximate the colors.

Example: For an Epson with 720 dpi, to print an image 9 by 7.5 inches (landscape) we recommend that you start at a 4x4 grid which gives an effective printed resolution of 180 dpi. Your image width and height would therefore be:

width = 9.0 * 180 = 9.0 * (720/4) = 1620

height = 7.5 * 180 = 9.5 * (720/4) = 1350

Finally, color laser printers vary in their abilities to approximate continuous tones. This means that the rules to apply will be somewhere between dye-sub and inkjet properties.

overlay_aerial

General Module Function:

The overlay_aerial module will take as input a field and then map an image onto the horizontal areas of the grid. The image can be projected from one coordinate system to another. It can also be georeferenced if it has an accompanying All vertical surfaces (Walls) can be included in the output but will not have image data mapped to them.

Module Input Ports:

1) surface_to_texture (Blue-Black): This port accepts the grid to be mapped onto.

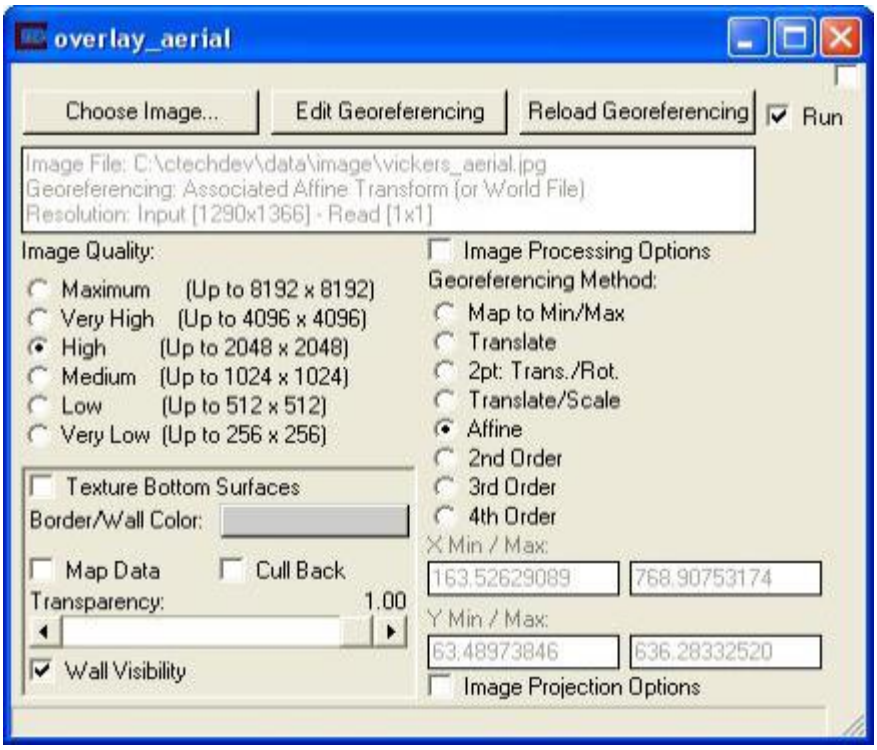
2) filename (Grey-Blue-Grey): This port allows for sharing the image filename.

Module Output Ports:

1) filename (Grey-Blue-Grey): This port allows for sharing the image filename.

2) out_surfaces (Red): This port outputs the mapped renderable object.

3) out_walls (Blue-Black): This port outputs the vertical surfaces. This can be used to texture map the vertical surfaces.



Module Control Panel:

Choose Image: This button will bring up a file browsing menu to allow for the selection of the image file from among the following types:

- *.png
- *.bmp
- *.jpg
- *.jpeg
- *.gif
- *.hav
- *.avi
- *.mpg
- *.bay
- *.giff
- *.pcx
- *.pnm
- *.ras
- *.sun
- *.rgb
- *.sgi
- *.tif or *.tiff
- *.xpm
- *.sid"

Edit Georeferencing: This button will bring up a window that will allow either a world file or GCP file to be created. NOTE: Please examine the Utility help section titled Georeference Image for instructions on how to use this feature.

Reload Georeferencing: This button will cause the file to be re-read and any new georeferencing created with the Edit Georeferencing button.

Image Processing Options: This toggle will bring up a window that allows for the adjustment of image brightness, sharpness, etc..

Image Quality: This selector limits the max resolution of the image being read. Most graphics cards support the ***High*** resolution of 2048, but relatively few support 4096 and only professional level cards and some of the newest DirectX 10 cards support 8192. Obviously higher resolution images will take more memory and more time to read, but will look much better when zoomed in.

Georeferencing Method: There are 8 different texture mapping modes as follows:

1) Map to Min/Max - Map image to the min/max extents of the input surface, or a user-defined value (can be typed into overlay_aerial directly).

2) Translate - Translate the image. Only requires a single GCP. No rotation or scaling is performed.

3) 2 pt: Trans./Rot. - Translate, Scale, and rotate the image. The image scaling is always the same in X&Y. Only a valid option if you have 2 GCPs. Good option if you only know 2 GCP points, and they are co-linear or near co-linear.

4) Translate/Scale - Translate and scale the image. Scale in X and Y are not the same. This keeps the image orthorectified. Can be used with 2 or more GCP points.

5) Affine - Perform a full affine transformation (1st order transformation) on the image. Requires a world file or 3 or more GCP points (from a gcp file). This is the default option which can be fully described with a World File.

6) 2nd Order - Perform a 2nd order polynomial transformation. This requires 6 or more GCP points (from a gcp file). It will map straight lines in the image into arcs. Allows an image that was georeferenced previously into LAT/LON coordinates to be "straightened" out and handled correctly. This can also be used to adjust for minor problems in the image due to topography. This option cannot be described with a World File because it uses a second order polynomial with more terms than are available in a world file. It requires the use of a GCP file.

7) 3rd Order - Perform a 3rd order polynomial transformation. Requires 10 or more GCP points. Allows you to adjust for drift in the image, "wedge" shaped photography, and more.

8) 4th Order - Perform a 4th order polynomial transformation. Requires 15 or more GCP points. Allows adjustments to be made where different portions of the image move in opposite directions. Requires many GCP points to use effectively.

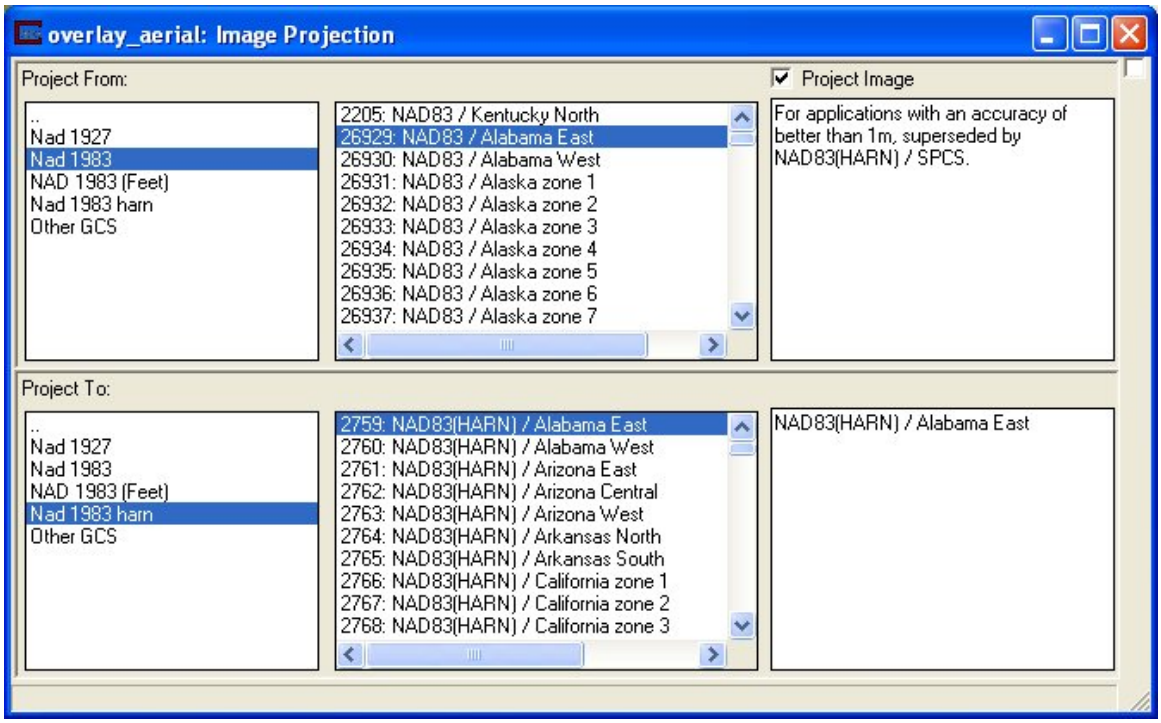


Image Projection Options: This toggle allows for the reprojection of the image. Each coordinate system is divided into either Geographic or Projected coordinate systems. The coordinate system types are navigated by selecting the appropriate system type in the far left window. When a general coordinate system has been selected a specific coordinate system can be selected from the center window. If there are any details regarding the selected specific coordinate system, they will appear in the text window on the right. A specific coordinate system must be selected both to project from and to project to, and then the Project Image toggle must be turned on.

Texture Bottom Surfaces: This toggle will tell the module to texture all surfaces whose normals are point downward.

Border / Wall Color: Since vertical surfaces (Walls) are not being textured this color button allows the color for these walls to be set.

Map Data: This toggle will cause the module to pass the first nodal data component through to the renderable object. NOTE: This will cause there to be a blending between the the nodal data color value and the image color.

Cull Back: This toggle when turned on will remove all faces that the user cannot see from the renderable object. This is useful for making a clearer picture when the transparency is not at one hundred percent.

Wall Visibility: This toggle when turned on will display the vertical surfaces (WALLS).

texture_walls

General Module Function:

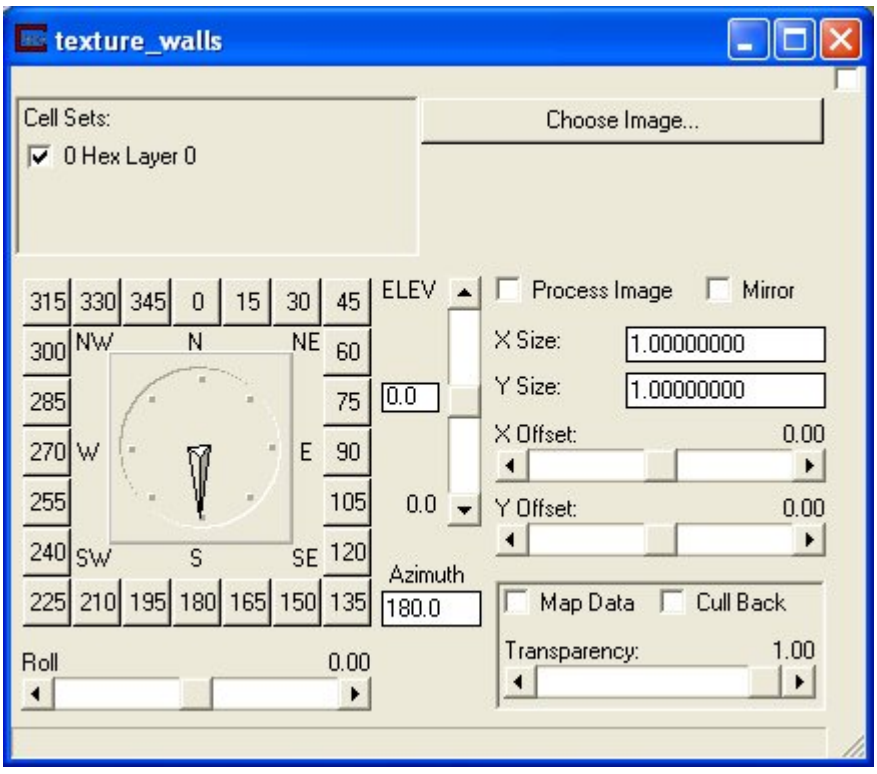
The texture_walls module provides a means to project images onto surfaces such as walls of buildings to add more realism to your visualizations.

Module Input Ports:

1) surface_to_texture (Blue-Black): This port takes the field to be textured.

Module Output Ports:

1) out_surfaces (Red): This port ouputs a renderable geometry consisting of the original field that has been wrapped in the selected image.



Module Control Panel:

Cell Sets: This option box allows for the selection of cell sets to be textured.

Choose Image: This button sets the image file name.

Azimuth: This value rotates the image to be textured around the Z axis in a 360 degree range based upon the center of the object.

ELEV: This value rotates the image along the horizontal plane in a 180 degree range.

Roll: This value rotates the up direction of the Z axis in a 360 degree range. It is normally used only in special circumstances.

Process Image: This toggle will bring up a window that allows for the alteration of the image brightness, contrast, etc.

Mirror: This toggle will change the behavior of the module in cases where the size of the image is smaller than that of the surface it is being mapped onto. When this toggle is turned off the image will just repeat when the image reaches its maximum resolution. If this toggle is turned on the image will instead flip over when it reaches its maximum resolution. This can make a more seamless texture when using certain images.

X Size: This value will scale the size of the image in the x direction.

Y Size: This value will scale the size of the image in the y direction.

X Offset: This slider will cause the image to shift in the x direction when being mapped onto the surface.

Y Offset: This slider will cause the image to shift in the y direction when being mapped onto the surface.

Map Data: This toggle when turned on will cause the first nodal data component to be displayed along with the image. NOTE: Since the data value is represented as a color in the display the same as the image, there will be a blending of the two when this option is used.

Cull Back: This toggle when turned on will remove all of the cell faces that are not facing the viewer. This is useful when Transparency is less than one hundred precent as the back faces can obstruct the view somewhat.

Transparency: This slider alters the transparency of the output renderable geometry.

texture_geology

General Module Function:

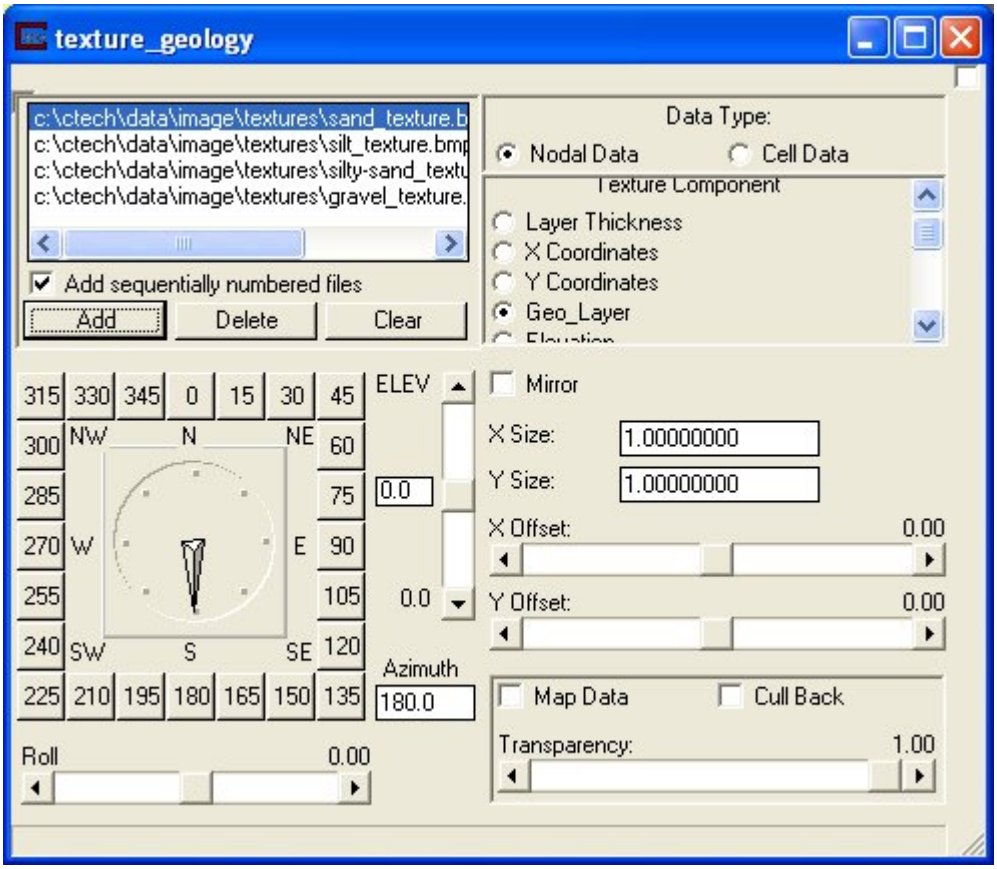
The texture_geology module will texture multiple images onto a field based on the geologic data in the field.

Module Input Ports:

1) in_field (Blue-Black): This is the field that images will be mapped onto.

Module Output Ports:

1) out_obj (Red): This port outputs a renderable geometry for the viewer.



Module Control Panel:

Add sequentially numbered files: This toggle when turned on will cause files with sequentially numbered names to be added when the first of the sequence is selected as a file. For example (image01.bmp, image02.bmp, etc.).

Add: This button will bring up a file browser and allow for the selection of an image file.

Delete: This button will remove the selected file from the file list.

Clear: This button will remove all files from the file list.

Data Type: This radio button will switch the data being looked at from nodal to cell data. NOTE: There are only three components that will work with this module: Geo_Layer; Material_ID; and Indicator.

Azimuth: This value rotates the image to be textured around the Z axis in a 360 degree range based upon the center of the object.

ELEV: This value rotates the image along the horizontal plane in a 180 degree range.

Roll: This value rotates the up direction of the Z axis in a 360 degree range. It is normally used only in special circumstances.

Process Image: This toggle will bring up a window that allows for the alteration of the image brightness, contrast, etc.

Mirror: This toggle will change the behavior of the module in cases where the size of the image is smaller than that of the surface it is being mapped onto. When this toggle is turned off the image will just repeat when the image reaches its maximum resolution. If this toggle is turned on the image will instead flip over when it reaches its maximum resolution. This can make a more seamless texture when using certain images.

X Size: This value will scale the size of the image in the x direction.

Y Size: This value will scale the size of the image in the y direction.

X Offset: This slider will cause the image to shift in the x direction when being mapped onto the surface.

Y Offset: This slider will cause the image to shift in the y direction when being mapped onto the surface.

Map Data: This toggle when turned on will cause the first nodal data component to be displayed along with the image. NOTE: Since the data value is represented as a color in the display the same as the image, there will be a blending of the two when this option is used.

Cull Back: This toggle when turned on will remove all of the cell faces that are not facing the viewer. This is useful when Transparency is less than one hundred precent as the back faces can obstruct the view somewhat.

Transparency: This slider alters the transparency of the output renderable geometry.

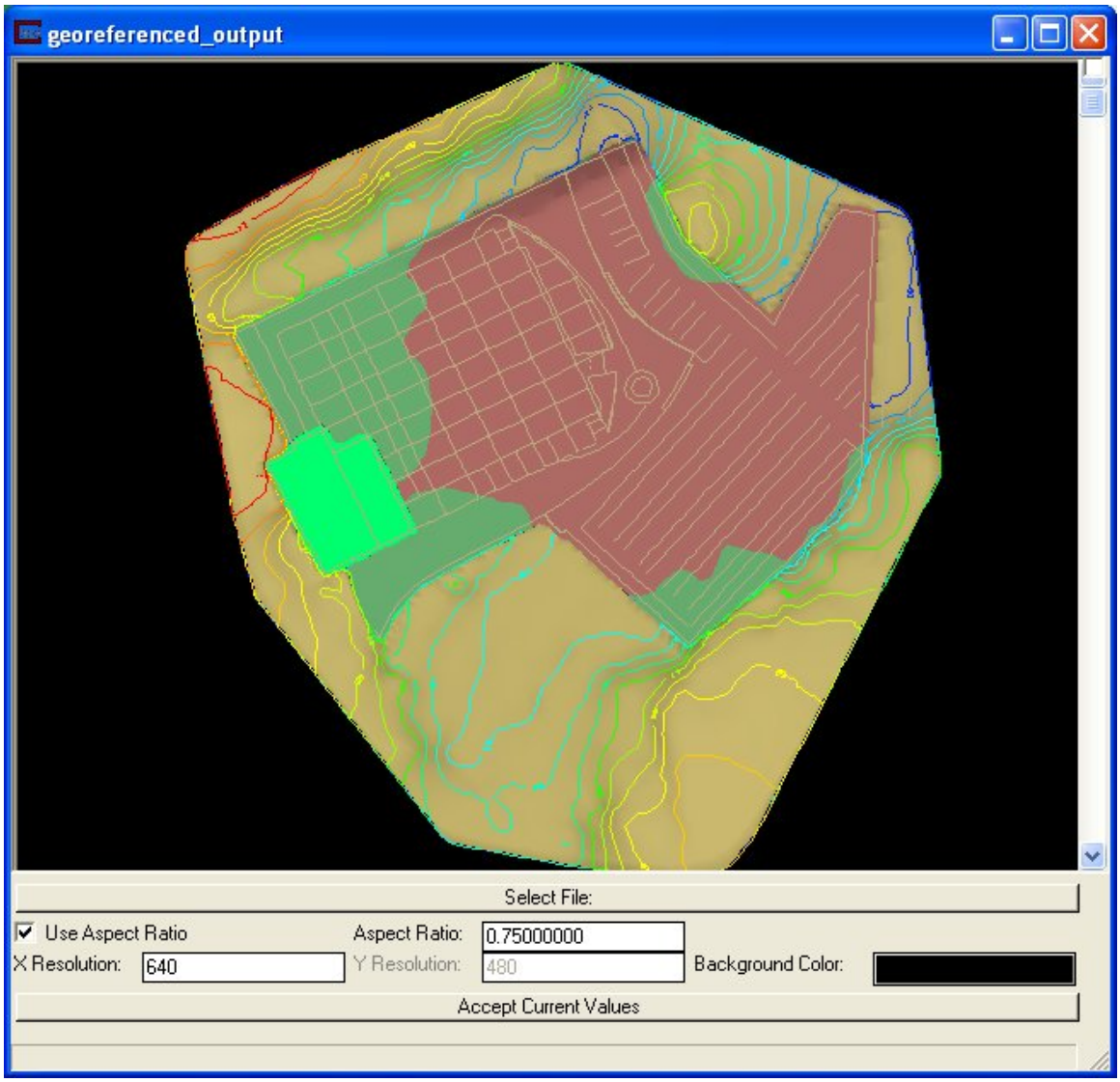
georeferenced_output

General Module Function:

This module will output a image in one of the following formats: BMP; TIF; JPG; and PNG. It will also output a world file that will allow the image to be placed correctly in applications that allow georeferencing.

Module Input Ports:

1) child_objs (Red): This port takes any number of rendered objects.



Module Control Panel:

Select File: This button will open a file browser and allow for the selection of the output image filename.

Use Aspect Ratio: This toggle when turned on will automatically set Y Resolution of the image based on the X Resolution and the Aspect Ratio.

Aspect Ratio: The ratio of the the Y Resolution to the X Resolution.

X Resolution: The resolution in the x direction of the output image.

Y Resolution: The resolution in the y direction of the output image.

Background Color: This color button allows the background color to be changed.

ImagesToAnimation

General Module Function

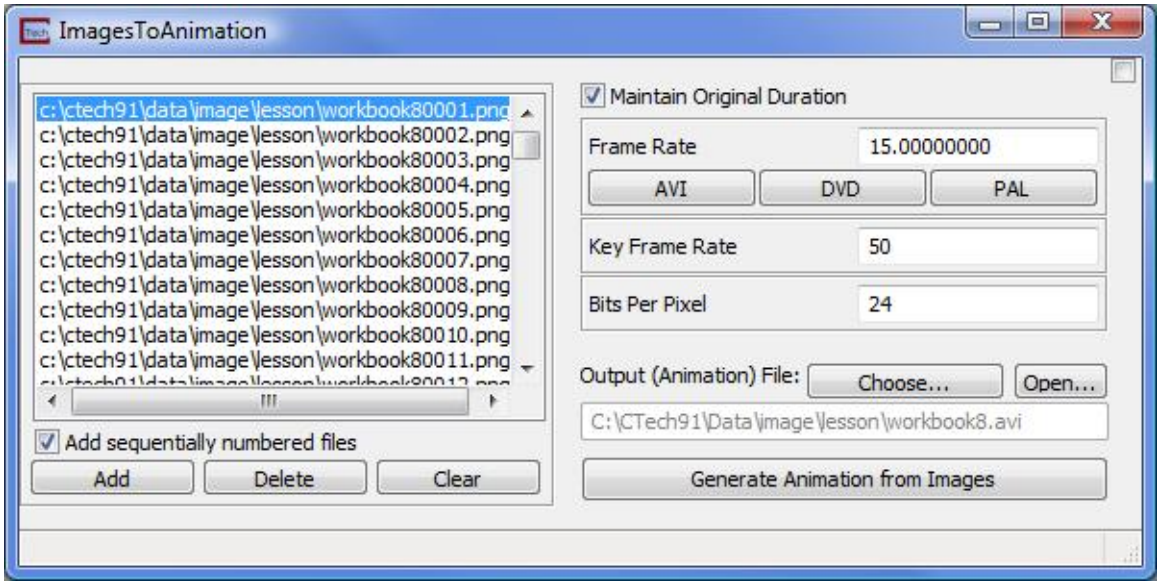
ImagesToAnimation incorporates the functionality of Gromada's VideoMach into an easy to use module.

Animation CODECS try to default to TechSmith Screen Capture Codec, or, if that doesn't exist, to the HuffYUV codec (both are lossless).

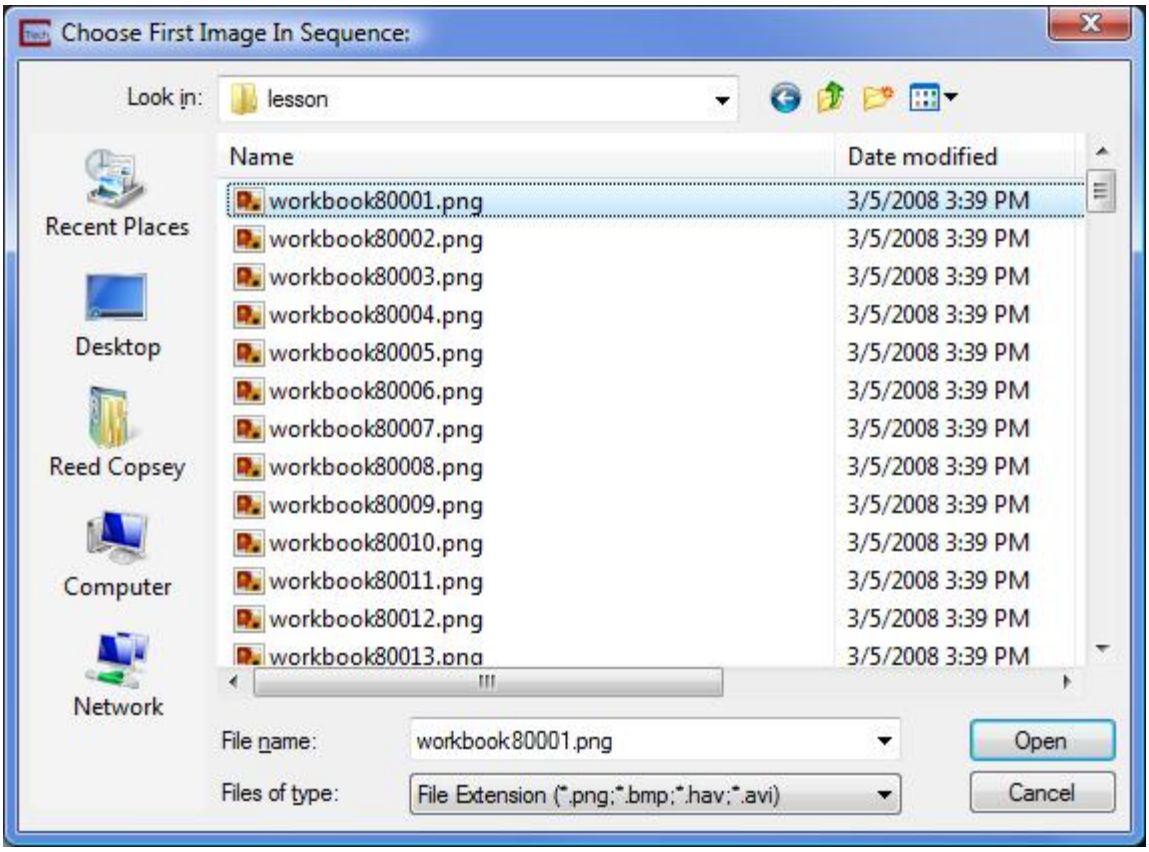
ImagesToAnimation supports several animation file formats including:

1. AVI Windows Audio Video Interleaved (avi)
2. Windows Media Video - WMV produces some of the smallest files and this output is virtually guaranteed to run on any up to date Windows computer. However, the quality can be poor unless the data rate is increased.
3. MPG Moving Pictures Expert Group, MPEG-1 (mpg,mpeg)
4. HAV High quality Audio Video (hav) HAV is a format that can be played with the freeware program Imagen (formally HAV player). This format has some distinct advantages, specifically it uses lossless compression. This results in the highest quality output. Surprisingly, HAV files are often as small as or smaller than lower quality AVI or MPG files.

First, instance the ImagesToAnimation module. The user interface for it is shown below.



Click on the Add button to bring up the panel to specify your image files.



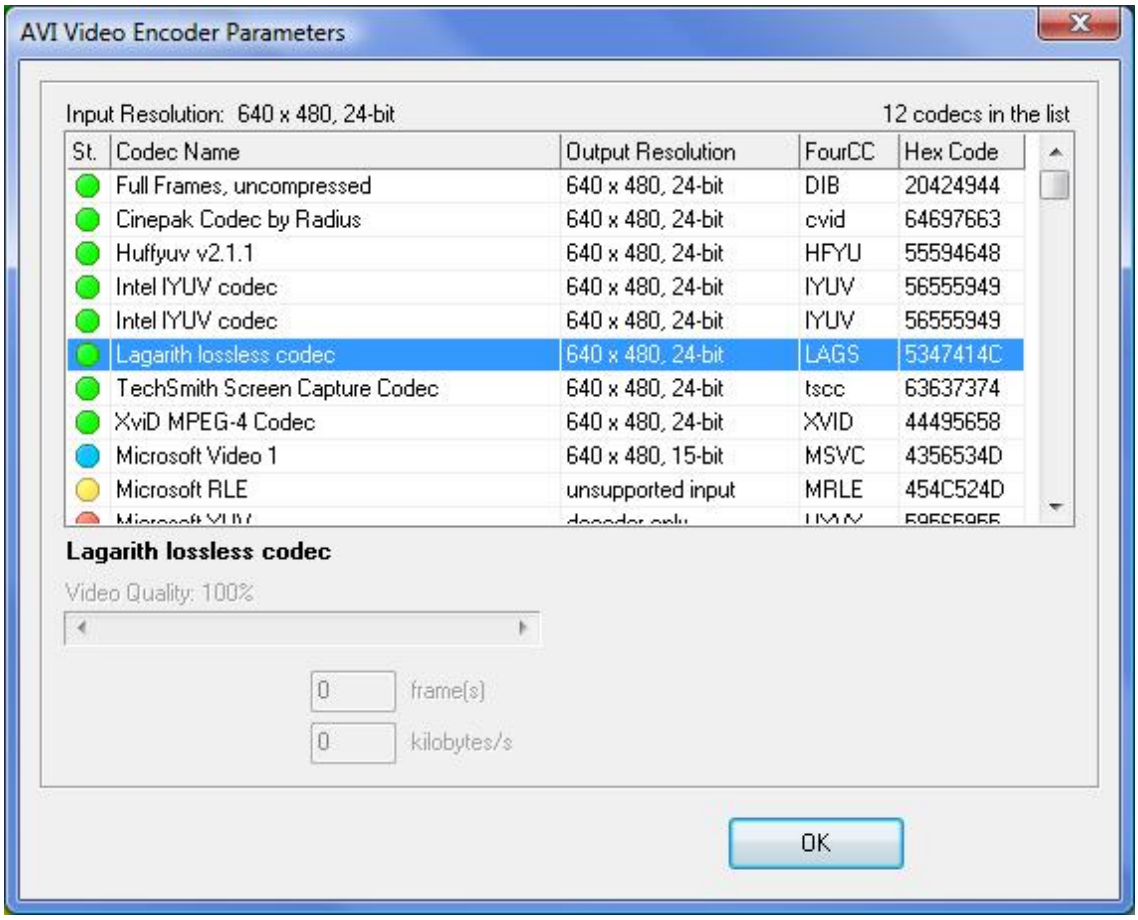
Click on the first file of the sequence you created. You only need to select the first file, then it will automatically select and process all files in the numbered sequence. You also need to "Choose" the Output (Animation) File. The suffix you choose determines the type of animation file to be created. The choices are hav (lossless high quality Gromada format), avi (windows standard) and mpg/mpeg (MPEG 1 format). The example above is set to workbook8.avi

Click on the "Generate Animation from Images" button to make an AVI file by using the input data (your PNG files).

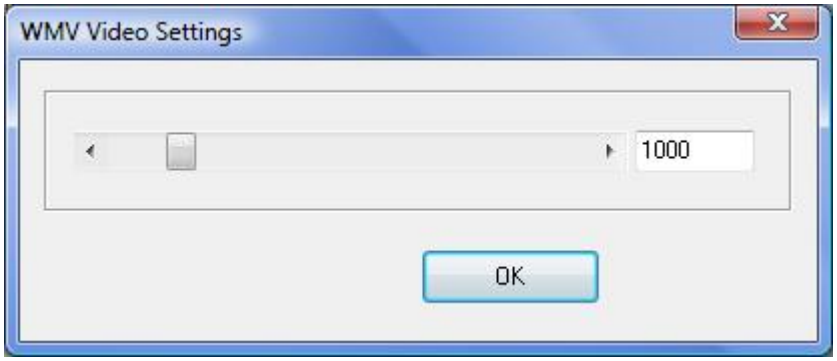
When you do, the first thing that will happen is a window will pop up which allows you to set the Output Animation Codec:

This shows the selected default CODEC for the type of animation file you've created. There are four supported bitmap animation file types which are AVI, WMV, MPG (MPEG), and HAV.

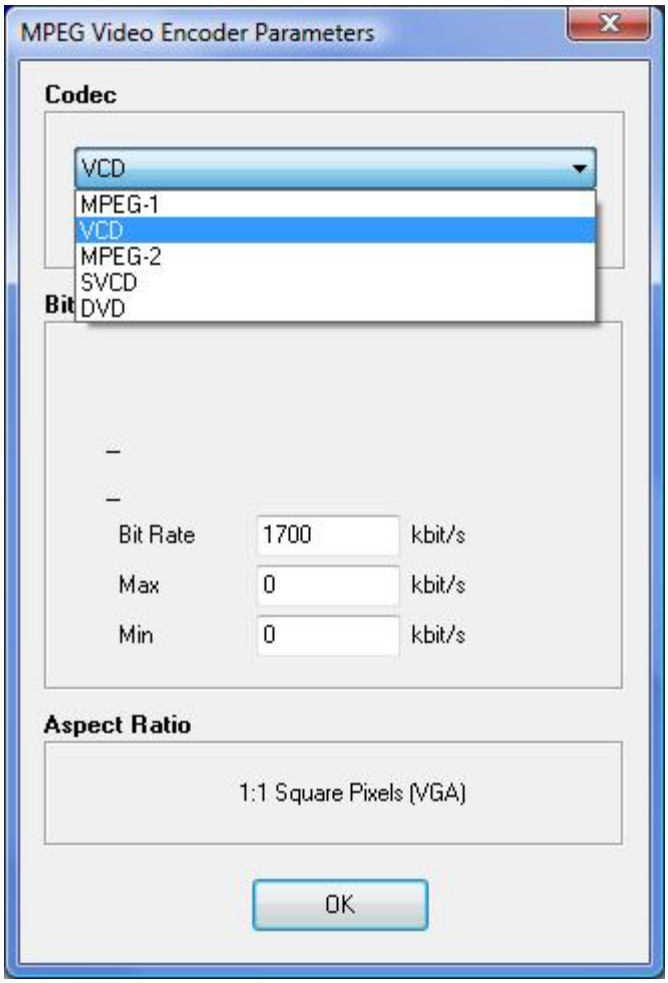
The AVI parameters window has the most options. The first thing to take notice of is the colored circles on the leftmost column. Only those colored GREEN are fully supported. Blue and yellow may work but will likely have issues and red is not recommended. With each CODEC type, additional options may be available.



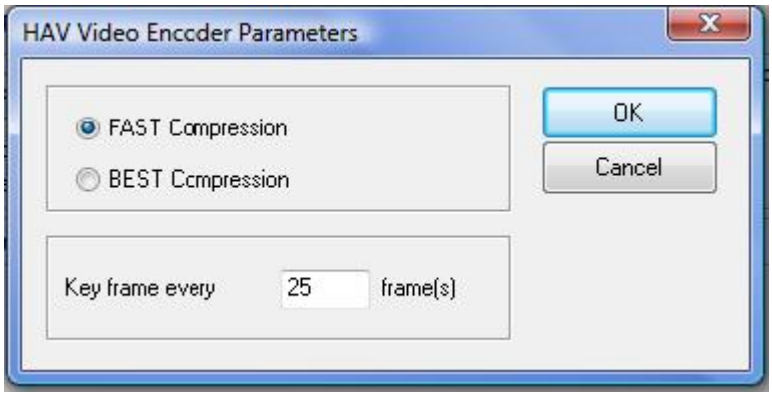
The Windows Media Video (WMV) CODEC has only one parameter which is not labeled (this is a Windows generated panel). The parameter is bit rate which defaults to 1000 and has a max value of 8,000. Higher values give higher quality and a larger file size.



The MPEG encoder parameters provides several useful options including DVD.



The HAV format has only two types and the ability to set the key frames.



Creating .AVI Files

In addition to the ImagesToAnimation module, C Tech recommends a robust and easy to use program named VIDEOMACH for creating and playing back video. The only capability in VideoMach that is not in EVS-PRO's ImagesToAnimation module is the ability to add sound to animations.

VIDEOMACH is shareware and Imagen is freeware.

Both of these products are very simple to use and offer significant advantages over the status-quo. VideoMach is a combination of video format converter and high-performance image/video processor. In many segments it can replace complicated video editors and post-processing tools, although it is not a video editor. VIDEOMACH is so easy that you can start working with digital video soon after installation, without consulting manuals or help files.

VIDEOMACH makes it easy to:

1) join captured video clips, animations and photos into a single video file (for example, you can use it to create AVI or FLIC out of images)

2) grab and save individual frames from a movie

3) rotate, crop, resize and resample image/video files

4) change brightness, contrast, color depth, frame rate and compression

5) apply a large number of image manipulation functions and filters

Features:

1) supported multimedia formats: BMP, DIB, RLE, GIF, JPG, PCX, TGA, AVI, FLC, FLI, HAV

2) built-in image functions: Vertical and Horizontal Mirror, Negative, GrayScale, Saturation, Brightness, Contrast, Adjust RGB, Gamma Correction, Posterize, Solarize, Equalize, Soften, Blur, Sharpen, Emboss, Find Edges, Edge Enhance, Trace Contour, Despeckle, Chaos, Uniform and Gaussian Noise, Interference Noise, Hue, Oil Painting, Count Unique Colors, Crop, Verical and Horizontal Flip, Resize, Resample, Swap Red and Blue Components and Rotate

3) automatic loading/saving of image sequences (numbered images)

4) Very Fast

5) year 2000 compliant

Download VIDEOMACH here: http://www.gromada.com/VideoMach_dl.html

Also, Gromada makes a better multimedia player. It has several advantages over Microsoft's players. Most notable are the ability to "play at maximum speed", "change the speed of playback" and my favorite "play image sequences" without converting

Imagen. It's a multimedia player mainly developed to support our HAV format including some special features very useful to most of video producers. Here is the quick list of what it offers:

* support for a total of 14 different formats (i.e. AVI, HAV, FLIC, MPEG-1, JPEG, PNG, TIFF, BMP, SGI etc.)

* capability to play image sequences (sequentially numbered images) which are frequently generated by many video rendering programs or video capture devices

* optionally it can be called with command line options (e.g. play and close)

* Imagen is freeware, use it for your private or commercial purposes with no restrictions

Please note that this is a first public release (v0.90) and still under development.

Direct link to Imagen package:

http://www.gromada.com/!JumpHtml(`http://www.gromada.com/

image_transition

General Module Function

image_transition receives two images, each coming from Read_Image and is used to transition from one image to another in any one of a variety of methods, including image fades and wipes.

Module Input Ports

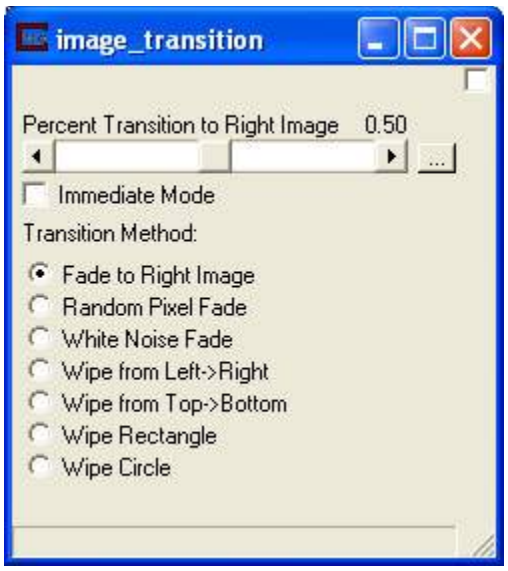
image_transition has three ports

The first two input ports each accept images from Read_Image.

The third input port accepts a float value between 0 and 1 which indicates the position of the Percent Transition to Right image slider.

Module Output Ports

image_transition has one output port. The output port outputs a new image that is the blended transition of the two input images. The resolution will match the resolution of the input fields. This is typically connected to the DisplayImage module for viewing, a mapper for texture mapping, or the Output_Images module.



Module Control Panel

The control panel for image_transition is shown in the figure above. The user interface shown above controls the method and amount of transition from image from the left input port to the right input port.

The ***Percent Transition to Right Image*** slider adjusts how much of the right image will be visible. This slider is affected by the Immediate Mode toggle, which allows changes to the image to occur as the slider is changed in real time.

The Transition Method can be any of the following:

***Fade to Right Image*:** Fade from the left image to the right image, where any pixel is a percentage of the right image as specified by the slider.

***Random Pixel Fade*:** Replace a random percentage of pixels from the left image with ones from the right image, roughly following the percent transition slider. This is a reproducible random effect, so once a pixel appears as the right image, moving the slider to the right will never cause it to switch back to the left image.

White Noise Fade: Pixels are randomly chosen from the right image, with the number of pixels in the right image roughly following the position of the slider. This method is non-reproducible, in that each time it is run, different pixels will be chosen. This is very similar to the above fade, but noticeably more "noisy".

***Wipe from Left->Right*:** Perform an image wipe, from the left edge of the image to the right edge, replacing the left image with the right.

***Wipe from Top->Bottom*:** Similar to above, except wipe from the top of the image down.

***Wipe Rectangle*:** Similar to above, except wipe outward from the center of the image in a rectangle following the aspect ratio of the original image.

***Wipe Circle*: Similar** to above, except wipe outward from the center of the image in a circle.

The*Reverse Wipe Direction* toggle reverses the direction a wipe is performed. If Wipe from Left->Right is the current method, and this toggle is on, the wipe will be performed from right to left.

Related Modules

->read_image

Read_Image

General Module Function

Read_Image reads an image file of a specified type and converts it into an 2D uniform field. It can optionally perform some basic image manipulation upon the read image. It also automatically looks for a world file or GCP file associated with the image in the same folder. Since this is done automatically, the world file name must conform to industry standards.

The naming conventions for the world file require that the world file must have the same base name and the extension must be the first and last letter of the suffix followed by a 'w'. For example:

Image World file GCP file

801022_ot.jpg 801022_ot.jgw 801022_ot.gcp

coasth.png coasth.pgw coasth.gcp

sitecrop.tif sitecrop.tfw sitecrop.gcp

If you world or gcp file name does not conform to this standard, you need to rename the file.

World Files have six lines that should contain the following information:

FIRST LINE: The dimension of a pixel in map units in the x direction
SECOND LINE: 0 (the rotation factor for rows - in this case none)
THIRD LINE: the rotation factor for columns- in this case none)
FOURTH LINE: The dimension of a pixel in map units in the y direction. The y-scale is usually the negative value of the x-scale.
FIFTH LINE: The x coordinate of the center of the upper-left pixel in map units.
SIXTH LINE: The y coordinate of the center of the upper-left pixel in map units.

WARNING: If the image portion of a georeferenced file is cropped or resized, the corresponding World file must be appropriately edited.

The world file information is passed to the texture_map module to georeference the image mapped on objects.

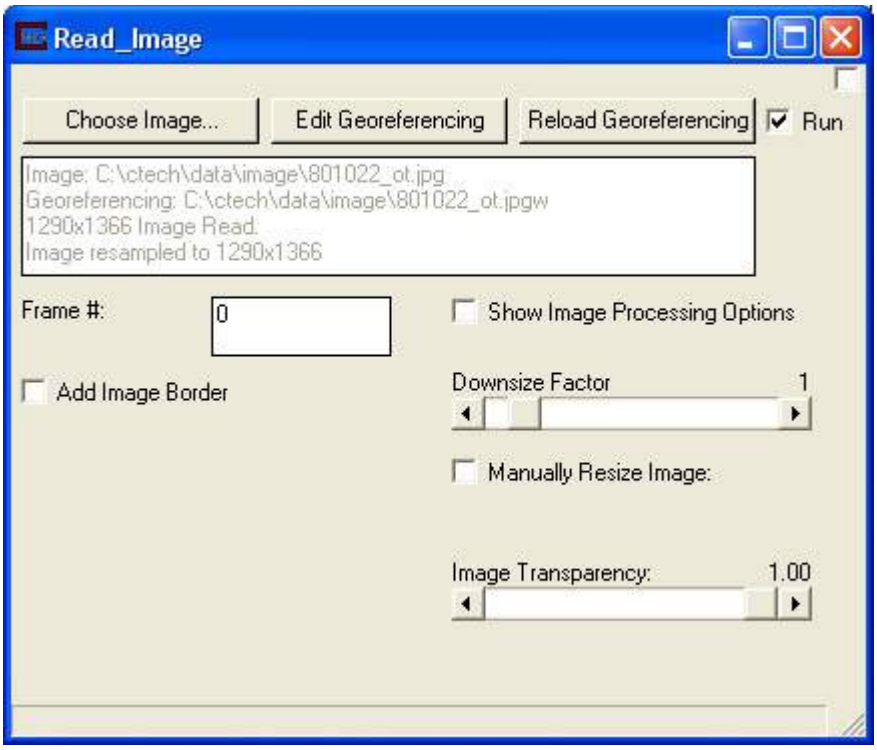
If you do not have a World file or GCP file, but are able to identify the coordinates (real world x-y) associated with objects (pixels) in your image, click on the **EditGeoreferencing** button to open the standalone utility program Georeference Image which can create world files or .gcp (ground control point) files for images.

Module Input Ports

Read Image has no input ports. It obtains the input data by reading a file with a file browser

Module Output

Read_Image has one output port. The port passes the mesh and data components to other modules which accept uniform field data types. Data passed from Read_Image is usually either viewed with DisplayImage or sent to modules in the Subsetting and/or Processing Libraries. The x and y nodal dimensions of the field will equal the pixel dimensions of the image and the values assigned each node are obtained from the RGB value of each pixel. The second port is used to send the 2D uniform field data directly to the viewer for rendering.



Module Control Panel

The user interface (shown above) allows selection of a file, a toggle for reversed scan line images, and various options for choosing image filters to apply as the image is read.

The ***Manually Resize Image*** toggle makes two inputs available to resize the image to any specified resolution as it is read.

The ***Frame #:*** input field is used for reading from animations. An animation file can be specified (such as a HAV or AVI file), and a specific frame read as the image.

The ***AddImageBorder*** toggle when selected will make additional controls visible for:

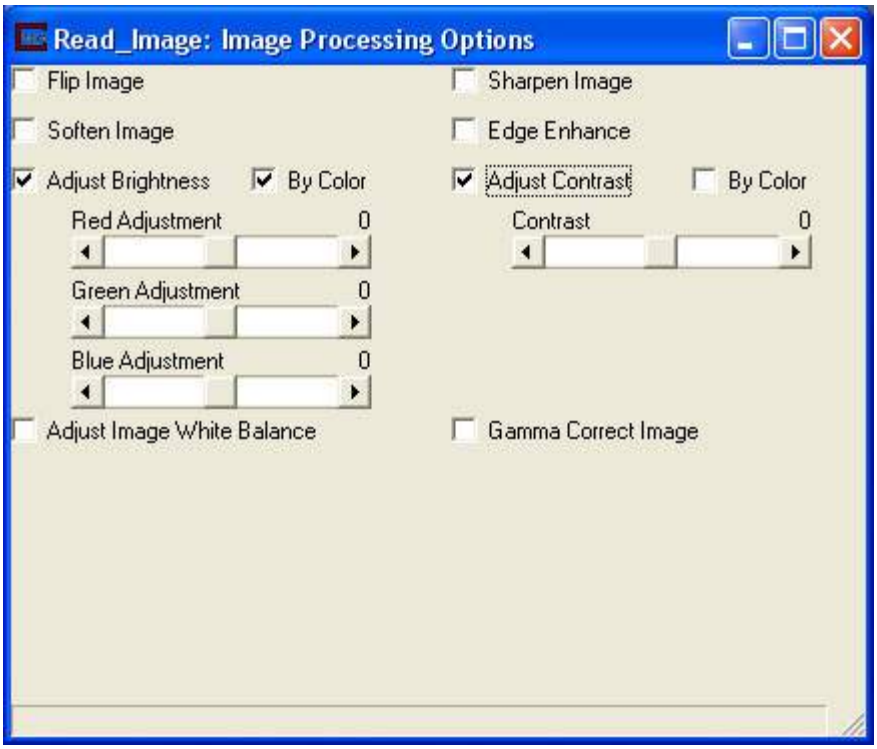
Border Color: This button allows you to set the color of a border applied to the image. This border is useful when your image is smaller than your object's (being texture mapped) extent. This prevents ugly streaking.

Border Transparency: Allows the border to become transparent. Transparent texture maps make the surface (or object) disappear or become transparent.

Border Size: Larger borders are necessary if the object being texture mapped has coarse resolution. You can tell if you need a large border if the border appears saw-toothed.

***ImageTransparency*:** Controls the transparency of the entire image. Transparent texture maps make the surface (or object) disappear or become transparent.

The ***ShowImageProcessing Options*** toggle when selected will open an addition window shown below with additional controls.



The ***Flip Image*** toggle when selected will invert the image about the x axis, i.e. in the vertical direction. The renderers assume that the image origin is at the lower left. Use this toggle if the image is rendered upside-down.

The ***Soften Image*** toggle makes the image smoother by applying low intensity pixel averaging.

The ***Sharpen Imag***e toggle increases the visibility of details on the image.

The ***Edge Enhance*** toggle enhances the edges of the objects and discards other image information.

Sliders are available to adjust the image ***brightness***, ***contrast***, or ***gamma correct*** the image. When Adjust Brightness or Adjust Contrast is selected, a new toggle will appear beside the first labeled By Color. This allows the image brightness or contrast to be adjusted on a per-color basis.

File types: The following image formats (with extensions) are currently supported:

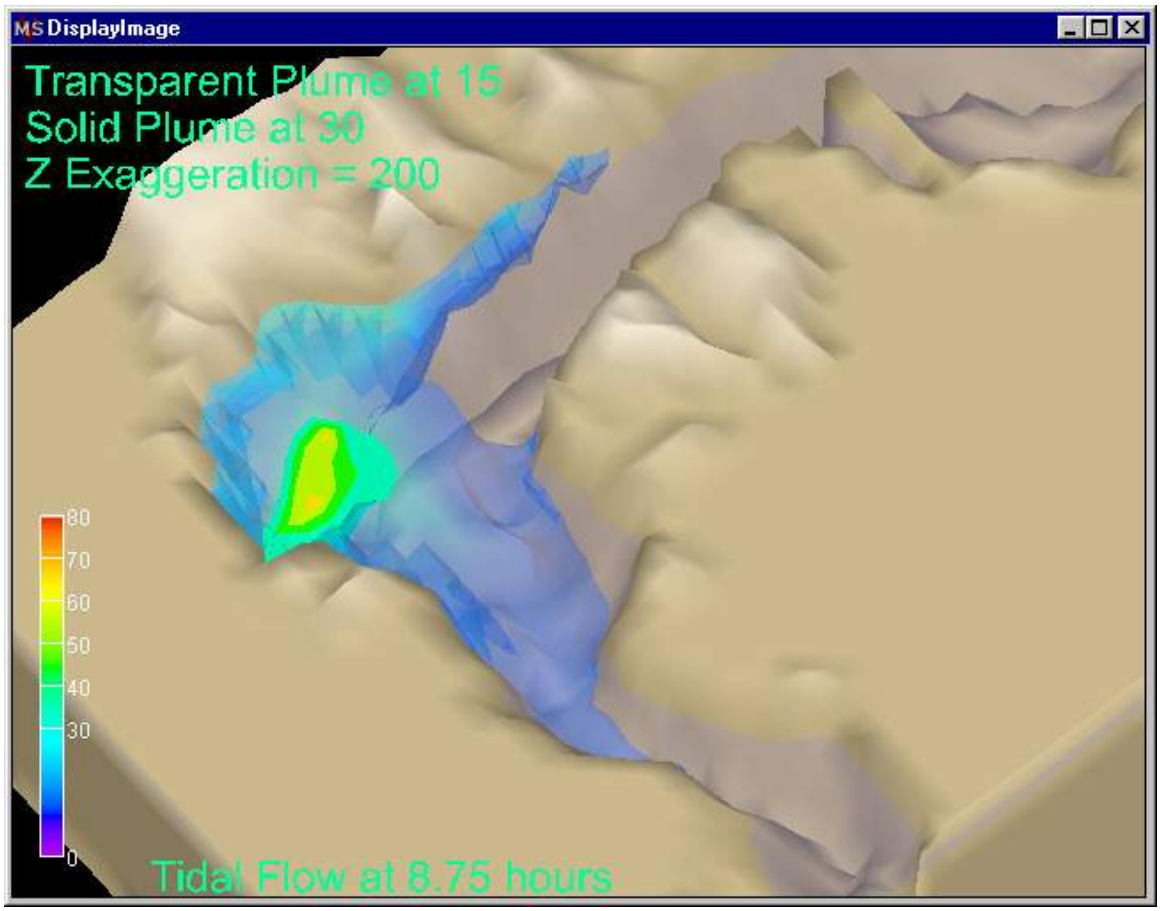
* Portable Network Graphics (png)
* Windows Device Independent Bitmap 8/15/24 bit (bmp, rle, dib)
* JPEG Joint Photographic Expert Group, JFIF 1.02 (jpg, jpeg, jif, jiff, jpe, J)
* GIF Graphics Interchange Format (gif, giff)
* TGA Truevision Targa 8/15/24/32 bit (tga)
* TIFF Tagged Image File Format (tif, tiff)
* XPM X-Pixmap (xpm)
* BAY Bayer Image (bay)
* FLC AutoDesk FLIC/FLIC-Pro, 8 bit (fli, flc)
* PCX Zsoft Paintbrush, 8/24 bit (pcx)
* PNM Portable Image 8/15/24 bit (pnm, ppm, pgm)
* RAS Sun Raster Image 8/24/32 bit (ras, sun)
* RGB Silicon Graphics Image 8/24/32 bit (rgb, rgba, bw, sgi)

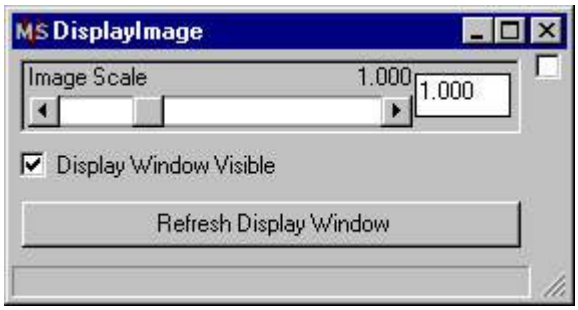
AnimationFile types: Single frames can be read from any of the following animation formats (with extensions):

* HAV High Quality Audio Video (hav)
* AVI Windows Audio Video Interleaved (avi)
* MPEG Motion Pictures Expert Group, Version 1 (mpg, mpeg, mpe, m1v)

DisplayImage

DisplayImage: This module is a viewer for images. It should only be connected to Read_Image. When images are connected to the Viewer (NOT RECOMMENDED), instead of DisplayImage, each pixel is rendered as a quadrilateral. This is VERY inefficient.





texture_cylinder

General Module Function

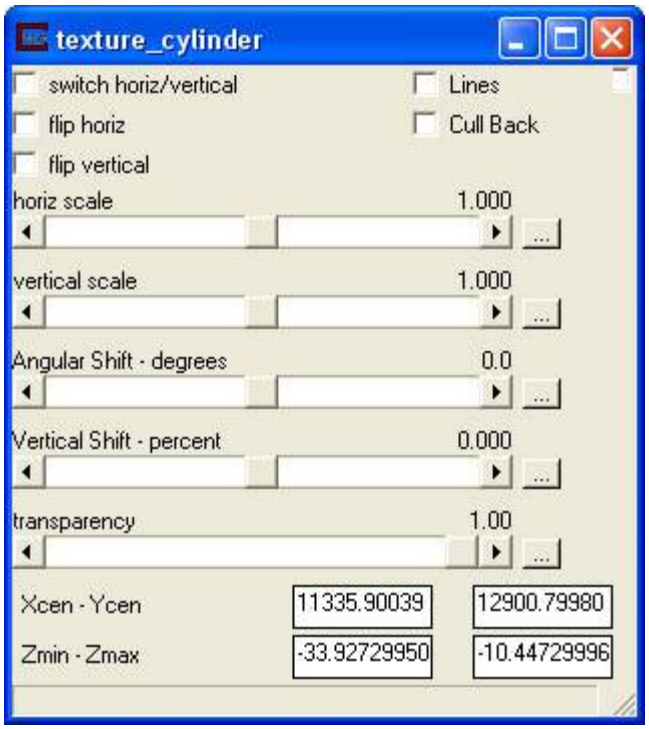
texture_cylinder provides a means to (texture map) project images onto a cylinder.

Module Input Ports

texture_cylinder has two input ports. The left input port should be a spheroid object with, and the right input port should be an image from the Read_TGA_BMP module.

Module Output Ports

texture_cylinder has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version directly to the Viewer.



Module Control Panel

The three toggles on the left provide a means to flip or mirror the image file for texture mapping. This is useful for quickly transforming an image file that has been produced using different scanning orders.

The Lines toggle determines whether lines are included in the output. The lines toggle will allow lines (such as a bounding box matching the image spatial extents) to be displayed.

Cull Back - A toggle for use with the transparency option. This feature may produce a better transparency when a surface is shown with other transparent surfaces, or if the same surface has high relief and is causing a confusing visualization. It's use will require experimentation to determine if it is appropriate.

horiz scale and vertical scale - Sliders for enlarging or shrinking the image. Horiz represents an East/West scaling, while vert represents North/South scaling

angular shift and vertical shift - Sliders for moving the image in Azimuth or vertical directions.

Transparency - A slider for adjusting the opacity of the texture mapped image. A value of 1.00 is completely opaque and a value of 0.00 is completely invisible.

Xcen - Ycen - Type-ins for the X-Y center of the cylinder (pseudo cylindrical object) to be texture mapped.

Zmin - Zmax - Type-ins for the minimum and maximum Z extents of the texture mapped image.

texture_sphere

General Module Function

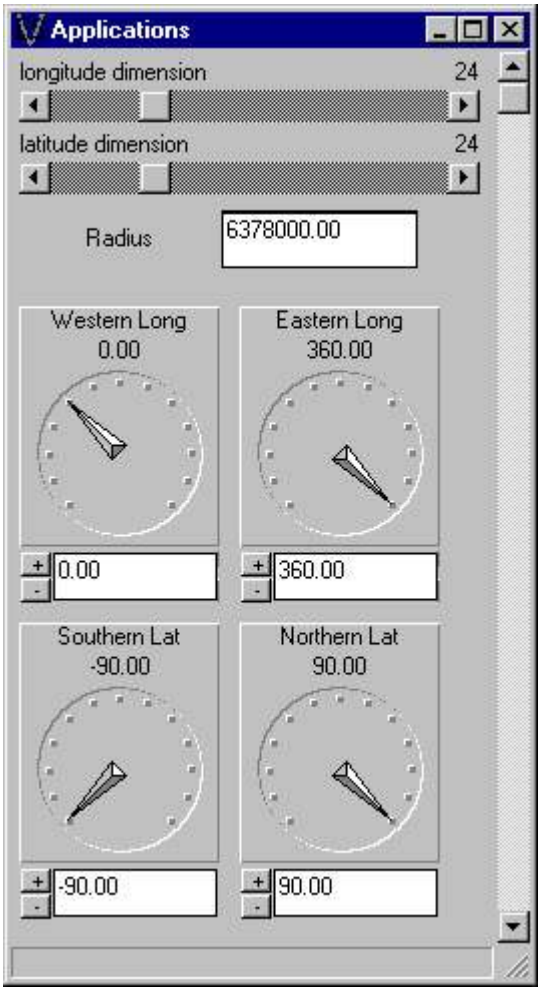
texture_sphere provides a means to (texture map) project images onto a sphere.

Module Input Ports

texture_sphere has two input ports. The left input port should be a spheroid object with, and the right input port should be an image from the Read_TGA_BMP module.

Module Output Ports

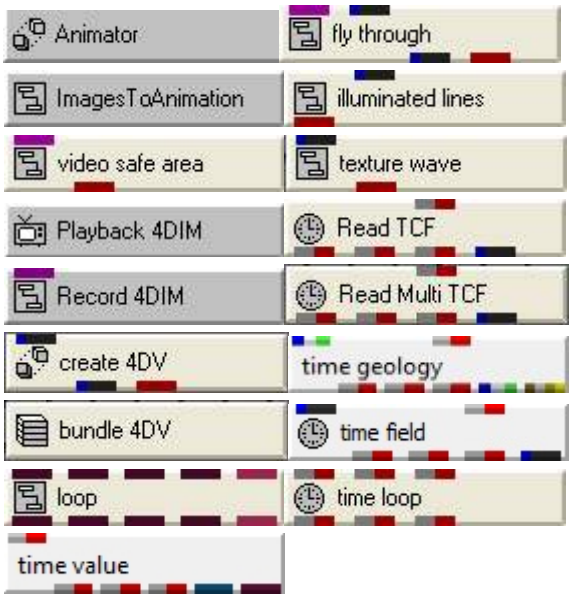
texture_sphere has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version directly to the Viewer.



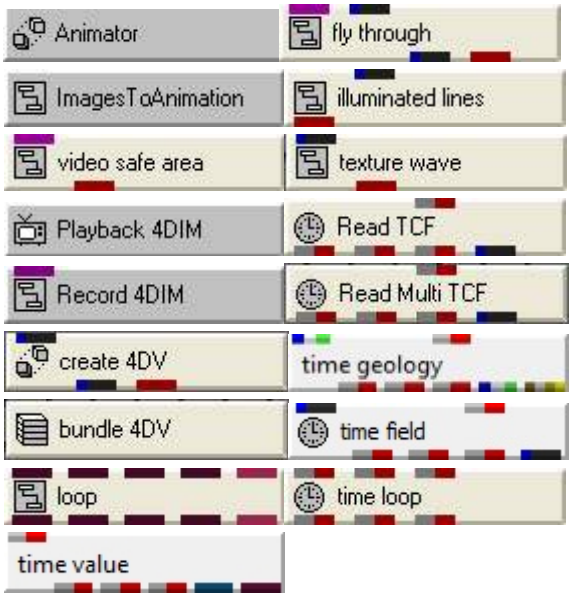
Module Control Panel

The control panel for texture_sphere is shown in the figure above. The three toggles on the left provide a means to flip or mirror the image file for texture mapping. This is useful for quickly transforming an image file that has been produced using different scanning orders. The Lines toggle determines whether lines are included in the output. The lines toggle will allow lines (such as a bounding box matching the image spatial extents) to allow lines to be displayed. If the input surfaces have nodal data, the texture map will be overlayed on colored surfaces. This is generally not desirable. The No Data toggle will remove the nodal data before texture mapping.

Animation Modules



Animation Modules



Animator

This module is available only in EVS-PROand MVS

General Module Function

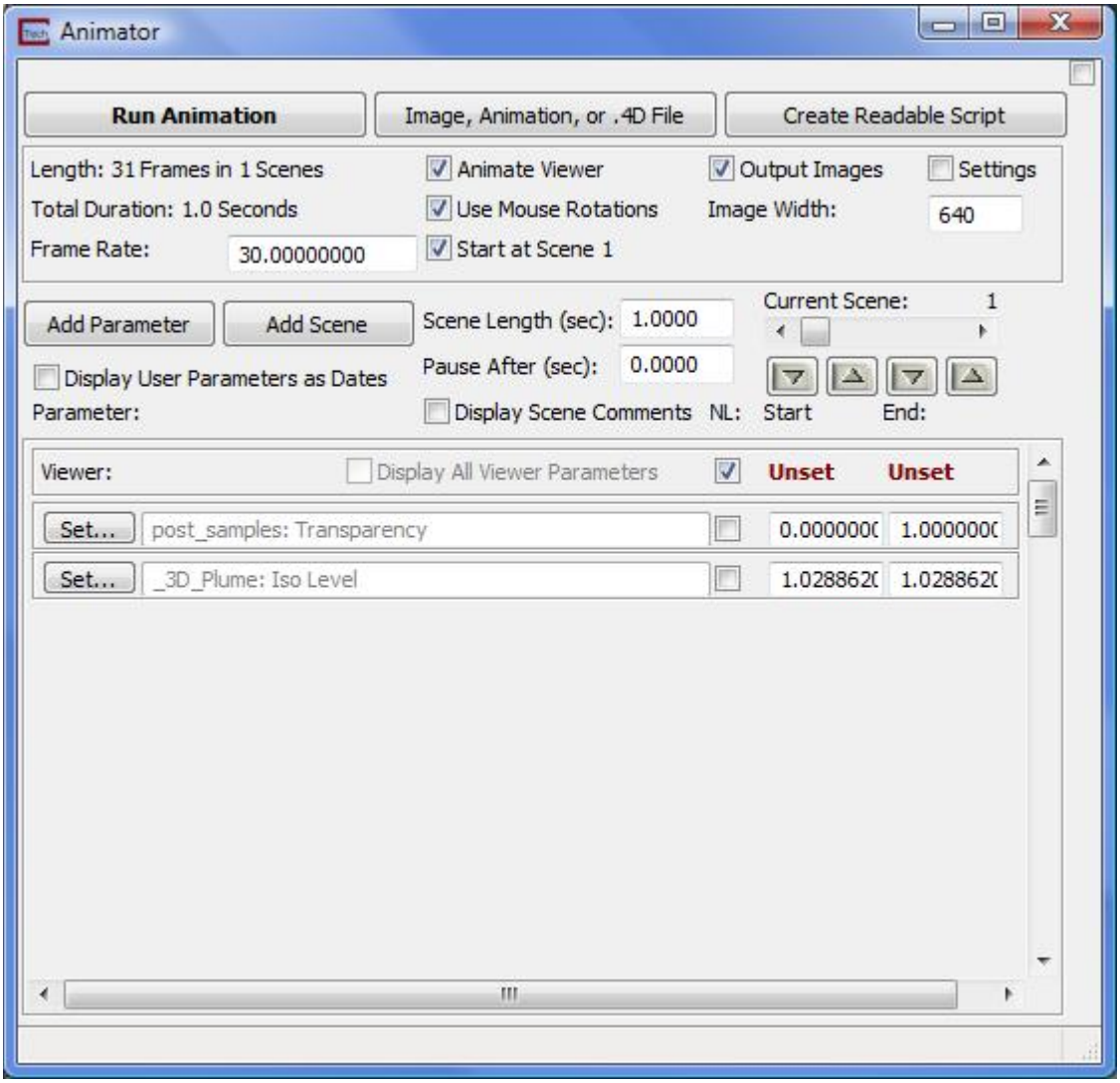
The Animator module is now tightly coupled with your application. It is aware of those modules and their associated parameters in the application. It allows you receive the values or set the values in each module without the need for typing.

Module Input&Output Ports

Animator has no input or output ports.

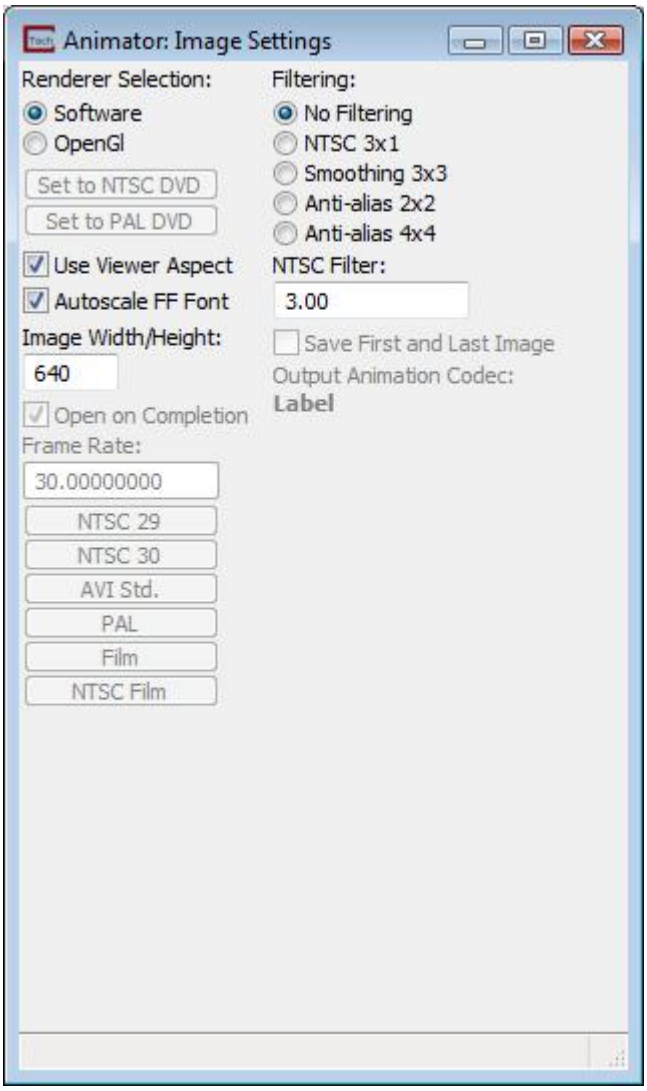
IMPORTANT NOTE: Be sure to **TURN OFF** "Animate Viewer" in the Animator module if you're controlling fly_through with the Animator.

Module Control Panel



The control panel for Animator is shown in the figure above. Please note that the example figures in this help topic show the Animator with settings for a specific application. Each of the user parameters is discussed below**.**

1. **Run Animation**: This button causes the Animator to run, controlling all modules, the viewer and creating images or an animation file (e.g. avi, hav, etc) as specified.
2. **Image, Animation, or .4D File**: This button opens a browser allowing you to specify the type of output desired and the folder. The suffix chosen determines the type of output (image sequence, animation or 4DIM file). For images you specify an image "base" name to create a sequence of image files. Any supported image file type may be specified though we strongly recommend using PNG (.png). The name specified should be of the form base.png (or other suffix such as .bmp, .tif, .tga, etc.) and the files created will be base0001.png through base0101.png (for 101 total frames).
3. **Create Readable Script**: This button causes the Animator to run, controlling all modules, the viewer and creating images or an animation file (e.g. avi, hav, etc) as specified.
4. **Length:** is a text output telling you the total number of frames and scenes in your animation.
5. **Total Duration** is the duration of the animation in seconds.
6. **Animate Viewer** is a toggle which determines if the Viewer will be controlled in the animation.
7. **Use Mouse Rotations:** When the toggle is on, each scene ending viewer position can be set by the actual view as determined by your mouse rotations and/or Az-El selections. BE SURE TO SET CAMERA AUTO-NORMALIZE TO **NONE**.
8. **Output Images / Output Animation / Record 4DIM** is a toggle which determines if an animation file will be created during playback. Turn this off during test playbacks if you just want to watch the progress in the Viewer. The name of this toggle depends on the type of file selected for the image or animation.
9. **Image Width** is a type-in that sets the basic size for images or bitmap animations. Unless overridden in Settings, the height is determined by the aspect ratio of your Viewer.
10. **Settings** is a toggle which opens the window below:



a) **Renderer Selection** chooses the method for creating images for image output or integration into an animation file.

- *Software* uses C Tech's software renderer and
- *OpenGL* uses your graphics card's OpenGL renderer

b) **Filtering** selects any filtering options for each image frame. For computer playback *No Filtering* is the most common option and for output to DVD (interlaced) or other television video, the *Anti-alias 4x4* option is best.

c) **NTSC Filter** is a type in which controls the "blurring" for the 4x4 option.

d) **Set to NTSC DVD** is a button which sets the recommended settings for NTSC DVD output.

e) **Set to PAL DVD** is a button which sets the recommended settings for PAL or SECAM DVD output.

f) Use Viewer Aspect will cause the Image Height to be generated based upon the value in the Image Width field and the Aspect Ratio of the Viewer. If this toggle is not selected an Image Width and Image Height should be set.

g) Autoscale FF Font this toggle will cause all Forward Facing text to be scaled according to the size of the Viewer.

h) Image Width & Height: These type-ins specify the size of the image file to create if an image type (not .4d) file name is specified with the "Image or 4D file" button (these are connected to the same type-ins in the main window). If the Use Viewer Aspect toggle is selected the Height field will be generated based upon the Image Width field and the Aspect Ratio of the Viewer.

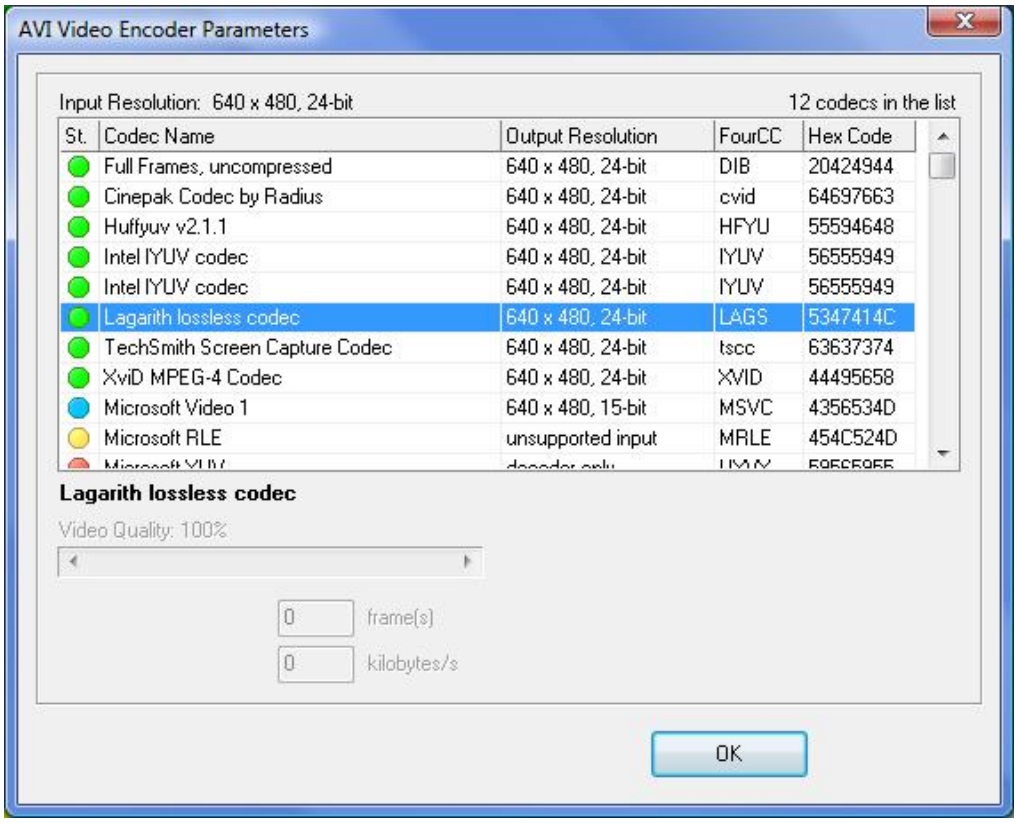
i) Open on Completion is a toggle which will cause the animation to open in Windows Media Player (or your default application) upon completion.

k) Save First and Last Image is a toggle which will save the first and last image. When an animation format is selected (e.g. AVI) each frame of the animation is not saved. This toggle causes the first and last frames to be saved. Often in video editing software (e.g. Adobe Premier Pro) it is useful to have the first and last frames if you want to create a pause before and/or after the animation sequence.

l) Frame Rate: This type-in specifies the frame rate in frames per second for the animation. A lower frame rate creates a smaller but jerkier animation. For playback on computers, typical frame rates are 10-20 frames per second. The buttons below set commonly used frame rates. (this is connected to the same type-in in the main window)

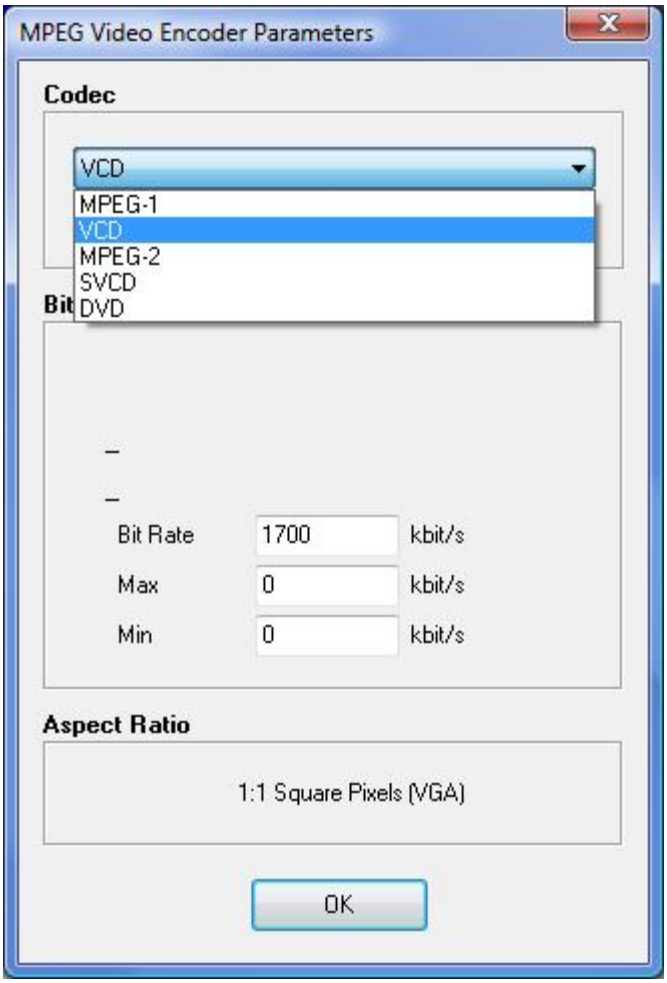
m) Output Animation Codec: This shows the selected default CODEC for the type of animation file you've created. There are four supported bitmap animation file types which are AVI, WMV, MPG (MPEG), and HAV.

The AVI parameters window has the most options. The first thing to take notice of is the colored circles on the leftmost column. Only those colored GREEN are fully supported. Blue and yellow may work but will likely have issues and red is not recommended. With each CODEC type, additional options may be available.

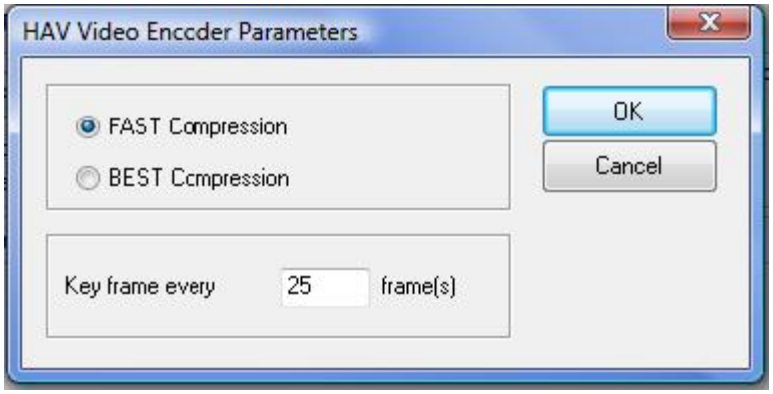


The Windows Media Video (WMV) CODEC has only one parameter which is not labeled (this is a Windows generated panel). The parameter is bit rate which defaults to 1000 and has a max value of 8,000. Higher values give higher quality and a larger file size.

The MPEG encoder parameters provides several useful options including DVD.



The HAV format has only two types and the ability to set the key frames.



to use from among those installed on your computer. Only a few CODECS are included with windows. Others are available with various software packages. This tries to default to TechSmith Screen Capture Codec, or, if that doesn't exist, to the HuffYUV codec (both are lossless). If neither exists, the first one in the list will be selected.

11) Frame Rate: This type-in near the top left corner specifies the frame rate in frames per second for the animation. A lower frame rate creates a smaller but jerkier animation. For playback on computers, typical frame rates are 10-20 frames per second.

12) Start at Scene 1 is a toggle which forces Run Animation to start at your first scene. Otherwise it begins at the Current Scene as specified in the slider.

13) Image Width & Height: These type-ins specify the size of the image file to create if an image type (not .4d) file name is specified with the "Image or 4D file" button.

[create_animation_panel4.jpg]

14) Add Parameter: This button causes a window to appear where you can specify all available parameters and properties for each controllable module in your application. The window is shown above.

15) Add Scene: This button appends a new scene to your animation. The initial values for every parameter will be obtained from the current values in your application.

16) Scene Length sets the length of the current scene in seconds. This value multiplied by the frame rate equals the total number of frames for the scene.

17) Current Scene is a slider to specify the scene for which you want to see or set values.

18) Pause After sets a pause duration in seconds after the current scene. This is an efficient way to create a pause in your animation.

19) Display User Parameters as Dates is a toggle intended for temporary use to see any parameters that represent dates as a date format.

20) Display Scene Comments: This is a toggle that opens a large type-in that lets you describe the purpose or theme of the scene. These comments are included in the Readable Script. This is shown in the figure above.

21) NL are toggles for each module parameter that determine if the interpolation of the Start and End values should be non-linear. The ***NL*** toggle Non-linear interpolation provides "soft" acceleration when moving or rotating objects (or interpolating any other feature). Instead of linear interpolation, a SINE function is used so the slope (rate of change of the parameter) is zero at the beginning and end. Note: By default, this is on for the Viewer since the behavior (in our opinion) is advantageous for rotations and other Viewer manipulations, however for all other parameters it is off by default.

22) Start: The Start type-in is the initial parameter value for the sequence.

23) End: The End type-in is the final parameter value for the sequence.

22)  **Down Button** This button transfers the current value of all animation parameters in each referenced module into the Start or End

values of the Animator. In other words, this lets you set how all parameters for the Start or End of the current Scene.

23) **Up Button** This button transfers the current Start or End values in the Animator to all animation parameters in every referenced module. In other words, this lets you see how your Viewer will look at the Start or End of the current Scene.

ImagesToAnimation

General Module Function

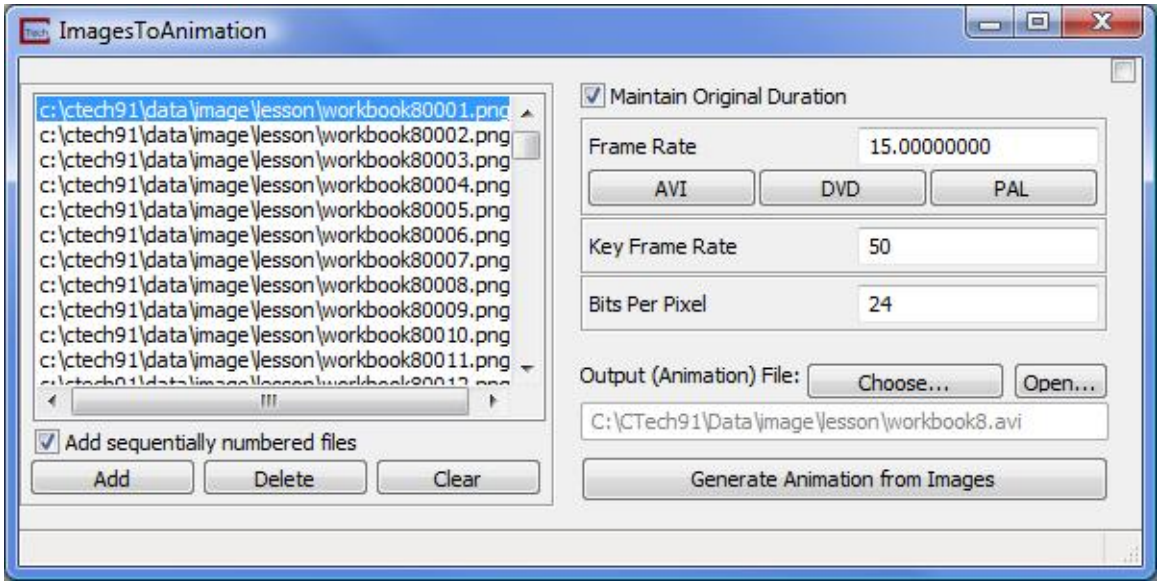
ImagesToAnimation incorporates the functionality of Gromada's VideoMach into an easy to use module.

Animation CODECS try to default to TechSmith Screen Capture Codec, or, if that doesn't exist, to the HuffYUV codec (both are lossless).

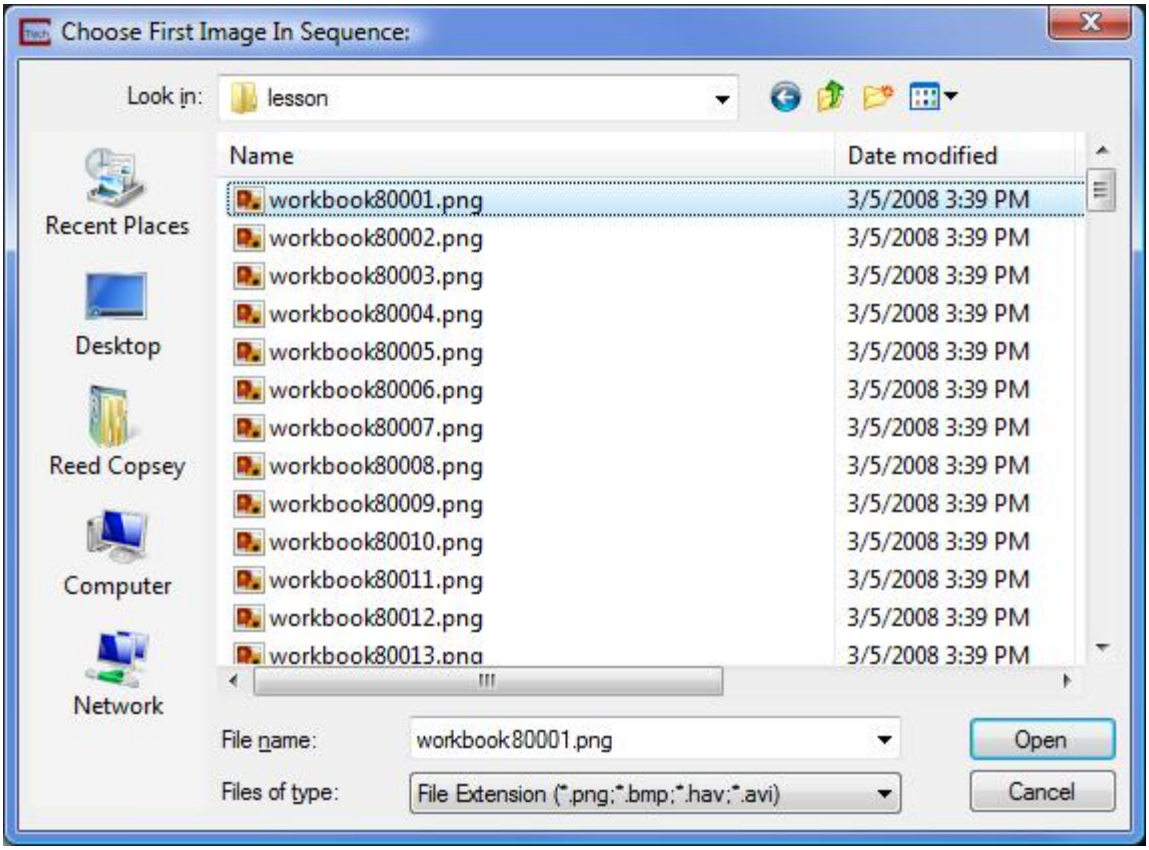
ImagesToAnimation supports several animation file formats including:

1. AVI Windows Audio Video Interleaved (avi)
2. Windows Media Video - WMV produces some of the smallest files and this output is virtually guaranteed to run on any up to date Windows computer. However, the quality can be poor unless the data rate is increased.
3. MPG Moving Pictures Expert Group, MPEG-1 (mpg,mpeg)
4. HAV High quality Audio Video (hav) HAV is a format that can be played with the freeware program Imagen (formally HAV player). This format has some distinct advantages, specifically it uses lossless compression. This results in the highest quality output. Surprisingly, HAV files are often as small as or smaller than lower quality AVI or MPG files.

First, instance the ImagesToAnimation module. The user interface for it is shown below.



Click on the Add button to bring up the panel to specify your image files.



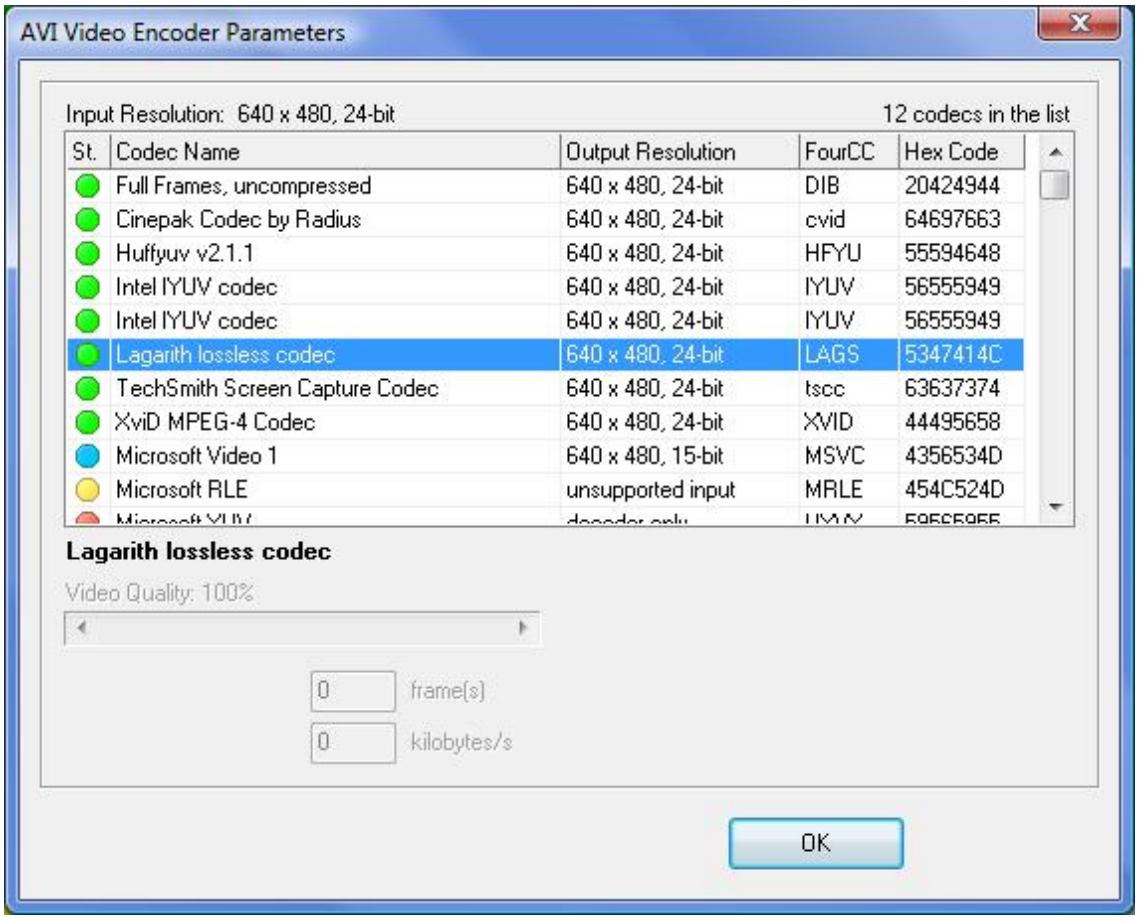
Click on the first file of the sequence you created. You only need to select the first file, then it will automatically select and process all files in the numbered sequence. You also need to "Choose" the Output (Animation) File. The suffix you choose determines the type of animation file to be created. The choices are hav (lossless high quality Gromada format), avi (windows standard) and mpg/mpeg (MPEG 1 format). The example above is set to workbook8.avi

Click on the "Generate Animation from Images" button to make an AVI file by using the input data (your PNG files).

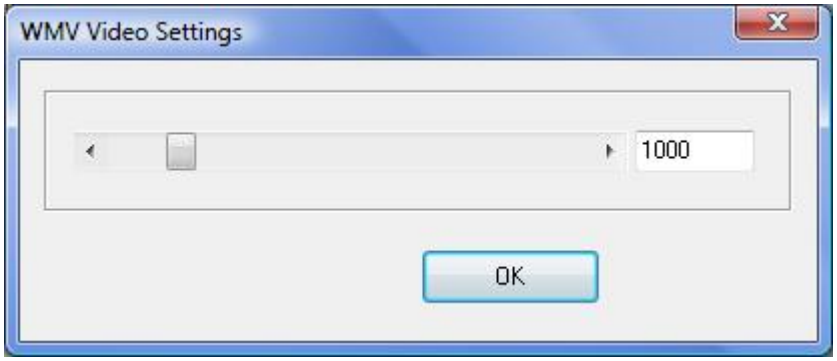
When you do, the first thing that will happen is a window will pop up which allows you to set the Output Animation Codec:

This shows the selected default CODEC for the type of animation file you've created. There are four supported bitmap animation file types which are AVI, WMV, MPG (MPEG), and HAV.

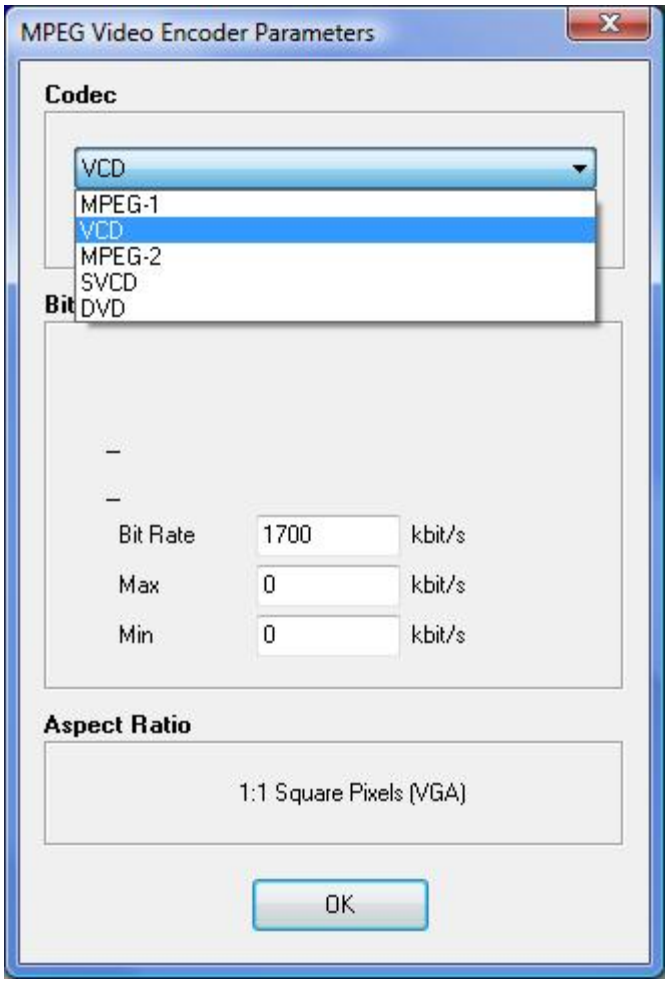
The AVI parameters window has the most options. The first thing to take notice of is the colored circles on the leftmost column. Only those colored GREEN are fully supported. Blue and yellow may work but will likely have issues and red is not recommended. With each CODEC type, additional options may be available.



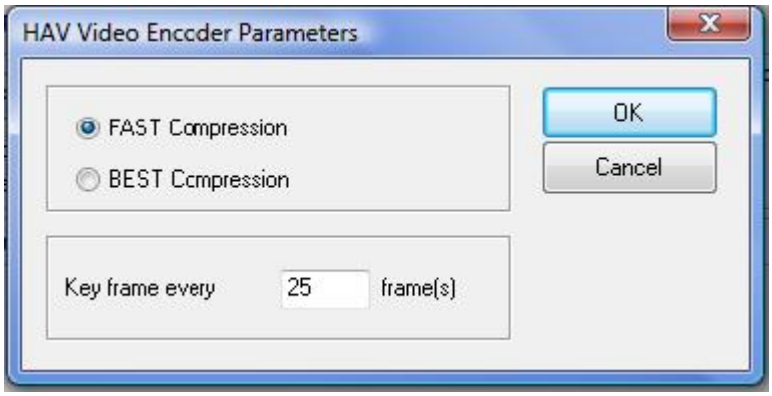
The Windows Media Video (WMV) CODEC has only one parameter which is not labeled (this is a Windows generated panel). The parameter is bit rate which defaults to 1000 and has a max value of 8,000. Higher values give higher quality and a larger file size.



The MPEG encoder parameters provides several useful options including DVD.



The HAV format has only two types and the ability to set the key frames.



Creating .AVI Files

In addition to the ImagesToAnimation module, C Tech recommends a robust and easy to use program named VIDEOMACH for creating and playing back video. The only capability in VideoMach that is not in EVS-PRO's ImagesToAnimation module is the ability to add sound to animations.

VIDEOMACH is shareware and Imagen is freeware.

Both of these products are very simple to use and offer significant advantages over the status-quo. VideoMach is a combination of video format converter and high-performance image/video processor. In many segments it can replace complicated video editors and post-processing tools, although it is not a video editor. VIDEOMACH is so easy that you can start working with digital video soon after installation, without consulting manuals or help files.

VIDEOMACH makes it easy to:

1) join captured video clips, animations and photos into a single video file (for example, you can use it to create AVI or FLIC out of images)

2) grab and save individual frames from a movie

3) rotate, crop, resize and resample image/video files

4) change brightness, contrast, color depth, frame rate and compression

5) apply a large number of image manipulation functions and filters

Features:

1) supported multimedia formats: BMP, DIB, RLE, GIF, JPG, PCX, TGA, AVI, FLC, FLI, HAV

2) built-in image functions: Vertical and Horizontal Mirror, Negative, GrayScale, Saturation, Brightness, Contrast, Adjust RGB, Gamma Correction, Posterize, Solarize, Equalize, Soften, Blur, Sharpen, Emboss, Find Edges, Edge Enhance, Trace Contour, Despeckle, Chaos, Uniform and Gaussian Noise, Interference Noise, Hue, Oil Painting, Count Unique Colors, Crop, Verical and Horizontal Flip, Resize, Resample, Swap Red and Blue Components and Rotate

3) automatic loading/saving of image sequences (numbered images)

4) Very Fast

5) year 2000 compliant

Download VIDEOMACH here: http://www.gromada.com/VideoMach_dl.html

Also, Gromada makes a better multimedia player. It has several advantages over Microsoft's players. Most notable are the ability to "play at maximum speed", "change the speed of playback" and my favorite "play image sequences" without converting

Imagen. It's a multimedia player mainly developed to support our HAV format including some special features very useful to most of video producers. Here is the quick list of what it offers:

* support for a total of 14 different formats (i.e. AVI, HAV, FLIC, MPEG-1, JPEG, PNG, TIFF, BMP, SGI etc.)

* capability to play image sequences (sequentially numbered images) which are frequently generated by many video rendering programs or video capture devices

* optionally it can be called with command line options (e.g. play and close)

* Imagen is freeware, use it for your private or commercial purposes with no restrictions

Please note that this is a first public release (v0.90) and still under development.

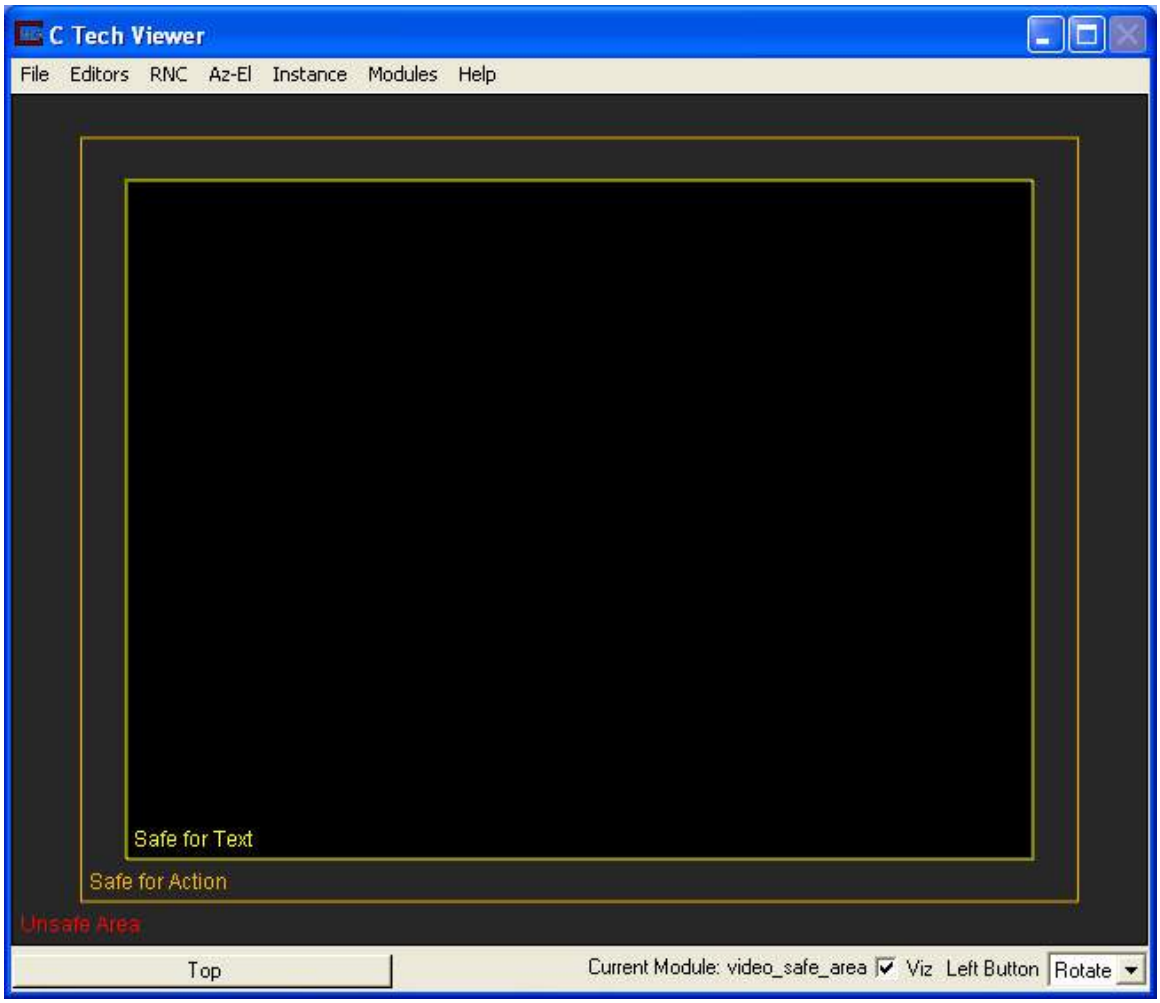
Direct link to Imagen package:

http://www.gromada.com/!JumpHtml(`http://www.gromada.com/

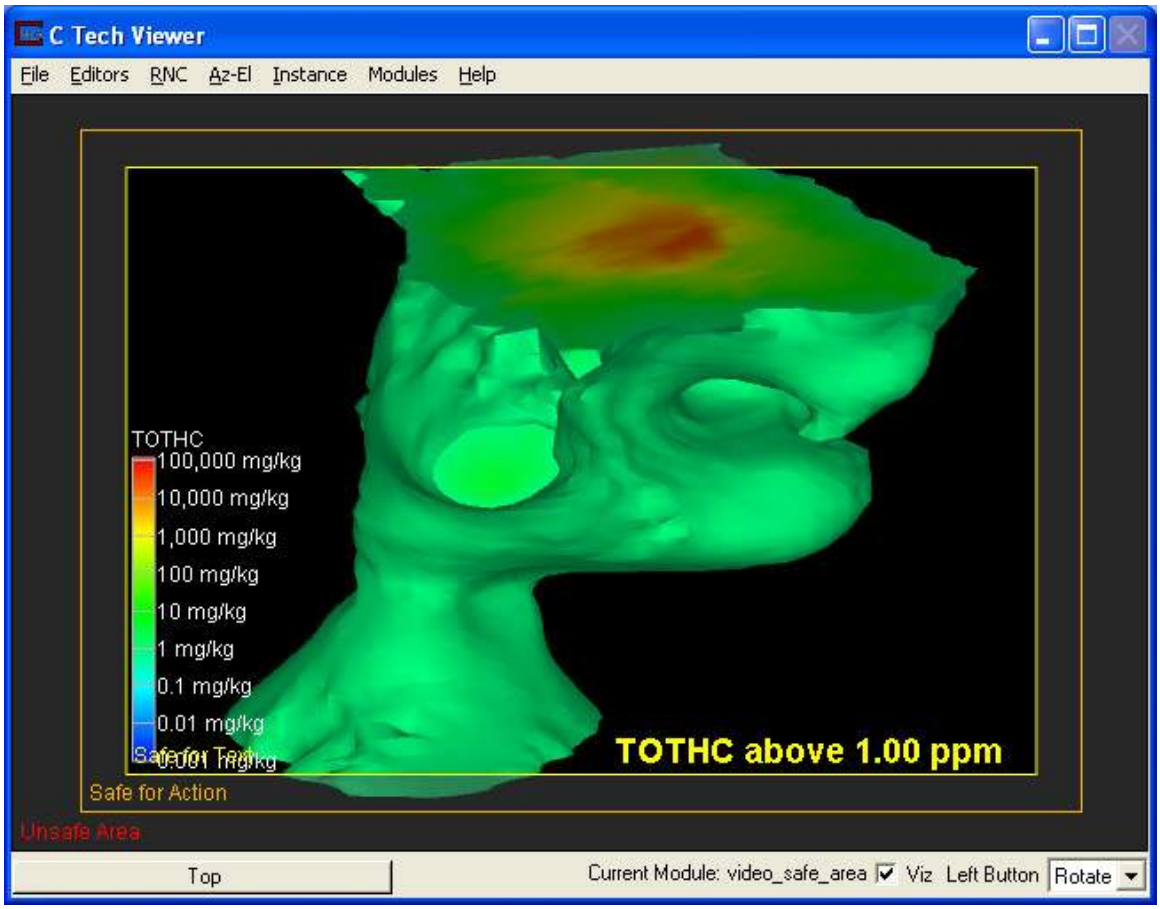
video_safe_area

General Module Function

The video_safe_area module is used when creating an animation for DVD or Video. It displays the areas that are usable for both text and animation purposes for several standard video formats. This allows you to properly setup your animation in order to get the best possible output on multiple television sets.

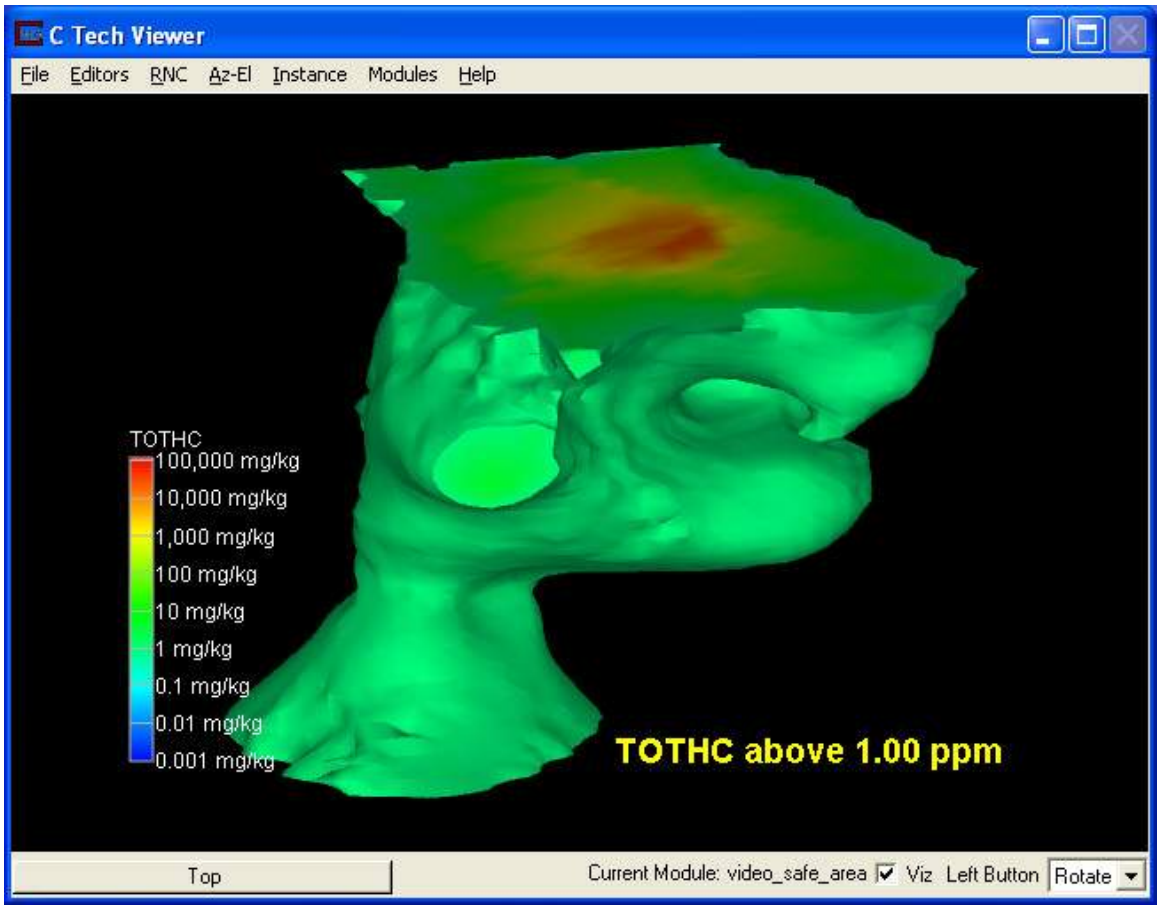


Notice in the picture below that the title is restricted the safe for text area, while the plume is displayed inside the safe for action border. Since it contains text and sharp lines, the legend should be placed inside the safe for text border.



NOTE: This module should have the visible toggle unchecked when the actual animation for video is produced to avoid having the safe areas displayed in the video.

Once you turn off the visible toggle, you will notice a large border around your content. This border is vital for proper output to a television.

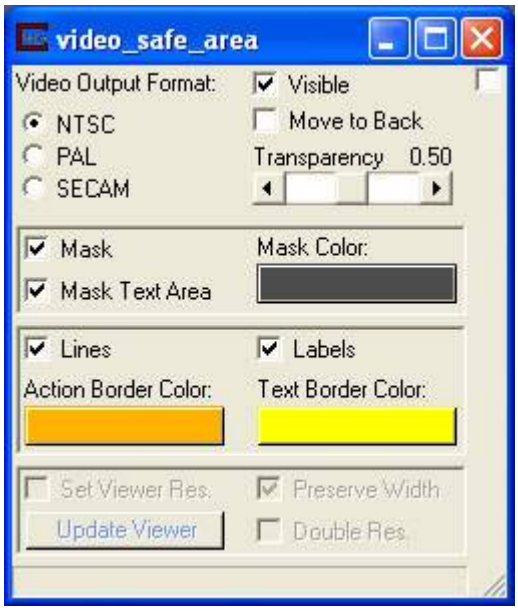


Module Input Ports

The video_safe_area module has one input port that accepts the output from a viewer module.

Module Output Ports

The video_safe_area module has one output port that connects to a viewer module's input and displays a viewable "safe area" in the viewer window.



Module Control Panel

The control panel for the video_safe_area module can be seen in the image above.

The **VideoOutput Format** changes the safe areas in the viewer window to match the default width and height values for the selected video format.

The **Visible** toggle turns the safe area display on and off. This toggle should always be off when making the actual video so the safe areas are not recorded.

The **Move to Back** toggle will put the safe area display behind any graphics in the viewer.

The **Transparency** slider changes the opacity of the safe area mask.

The **Mask** toggle turns the safe area masks on and off. The mask is a visual tool to help visualize which graphics fall into which safe area.

The **Mask Text Area** toggle turns the masking surrounding the text area on or off.

Mask Color alters the color of the masking.

The **Lines** toggle turns the lines defining the safe areas on and off.

The **Labels** toggle turns the labels defining the safe areas on and off.

The **Action Border Color** button selects the color of the action border.

The **Text Border Color** button selects the color of the text border.

Selecting **Set Viewer Res.** sets the resolution of the viewer to the default for the video format that has been selected.

If the **Preserve Width** toggle is selected when the Set Viewer Res. toggle is chosen, the current resolution width of the viewer will be maintained while the resolution height of the viewer will be based upon the appropriate ratio for the video format that has been selected.

If the Preserve Width toggle is unselected the **Double Res** toggle can be selected. The Double Res toggle will double the resolution of the viewer,

while keeping the appropriate width-height ratio for the video format that has been selected. This should only be used while using the Screen Renderer output of Output_Images with the 4x4 anti-aliasing option.

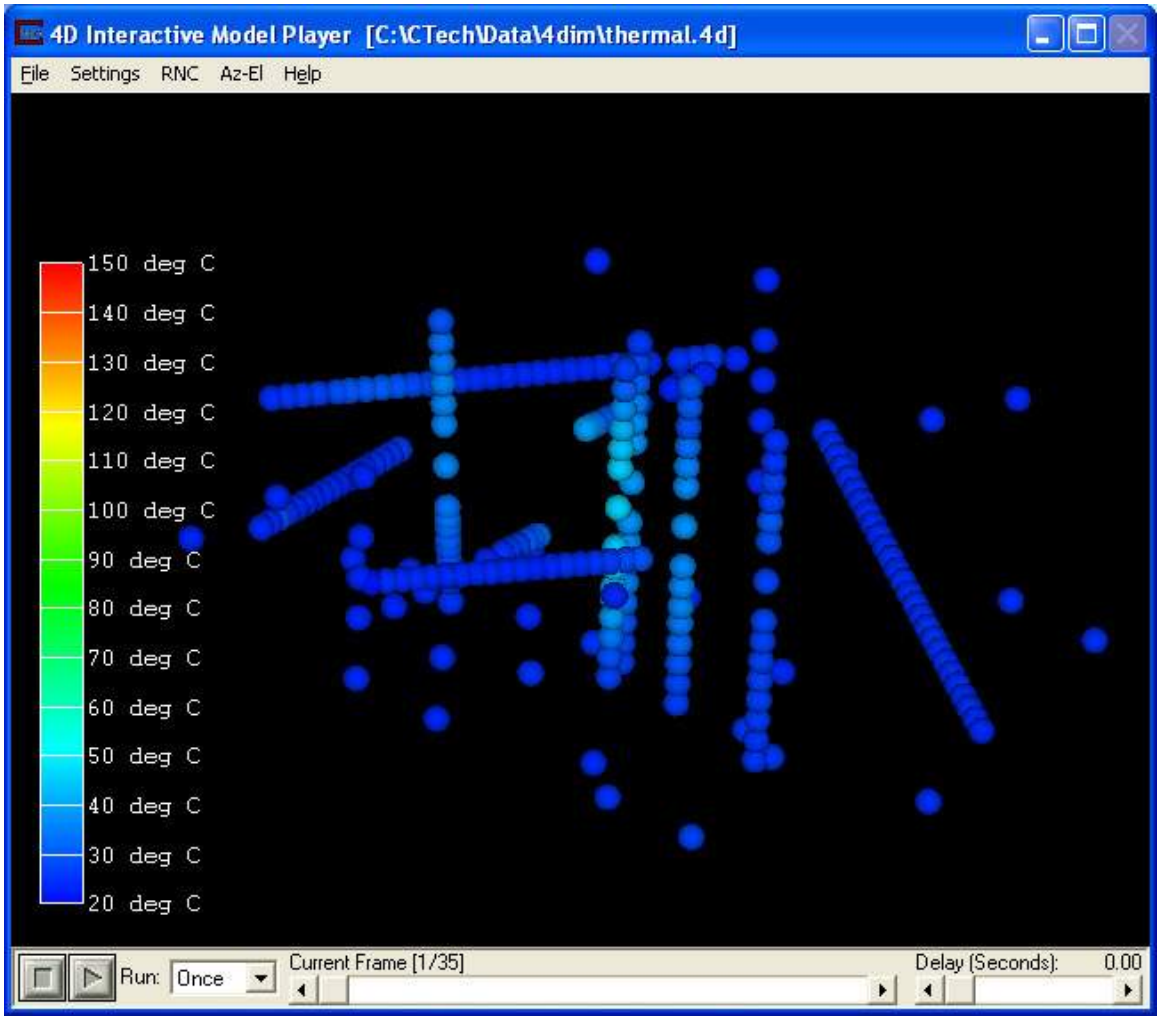
The **Update Viewer** button will set the viewer to the correct width and height if the Set Viewer Res toggle has been selected.

4D Interactive Model Player

The module for the player integrated in C Tech software is:

The Playback _4DIM module and the stand-alone 4DIM Player are identical in functionality with very minor exceptions. The players include an integrated C Tech Viewer with many of its features. Mouse interactions are nearly identical to the Viewer's. The bottom of the player window includes script controls and more.

After loading the file specified in the previous topic, the player's window should look like:



Shortcuts

There are a few keyboard shortcuts worth noting. These will give you quick control over the player.

- CTRL-F sets the player to FULL SCREEN mode. This is not equivalent to the maximize button in the upper right corner since this removes the normal borders.
- With the player as the active window, ESC(ape) exits Full Screen mode
- CTRL-H toggles the Auto-Hide mode.

The VCR type buttons on the player panel allows you to run or pause the selected scipt. The function of each button is:

 **Stop**: Stop script from playing.

 **Pause**: Pause script.

 **Play**: The play button will open the Choose 4DIM View window. This window selects the current script to be played by the 4DIM player. There are standard scripts that can be played with every 4DIM or complex scripts that can be created and bundled for specific 4DIM's. If a 4DIM has been paused while playing a script this button will instead resume the playing of the script.

Start by hitting the Play button.Select "Play Each Frame" from the Standard Script Library and then select the "Play Script" button.Notice that the animation plays from the beginning to the end only one time. There are options to change this behavior.

The **Run** option menu allows for specifying three different ways of playing the animation:

Once displays the script one scene at a time.

Cycle will display the script one scene at a time, when the script is complete it will repeat this process.

Bounce also displays the script one scene at a time, however, when the script is complete it will then play the scenes in the reverse order. It will continue to bounce until stopped.

Change to the **Cycle** option and Play again. Note now that the animation runs until you Pause or Stop. In this mode it runs from start to finish and then resets to the beginning and runs again.

Change to the **Bounce** option and Play again. In this mode it plays from start to finish and then from finish to start and so on.

The **Current Frame** slider provides a visual display of the current frame, and more importantly provides a direct method of examining each frame. The slider button (moving square) can be dragged forward or backward to advance or retreat the current frame in real time.

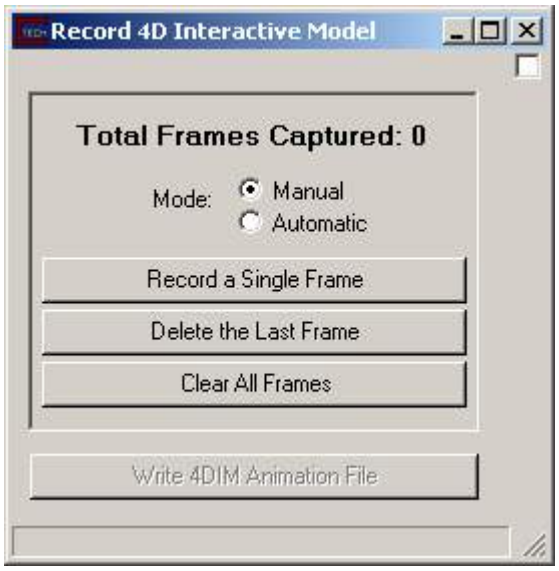
Now let's control the animation using the slider. You should have noticed that it moved when you play with any of the modes above. Press Stop (or Pause) and "click and hold" on the square moving button on the slider. As you move the slider, the animation responds by displaying the frame that you specify by your motion.

You can use the **Delay** slider to slow down the replay. The slider controls the amount of time between frames in seconds. You can set the amount of time from one-hundredth of a second to two seconds.

Now that you know how to play the animation, the real fun begins.

Recording (Capturing) 4DIM Files

The Record_4DIM module connects to the Viewer in EVS-PRO and MVS in the same way that the Output_Images module does. The user interface for Record_4DIM is shown below.

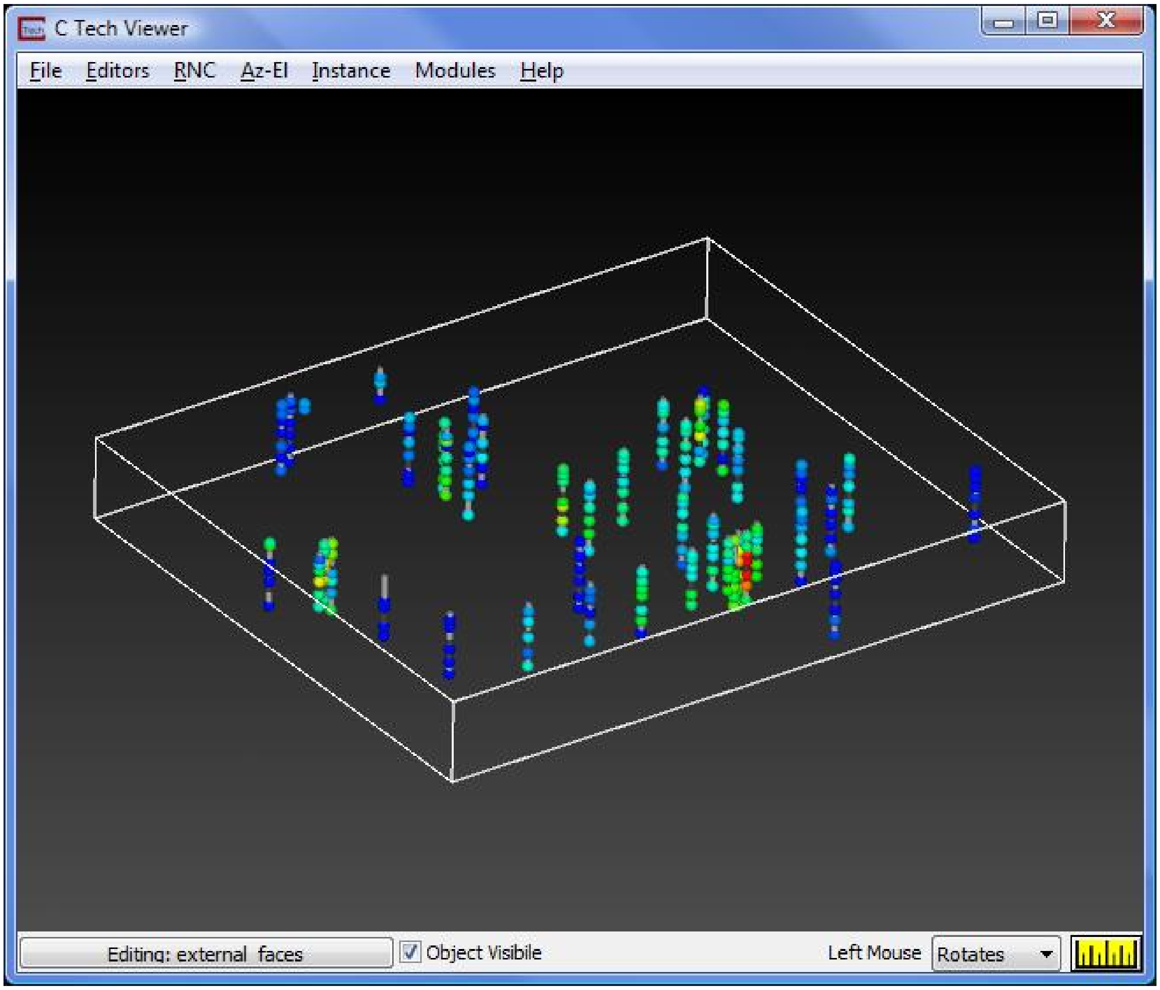


Limitations

- In some circumstances Transform_Group cannot be used with 4DIMs. It can cause the 4DIM extents to be different than they were in the EVS/MVS Viewer. This has been noted when doing rotations.
  - In most cases, the transform_field module can be used instead, however it does not allow for multiple objects to be connected to its input.
- volume_renderer is not compatible with 4DIMs
- 4DIM files will not record any object whose cache has been disabled. This occurs when large fields are connected to the Viewer. When this occurs (for external_faces in this example), the following message appears in the Status Window:

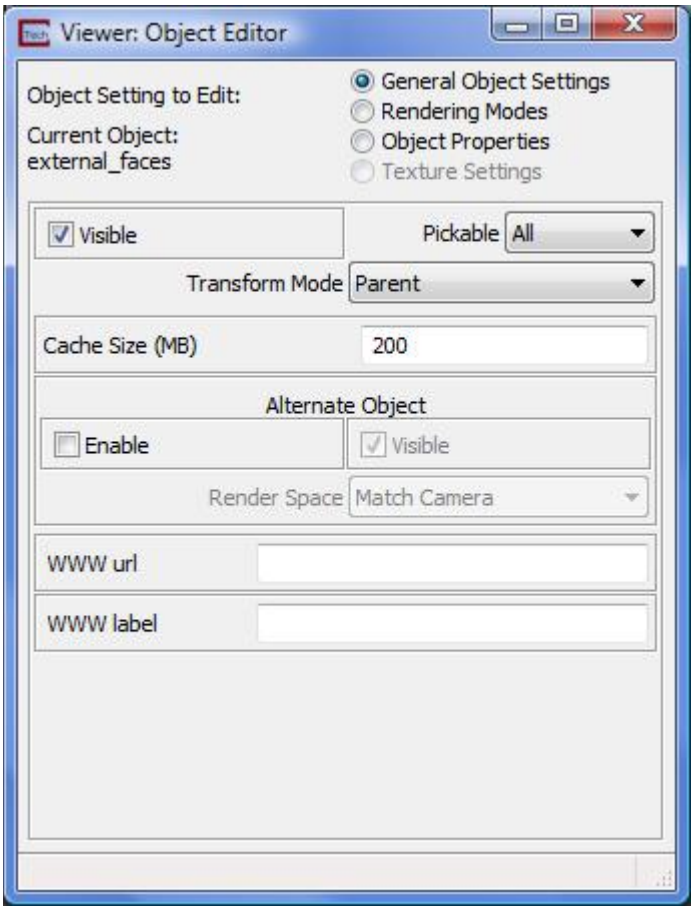
```
--- Warning from: module: external_faces ---
Field is too big (140 MB) to be put into GDobject's cache (128
MB). Drawing the
bounds only. Consider increasing the cache size or reducing the
field's
complexity.
```

You will also know this has happened when you see an object in your viewer that is only the white bounds of what SHOULD be displayed. Such as:



When this occurs, the procedure to fix it is:

1. Select the object using the "object selector" button (All Objects) in the lower left corner of the Viewer.
2. Under Object_Editor on the Viewer, under the "General" category, increase the cache size from the default value of 128 (Mb) to a larger value. In the example figure below we increased it to 200 Mb.



Operation

When in Manual mode, frames (3D Models) are saved only when the "Record a Single Frame" button is depressed. When in Automatic mode, every time the *model is changed* a frame is appended the 4DIM animation. The definition of *model is changed* is not the same as the automatic mode in Output_Images. For this module, a change is defined as a change to one or more of the 3D objects in the Viewer. Merely manipulating the view with Az-El or your mouse does not constitute a change. The reason for this is that recording frames that represent Viewer manipulations is a waste. 4DIM files can be manipulated exactly the same way you manipulate the Viewer. With 4DIM files we only want to save frames that represent changes to the **content** in the Viewer.

Before the 4DIM file is written, you have the option of deleting the last frame (this can be done repeatedly) or clearing all frames. When creating small 4DIMs manually, this can be useful.

4DIM animation frames are stored in memory (not on disk) **until** you hit the "Write 4DIM Animation File" button. Because each frame is a complete 3D representation of your model (vs. an image), 4DIM files can become quite large. Increasing your physical memory (RAM) and/or virtual memory will allow you to create larger files. Be sure that you experiment with your

system and memory before writing very large (over 200 frames or large models with fewer) files.

What is saved?

Record_4DIM records all objects (geometries) in the view: However, some geometries may not display properly when the animation is played back. In particular, volume rendering is not supported.

Recording is always done with "top" object selected. Data is captured as the viewer content (object) changes. Geometry that does not change from frame-to-frame is not re-saved. Instead, a reference is made to the previous frame so that data does not need to be duplicated. Invisible objects (visible set to zero) are not captured.

View attributes will not be saved as part of the animation.

Attributes that can be saved

1. Visibility
2. Transparency
3. Most object modes (rendering modes and line modes)
4. Background color and background type
    1. If Locked 2 or 4 color backgrounds are used, they cannot be changed by the user in the 4DIM player

View, Light and Camera Attributes

The following lists the view attributes you can change.

You can change all view attributes.

All light attributes can be changed.

The following camera attributes can be changed:

lens

clipping plane

depth cueing

create_4DV

General Module Function

This module is available only in EVS PRO and MVS.

The create_4DV module creates an animation script (*.4DV) for a 4DIM file.Once this script has been created itcan be used in one ofthree ways:

- It can bebundled into the 4DIM using the bundle_4DV module (preferred method due to simplicity for end user)
- The 4DV file script can be added to the 4DIM (as an available script)
- Once the 4DIM (.4d) file is opened, if you double click (in Windows explorer) on the 4DV file is will automatically play.

Module Input Ports

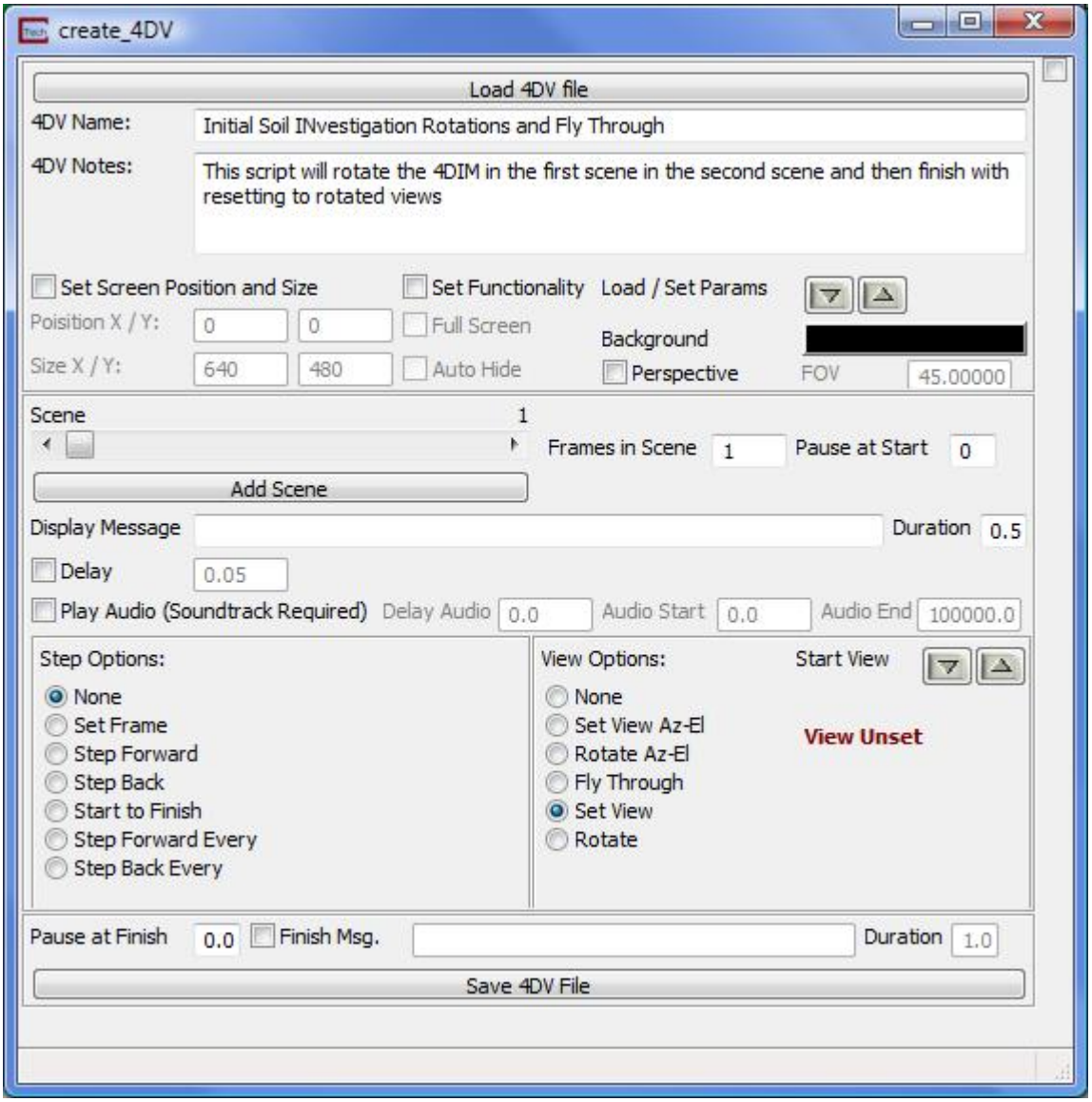
1) lines_field (Blue-Black) : This port takes a field consisting of lines. This line field can come from the read_lines, draw_2D_lines, or draw_3D_lines.

Module Output Ports

1) path_field (Blue-Black) : This port outputs a field consisting of any Fly Through's the script will go through. This allows the user to see the exact path the Fly Through will take.

2) out_obj (Red) : This port outputs a renderable object consisting of any Fly Through's the script will go through.

Module Control Panel



The global options for the create_4DV module can be seen in the image above.

The *4DV_Name* is the name for the script you are creating. When selecting a script to run in 4DIM player or with the Playback_4DIM module, this name will be displayed.

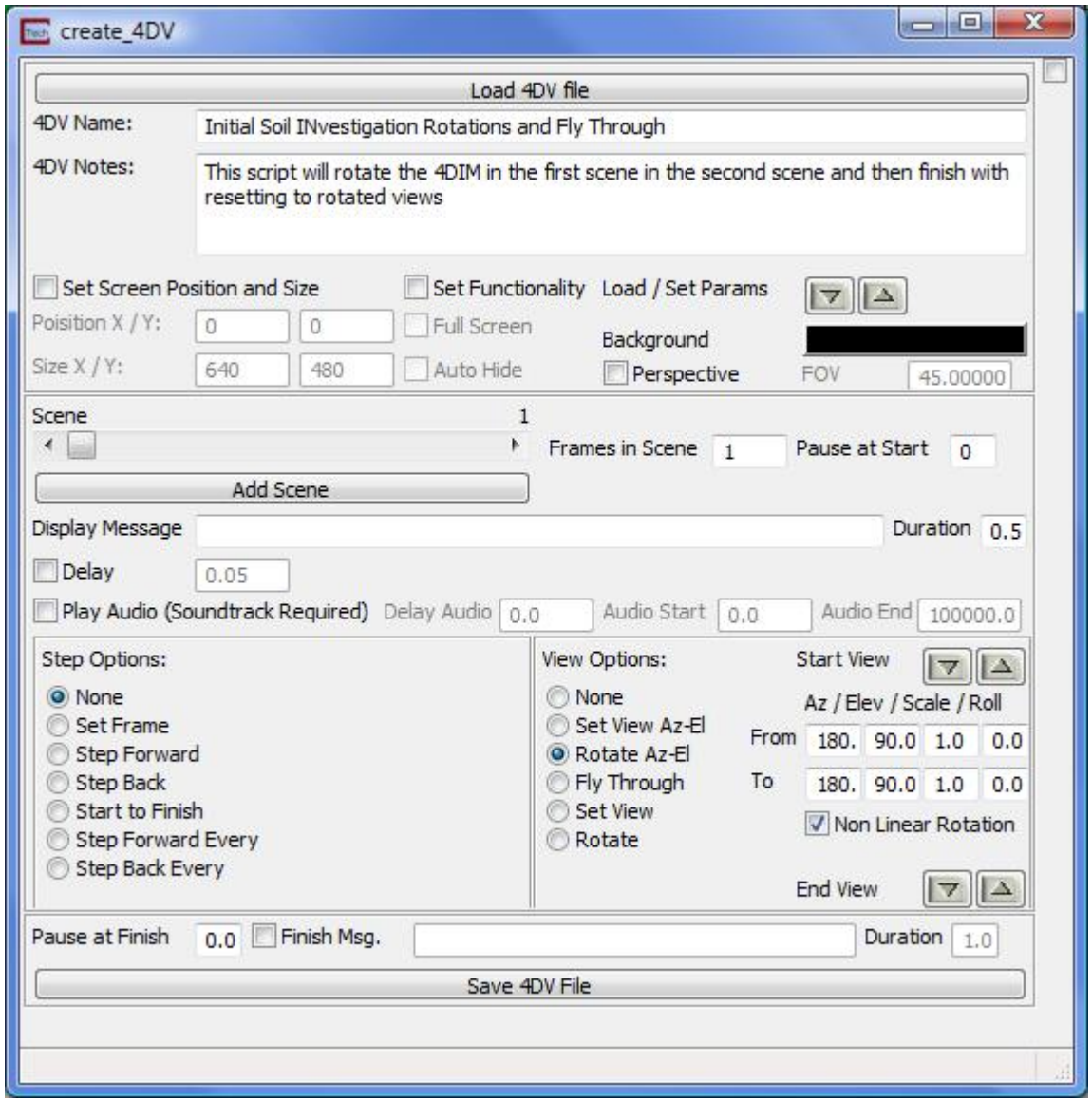
The *4DV_Notes* is used to describe the type of animation being created. When selecting a script to run these notes will also be displayed and can help distinguish one script from another.

The *Set Screen Position and Size* toggle will allow the user to set the size of the 4DIM and the position of the player.

The *Set Functionality* toggle will allow the user to set both the *Full Screen* toggle and the *Auto Hide* toggle. The Auto Hide toggle causes the play and file controls to be hidden when the mouse is not in the vicinity of the tool bars.

The *Perspective* toggle will turn on perspective viewing. This toggle will also cause the *FOV(field of view)* field to become active.

The *Load / Set Params* buttons will either load the position, size, and functionality settings from the 4DIM, or set the 4DIM settings from the create_4DV module.



The current scene options can be scene in the image above.

The ***Scene*** slider controls which scene is currently being edited.

The **Add Scene** button will add another scene to the script.

The ***Frames in Scene*** field sets the number of frames in the scene. A high number of frames will make any animations involving an interpolation (i.e. a rotation from one view to another) appear smoother, but will take more time. A small number of frames will make interpolated animations faster, but they will appear to jerk more between frames.

The ***Pause at Start*** field tells the 4DIM to pause before beginning the scene animation.

The ***Display Message*** text allows a message to be displayed in the 4DIM. This message will be displayed at the beginning of the scene and will last for the specified duration.

The ***Delay*** field gives the duration of the delay in seconds between each frame of the scene.

The ***Play Audio*** toggle will play the soundtrack bundled into the 4DIM from the specified start position to the specified end position. The *Delay* is in seconds from the start of the scene. The soundtrack is added to the 4DIM using the bundle_4DV moduel.

The **Step Options** are listed below:

1) *None*: No special stepping.

2) *Set Frame*: This will set the current frame in the 4DIM. This value is not related to the number of frames in the current scene.

3) *Step Forward*: This option will cause the frame in the 4DIM to advance once for every set number of frames in the Step Every field. So if the Step Every value is set to 5, for every time our current scene advances 5 frames the 4DIM frame will advance 1.

4) *Step Back*: This option is the same as above except that for every set number of frames in the Step Every field the 4DIM frame will go back 1 frame.

5) *Start to Finish*: This option will cause the 4DIM to advance from the Start Frame to the End Frame. The End Frame may be smaller than the Start Frame which will cause the 4DIM player to run in reverse.

6) *Step Forward Every*: Will cause the 4DIM player to step forward through every frame.

7) *Step Back Every*: Well cause the 4DIM player to step in reverse through every frame.

The two button groups *Start Values* and *End Values* will load the current frame from the 4DIM player into the appropriate box, or load the frame number from the current scene into the 4DIM.

The **View Options** are listed below:

1. None: No view changes.
2. *Set View Az-El*: Allows the user to type in an *Azimuth, Elevation, Scale* and *Roll*. Views set in this manner are always OBJECT CENTRIC. To have more complex views or rotations, use options 5 or 6. These values can be loaded in from the 4DIM using the down arrow of the *Start View* button group. The 4DIM view can be set using the up arrow.
3. *Rotate Az-El*: Allows the user to rotate the view from a starting view to an end view. The start view can loaded in from the Viewer by setting Start and End positions in Az-El and using the down arrow of the Start View button group. Rotations set in this manner are always OBJECT CENTRIC. The end view can be loaded using the down arrow from the End View group. The current 4DIM view can be set using the up arrow of either group. A *Non-Linear* rotate will cause the view to rotate more quickly at the beginning and end, and more slowly in the middle.
4. *Fly Through*: This option will create a scene where the view follows a specified path. This path can be created by using either draw_2D_lines,

draw_3D_lines or by creating and reading and EVS Line File (*.elf). Which line is followed can be specified using the *Line #* field. The *Spline Path* toggle will cause the fly through to be smoothed while still traveling through all of the points in the line. The *Pitch, Scale,* and *Normalized Back Off* can also be set for this fly through. The Pitch determines from which angle above or below the line you are looking at the next point. A Pitch greater than zero mean you are looking up from the current position along the line to the next position, equal to zero means you are looking directly down the line, and less than zero means you are looking down to the next position. The Scale term is used in the same way as the Az-El panel in the viewer. The Normalized Back Off value is discussed in the help for the fly_through module. The smoothness of the Fly Through is determined by the number of frames in the scene. For more information on these parameters, and for a more in depth discussion on fly throughs, please look at the help for the Control_Fly_Through module.

5. *Set View*: Allows the user to set arbitrary rotations, scale and translations using the mouse. The 4DIM view can be set using the up arrow.
6. *Rotate*: Allows the user to set arbitrary rotations, scale and translations using the mouse from a starting view to an end view. The start view can loaded in from the 4DIM using the down arrow of the Start View button group. The end view can be loaded using the down arrow from the End View group. The current 4DIM view can be set using the up arrow of either group. A *Non-Linear* rotate will cause the view to rotate more quickly at the beginning and end, and more slowly in the middle.

The script finish options are displayed in the image above.

The *Pause at Finish* field will cause the 4DIM player to pause for the specified time in seconds after the last frame of the last scene.

The *Finish Msg.* toggle will display the set message text for the *Duration* in Seconds.

The *Save 4DV File* will set the 4DV file name and create the 4DV file. This should be done after the entire script has been completed.

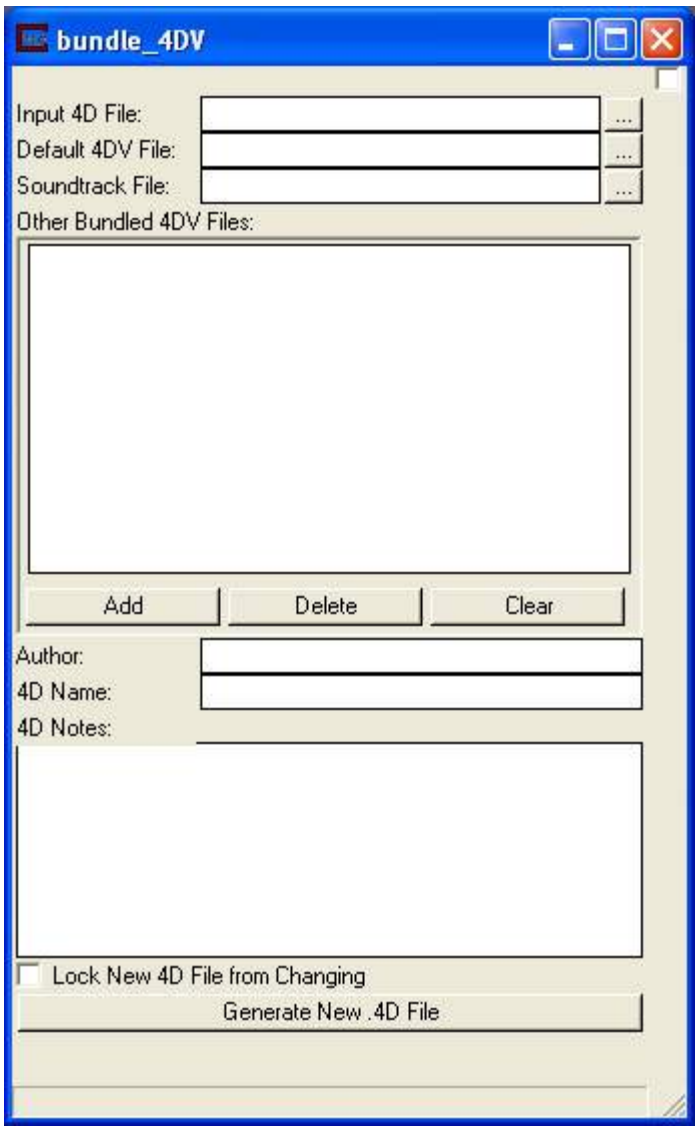
bundle_4DV

General Module Function

This module is available only in EVS PRO and MVS.

The bundle_4DV modulewill take a 4DIM file,one or more4DV scripted animation files, and a Soundtrack File(e.g. .mp3 file)andintegrated them into anew animated 4DIM file.

Module Control Panel



The bundle_4DV control panel can be seen in the image above.

The *Input 4D File* browser will allow the user to select a precreated 4DIM file.

The *Default 4DV File* browser will allow the user to select a precreated 4DV file. The Default 4DV file will be the script loaded by the 4DIM player when it loads the 4D file. 4DV files can be created with the create_4DV module.

The *Soundtrack File* browser will allow the user to select a sound file with the following format: .m4a; .wma; .asf; mp3. This soundtrack file can be played along with the 4DIM by creating a 4DV script file.

The *Other Bundled 4DV Files* list allows the user to add additional 4DV script files to the newly created 4D file.

The *Author* field is an optional field for documenting the creator of the 4D file.

The *4D Name* field allows the user to set the name of the 4D. This is not the filename.

The *4D Notes* field allows the user to describe the contents of the 4D.

The *Lock New 4D File from Changing* will keep the newly created 4D from having working with the bundle_4DV module in the future. This insures that distributed 4D files will not be added to.

The *Generate New .4D File* button selects the name of the 4D file and creates the 4D file.

loop

General Module Function

The loop module iterates an operation. For example, you could use a loop object to control the movement of an object in your application; such as incrementing the movement of a slider for a slice plane.

Module Input Ports

This module has five input ports which accept other loop counter information. From left to right they are:

Start

End

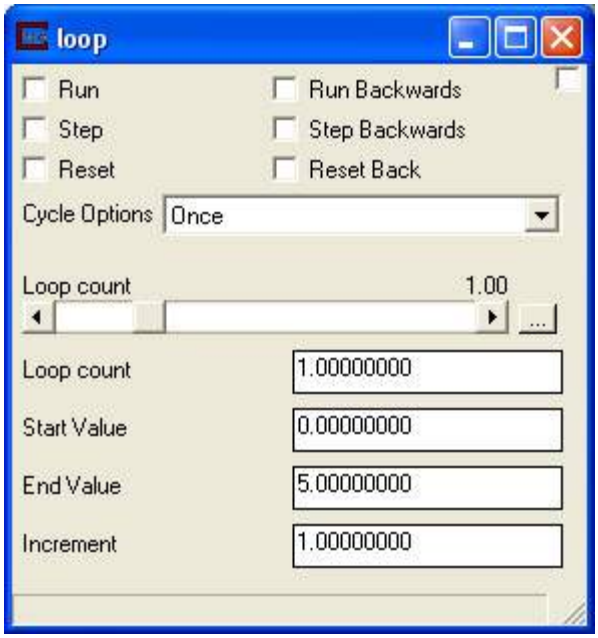
Increment

count

done

Module Output Ports

The output ports are identical to the input ports.



Module Control Panel

The user interface for loop is shown above.

The ***Run*** toggle when set will kick off an iteration of the loop. It is reset to off when it is finished.

The ***Step*** toggle will make one increment of the loop as specified int the Increment type-in.

Reset, when toggled, resets the count subobject to the value of the start_val subobject. This value is reset to off right after it is changed.

The ***Run Backwards***, ***Step Backwards*** and ***Reset Back*** are self explanatory.

The ***Cycle Options*** allow for Once (one iteration), Cycle (keep running from beginning), and Bounce (run to end, then backwards, then forwards, etc.)

The **Loop count** is updated during running of the module, but the user may jump to any part of the specified iteration by typing in a valid value. It is available as a slider and a type-in. The slider is particularly useful as you can control the output of loop by moving it in real-time.

Start Value indicates the starting value of the iteration and refers to the slider value of the module to which loop is outputting.

End Value indicates the ending value of the iteration and refers to the slider value of the module to which loop is outputting.

Increment indicates the interval at which to count from the starting value to the ending value. For example, to move a slice plane in 10 steps from -50.0 (Start Value) to 40.0 (End Value) the user would choose an increment of 10.00. These increments may have non-integer values.

fly_through

General Module Function

fly_through is an animation modules which facilitates controlling the Viewer or creating an animation in which the view follows a complex 3D path on, through, or around your model. The method by which this module controls fly-throughs allows the user to pause at any time and interact with the model using their mouse or the Az-El panel.

Az-El parameters (azimuth, elevation, scale, field of view, rotation/scaling center, etc.) are updated by fly_through in real time. This can be seen by running fly_through with the Az-El window open. However, please note that this will slow your animation substantially because of the need to continously update the parameters in Az-El.

Unlike this module's predecessors, there is no need to "reset" to a normal mode, as fly_through's normal mode of operation is identical to using Az-El with advanced centering options.

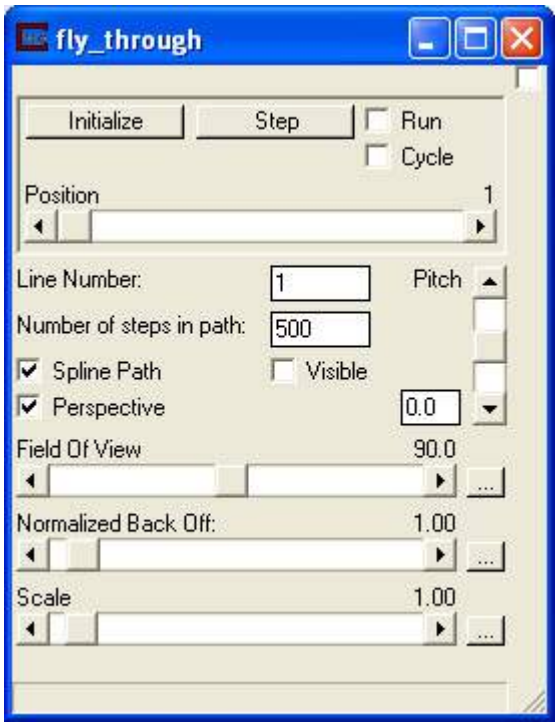
IMPORTANT NOTE: Be sure to **TURN OFF** "Animate Viewer" in the Animator module if you're controlling fly_through with the Animator.

Module Input Ports

fly_through has two input ports. The magenta port must be connected to the output port on the Viewer and the blue/black (field) accepts the output from any module with that can provide a polyline path. Examples of appropriate modules would be read_lines, draw_3D_lines, etc.

ModuleOutput Ports

fly_through has two output ports. The blue/black (field) port outputs the subset and/or splined version of the input path. The red port connects to the Viewer to allow display of the subset and/or splined path.



Module Control Panel

The control panel for fly_through is shown in the figure above.

The ***Initialize*** button sets the viewer orientation to the beginning of the fly through. This should be pressed after all modules connections are made and a valid (flying) path input is supplied to the module. It can also be used to reset the position to the start of the specified Line Number.

The ***Step*** toggle advances one step along the fly-through path.

The ***Run*** toggle allows you to start and stop the fly-through animation. It is reset to off when it is finished.

The ***Cycle*** toggle allows your animation to rerun when it reaches the end. (keep running from beginning), If your path closes on itself this can make a continuous animation.

The ***Position*** slider determines the camera position along the length of the specified path. The value is updated during running of the module, but the user may jump to any point along the path part of the specified iteration by adjusting the position slider. This parameter may also be controlled using the Animator. Note: when controlling position with the Animator, only integer values apply. If you want an Animator driven animation with more steps than the specified number (default 500), you must adjust the *Number of steps in path* parameter in this module.

The ***Line Number*** type-in lets you specify which polyline segment to use. This is very useful since read_lines and other modules can have inputs which contain multiple paths.

The ***Number of steps in path*****:** type-in lets you specify the total number of steps into which the polyline is divided. More steps result in a slower and smoother fly-through.

Spline Path is a toggle which controls whether the incoming path is splined before being subset into the specified number of steps. Unless you are smoothing or splining with polyline_spline this should normally be ON.

Visible is a toggle which controls the visibililty of the (splined) flight path if the red output port of fly_through is connected to the Viewer.

Perspective is a toggle turns perspective on or off. **Fly-throughs generally must use perspective** because without it there is no concept of distance from the camera to the objects in the Viewer.

Field of View is a slider that sets the total included angle (in degrees) of the camera lens used to see the scene. and perspective is based on the concept of a lens angle. Larger angles have a wider field of view. This parameter may also be controlled using the Animator.

Normalized Back off is a slider that sets a back-off distance between the point on your path and the camera. The units are dimensionless (not in your model units of feet or meters). This slider ranges from 0 to 20.0. In perspective view, back-off is similar to the inverse of scale. This parameter is only active when Perspective is selected and it has a noticable effect on the clipping planes. This parameter may also be controlled using the Animator.

Scale is a slider that is identical to the Scale in the *Azimuth & Elevation* panel. This slider ranges from 0.01 to 25.0. If the buttons on the Scale and Normalized Back Off sliders are in the same relative position, the apparent scale of the model is relatively unchanged, however the clipping and point of view is altered. This parameter may also be controlled using the Animator.

Sample Applications

A sample network that demonstrates the use of fly_through is fly_thru_full-site.v in the Pro folder. The applications is shown below:

{bmc fly_through_app.bmp)

In order to help users understand the basics of fly through animation, each modules function is summarized below.

Post_samples creates the display of the measured data. Any other objects such as plumes, slices, etc. could be added (though you would need to add Krig_3D to create a 3D volumetric model).

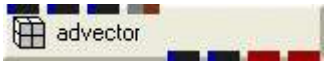
Axes provides annotation of the spatial extents.

Read_lines is used to read the 3D path which was previously drawn and saved with draw_3D_lines.

fly_through takes the output from *read_lines* and controls the *Viewer's* apparent camera position.

The Viewer window is shown below half way through the fly through.

advector



General Module Function

The advector module combines streamlines capability and a tool for sequential positioning of glyphs along the streamlines trajectory to simulate advection of weightless particles through a vector field (for example, a fluid flow simulation such as modflow). The result is an animation of particle motion, with the particles represented as any EVS geometry (such as a jet or a sphere). The glyphs can scale, deflect or deform according to the velocity vector it passes. At least one of the nodal data components input to advector must be a vector. The direction of travel of streamlines can be specified to be forwards (toward high vector magnitudes) or backwards (toward low vector magnitudes) with respect to the vector field. The input glyphs travel along streamlines (not necessarily visible in the viewer) which are produced by integrating a velocity field using the Runge-Kutte method of specified order with adaptive time steps.

Module Input Ports

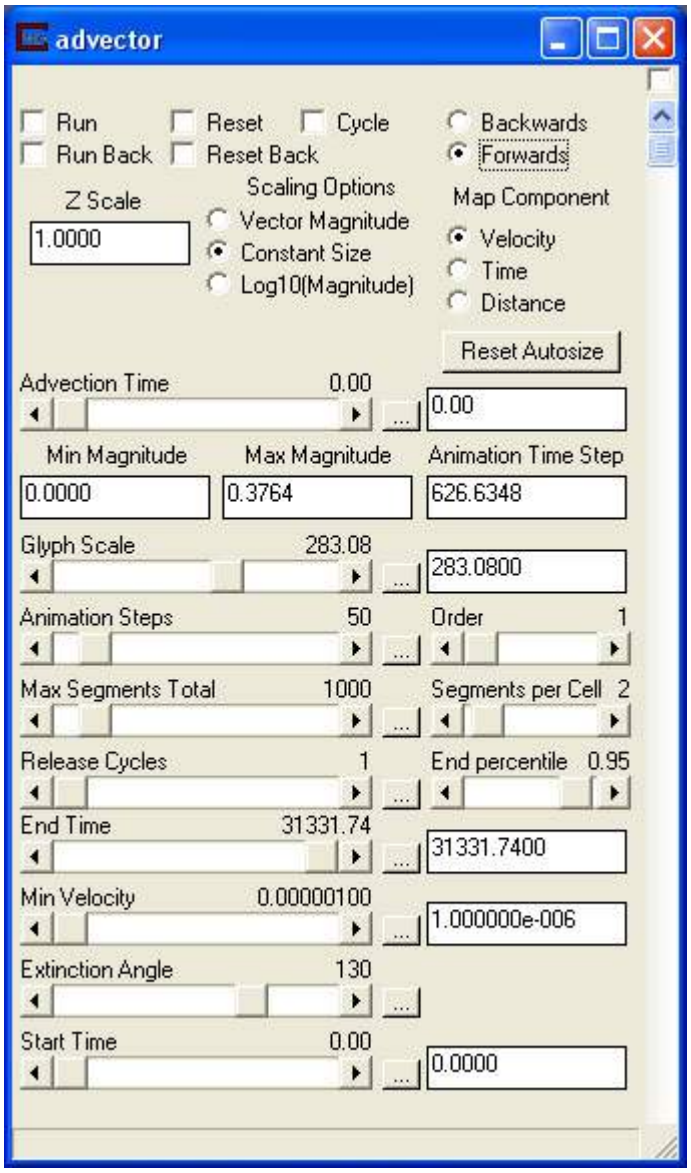
advector has three input ports.

- Data passed to the first port (the left port) can have a mesh of any type, and must have at least one vector nodal data component. To generate streamlines of scalar data, a module such as gradient must be used to calculate the gradient of a scalar field.
- The second port accepts data specifying the starting location of the streamlines. This data is typically generated by the create_grid, slice or isolines modules.
- The third port accepts any geometry mesh (glyph, usually read in with the load_glyph module) which will be rendered at each starting point and moved sequentially through the vector field during running of the advector module.

Module Output Ports

advector contains four output ports.

- The leftmost port outputs a mesh composed of the original mesh plus the meshes representing the particles.
- The second output port creates a new unstructured polyline mesh representing the streamlines, and also contains a nodal data component referencing the velocity component.
- The third output creates a renderable version of the glyphs.
- The fourth output port sends streamline polylines to the viewer.



Module Control Panel

The control panel for advector is shown in the figure above. Many of the parameters are related to streamline interpolation which is discussed in the streamlines help. Below is a discussion of those parameters unique to advector.

The ***Run*** toggle starts or stops advection.

The ***Cycle*** toggle starts the advection again at Start Time when End Time is reached.

The ***Reset*** Time toggle resets Time to the value of Start Time.

The ***Direction*** radio buttons allow the user to specify forward or backward streamlines. Forward streamlines start from the specified starting points and travel to the maximum velocity location. Backward streamlines travel from the specified starting points to the minimum or zero velocity location.

The ***Scaling Options*** radio buttons establishes how glyphs are sized based on the data values.

- ***Vector Magnitude*** sizes the glyphs proportional to the velocity at each node.
- ***Constant Size*** makes all glyphs the same size.
- ***Log10(Magnitude)*** scales by the log of the velocity. When velocities span several orders of magnitude working in log space is useful. When selected, an additional parameter is visible:
  - ***Magnitude for Zero Size*** defines the lower clamping velocity that will correspond to a ZERO (0.0) size glyph regardless of "Glyph Scale".

The ***Advection Time*** slider and type-in allow you to control the particle animation and also display the progress when Run is selected. The time runs from Start Time to End Time during running of advector.

The ***Advection Time Step*** type-in allows you to set the duration of each animation step.

The ***Glyph Scale*** slider adjusts the sizes of the glyphs. Default range is 0.0 to 100.00, but the type-in box allow for increasing the range to any value. The default scale of 1.00 is approximately 1/100th the x,y extents of the model.

The ***End percentile*** slider allows you to control the automatic scaling of glyphs based on the nth percentile value (versus the maximum 100^{th}%). This addresses datasets where there are only a few nodes with extremely high values (like wells).

The ***Map Component*** radio buttons allow the user to color by Velocity, Time or Distance.

The ***Min Magnitude*** and ***Max Magnitude*** type-ins display the min/max velocities represented in the input field. Changing these values allows the user to reset the min-max values for coloring purposes. NOTE: once you change these values they will not automatically update.

The ***Z Scale*** type-in reflects the value of the z exaggeration port.

The **Segment per Cell** slider is used to set the number of integration steps to be used in each cell (i.e., the number of divisions of the cells) to calculate the streamline. The default is one and the range is 1 to 16.

The ***Max Segments Total*** slider is used to set the maximum allowable number of streamline segments that will be completed for each streamline. If the number of segments along a streamline exceeds the max number, the streamline is terminated at the end of the last (max) segment. The default is 15 and the range is 1 to 1000.

The **Order** slider is used to set the order of the integration. Higher order integration is more accurate, but executes much slower. The default is one and the range is 1 to 4.

The **Min Velocity** slider and type-in is used to specify the minimum velocity that will be considered in the integration. If the magnitude of the velocity field in a region is less than this minimum value, streamlines will end in that region (or will not be produced if the gradient at a starting point is less than the min). Setting this to a lower value will produce longer streamlines (and typically more). Higher values tend to produce less streamlines and shorter streamlines. If streamlines are not visible on the data set, setting this to a lower value may produce streamlines. The default value is 0.000001.

The ***Extinction Angle*** slider is used to specify the maximum allowable angle between successive line segments before integration (streamline generation) should be terminated. The default value is 130 degrees.

The ***Animation Steps*** slider controls the incrementation of the glyph stops between the Start Time and the time along the original streamline continuum for each advection step. For example, if the End Time is 10000 and the Animation Steps slider is 10 then there will be ten steps (frames) when the animation is run. The Animation Time Step Type-in box will reflect the time increment based on the slider choice. For the above example (10 steps) the type-in would read 1000.

The ***Release Cycles*** slider determines the number of times the particle glyphs will be sent from their release points. This could be thought of as the number of pulses of particle releases. For example, if the End Time is 10000, then a pulse of particles is released at the Start Time and again at 5000.

The ***End Time*** slider and type-in reflects the time value along the original streamline at which to halt advection of all particles. The default is 1000.0, but the type-in box allows for increasing the range to any value. Typically with groundwater flow problems (very low velocities) this number needs to be very large to achieve ample travel distance of particles.

Note that adjustments to either the End Time, Animation Steps, or Animation Time Steps will directly affect one or both of the remaining parameters. For example, adjusting the Animation Time Steps in the above example from 1000 to 2000 would increase the End Time from 10000 to 20000.

The ***Start Time*** slider reflects the time value along the original streamline continuum at which to start advection. The default is 0.0.

Related Modules

->glyph

->streamlines

advect_surface

General Module Function

The advect_surface module combines streamline_surface capability and a tool for sequential positioning of glyphs along the streamlines trajectory to simulate advection of particles down a surface. The result is an animation of particle motion, with the particles represented as any EVS geometry (such as

a jet or a sphere). The glyphs can scale, deflect or deform according to the velocity vector. The direction of travel of streamlines can be specified to be downhill or uphill (for the slope case). The input glyphs travel along streamlines (not necessarily visible in the viewer) which are produced by integrating a velocity field using the Runge-Kutte method of specified order with adaptive time steps.

The advect_surface module is used to produce streamlines and particle animations on any surface based on its slopes. The direction of travel of streamlines can be specified to be downhill or uphill for the slope case. A physics simulation option is also available which employs a full physics simulation including friction and gravity terms to compute streamlines on the surface.

Module Input Ports

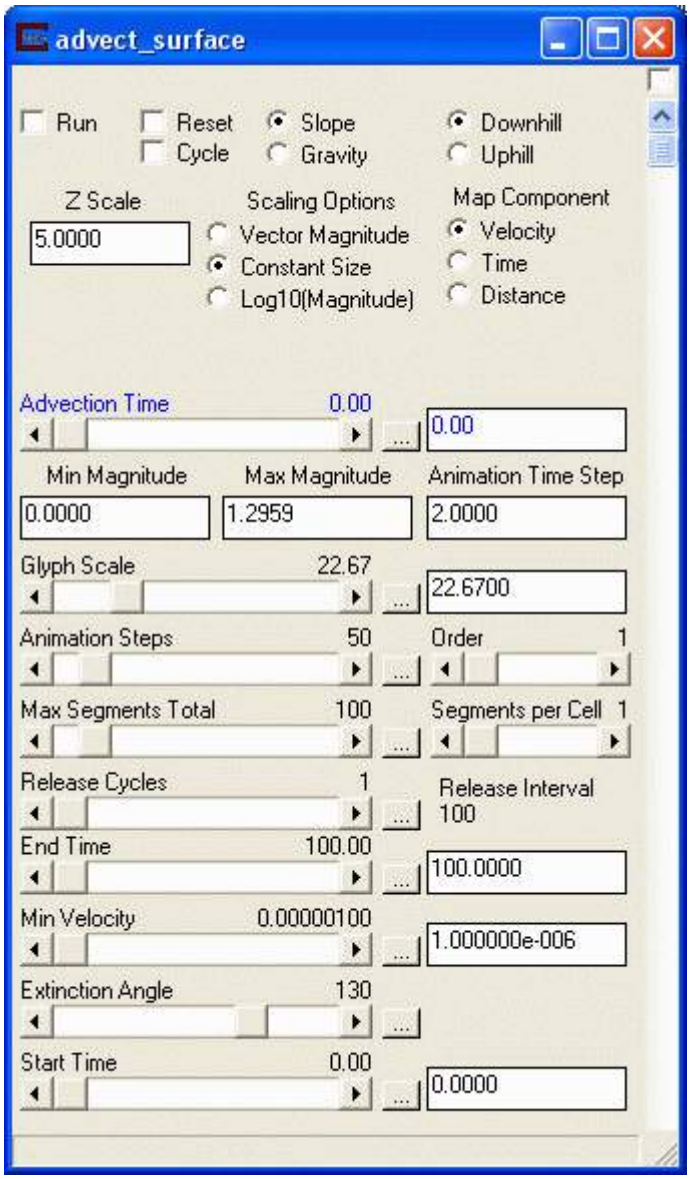
Advect surface has four input ports.

- Data passed to the first port (the left port) must be a surface.
- The second port accepts data specifying the starting location of the streamlines. This data is typically generated by the slice or isolines or place_glyph modules.
- The third port accepts the glyph, usually read in with the load_glyph module, to be used as the advector particles.
- The last port is the z exaggeration factor. This input causes the stream path generation to be calculated in an unscaled system ensuring that velocities are not scaled and are accurate, then scales the position of the streamlines.

Module Output Ports

advect_surface has two output ports.

- The leftmost output port creates a new unstructured polyline mesh representing the streamlines. The output also contains a nodal data component referencing the velocity component.
- The right output port can send either streamline polylines or renderable streamribbons to the viewer.



Module Control Panel

The control panel for advect_surface is shown in the figure above. Many of the parameters are related to streamline interpolation which is discussed in the streamline_surface help. Below is a discussion of those parameters unique to advect_surface.

The *Map Component* radio button allows you to choose which computed data component to use for coloring the lines. Choices are Velocity, Time, Distance and Uncolored.

The *Direction* radio buttons allow the user to specify forward or backward streamlines. Forward streamlines start from the specified starting points and travel to the maximum velocity location. Backward streamlines travel from the specified starting points to the minimum or zero velocity location.

The *Coloring Options* radio buttons allow the user to specify Magnitude (normal) or Log10(Magnitude) representation of the velocity data for coloring purposes. When velocities span several orders of magnitude this is useful.

The *Physics* radio buttons allow the user to specify whether streamlines will be computed based on the slopes of the surface only or whether a full physics simulation including friction and gravity terms will be used to compute streamlines on the surface. When *Gravity* is selected *Segments perCell* and *Order* do not apply but additional parameters appear for the module. These are:

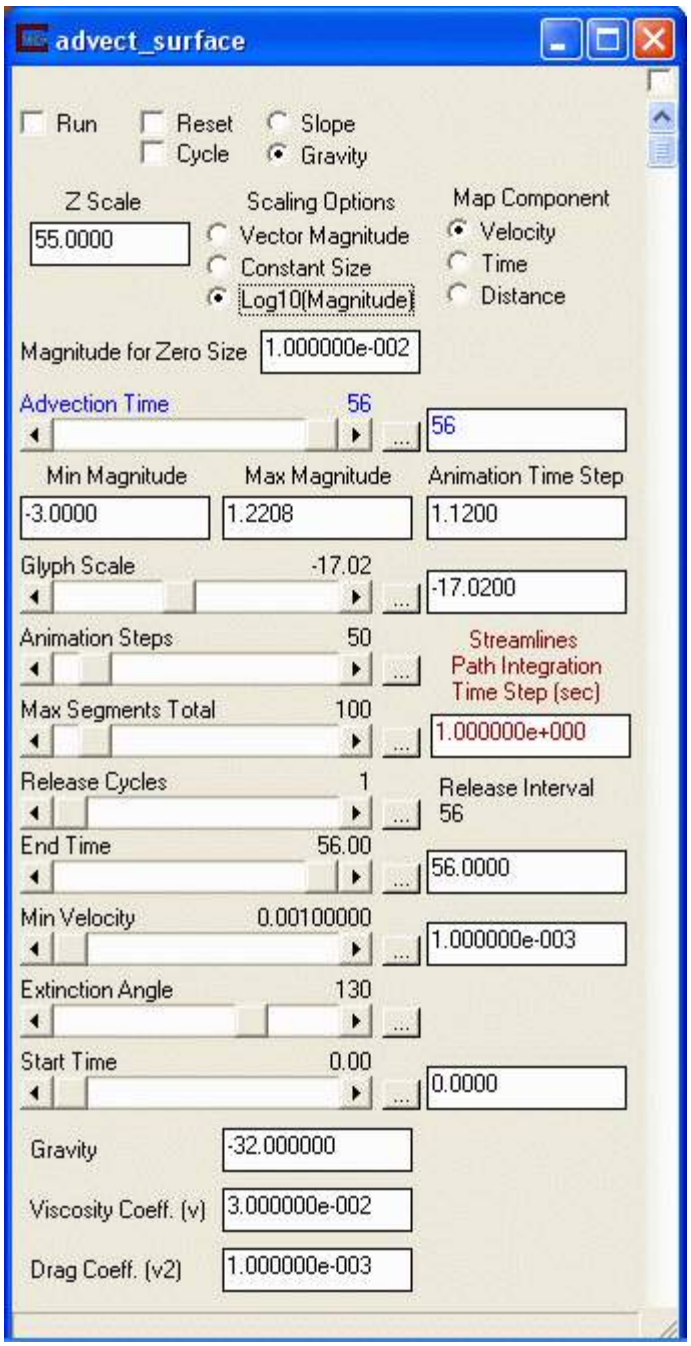
Integration Time Step is the time step for the numerical integration of the paths. For typical gravity units (like 32 feet per second-squared) this value is in seconds.

Gravity is the coefficient of gravity for your units. If your coordinate units are feet, the appropriate (default) value would be 32 feet per second-squared.

Viscosity Coefficient (v) is the friction term that depends on velocity.

Drag Coefficient (v2) is the friction term that depends on velocity-squared.

The panel when Gravity is selected is show below.



The **Min Magnitudeand *Max Magnitude*** type-ins display the min/max velocities represented in the input field. Changing these values allows the user to reset the min-max values for coloring purposes. NOTE: once you change these values they will not automatically update.

The ***Z Scale*** type-in reflects the value of the z exaggeration port.

The **Segment per Cell** slider is used to set the number of integration steps to be used in each cell (i.e., the number of divisions of the cells) to calculate the streamline. The default is one and the range is 1 to 16.

The **Max Segments** *Total* slider is used to set the maximum allowable number of streamline segments that will be completed for each streamline. If

the number of segments along a streamline exceeds the max number, the streamline is terminated at the end of the last (max) segment. The default is 15 and the range is 1 to 1000.

The ***Release Cycles*** slider determines the number of times the particle glyphs will be sent from their release points. This could be thought of as the number of pulses of particle releases. For example, if the End Time is 10000, then a pulse of particles is released at the Start Time and again at 5000.

The **Order** slider is used to set the order of the integration. Higher order integration is more accurate, but executes much slower. The default is one and the range is 1 to 4.

The **Min Velocity** slider and type-in is used to specify the minimum velocity that will be considered in the integration. If the magnitude of the velocity field in a region is less than this minimum value, streamlines will end in that region (or will not be produced if the gradient at a starting point is less than the min). Setting this to a lower value will produce longer streamlines (and typically more). Higher values tend to produce less streamlines and shorter streamlines. If streamlines are not visible on the data set, setting this to a lower value may produce streamlines. The default value is 0.000001.

The ***Extinction Angle*** slider is used to specify the maximum allowable angle between successive line segments before integration (streamline generation) should be terminated. The default value is 130 degrees.

modpath_advector

General Module Function

The modpath_advector module combines MODPATH capability and a tool for sequential positioning of glyphs along the MODPATH lines trajectory to simulate advection of weightless particles through a vector field. The result is an animation of particle motion, with the particles represented as any EVS geometry (such as a jet or a sphere). The glyphs can scale, deflect or deform according to the velocity vector it passes. The direction of travel of streamlines can be specified to be forwards (toward high vector magnitudes) or backwards (toward low vector magnitudes) with respect to the vector field. The input glyphs travel along streamlines (not necessarily visible in the viewer) which are produced by integrating a velocity field using the Runge-Kutte method of specified order with adaptive time steps.

Module Input Ports

The modpath_advector module has four input ports.

- The first (leftmost) accepts field data usually from Load_EVS_Field or Read_TCF.
- The second port accepts data specifying the starting location of the particle paths, this data is typically generated by a slice, create_grid, isolines or place_glyph module.
- The third port accepts any geometry mesh (glyph, usually read in with the load_glyph module) which will be rendered at each starting point

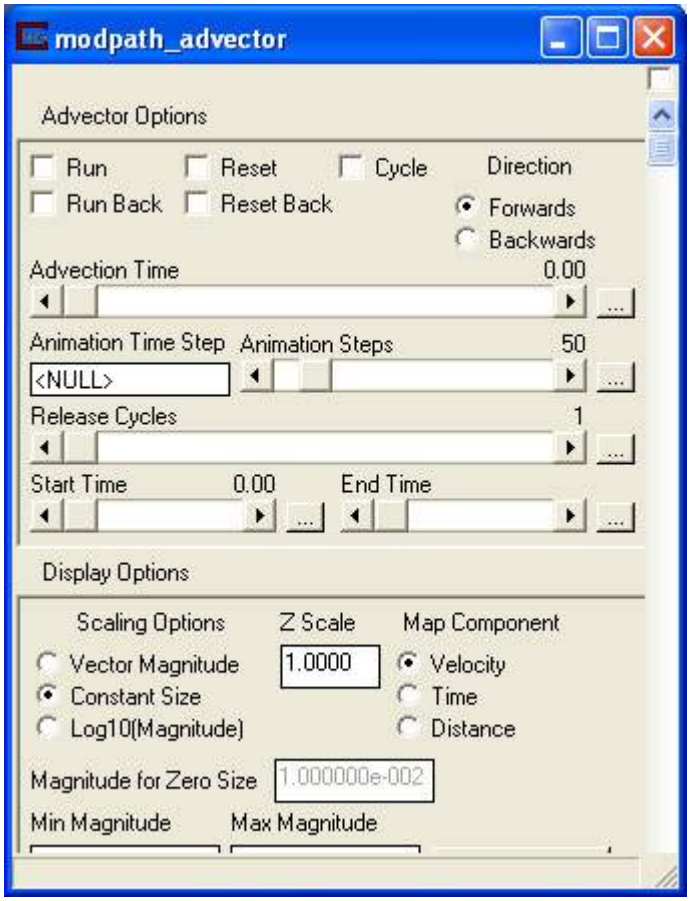
and moved sequentially through the vector field during running of the advector module.

- The last port is the z exaggeration factor. This input causes the path generation to be calculated in an unscaled system ensuring that velocities are not scaled and are accurate, then scales the position of the paths.

Module Output Ports

Modpath_advector contains four output ports.

- The leftmost port outputs a mesh composed of the original mesh plus the meshes representing the particles.
- The second output port creates a new unstructured polyline mesh representing the MODPATH lines, and also contains a nodal data component referencing the velocity component.
- The third output creates a renderable version of the glyphs.
- The fourth output port sends MODPATH polylines to the viewer.



Module Control Panel

The control panel for modpath_advector is shown in the figure above.

Advector options :

The ***Run*** toggle starts or stops advection.

The ***Cycle*** toggle starts the advection again at Start Time when End Time is reached.

The ***Reset*** Time toggle resets Time to the value of Start Time.

The ***Direction*** radio buttons control the direction of travel of the glyphs.

The ***Advection Time*** slider and type-in allow you to control the particle animation and also display the progress when Run is selected. The time runs from Start Time to End Time during running of advector.

The ***Advection Time Step*** type-in allows you to set the duration of each animation step.

The ***Animation Steps*** slider controls the incrementation of the glyph stops between the Start Time and the time along the original streamline continuum for each advection step. For example, if the End Time is 10000 and the Animation Steps slider is 10 then there will be ten steps (frames) when the animation is run. The Animation Time Step Type-in box will reflect the time increment based on the slider choice. For the above example (10 steps) the type-in would read 1000.

The ***Release Cycles*** slider determines the number of times the particle glyphs will be sent from their release points. This could be thought of as the number of pulses of particle releases. For example, if the End Time is 10000, then a pulse of particles is released at the Start Time and again at 5000.

The ***End Time*** slider reflects the time value along the original streamline at which to halt advection of all particles. The default is 1000.0, but the range may be increased to any value. Typically with groundwater flow problems (very low velocities) this number needs to be very large to achieve ample travel distance of particles.

Note that adjustments to either the End Time, Animation Steps, or Animation Time Steps will directly affect one or both of the remaining parameters. For example, adjusting the Animation Time Steps in the above example from 1000 to 2000 would increase the End Time from 10000 to 20000.

The ***Start Time*** slider reflects the time value along the original streamline continuum at which to start advection. The default is 0.0

Display Options:

The ***Scaling Options*** radio buttons establishes how glyphs are sized based on the data values.

Vector Magnitude sizes the glyphs proportional to the velocity at each node.

Constant Size makes all glyphs the same size.

Log10(Magnitude) scales by the log of the velocity. When velocities span several orders of magnitude working in log space is useful. When selected, an additional parameter is visible:

Magnitude for Zero Size defines the lower clamping velocity that will correspond to a ZERO (0.0) size glyph regardless of "Glyph Scale".

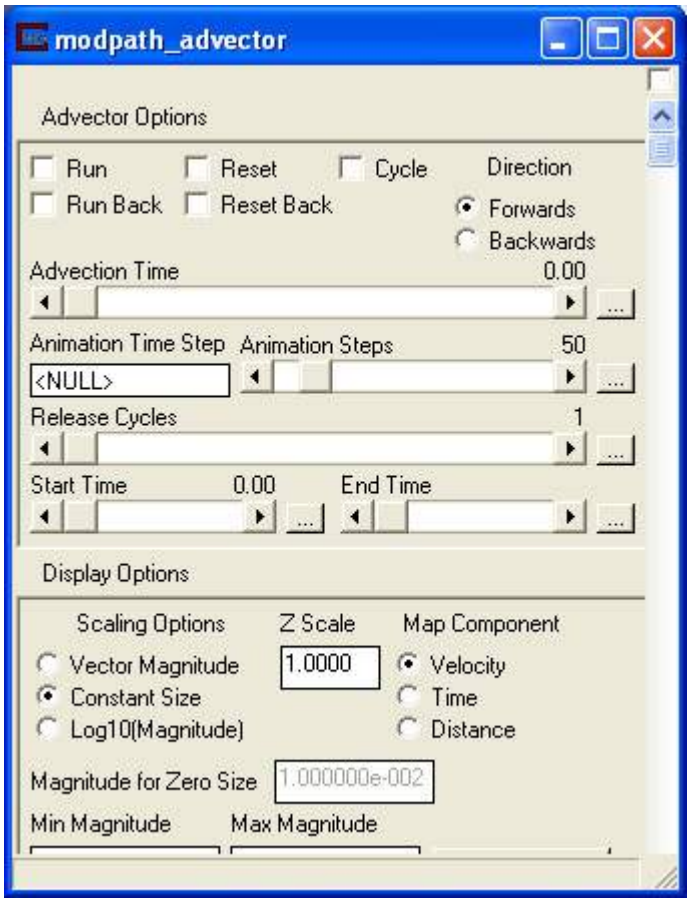
Min / Max Magnitude sets limits on the glyph scale.

The ***Glyph Scale*** slider adjusts the sizes of the glyphs. Default range is 0.0 to 100.00, but the type-in box allow for increasing the range to any value. The default scale of 1.00 is approximately 1/100th the x,y extents of the model.

The ***End percentile*** slider allows you to control the automatic scaling of glyphs based on the nth percentile value (versus the maximum 100th%). This addresses datasets where there are only a few nodes with extremely high values (like wells).

Z Scalecontrols the z exaggeration factor.

Map Componentselects which MODPATH data component to color the out put lines by.



Model Options:

The ***DWR File****:* shows which .dwr/.dwz file has been selected, this file contains MODFLOW package data such as drains, wells, recharge, etc... For details on the DWR file format, see here.

The ***Time Options*** radio buttons indicate a time based stopping condition for the particles.

Run to end indicates that the particle will run until it cannot escape a cell due to lack of flow.

Run for date allows a particle to travel from a start date to an end date.

The ***duration*** option allows the particles to run for a duration of time in days.

The ***Sink Options*** radio buttons indicate stopping conditions for the particles based on the amount of flow through a cell. The Pass through cells with weak sinks option allows all particles to pass through a cell, unless that cell has a strong sink. The Stop in cells with weak sinks button does not allow a particle to pass unless the flow out of a cell is greater than or equal the flow into it. The last option, Stop if discharge to sink is > specified fraction of inflow to cell, allows the user to specify what fraction of inflow will stop particles in that cell. This fraction value is set using the Fraction slider which is visible when the last option is selected. Only values between zero and one are useful.

drive_glyphs

General Module Function

The drive_glyph module provides a way to move any object (glyph or object from Read_DXF, etc.) along multiple paths to create a "driving" animation.

Module Input Ports

drive_glyphs has three input ports.

Data passed to the first port is the paths to follow (normally from read_lines).

The second port accepts the glyph or vehicle to drive, usually read in with the load_glyph module.

The third port is a float parameter for the position of the glyphs.

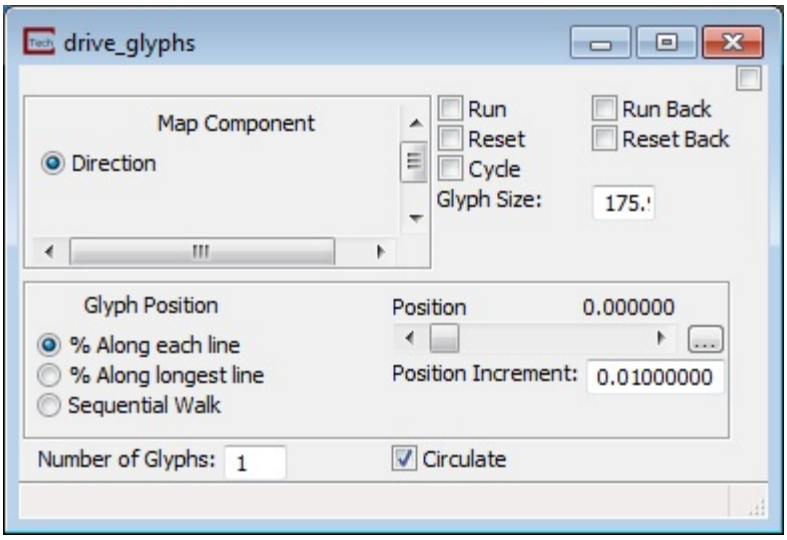
Module Output Ports

drive_glyph has three output ports.

The leftmost output port is a float parameter for the position of the glyphs along the input paths.

The center port is the animated glyphs.

The right output port is the animated glyphs in a renderable form for the viewer.



Module Control Panel

drive_glyph's control panel is shown above.

- The **Map Component** selector takes both the nodal data and the cell data from the input lines and allows the glyphs to be colored according to the selected component. For nodal data the color of the glyph is linearly interpolated between each node along the line, while with cell data the color of the glyph is constant throughout the cell.
- The **Run** toggle will increment the Position slider by the Position Increment until the value is 1.
- The **Run** Back toggle will decrement the Position slider by the Position Increment until the value reaches 0.
- The **Reset** toggle will set the Position to zero.
- The **ResetBack** toggle will set the Position to 1.
- The **Cycle** toggle will cause the Position slider to transition to the opposite end if the slider value reaches zero or one. This allows for a looping animation.
- The **Glyph Size** field sets the size of the glyph in real world units.
- **Glyph Position** indicates how each glyph will be placed along a line.
- If "***% Along each line*** is selected then the glyphs positions will be determined based upon each individual line length.
- If "***% Along longest line*** is selected then the glyphs position will be determined based on the longest line in the input.
- ***Sequential Walk*** will allow a glyph to walk along each line in succession.
- The **Position** slider gives the percentage along a line to use for glyph positioning.
- The **Position increment** changes the value the percentage is incremented by.

- **Number of glyphs** indicates how many glyphs at most will be present along the line.
- The **Circulate** toggle is used for looping or cycling animations. If a glyphs position is greater than the line length the glyph will be placed the appropriate distance from the start of the line. Or if the glyph position is before the end of the line it will be placed the appropriate distance from the end of the line. In example if you have selected 3 glyphs to be placed on a line (meaning that each glyph should be placed 33% of the line distance apart) and the Position is at zero. Then there will be one glyph at zero and another at 100 - 33 or 66% and a final glyph at 100 - 66 or 33% of the distance along the line length.

illuminated_lines

General Module Function

Display of Illuminated Lines using texture mapped illumination model on polylines with line halo and animation effects.

Prerequisites

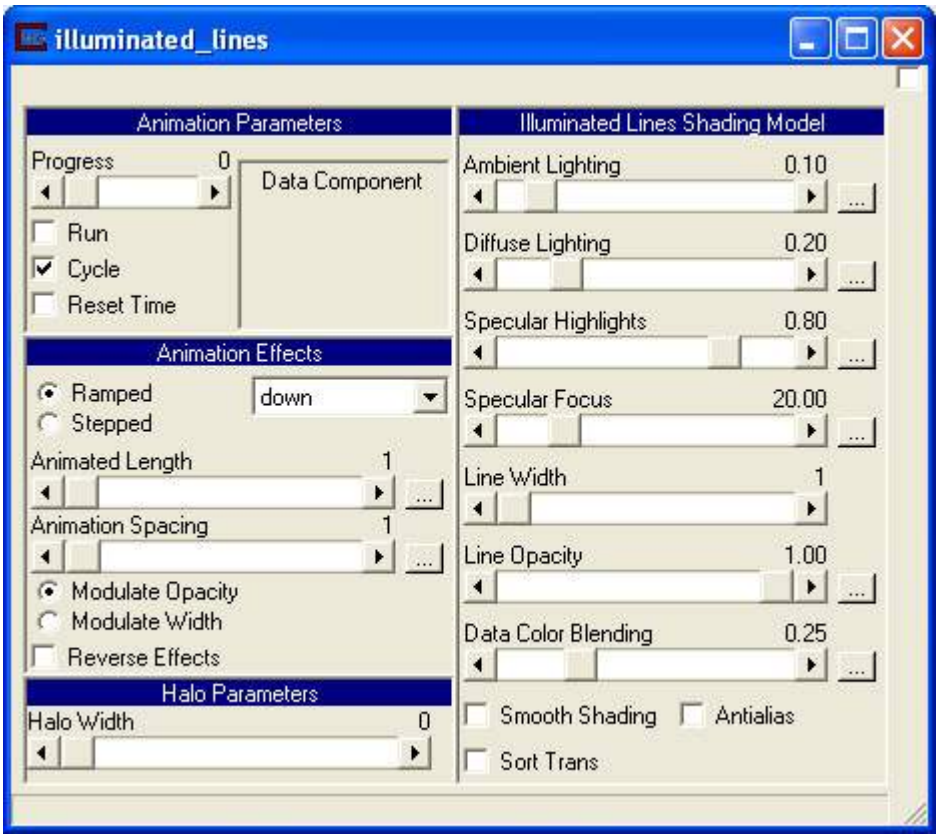
This module requires OpenGL rendering to be selected. This module utilizes special OpenGL calls to implement the illuminated line technique. If this module is used with another renderer, such as the software renderer or the Output_Images module (not set to Automatic), lines will be drawn in the default mode with illuminated line features disabled.

Module Input Ports

This module requires the input mesh to contain one Polyline cell set. Any other type of cell set will be rejected, and any additional cell sets will be ignored. Any scalar node data may be present, or none for purely geometric display.

Module Output Ports

illuminated_lines has one output port that connects to the viewer.



Module Control Panel

The user interface is shown above and detailed descriptions of each parameters are given below.

AnimationParameters

Progress : This slider displays the progress of the animation and (when not Running) allows you direct control of the animation by moving the slider in real-time.

Run : This toggle enables the animation loop to run.

Cycle : If enabled, the animation loop continues forever in a cyclic fashion. If unset, the animation stops after once complete cycle. This is similar to the cycle parameter of the "Loop" module.

Reset Time : This is one-shot control to reset the animation to the start of a cycle

Data Component allows you to choose which data component of your incoming lines you wish to color by. When input is from streamlines, the choices are velocity, time and distance.

Animation Effects

Ramped/Stepped This choice selects the style of effect variation. Stair creates a linearly increasing or decreasing value, while step makes a binary chop effect. In *Ramped* mode, the blending can be selected to start small then get big, or the reverse or both. The values are *down, up, up&down* respectively. *Stepped* causes abrupt changes in effect.

AnimatedLength This slider sets the length of the effect along the polyline.

AnimationSpacing This slider sets the spacing between effects along the line.

ModulateOpacity In this mode the line segment varies in transparency from completely transparent to opaque.

ModulateWidth In this mode the line width is varied between 1 (very thin) to fat, based on the effect modes and shape controls.

Reverse Effect As the animation effect is applied between two zones, such as the dash and the space between the dash, this toggle reverses the area where the effect is applied.

Halo Parameters

Halo Width The width control for the halo effect defines the size of the transparent mask region added to the edge of each line. A value of zero turns off the halo effect.

Illuminated Lines Shading Model

AmbientLighting This value provides a base shadow value, a constant added to all shading values.

DiffuseLighting Pure diffuse reflection term, amount of shading dependent on light angle

SpecularHighlights Amount of specular reflection hi-lights based on light and viewer angle

Specular Focus Tightness of specular reflection, low values are dull, wide reflections, high values are small spot reflections.

Line Width Controls line width. Normal 1-pixel lines are 1, can be increased in whole increments. Wide lines are drawn in 2D screen space, not full 3D ribbons. If you want full ribbons, use streamline module ribbon mode.

Line Opacity Variable transparency of all lines. A value of 1.0 is fully opaque, while a value of zero makes lines invisible.

DataColor Blending If node data is present, this controls the relative mix of data color and shading color. A value of zero sets full contribution of data color, while at 1.0 no data color is used and the line shade is dominated by illumination effects.

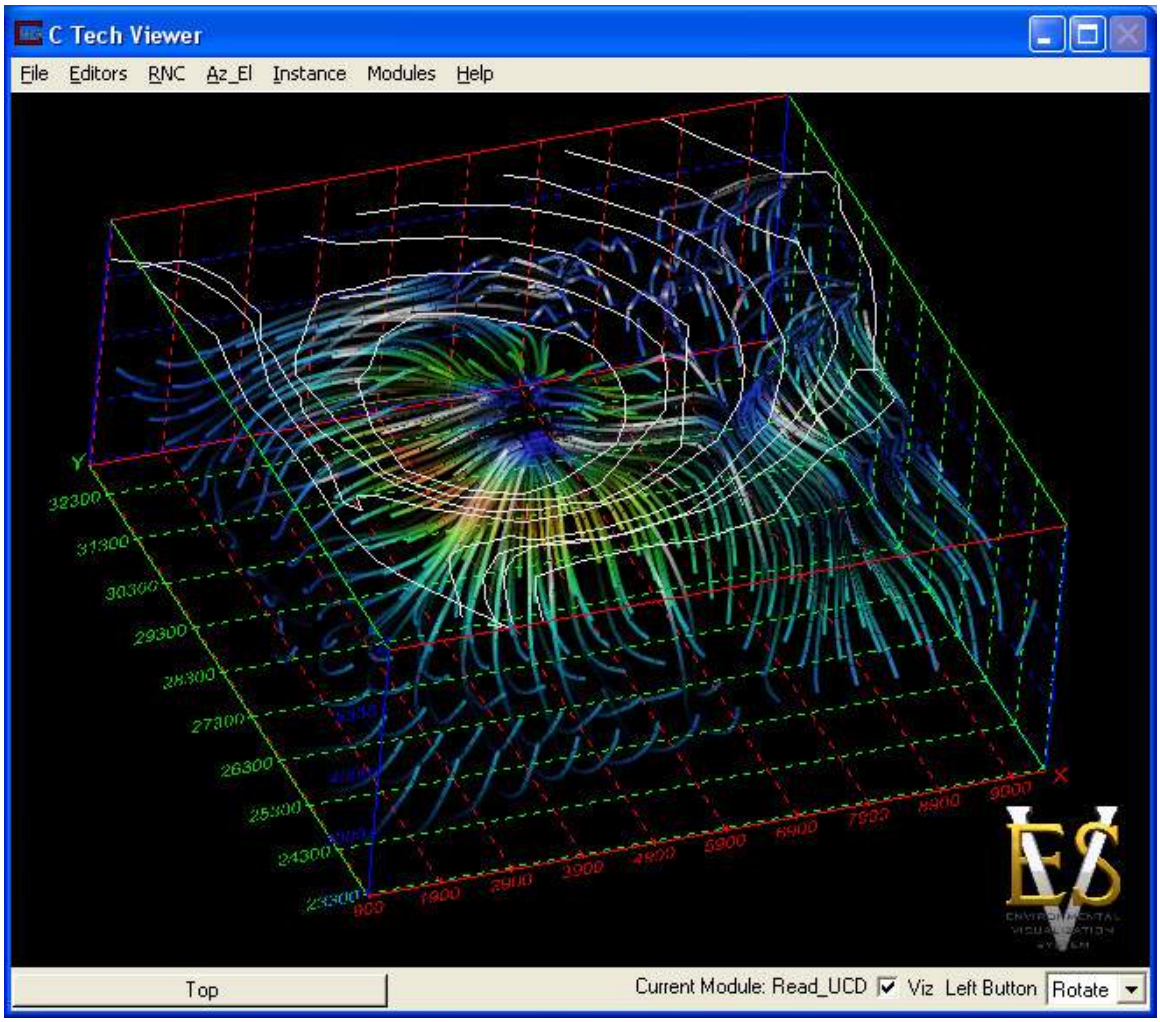
Smooth Shading This enables an additional interpolation mode for blended node data colors. In the off state, data is sampled once per line segment. When enabled, linear interpolation is used between end points of each segment. This can be helpful if large gradients are present on low resolution polylines.

Antialias This effect, sometimes called "smooth lines" blends the drawing of lines to create a smooth effect, reducing the effects of "jaggies" at pixel resolution.

Sort Trans This mode assists visual quality when transparency or antialiasing modes are used, helping to reduce artifacts caused by non-depth sorted line crossings.

Example

An example of the output from illuminated_lines is shown below.



texture_wave

General Module Function

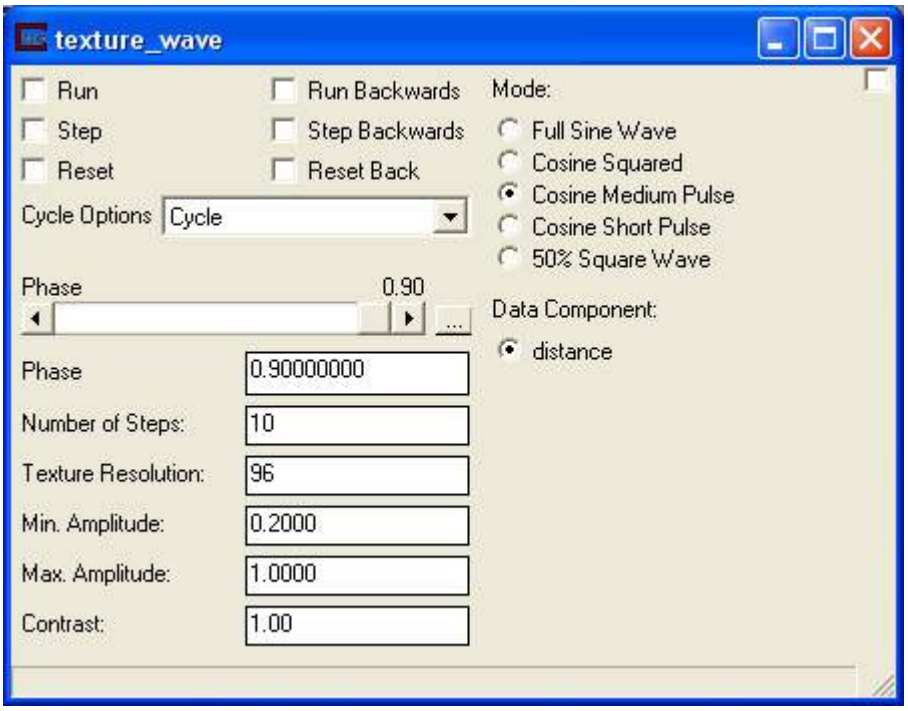
The texture_wave module utilizes transparency and texture mapping similar to texture_colors and illuminated_lines technology to create an animated effect. However, unlike illuminated_lines, this module works with both OpenGL and Software Rendering.

Module Input Ports

texture_wave has a single input port that accepts the grid with nodal data that you want to color with this technique. This would normally be tubes or streamribbons.

Module Output Ports

texture_wave has only one output port that can be connected to the Viewer.



Module Control Panel

The control panel for texture_wave is shown in the figure above.

The ***Run*** toggle enables the animation loop to run.

The ***Step*** toggle will make one increment of the loop as specified int the Increment type-in.

Reset, when toggled, resets the animation.

The ***Run Backwards***, ***Step Backwards*** and ***Reset Back*** are self explanatory.

The ***Cycle Options*** allow for Once (one iteration), Cycle (keep running from beginning), and Bounce (run to end, then backwards, then forwards, etc.)

The **Phase** is the parameter that changes during the animation loop.

Number of Steps: determines the number of steps in the animation.

Texture Resolution is the internal resolution of the image used for texture-coloring.

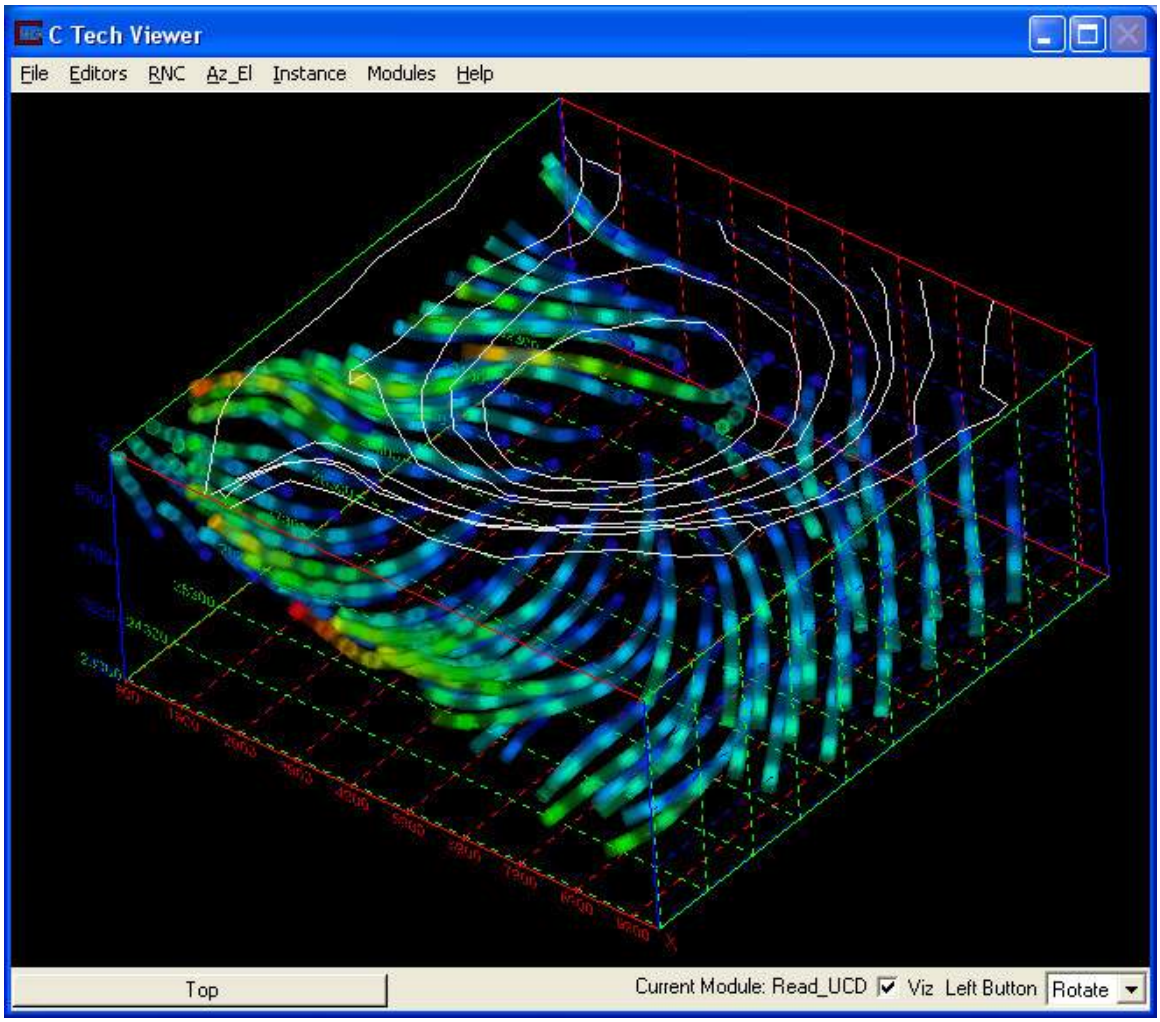
Min Amplitude is the minimum opacity of the objects.

Max Amplitude is the maximum opacity of the objects.

Contrast affects the contrast (similar to color saturation).

In the image below, we used streamlines which are passed to tubes, which are then connected to texture_wave. The transparency, colors, and animation effects on the tubes is all performed by texture_wave.

The Viewer window is shown below.



Read_TCF

General Module Function

The Read_TCF module is one of a limited set of Time_Data modules. These modules are specifically designed to create models and animations of data that changes over time. This type of data can result from water table elevation and/or chemical measurements taken at discrete times or output from Groundwater simulations or other 3D time-domain simulations.

The Read_TCF module creates a field using a Time Control File (.TCF) to specify the date/time, field and corresponding data component to read (in netCDF, Field or UCD format), for each time step of a time_data field. All file types specified in the TCF file must be the same (e.g. all netCDF or all UCD). The same file can be repeated, specifying different data components to represent different time steps of the output.

Read_TCF effectively includes an inter_time_step module internally in that it performs the interpolation between appropriate pairs of the files/data_components specified in the TCF file. Its internal structure only

requires reading two successive time steps rather than the complete listing of time steps normally represented in a time_data field.

Module Input Ports

Read_TCF has one input port that accepts the current time.

Module Output Ports

Read_TCF has four output ports. From left to right:

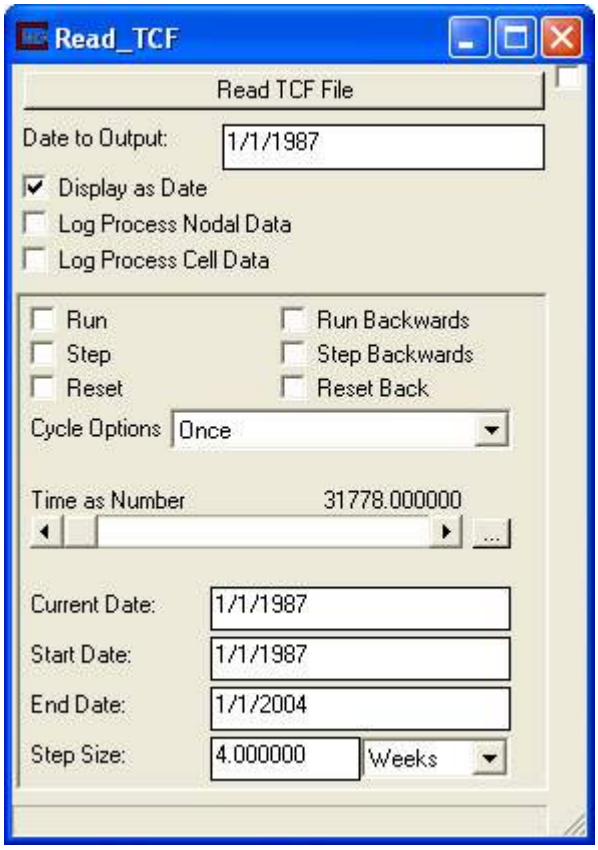
The first port outputs the starting time.

The second port outputs the ending time.

The third port outputs the current time.

The fourth port outputs a normal field (mesh/grid plus data) representing the interpolated time step.

Module Control Panel



The control panel for Read_TCF is shown in the figure above.

The ***Read TCF File*** button allows you to specify the TCF file to read.

Click here for an example of a TCF fileand a description of the format.

The ***Date to Output*** is a type-in box that displays (or allows you to enter) the date and time for the interpolated time step.

The ***Display as Date*** toggle, if off displays the date as a number.

The ***Log Process Nodal Data*** toggle is used if the nodal data is log processed so that the interpolation is linear vs. geometric.

The ***Log Process Cell Data*** toggle is used if the cell data is log processed so that the interpolation is linear vs. geometric.

The cycle options are the same as those found in the loop module and an in depth help discussion can be found for them there.

TCF File Format and Example

The listing below is the full contents of the Time Control File ***control_tce_cdf.tcf***. Blank lines or any lines beginning with a "#" are ignored. Valid lines representing time steps must be in order of ascending time and consisting of:

a) a date and/or time in Windows standard format

b) a file name with an absolute path or just the filename (if the data files are in the same directory as the TCF file). **This is not a true relative path** (..\file.cdf and subdir\file.cdf don't work, but file.cdf does), but gives some of the relative path abilities.

c) the data component to use for that time step. (You can specify -1 in the third column, which causes ALL the data components to pass through.)

NOTE: These three items on each line **must** be separated with a comma ",".

```
# This file contains the list of control commands for the
# TCE time data in netCDF format.

# The format is a date/time, then the file, then the nodal data
component.
# The END on the last line is optional.

# Each line MUST be comma delimited
# (since spaces can exist in the time and filename)

6/1/1990 12:00 AM, $XP_PATH<0>/data/netcdf/time_data/tce_01.cdf,
0
12/1/1990, $XP_PATH<0>/data/netcdf/time_data/tce_02.cdf, 0
2/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_03.cdf, 0
5/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_04.cdf, 0
8/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_05.cdf, 0
11/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_06.cdf, 0
3/1/1992, $XP_PATH<0>/data/netcdf/time_data/tce_07.cdf, 0
6/1/1992, $XP_PATH<0>/data/netcdf/time_data/tce_08.cdf, 0
10/1/1992, $XP_PATH<0>/data/netcdf/time_data/tce_09.cdf, 0
3/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_10.cdf, 0
4/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_11.cdf, 0
8/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_12.cdf, 0
12/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_13.cdf, 0
```

```
3/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_14.cdf, 0
6/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_15.cdf, 0
9/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_16.cdf, 0
11/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_17.cdf, 0
3/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_18.cdf, 0
5/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_19.cdf, 0
8/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_20.cdf, 0
10/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_21.cdf, 0
1/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_22.cdf, 0
5/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_23.cdf, 0
9/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_24.cdf, 0
11/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_25.cdf, 0
12/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_26.cdf, 0
3/1/1997 12:00 AM, $XP_PATH<0>/data/netcdf/time_data/tce_27.cdf,
0
6/1/1997, $XP_PATH<0>/data/netcdf/time_data/tce_28.cdf, 0
9/1/1997, $XP_PATH<0>/data/netcdf/time_data/tce_29.cdf, 0
12/1/1997, $XP_PATH<0>/data/netcdf/time_data/tce_30.cdf, 0
3/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_31.cdf, 0
6/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_32.cdf, 0
9/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_33.cdf, 0
11/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_34.cdf, 0
5/1/1999, $XP_PATH<0>/data/netcdf/time_data/tce_35.cdf, 0
10/1/1999, $XP_PATH<0>/data/netcdf/time_data/tce_36.cdf, 0
3/1/2000, $XP_PATH<0>/data/netcdf/time_data/tce_37.cdf, 0
7/1/2000, $XP_PATH<0>/data/netcdf/time_data/tce_38.cdf, 0
11/1/2000, $XP_PATH<0>/data/netcdf/time_data/tce_39.cdf, 0
3/1/2001, $XP_PATH<0>/data/netcdf/time_data/tce_40.cdf, 0
5/1/2001, $XP_PATH<0>/data/netcdf/time_data/tce_41.cdf, 0
10/1/2001, $XP_PATH<0>/data/netcdf/time_data/tce_42.cdf, 0

END
```

Read_Multi_TCF

General Module Function

The Read_Multi_TCF module is one of a limited set of Time_Data modules. These modules are specifically designed to create models and animations of data that changes over time. This type of data can result from water table elevation and/or chemical measurements taken at discrete times or output from Groundwater simulations or other 3D time-domain simulations.

The Read_Multi_TCF module creates a field using one or more Time Control Files (.TCF). **Click here for an example of a TCF fileand a description of the format.**

The Read_Multi_TCF module creates a mesh grid with the interpolated data from a user specifed number of TCF files (n). It outputs the first data component from the first (n-1) TCF files and all of the time interpolated data components from the nth TCF file.

For example, if you were trying to create a time animation of the union of 3 analytes (e.g. Benzene, Toluene & Xylene), Read_Multi_TCF allows you to select all three separate TCF files. Only the first data component from Benzene.tcf (nominally the concentration of benzene) is output as the new first data component. The first data component from Toluene.tcf (nominally the concentration of toluene) is output as the new second data component. All of the data components from Xylene.tcf are then output (typically xylene, confidence_xylene, uncertainty_xylene, Geo_Layer, Material_ID, Elevation, etc.). This allows you to explode layers and do other typical subsetting and processing operations on the output of this module.

The TCF files should be created using identical grids with date ranges that overlap the time period of interest.

Read_Multi_TCF effectively includes an inter_time_step module internally in that it performs the interpolation between appropriate pairs of the files/data_components specified in the TCF file. Its internal structure only requires reading two successive time steps rather than the complete listing of time steps normally represented in a time_data field.

Module Input Ports

Read_Multi_TCF has one input port that accepts the current time.

Module Output Ports

Read_Multi_TCF has four output ports. From left to right:

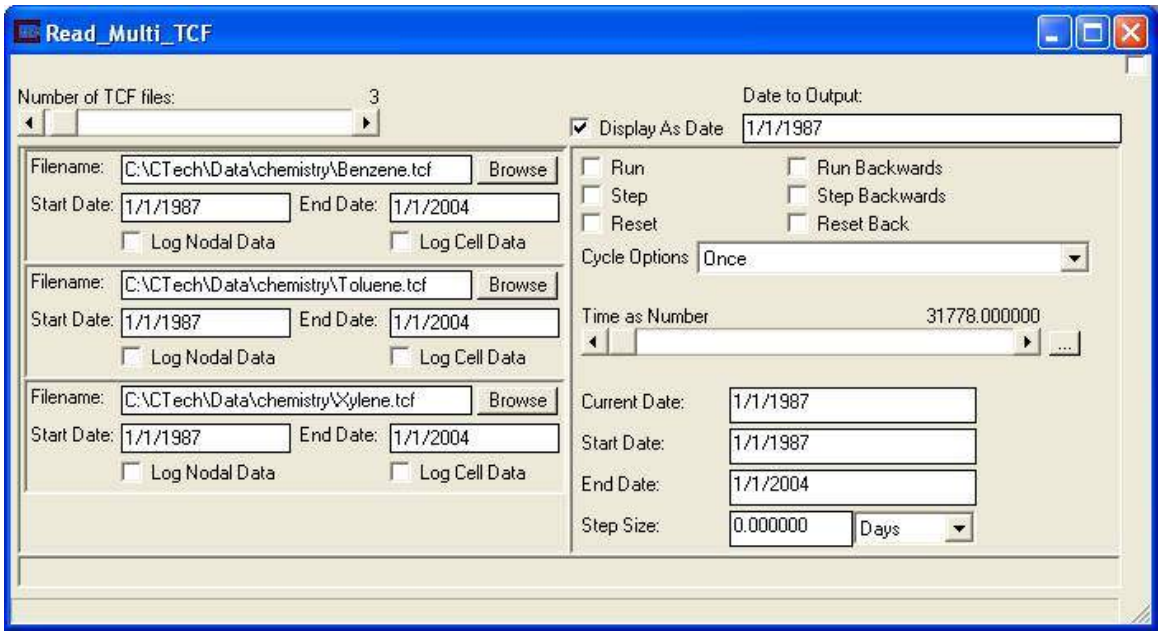
The first port outputs the starting time.

The second port outputs the ending time.

The third port outputs the current time.

The fourth port outputs a normal field (mesh/grid plus data) representing the interpolated time step.

Module Control Panel



The control panel for Read_Multi_TCF is shown in the figure above. Please note that the example above had three TCF files read as per the discussion about unions above.

The ***Number of TCF files*** indicates the number of TCF files the user wishes to read in.

The ***Browse*** button allows you to specify the TCF file to read.

The ***Start DAte / End Date*** fields indicate the starting and ending dates found in the TCF file.

The ***Log Process Nodal Data*** toggle is used if the nodal data is log processed so that the interpolation is linear vs. geometric.

The ***Log Process Cell Data*** toggle is used if the cell data is log processed so that the interpolation is linear vs. geometric.

The ***Display as Date*** toggle, if off displays the date as a number.

The ***Date to Output*** is a type-in box that displays (or allows you to enter) the date and time for the interpolated time step.

The cycle options are the same as those found in the time_loop module and an in depth help discussion can be found for them there.

TCF File Format and Example

The listing below is the full contents of the Time Control File ***control_tce_cdf.tcf***. Blank lines or any lines beginning with a "#" are ignored. Valid lines representing time steps must be in order of ascending time and consisting of:

a) a date and/or time in Windows standard format

b) a file name with an absolute path or just the filename (if the data files are in the same directory as the TCF file). **This is not a true relative**

path (..\file.cdf and subdir\file.cdf don't work, but file.cdf does), but gives some of the relative path abilities.

c) the data component to use for that time step. (You can specify -1 in the third column, which causes ALL the data components to pass through.)

NOTE: These three items on each line **must** be separated with a comma ",".

```
# This file contains the list of control commands for the
# TCE time data in netCDF format.

# The format is a date/time, then the file, then the nodal data
component.
# The END on the last line is optional.

# Each line MUST be comma delimited
# (since spaces can exist in the time and filename)

6/1/1990 12:00 AM, $XP_PATH<0>/data/netcdf/time_data/tce_01.cdf,
0
12/1/1990, $XP_PATH<0>/data/netcdf/time_data/tce_02.cdf, 0
2/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_03.cdf, 0
5/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_04.cdf, 0
8/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_05.cdf, 0
11/1/1991, $XP_PATH<0>/data/netcdf/time_data/tce_06.cdf, 0
3/1/1992, $XP_PATH<0>/data/netcdf/time_data/tce_07.cdf, 0
6/1/1992, $XP_PATH<0>/data/netcdf/time_data/tce_08.cdf, 0
10/1/1992, $XP_PATH<0>/data/netcdf/time_data/tce_09.cdf, 0
3/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_10.cdf, 0
4/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_11.cdf, 0
8/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_12.cdf, 0
12/1/1993, $XP_PATH<0>/data/netcdf/time_data/tce_13.cdf, 0
3/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_14.cdf, 0
6/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_15.cdf, 0
9/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_16.cdf, 0
11/1/1994, $XP_PATH<0>/data/netcdf/time_data/tce_17.cdf, 0
3/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_18.cdf, 0
5/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_19.cdf, 0
8/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_20.cdf, 0
10/1/1995, $XP_PATH<0>/data/netcdf/time_data/tce_21.cdf, 0
1/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_22.cdf, 0
5/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_23.cdf, 0
9/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_24.cdf, 0
11/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_25.cdf, 0
```

```
12/1/1996, $XP_PATH<0>/data/netcdf/time_data/tce_26.cdf, 0
3/1/1997 12:00 AM, $XP_PATH<0>/data/netcdf/time_data/tce_27.cdf,
0
6/1/1997, $XP_PATH<0>/data/netcdf/time_data/tce_28.cdf, 0
9/1/1997, $XP_PATH<0>/data/netcdf/time_data/tce_29.cdf, 0
12/1/1997, $XP_PATH<0>/data/netcdf/time_data/tce_30.cdf, 0
3/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_31.cdf, 0
6/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_32.cdf, 0
9/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_33.cdf, 0
11/1/1998, $XP_PATH<0>/data/netcdf/time_data/tce_34.cdf, 0
5/1/1999, $XP_PATH<0>/data/netcdf/time_data/tce_35.cdf, 0
10/1/1999, $XP_PATH<0>/data/netcdf/time_data/tce_36.cdf, 0
3/1/2000, $XP_PATH<0>/data/netcdf/time_data/tce_37.cdf, 0
7/1/2000, $XP_PATH<0>/data/netcdf/time_data/tce_38.cdf, 0
11/1/2000, $XP_PATH<0>/data/netcdf/time_data/tce_39.cdf, 0
3/1/2001, $XP_PATH<0>/data/netcdf/time_data/tce_40.cdf, 0
5/1/2001, $XP_PATH<0>/data/netcdf/time_data/tce_41.cdf, 0
10/1/2001, $XP_PATH<0>/data/netcdf/time_data/tce_42.cdf, 0

END
```

time_value

General Module Function

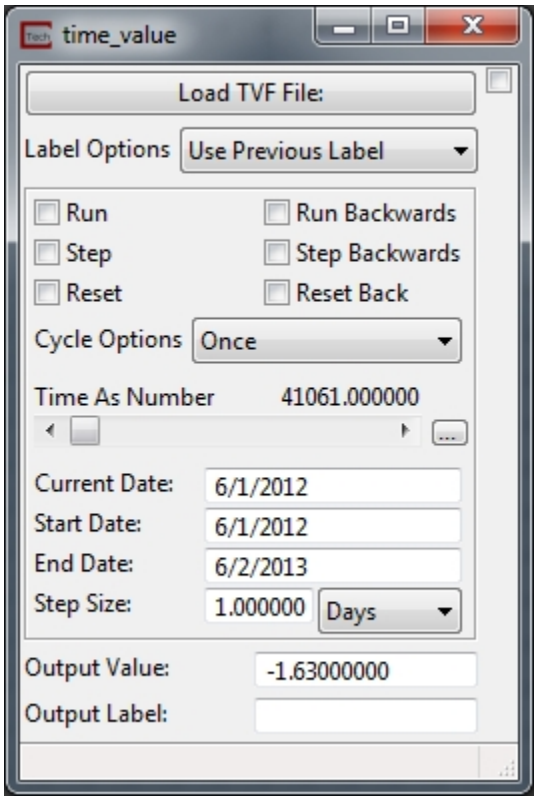
The time_value module is used to parse a TVF file consisting of dates, values, and (optional) labels. The starting and end dates are read from the file and the controls can be used to interpolate the values to the date and time of interest.

Input Ports

- DateAndTime (Grey-Red) : This port is used to set the current date and time of interest if there are other time animations modules in use.

Output Ports

- StartDate (Grey-Red) : This port is used to output the first date read from the file.
- EndDate (Grey-Red) : This port is used to output the last date read from the file.
- DateAndTime (Grey-Red) : This port is used to output the current date and time of interest.
- CurrentDateAndTimeLabel (Teal) : This port is used to output the current label associated with the date and time of interest.
- CurrentDateAndTimeValue (Maroon) : This port is used to output the current interpolated value associated with the date and time of interest.



Module Control Panel

Use the *Load TVF File* button to load the (*.tvf) file of interest. Dates must be in order from earliest to latest and not repeating. Only the label column is optional.

If labels are present in the file then *Label Options* allow the user to select:

- *Use Previous Label*: either use the last valid label from a previous date
- *Use Nearest Label*: use label closest to the current date.

Note: The empty string "" is a valid label.

The *Cycle Options* are the same as those found in the loop module and an in depth help discussion can be found for them there.

Output Value is the current interpolated value associated with the date and time of interest.

Output Label is the current label associated with the date and time of interest.

TVF File Format

TVF files provide a way to generate a time varying numeric and option string (label). The file is similar to the TCF file, but does not reference information in external files.

The file consists of two or more rows, each having 2 or 3 columns of information. The columns must contain:

1. Date and/or time in Windows standard format
2. A numeric (float) value (required)
3. A string consisting of one or more words. These need not be in quotes. Everything on the row after the numeric value will be used. (optional)

Dates must be in order from earliest to latest and not repeating. Only the label column is optional.

An example file follows:

```
06/01/12    -1.63   Spring Rains

06/04/12    -1.87

06/07/12    -2.17

06/10/12    -1.87

06/13/12    -1.9

06/16/12    -2.2

06/19/12    -1.9

06/22/12    -1.96   Summer

06/25/12    -1.81

06/28/12    -1.84

07/01/12    -1.69

07/04/12    -1.39

07/07/12    -1.33

07/10/12    -1.12

07/13/12    -0.85
```

```
07/16/12    -1.03

07/19/12    -1.06

07/22/12    -0.76

07/25/12    -0.61 Flood Event

07/28/12    -0.31

07/31/12    -0.31

08/03/12    -0.52

08/06/12    -0.37

08/09/12    -0.61

08/12/12    -0.85

08/15/12    -0.79

08/18/12    -0.76

08/21/12    -0.58

08/24/12    -0.64

08/27/12    -0.49

08/30/12    -0.46

09/02/12    -0.67

09/05/12    -0.91

09/08/12    -0.82

09/11/12    -1.09 ""

09/14/12    -1.27

09/17/12    -1.3

09/20/12    -1.33

09/23/12    -1.51   Fall

09/26/12    -1.42

09/29/12    -1.69
```

```
10/02/12    -1.69

10/05/12    -1.78

10/08/12    -1.84

10/11/12    -1.96

10/14/12    -2.17

10/17/12    -2.29

10/20/12    -2.26

10/23/12    -2.05

10/26/12    -2.05

10/29/12    -1.84

11/01/12    -2.05

11/04/12    -2.23

11/07/12    -2.08

11/10/12    -2.2

11/13/12    -2.41

11/16/12    -2.62

11/19/12    -2.83

11/22/12    -2.62

11/25/12    -2.5

11/28/12    -2.29

12/01/12    -2.11

12/04/12    -2.2

12/07/12    -1.9

12/10/12    -2.08

12/13/12    -1.93

12/16/12    -1.81
```

```
12/19/12    -1.75

12/22/12    -1.63   Winter

12/25/12    -1.36

12/28/12    -1.45

12/31/12    -1.24

01/03/13    -1.21

01/06/13    -1

01/09/13    -1.27

01/12/13    -1.21

01/15/13    -1.18

01/18/13    -1.15

01/21/13    -1.12

01/24/13    -1.33

01/27/13    -1.39

01/30/13    -1.24

02/02/13    -1.3

02/05/13    -1.57

02/08/13    -1.66

02/11/13    -1.81

02/14/13    -1.69

02/17/13    -1.78

02/20/13    -1.78

02/23/13    -1.84

02/26/13    -1.72

03/01/13    -2.02

03/04/13    -2.23
```

```
03/07/13    -2.08

03/10/13    -2.02

03/13/13    -2.32

03/16/13    -2.11

03/19/13    -2.41

03/22/13    -2.65   Spring

03/25/13    -2.38

03/28/13    -2.47

03/31/13    -2.47

04/03/13    -2.32

04/06/13    -2.17

04/09/13    -2.14

04/12/13    -2.41

04/15/13    -2.65

04/18/13    -2.47

04/21/13    -2.35

04/24/13    -2.32

04/27/13    -2.38

04/30/13    -2.08

05/03/13    -1.93

05/06/13    -1.84

05/09/13    -1.57

05/12/13    -1.84

05/15/13    -1.57

05/18/13    -1.57

05/21/13    -1.69
```

```
05/24/13    -1.93

05/27/13    -1.78

05/30/13    -1.57

06/02/13    -1.84
```

time_geology

General Module Function

The time_geology module allows you to extract a surface from a set of time-based surfaces. The time for the extracted surface can be any time between the start and end of the surface set. It will interpolate between adjacent known times.

Module Input Ports

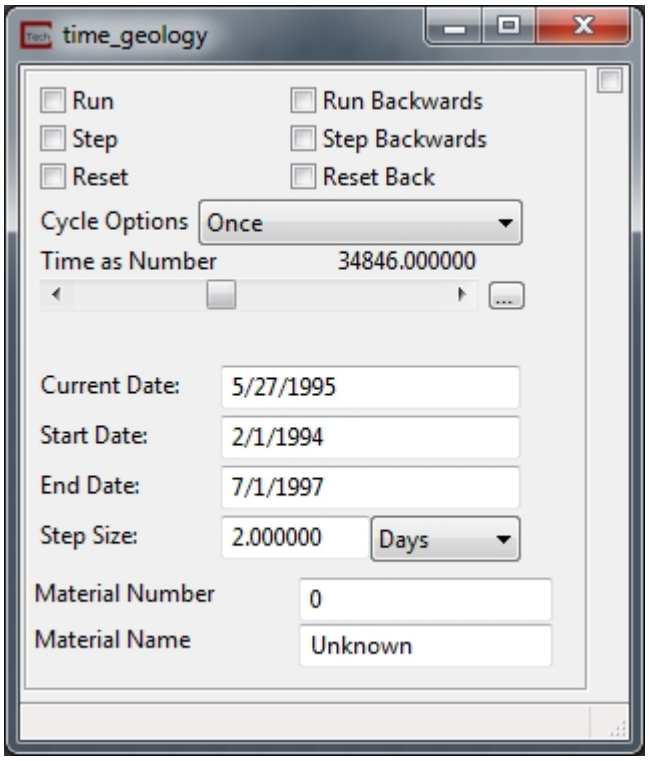
time_geology has two input ports.

1. The leftmost port accepts a set of surfaces from Krig_3D_Geology, Spline_Geology or the output of those modules saved as an EVS Field File (e.g. .EFF).
2. The right port is the current time.

Module Output Ports

time_geology has five output ports.

1. The first port (grey-red) outputs the starting time.
2. The second port (grey-red) outputs the ending time.
3. The third port (grey-red) outputs the current time.
4. The fourth (blue-white-green) outputs the interpolated surface (geology format) at the specified time.
5. The fifth port (brown-grey-green/brown-yellow/brown) rovides geologic material information for the Legend module.



The control panel for time_geology is shown in the figure above.

- The ***Run*** toggle when set will kick off an iteration of the loop. It is reset to off when it is finished.
- The ***Step*** toggle will make one increment of the loop as specified by the Step Size (and units).
- ***Reset***, when toggled, resets the Current Date to the Start Date
- The ***Run Backwards***, ***Step Backwards*** and ***Reset Back*** are self explanatory.
- The ***Cycle Options*** allow for Once (one iteration), Cycle (keep running from beginning), and Bounce (run to end, then backwards, then forwards, etc.)
- The **Time as Number** slider is updated during running of the module, or may be controlled by moving it in real-time. but the user may jump to any part of the specified iteration by typing in a valid value
- ***CurrentDate*** shows the current date represented by the time number. You may enter dates in standard short date format.
- ***StartDate*** indicates the starting value of the loop and sets the lower limit of the Time as Number slider.
- ***EndDate*** indicates the ending value of the loop and sets the upper limit of the Time as Number slider.
- ***Step Size*** indicates the time interval for each step of the loop. The units pull down menu lets you choose whether the step size represents *Seconds, Minutes, Hours, Days, Weeks or Years.*

- ***Material Number***provides the ability to set the material number for the geologic info port
- ***Material Name***provides the ability to set the material name for the geologic info port

time_field

General Module Function

The time_field module allows you to extract a field (grid with data) from a set of time-based fields. The time for the extracted field can be any time between the start and end of the set of fields. It will interpolate between adjacent known times.

Module Input Ports

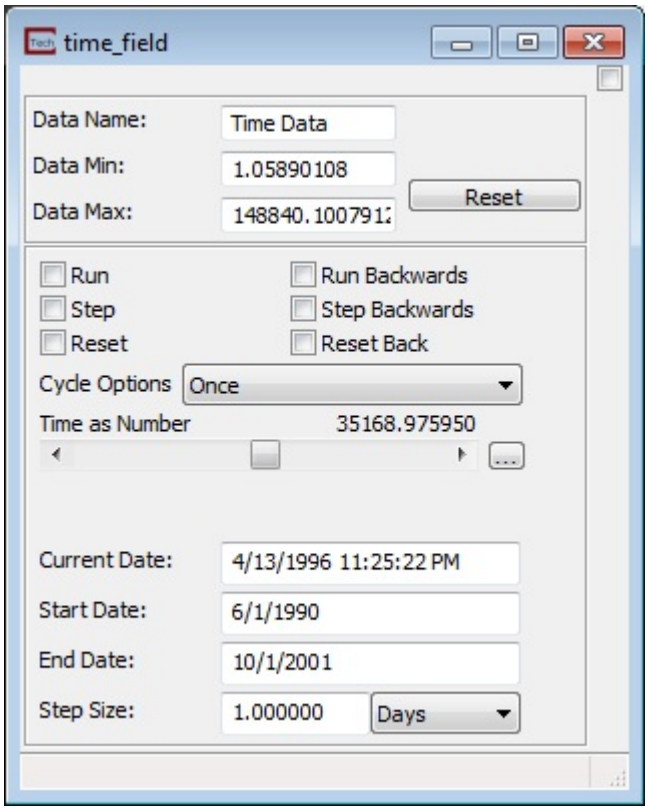
time_field has two input ports.

1. The leftmost port accepts any 1D, 2D or 3D mesh. The mesh can contain nodal data and/or cell data.
2. The right port is the current time.

Module Output Ports

time_field has four output ports.

1. The first port outputs the starting time.
2. The second port outputs the ending time.
3. The third port outputs the current time.
4. The fourth (blue/black) outputs the interpolated field at the specified time.



The control panel for time_field is shown in the figure above.

- ***Data Name*** allows you to specify the name of the interpolated field. Normally this will be the analyte name.
- ***Data Min*** allows you to specify the minimum value you wish to display over the entire time range.
- ***Data Max*** allows you to specify the maximum value you wish to display over the entire time range.
- The ***Run*** toggle when set will kick off an iteration of the loop. It is reset to off when it is finished.
- The ***Step*** toggle will make one increment of the loop as specified by the Step Size (and units).
- ***Reset***, when toggled, resets the Current Date to the Start Date
- The ***Run Backwards***, ***Step Backwards*** and ***Reset Back*** are self explanatory.
- The ***Cycle Options*** allow for Once (one iteration), Cycle (keep running from beginning), and Bounce (run to end, then backwards, then forwards, etc.)
- The **Time as Number** slider is updated during running of the module, or may be controlled by moving it in real-time. but the user may jump to any part of the specified iteration by typing in a valid value
- ***CurrentDate*** shows the current date represented by the time number. You may enter dates in standard short date format.

- ***StartDate*** indicates the starting value of the loop and sets the lower limit of the Time as Number slider.
- ***EndDate*** indicates the ending value of the loop and sets the upper limit of the Time as Number slider.
- ***Step Size*** indicates the time interval for each step of the loop. The units pull down menu lets you choose whether the step size represents *Seconds, Minutes, Hours, Days, Weeks or Years.*

time_loop

General Module Function

The time_loop module is one of a limited set of Time_Data modules. These modules are specifically designed to create models and animations of data that changes over time. This type of data can result from water table elevation and/or chemical measurements taken at discrete times or output from Groundwater simulations or other 3D time-domain simulations.

The time_loop module allows you to loop through a series of times or specify a time for interpolation from a time field.

Module Input Ports

time_loop has three input ports. From left to right:

The first port accepts the starting time.

The second port accepts the ending time.

The third port accepts the current time.

Module Output Ports

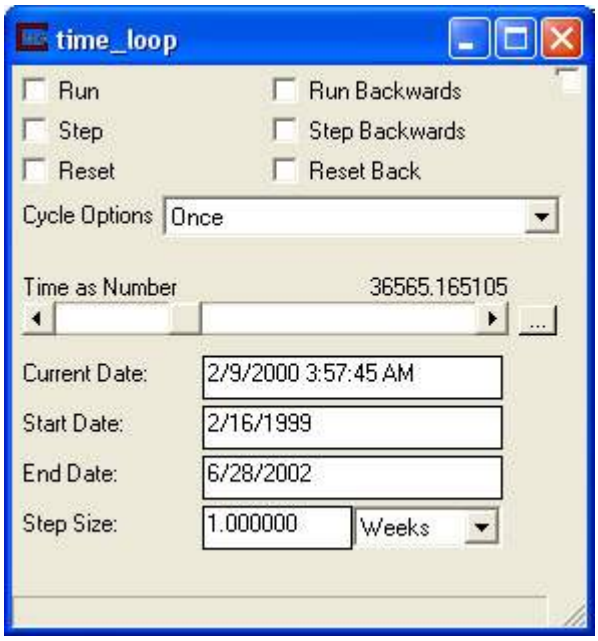
time_loop has three output ports. From left to right:

The first port outputs the starting time.

The second port outputs the ending time.

The third port outputs the current time.

Module Control Panel



The control panel for time_loop is shown in the figure above.

The ***Run*** toggle when set will kick off an iteration of the loop. It is reset to off when it is finished.

The ***Step*** toggle will make one increment of the loop as specified by the Step Size (and units).

Reset, when toggled, resets the Current Date to the Start Date

The ***Run Backwards***, ***Step Backwards*** and ***Reset Back*** are self explanatory.

The ***Cycle Options*** allow for Once (one iteration), Cycle (keep running from beginning), and Bounce (run to end, then backwards, then forwards, etc.)

The **Time as Number** slider is updated during running of the module, or may be controlled by moving it in real-time. but the user may jump to any part of the specified iteration by typing in a valid value

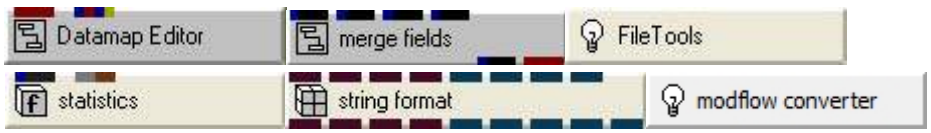
CurrentDate shows the current date represented by the time number. You may enter dates in standard short date format.

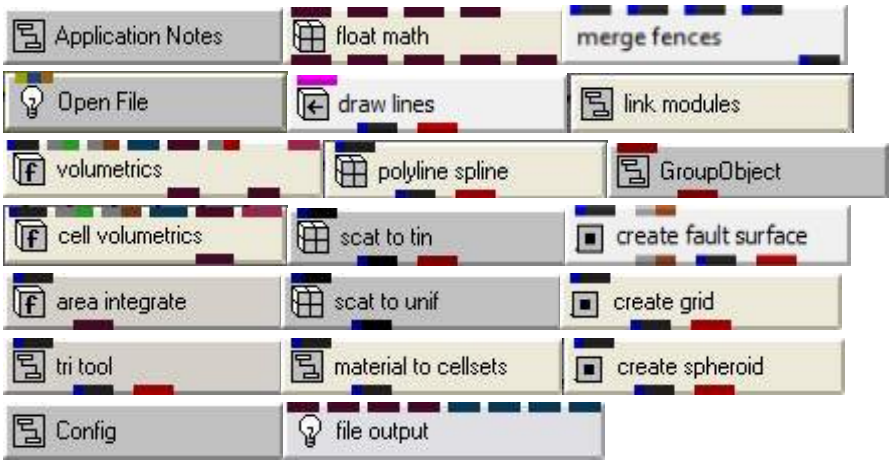
StartDate indicates the starting value of the loop and sets the lower limit of the Time as Number slider.

EndDate indicates the ending value of the loop and sets the upper limit of the Time as Number slider.

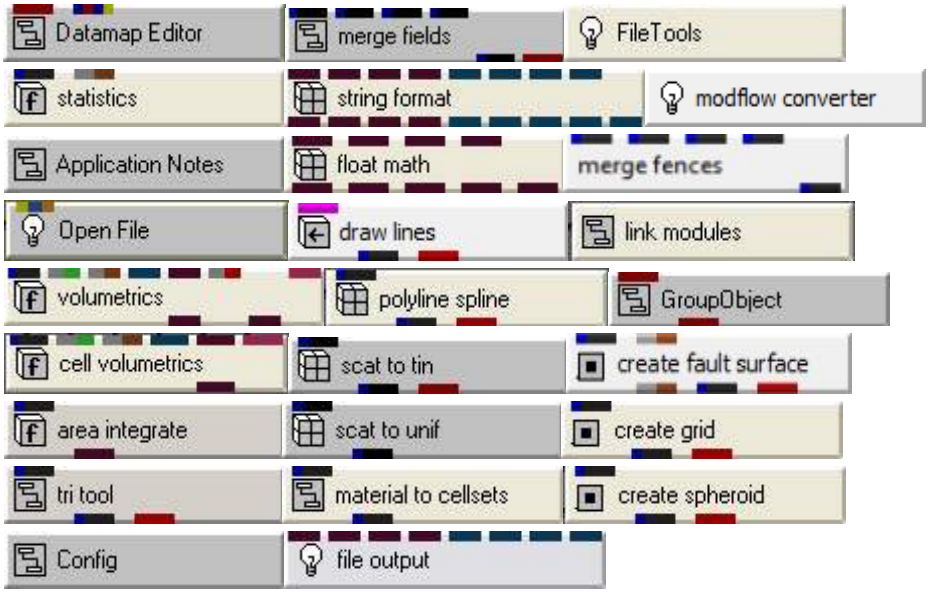
Step Size indicates the time interval for each step of the loop. The units pull down menu lets you choose whether the step size represents *Seconds, Minutes, Hours, Days, Weeks or Years.*

Tools Modules

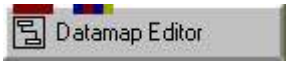




Tools Modules



Datamap_Editor



General Module Function

This simple module allows the user to create complex multi-range datamaps. The user simply connects the Datamap_Editor module to the object's red port.

Module Input Ports

Datamap_Editor has two input ports.

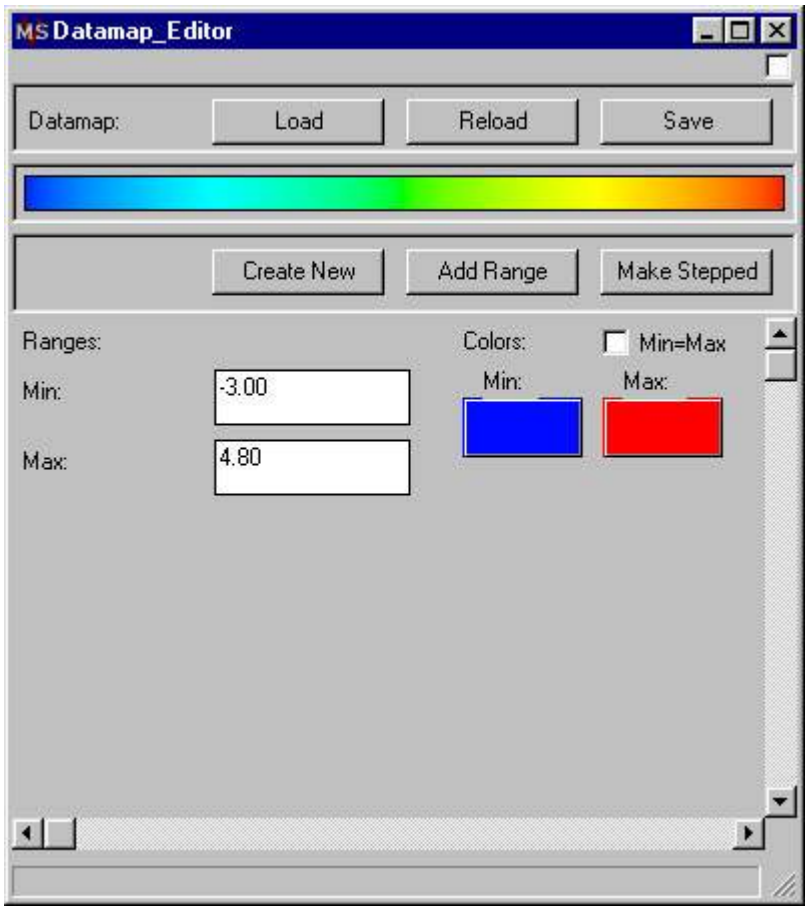
The first port (red) should be connected to the module whose rendered object you would like to color.

The second port (blue-red-blue-beige) is found on only a few modules in EVS/MVS. The modules with this port are contour_data, isolines, Datamap_Editor, and Legend.

These modules pass data corresponding to one or more subsetting levels (or solid contour levels). When this data is passed to Datamap_Editor, it is used to set the precise break points where datamap ranges (colors) will change. You still need to specify the colors for each of these ranges.

Module Output Ports

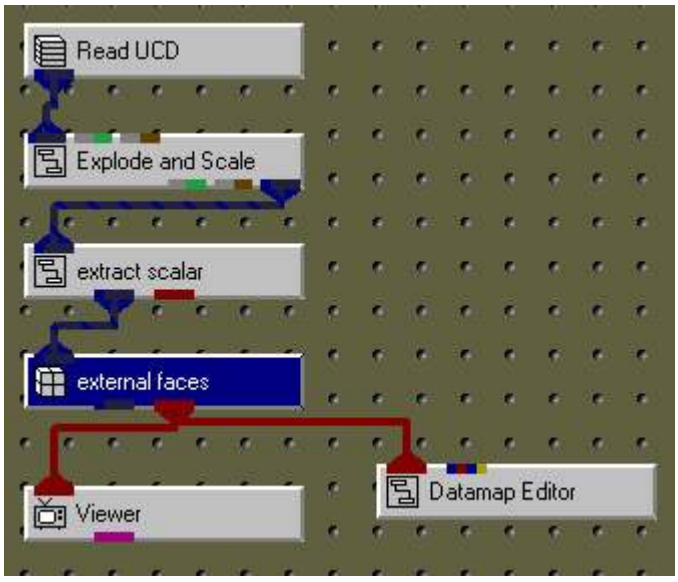
None



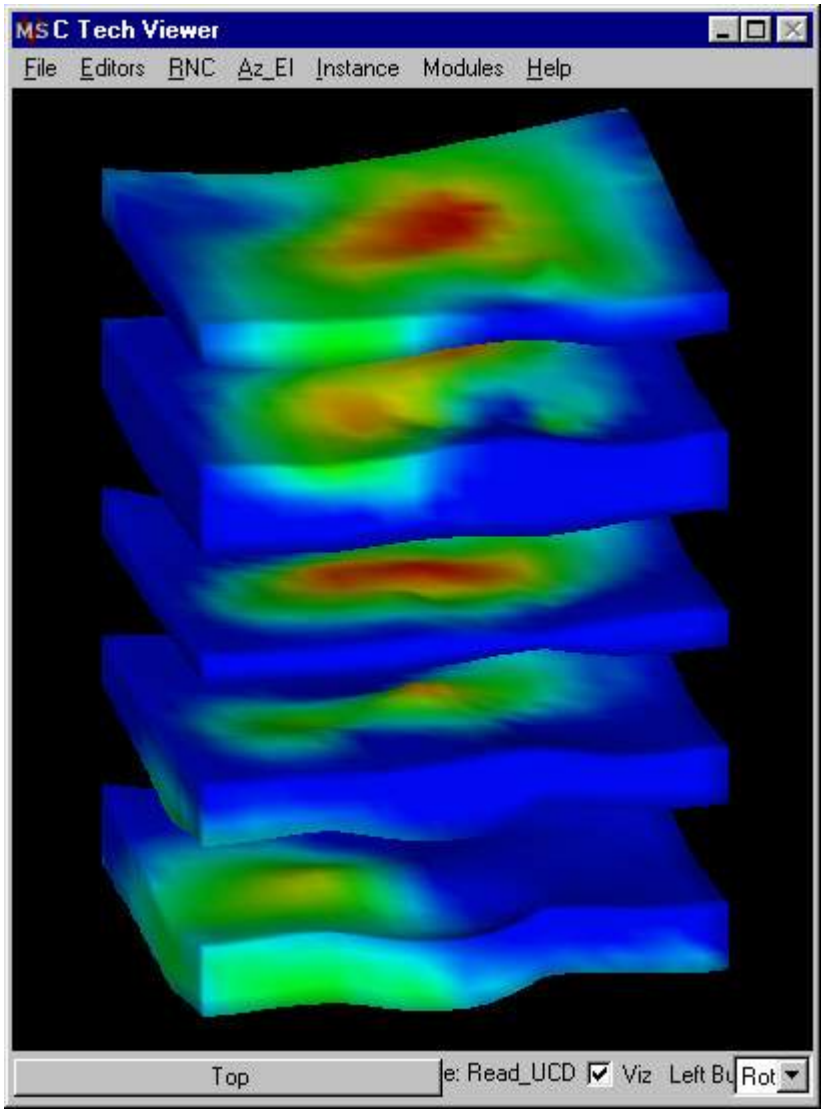
Module Control Panel

The Datamap_Editor control panel is shown above. Because of the numerous options associated with this module, a short tutorial follows.

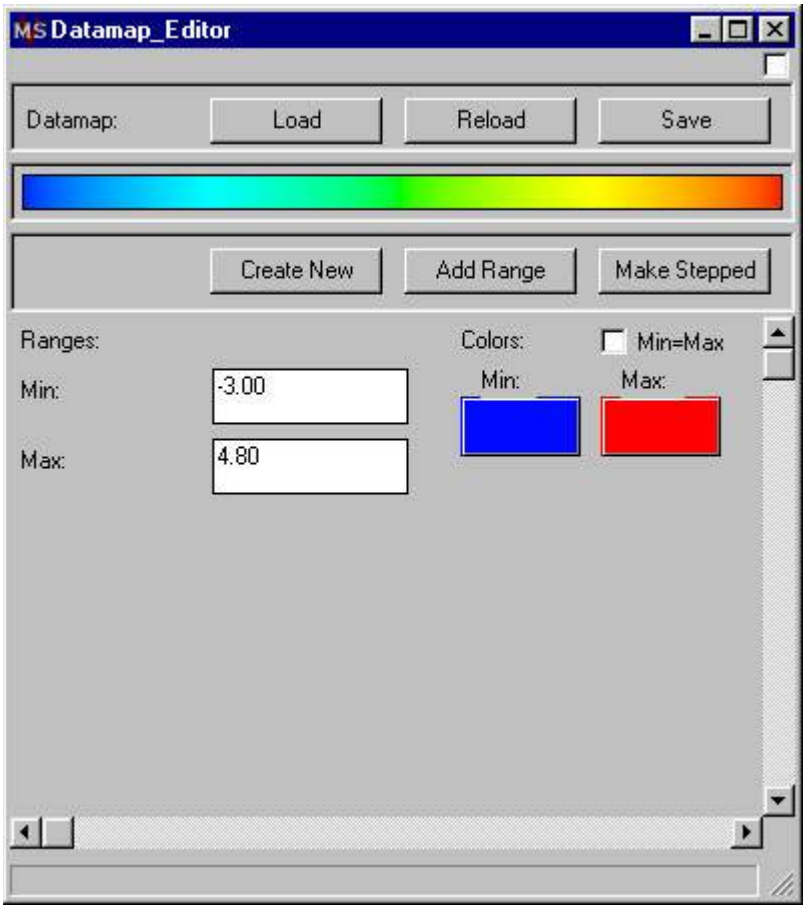
Let's begin by building a simple network. We'll read the UCD file initial_soil_investigation_subsite.inp for this example.



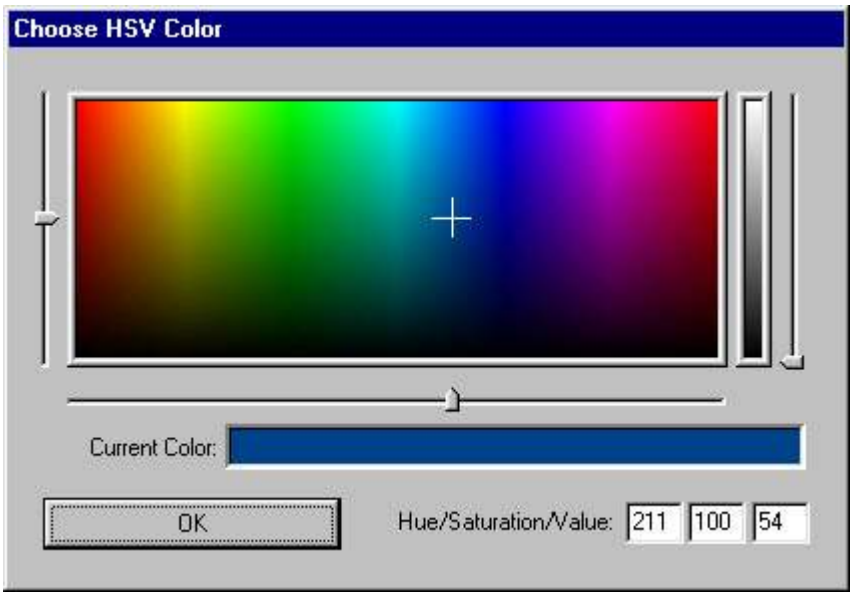
The picture in your viewer should match that below.



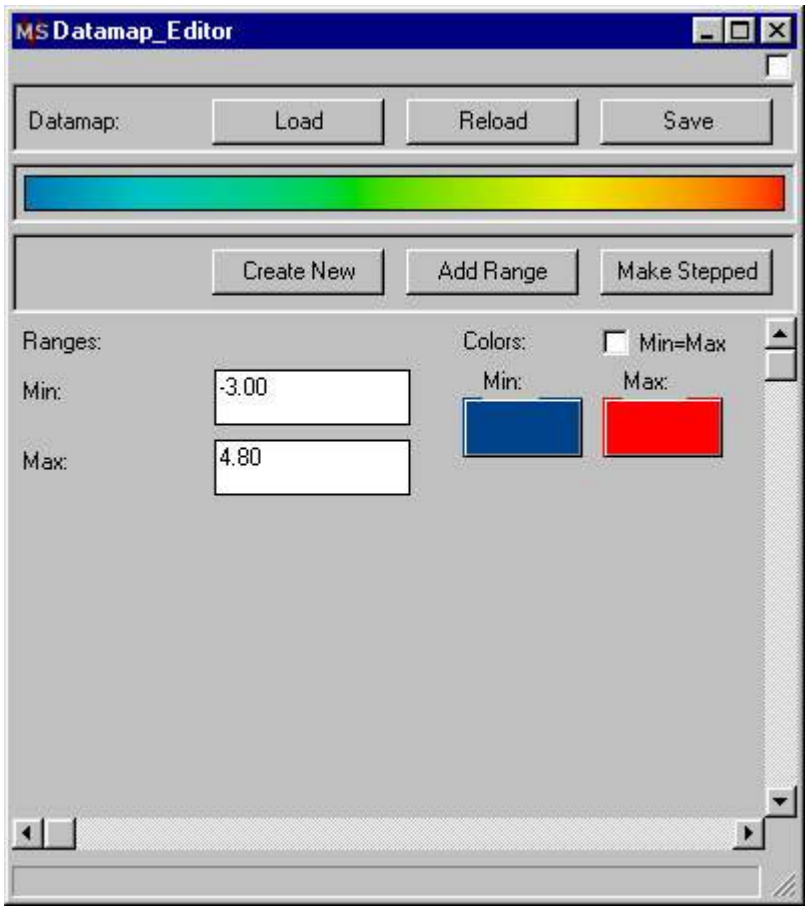
If you open the Datamap_Editor's panel, it will look like this:



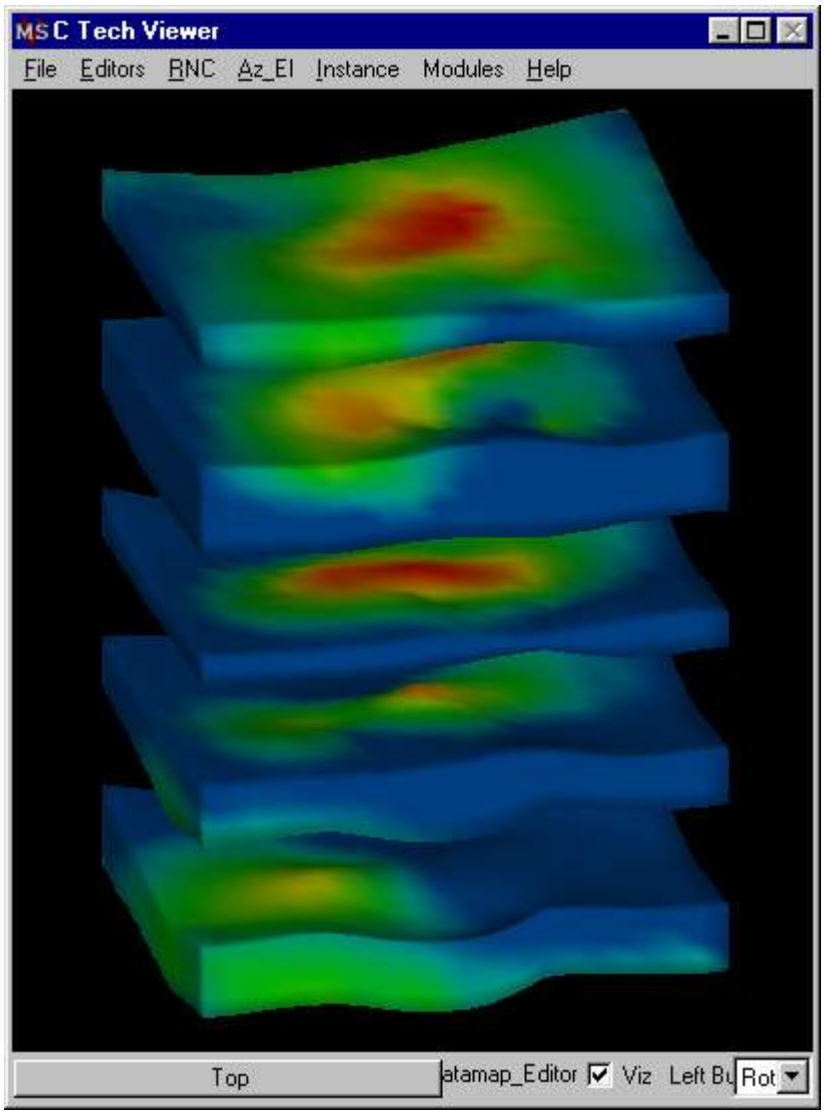
Press the blue button, which corresponds to the color that the data minimum maps to. It will pop up a color selector window. Pick a color similar to that shown below and hit OK.



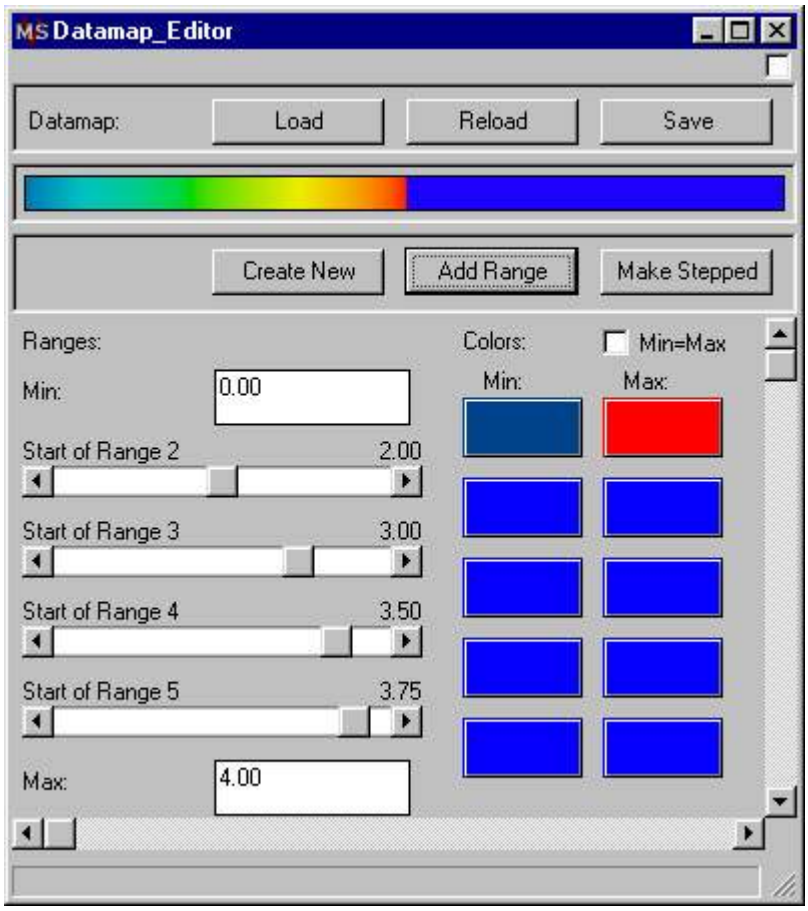
At this point, your datamap editor should reflect the changes that you have made to the min color and the resulting datamap.



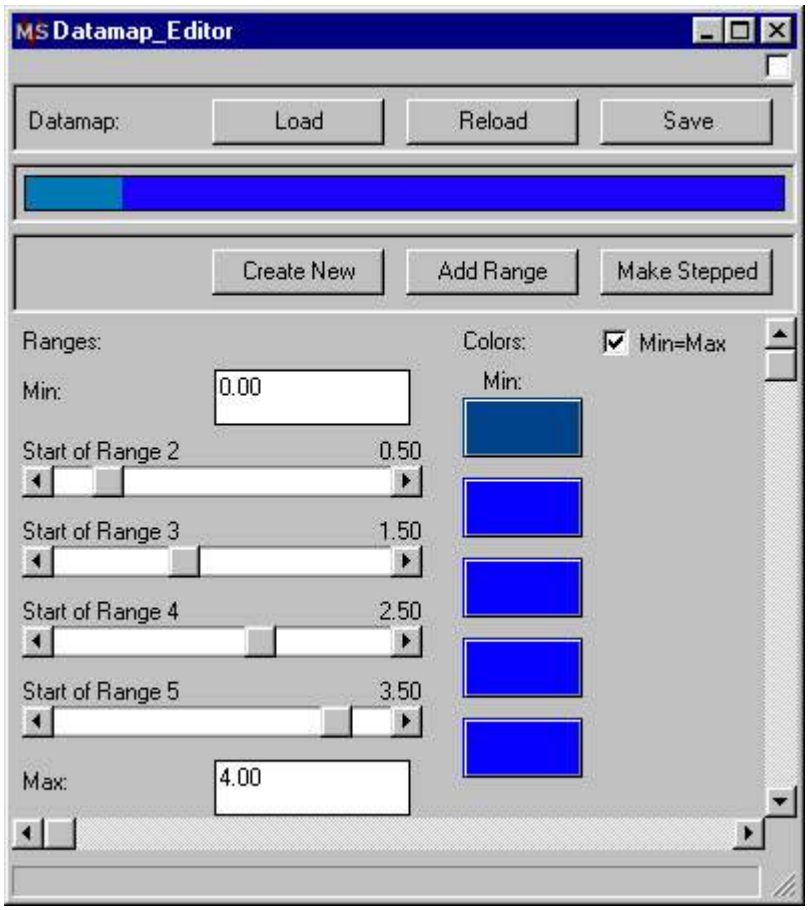
The picture in your viewer should match that below. This is the simplest use of the Datamap_Editor. You can change the color that the data Min or Max maps to.



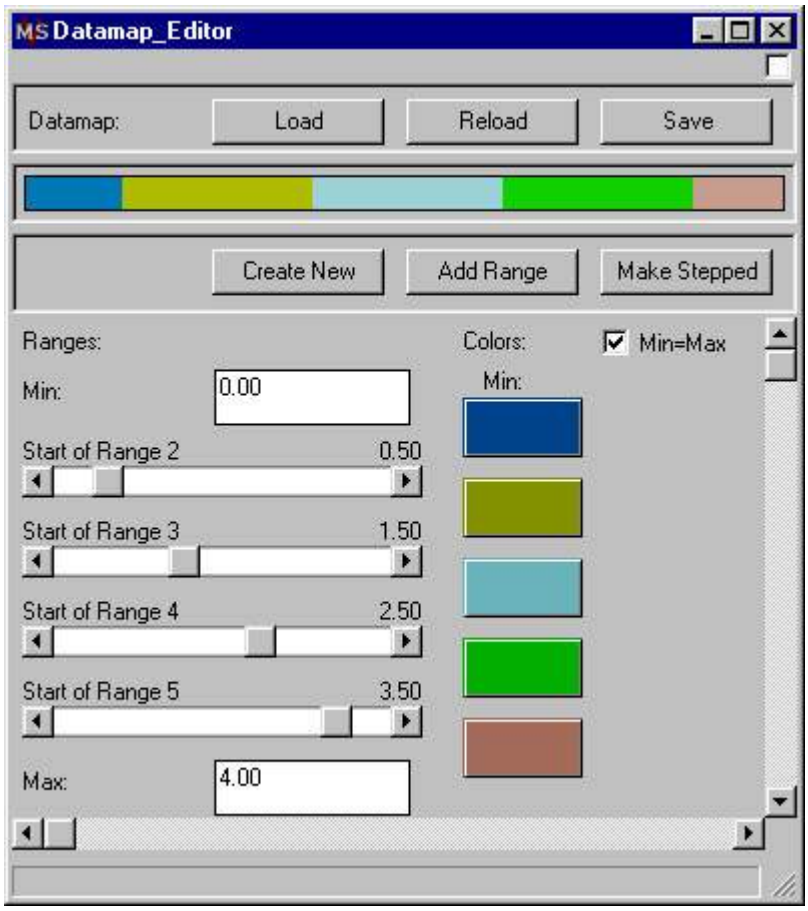
Now let's set extract_scalar's data component to Geo_Layer and then go back to the Datamap_Editor's panel and hit the Add_Range button 4 times. Datamap_Editor should now look like:



Hit the "Make Stepped" button and the panel will change to look like below. Make Stepped is specifically for data like geo_layer and material colors:

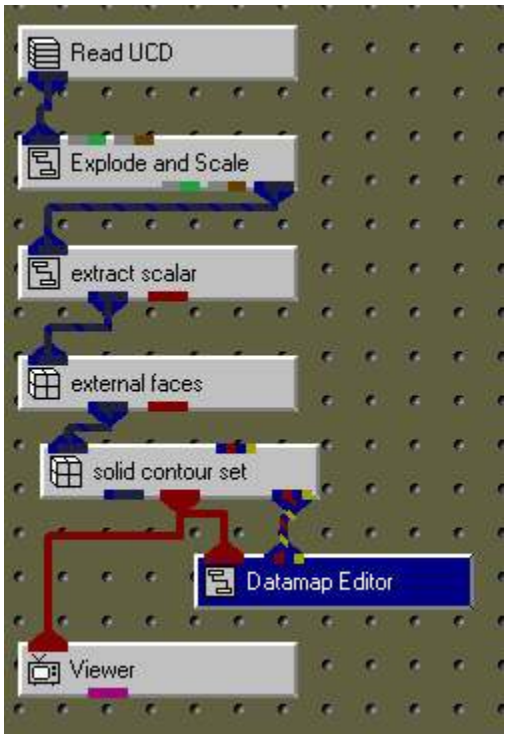


Select each of the four Blue buttons for ranges 2-5 (you can change the first one also if you want to) and change the color to the color you desire for each geologic layer. Your Datamap_Editor should now look something like:

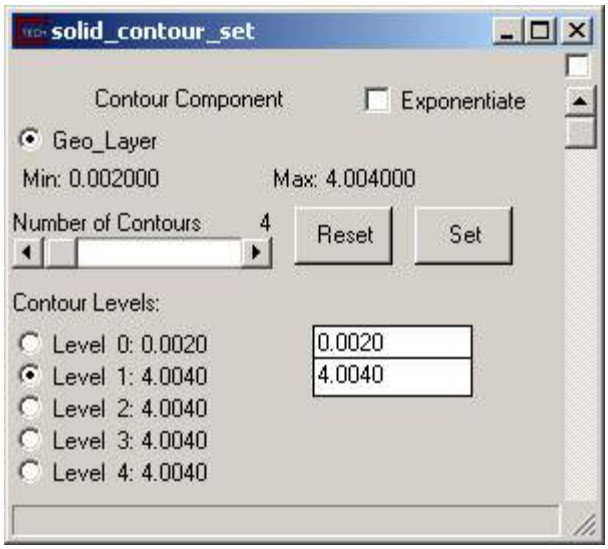


And the Viewer will now show layers with those colors.

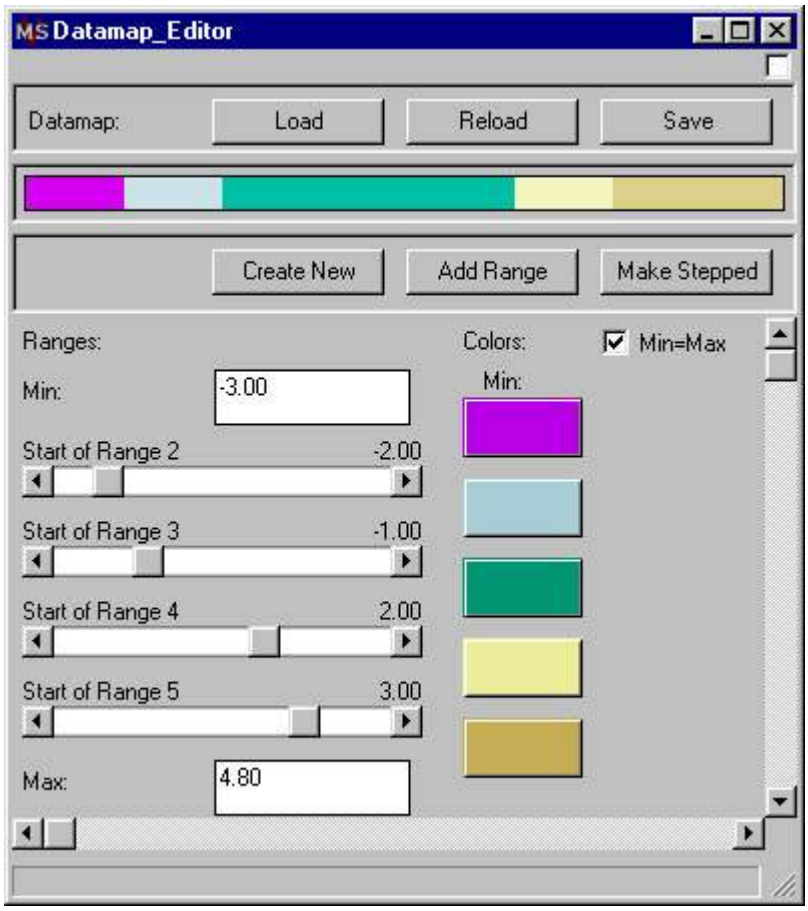
Now let's change the network to include the module solid_contour_set. This module (and a few others like isolines) has a special output port made to pass contour levels. This information can be used by the Datamap_Editor to set the break points in the datamap to exactly correspond to the levels. Datamap_Editor then allows you to set each color individually.



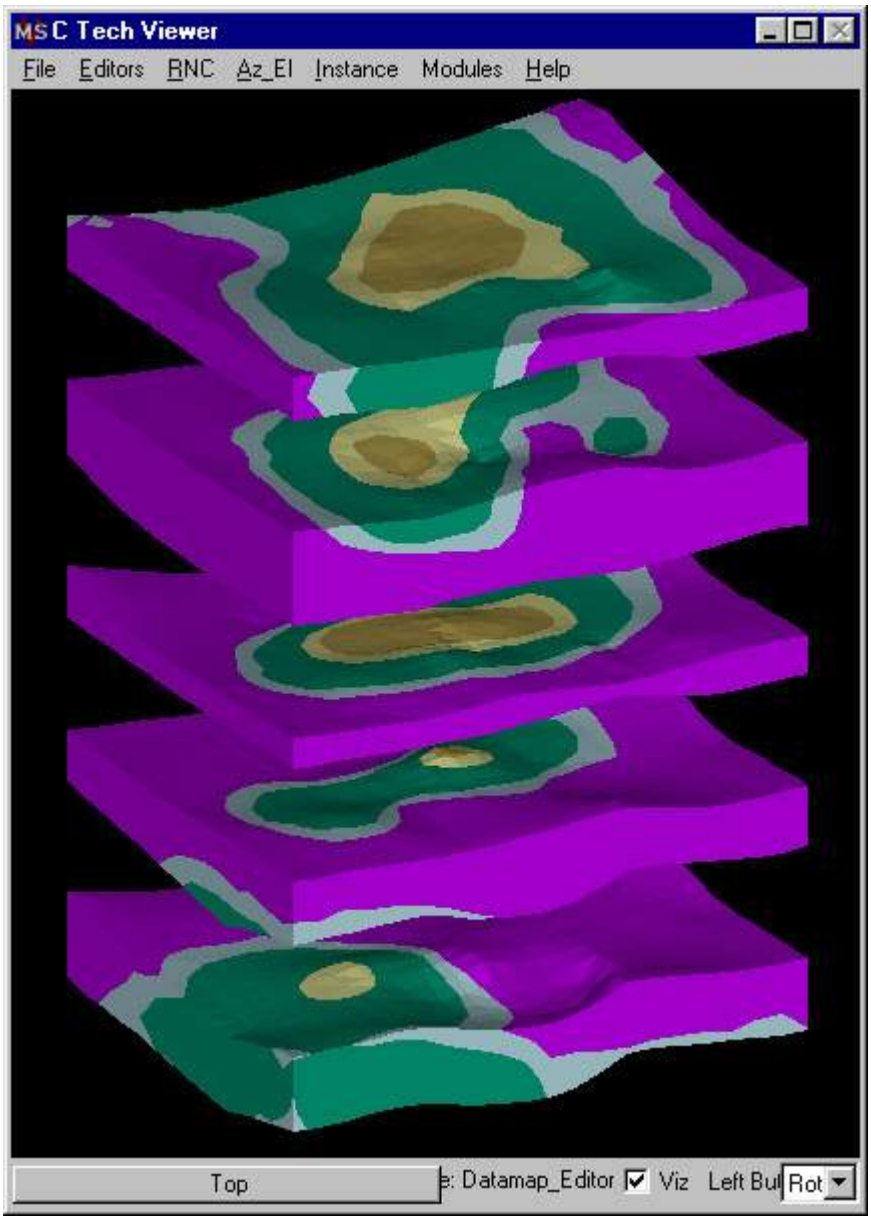
Set the levels and settings in solid_contour_set to match those shown below.



Open the Datamap_Editor panel and turn on the "Min=Max" toggle. The panel should have 5 identical blue buttons. Set the color for each range and your Datamap_Editor should now look something like this.



With the colors for each solid contour set, the Viewer should look like that below.



statistics

General Module Function

The statisitics module is used to analyze the statistical distribution of a single data field. The data field can contain a number od data components. Statistical analyses can only be performed on scalar data components. An error occurs if a statisitical anlaysis is performed on non-scalar data. Output from the statistics module appears in the EVS Status Window. Output consist of calculated min and max values, the mean and standard deviation of the data set, the distribution of the data set, and the coordinate extents of the model.

Module Input Ports

Statistics has two input ports.

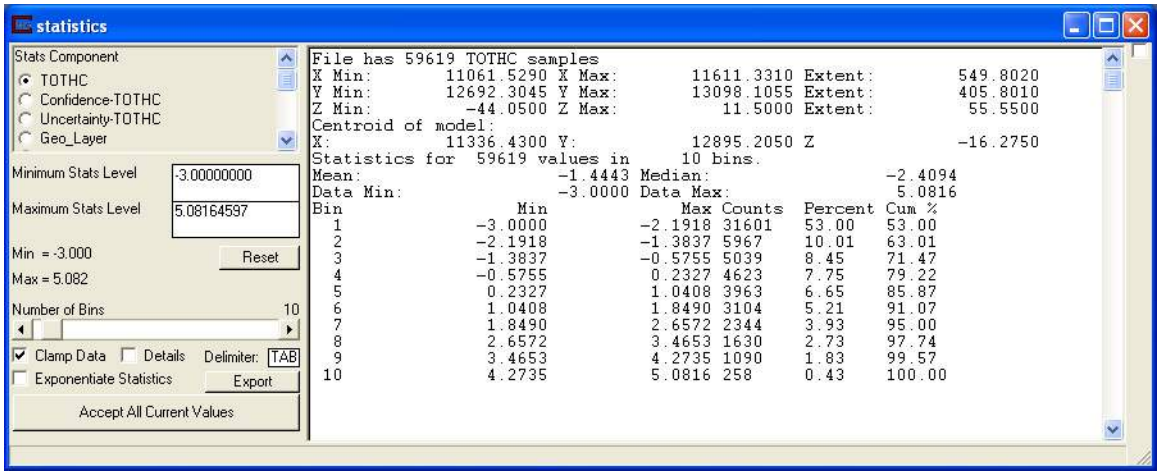
The first port (the leftmost one) should contain a mesh with nodal data. If no nodal data is present, statistics will only report the extents and centroid of your mesh. Data sent to the statistics module for analysis will reflect any data transformation or manipulation performed in the upstream modules. Any mesh data sent to the port is used for calculating the X, Y and Z coordinate ranges. The mesh coordinates have no affect on the data distribution. Cell based data is not used.

The second port is the z scale factor, which affects the centroid of the model.

Module Output Ports

By default, there are no output ports on the statistics module. However the Export button causes 6 output ports to be visible. These are (from left to right):

- mean
- median
- min
- max
- number of points
- The string (characters) representing the total output from Statistics.



Module Control Panel

The control panel for statistics is shown in the figure above.

1. Radio buttons are used to pick the *Stats* (data) *Component* on which the statistical anaylsis is to be performed. The list of radio buttons will display all avalilable data components passed to the input port. The first data component is the default. If the chosen data component is a vector, an error will appear in the EVS Status Window indicating this.

2. The *Minimum Stats Level and Maximum StatsLevel* edit boxes are used to set the upper and lower limits on the data bins for statistical analysis. The default values are the minimum and maximum values in the data component. If the statistical distribution should focus on only a portion of the data, these values can be changed to reflect only that desired range of data. The Min and Max values in the data component are shown below these edit fields.
3. The *Reset* button restores the Min/Max values to their default data extremes.
4. The *Number of Bins* slider is used to set the number of distribution bins to be used in the analysis. The default is 10 and the range is from 2 to 255. This value is generally changed to get unary or aesthetically pleasing bin values. For example, if the data min is -3.0 and the data max is 2.0, setting the Number of Bins to 6 will result in unary bin values (e.g. -3, -2, -1, etc.)
5. The *Clamp Data* toggle will actually clamp the data to the Minimum and Maximum Stats Level before processing it for statistics.
6. The *Details* toggle will display additional statistical information such as variance, deviation and quartile values.
7. The *Delimiter* type in controls the spacing of the output window. The text of the output window will be broken up according to whatever is in the Delimiter box, there is one special delimiter tag and that is the word "TAB" which allows the text to be tab delimited.
8. The Details toggle will display additional statistical information such as variance, deviation and quartile values.
9. The *Export* button causes 6 output ports to be visible. These are described above.
10. The *Accept All Current Values* button causes the module to re-run.

The Accept All Current Values button is used to tell statistics to calculate the distribution using the current settings. Results of the statistical analysis are sent to the EVS/MVS Status Window and the output panel on the right side of statistics' main panel.

Application_Notes

General Module Function

The Application_Notes module allows you to add textual notes to your applications that are saved with the application. This provides a means to document your work or provide instructions on the use or function of the application.

Suggested uses are documentation of:

1) unusual assumptions

2) required data files

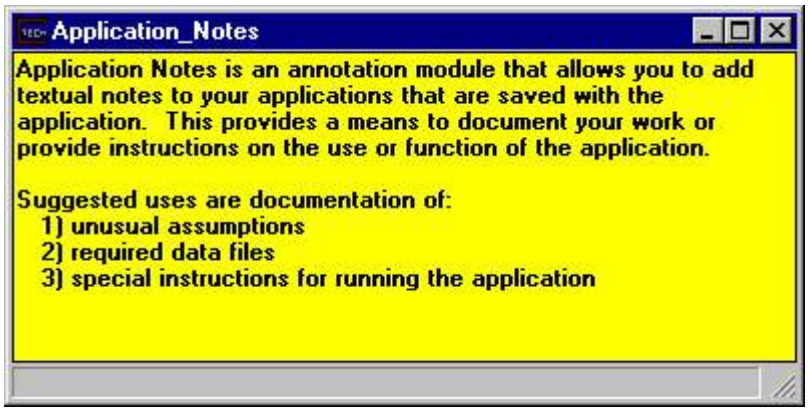
3) special instructions for running the application

Module Input Ports

No input ports.

Module Output Ports

No output ports.



Module Control Panel

The window allows you to type in a message of (virtually) any length. There are no scrollbars, but you can scroll up and down using the arrow keys or "Page Up" and "Page Down" keys on your keyboard.

Open_File

General Module Function

It completely replaces the functionality of the Edit_Data_File, Play_AVI_WAV, Open_Notes_File, Open_Excel_File, and Open_Word_File modules, which have now been removed from EVS.

This new module allows you to select any file on your system, and will automatically open it using your Windows file associations. You can force it to open up in Notepad (to make it work like Edit_Data_File or Edit_Notes_File), as well as request to use the "edit" shell context if available. If there is no editor associated with a file, it will open up the file in the default program, acting the same as a double click in Windows Explorer.

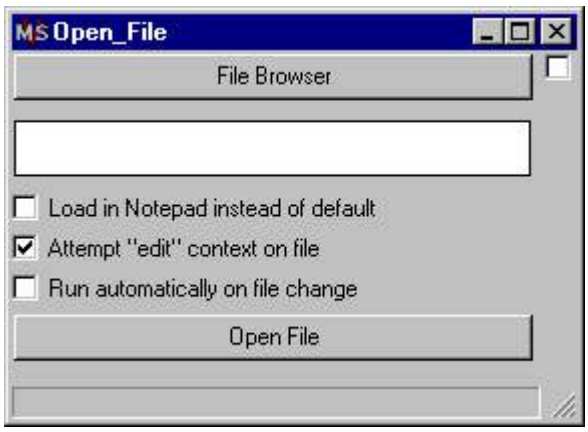
This allows you to launch your preferred application for playing animations, editing data, or anything else you wish, directly from within EVS.

Module Input Ports

It has an input port that accepts file names from all modules that read files (e.g. Krig_3D, Krig_2D, post_samples, etc.)..

Module Output Ports

None



Module Control Panel

The EDIT DATA FILE control panel is shown above. Clicking on the DATA File Browser push button opens a standard windows file browser. The filename and location can be specified in this browser. Clicking on the OPEN DATA FILE push button opens the data file.

volumetrics

General Module Function

The volumetrics module is used to calculate the volumes and masses of soil, and chemicals in soils and ground water, within a user specified isosurface (surface of constant concentration), and set of geologic layers. The user inputs the units for the nodal properties, model coordinates, and the type of processing that has been applied to the nodal data values, specifies the subsetting level and soil and chemical properties to be used in the calculation, and the module performs an integration of both the soil volumes and chemical masses that are within the specified isosurface. The results of the integration are displayed in the EVS Status Window window, and in the module output window.

The volumetrics module computes volumes and masses of analytes using the following method:

- A subset (plume) within the input grid is computed using the subsetting level specified in the volumetrics module
- Each cell having any nodes within the subset is analyzed
- The step above is limited to selected geologic units
- The portion of the cell above threshold is computed
- The mass of analyte within the cell is integrated based on concentrations at all nodes (and computed cell division points)
- The volumes and masses of all cells are summed
- Centers of mass and eigenvectors are computed

- For soil calculations the mass of analyte is directly computed from the computed mass of soil (e.g. mg/kg). This is affected by the soil density parameter.
- For groundwater calculations, the mass of analyte (Chemical Mass) is computed by first determining the volume of water in each cell. This uses the porosity parameter and each individual cell's volume. From the cell's water volume, the mass of analyte is directly computed (e.g. mg/liter).
- The volume of analyte (Chemical Volume) is computed from the Chemical Mass using the "Chem Density" parameter.

The volumetrics module will give more accurate results and should be used instead of the deprecated volume_integrate module.

Module Input Ports

The volumetrics module has seven input ports.

The first input port (the leftmost port) accepts a 3D data field.

The second input port is the explode distance of the model.

The third input port is the z exaggeration port.

The fourth port accepts a string as input; this string is then written to the output file if the Output Results File toggle has been selected.

The fifth input port takes a float value representing the subsetting level.

The sixth port takes a float value representing the date.

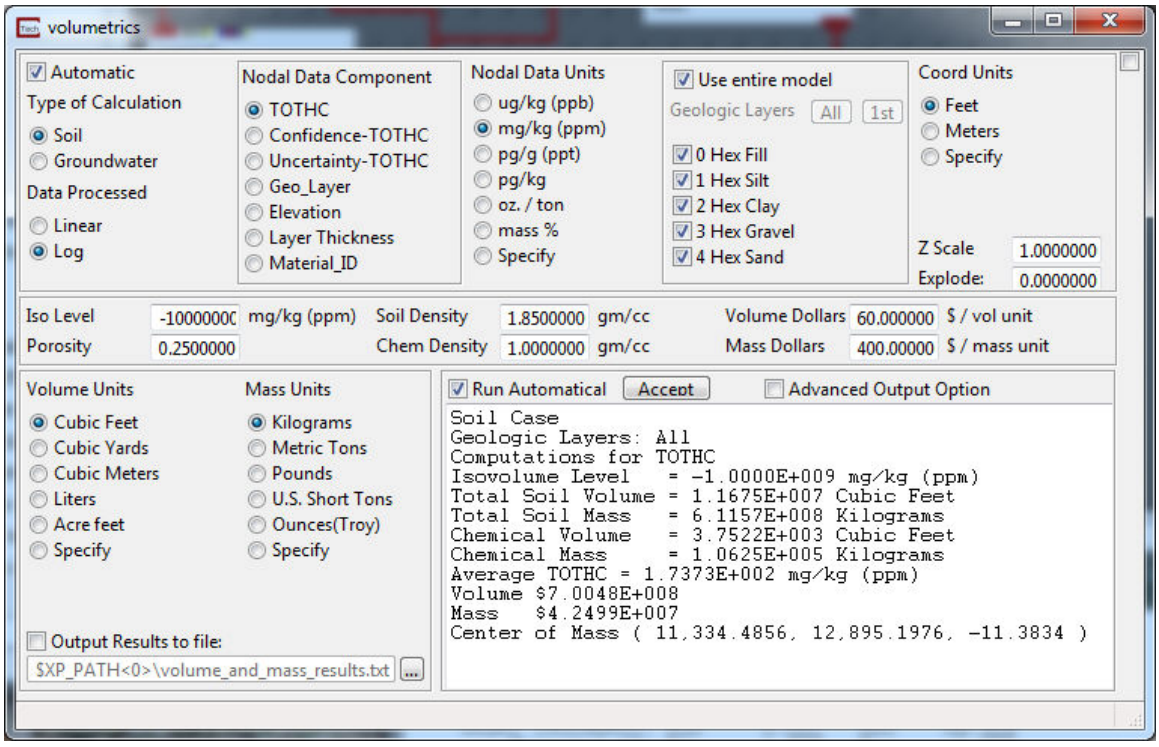
The seventh port takes an integer as its input; this integer causes the module to run when changed.

Module Output Ports

volumetrics has two output ports.

The first port (the leftmost) outputs a float representing the subsetting level.

The second port outputs the volume of the 3D field as a float.



Module Control Panel

The control panel for volumetrics is shown above.

The ***Automatic*** toggle will try to automatically determine the following options based upon the model data that is in the input field: Type of Calculation(Soil or Groundwater); Data processed, this is based upon the selected Node Data; Node Data Units, once again dependent on the selected Nodal Data Component; and Coordinate Units. To change any of these values the automatic toggle should be turned off.

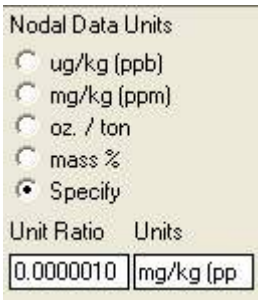
The ***Type of Calculation*** toggle changes the type of nodal data units that are available for selection.

The ***Data Processed*** radio list is used to specify whether the input data field has been processed to compute the Log 10 of the nodal values by Krig_3D or the module that is supplying the data to volumetrics. The default value is on, which means that Log 10 processing has been completed. Note that most analyte (e.g. chemistry) related data are log processed, while other types of data are not. If the user is not sure which processing has been completed, the Statistics module can be used to examine whether negative values that resemble log 10 data are present in the data set. When Log 10 processed data is passed to volumetrics, it exponentiates the nodal values in the data file, and completes the integration within the isosurface assuming that the data are exponentially distributed. This algorithm essentially returns the data to a normal distribution to provide accurate volumetrics estimates.

The ***Nodal Data Component*** lists the nodal data components that can be used to determine volumetrics.

The ***Nodal Data Units*** radio list allows the user to select the units of the Nodal Data Component. These options can change depending on the

selection of the Type of Calculation. The last option in the list is Specify, as shown below, this allows the user to specify their own units for calculation.



NOTE: The Unit Ratio is the ratio of the user specified units to the units we use for calculation. For example, the units we use for the Nodal Data Component in our calculations are grams/grams. If you wanted to use the units of mg/kg (ppm) you would need to find the ratio to convert the units which is in this case .001/1000 or .000001.

The ***Geologic Layers*** panel has a toggle which is dominant and other selectors:

- The**Use entire model**toggle insures that all layers will be visible. If any upstream module drops the layers and then recreates them, without this toggle on only the first layer of the model may be visible. You must turn this toggle off in order to use the layer check boxes to subset layers.
- Buttons to select "**All**" or only the "**1st**" layer quickly populate the checkboxes below. These options are only relevant if the *Use entire model*toggle is OFF.
- The layer selection list allows the user to select which of the geologic layers the volumes and masses will be calculated in. The layer number for which the estimates are made is printed in the EVS Console Window, along with the results.

NOTE: There are two different ways to perform volume calculations on individual geologic layers.

1) IF you have not used a plume_volume (or similar) module upstream of volumetrics you can use the ***Geologic Layers*** selection list allows you to choose the cell sets (geologic layers) that you want to perform computations on. Since geologic layers are segregated into "cell sets" this works only if there are no upstream modules LIKE plume_volume, which affects the cell sets.

If you use plume_volume upstream, it creates a single cell set for all hexahedron (hex) cells and another for all tetrahedron (tet) cells. This merges all geologic layers and makes it impossible to perform geologic layer subsetting INSIDE of volumetrics.

2) However, if you use the Explode_and_Scale or a select_cells module UPSTREAM of plume_volume you can pick one or more geologic layers that will be included in your volume calculations. In

this case, make sure all cell sets (hex and tet) are selected. In other words, perform your computations on everything.

When there are HEX and TET cell sets, you need BOTH. The hex cells are those cells in your grid that are completely within your threshold (subsetting level). These WHOLE cells are output from plume_volume unchanged EXCEPT that they are all lumped together in a single cell_set. Any cells that are not COMPLETELY inside the subsetting level are cut into smaller pieces that are completely inside. Those smaller pieces are composed of Tetrahedrons. All of the tetrahedrons are output in a single TET cell set.

Remember, you don't get the correct volume unless you include both Hex and Tet cell sets.

The ***Coord Units*** radio list is used to select the unit of measurement for the X, Y, and Z coordinates of the model data. The user should be certain of the nodal and coordinate units selected, as obviously, the outputs of the calculations are significantly affected by the units of the data. For the Specified Ratio the users input units need to be converted to Liters.

The ***Z Scale*** input field specifies the value of z exaggeration and is used to unscale the model for the purpose of calculations.

The ***Explode*** input field specifies the value the model has been exploded by and is used to unexplode the model for the purpose of calculations.

The ***subsetting level*** input field specifies the value in user units of the isosurface within which the volumetrics estimates will be completed. Note that the unit's text following the subsetting level input field will change to be consistent with the Nodal Data Units radio button selected, to remind the user what data units are being used.

The ***Soil Density*** and ***Porosity*** input fields allow the user to input the properties of the soil matrix in which the chemicals reside. Note that if the mass of chemicals in a combined soil and ground water plume are to be estimated, one of the geologic layers should be set up to have a boundary within it that corresponds to the water table position. In essence, this will create two layers out of one geologic unit that can be used to separate the soil domain from the ground water domain. The user can then choose the appropriate Nodal Data Units for each layer in the two domains, and obtain volumetrics estimates by summing the results in individual layers. There are several other alternative methods for completed volumetrics estimates in continuous soil and ground water plumes, which involve either setting up separate soil and ground water models, or using the Field Math module to remove and include specified areas of the domains.

The ***Chem Density*** input field allows the user to input the density of the chemical constituent for which mass estimates are being completed. Note that this value is used to calculate the volume of chemical in the specified isosurface, as the mass units are calculated directly from the nodal data.

The ***Volume Dollars*** type in is used along with the total volume of the chemical to indicate the cost of the removal of the chemical.

The ***Mass Dollars*** type in is used, along with the total chemical mass, to determine the value of the chemical mass.

Volume Units is used to select which units the volume should be calculated in. For the Specified Unit Ratio the units to convert to are liters. For example if your units were Cubic Meters the ratio would be 1000.

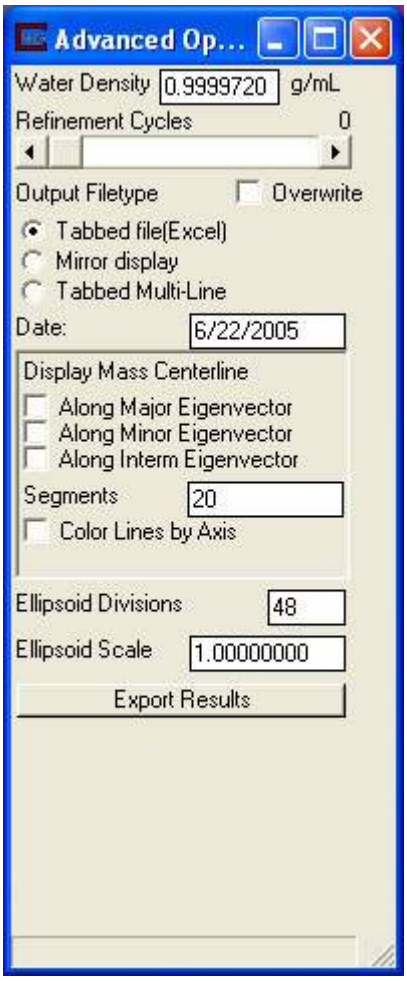
Mass Units is used to select which units the mass should be calculated in. For the Specified Unit Ratio the units to convert to are Kilograms.

The ***Output Results File*** toggle causes volume_and_mass to write a file to the ctech folder (volume_and_mass_results.txt) that contains all volumetrics information in a format suitable for input to programs like Excel (tab delimited .txt file). This file is written to in an append mode. It will grow in size as you use volumetrics. You should delete or move the file when you're done with it.

The ***Run Automatically*** toggle, when selected, causes the module to run as soon as any of the input parameters have changed. When not selected the accept button must be pushed for the module to run.

PRO and MVS only

For PRO and MVS users there is an advanced window that can be opened by checking the Advanced Output Options toggle.



The advance panel provides many capabilities including Spatial Moment Analysis.

- Spatial Moment Analysis involves computing the zeroth, first, and second moments of a plume to provide measures of the mass, location of the center of mass, and spread of the plume.
- The zeroth moment is a mass estimate for each sample event and COC. The estimated mass is used to evaluate the change in total mass of the plume over time.
- The first moment estimates the center of mass of the plume (as coordinates Xc , Yc, & Zc).
- The second moment indicates the spread of the contaminant about the center of mass (σ_{xx},σ_{yy} andσ_{zz}), or the distance of contamination from the center of mass. This is somewhat analogous to the standard deviation of the plume along three orthogonal axes represented as an ellipsoid created using the eigenvalues as the ellipsoid major and minor axes, and the eigenvectors to orient the ellipsoid. The orientation of the ellipsoid is aligned with the primary axis of the plume (not the coordinate axes).

- The Second Moment ellipsoid represents the spread of the plume in the x, y and z directions. Freyberg (1986) describes the second moment about the center of mass as the spatial covariance tensor.
- The components of the covariance tensor are indicative of the spreading of the contaminant plume about the center of mass. The values of σ_{xx}, σ_{yy} and σ_{zz} represent the axes of the covariance ellipsoid. The volumetrics module provides a scaling parameter that allows you to view the ellipsoid corresponding to the one-sigma (default) or higher sigma (higher confidence) representation of the contaminant spread.

The ***Water Density*** type in window allows the user to specify the density of water. The default of 0.9999720 g/mL is the Density of Water at 4.5 degrees Celsius.

The ***Refinement Cycles*** slider specifies how finely each element will be subdivided to perform the integration. The default value is 0, which provides the fastest computation time, and resulting volumetrics estimates that are a good first approximation. Generally, the number of refinement cycles should be set to a value between 1 and 4, and the user should experiment with increasing the refinement cycles until subsequent calculations show a change that is insignificant to the problem at hand.

The ***Output Filetype*** radio list is used to select the format of the output file. The default is a tab spaced single line output, the second choice will format the output the same as the display window, and the third option will format the output separated by tabs on multiple lines. Changing these options will not cause the module to run, you must hit accept or change an input value for the module to run.

Overwrite causes the output file to be overwritten instead of appended to. This toggle will only be selected for one run and then will unselect itself and begin appending again, unless it is rechecked. Selecting this toggle will not cause the module to run, you must hit accept or change an input value for the module to run.

The ***Date*** type in allows you to set the date, which is output only in the Tabbed Multi-Line file.

The ***Display Mass Centerline*** allows you turn on and off the lines lying along the Major, Minor, and Intermediate Eigenvectors. These vectors represent the second moment of mass, and by default have chemical data mapped to them. These lines are of the same orientation as the second moment ellipse but they stretch only to the extents of the model. To output these lines the Export Results button must be pushed.

The ***Segments*** type in allows you to control the number of segments making up each line, the larger the number of segments the closer the node data along the line will match the node data of the model, but at the cost of speed.

The ***Color Lines by Axis*** toggle strips the node data from the lines leaving them colored by the axis the represent.

EllipsoidDivisions is an integer value determines the number of faces used to approximate the analytically smooth ellipsoid. The higher the resolution the smoother the ellipsoid. The *Export Results* button must be used to have an output port for the ellipsoid.

EllipsoidScale is a scaling factor for the second moment ellipsoid. A value of 1.0 (default) is analogous to one-sigma (67%) statistical confidence. Higher values would provide an indication of the size of the eigenvalues with a higher statistical confidence.

The ***Export Results*** button adds an additional seven output ports to the original two for a total of nine output ports.

The first port (the leftmost one) exports a float representing the subsetting level.

The second port exports a float representing the Soil Volume.

The third port exports a float representing the Soil Mass.

The fourth port exports a float representing the Chemical Volume.

The fifth port exports a float representing the Chemical Mass.

The sixth port exports a field representing the Ellipse of the second moment of mass.

The seventh port exports a renderable object representing the Ellipse of the second moment of mass.

The eight port exports a field representing the Mass Centerlines of the second moment of mass.

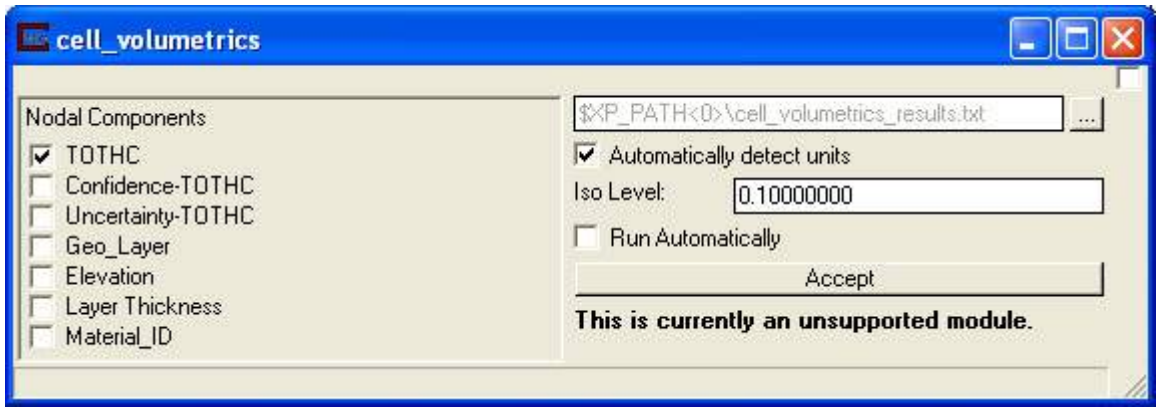
The ninth port exports a renderable object representing the Mass Centerlines of the second moment of mass.

cell_volumetrics

(Available only in MVS)

General Module Function

The cell_volumetrics module is currently an unsupported module.



Module Control Panel

The control panel for cell_volumetrics is shown above.

area_integrate

General Module Function

The area_integrate module is used to calculate the areas of the entire field input. The input data to area_integrate must be a two dimensional data field output from Krig_2D, slice, or any subsetting module which outputs two-dimensional data (slice, plume_volume with 2D input, or plume_shell). The results of the integration are updated each time the input changes.

Module Input Ports

The area_integrate module has only one input port, which accepts a 2D data field from Geologic_Surface, slice, or any of the modules that output 3D fields.

Module Output Ports

area_integrate has one output port that exports the area in your user units squared.

has no output ports, as its output is displayed as text in the control panel.

Module Control Panel

The control panel for area_integrate is shown above. Note that if your input field has specified units (like feet) the results will show that.

tri_tool **(This module is available only in MVS)**

tri_tool is an MVS module primarily for use with surf_cut. It can subdivide triangular and quadrilateral cells until none of the sides of the output triangles exceed a user specified length (a default value is calculated as 5% of the x-y extent of your input surface). This increases the accuracy of surf_cut especially when the input surface comes from scat_to_tin and the nodes used to create the TIN are poorly spaced. It can also correct the normals of a surface. It does this by organizing all of the triangles and quadrilaterals in a surface into disjoint patches, and then allowing the user to select which patches have normals that need to be flipped. The maximum number of triangles in a patch is 130,000, any triangles above this number will be considered to be in the next patch.

Module Input Ports

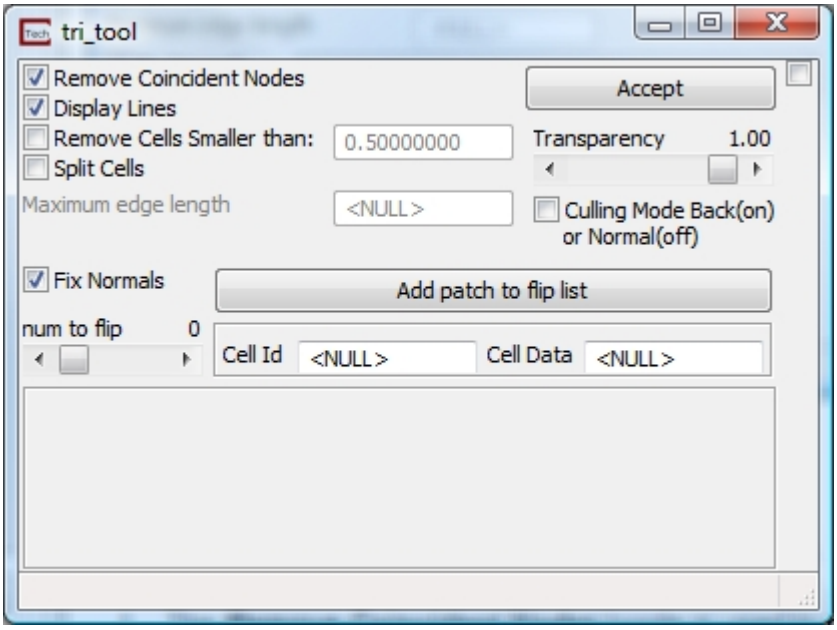
tri_tool has one input port which accepts a surface composed of triangular and/or quadrilateral cells.

Module Output Ports

tri_tool has two output ports.

The first port outputs a surface consisting of the newly created or flipped triangles.

The second port outputs a renderable surface to a Viewer. This surface consists of triangles that are colored according to the disjoint patch they belong to.



Module Control Panel

The control panel for tri_tool is shown in the figure above.

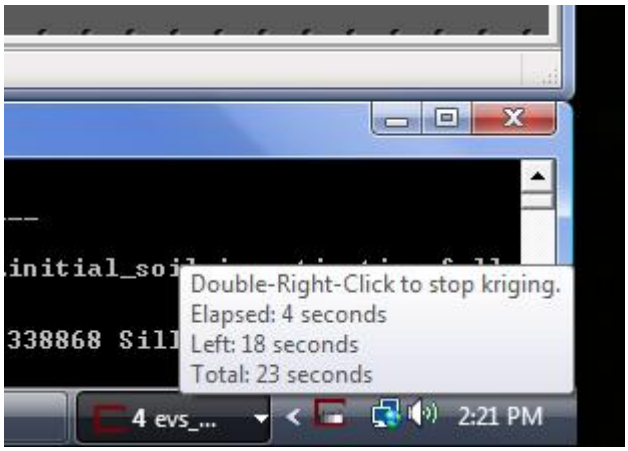
1. The ***Remove Coincident Nodes*** toggle is used to merge nodes that have multiple nodes occupying the same coordinate space.
2. The ***Split Cells*** toggle is used to ensure that none of the triangles have an edge longer than the maximum edge length.
3. ***Display Lines*** toggles the line rendering mode between Regular and None. This displays the connectivity between nodes for each cell.
4. The ***RemoveCells Smaller than*** toggle is used to remove extremely small cells (based on area in your coordinate units squared) that sometimes are generated with CAD triangulation routines that might have their normal vectors reversed and would contribute to poor cutting surface definition. Try this option if you find that surf_cut is giving anomalous resuts.
5. The ***Maximum edge length*** type in allows the maximum length of each triangle side to be set for when the Split Cells toggle is set.
6. The ***Fix Normals*** toggle is used to check to make sure all of the triangles in selected patches of the surface have the same normal. If the normal is backwards, you can flip the normal of the patch in two ways. The first way is Alt + Right click on a cell in the patch that you

wish to flip and then click the Add patch to flip list button. You only need to do this for one cell in each patch. Another way to do this is to increase the num to flip slider by one and enter the Cell Id and Cell Data value of a cell in the patch you wish to flip. The Cell Id and Cell Data values should be gotten from the surface being output from tri_tool, and not the surface being input.

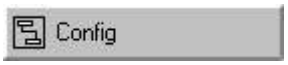
7. The ***Culling Mode*** toggle controls whether back facing surface are visible. Generally you will want this ON when making the object(s) transparent.
8. The ***Transparency*** slider determines the opacity of the objects.

Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Config



General Module Function

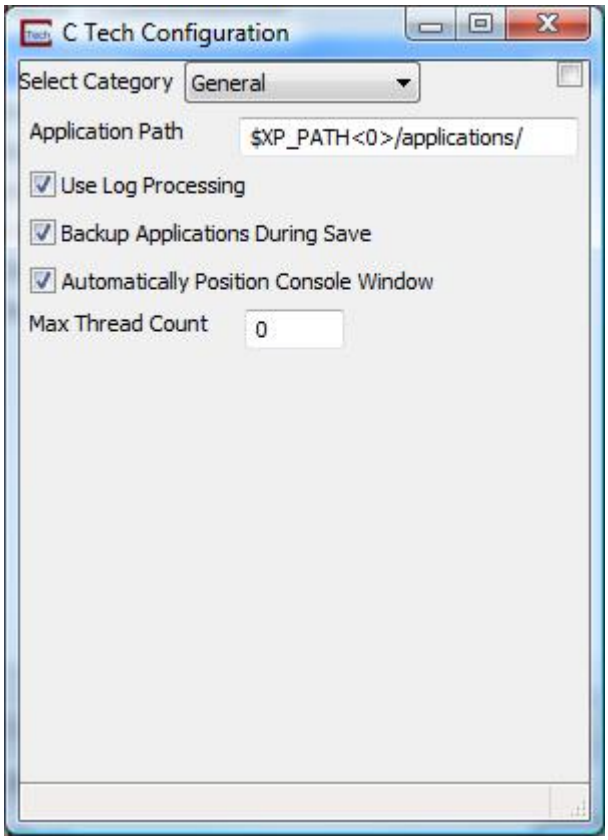
The Config module provides a means for setting certain EVS/MVS modules and file paths to user defined default values. This may be useful for certain projects or to customize defaults for different users within a group. The module currently provides the ability to change defaults for 11 modules and allows setting customized data path defaults for most EVS/MVS data types.

Module Input Ports

No input ports.

Module Output Ports

No output ports.

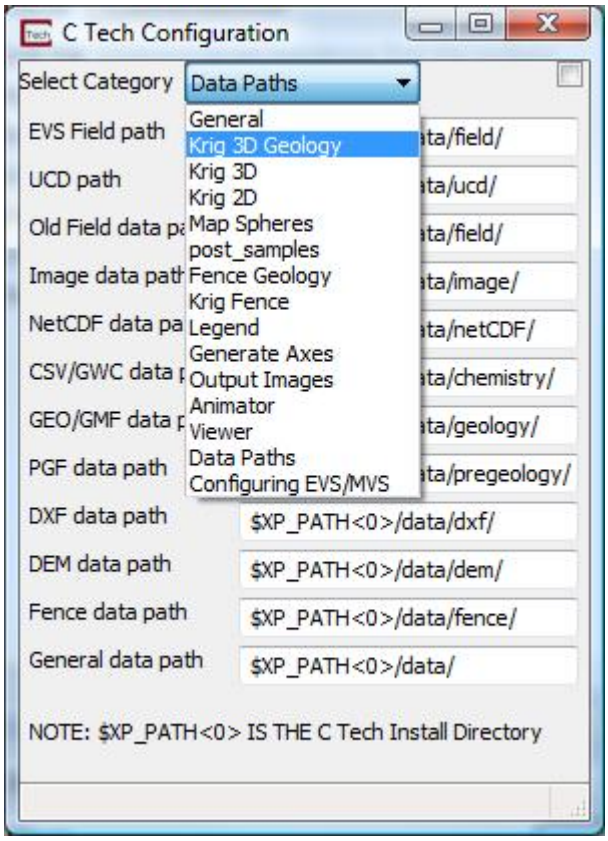


Module Control Panel

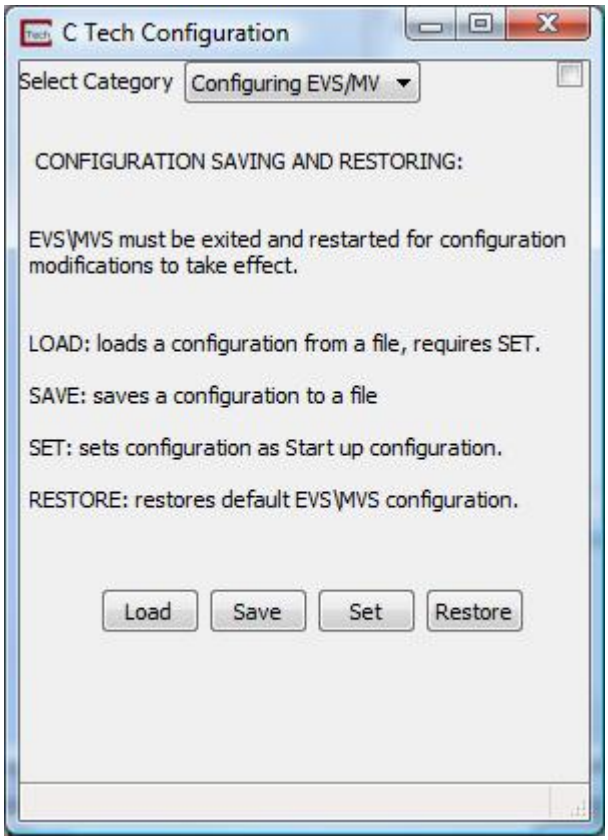
This module has a control panel which initially provides a pull-down list whereby each list selection activates type-ins for the available default settings. The final list selection, entitled Configuring EVS/MVS provides the necessary options for loading, saving and setting a previously defined configuration. The modules currently available for configuring defaults are:

- Krig_3D_Geology
- Krig_3D
- Krig_2D
- Post_samples
- Fence_Geology
- Krig_Fence
- Legend
- axes
- Animator
- Output_Images
- Viewer

The Data Paths selection allows setting of new paths for most EVS data types.



The Configure EVS/MVS panel allows the user to Load, Save, Set and Restore a configuration. The recommended approach for creating a new configuration of is to make the desired adjustments, save the configuration to a file, load the configuration, set the configuration, then exit/restart EVS/MVS. For making a configuration adjustment you do not wish to use again, then simply use Set and then exit/restart EVS/MVS. To use the provided EVS/MVS defaults select Restore, then exit/restart EVS/MVS. Note that for any modifications to take effect you must exit and restart EVS/MVS.



merge_fields

(This module is available only in MVS)

General Module Function

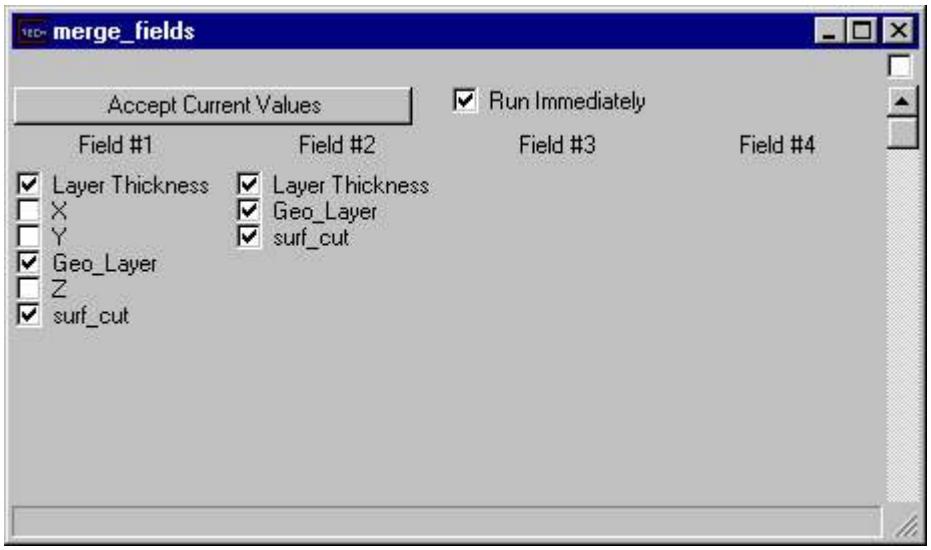
merge_fields combines the input fields from up to 4 separate inputs into a unified single field with any number of nodal data components, which can be output to virtually any filter or mapper module, OR directly to the Viewer. This is useful when you want to slice through or otherwise subset multiple fields using the same criteria or object.

Module Input Ports

merge_fields has four identical input ports. Each port can accept any type of mesh with nodal data. At least two input ports must be used and up to four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right. The nodal data components must be in the same order (not all must be selected) and must correspond to the same criteria in order for the output to make sense.

Module Output Ports

merge_fields has two output ports. The first output port (closest to the left) outputs a mesh containing the merged input fields. The second port outputs a renderable object.



Module Control Panel

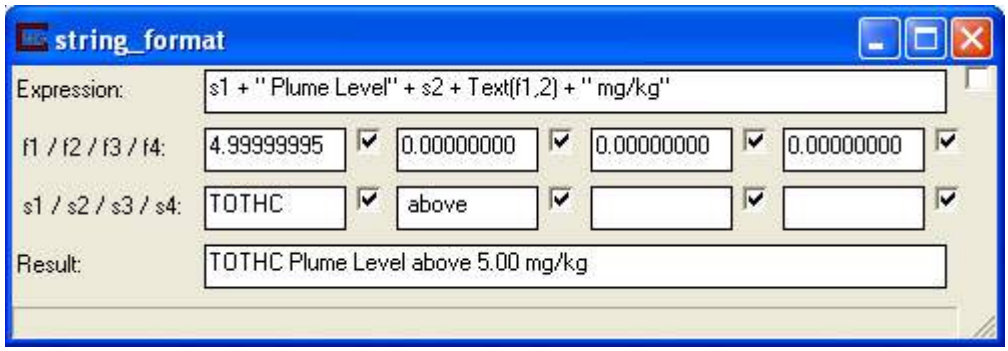
The control panel for merge_fields is shown in the figure above. For each field input, you must select the data components you want in the output. Be sure that the order of components is identical in each list. If the "Run Immediately" toggle is on, as soon as the number of components for each field is the same, the module will run automatically. Otherwise, merge_fields will run when you push accept.

Related Modules

->combine_comp

string_format

string_format, used in conjunction with modules like plume_shell, plume_volume, loop and others provides a very simple way to create complex titles that change as the parameters in your module change. It also provides a simple means to perform mathematical operations on numbers coming from up to 4 numeric input ports. By using multiple string_format modules, a title (string) of any complexity may be created.



The panel for string_format is shown above. The expression shown

s1 + " Plume Level" + s2 + Text(f1,2) + " mg/kg"

1) with a value of 5.0 (actually 4.99999995) input to the leftmost numeric input port and

2) the iso_component input to s1 (string 1) and

3) the above/below string input to s2, results in the following string output:

TOTHC Plume Level above 5.00 mg/kg

The most wonderful thing about this result is that changing any of the parameters in the plume_volume module that which is connected to 3 inputs on string_format automatically update the result. Even in an animation, changes that affect plume_volume are automatically reflected in the output of string_format. This behavior can be changed by unchecking the checkbox next to the desired field. When this box is unchecked the output will not change when that value changes until it is checked again. However if the value changes in any other field that is checked the module will run will all of the current values.

In addition to the functions, variables, and operations available in the other math operations, the string_format module can also use the following functions on string variables and values:

String(number,[decimals],[format]) or **Text(number,[decimals],[format])**:

This function takes one to three arguments.

The first is a number. This number is converted into a string for display purposes.

An optional second argument can be used, which is the number of decimal points to use during the conversion.

If this number is negative, the function attempts to automatically determine the optimal number of decimal points to use based on the value. By changing this negative value, you can control the number of decimal points used in the automatic conversion.

The default setting if unspecified is a value of -2.

A third optional argument is a formatting specifier. **Note that the format specifier must be in single or double quotes 'r;e' or "e".** The following values can be used:

'n' (default): Number format: The value is converted to a string of the form "-d,ddd,ddd.ddd..." The symbols used for thousands and decimal separators are determined in the Regional Settings of the Windows Control Panel.

'f': Fixed format: The value is converted to a string of the form "-ddddddd.ddd"

'e': Scientific Notation: The value is converted to a string of the form "-d.ddd...E+ddd". The resulting string starts with a minus sign if the number is negative. One digit always precedes the decimal point.

'g': General format: The value is converted to the shortest possible decimal string using fixed or scientific format.

'm': Money format: The value is converted to a string that represents a currency amount. The format is controlled by the Regional Settings in the

Windows Control Panel, and will include the currency symbol, thousands separator, and decimal separator as specified in the Regional Settings.

Examples: (quotes " " are shown, but are not a part of the output)

- String(5.5) = "5.5"
- String(5.5,3) = "5.500"
- String(1000000,1) = "1,000,000.0"
- Text(5423.1,2,'f') = "5423.10"
- Text(5423.1,2,'m') = "$5,423.10"

String Operators: Many functions (operators) are available to parse and assess the characteristics of strings. These are discussed here:
Pop-up available String Operatorshere or **Jump to availableString Operatorshere**.

Complex Example: If you connect a filename port to s1 after browsing for a file (browsing always gives you \ slashes, doubled in the saved app), the following works:

Mid(s1,1+RPos(s1,"\"))

This will strip off just the filename portion. (ie: If you do c:\ctech\data\analyte\initial_soil_investigation_subsite.apdv, the above returns initial_soil_investigation_subsite.apdv)

The 1+ is because we want the character after the "\".

Date(date_number, format)
The date function allows you to format a wide variety of date and time strings (labels). **Note that the format specifier must be in single or double quotes 'r;c' or "c".**
Pop-up available Date Formats here or **Jump to available date formats here**.
To create the current date/time use the following:

- date(now()) = 10/16/2009 8:59:14 AM
- date(now(),"MMMM dd, YYYY") = October 16, 2009

Season(date_number)
Season is a special date operator that outputs either *Winter, Spring, Summer or Fall,* based on the date_number input. To display Summer, 2002 you would use Season in conjunction with Date (displaying only the year).

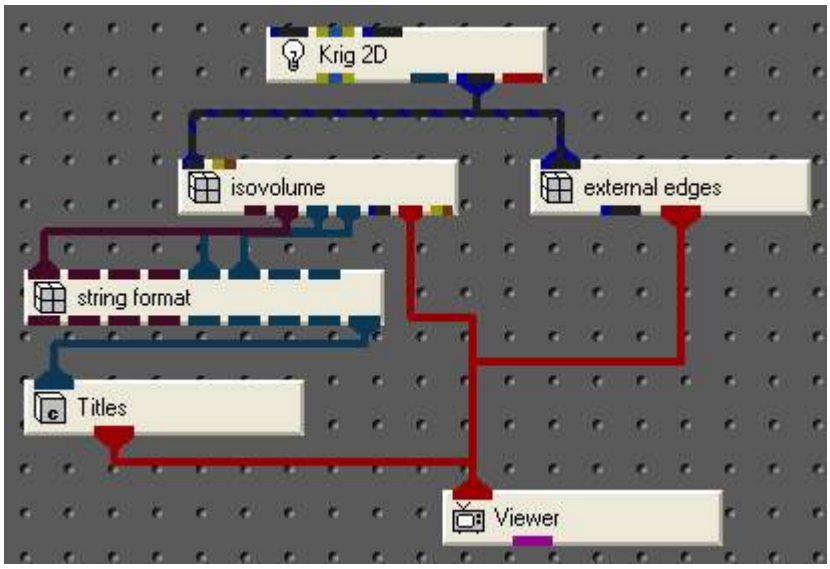
Now()

Returns the date_number referring to the current date and time.

String_format can also have complex math operations in the expression.

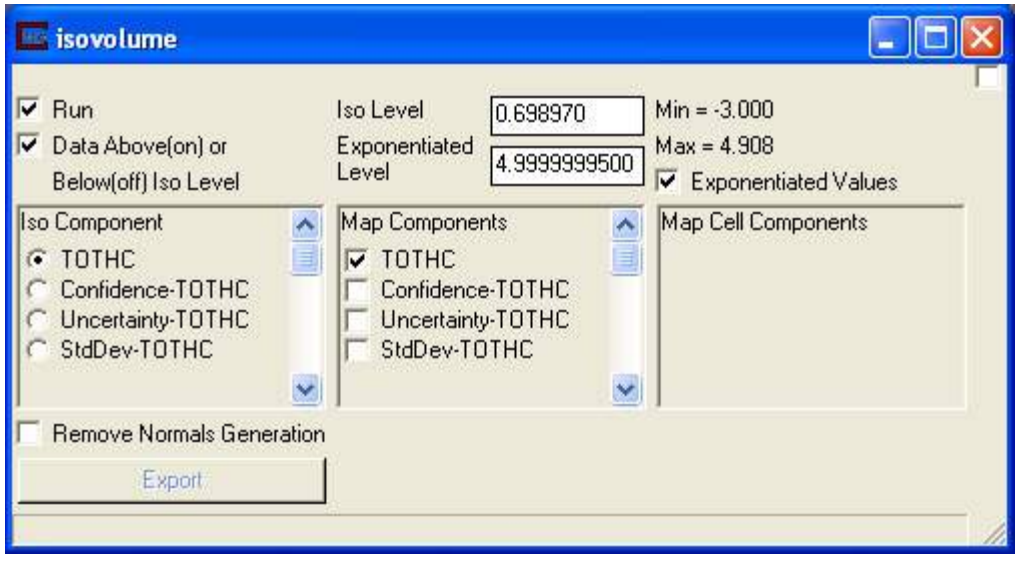
Mathematical Operators here

Any of these operators may be used.

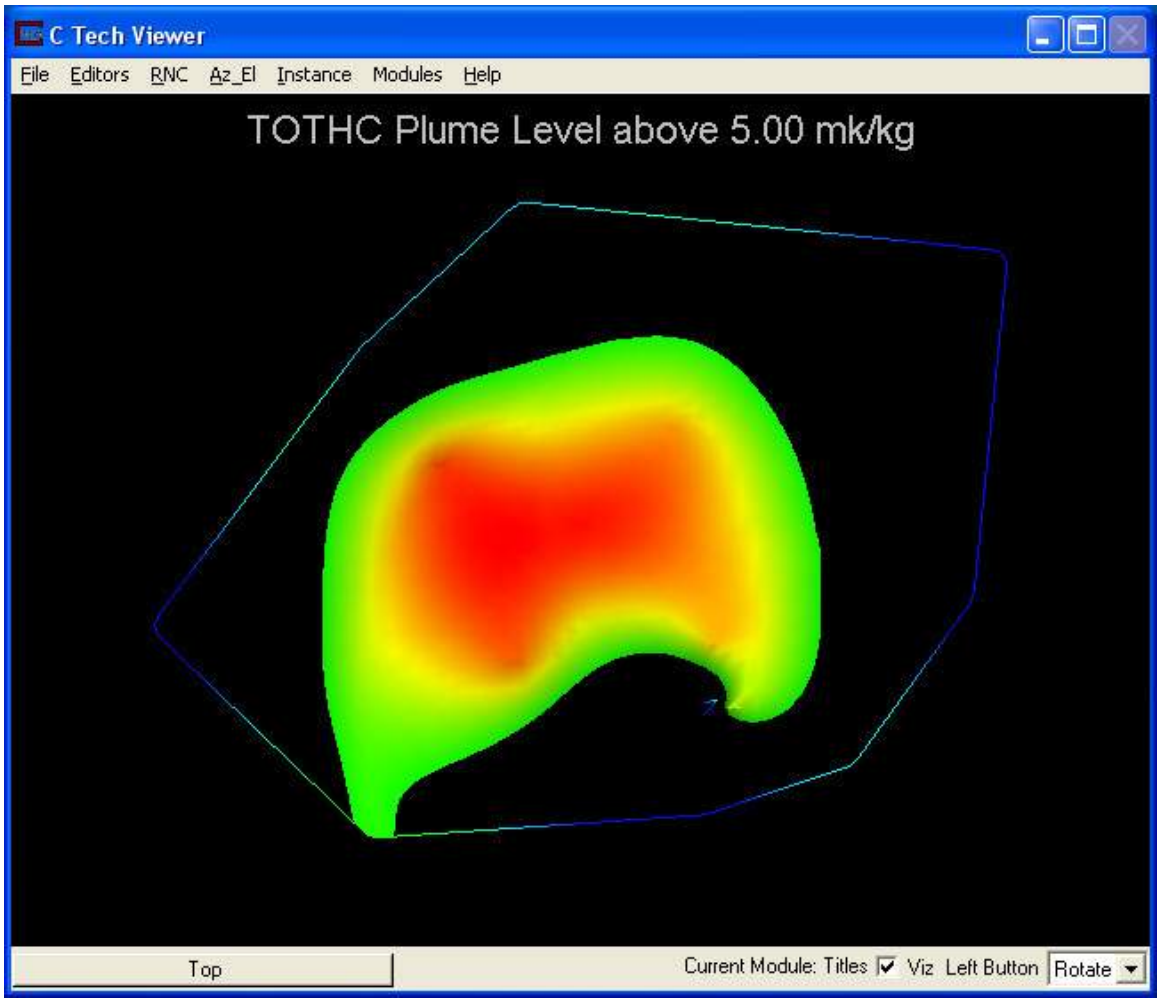


An example of an application using this module is shown above.

The parameters in plume_volume are shown below.



The output from this application is shown below.



String Operators

Built In Operators:

The following is a list of all built in operators available in string expressions. All operator names are case insensitive, though actual strings are case sensitive.

All operators that use or return numeric values are 1 based (1 is the first item).

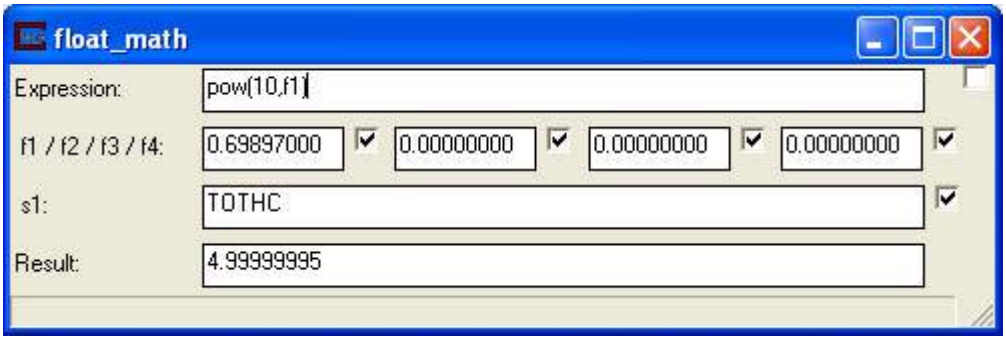
Left(text, num)	This function takes two parameters. The first is a string, the second is the number of characters to return. The first "num" characters are returned.	Left("abcdef",3) = "abc"

Mid(text, start, [length])	This function takes three parameters. The first is a string. The second is the first character to return. The third (optional) parameter is the number of characters to return, by default 1024.	Mid("abcdef",3,2) = "cd" Mid("abcdef",3) = "cdef"
switch_delimited	This function returns a subset of text given: the index of the subset; a string of text values; and a string representing the delimiter that defines how the text array is parsed.	switch_delimited(2, "zero|one|two|three", "|") = "two"
Switch(index,text1,[text2],[text3],[])	This function takes 2 or more parameters. The first is an integer 0-based index of which of the other parameters to return. The second and subsequent parameters must all be strings. The "index" parameter is returned.	Switch(1,"one","two","three") = "two"
Upper(text)	Convert the string in "text" to upper case.	Upper("abcdef") = "ABCDEF"
Lower(text):	Convert the string in "text" to lower case.	lower("aBcDeF") = "abcdef"

Title(text) or Proper(text)	Convert the string in "text" to title case.	Title("make this look like a title") = "Make This Look Like A Title" Proper("FIX the CaPiTaliZaTIon HEre") = "Fix The Capitalization Here"
Len(text)	Returns the number of characters in a string	
Right(text,number_of_chars)	Return the right-most characters in text	right("c:/ctech/mydata.apdv",4) = ".apdv"
NL() or CR()	Add a newline character	"Line 1" + NL() + "Line 2" = Line 1 Line 2
QUOTE()	Add a double quotation mark character	"One " + QUOTE() +"Two" + QUOTE() + " Three" = "One "Two" Three"
APOSTROPHE()	Add an apostrophe character	"The analyte" + APOSTROPHE() + "s concentration" = "The analyte's concentration"
TAB()	Will insert a tab in the text	"30" + TAB() + "ppm" = "30 ppm"
Pos(text,search_text)	Return the integer position of first occurrence of substring in text	Pos("c:\ctech\mydata.apdv","\") = 3
RPos(text,search_text)	Return the integer position of last occurrence of substring in text (searches backwards)	Pos("c:\ctech\mydata.apdv","\") = 9

float_math

This module provides a simple means to perform mathematical operations on numbers coming from up to 4 input ports. By using multiple float_math modules, any number of values may be combined.



The panel for float_math is shown above. The default equation is f1 + f2 + f3 + f4 which adds all four input ports.

Pop-upAvailable Mathematical Operators hereorJump to Available Mathematical Operators here

Any of these operators may be used.

The output (rightmost output port) is the numeric value resulting from the equation.

The value will update when any of the input values are changed unless the checkbox next to the input value is turned off.

Related Modules

string_format

draw_lines

General Module Function

The draw_lines module enables you to create both 2D and 3D lines interactively with the mouse.

The mouse gesture for line creation is: depress the alt key and then click the left mouse button on any pickable object in the viewer. The first click establishes the beginning point of the line segment and the second click establishes each successive point.

draw_lines allows adding of points that are outside the model extents, undoing of the last picked point, and the clearing of all picked points. Unlike most modules which create mesh data to used by other modules, the draw_lines module receives input from the viewer, and also passes on field data to be used by other modules.

There are two drawing modes:

1) Top View Mode creates 2D lines which are always at Z=0.0. You must be in a Top View to draw with this mode, but you may pick points anywhere in the Viewer screen.

2) Object Mode creates 3D lines which are drawn by probing objects in your model. You cannot draw at a point without having an object there or specifying a coordinate using the x-y-z type-ins.

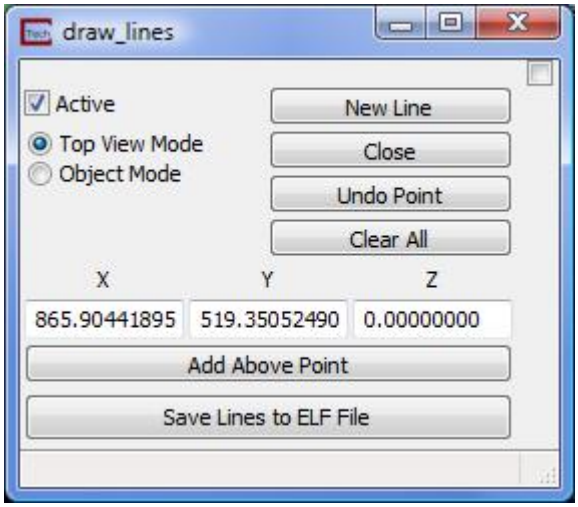
NOTE: Because draw_lines saves your lines with your application, when an application is saved, the purple port is automatically disconnected from the Viewer. This ensures that when you load an application the resulting objects (lines, fence-diagrams, etc.) will look exactly the same as when you saved the application. However, if you wish to draw new lines you will need to reconnect the purple port from the Viewer.

Module Input Ports

The draw_lines module has one input port. This port is connected to the output port of the viewer within which the line will be drawn.

Module Output Ports

The draw_lines module has two output ports. This first output port sends a mesh to downstream modules. The second output port sends the line as a renderable object to the viewer to be rendered.



Module Control Panel

The draw_lines user interface allows interactive creation of points or lines onto any connected object and then displays the points or lines in the viewer.

The **Active** toggle turns on and off the functionality of the module, letting you Alt+Left Click on an object in the viewer without drawing a line.

The **New Line** button is used to if multiple lines are desired. It should be used when a line is finished and the user wished to begin a new line. It is not needed to start a line.

The **Close** button is used to draw a point back to your starting point and thereby create a closed polyline. This is useful for modules which require closed polylines as input (e.g. triangulate_polygons).

The **Undo Point** button will undo the last entered point of the line.

Clear All will remove all points from all lines.

The **X, Y** and **Z** type ins by default reflect the last point entered into the line. If the user wants to add a point by hand the coordinates may be typed in and added to the line by pressing the Add Above Point button. If the user is entering a coordinate by hand they are not limited to using one within the extents of the model in the viewer.

The **Save Lines to ELF File** button allows the user to save all of the created lines in the EVS Line File format (*.elf).

polyline_spline

General Module Function

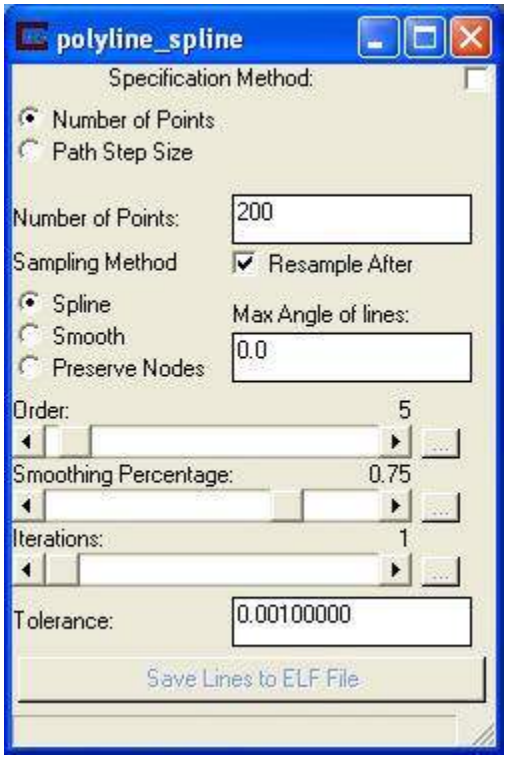
The polyline_spline module accepts a 3D polyline and can either increase or decrease the number of line segments of the polyline. A splining algorithm smoothes the line trajectory once the number of points are specified. This module is useful for applications such as a fly over application (along a polyline path drawn by the user). If the user drawn line is jagged with erratically spaced line segments, polyline spline smoothes the path and creates evenly spaced line segments along the path.

Module Input Ports

polyline_spline has one input port. This port should be a 2D or 3D polyline (perhaps from ContinuousSketch or draw_line).

Module Output Ports

polyline_spline has two output ports. The first output port (closest to the left) outputs the 3D or 2D polyline mesh. The second port outputs a renderable version of the polyline mesh to the viewer.



Module Control Panel

The user interface for polyline_spline is shown above.

The output line segment size can be adjusted via two modes; Number of Points mode or Path Step Size mode. Choosing the Number of Points toggle allows line smoothing adjustments by specifying the number of line segments (endpoints). Choosing the Path Step Size toggle allows smoothing adjustments by specifying the step size (in the units of the input mesh). The largest angle between any two line segments created by this module is displayed in the Max Angle of lines field.

If the **Spline** toggle is selected, the line is segmented and the original line nodes are used to determine the curve of the line. This option preserves the original nodes.

If the ***Smooth line*** toggle is ON, the line is both segmented and smoothed. All sharp corners are rounded off, how round the corners are made is determined by the Order, the number of Iterations, and the Smoothing Percentage.

If the ***Resample After*** toggle is ON, the line is segmented before **and after** being smoothed or splined.

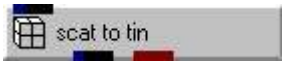
The ***Order*** slider determines the number of points to be used for splining. More points creates a smoother line that departs more from the original.

The ***Smoothing Percentage*** slider determines the degree of smoothing. Higher values creates a smoother line that departs more from the original.

The ***Iterations*** slider determines the number of smoothing iterations. This is analogous to multiple polyline_spline modules connected in series. This provides a means to further smooth erratic lines while staying close to the original points.

The ***Tolerance*** parameter determines how close two disjoint points must before the two polylines are merged.

scat_to_tin



General Module Function

The scat_to_tin module is used to convert scattered sample data into a three-dimensional surface of triangular cells representing an unstructured mesh.

"Scattered sample data " means that there are discrete nodes in space. An example would be geology or analyte (e.g. chemistry) data where the coordinates are the x, y, and elevation of a measured parameter. The data is "scattered" because there is not necessarily an implicit grid of data.

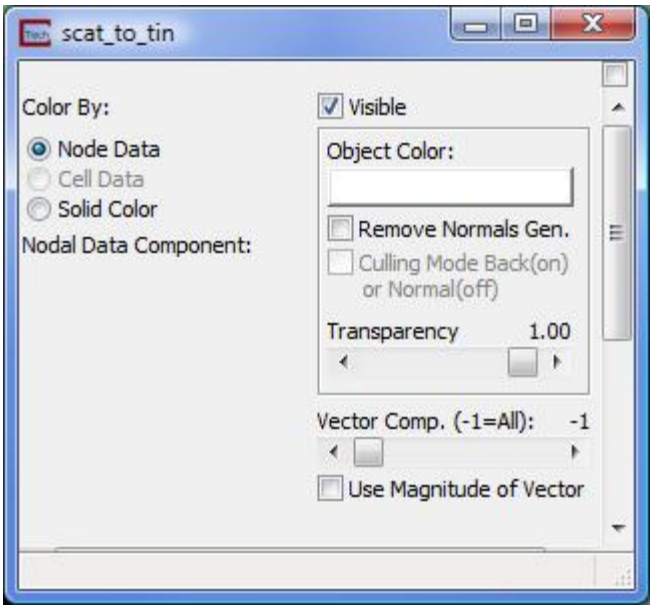
scat_to_tin uses a proprietary version of the Delaunay tessellation algorithm.

Module Input Ports

scat_to_tin has only one input port. Input to this port must contain scattered sample data. Any source of points, even the nodes comprising lines may be used as input. Examples are file_statistics, Load_CAD (Read_DXF), etc..

Module Output Ports

scat_to_tin has two output ports. The blue output port outputs the triangular 2D elements having interpolated values based on the input data. The red port can be connected directly to the viewer.



Module Control Panel

The user interface for this module is shown above.

Related Modules

scat_to_unif

scat_to_unif

General Module Function

The scat_to_unif module is used to convert scattered sample data into a three-dimensional uniform field. Also, scat_to_unif can be used to take an existing grid (for example a UCD file) and convert it to a uniform field. scat_to_unif converts a field of non-uniformly spaced points into a uniform field which can be used with many of EVS's filter and mapper modules. "Scattered sample data " means that there are disconnected nodes in space. An example would be geology or analyte (e.g. chemistry) data where the coordinates are the x, y, and elevation of a measured parameter. The data is "scattered" because there isn't data for every x/y/elevation of interest.

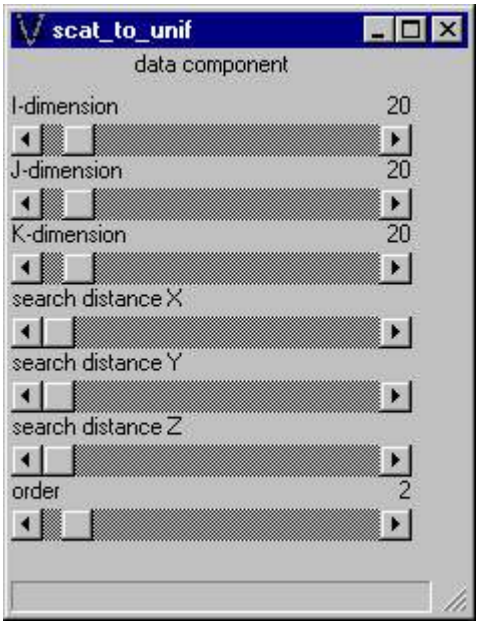
scat_to_unif lets you define a uniform mesh of any dimensionality and coordinate extents. It superimposes the input grid over this new grid that you have defined. Then, for each new node, it searches the input grid's neighboring original nodes (where search_cube controls the depth of the search) and creates data values for all the nodes in the new grid from interpolations on those neighboring actual data values. You can control the order of interpolation and what number to use as the NULL data value should the search around a node fail to find any data in the original input.

Module Input Ports

scat_to_unif has only one input port. Input to this port must contain scattered sample data. Currently, only the file_statistics module provides this output.

Module Output Ports

scat_to_unif contains one output port. The output port outputs the 3D uniform field having interpolated values based on the input data.



Module Control Panel

This module's control panel has the following parameters:

data_component: Radio buttons to choose the data for the field.

I-dimension: resolution of grid in x-direction. Max is 300.

J-dimension: resolution of grid in y-direction. Max is 300.

K-dimension: resolution of grid in z-direction. Max is 300.

search_distance_X: Initially set to the x extent of the data

search_distance_Y: Initially set to the y extent of the data

search_distance_Z: Initially set to the z extent of the data

order: An integer that controls the order of interpolation done on the adjacent nodes. Max value is 30.

Related Modules

scat_to_tin

material_to_cellsets

(This module is available only in MVS)

General Module Function

Material_to_cellsets is intended to receive a 3D field into its input port which has been processed through a module like plume_volume. If the original field (pre-plume_volume) had multiple cell sets related to geologic units or materials the output of plume_volume will generally have only two cell sets which comprise all hexahedron and all tetrahedron cells. The ability to control the visibility of the layer-cell sets is normally lost.

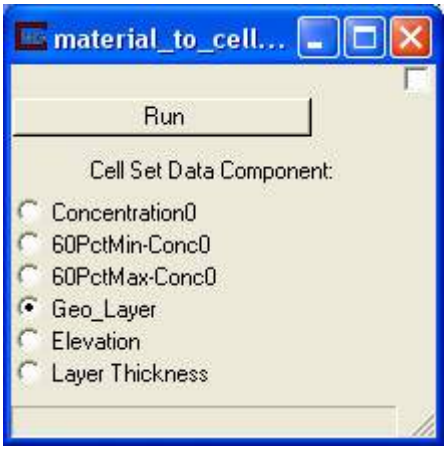
This module takes plume_volume's output and recreates the cell sets based on nodal data. However, since each geologic layer will likely have two cell sets each (one for all hexahedron and all tetrahedron cells), the output tends to have twice as many cell sets as the original pre-plume_volume field).

Module Input Ports

material_to_cellsets has one input ports. Material_to_cellsets receives a 3D field into its input port which has been processed through a module like plume_volume.

Module Output Ports

material_to_cellsets has one output port which outputs a field with recreated cell sets based on nodal data. Normally the Geo_Layer data component is used.



Module Control Panel

The user interface for material_to_cellsets is shown above. Clicking the Run button causes it to recreate the output field. This should be done AFTER selecting the appropriate data component for establishing cell set relationships.

file_output

General Module Function

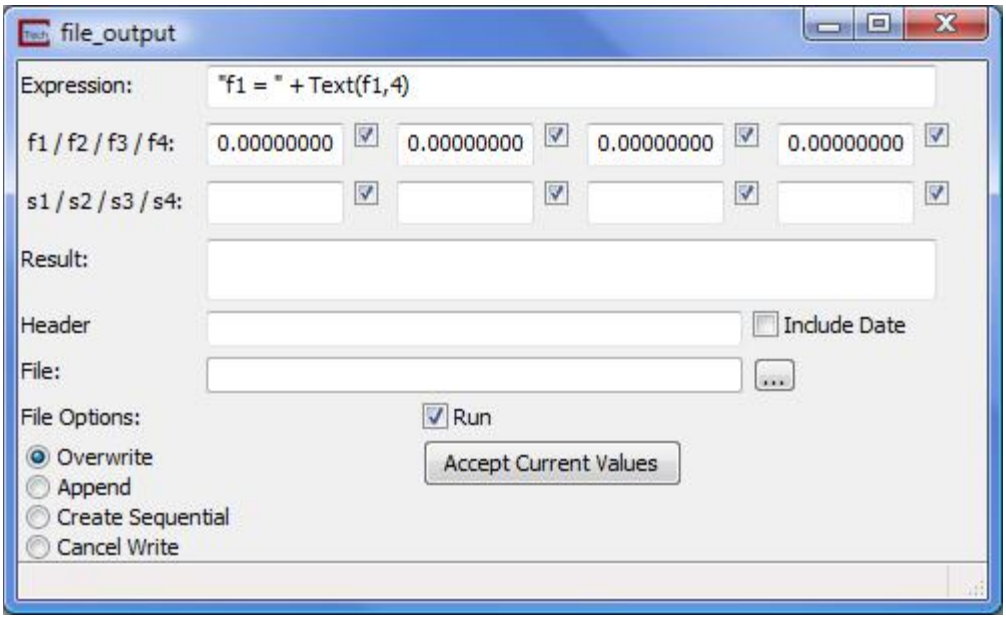
The file_output module creates a formatted string based upon the values passed to it. This string is then written to the selected ascii text file. Certain modules such as Krig_3D, Krig_2D, and Krig_Fence output a formatted string for just this purpose.

Module Input Ports

Ports 1– 4 are numeric values (Brownish Red): These ports will take float values.

Ports 5 – 9 are strings (Blue Green): These ports will take string values.

Module Control Panel



The control panel for file output is shown above.

The **Expression**, f1 – f4, s1 – s4, and result fields are the same as that of string format, and their descriptions can be found in the help for that module.

The **Header** text field will allow the user to type in a string that will be written on the line above the result string in the output file.

The **IncludeDate** toggle when selected will append the current date to the header string. If no header has been included the date will appear by itself on the line above the result string.

The **File** text field is the current filename to be written to. This file may be selected by selecting the "" button at the end of the field.

The **File Options** radio box lists the different ways the file can be written to or created.

1. Overwrite: This option will cause the file to be overwritten with the selected output any time there is a change in one of the input parameters.
2. Append: This option will cause the selected output to be appended to the end of the file.
3. Create Sequential: This option will cause a new file to be created every time the input changes. These files will have names based upon the selected output file and appended with a numeric sequence (ex. Krig_2D_output_001.txt). Every time the input changes the sequence

number increases (ex. Krig_2D_output_002.txt, Krig_2D_output_003.txt, etc.).

4. Cancel Write: This option will create and write the result string to the selected file, only if the file does not currently exist. All input changes will be ignored.

The **Run** toggle when not selected will prevent the module from running.

The **AcceptCurrentValues** button will cause the module to run even if the Run toggle is not selected.

FileTools

General Module Function

For subsetting the FileTools module reads .apdv, aidv, or gmf files and creates a simplifies the by first deleting all the lines where the parameter data is -1.00E+09 (missing data). Then it keeps every nth line of the file starting with the first line. This is very useful when you have an overly large data file that is causing the kriging to choose all points from the same local region.

For refining the File_tools module reads .pgf files and removes duplicate material values on the same boring so that materials are not over defined. If the topmost material in a Boring is not duplicated an additional sample will be added to the boring.

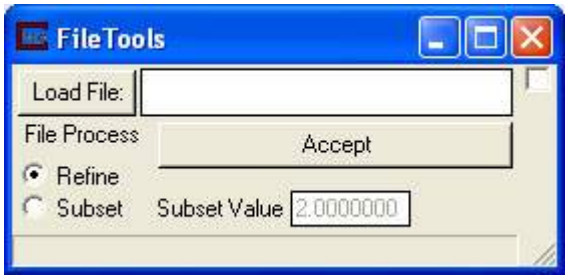
Module Input Ports

No input ports

Module Output Ports

No output ports

Module Control Panel



The control panel of File Tools is shown above. The Load File button opens a File browser which lists the files that are present in the current directory shown in the directory window. The file name this module creates is dependent on the options selected. If initial_soil_investigation_subsite.apdv and the subset option is chosen, the output file will be initial_soil_investigation_subsite_subset.apdv . If refine is chosen, the file initial_soil_investigation_subsite.pgf will be initial_soil_investigation_subsite_refined.pgf.

modflow_converter

General Module Function

The modflow_converter module will import MODFLOW data from the following sources. For each, it is important that the reference files be present and any files referenced in those files also be in the same folder.

- MODFLOW98 :
  - Requires: *.BAS, *.BCF files
- MODFLOW2000
  - Requires *.NAM,*.DIS, *.BAS6 files
- GMS 8.1
  - Requires *.GPR file
- Visual MODFLOW
  - Requires *.VMG file
- Groundwater Vistas
  - Requires same files as MODFLOW

It will create time based EVS field files (.eff or .efz when Compress Files option is toggled) for the selected *Output Data.*

The MODFLOW simulation should be run to completion before the conversion is attempted.

For GMS and Visual MODFLOW the project files will contain links to files generated by the simulation.

Certain data will require additional files to be present.

For example head data will be created once the MODFLOW simulation is complete, usually generating a binary *.hed file.

This file should be referenced in the NAM file and then can be included in the generated EVS Field Files.

Concentration data is created when the MT3D simulation is run to successful completion and should be referenced in the NAM file as well.

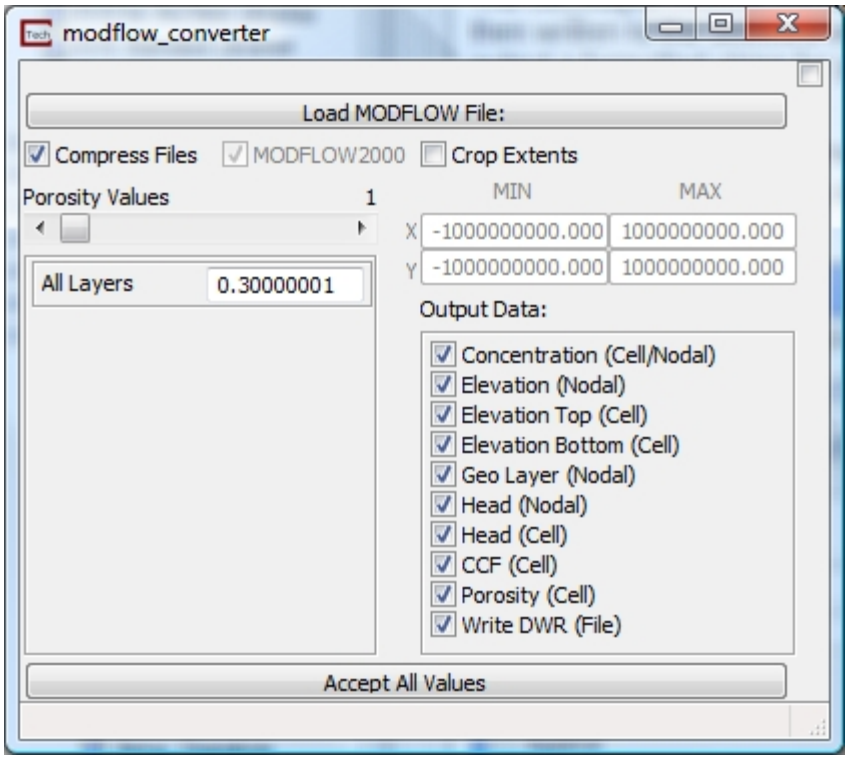
Module Input Ports

No input ports

Module Output Ports

No output ports

Module Control Panel



The control panel of modflow_converter is shown above.

- The Load MODFLOW File button will select the model file to be imported.
- The Compress Files toggle when selected will change the output from (*.eff) field file to the compressed (*.efz) field file.
- The MODFLOW2000 toggle is active if a *.nam is selected. When selected the module expects the model file to be associated with the MODFLOW2000 format as opposed to MODFOW98.
- The Crop Extents toggle when selected will return only the portion of the model that lies within the X/Y min-max range.
- The Porosity values slider is set to 1 by default. In this instance the value indicated (0.3 by default) in the "All Layers" field will be the default value for porosity on all layers. The slider can alternately be increased to the number of layers in the model and the default value for each layer can be set.
- Each type of data in the output can be toggled on or off in the Output Data frame. NOTE: Concentration (MT3D data) will generate a separate set of time based EVS field files.
- When the Accept All Values button is selected the module will parse the selected MODFLOW based file. It will create three folders in the project directory:
    1. GENERAL;

2. MODFLOW and
3. If Concentration has been selected MT3D.

The GENERAL folder will have two files which contain the coordinates of each node and the connectivity for the cells. The MODFLOW folder will contain the time based field files that include the selected MODFLOW data (i.e. Head, CCF, etc.). The MT3D folder will contain the time based files that include Concentration data.

modflow_converter

General Module Function

The modflow_converter module will import MODFLOW data from the following sources. For each, it is important that the reference files be present and any files referenced in those files also be in the same folder.

- MODFLOW98 :
  - Requires: *.BAS, *.BCF files
- MODFLOW2000
  - Requires *.NAM,*.DIS, *.BAS6 files
- GMS 8.1
  - Requires *.GPR file
- Visual MODFLOW
  - Requires *.VMG file
- Groundwater Vistas
  - Requires same files as MODFLOW

It will create time based EVS field files (.eff or .efz when Compress Files option is toggled) for the selected *Output Data.*

The MODFLOW simulation should be run to completion before the conversion is attempted.

For GMS and Visual MODFLOW the project files will contain links to files generated by the simulation.

Certain data will require additional files to be present.

For example head data will be created once the MODFLOW simulation is complete, usually generating a binary *.hed file.

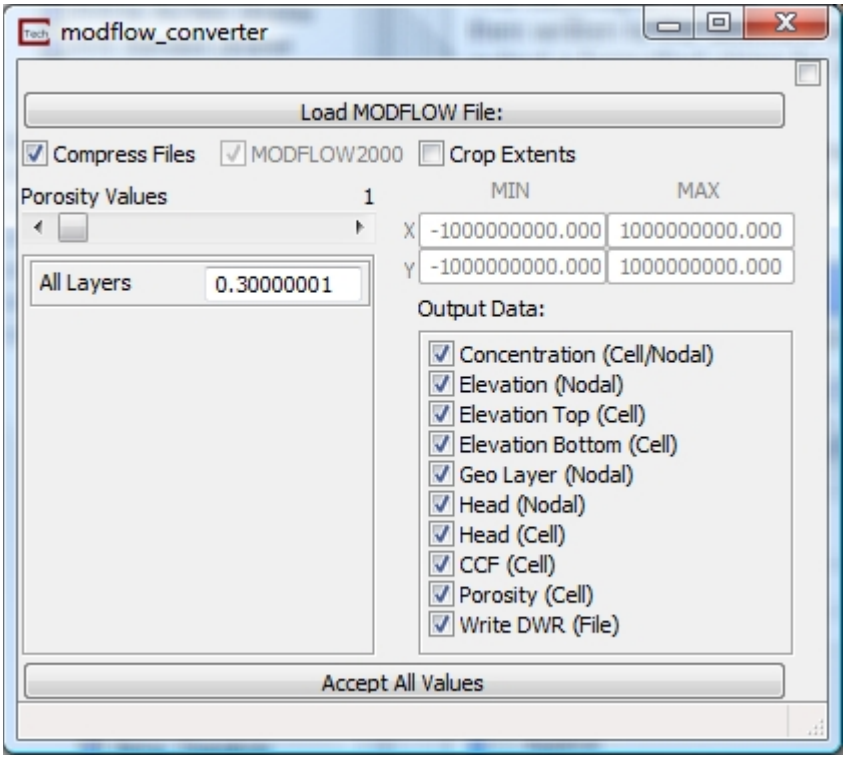
This file should be referenced in the NAM file and then can be included in the generated EVS Field Files.

Concentration data is created when the MT3D simulation is run to successful completion and should be referenced in the NAM file as well.

Module Input Ports
No input ports

Module Output Ports
No output ports

Module Control Panel



The control panel of modflow_converter is shown above.

- The Load MODFLOW File button will select the model file to be imported.
- The Compress Files toggle when selected will change the output from (*.eff) field file to the compressed (*.efz) field file.
- The MODFLOW2000 toggle is active if a *.nam is selected. When selected the module expects the model file to be associated with the MODFLOW2000 format as opposed to MODFOW98.
- The Crop Extents toggle when selected will return only the portion of the model that lies within the X/Y min-max range.
- The Porosity values slider is set to 1 by default. In this instance the value indicated (0.3 by default) in the "All Layers" field will be the default value for porosity on all layers. The slider can alternately be

increased to the number of layers in the model and the default value for each layer can be set.

- Each type of data in the output can be toggled on or off in the Output Data frame. NOTE: Concentration (MT3D data) will generate a separate set of time based EVS field files.
- When the Accept All Values button is selected the module will parse the selected MODFLOW based file. It will create three folders in the project directory:
  1. GENERAL;
  2. MODFLOW and
  3. If Concentration has been selected MT3D.

The GENERAL folder will have two files which contain the coordinates of each node and the connectivity for the cells. The MODFLOW folder will contain the time based field files that include the selected MODFLOW data (i.e. Head, CCF, etc.). The MT3D folder will contain the time based files that include Concentration data.

Modpath DWR/DWZ File Example

Discussion of DWR/DWZ Files

DWR/DWZ files contain the package parameters from MODFLOW projects. These are stored as either a single record, for steady state simulations, or as a group of records based on date for transient simulations. The different packages supported are: DRAINS; WELLS; RECHARGE; ET (evapotranspiration); CONSTANT_HEAD; GENERAL_HEAD; RIVER_LEAKAGE; and STORAGE.

Format:

You may insert comment lines in DRW/DWZ input files. Comments can be inserted anywhere in a file and must begin with a '#' character. The line numbers that follow refer to all **non-commented** lines in the file.

Line 1: Should contain the word GRID.

Line 2: The GRID flag should be followed on the next line by the following grid parameters in order: the grid rotation about the z axis, the translation of the grid from the origin in the x direction, the translation of the grid from the origin in the y direction, the translation of the grid from the origin in the z direction.

Line 3: Should contain the word TIME followed on the same line by one of the following abbreviations indicating the time units of the file: "yr" – year, "d" – day, "h" – hours, "m" – minutes, "s" – seconds.

Line 4: Will contain either the word "STEADY_STATE" for steady state simulations, or the word "Date" followed by a date in the standard short date format.

Lines 5+: Should contain one of the package headers mention above (DRAINS,WELLS,etc..), followed on the subsequent lines with the coordinates of the center of the cell, the flow due to that package in that cell, and the face (if applicable) at which the flow is occurring. The faces are defined in the following order: top - 6, bottom - 5, right - 1, left - 2, front - 4, and back - 3. If the flow is not followed by a face number or is given a face number of 0 then the flow is applied to the entire cell and not to a cell face.

The word END on any line prevents further parsing of the file.

Steady State File example:

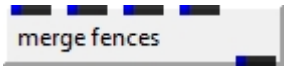
```
# EVS\MVS generated DWR file
GRID
0.000000 0.000000 0.000000 0.0
TIME d
STEADY_STATE
DRAINS
57500.000000 67500.000000 25.000000 -26006.757813 6.000000
WELLS
62500.000000 42500.000000 25.000000 100000.000000 0.000000
CONSTANT_HEAD
2500.000000 72500.000000 25.000000 -3394.514160
2500.000000 67500.000000 25.000000 -3415.331787
2500.000000 62500.000000 25.000000 -3453.412109
END
```

Transient File example:

```
# EVS\MVS generated DWR file
GRID
0.000000 0.000000 0.000000 0.0
TIME d
DATE 5/31/2146
DRAINS
57500.000000 67500.000000 25.000000 -26006.757813 6.000000
WELLS
62500.000000 42500.000000 25.000000 100000.000000 0.000000
CONSTANT_HEAD
2500.000000 72500.000000 25.000000 -3394.514160
2500.000000 67500.000000 25.000000 -3415.331787
2500.000000 62500.000000 25.000000 -3453.412109
DATE 10/28/2392
DRAINS
57500.000000 67500.000000 25.000000 -25082.052734 6.000000
WELLS
```

```
62500.000000 42500.000000 25.000000 90000.000000 0.000000
CONSTANT_HEAD
2500.000000 72500.000000 25.000000 -3022.231934
2500.000000 67500.000000 25.000000 -3042.281006
2500.000000 62500.000000 25.000000 -3079.266602
```

merge_fences



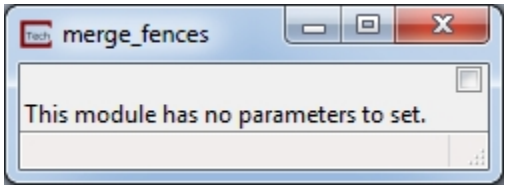
The merge_fences module is used to merge the output from multiple Krig_Fence modules into one data set (i.e., to merge cross sections into a fence diagram). This is useful for performing uniform data manipulation procedures on fence data from several Krig_Fence outputs. For example, if several Krig_Fence modules are used, they should all pass through a merge_fences module before being passed to explode and scale. Therefore, all fences will be exploded and scaled the same amount and only one dialog box is needed to control all fences. merge_fences should always be used when more than one krig_fence module is used.

Module Input Ports

merge_fences has four identical input ports which can accept data from krig_fence modules.

Module Output Ports

merge_fences has one output port which contains a reference to all fence data sent to the input port. The fence data is unaltered in the output.



link_modules

General Module Function

This module uses a method similar to the Animator to create invisible links between parameters within modules that are not normally directly linkable. Any animatable parameter can be used as a link with this module. The full set of math operations areavailablein these links, allowing complex expressions to be made, as well.

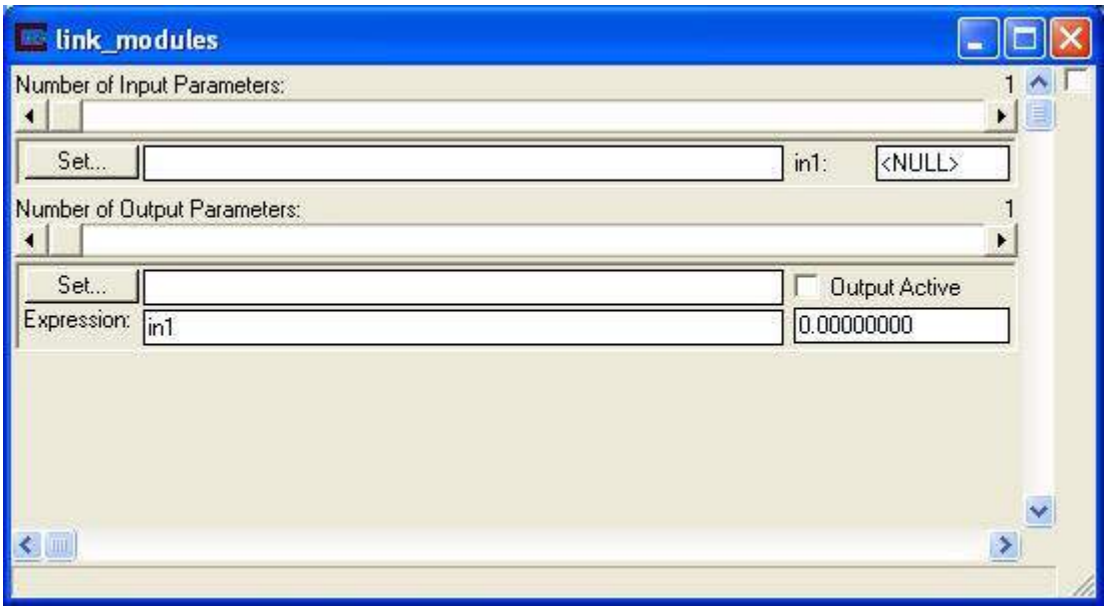
As an example, this module can be used to link together two slice_horizontal modules, even though the position parameter is not available as a link in the module. The position of the a slice_horizontal can be linked to the position of another slice horizontal, so that when the first slice is moved, the second will always stay at the same location.

Module Input Ports

link_modules has no input ports.

Module Output Ports

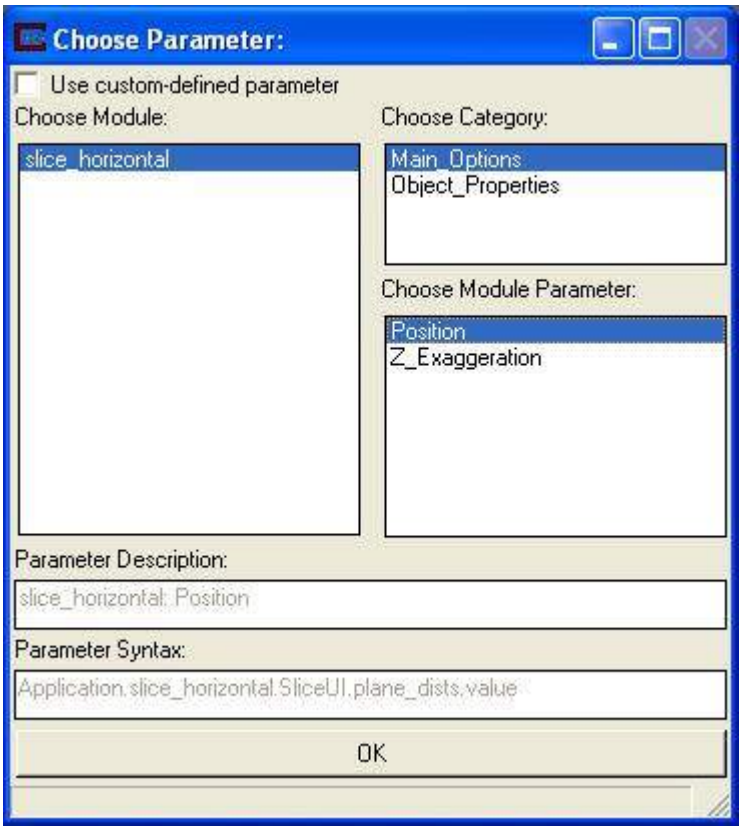
link_modules has no output ports.



Module Control Panel

The control panel for link_modules is shown in the figure above.

The ***Number of Input Parameters*** slider indicates how many variables from all of the instanced modules should be used for input. The ***Set*** button opens the window shown below. This window allows you to choose which variable of which module to use for input. Each variable is assigned a name (**in1, in2, in3**,) from which it can be referred to in output portion of the module. This variable name is followed by the actual value of the selected variable.



The ***Number of Output Parameters*** slider indicates how many variables from all of the instanced modules should be affected by the Input Parameters. The ***Set*** button opens the window shown above. This window allows you to choose which variable of which module that you wish to have affected by an input variable. The output parameter chosen is listed in the field to the right of the button. The field under the chosen parameter is used to set the new value for the output parameter. The syntax is the same as the data_math module, allowing the full spectrum of mathematical operations to be used inside the link between the modules. Each input is assigned as a variable, and each variable can be used in any of the output parameters. This allows a single link_modules module to control any number of links between parameters.

GroupObject

General Module Function

GroupObject is a renderable object that contains other subobjects that have the attributes that control how the rendering is done. Unlike DataObject, GroupObject does not include data. Instead, it is meant to be a node in the rendering hierarchy that groups other DataObjects together and supplies common attributes from them. This object is connected directly to one of the viewers (for example, SimpleViewer3D) or to another DataObject or to GroupObject. A GroupObject is included in all the standard viewers provided with the EVS applications chooses.

GroupObject combines the following:

* DefaultDatamap to convert scalar node or cell data to RGB color values. By default, the datamap's minimum and maximum values are 0 and 255, respectively. This datamap is inherited by any children objects if they do not have their own datamaps.

* DefaultProps to control color, material, line attribute, and geometrical attributes.

* DefaultModes to control point, line, surface, volume, and bounds rendering modes.

* DefaultPickInfo to contain information when this object is picked.

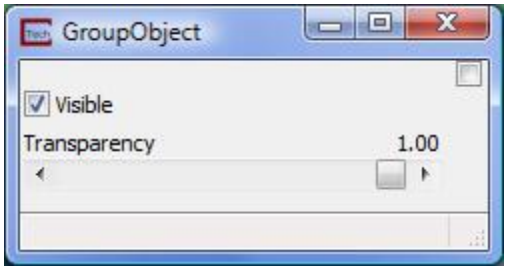
* DefaultObject to control visibility, pickability, caching, transform mode, surface conversion, and image display attributes.

Module Input Ports

This module has a single red input port that can accept one or more objects (red ports) as inputs. Each input will become a child object of the output. Multilevel object hierarchies can be built in this manner.

Module Output Ports

The renderable object that can be connected directly to a viewer.



Module Control Panel

The user interface for this module is shown above.

Related Modules

Render_Field

Transform_Group

create_grid

General Module Function

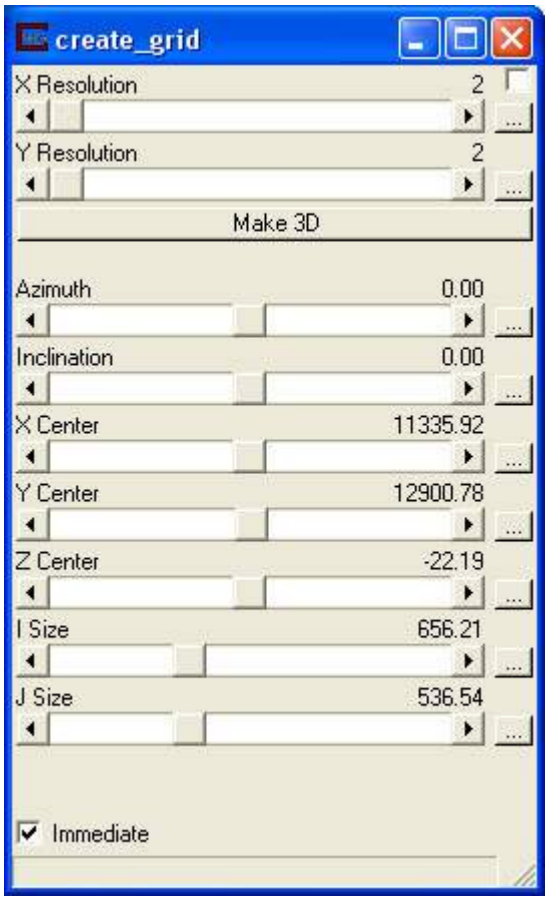
The create_grid module produces a 2D or 3D uniform grid that can be used for any purpose, however the primary application is as starting points for streamlines or advector. In 2D (default) mode it creates a rectangle of user adjustable grid resolution and orientation. In 3D mode it creates a box (3D grid of nodes).

Module Input Ports

create_grid has one input port which accepts any 3D mesh data. The purpose of this input is to help place your grid in the spatial extent of your existing model.

Module Output Ports

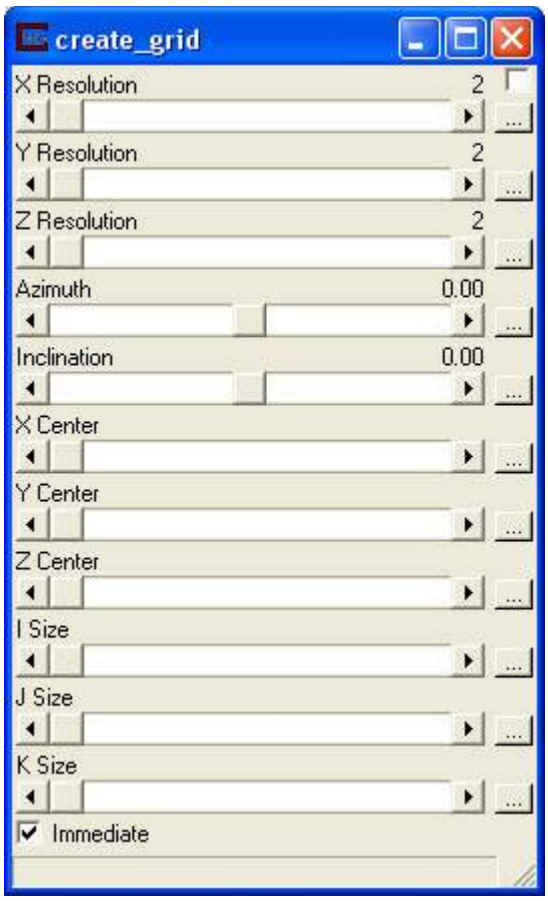
create_grid has two output ports. The first port passes a 2D mesh representing the plane The second port passes the renderable geometry to the viewer.



Module Control Panel

The user interface for create_grid consists of numerous sliders for manipulation of the resulting grid's spatial extent and orientation.

The X, Y and Z Resolution sliders control the number of the nodes in the x-y-z directions of the grid. The Z Resolution slider is not visible unless you press the Make3D button as shown below.



The Azimuth and Inclination sliders control the grid orientation.

The X, Y and Z Center sliders control the position of the grid.

The I, J and K Size sliders control the size of the grid in the x, y, and z directions (before rotations).

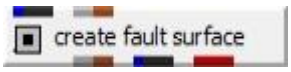
The Immediate toggle makes all sliders immediate.

Related Modules

->**Advector**

-> **Streamlines**

create_fault_surface



(Available only in MVS)

General Module Function

The create_fault_surface module creates a 3D grid that is aligned to a specified strike and dip.

Module Input Ports

1) This input field helps auto size and position your grid.

2) z_scale (Grey-Brown) : This port is for the z exaggeration factor.

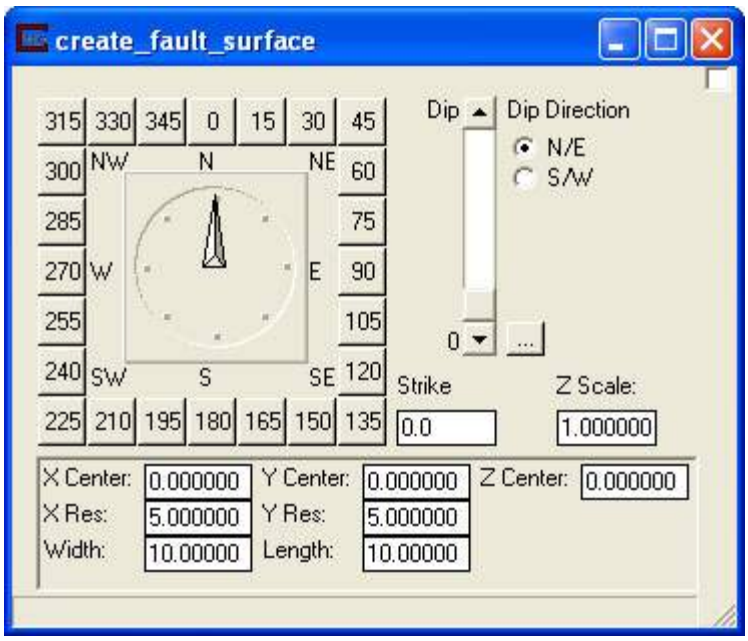
Module Output Ports

1) z_scale (Grey-Brown) : This port outputs the z exaggeration factor.

2) out_fld (Blue-Black) : This port outputs a 3D field aligned to a specified strike and dip.

3) out_obj (Red) : This port outputs a renderable object representing the created fault surface

Module Control Panel



The *Strike* value of the grid can either be dialed in or set in the field.

The *Dip* value can be set using the slider.

The *X Center, Y Center* and *Z Center* set the position of the center of the grid.

The *X Res* and *Y Res* field set the resolution of the grid in nodes in the x and y directions.

The *Width* field scales the grid in the X direction.

The *Length* field scales the grid in the Y direction.

create_spheroid

General Module Function

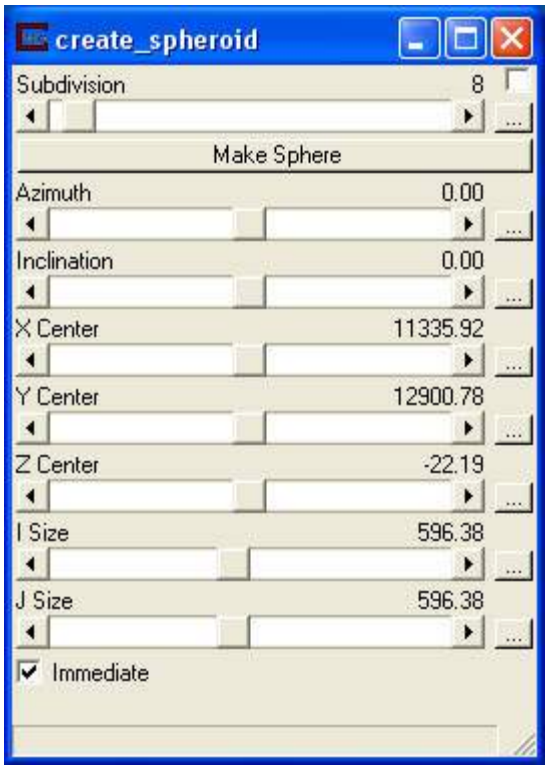
The create_spheroid module produces a 2D circular disc or 3D spheroidal or ellipsoidal grid that can be used for any purpose, however the primary application is as starting points for streamlines or advector.

Module Input Ports

create_spheroid has one input port which accepts any 3D mesh data. The purpose of this input is to help place your grid in the spatial extent of your existing model.

Module Output Ports

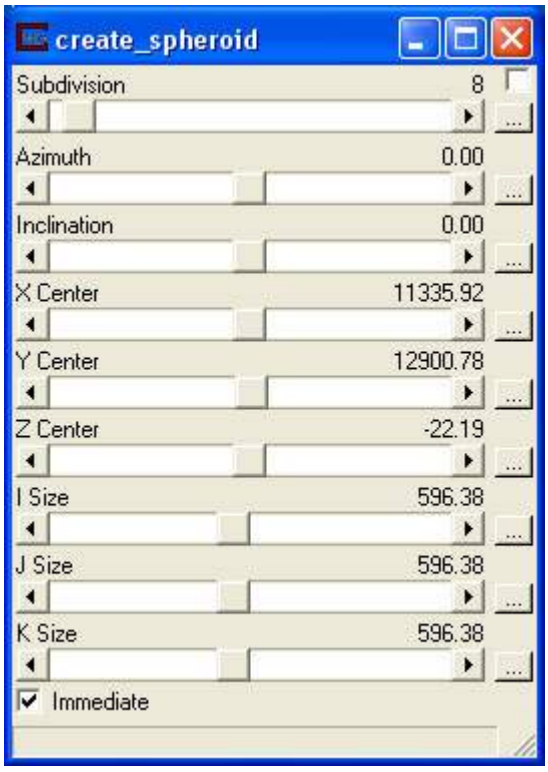
create_spheroid has two output ports. The first port passes a 2D mesh representing the plane The second port passes the renderable geometry to the viewer.



Module Control Panel

The user interface for create_spheroid consists of numerous sliders for manipulation of the resulting grid's spatial extent and orientation.

The Subdivision slider control the number of the nodes around the disc. The Azimuth and Inclination sliders are not visible unless you press the ***MakeSphere*** button as shown below.



The Azimuth and Inclination sliders control the grid orientation.

The X, Y and Z Center sliders control the position of the grid.

The I, J and K Size sliders control the size of the grid in the x, y, and z directions (before rotations). When these are different you have an ellipsoid.

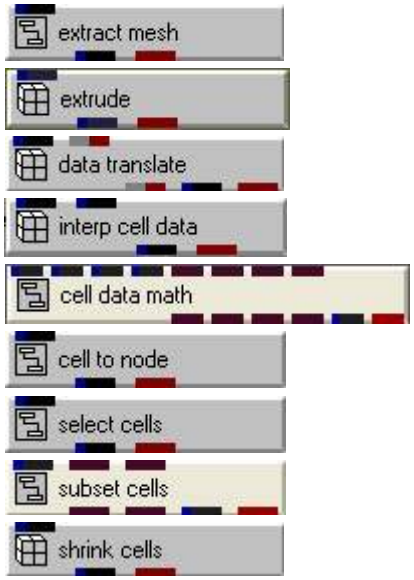
The Immediate toggle makes all sliders immediate.

Related Modules

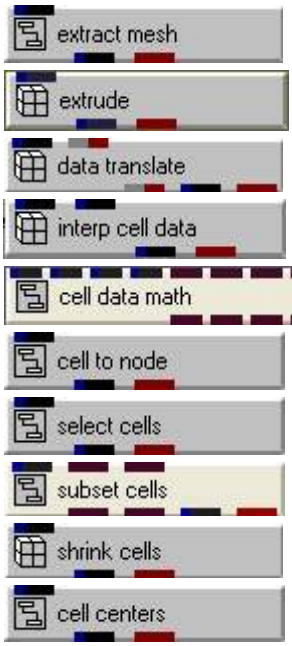
->**Advector**

-> **Streamlines**

Cell Data Modules



Cell Data Modules



select_data

General Module Function

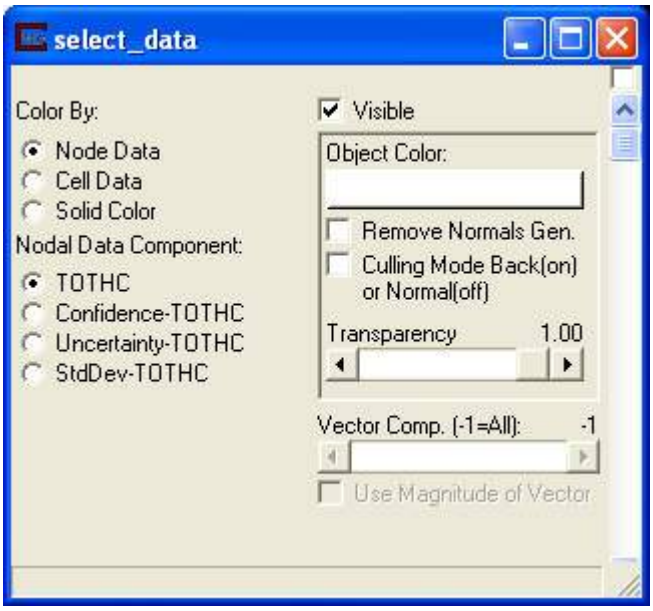
The select_data module extracts a single data component from a field. Select_data can extract scalar data components or vector components. Scalar components will be output as scalar components and vector components will be output as vector components.

Module Input Ports

Select_data has only one input port which accepts any nodal data (multiple scalar and/or vector).

Module Output Ports

Select_data has two output ports. The first port (closest to the left) outputs a new nodal data component containing only the extracted nodal data component. If mesh data was included in the input, it will be contained in the output. The second port will output a renderable object if the data are faces or lines.



Module Control Panel

The control panel for select_data is shown in the figure above.

The Color By radio button controls which type of data to export.

1)The Node Data component list contains all of the nodal data components piped into the input port. You can also choose to extract a single scalar component of a vector data component.

2) The Cell Data component list contains all of the cell data components piped into the input port. Only one component can be selected at a time and the default selection is the first (0th) data component. Any modules downstream of select_data will receive only the selected data component.

3) If Solid Color is selected the output field will have neither cell or nodal data but will be colored according to the Object Color button on the right panel.

The right side panel provides control over several commonly edited object properties including transparency, Normals generation, culling mode and visibility.

Related Modules

->extract_scalar

-> combine_comp

-> combine_vect

extract_mesh

This module has been deprecated and superceded by select_data

General Module Function

The extract_mesh module strips scalar nodal data from a mesh and passes it on without any scalar data, cell data or vector data associated with it.

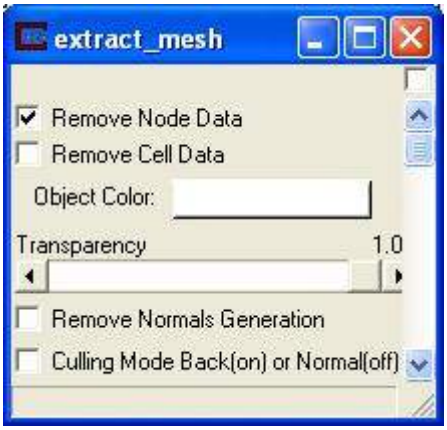
Module Input Ports

Extract mesh has only one input port which accepts any mesh and optionally removes nodal or cell data. The mesh data in the input passes through extract scalar unaffected.

Module Output Ports

Extract mesh has two output ports. The first port (closest to the left) outputs the mesh nodal coordinates The second port (red) is a renderable object if the input is faces or lines. The rendered object from the second port will have no other data attached to it

Module Control Panel



Remove Node Data toggles the node data to be removed from the input mesh.

Remove Cell Data toggles the cell data to be removed from the input mesh.

Object Color changes the color of the output mesh.

Transparency is a slider indicating the opacity, or transparency of the output mesh.

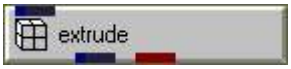
Remove Normals Generation removes normals from the output mesh.

Cull Mode toggles the culling mode to be Back or Normal.

Related Modules

-> select_data

extrude



General Module Function

The extrude module accepts any mesh and adds one to the dimensionality of the input by extruding the mesh in the Z direction. The interface enables changing the height scale for extruded cells and extruding by a constant, any nodal or cell data component. This module is often used with the Read_Shapefile module to convert polygonal shapefiles into extruded volumetric cells.

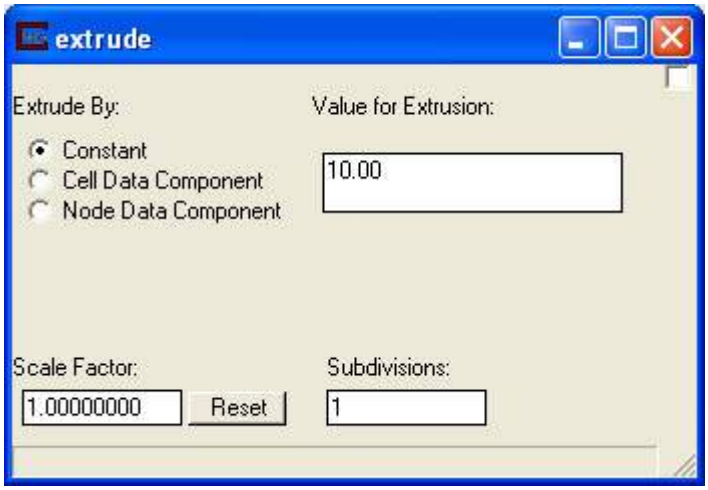
Module Input Ports

The extrude module has a single input port that accepts any non-volumetric mesh. Point, line, or polygonal input is accepted, as well as any number of mesh or cell data components.

Module Output Ports

The extrude module has two output ports. The blue/black output port is the new mesh which will have a dimensionality one greater than the input. The new mesh will include all of the nodal and cell data in the input field. The red port outputs a renderable object which will be colored by the first nodal data component in the input field, the first cell data component if the input field has no nodal data, or a constant color if the input mesh has no data components.

Module Control Panel

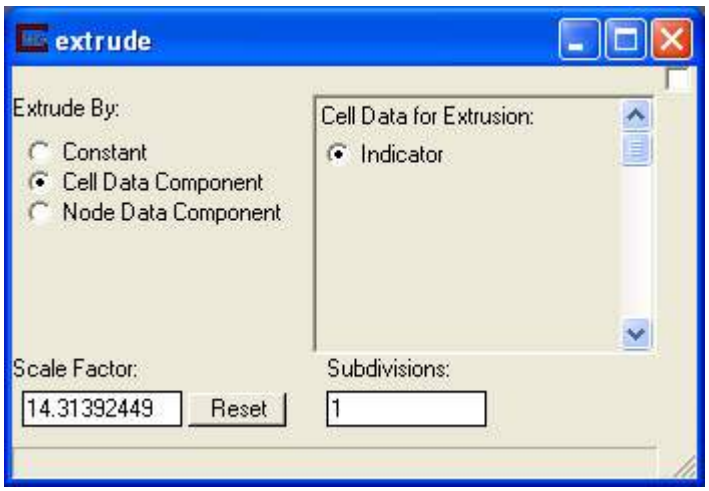


The control panel for extrude is shown above. The "Extrude By" radio buttons control the method of extrusion. The "Node/Cell Data for Extrusion" radio buttons determine which data component will be used. The output will contain all data components.

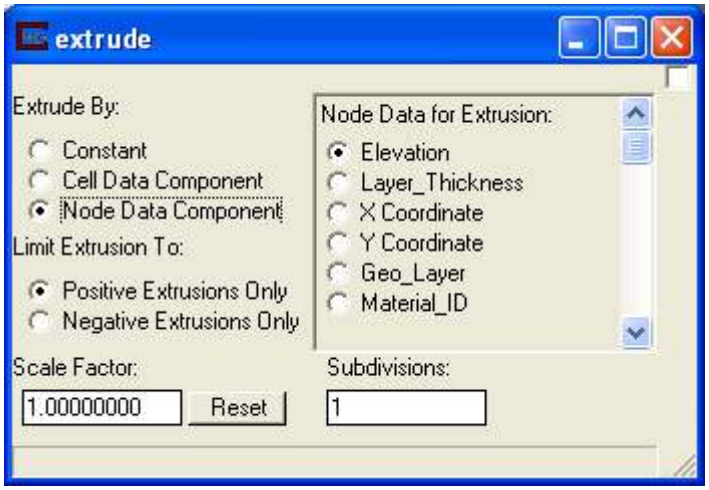
The Scale Factor is multiplied by the data value to determine the extrude height.

The Subdivisions field determines how many cells vertically to create.

When Constant is selected, the cells in the output mesh will be extruded by the "Value for Extrusion" times the "Scale Factor".



When Cell Data Component is chosen, the output cells will be extruded by the Scale Factor times the value of whichever cell data component is selected on the right.



When Node Data Component is chosen, the output cells will be extruded by the Scale Factor times the value of whichever nodal data component is selected on the right. With nodal data extrusion you must select "Positive Extrusions Only" or "Negative Extrusions Only". Since each node of a triangle or quadrilateral can have different values, it is possible for a single cell to have both positive and negative data values at its nodes. If this type of cell is extruded both directions, the cell topology can become tangled.

For this reason, nodal data extrusions must be limited to one direction. To extrude in both directions, merely use two extrude modules in parallel, one set to positive and the other to negative.

Related Modules

data_translate

data_translate

General Module Function

The data_translate module accepts **nearly** any mesh and translates the grid in x, y, or z based upon either a nodal or cell data component or a constant.

The interface enables changing the Scale Factor for z translates to accommodate an overall z exaggeration in your applications. This module is most useful when used with the Read_Shapefile module to properly place polygonal shapefile cells at the proper elevation.

Module Input Ports

The data_translate module has two input ports. The leftmost port accepts the Z Exaggeration factor from modules such as Explode_and_Scale.

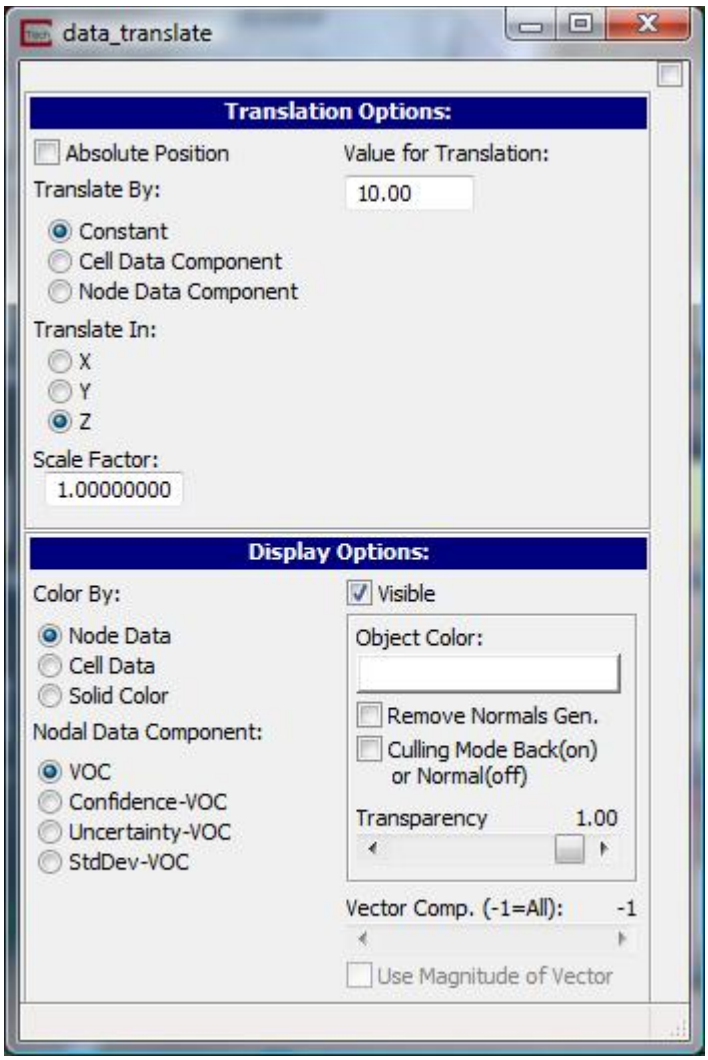
The right blue/black port accepts any mesh to be used with cell data or any non-volumetric mesh to be used with nodal data. Volumetric cells cannot be translated by their nodal data because cell topology can easily be corrupted.

Module Output Ports

The data_translate module has three output ports. The leftmost port outputs the Z Exaggeration factor to modules such as Explode_and_Scale.

The center blue/black output port is the new mesh which will have the translated version of the input. The new mesh will include all of the nodal and cell data in the input field. The red port outputs a renderable object which will be colored by the first nodal data component in the input field, the first cell data component if the input field has no nodal data, or a constant color if the input mesh has no data components.

Module Control Panel



The control panel for data_translate is shown above and below.

The **Translate By:** radio buttons control the method of translation. Depending on the method selected, the options in the module (and the appearance of the panel) vary.

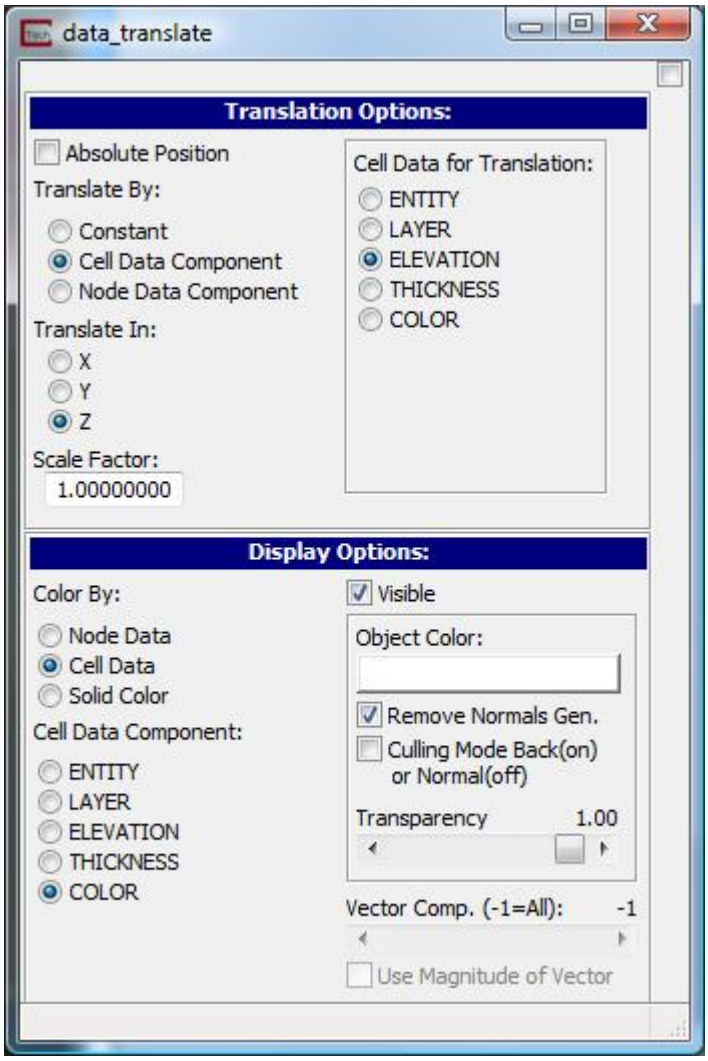
The **Translate In:** radio buttons determine the axis of translation.

Warning: The scale factor is always applied. If translating along any axis other than z, it is unlikely that you want to use the Z Exaggeration factor used elsewhere in your application.

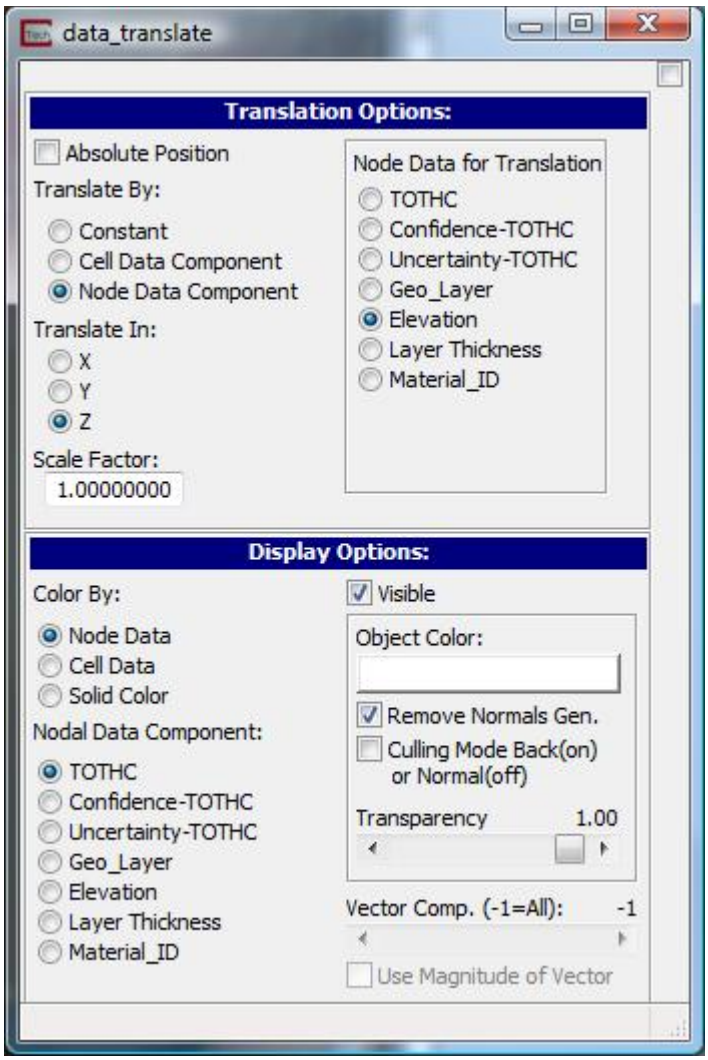
- When translating by a Constant, the amount is affected by the Z Scale Factor.
- When translating by Cell Data, a radio box appears to allow specification of the cell data component
- When translating by Node Data, a radio box appears to allow specification of the nodal data component

Under the **Display Options** subpanel you can specify the data for coloring as well as typical object properties.

The control panel shown below shows a cell data case that was used in the application shown at the bottom of this topic.



The control panel shown below shows a nodal data case.



When Node Data Component is chosen, the output cells will be translated by the Scale Factor times the value of whichever nodal data component is selected on the right. With nodal data you can only translate non-volumetric cells, otherwise the cell topology can become tangled.

Related Modules

extrude

interp_cell_data

General Module Function

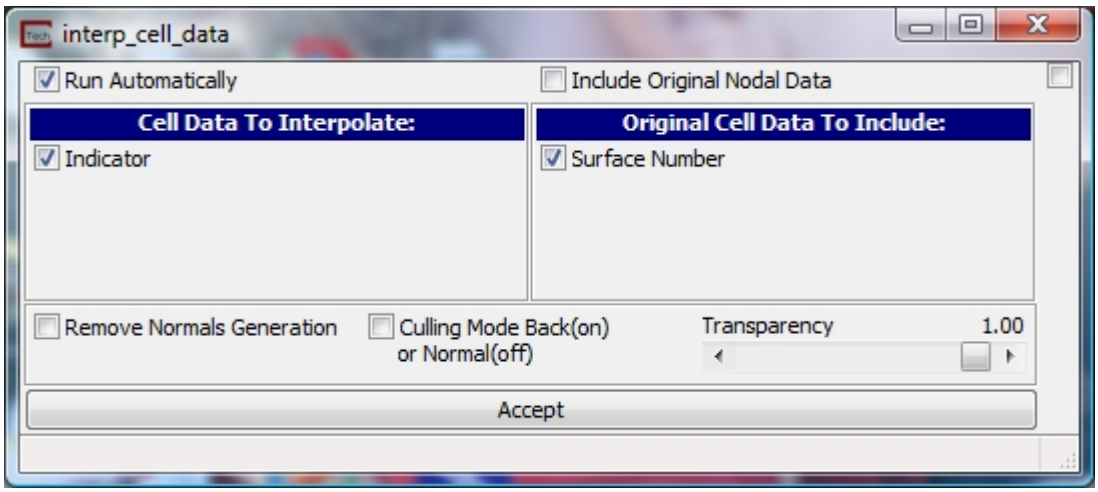
The interp_cell_data module interpolates cell data from one field to another using a Nearest Neighbor interpolation. Typical uses of this module are mapping of cell data from a 3D mesh onto a geologic surface or a 2D fence section. In these applications the 2D surface(s) simply provide the new geometry (mesh) onto which the adjacent cell values are interpolated.

Module Input Ports

The interp_cell_data module has two input ports.
1) in_field (Blue/Black): This port will accept a field which contains the parameters to be interpolated (all cell data components can be interpolated).
2) in_mapto (Blue/Black): This port accepts a field that the data will be interpolated to. Cell or nodal data are not needed for data sent to the right input port, because the mesh from the right input port is used strictly to obtain coordinate locations and cells where the left port's cell data will be interpolated.

Module Output Ports

Interp_cell_data has two output ports.
1) output_field (Blue/Black): This port outputs the data field with interpolated cell data. This port can be connected to any module that can process data fields with cell data, such as the contour cell module.
2) out_obj(Red): This port outputs a renderable geometry of the surface that can be input to the viewer.



Module Control Panel

The control panel for interp_cell_data is shown in the figure above.
The ***Run Automatically*** toggle will allow the module to run for every change made to the control parameters or input.
The ***Include Original Nodal Data*** toggle will cause all nodal data sent to the second input port(in_mapto) to be copied to the output field. NOTE: Leaving this toggle on will cause the nodal data to be displayed and not the interpolated cell data.
The ***Cell Data To Interpolate*** frame lists all of the cell data from the first input port(in_field) that is available for interpolation.
The ***Original Cell Data To Include*** frame lists all of the cell data from the second input port(in_mapto) that can be included in the output. This data will occur after the interpolated data in the field.

The ***Remove Normals Gen***eration toggle controls how vertices and edges are rendered. When this is ON, it makes more distinct edges, but a more faceted overall surface.

The ***Culling Mode*** toggle controls whether back facing surface are visible. Generally you will want this ON when making the object(s) transparent.

The ***Transparency*** slider determines the opacity of the objects.

cell_data_math

General Module Function

The cell_data_math module (only in EVS PRO and MVS) is used to perform mathematical operations on cell data fields. Up to four data fields can be input to cell field math. Mathematical expressions can involve any or all of these input fields.

Module Input Ports

cell_data_math has eight input ports.

There are four identical input ports that can accept any type of mesh, but the meshes must have the same cell types, the same number of nodes, and the same number of cells (i.e., they must have identical geometries). At least one input port must be used and up to four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right.

Cell data passed to ports one, two, three and four are referred to as Acx, Bcx, Ccx and Dcx (where x refers to the number of the data component) in the appropriate mathematical expression.

NOTE: MVS users also get Ax, Ay, Az which are the coordinates of the cell center. These can use this to affect the resultant cell data component and when MVS is run, the user interface adjusts accordingly (automatically). Also note that unlike field_math, you cannot affect the coordinates of the output.only the cell data.

Coordinate data passed to ports one, two, three and four are referred to as Ax, Bx, Cx and Dx (y, or z) in the appropriate mathematical expression. For example if you want to refer to X coordinate of the first input you would use Ax in the expression, if you want to refer to Z coordinate of the second input you should use Bz in the expression.

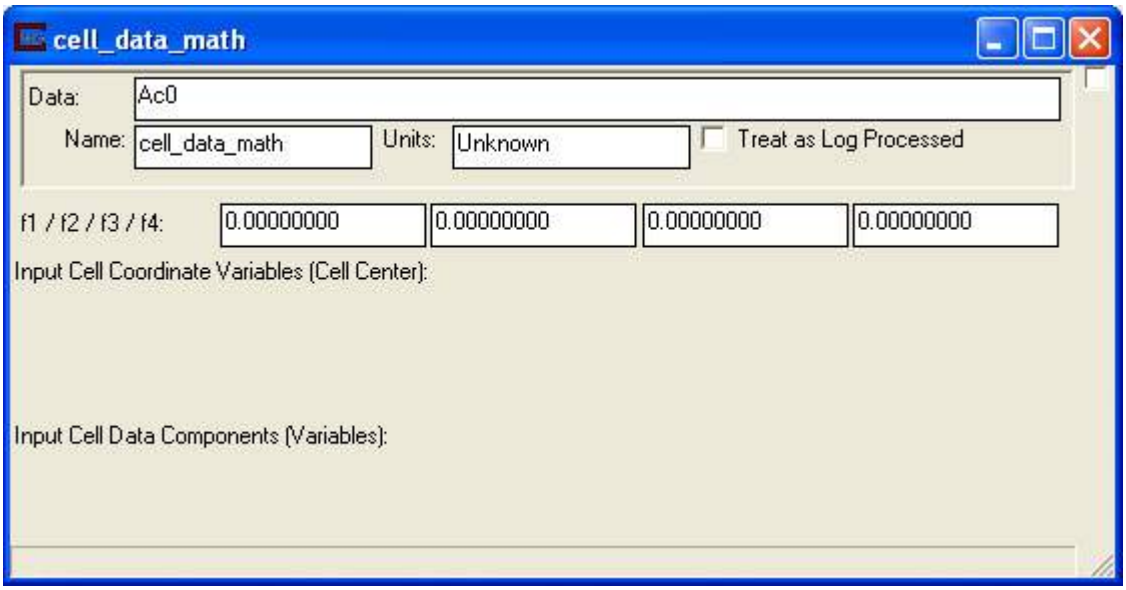
For example if you want to refer to third cell data component of the first input you would use Ac2 (numbers start at zero) in the expression, if you want to refer to first cell data component of the second input you should use Bn0 in the expression.

There are also four identical input ports that accept floating point numbers (variables). None of these are required to be used, but up to four can be used. The first of these ports is the fifth from the left and the ports are numbered sequentially in ascending order to the right. These **variables** are referred to as f1, f2, f3 and f4 in your mathematical expressions.

Module Output Ports

Cell_data_math has six output ports. The first four output port (from the left) pass the four variables so they can be shared with other modules. The next

port is the output mesh containing revised cell data that are a result of the computation. The last (red) output port is renderable version of the output mesh.



Module Control Panel

The control panel for cell_data_math is shown in the figure above. The Data type-in box is used for inputting mathematical expressions. The result of this expression will added to the input field as a data component. The name and units of that component can be entered in the fields below the Data box. The treat as log processed toggle should be selected if the newly created data component is in log space.

An example of an equation to blend the values of (interpolate between) two different fields having cell data with a logarithmic distributions as f1 ranges from 1 to 0.0 is:

log10(pow(10,Ac0)*f1+pow(10,Bc0)*(1.0-f1))

You can also use

interplog(Ac0, Bc0,f1)

To perform interpolation between non-log processed data use:

Ac0 * f1 + Bc0 * (1.0-f1) OR interp(Ac0, Bc0,f1)

Mathematical Operators

Related Modules

->coordinate_math

cell_to_node

General Module Function

The cell to node module is used to translate cell data components to nodal data components. Cell data components are data components which are associated with cells rather than nodes. Nearly all modules in EVS currently

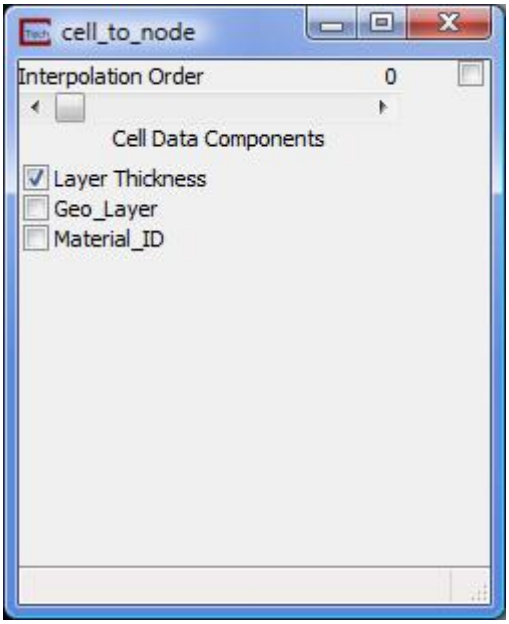
support only node based data. Therefore, cell to node can be used to translate cell based data to a nodal data structure consistent with other EVS modules.

Module Input Ports

Cell to node has only one input port which accepts an unstructured mesh with cell based data. There can be several data components, either vector or scalar, in the cell based data field.

Module Output Ports

Cell to node has two output ports. The first port (closest to the left) outputs a reference to a merged object that contains the new nodal data, plus the existing unstructured mesh and cell based data.



Module Control Panel

The control panel for cell to node is shown in the figure above. The slider is used to set the order of the interpolation algorithm. The default is 0 which stipulates a simple averaging of all adjoining cells. For values of 1 to the maximum of 32, a distance weighting function is used. Radio buttons are used to select which cell based data component is to be interpolated to the nodes.

Related Modules

->extract_cell_comp

select_cells

General Module Function

select_cells provides the ability to select individual geologic layers for output. If connected to Explode_and_Scale multiple select_cells modules will allow selection of specific geologic layers for downstream processing. One example would be to texture map the top layer with an aerial photo after one

select_cells and to color the other layers by geologic layer with a parallel select_cells path. This can be accomplished by multiple Explode_and_Scale modules, but that would take much more memory.

Module Input Ports

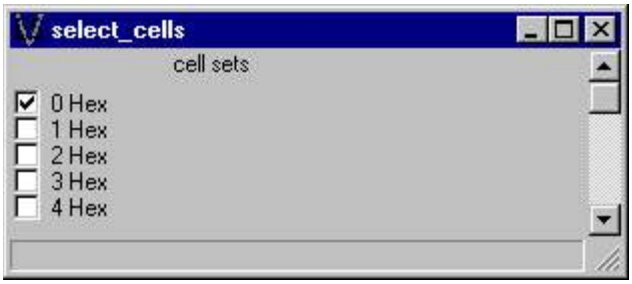
select_cells has only one input port which is an EVS field. Typically, this should be from Explode_and_Scale.

Module Output Ports

select_cells has two output ports. The first output port (closest to the left) outputs a subset of the input field based on which cell_sets (geologic layers) are chosen. The second port outputs a renderable geometry directly to the Viewer, but only if the input is 2D which is not typical.

Module Control Panel

The control panel of select_cells is shown in the figure below. The check boxes are used to select which geologic layers will be passed through the module.



shrink_cells

General Module Function

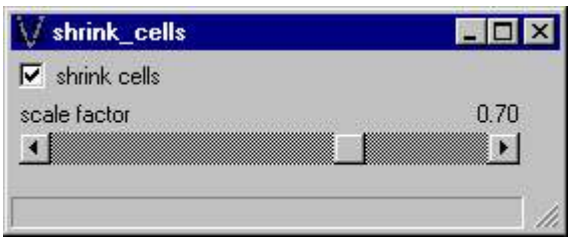
The shrink_cells module produces a mesh containing disjoint cells which can be optionally shrunk relative to their geometric centers. It creates duplicate nodes for all cells that share the same node, making them disjoint. If the shrink cells toggle is set, the module computes new coordinates for the nodes based on the specified shrink factor (which specifies the scale relative to the geometric centers of each cell). The shrink factor can vary from 0 to 1. A value of 0 produces non-shrunk cells; 1 produces completely collapsed cells (points). This module is useful for separate viewing of cells comprising a mesh.

Module Input Ports

shrink_cells has one input port. Data passed to this port can be any type of mesh.

Module Output Ports

shrink_cells has two output ports. The leftmost output port creates a new mesh that contains disjoint cells shrunk relative to their geometric centers. It also contains a Node_Data and Cell_Data if present in the input mesh. The second output port sends a renderable version of the output field to the viewer.



Module Control Panel

The control panel for shrink_cells is shown in the figure above.

The shrink toggle. If off results in an output mesh which contain disjoint cells with the same coordinates as the input mesh. If set on, the module computes new coordinates for the nodes based on the shrink factor that specifies the scale relative to the geometric centers of each cell. The default is on.

The shrink_factor slider is a float slider to adjust the sizes of the cells. The shrink factor value specifies the scale relative to the geometric centers of each cell. The shrink factor value can vary from 0 to 1, 0 producing non-shrunk cells and 1 producing completely collapsed cells (points). The default is 0.3. The range is 0.0 to 1.0.

cell_centers

General Module Function

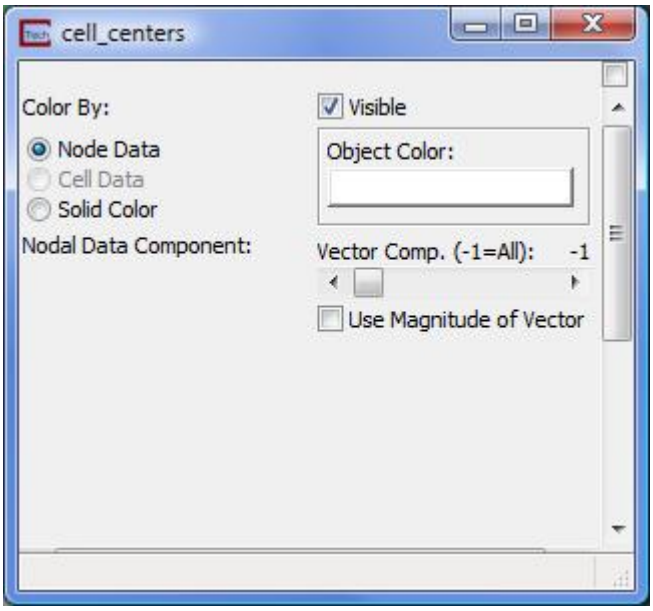
cell_centers module produces a mesh containing Point cell set, each point of which represents a geometrical center of a corresponding cell in the input mesh. The coordinates of cell centers are calculated by averaging coordinates of all the nodes of a cell. The number of nodes in the output mesh is equal to number of cells in the input mesh. If the input mesh contains Cell_Data it becomes a Node_Data in the output mesh with each node values equal to corresponding cell value. **Nodal data is not output directly.** You can use this module to create a position mesh for the glyph module. You may also use this module as mesh input to the interp_data module, then send the same nodal values as the input grid, to create interpolated nodal values at cell centroids.

Module Input Ports

cell_centers has one input ports which can contain any type of mesh.

Module Output Ports

cell_centers has two output ports. The first output port (closest to the left) outputs a new mesh that consists of points representing geometrical centers of the corresponding cells in the input mesh. It also may contain a Node_Data that corresponds to Cell_Data in the input mesh. The second port outputs a renderable version of the output field



Module Control Panel

The user interface for this module is shown above.

interactive_labels

General Module Function

The incteractive_labels module allows the user to place formatted labels at probed locations within the Viewer. The data displayed is whatever data is visible at the

Module Input Ports

1) in_view (Purple) : This port takes input from the Viewer.

2) z_scale (Grey-Brown) : This port takes the z exaggeration factor.

3) f1 (Dark Brown) : This port takes a float value that can be used in the formatted label.

4) s1 (Greyish Blue) : This port takes a string value that can be used in the formatted label.

Module Output Ports

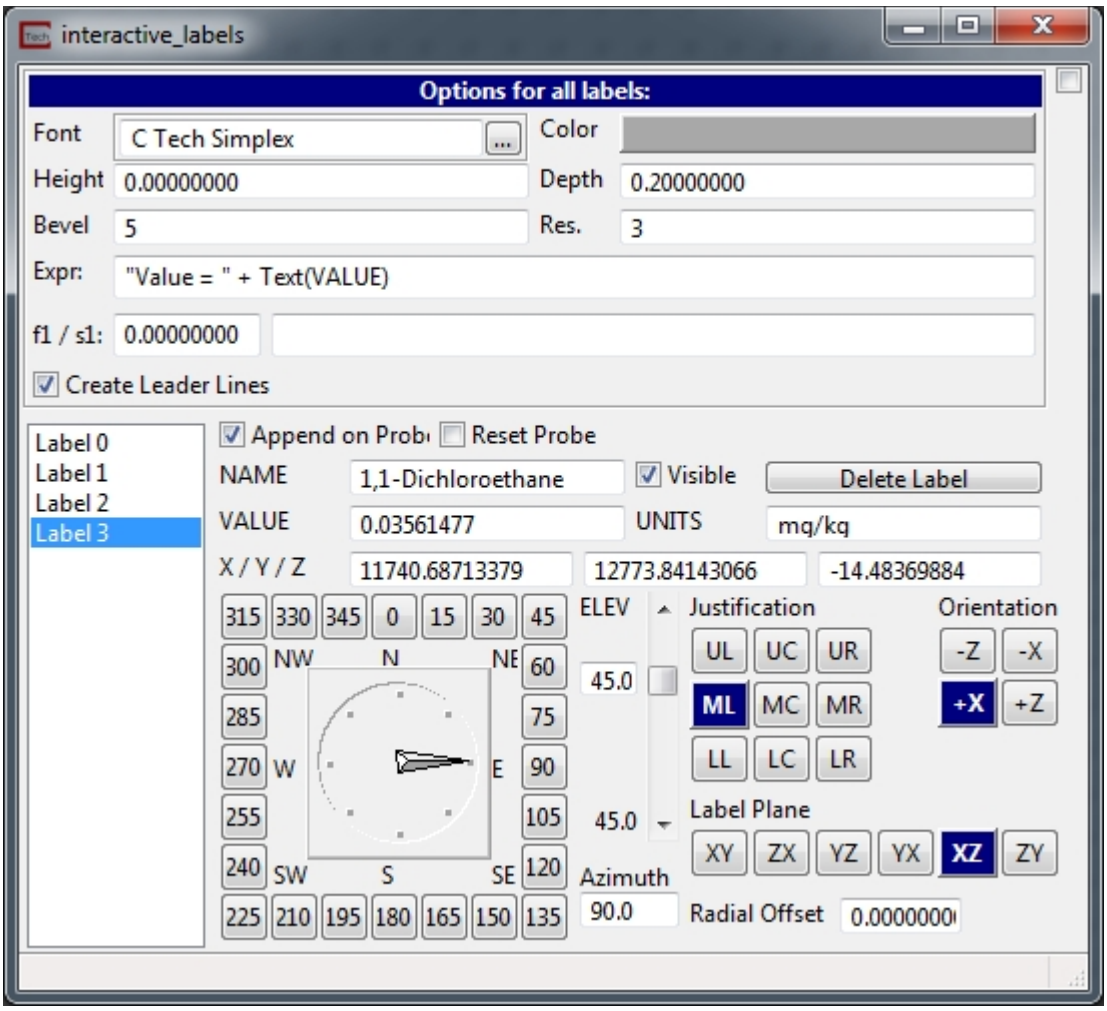
1) z_scale (Grey-Brown) : This port outputs the z exaggeration factor.

2) f1 (Dark Brown) : This port outputs the float value passed in via the input port.

3) s1 (Greyish Blue) : This port outputs the string value passed in via the input port.

4) out_obj (Red) : The port outputs a group renderable labels to the Viewer.

Module Control Panel



The Options for all labels window defines a group of parameters that are applied to all labels. These parameters cannot be set differently for individual labels.

The *Font, Color, Height, Bevel*, and *Res.* fields are the common font options for all labels.

The *Exrpr:* is the format string for the labels. The format string allows the same input as the string_format module with the additional variables of: NAME; VALUE; UNITS; X; Y; and Z. These additional variables correspond to the data values at the probed location.

The *f1* field is a float field that can be passed in and allows for the f1 variable to be used in the expression.

The *s1* field is a string field that can be passed in and allows for the s1 variable to be used in the expression.

There are many options that can be set for individual labels. The list box to the far left lists all of the labels created and the current label being edited.

The *Create Leader Lines* toggle will a leader line from the probed point to the label to be created.

The *Append on Probe* toggle will cause an additional label to be created when the Viewer has been probed in using the Alt + Left Mouse button.

The *Reset Probe* toggle when selected will allow the user to change the probe location of the label currently being edited.

The *NAME, VALUE, UNITS, X, Y*, and *Z* fields are populated based upon the displayed data at the probed location. Changing these values will not affect the data at the probed location.

The *Visible* toggle will turn on or off the currently selected label.

The *Delete Label* button will remove the currently selected label.

The *Azimuth* and *Elev* fields will rotate the label around the probed location.

The *Radial Offset* field allows the user to change the distance from the probed location to the label. There will be a line connecting the label to the probed location.

Justification, Orientation and the *Label Plane* are all label alignment options.

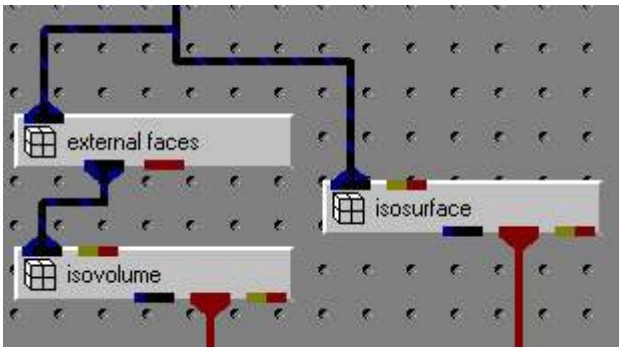
_3D_Plume

This is a deprecated module whose function has been superceded by plume_shell

General Module Function

The 3D_Plume module (previously named IsoVolume) is identical to a subnetwork containing the isovolume module with constant_shelland external_faces. This module creates a superior visualization of a plume that can be sent directly to the viewer for rendering. This is not a subsetting module (as is isovolume or contour). It is used exclusively for plume visualization of 3D fields (such as the output of Krig_3D).

The figure below shows a module network showing the three individual modules that make up the 3D_Plume module.

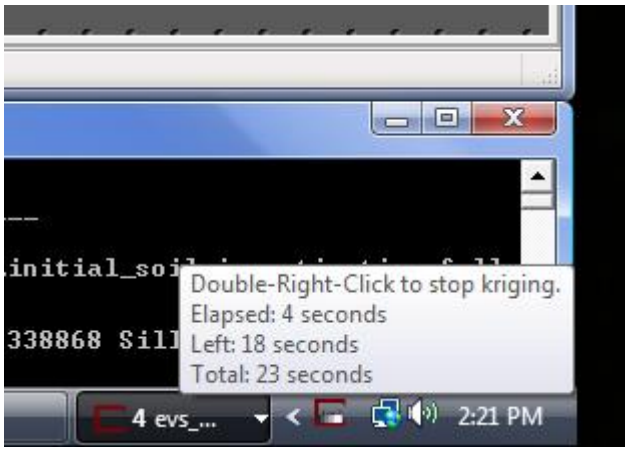


Module Input Ports

3D_Plume has two input ports. The leftmost port accepts unstructured mesh data. The second port provides a means to share the subsetting level of other modules.

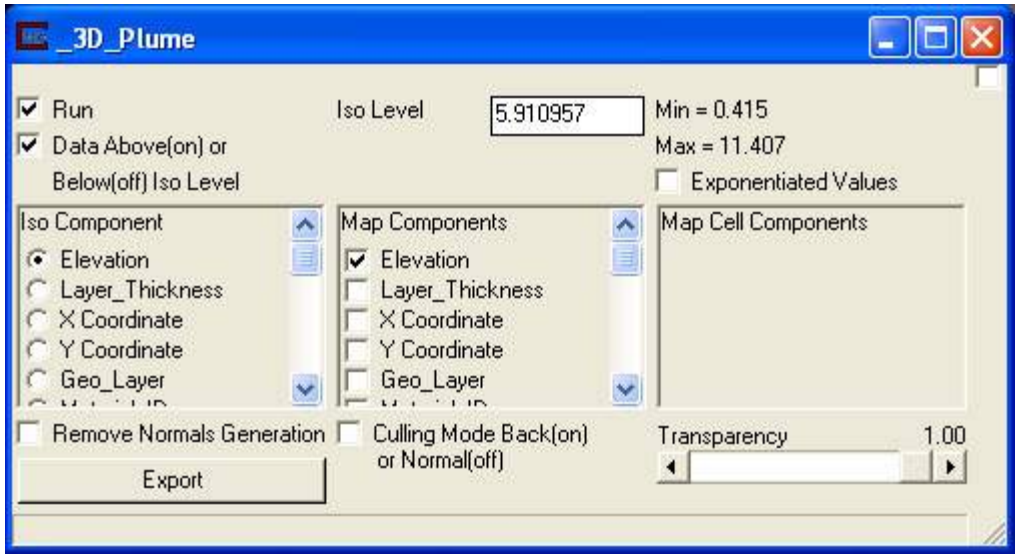
Module Status: **Interruptible**

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Output Ports

3D_Plume has three output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains cells representing the external faces portion of the 3D_Plume. Nodal data sent to this output port consists of the nodal data of the map component in the isovolume portion of the subnetwork. This blue output is used to input to isolines to place isolines on the portions of the plume that intersected the exterior surfaces of the input domain. The second port outputs a rendered geometry directly to the Viewer. The third output port provides a means to share the subsetting level of this module with others.



Module Control Panel

The control panel for 3D_Plume is shown in the figure above.

Iso Components refers to the model data component used to create the subset of the original input field. When an iso component is selected, the min and max values of the variable are displayed in the lower right corner of the dialog box. The default iso component is the first (0th) component in the column.

Map Components determines which model data components in an unstructured mesh will be sent to the leftmost output port. Also, the first map component selected will be used to color the output. For example, choosing iso component concentration and map component uncertainty will create a volumetric subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The ***Map Cell Components*** option box selector lets you map cell data (if any) to the surfaces output by 3D_Plume. This is markedly faster than using interp_cell_data. The cell data will only be visible if all nodal data is unselected (under *Map Components*).

The ***Iso Level*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default iso level value is the arithmetic average of the minimum and maximum values in the iso component. If your input data has been kriged with log processing, the values here will be the Log of your input data.

The ***Exponentiated Values*** toggle makes another type-in field visible which convert real units to logarithmic units for you automatically.

The ***Run*** toggle when not selected will prevent the module from running.

The ***Data Above(on) or Below(off) Iso Level*** check box (Above Box) is used to display data above the iso level or below the iso level. For example, to display a volumetric subset of all concentrations greater than or equal to 1 ppm, set iso level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the 3D_Plume of 1 ppm and below, simply turn the Above Box off (no check in box).

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Culling Mode Back(on) or Normal(off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

The ***Opacity*** slider changes the opacity (opposite of transparency) of the entire 3D Plume Group.

The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. The additional ports represent:

a. Iso_Component name: This is a string with the name ot the selected data component.
b. Above/Below: This is a string containing the word "Above" or "Below" depending on the state of the toggle above.
c. Iso_Level: This is a real number representing the subsetting level.
d. Exponentiated Iso_Level: This is a real number representing the exponentiated subsetting level.

Related Modules

->isosurface

-> isolines

-> isovolume

solid_contour

This is a deprecated module whose function has been superceded by contour_data

General Module Function

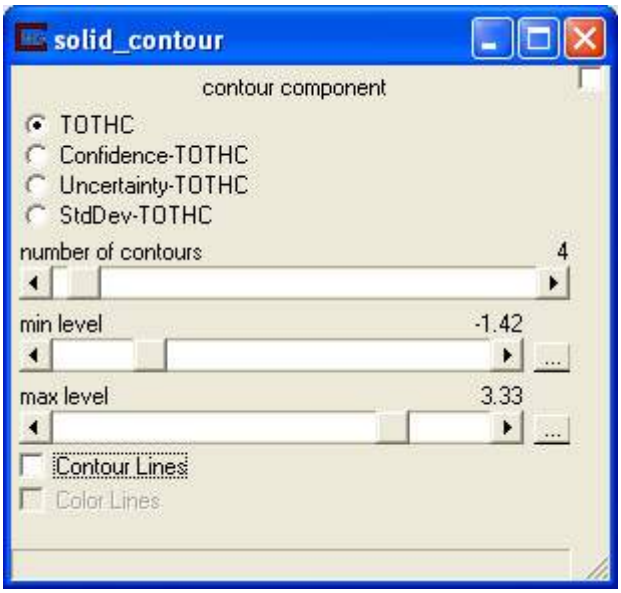
The solid_contour module has identical functionality to Solid_3D This module is used for applying solid contours. The user may the number of solid contours to map, however the contour spacing will be an even incremental spacing between the subsetting level and the max/min value in the mesh. If you wish to apply solid contours with user specified contour intervals, you must use Solid_3D_Set or solid_contour_set.

Module Input Ports

solid_contour has one input port which accepts unstructured mesh and nodal data.

Module Output Ports

solid_contour has four output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains cells representing the external faces portion of solid_contour. This port contains no nodal data, only cell data that corresponds to the nodal data of the input. The second blue/black output is the isolines that delineate the boundaries between the solid contours. The third port is the solid contours and connects directly to the Viewer. The fourth port is the isolines and connects directly to the Viewer.



Module Control Panel

The panel for solid_contour is shown above. The parameters are identical to that of plume_shell parameters with an additional slider for setting the number of solid contour levels. The slider choice will result in *n* number of contours at even increments between the min and/or max subsetting level. If the even spacing is not desirable, the module solid_contour_set module can be used to set un-evenly spaced contour levels.

Note that the coloring of each solid region is based on an average of the bordering contour levels. If this is objectionable try selecting the Solid_3D object in the Viewer, then choose datamap_editor-->Options-->Edit Range/Data-->then adjust Range Size slider to match the number of contours. This should solve any averaging issues with Solid_3D.

If Contour Lines is selected a isoline will be added at each of the contour boundaries.

If Color Lines is selected the countour lines will be colored according to data.

Related Modules

-> isolines

-> Solid_3D_Set

contour

This is a deprecated module that has been supplanted the intersection module or the use of two plume_volume modules.

General Module Function

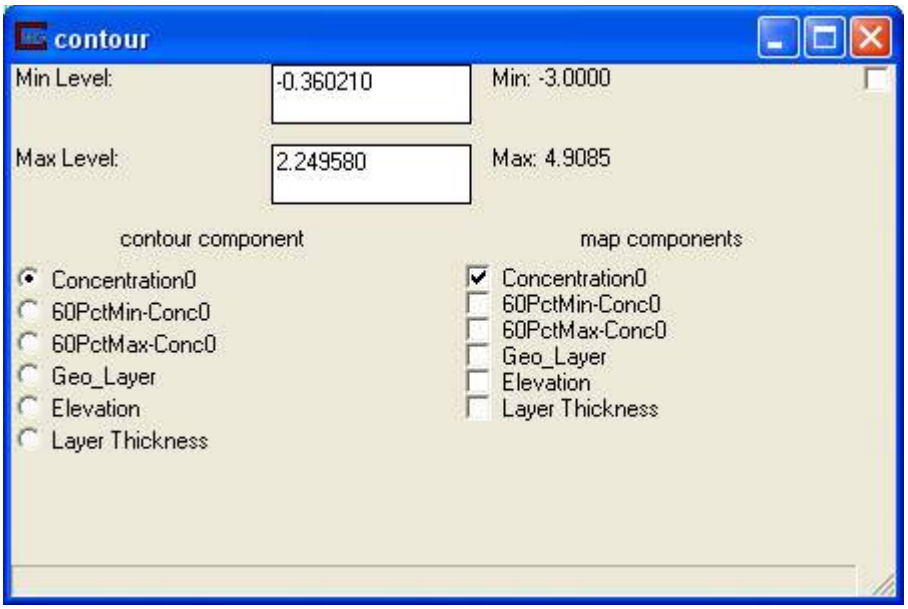
contour is similar to plume_volume in that it subsets both 2D and 3D input fields, but it provides the additional ability to specify minimum and maximum contour levels and outputs only those regions falling in between.

Module Input Ports

contour has three input ports. The leftmost port accepts unstructured mesh data. The other ports provide a means to share the subsetting levels of other modules.

Module Output Ports

contour contains four output ports. The first output port (far left) outputs a new unstructured mesh which contains cell type tri(angle) representing the external faces of the contour. Nodal data sent to this output port consists of the nodal data of the map component within the contour level. The second port outputs a rendered geometry directly to the Viewer. The third and fourth output ports provide a means to share the subsetting levels of this module with others.



Module Control Panel

The control panel for contour is shown in the figure above. The left column, labeled contour component, consists of a series of radio buttons representing all available model data components. The right column, labeled map components, is a series of check boxes next to the same model data components. User input sliders labeled min level and max level are displayed above the contour component radio buttons. At the right ends of the sliders the minimum and maximum values for the data selected under iso component are displayed.

Iso component refers to the nodal data used to create the data subset. When an iso component is selected, the min and max values of the variable are displayed at the right end of the min and max level sliders. The default iso component is the first (0th) component in the column.

Map component determines which data components will be sent to the leftmost output port in an unstructured mesh format. Note that the first map component selected will be used to color the contour, regardless of how many others are selected. For example, choosing iso component concentration and map component uncertainty will create an contour of

concentration colored by uncertainty. By default, only the first map component is selected.

The min and max level sliders are used to define the iso component value at which the subset defined by contour is to be drawn. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default level values are 1/3 and 2/3 of the way between the minimum and maximum values for the selected iso component.

Related Modules

-> **isolines**

-> **plume_volume**

draw_3D_lines

This is a deprecated module that has been supplanted by draw_lines

General Module Function

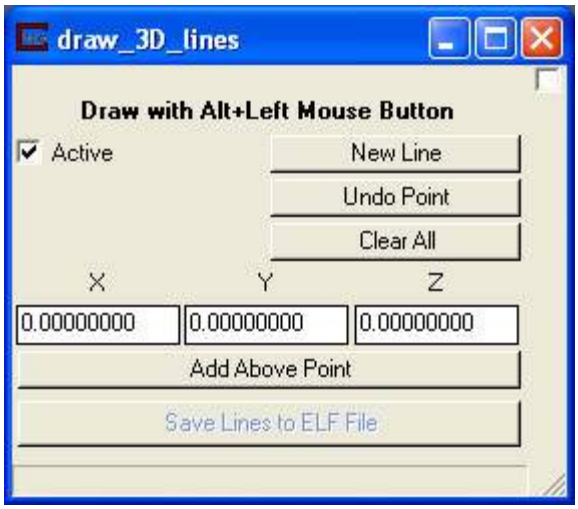
The draw_3D_lines module enables you to create a 3D drawing with individual clicks of the mouse. The default mouse gesture for line creation is: depress the alt key and then click the left mouse button on any pickable object in the viewer. The first click establishes the beginning point of the line segment and the second click establishes the each successive point. draw_3D_lines allows adding of points that are outside the model extents, undoing of the last picked point, and the clearing of all picked points. Unlike most modules which create mesh data to used by other modules, the draw_3D_lines module receives input from the viewer, and also passes on field data to be used by other modules.

Module Input Ports

The draw_3D_lines module has one input port. This port is connected to the output port of the viewer within which the line will be drawn.

Module Output Ports

The draw_3D_lines module has two output ports. This first output port sends a mesh to downstream modules. The second output port sends the line as a renderable object to the viewer to be rendered.



Module Control Panel

The draw_3D_lines user interface allows interactive creation of points or lines onto any connected object and then displays the points or lines in the viewer.

The Active toggle turns on and off the functionality of the module, letting you Alt+Left Click on an object in the viewer without drawing a line.

The New Line button is used to if multiple lines are desired. It should be used when a line is finished and the user wished to begin a new line. It is not needed to start a line.

The Undo Point button will undo the last entered point of the line.

Clear All will remove all points from all lines.

The X, Y and Z type ins by default reflect the last point entered into the line. If the user wants to add a point by hand the coordinates may be typed in and added to the line by pressing the Add Above Point button. If the user is entering a coordinate by hand they are not limited to using one within the extents of the model in the viewer.

The Save Lines to ELF File allows the user to save all of the created lines in the EVS Line File format (*.elf).

Map_Spheres

This is a deprecated module that has been supplanted by post_samples

(This module has some features available only in EVS PRO and MVS)

General Module Function

The Map Spheres module is used to visualize sampling locations and the values of the properties in .apdv files, or the lithology specified in a .geo file, along with a representation of the borings from which the samples/data were collected. Map Spheres has the capability to process property values to make the posted data values consistent with data used in kriging modules, to post spheres, or square or hexagonal discs at the sampling locations that are colored and sized according to the magnitude of the property value, and to

label the sampling locations with several different types of information. An interpolation functionality is now available with Map_Spheres whereby the user may calculate a linear interpolation (range 0 to 100 percent) between two consecutive data components. This functionality is useful for creating animations of time-series data with Map_Spheres. Mouse interactive querying of data is now available with Map_Spheres by clicking the left mouse while holding the alt key.

Map Spheres can also represent downhole geophysical logs or Cone Penetration Test (CPT) logs with colored and sized tube diameters according to the magnitude of the data. Map Spheres can display nonvertical borings and data values collected along their length, and can also explode borings and sample locations to show their correct position within exploded geologic layering. When used to read geology files, map spheres will place sample indicators at the top surface of each geologic layer, that are colored according to the layer they are depicting and/or will color the borings by lithology.

Module Input Ports

Map Spheres has four input ports. The first (leftmost) port can only be connected to Krig 3D Geology, which provides the geologic layering information that allows the boreholes and spheres to be exploded (This feature available only in EVS PRO and MVS) into their appropriate geologic layers. The second, and third rightmost ports can be connected to explode and scale, which provides the explode and scale factors, respectively. If geologic layering is not being modeled, then only the right port needs to be connected to Explode and Scale to provide the scale factor for displaying the boreholes and spheres.

The Yellow-Blue-Orange port (second from left to right) allows the sharing of analyte (e.g. chemistry) or Geology file names between similar modules. This should simplify the task of specifying data file names common to multiple modules in your application. The output ports come in 3 color varieties:

Module Output Ports

Map Spheres has five output ports:

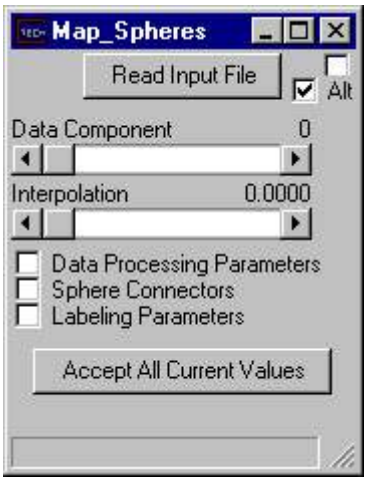
The leftmost Yellow-Blue-Orange port allows the sharing of analyte (e.g. chemistry) or Geology file names between similar modules.

The second (gray-green) port is the explode factor.

The third (gray-brown) port is the Z-exaggeration factor.

The fourth (red) port outputs a geometry that can be input to the Viewer for rendering.

The last (blue-black) port outputs the bounds of the data field to the Generate Axes module, which provides automatic placement of the axes in the scaled or unscaled data field.



Module Control Panel

The control panel of Map Spheres is shown in the figure above. The **Read Input File** button opens a *File Browser* which lists the *.geo, *.pgf, or *.apdv files that are present in the current directory shown in the directory window. The format of .apdv files is described in the apdv file format help topic. The format of .geo files is described in the geo file format help topic.

Note that this module will not begin running until a valid chem file has been selected and the **Accept All Current Values** button is pushed.

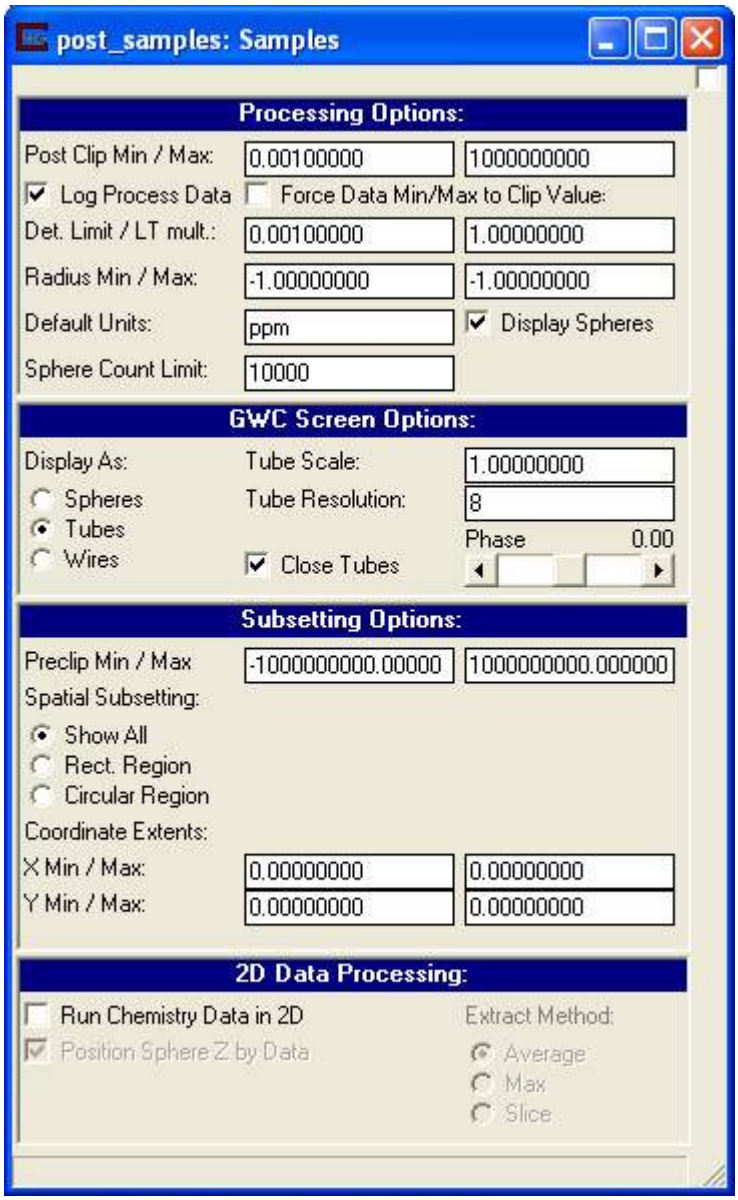
To display nonvertical boreholes, the only requirement is that samples from any given borehole have the same location name, with the actual coordinates of each sampling location. The Data Component slider allows the user to select which of the property values in a .apdv file will be displayed by Map Spheres. For example, when reading CPT data the user may choose either tip resistance, sleeve friction or friction ratio using the slider. When reading contaminant data the user may choose among the various contaminants listed in the file. The default value is 0, which selects the first value of 15 that can be present in the file. When reading .geo files, the data for all of the geologic layer tops will be read and displayed when the module is executed. When the Data Component slider is set to 1 for .geo files, the tubes and spheres will be colored by the geologic material color ID specified in the .geo or .gmf file.. Note that this module will not begin running until a valid input file has been selected and the Accept All Current Values button is pushed.

The Interpolation slider performs a linear interpolation between two consecutive data components The selected data component is considered as the first and the next consecutive data column in the .apdv file is second. For example, if the data component slider is set to 0, an interpolation slider setting of 0.25 would produce a value one-quarter of the way between data component #0 and data component #1.

Module Parameter Subpanels

Map Spheres has three subpanels, which allow the user to set the parameters used for data processing the input data, to specify the type of connector to be used to represent a boring, and to specify the types of

labeling that will be displayed. Clicking on either the check boxes next to the subpanel names, or on the names themselves will bring up the subpanel parameter screens. Note that the subpanels must be closed by clicking on the box in the **Module Control Panel**. They cannot be closed by double clicking on the subpanel's window control icon (the small horizontal bar in the upper left corner of the window).



The Data Processing Parameters subpanel is shown in the figure above. It is important to note that all data processing actions are applied directly to the data in memory, and that the original data file is not altered. Because Map Spheres only outputs a data field to Generate Axes, or a geometry to the viewer, the data processing within Map Spheres applies only to the Sample indicators or tube coloring in the display. When setting data processing parameters in Map Spheres, the user should refer back to the preprocessing and/or postprocessing parameters set in any other modules that will

contribute to the final display to assure that consistent parameters are being used (i.e., to correctly specify whether the data has been log transformed, scaled, and/or clipped). This is particularly important when a color scale is being used to display parameter value distributions on an isosurface and spheres in the same display.

The Data Processing Options radio buttons (below and to the right of the Explode Distance input field) allow the user to specify whether the data will be used as is, or will be processed to compute the log base 10 of the parameter value before kriging. This parameter should be set to be consistent with the data processing that is being performed by the kriging or other modules that are contributing to the display.

The number entered into the Clip Min input field will be used during data processing to replace any sample property value that is less than the specified number. The default value for Clip Min is 0.001, but the user can enter any number. The number entered into the Clip Max input field will be used during preprocessing to replace any nodal property value that is greater than the specified number. The default value for Clip Max is 1000000.00, but the user can enter any number less than your data maximum to adjust the color mapping by clamping.

The *Reset MinMax to Above Values* toggle uses the Clip Min and Clip Max values to set clamping and to reset the Min and Max values used for coloring. This allows multiple datasets or data measured at different times to have exactly the same datamaps for coloring.

This is accomplished by adding 2 invisible spheres (having zero radius) to your dataset located inside of one of the existing points. This can create a problem if the output of Map_Spheres is connected to other modules such as glyph. It this case we recommend NOT using this toggle and using the set_minmax module between Map_Spheres and glyph.

Note that setting the clip max outside your data maximum will not shift the datamapping outside your data range unless the above toggle is on.

Note that if the log10 of the data is taken, the Clip Min value must be used to replace any zero values in the data with a specified Clip Min value that is greater than zero, to eliminate possible errors associated with attempting to take the log of 0 or negative number (which is undefined). If duplicate data values exist in .apdv files (data points with exact X, Y, and Z coordinates), these values are averaged to produce one data point for each unique coordinate.

The Det. Limit refers to the detection limit used when creating the input file. Any non-detect flag in the file (please see the help file for the file type being read in for a list of non-detect flags) will be replaced by the Det. Limit value.

The LT mult field is used to input the Less Than Multiplier. This value is used whenever the '<' character preceeds a file value. The file value will be replaced by the product of the file value and the LT multiplier.

The Rmin / Rmax input fields are used to specify the minimum and maximum radii of the spheres (or other sample indicators in user units), that will be used to scale the sample indicators to the value of the property being

represented. The scaling procedure considers the extent in X-Y of the domain of the .apdv file being read by Map_Spheres, to produce sample indicators that are easily seen on the display. If the X-Y domain is less than 100 user units, than the Rmin and Rmax parameters specify the actual radii of the sample indicators in user units. If the X-Y domain is greater than 100 user units, then Rmin and Rmax specify the radius of the sample indicators as a percentage of the total extent of the domain. As an example, if the X-Y domain in the .apdv file is 100 feet, than Rmin and Rmax values of 10 will produce sample indicators that have a constant 10 foot radius. If the domain is 1000 feet, than Rmin and Rmax values of 10 will produce sample indicators that are 100 feet in radius. The default values for Rmin and Rmax are both 1.0. If different values are specified for Rmin and Rmax, then the sample indicators will be sized according to the data values which are normalized to a scale ranging between Rmin and Rmax units (either in user units, or a percentage of the total X-Y domain extent as described above). The user can specify any values for Rmin and Rmax that provide the display desired. When posting geophysical logs or CPT logs (with apdv files), the Rmin and Rmax are commonly adjusted to different values to display the relative magnitude of the deflections along the borehole trace.

The Zscale parameter specifies a multiplier by which all Z coordinate values in the data input set will be multiplied before displaying. This parameter is used to incorporate vertical exaggeration into the display, and should match the scaling parameters used in other modules that will contribute to the display. The default value of Zscale is 5, but the user can enter any value that will provide the display desired. Note that if the right port of Map Spheres is being connected to Explode and Scale, the Zscale parameter is overridden by the Z Exaggeration factor being passed to Map Spheres from Explode and Scale.

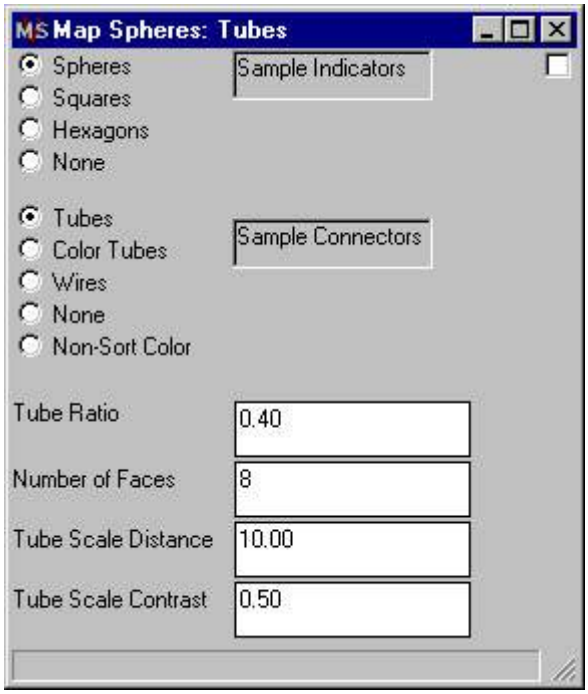
The Top parameter specifies the elevation at which all tops of the sphere connectors will be uniformly placed. Note that if the boring top elevations are specified in the .apdv file, then this parameter is not used, and the sphere connectors will be placed at the actual top surface of the model. A discussion of how to use either depths or elevations for the top coordinates of boreholes is provided in the apdv file format section of the Help System. However, if the site has relatively little topography, and the user wishes to quickly prepare a display with posted sample indicators and borings, then this parameter can be used to set a uniform top elevation for all borings. The default value is 0, which will work well with .apdv files that only specify the depth to each sample. The user can use any value for Top that will provide the display desired.

The Explode Distance parameter (This feature available only in EVS PRO and MVS) specifies the distance in user units that the sample indicators (and connectors) will be exploded at geologic layer boundaries supplied by Krig_3D_Geology. Note that if the second port of Map_Spheres is connected to the Explode and Scale module, then the explode distance will be passed to Map Spheres. It can be overridden by entering a different value for the Explode Distance in this panel if there is no connection.

The Extents Display Options radio box provides a means to subset the display of your sample data. The default is all which will display all of your data. When Rectangular Region (shown above) is selected four type-in fields become visible and can be used to specify a rectangular region within which data will be displayed. Similarly, Circular region causes three type-in fields to become visible and can be used to specify a circular region (x-y center and radius) within which data will be displayed.

The Indicator Display Options radio box also provides a means to subset the display of your sample data. The default is all which will display all of your data. The other options define range domains within which data will be displayed. Note that these ranges apply to the only the sample results and should not be confused with the range of data extents (coordinates) which are adjusted using the Extents Display Options.

The Data Extents display at the bottom provides a display of the extents of the entire dataset.

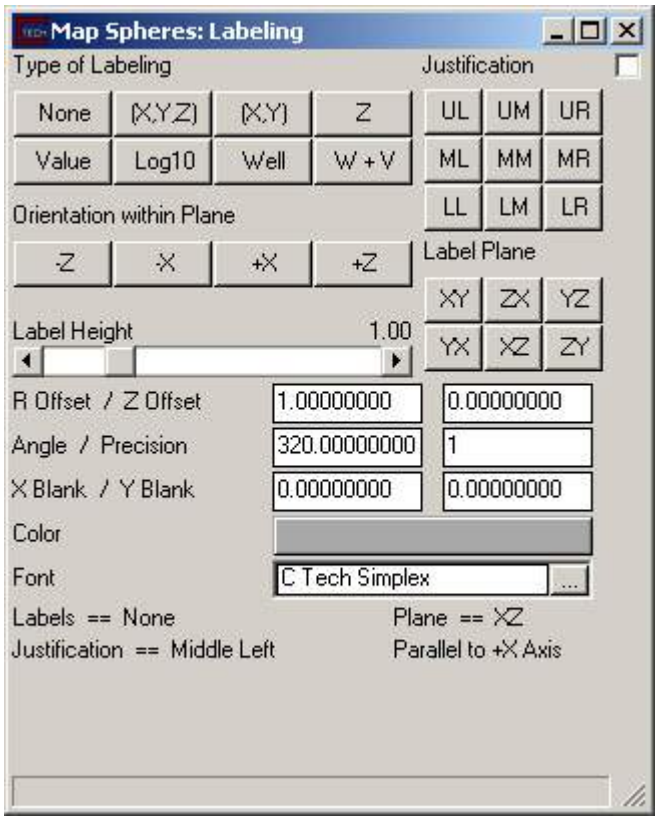


The Sphere Connectors subpanel is shown in the figure above. It provides means to specify the Sample Indicator and the Type of Connection via radio buttons which allow the user to select: Spheres, Squares, Hexagons, or None for glyphs located at each sample location, and whether: Tubes, Colored Tubes, Wires, or None will specify the items in the display to connect the posted indicators, or whether no connection will be used. Note that for large and/or tightly spaced (downhole) data sets, the spheres should be turned off, and only colored tubes should be used. See the Module Hints at the end of this help section for more details.

NOTE: The non-sort toggle is currently functional, but unsupported. One requirement with the non-sort is that the CSV file must contain a "name" column and a "top" column, or a crash will result.

The Tube Ratio parameter sets the ratio of the radius of the tube to the radius of the spheres. For example, if Rmin of the spheres is 8, and the Tube Ratio is set to 0.5, then the tube radius displayed will be 4 user units. The default value of 0.4 generally produces good results, but the user can input any value from 0 to 1. The Number of Faces parameter applies only to Tube connectors, and specifies the type of geometry that will be displayed. The default Number of Faces is 8, which produces a fairly smooth sided cylinder. If a value of 3 was used, then the connectors would represent a triangular tube. This value should also be adjusted when trying to reduce memory overhead during processing. See the Module Hints at the end of this help section for more details.

The Tube Scale Distance parameter specifies the length (in user units) along the tube at which the tube will be displayed with contrasting colors. This parameter allows a direct indication of depth down the tube to be displayed. The default value of 10 gives the tube alternating colors in 10 user units increments, but any user convenient value can be input by the user. The Tube Scale Contrast parameter specifies the change in hue of the tube between alternate scale distances. The default value of 0.5 produces tubes which alternate from gray to black at the specified scale distances. A Tube Scale Contrast of 1 will produce colors which alternate from light gray to dark gray, and a value of 0 will make the entire tube medium gray.



The Labeling Parameters subpanel is shown in the figure above. This panel allows the user to specify which types of information will be displayed as labels at each boring and sample location. The current labeling parameters that have been set are listed along the bottom of the subpanel. The Type of

Labeling push buttons are used to specify which information will be displayed next to the boring or sphere. Each button specifies a certain type of value that will be displayed. The top row of buttons provide the type of coordinate information to be displayed. The default value is to display the well name next to the top of the boring location, but the X, Y, or Z coordinates (and combinations thereof) can also be displayed at the top of the borehole or next to the spheres. The first two buttons on the bottom row specify whether the property value, or the Log 10 of the property value will be displayed. Note that the property values displayed **do not** reflect any clipping that was done on the data. The Well button displays the well name at the Top of the boring (the default). The W+V button displays the Well Name at the top of the boring, and the Value of the property next to each sphere.

For more information on Font Selection Click Here.

Because all of EVS's displays are in three dimensional space, the user has to specify which plane the label text strings will be written onto, what direction they will be written, and the justification to be used to allow the text to be viewed correctly from a given perspective. The justification push buttons are used to set the location of the label relative to the center of the sphere to be labeled. The layout of these push buttons corresponds to the location of the sphere relative to the label location. The label can exist in any of nine locations relative to the sphere. For example:

- Selecting MC (middle center) the center of the label is placed directly over the center of the sphere.

- Selecting UC (upper center) places the center of the label directly below the center of the sphere (sphere is above the center of the label).

- Selecting LL (lower left) places the left end of the label just above the center of the sphere.

- Selecting MR (middle right) places the right end of the label directly to the left of the center of the sphere.

The other justification push buttons act in a similar manner.

The Label Plane push buttons define which plane the label is to reside in. There are 6 possible planes which correspond to the six faces of a box. The label planes are defined as being parallel to the axes listed in the push button name. The order of the axes in the plane names define the side from which the plane is viewed. For example, the XY label plane will produce labels in a plane parallel to the X and Y axes that can be viewed correctly from a viewpoint along the positive Z axis (the normal vector in a right handed coordinate system). The YX label plane will produce labels in the XY plane that can be viewed correctly from a viewpoint along the negative Z axis. Again, the order of the axis listing determines whether a label is forward and right side up when viewed from the negative or positive end of the third axis, relative to a right handed coordinate system.

The Orientation Within Plane buttons define which axis the labels should parallel within the label plane. Within any given label plane, there are four possible label orientations. For example, in the XZ plane the labels can be parallel to the +X, +Z, -X or -Z axis. A +X orientation will cause the labels to

parallel the +X axis in the XZ plane and read from left to right, right side up when viewed from along the +Y axis. A -X orientation creates labels parallel to the -X axis in the XZ plane, which, when viewed from along the +Y axis, will appear upside down.

The orientation of the label also affects how the Justification push buttons affect the label. As was described above, the Justification push buttons move the text around a label box relative to the center of the sphere. The label string can be thought of being contained in a box that is oriented parallel to the Orientation Within Plane push button selection. Therefore, if +Z orientation is chosen, the UC justification location appears to move the label to the middle left of the tick mark in Cartesian coordinates, when viewed normal to the XZ plane. The LL will appear to move the label to the lower right of the tick mark when viewed normal to the XZ plane.

The user will need to experiment with the values of Label Plane, Justification, and Orientation to obtain the label justification desired. The best way to understand the effects of setting different Label Planes and Orientations with the planes is to experiment with posting a set of axes using the Generate Axes module along with Map Spheres, and trying different settings for the labels.

The Label Height parameter allows the user to specify the height of the label in user units. The default value is one, but any value between 0 and 5 can be selected.

The Angle parameter specifies the radial angle that the label justification point will be positioned at a distance of R Offset (see below) in the Label Plane. An angle of 0 places the label directly to the right of the sphere, and an angle of 90 places the label justification point directly above the sphere.

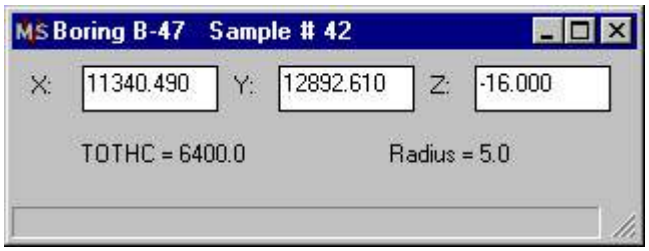
The R Offset parameter specifies the radial distance, in user units, in the Label Plane, that the label justification point will be offset from the sphere position. The Z Offset parameter specifies the distance, in user units, that the label string will be placed away from the central Z coordinate of the Sphere. Note that an appropriate value for this parameter should be greater than the Sphere Rmax value chosen, or else the label will appear within the sphere.

The Precision parameter specifies the number of decimal places that will be displayed in the label

The X Blank and Y Blank parameters specify the X and Y distances that must exist between adjacent labels, or else only the greatest of the overlapping labels will be posted. This parameter allows the user to avoid having many overlapping labels that are unreadable in the display.

The Gray parameter specifies what shade of gray will be used for the text of the label. A value of the default value of 128 produces a gray label, a value of 0 produces a black label, and a value of 256 produces a white label. The font specifies which of 6 available fonts will be used for the label text.

Interactive Querying of Your Data



Map_Spheres allows you to probe any sphere (.apdv, .geo, .pgf files) and determine information about that sample and its borehole ID. To probe a sample (spheres must be on), click on it with the Alt key and Left Mouse Button. The following window will appear in the upper right hand corner of your desktop.

X (Easting)

Y (Northing)

Z (Elevation)

Concentration (Attrib)

Radius

Further explanations of the various parameter settings and their effects on the visualization are presented in Workbook2.

Map_Spheres Module Hints

Using Map_Spheres with .geo files

When using Map Spheres to post lithologic layer data (.geo files), the user must NOT connect to/from Explode and Scale. This is due to competing functionality for posting chemical data with the exploded layers from Krig 3D Geology. Therefore if viewing lithologic layer data (.geo files), the user must simply read the .geo file on a path parallel to the Krig 3D Geology module, and then send that data to the viewer. The parallel network containing Krig 3D Geology and Explode and Scale MUST have explode distance set to 0.0.

The .geo files read into Map Spheres MUST contain boring names.

Using Map_Spheres with geophysical logs in .apdv files

When using Map Spheres to post tightly spaced downhole data such as geophysical logs, the user must turn Sample Indicators to None and Sample Connectors to Color Tubes. The Tube Ratio may be adjusted, but the output is directly related to the Rmin and Rmax values chosen in the Data Processing Parameters.

Map_Sphere's "Sphere Connector" panel also has a parameter that is "Number of Faces". This is how many sides the "cylindrical" tubes have. The default is 8 meaning that the tubes are actually octagonal. To keep the memory overhead of large files down, try setting it to a lower number such as:

2 gives a flat planar strip which is oriented in the "X-Z" plane

3 gives triangular tubes

4 gives square tubes, etc.

Read_netCDF

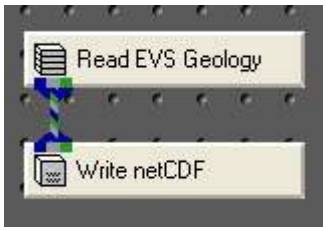
General Module Function

This is a deprecated module that has been supplanted by Load_EVS_Field

Read_netCDF reads a dataset from a netCDF file into an EVS field. If the netCDF file was written with the Write_netCDF module, then Read_netCDF outputs the same object that was written (a field or other EVS object). If the file contains only a standard netCDF object, without EVS header information, Read_netCDF produces an object containing the data elements of the netCDF file.

As of Version 6.0 the output of Krig_3D_Geology and Spline_Geology can (and should be) saved as a netCDF file using Write_netCDF.

We strongly recommend that you convert older .egf files to the newer format using the following simple process:



Module Input Ports

The Read_netcdf module is shown above. Read Field has no input ports. It obtains the Field input data by reading a file with a file browser.

Module Output Ports

Read_netCDF has two output ports. The first port (closest to the left) passes the mesh and data components to other modules which accept Field data types. The second port is used to send data directly to the viewer for rendering.

Module Control Panel

The user interface for Read_netCDF consists solely of the file browser shown above. The file browser is used for selecting the file. Double clicking on a filename or selecting a file name and choosing OK reads the selected Field file into memory and closes the file browser. After the file has been read into memory, the mesh and data components are passed to any downstream modules in the application. Selecting a different file, after one has been read in, will replace the first file's mesh and data components with the new file's

mesh and data components and all modules will be updated with the new data.

Related Modules

->Read_UCD

->Read_Field

Read_DXF

General Module Function

This module has been deprecated and replaced by the read_CAD module.

The Read_DXF module will read an industry standard CAD DXF format graphics file, and place the graphic entities into the EVS viewer. This module provides the user with the capability to integrate site plans, buildings, and other features into the EVS visualization, to provide a frame of reference for understanding the three dimensional relationships between the site features, and characteristics of geologic, hydrologic, and chemical features. The DXF entities are treated as three dimensional objects, which provides the user with a lot of flexibility in the placement of DXF objects in relation to EVS objects in the visualization. The surfmap module allows the user to drape DXF line-type entities (not 3D-Faces) onto three dimensional surfaces. Read_DXF supports the following DXF entities:

1) lines,
2) polylines,
3) arcs,
4) circles,
5) simplex text, and
6) 3d faces.

When creating a DXF from AutoCAD for import to EVS, the user should specify that the "Entities" be exported when first prompted for the number of decimal places (this will be an option In the ACAD command line). After selecting the objects to include in the DXF, the user can specify only the number of decimal places needed to adequately place the object in the drawing (2 decimal places is usually sufficient). Including many decimal places in a DXF drawing increases the size of the file and the time required to read it into EVS with no tangible benefit. Also note that although Read_DXF can handle large DXF files, a lot of other information will typically be displayed when using EVS, and thus the simplest DXF overlay that conveys the required information generally produces the best displays.

Note: the default rendering mode is outline mode = "off" for this module. Surfaces no longer show cell boundary lines, but lines are still visible. To view outlines of surfaces, set outline mode equal to "on".

Module Input Ports

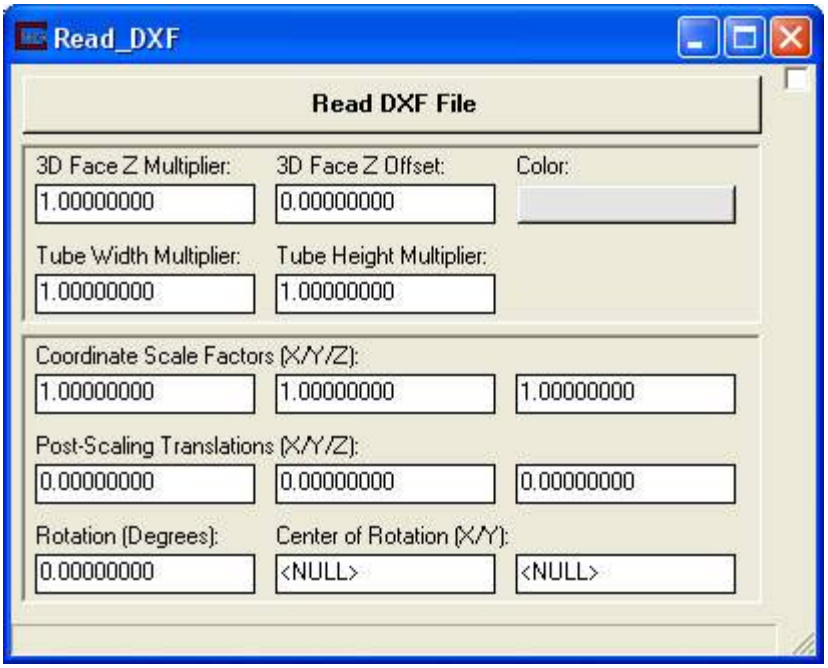
The Read DXF module is shown above. Read DXF has one input port for the Z Scale.

Module Output Ports

The first (left) port outputs the Z Scale.

The second port outputs a typical data field which can be input to surfmap, external edges and/or any of the Subsetting and Processing modules which have the same color port.

The right port outputs a renderable geometry, and can only be connected to the viewer.



Module Control Panel

The parameter input panel for Read DXF is shown above.

Clicking the ***Read DXF File*** button opens a standard windows style file browser which allows the user to select a DXF file from a specified directory. There are many different types of entities, and some differences in formats, that can be present in DXF files. EVS is currently able to process the common types of graphic entities such as lines, polylines, circles, and text. However, the user should be aware that some programs which produce DXF "compatible" files, may not actually be 100% compatible with the AutoCAD DXF format standard, and therefore may not import correctly to EVS. The module runs as soon as you read the file.

The ***3DFace Z Multiplier and Z Offset*** fields allow the user to scale the vertices of 3DFACE DXF entities, and offset their position in the Z axis relative to the original vertices. Note that these parameters only need to be used when special placing of 3D FACE entities is desired. The offset parameter must consider the scaling multiplier as described above for translation factors. These factors are applied after the primary scaling and translation factors listed above.

The color of the DXF entities which do not have assigned (DXF) entity colors are controlled by ***Color*** button. All entities that do not have set colors in the DXF receive the same color.

The ***Coordinate Scale Factors*** type-ins allow the user to scale the coordinates of the DXF entities in the X, Y, and Z axes. The default scale factors are 1.0, but the user can input any positive scaling values, by which the coordinates of the respective axes will be multiplied. The scale factors must be used to correctly place the DXF entities when Z exaggeration or other scale factors are being applied to the kriged data distributions to produce effective visualizations. In this case, the scale factor should be noted from the Explode and Scale or Kriging parameter input panels, and an equivalent, or nearly equivalent factor should be applied in the Read DXF scale factors (note that the scale factors usually only affect the Z axis scaling).

The ***Post Scaling Translations*** data input fields allow the user to translate the DXF entities in all three axes for placement in the visualization. These factors are useful if different coordinate systems are being used for the DXF file, and the geologic or hydrologic data. Note that if Coordinate Scale Factors are being applied, then the user must enter translation parameters that consider the scaling parameters. As an example, if a Coordinate Scale Factor of 10 is specified for the Z axis, and the user also enters a translation factor of 10, then the DXF entities will actually be translated 100 original units in the Z Axis.

The ***Rotation*** type-in rotates the entire output about the Z axis by the specified amount in degrees. The ***Center ofRotation(X/Y)*** type-ins determine the center for the rotation.

Explanations of the various parameter settings and their effects on the visualization are presented in Workbook 2.

Read_DXF Module Hints

Coloring DXF lines using the Viewer

Coloring lines from a DXF file can now be accomplished by first selecting the Read_DXF object in the Viewer, then choose Editors --> Object Editors. Within this panel choose Object Properties AND Type General. Then in the lower Object pull down list change the selection from primary to secondary. Now, changing the line color is possible via either HSV dials or sliders.

Read_Field

This is a deprecated module that has been supplanted by Load_EVS_Field

General Module Function

The Read Field module is used to read EVS field files, which generally have a *.fld* filename extension. An EVS field file contains structured cell geometries with implied connectivity and nodal or cell based data components. The cell

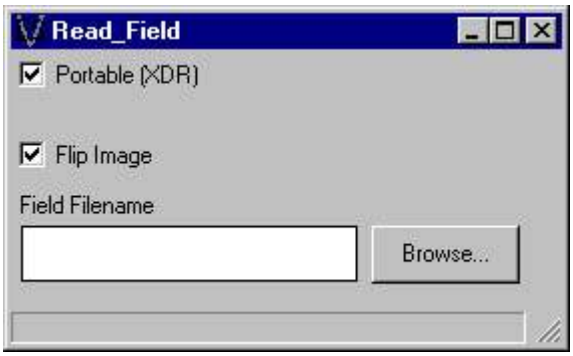
geometries generally consist of rectilinear cells similar to a finite-difference grid. Field files are not currently created within EVS. The field file format is discussed in the section below called Field File Formats Because modules in the current version of EVS use UCD data types, the Read_Field module is not utilized often.

Module Input Ports

The Read Field module is shown above. Read Field has no input ports. It obtains the Field input data by reading a file with a file browser.

Module Output Ports

Read Field has two output ports. The first port (closest to the left) passes the mesh and data components to other modules which accept Field data types. Data passed from Read Field is usually sent to modules in the Subsetting and Processing Libraries. The second port is used to send Field file data directly to the viewer for rendering.



Module Control Panel

The user interface for Read Field consists of two toggles and a file browser shown above. The Portable XDR toggle (checked by default) specifies whether the data is in the machine-dependent XCR format. With this parameter enabled, and when binary data is being read in, the machine assumes that the data is in XDR format. Note that these types of fields are not created by EVS, but may be available through outside sources and are thus accommodated by this option. The Flip Image toggle specifies whether image data is inverted before it is output. Image data could be expressed as a field and is typically described with the upper-left corner as the starting point. As a result, when it is read in as field data, the image may appear flipped vertically. The default is "ON" which usually interprets EVS fields properly.

The file browser is used for selecting the Field file (Field files have a *.fld* extension). The default directory path is the \EVS\DATA\FIELD. Double clicking on a filename or selecting a file name and choosing OK reads the selected Field file into memory and closes the file browser. After a Field file has been read into memory, the mesh and data components are passed to any downstream modules in the network

Field File Formats

This is a description and some examples for the EVS field data format. The field format can be binary or ASCII files. These examples are in ASCII

format. The standard field format is more memory efficient then the UCD format, but does not allow exploding geologic layers for visualization purposes. Otherwise it is probably better for Finite Difference code output.

Note: in EVS, coordinates apply to nodes (versus cells). If you write out the coordinates of cell centers (such as with block-centered flow data: MODFLOW), EVS will visualize these as nodes at the corners of hexahedral volume cells. To be most correct, MODFLOW models should be output with one extra node in all three axes. However, the compromise is generally acceptable. Note that the Groundwater Vistas Pre-Processor creates the 'correct nodal' corners of the s mesh and outputs a .fld and .dat file which can be read directly into EVS.

Generally, the .fld file describes the structure(s) of the data file(s) to be read and also describes how to read them. For example, if you want EVS to read a .dat file with coordinates you specify a combination of skip, offset and stride to read the x, y, then z values in the .fld file. Simply put, *'skip'* specifies the number of *lines* to skip before reading data; *'offset'* specifies *which value* in the line to read (e.g. offset of '0' means read 1st value, offset of '1' means read second value, etc.); *'stride'* specifies how many *values to jump* before starting to gather data again (e.g. for a file with x,y,z coords. the stride would be 3....if there is x,y,z and value the stride would be 4.

The files below are for reading a file which contains only coordinates AND a file which contains only data values. This is quite powerful in that you may generate only one coordinate file, then keep changing/adding new data files. A simpler yet less robust example would involve one .dat file which contains the coords AND the data component(s) in one file. The two .fld files are to show that you may have multiple components....note that three of the components are velocity vector.

File list and descriptions:

mt3d_smp.fld Simple field header file to read only one data component (head). This file defines the files that contain coordinate data and nodal parameter data. In other words, this tells EVS how to read the data file.

The text below is from the file: mt3d_smp.fld

```
# AVS FIELD the string "# AVS/Express" must be the first five
# characters in the file.
ndim=3 # REQUIRED - the number of dimensions
dim1= 51 # REQUIRED - dimension of axis 1
dim2= 31 # REQUIRED - dimension of axis 2
dim3= 8 # REQUIRED - dimension of axis 3
nspace=3 # REQUIRED - coordinates per point
veclen=1 # REQUIRED - components at each point
data=float # REQUIRED - data type
field=irregular # REQUIRED - field type
label=head # OPTIONAL - label for variable 1
unit=feet # OPTIONAL - unit label for variable 1
```

#
For each value in the vector: data reading instructions
#
variable 1 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 1 stride = 13
#
For each coordinate X, Y, and Z: data reading instructions
#
coord 1 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 1 stride = 4
coord 2 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 2 stride = 4
coord 3 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 3 stride = 4

**********End of text**********

Note: Any characters following (and including) # in a header line are ignored.

mt3d_all.fld Simple field header file to read all 12 data components (last 3 are vel. vector). This file defines the files that contain coordinate data and nodal parameter data.

The text below is from the file: mt3d_all.fld

```
# AVS FIELD
ndim=3
dim1= 51
dim2= 31
dim3= 8
nspace=3
veclen=12
data=float
field=irregular
label=head drawdown thickness Geolayer Elevation Conductivity Storage Concentration
Change_in_Conc Velocity_x Velocity_y Velocity_z
unit=feet feet feet number feet feet/day 1/feet mg/kg mg/kg feet/day feet/day feet/day
variable 1 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 1 stride = 13
variable 2 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 2 stride = 13
variable 3 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 3 stride = 13
variable 4 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 4 stride = 13
variable 5 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 5 stride = 13
variable 6 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 6 stride = 13
variable 7 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 7 stride = 13
variable 8 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 8 stride = 13
variable 9 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 9 stride = 13
variable 10 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 10 stride = 13
variable 11 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 11 stride = 13
variable 12 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 12 stride = 13
coord 1 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 1 stride = 4
coord 2 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 2 stride = 4
coord 3 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 3 stride = 4
```

**********End of text**********

mt3d_smp.crd Field coordinate data file (first column (node number) is not needed so offset = 1, 2, or 3) Coordinates can/should be in user's original coordinate system. If model was rotated (to be run finite difference) and/or translated, it should be put back in real coordinates. This allows for incorporating simulation data with measured data results and incorporating CAD drawings drawn in real coordinates. In this example, the first column is the node number and is not needed in the file (so we offset to skip over it during reading).

Note: numerous lines were dropped from this example.

The text below is from the file: mt3d_smp.crd

```
1 9625.112 10786.380 6.275
2 9714.992 10742.550 6.367
3 9804.871 10698.710 6.581
4 9894.751 10654.870 6.939
5 9984.630 10611.030 7.581
6 10074.510 10567.200 8.159
7 10164.390 10523.360 8.808
8 10254.270 10479.520 9.420
9 10344.150 10435.690 9.922
10 10434.030 10391.850 10.080
.
.
.
.
12640 12084.940 6248.842 -36.863
12641 12174.820 6205.005 -36.843
12642 12264.690 6161.168 -36.822
12643 12354.570 6117.331 -36.803
12644 12444.450 6073.494 -36.562
12645 12534.330 6029.657 -36.317
12646 12624.210 5985.820 -37.157
12647 12714.090 5941.983 -38.017
12648 12803.970 5898.146 -38.020
```

*********End of text**********

mt3d_smp.dat Field nodal data file (first column is not needed so offset = 1, 2, etc.). This file contains the calculated data from the model. In this example, the first column is the node number and is not needed in the file (so we offset to skip over it during reading).

Note: numerous lines were dropped from this example file to keep it small.

The text below is from the file: mt3d_smp.dat

```
1 .890 .000 19.1 0 6.28 15.0 .250 .000 .000 .000 .535E-02 .000
2 .890 .000 19.6 0 6.37 15.0 .250 .000 .000 -.271E-02 .425E-02 .000
3 .895 .000 20.6 0 6.58 15.0 .250 .000 .000 -.271E-02 .817E-02 .000
4 .900 .000 21.2 0 6.94 15.0 .250 .000 .000 .000 .774E-02 .000
5 .900 .000 21.2 0 7.58 15.0 .250 .401E-41 .000 -.247E-02 .672E-02 .000
6 .905 .000 21.3 0 8.16 15.0 .250 .811E-37 .000 -.487E-02 .897E-02 .000
7 .915 .000 21.9 0 8.81 15.0 .250 .532E-33 .000 -.470E-02 .101E-01 .000
8 .925 .000 22.0 0 9.42 15.0 .250 .217E-29 .000 -.452E-02 .107E-01 .000
9 .935 .000 22.1 0 9.92 15.0 .250 .000 .000 -.445E-02 .108E-01 .000
10 .945 .000 22.6 0 10.1 15.0 .250 .122E-21 .000 -.718E-02 .116E-01 .000
.
```

```
.
.
.
12638 .264 .406 2.36 4 -40.9 1.00 .100E-05 .124E-20 .000 .223E-02 -.549E-03 .905E-03
12639 .218 .456 2.37 4 -41.2 1.00 .100E-05 .111E-18 .000 .248E-02 -.791E-03 .836E-03
12640 .165 .513 2.42 4 -41.2 1.00 .100E-05 .726E-15 .000 .275E-02 -.111E-02 .123E-02
12641 .107 .573 2.51 4 -41.0 1.00 .100E-05 .484E-12 .000 .304E-02 -.153E-02 .147E-02
12642 .409E-01 .643 2.57 4 -40.7 1.00 .100E-05 .129E-09 .000 .340E-02 -.208E-02 .165E-02
12643 -.354E-01 .724 2.80 4 -39.8 1.00 .100E-05 .103E-07 .000 .373E-02 -.281E-02 -.490E-03
12644 -.123 .818 2.92 4 -39.2 1.00 .100E-05 .325E-06 .000 .388E-02 -.376E-02 -.452E-02
12645 -.218 .919 2.85 4 -39.3 1.00 .100E-05 .282E-05 -.331E-04 .374E-02 -.490E-02 -.813E-02
12646 -.311 1.02 2.84 4 -38.9 1.00 .100E-05 .122E-04 -.325E-03 .323E-02 -.613E-02 -.112E-01
12647 -.397 1.11 2.79 4 -38.7 1.00 .100E-05 .221E-04 -.814E-03 .260E-02 -.744E-02 -.127E-01
12648 -.466 1.19 2.68 4 -38.4 1.00 .100E-05 .337E-04 -.132E-02 .152E-02 -.838E-02 -.177E-01
```

**********End of text**********

Related Modules

->Read_UCD

Read_UCD

This is a deprecated module that has been supplanted by Load_EVS_Field

General Module Function

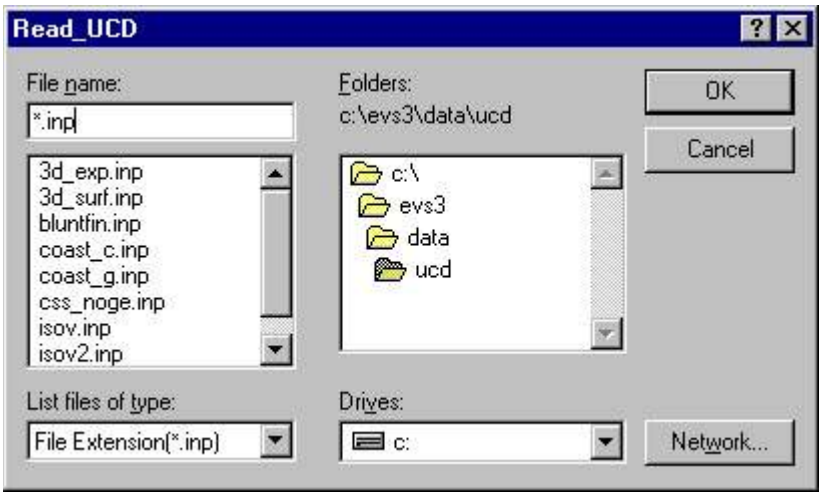
The Read UCD module is used to read EVS Unstructured Cell Data (UCD) files. These files have a default *.inp* filename extension, and contain ASCII text. UCD files contain cell structure data and nodal or cell based data components. The format of a UCD file is discussed below. UCD files are created using Write_UCD , the MODFLOW_*2*_UCD module, or externally by the user. A typical use of Read UCD involves using Krig Geology and/or Krig_3D to krig a .apdv file, writing a UCD file of the Krig_3D output using Write UCD, and reading the UCD file into another network using Read UCD. This eliminates the need for re-kriging the data every time a data set is used. It is important to understand, however, that a UCD contains the range of property values that are output by a kriging module, which may or may not represent the entire actual range of the input data (the .apdv or .GEO file). The range of data output by kriging modules depends on the grid density and kriging parameters used to produce the data field. Uniform maximum and minimum values that are written to a UCD can be controlled in the kriging modules, or by using the set_minmax module.

Module Input Ports

The Read UCD module is shown above. Read UCD has no input ports. It obtains the UCD input data by reading a file with a file browser.

Module Output Ports

Read UCD has one output port, which passes the mesh and data components to other modules. Data passed from Read UCD is usually sent to modules in the Subsetting and Processing Library.



Module Control Panel

The user interface for Read UCD consists solely of the file browser shown above. The file browser is used for selecting the UCD file (UCD files have a *.inp* extension). The default directory path is the \EVS\DATA\UCD. Double clicking on a filename or selecting a file name and choosing OK reads the selected UCD file into memory and closes the file browser. After a UCD file has been read into memory, the mesh and data components are passed to any downstream modules in the network. Selecting a different UCD file, after one has been read in, will replace the first UCD file's mesh and data components with the new UCD file's mesh and data components and all network modules will be updated with the new data.

UCD File Format

The format of UCD files is very similar to that commonly employed by node based finite element models. Essentially, UCD files contain three different sections with a header line that describes the type and amount of data that is contained in each section. A UCD file cannot contain blank lines or lines with leading blanks. Comments can be present at the top of the file, but are not allowed in other parts of the file. Leading comment lines must begin with the "#" symbol. An example UCD file is presented below, with explanation lines describing the contents of the lines (which can not be present in actual UCD files) placed in brackets "{}" and colored *magenta and italicized*. If the user wishes to create or manipulate UCD files, they should examine some of the example UCD files contained in the EVS\DATA\UCD directory in a text editor to become familiar with the UCD structure. The file described below is the hex.inp example UCD file, which can be found in the sample UCD directory. This UCD file consists of one hexahedral cell with 8 nodes, and each node has a single scalar data value.

Lines with leading "#" characters in a UCD file can be comments.

The user can include any number of comment lines.

{The first section of data in the file begins with a header line, which describes the number of nodes in the model, the number of cells in the model, the number of nodal data components, the number of cell data components, and the number of model data components (cell and model

data components are not currently used in EVS). The first section of the file looks like this:}

8 1 1 0 0 {8 nodes, 1 cell, 1 nodal data component, 0 cell data, 0 model data}

1 0.000 0.000 1.000 {node 1, xcoord = 0, ycoord = 0, zcoord = 1]

2 1.000 0.000 1.000 {node 2, xcoord = 1, ycoord = 0, zcoord = 1]

3 1.000 1.000 1.000 {node 3, xcoord = 1, ycoord = 1, zcoord = 1]

4 0.000 1.000 1.000 {and so forth for the 8 nodes in this hexahedron]

5 0.000 0.000 0.000

6 1.000 0.000 0.000

7 1.000 1.000 0.000

8 0.000 1.000 0.000

{The second section of the file contains the list of elements in the model, the material type (or layer) of each element, the type of element (EVS usually uses quads and hexahedrons), and the nodes that make up the vertices of the element, listed in order of connectivity starting from the upper left vertex, and progressing counterclockwise for each layer of vertices in the element (see the diagram below). Remember these lines starting with brackets are for explanation only and cannot actually be present in the file.}

1 1 hex 1 2 3 4 5 6 7 8 {Element 1, Material 1, Hexahedral Element Type,Node numbers for nodes at vertices of element. This element has 8 vertices, at nodes 1 through 8. Note that the nodes were identified by number in the first section of the file, and along with their X, Y, Z, coordinates}

{The third section of the file contains a first line that lists the number of data components for each node, and then a list of the number of properties belonging to each component. Most EVS modules output "scalar" properties, which by definition have only one property per data component. However, the MT3D model outputs a vector component describing the velocity of fluid flow at the model nodes, which has the three properties of X, Y, Z velocity present in one data component. An example of this type of "vector" data component is presented below.}

1 1 {One data component, One property in the first component "a scalar" The next n lines of the third section of the file contain text entries separated by commas, that describe the name and units of the property in each data component. There are as many of these text lines as there are data components.}

stress, lb/in**2 {Data Component 1 is the Stress property, Units are lb/in**2]

The last n lines in the third section of the file list the node numbers, and the values for the properties in each data component. There are as many lines as there are nodes in the model.}

1 4999.9999 {Node 1, 4999.9999 (Stress Property Value in lb/in**2)}

2 18749.9999 {Node 2, 18749.9999 (Stress Property Value in lb/in**2)}

3 37500.0000 { Node 3, and so on ...}

4 56250.0000

5 74999.9999

6 93750.0001

7 107500.0003

8 5000.0001

{The last node number and list of data component property values is the last line of the file.}

A more complex example of a UCD file is provided below, which is an edited version of the sample file sp0112.inp, which is located in the EVS\DATA\MODFLOW\PROB.SPI\SAMPLE directory. Note that this UCD file has no header lines, and has a vector property in the last nodal data component (in the third section of the file). Again, some explanations of the file structure are provided, which are enclosed in brackets "{}"and colored magenta. These are not in the actual UCD file.

22134 10500 12 0 0 {Line 1 of Section 1; 22134 nodes, 10500 elements, 12 nodal data components}

1 9625.112 10786.380 6.275 {Node 1, 9625.112 X Coord, 10786.380 Y Coord, 6.275 Z Coordinate}

2 9714.992 10742.550 6.367

3 9804.871 10698.710 6.581

4 9894.751 10654.870 6.939

5 9984.630 10611.030 7.581

. {Data lines for nodes 6 through 22130 Omitted from this example}

.

22131 12534.330 6029.657 -62.443

22132 12624.210 5985.820 -62.879

22133 12714.090 5941.983 -63.324

22134 12803.970 5898.146 -63.318 *{Node 22134 is last in Model}*

1 0 hex 1 52 53 2 1582 1633 1634 1583 *{Line-1 of Section 2; Element 1, Material 0, Hex Elem., Nodes }*

2 0 hex 2 53 54 3 1583 1634 1635 1584 *{at vertices of element 1 are 1,52,53,2,1582,1633,1634,1583}*

3 0 hex 3 54 55 4 1584 1635 1636 1585

4 0 hex 4 55 56 5 1585 1636 1637 1586

5 0 hex 5 56 57 6 1586 1637 1638 1587

. {Data lines for elements 6 through 10494 Omitted from this example}

.

10495 6 hex 20496 20547 20548 20497 22077 22128 22129 22078

10496 6 hex 20497 20548 20549 20498 22078 22129 22130 22079

10497 6 hex 20498 20549 20550 20499 22079 22130 22131 22080

10498 6 hex 20499 20550 20551 20500 22080 22131 22132 22081

10499 6 hex 20500 20551 20552 20501 22081 22132 22133 22082

10500 6 hex 20501 20552 20553 20502 22082 22133 22134 22083 *{Element 10500,.Material 6...Last in Model}*

10 1 1 1 1 1 1 1 1 3 {Line 1 of Section 3; 10 Data components, Comp.1 through 9 have 1 property, Comp 9 has 3}

head, feet {First data component is head in feet (a scalar property)}

drawdown, feet {Second data component is drawdown in feet}

thickness, feet {Third data component is layer thickness in feet}

Geolayer, number {Fourth data component is Geolayer number, which specifies which element material it beongs to}

Elevation, feet {And so forth...}

Conductivity, feet/day

Storage, unitless or 1/feet

Concentration, M/L^3

Change in Conc, M/L^3

Velocity, L/t {Velocity has three components, Vel in X, Vel in Y, Vel in Z, in L/t, a vector data component}

3 .895 .000 20.6 0 6.58 15.0 .250 .286E-03 -.143E-03 -.750E-02 .115E-01 .000

4 .900 .000 21.2 0 6.94 15.0 .250 .341E-03 -.314E-03 .000 .111E-01 .000

5 .900 .000 21.2 0 7.58 15.0 .250 .756E-05 -.171E-03 .000 .101E-01 .000

. {Data lines for nodes 6 through 22129 Omitted from this example}

.

22130 .735 .000 3.77 6 -63.0 105. .100E-05 .295E-02 -.667 -.525E-01 -.732E-01 .000

22131 .740 .000 3.85 6 -62.4 105. .100E-05 .504E-03 -.947 .000 -.330E-01 .000

22132 .740 .000 4.14 6 -62.9 105. .100E-05 .352E-06 -.733 .000 .501E-02 .000

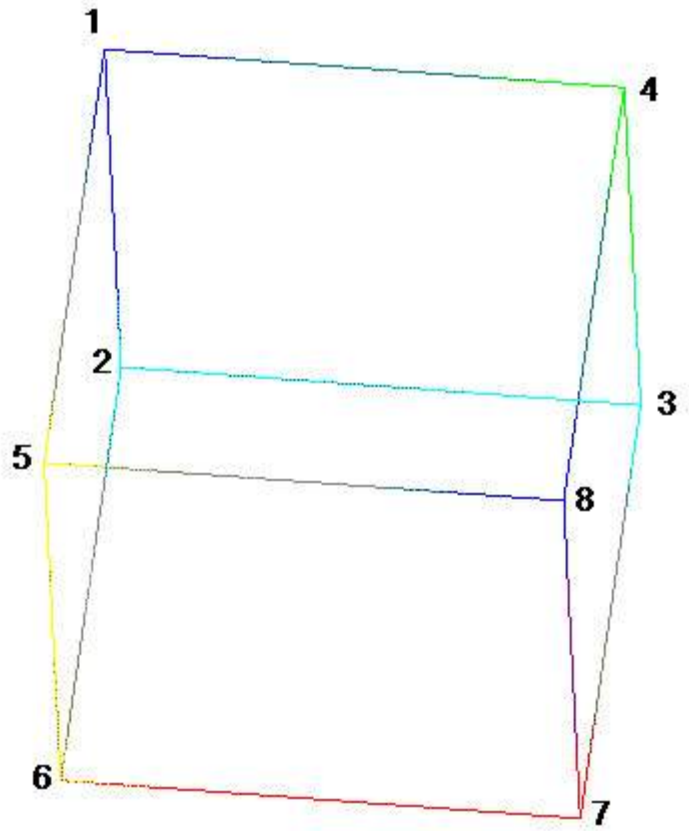
22133 .740 .000 4.44 6 -63.3 105. .100E-05 .000 -.644 .000 .525E-01 .000

22134 .740 .000 4.45 6 -63.3 105. .100E-05 .000 -.643 .000 .525E-01 .000 *{Node 22134, last in model}*

The node numbering order for hexahedrons used by EVS is demonstrated in the following figure:

1 .890 .000 19.1 0 6.28 15.0 .250 .742E-07 .000 .000 .750E-02 .000 *{Node 1, properties as listed above}*

2 .890 .000 19.6 0 6.37 15.0 .250 .000 .000 .000 .590E-02 .000



The nodes are numbered starting in the upper left corner of a hexahedron or quadrilateral element, and increase first counterclockwise, and then downward and counterclockwise as shown.

Related Modules

->Read_Field

Write_netCDF

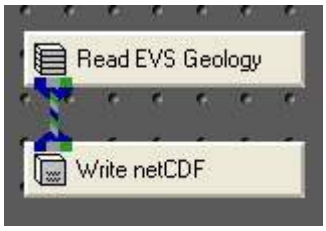
This is a deprecated module that has been supplanted by Save_EVS_Field

General Module Function

The Write_netCDF module creates a netCDF file containing all the mesh and nodal data component information sent to the input port. NetCDF files are similar to UCD files but they are in binary format and therefore cannot be edited. This module is useful for writing the output of modules which manipulate or interpolate data (Krig_3D , Krig_2D , etc.) to netCDF files so that the data will not need to be processed in the future. The process data in the netCDF can be read using Read_netCDF, which is much faster than reprocessing the data. Similarly, since these files are binary, they will read faster than a UCD file.

As of Version 6.0 the output of Krig_3D_Geology and Spline_Geology can (and should be) saved as a netCDF file using Write_netCDF.

We strongly recommend that you convert older .egf files to the newer format using the following simple process:



Module Input Ports

The Write netCDF module is shown above. The module has one input port which accepts any mesh and nodal data.

Module Output Ports

Write netCDF has no output ports.

Write netCDF Control Panel

The Write netCDF control panel is shown above. Clicking on the Write netCDF File push button opens a standard windows file browser. The filename and location can be specified in this browser. Clicking on the OK button tells Write netCDF to create the netCDF file. The user is not warned if the file they are creating currently exists. The file will be overwritten.

Related Modules

-> Write_UCD

Pre_Geology

This is a deprecated module that has been supplanted by make_geo_hierarchy

General Module Function

The Pre_Geology module reads a special input file format called a pgf file, and then allows the user to build geologic surfaces based on the input file's geologic surface intersections. This process is carried out visually (in the EVS Viewer) with the use of the Pre_Geology user interface, and by selecting spheres in the Viewer using the Alt-left mouse action. Once each surface is created, the user may create or append a gmf file, which can be used with other geology modules.

Module Input Ports

Pre_Geology has three input ports. The left port is the filename port. The second port accepts the scaling (Z exaggeration) factor and the right port is the refine distance..

Module Output Ports

Pre_Geology has six output ports.

The leftmost port is the filename port.

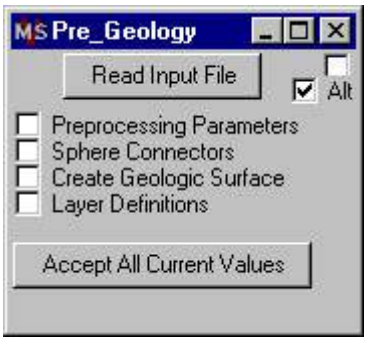
The second port (gray-brown) is the scaling (Z exaggeration) factor.

The third (gray-magenta) port outputs the refine distance to Indicator_Geology.

The fourth (red) port sends a renderable object to the viewer containing both tubes, spheres and surfaces.

The fifth (blue-black) port is the spheres and is usually used as input to axes.

The sixth (blue-black) port is the tin surface (if it exists).



Module Control Panel

The main panel for Pre_Geology is shown above. The Read Input File activates a browser for selecting any file with a .pgf extension. The Accept All Current Values button will activate the processing of the file with the current parameter settings.

Since this module relies heavily on procedure, we have supplied a short tutorial to familiarize the user with the most generalized simple use for this module. Here's a short step-by-step tutorial for reading an existing .pgf file, interactively creating geologic surfaces in a 3D environment, then creating and appending a gmf file with each surface you make. The resulting gmf file is suitable for use with other EVS/MVS geology modules and other EVS/MVS visualization tools:

Step 1 Create a Pre_Geology Network.

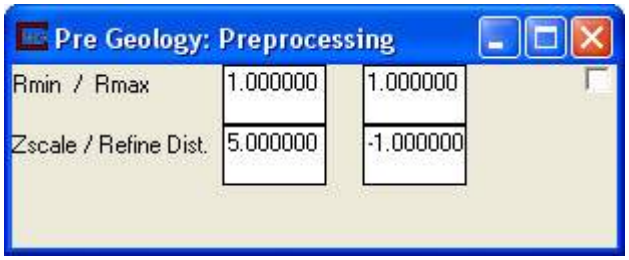
Step 2 Read a pgf file and prepare to create your first geologic surface.

Step 3 Create a geologic surface (TIN) by clicking your mouse on borings in EVS Viewer .

Step 4 Write your geologic surfaces to a gmf file using the Graphical User Interface.

Step 5 "OPTIONAL" Display your 3D geologic model while creating the geologic surfaces.

Now that you have experienced the basic functionality of Pre_Geology, the following listing of module features is provided with descriptions.



The **Preprocessing Parameters** subpanel is shown in the figure above.

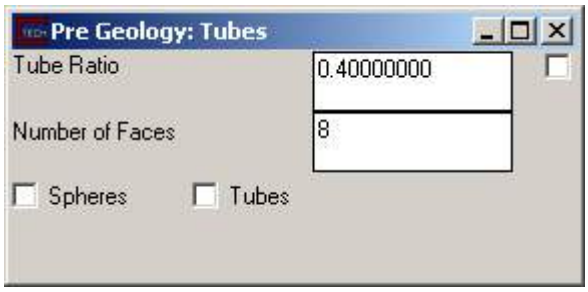
The **Rmin/Rmax** type-ins control the size of the spheres in the viewer. The Rmin affects sizing of the spheres with the lowest layer number, and the Rmax affects the sizing of spheres with highest layer number.

The **Z scale** sets the vertical exaggeration of the borings in the view.

The ***Refine Dist***(ance) parameter is -1 by default which causes no refinement to be performed. The purpose of this parameter is to be able to visualize the refinement that will be used by the Indicator_Geology module. That is also the reason that both of these modules have a refinement input & output port.

Understanding the ***Refine Dist***(ance) parameter requires an understanding of the basics of indicator kriging with a .PGF file as input. Since a PGF file has only a single point defining the bottom of intervals of a specific material, it is necessary to create an intermediate dataset that is used for the actual kriging. Since each interval represents a line segment with constant material characteristics, we approximate it by refining the interval into a set of points. The spacing between these points affects accuracy. Each segment will receive at least two points near the end points of the interval, but longer segments will broken up into segments no larger than the refine distance. Setting this value too small will result in much greater run times, as well as requiring higher values for the points parameter. Setting this value too large can create artificial inaccuracies in the prediction because the center of the intervals can have a lack of sample points, causing points in other borings to appear closer. If you leave the default setting of 0.00 for this parameter the expert system determines the spacing for you by searching the original PGF file for the shortest segment in any boring. The calculated refine distance will be twice the shortest segment length or one half the height of the first cell in the grid, whichever is larger. This tends to give reasonable results with most datasets. You can also use the Pre_Geology module to visualize the refined points directly by setting its refine distance parameter to be the same as the computed one in Indicator_Geology.

Note that these type-ins will not take effect in the Viewer until the Accept All Current Values button is pushed in the main panel.



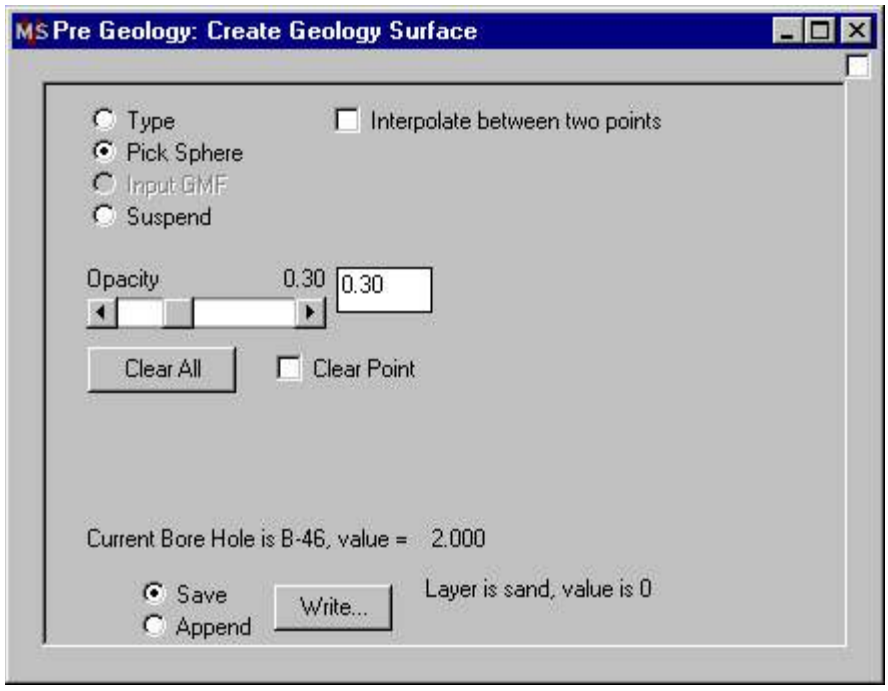
The Sphere Connectors subpanel is shown in the figure above.

The Tube Ratio affects the radius of the tubes connecting the spheres (points). The default is 0.4 and results in a tube radius that is 40% of the sphere radius, thus a setting of 1.00 would result in a tube radius that is 100% that of the sphere radius.

Number of Faces controls how many sides will be rendered for each tube. The default of 8 creates an octagonal tube, and 4 creates a square tube. The minimum setting of 2 creates a ribbon along the borehole trace. This setting is useful if trying to preserve memory while rendering a high number of borings (tens of thousands).

The Spheres toggle controls visibility of the spheres (indicating the boundaries between geologic materials).

The Tubes toggle controls visibility of the tubes (indicating the geologic borings).



The Create Geology Surface subpanel is shown above. Note that the indications in the above figure will only become visible after selecting your first sphere in the Viewer via the Alt-left mouse. THE CREATE GEOLOGY SURFACE PANEL MUST BE OPEN TO CREATE GEOLOGIC SURFACES. Also note

that the first triangle of the TIN (Triangulated Irregular Network) will not be visible until the first three spheres are selected.

The *Type* toggle activates type-in boxes for selecting a specific x,y,z location for a node of the TIN. The boxes are populated by default with the x, y, and z of the currently selected sphere.

The Pick Sphere toggle is the default mode whereby Alt-left mouse clicks place the node of the TIN surface at the sphere location.

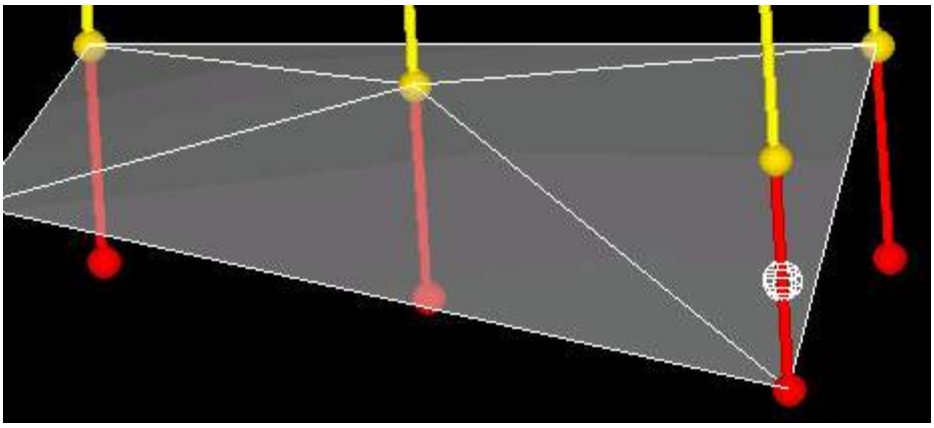
The *Input GMF* toggle has a very specific but important function, which requires a strict order of actions to be used. The purpose of the Input GMF function is to either a.) Read in a gmf (created previously in Pre_Geology) and make corrections to a single surface of the gmf file, or b.) Read in and APPEND to a gmf (created previously in Pre_Geology). In either case the user MUST re-instance Pre_Geology and first select the pgf file used to make the gmf file of interest, then open the Create Geology Surface panel, and then toggle the Input GMF File option. Here, the user must select the gmf file that coincides with the already selected pgf file. Next, the window called Select Surface appears and prompts for choosing the surface that you want to correct or append to.

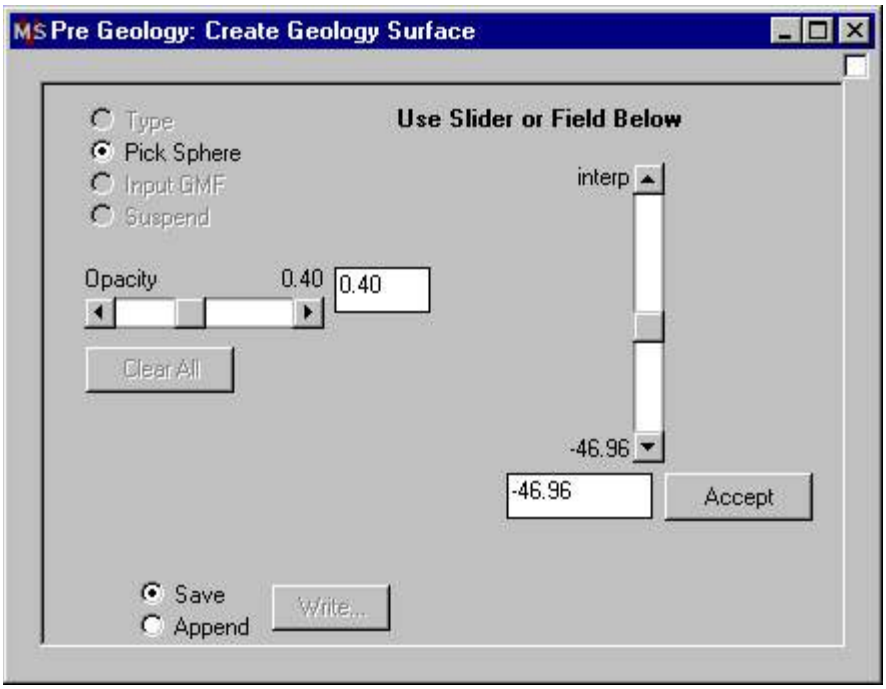
In case a.) - *Making corrections to a surface*, the user should make the adjustments to the selected surface and then choose Write in the Create Geology Surface panel (with Save toggled, NOT Append), and select the gmf file previously chosen and overwrite it. This action will make the adjustment only to the selected surface and any other changes will only be made to the initially selected surface. If the user would like to make changes to multiple surfaces in a gmf file, they must then re-instance Pre_Geology and repeat the above procedure.

In case b.) - *Appending to an incomplete gmf file*, the user should select the bottom surface and immediately choose the write function (with Save toggled, NOT Append), and overwrite when prompted. Then continue selecting deeper surfaces and choose the Append toggle to continue writing surfaces.

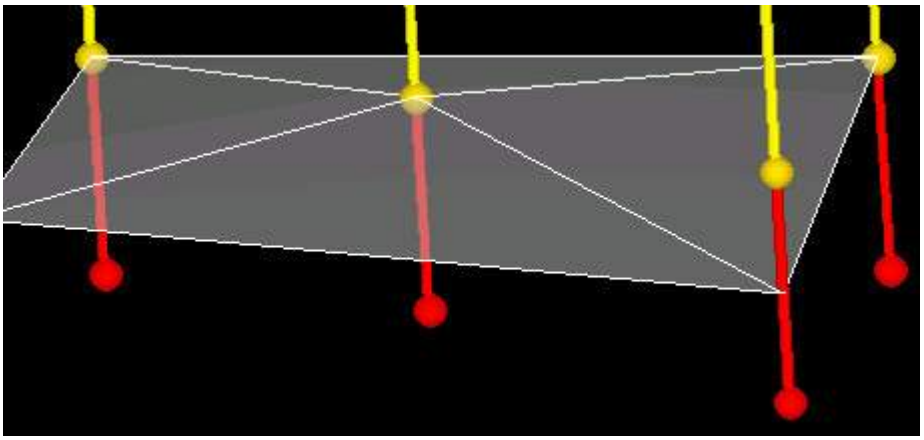
The *Suspend* toggle allows the user to use the alt-left mouse action for NON Pre_Geology tasks while this module is instanced. For example, if the user wanted to use some drawing modules like click_sketch.

The *Interpolate between two points* toggle activates functionality for selecting two spheres within the same borehole (a strict requirement). Then the user moves a slider to achieve a user-specified elevation. Once toggled, the user is prompted to select the two spheres and move the slider. This procedure will produce a view and panel which looks similar to this:





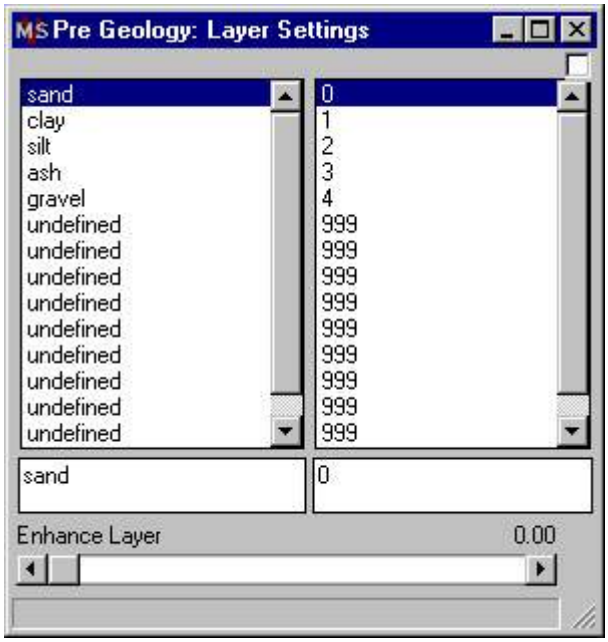
Once the mesh rendered sphere is placed at the desired elevation, the user pushes Accept and the TIN surface node should move to that location, producing the following view:



The *Opacity* slider adjusts the transparency setting for the TIN surface. A setting of 1.0 is completely opaque and 0.0 is invisible. The default 0.4.

The *Clear Point* toggle, when activate allows for clearing previously selected spheres from use in the TIN surface. The panel borders becomes red when activated.

The *Save* and *Append* toggles establish the two possible modes for writing the current surface to a gmf file with the *Write* button. Save mode should be selected for the first surface of a gmf file, or when correcting a single gmf file surface which was read in using Input GMF. The Append mode should be selected whenever newly created surfaces are written to the bottom of the gmf file. NOTE: Surfaces may NOT be inserted into the middle of the gmf file.



The Layer Settings subpanel is shown above. The main function of the Layer Settings panel is for assigning material color numbers to the various layers in the gmf file. The left half of the panel is for assigning user-defined material descriptions to the material numbers, which are input on the right side. The text descriptions are NOT used by EVS/MVS but serve as a useful way of keeping track of layer information, correlations and general organization. The pgf file (used to create the spheres and tubes in Pre_Geology) also results in coloring of the spheres and tubes in and may or may not correspond with the material color numbers in the resulting gmf file. It is recommended that the material numbers in the pgf file correspond with the expected layers in the gmf file to help visually resolve the existence of pinchouts in the geologic model.

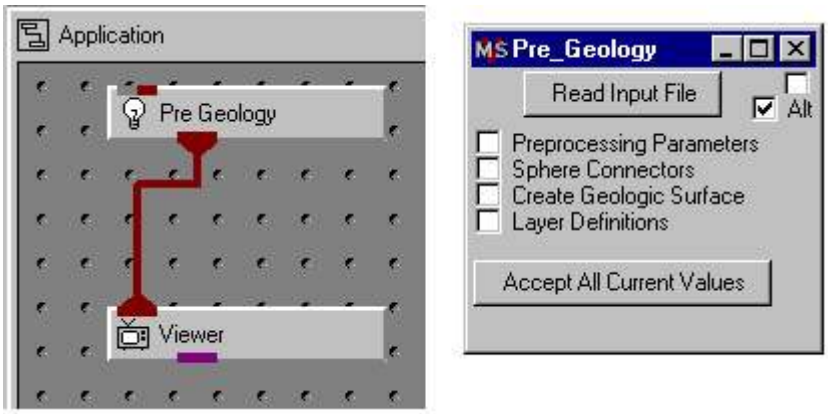
The user must select the desired material color number (and therefore text description) before writing the current surface. Once the surface is written to the gmf file, the proper material color number is assigned to that surface with the x, y and z values written beneath. Note that the currently selected layer setting is reported in the Create Geology Surface subpanel as a reminder of what you have selected. You may define up to 15 material

numbers. You may repeat material numbers and they may be listed in any order (for example if sand is material number 2 and clay is number 3, you may have alternating sand, clay as 2,3,2,3, etc.)

The *Enhance Layer* slider enlarges the spheres of the selected layer. For example, if layer settings is currently selecting sand/material number 2 and Enhance Layer is set to 0.5, then any spheres with material number 2 will be twice as large as the other spheres. This is helpful for spotting the spheres you are currently trying to correlate.

Tutorial_Pre_Geology

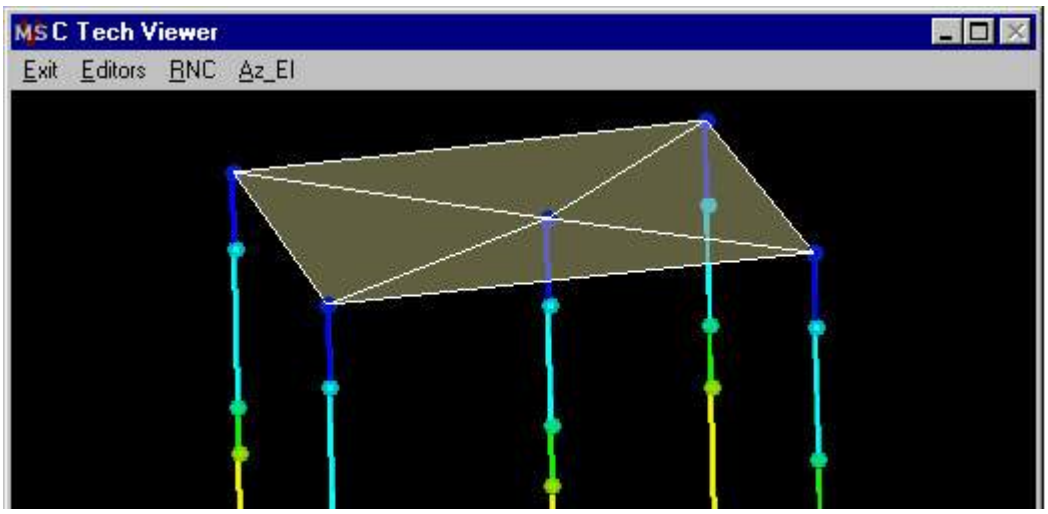
Step 1: Instance Pre_Geology and Viewer and then connect them. You should immediately see the Viewer and Pre_Geology control panel on your screen.



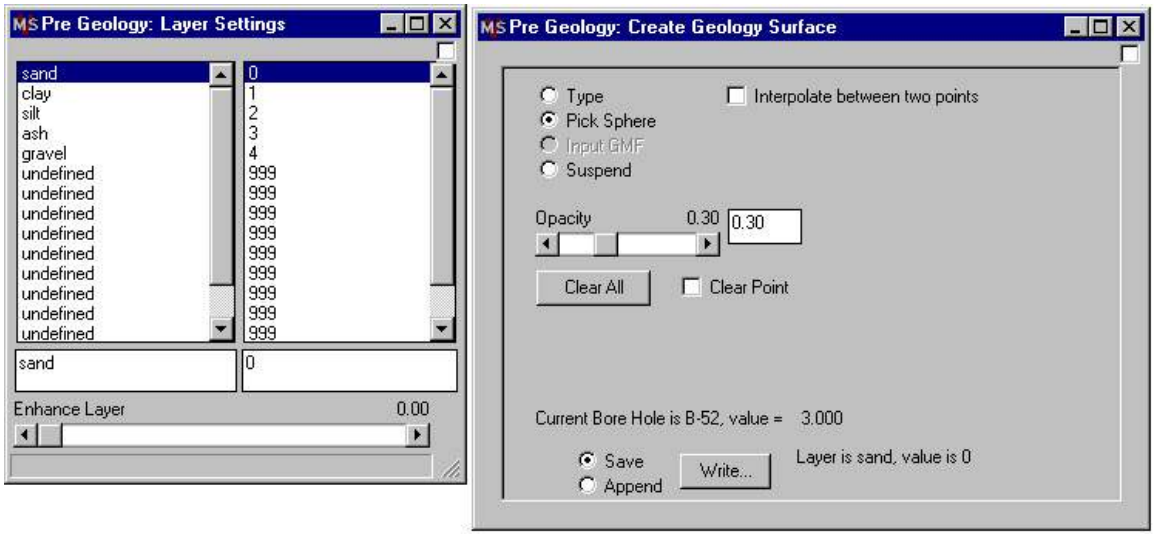
Step 2: Using the Read Input File button, open the interface and select on the pgf file named initial_soil_investigation_subsite.pgf. Then go back to the panel and toggle on Create Geologic Surface. Once you see the interface for this function you should move this to an "out-of-the-way' place on your screen, but still visible (a good place is over the network editor palette area). You are now ready to create a geologic surface via mouse interactive clicking in the view screen. Don't start clicking yet!

Note: You will not be able to perform this operation unless the Create Geologic Surface interface is open.

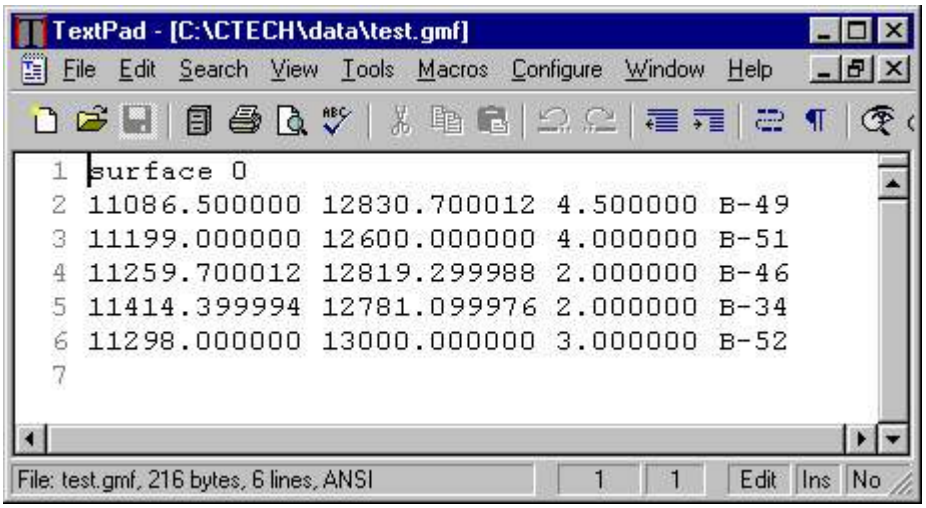
Step 3: Now that the Create Geologic Surface is open you will use the Alt-Left mouse action to select lithologic intersections. Since this surface will be a series of triangles you must select three unique points (spheres), before seeing the first triangle of the surface. Remember, you must click only one sphere per boring, and you should start at the top and work your way down to preserve the lithologic hierarchy. Once you have completed clicking all the points you should see something like the picture below. Now let's move to the next step where we will write this first surface to a gmf file.



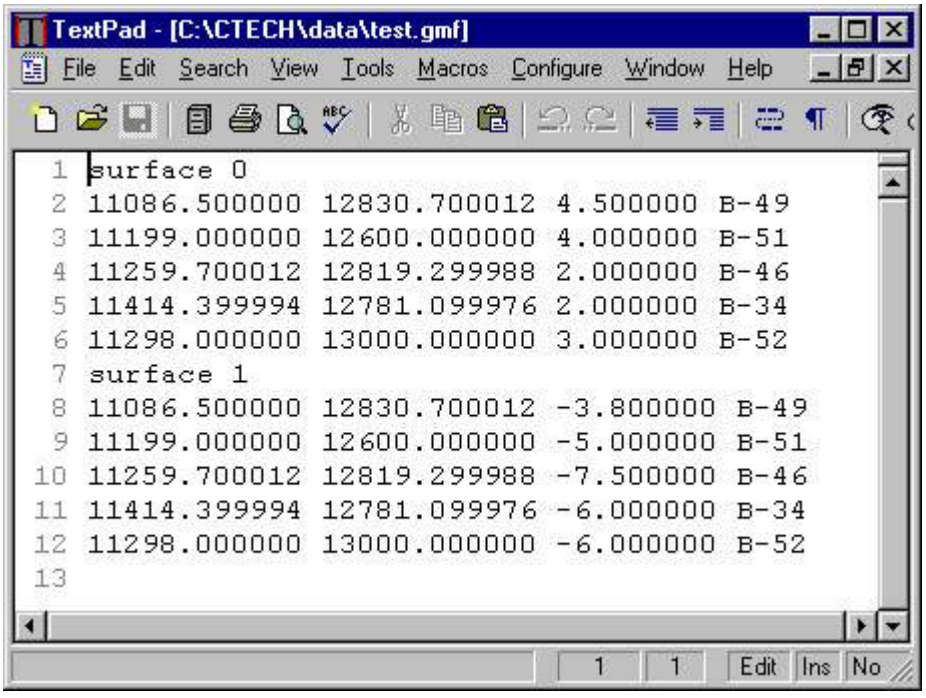
Step 4: Make the Create Geology Surface window active and also click on Layer Settings and select the top selection. Now the two open panels should look similar (but probably not identical) to the picture below:



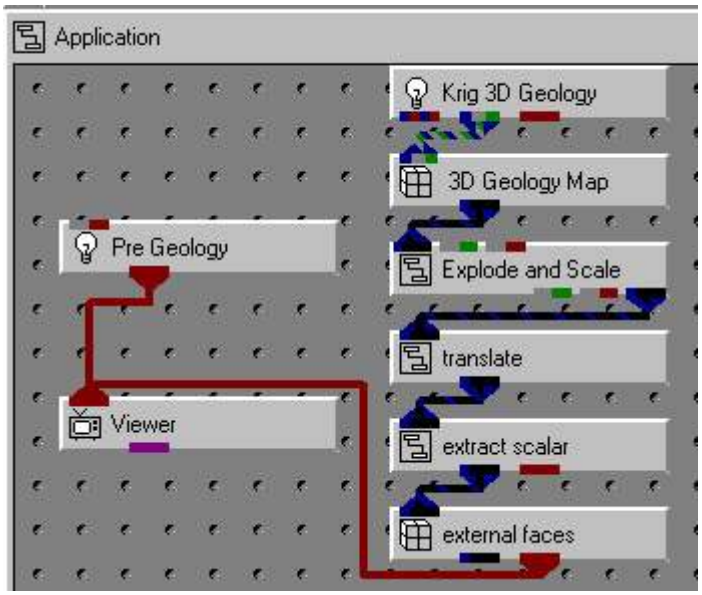
Now click the "Write" button and a file browser will appear prompting for a file name with a .gmf extension. Fill this out as test.gmf for this example and click OK. At this point the top surface of a gmf file has been created by EVS and written to a file named test.gmf. If you edit this file you should see the following text in your editor:



In order to create the second surface of your file, you must click Append on the Create Geology Surface panel, then click the 2nd selection in the Layer Settings panel. Repeat step 3 and step 4 at the intersection points between layer 1 and 2 to create the bottom of layer 1. Now edit your gmf file, and your text should look like this:



Step 5: ***Optional*** You may use the application shown below to display the 3D model resulting from your newly created gmf file (test.gmf). The non-default settings for this application are listed below just after the graphic of the module connections:



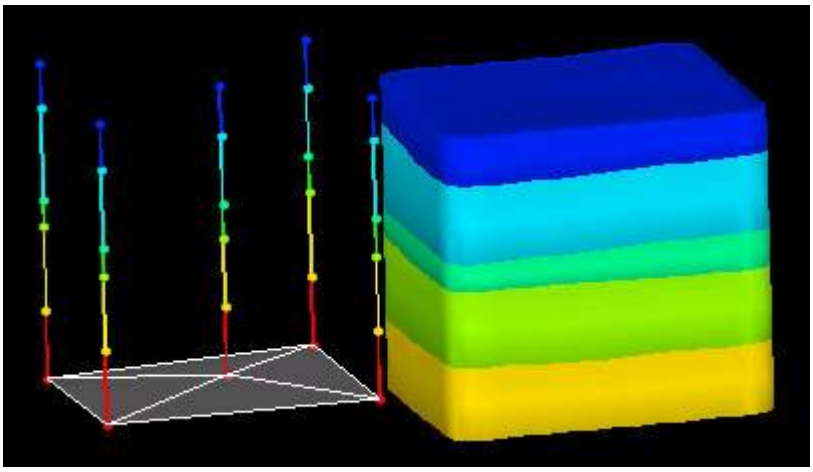
In order to achieve the view below, the following module settings must be set:

Krig_3D_Geology - Read and Accept test.gmf

Explode_and_Scale - set explode factor to "0"

Extract_Scalar - select Geo_Layer

Translate - set X Offset to 350



Note that this approach could have been applied from the outset of the process. Try this out by starting over making a file called test2.gmf, but this time read and accept test2.gmf in Krig_3D_Geology after creating each surface. This approach is very useful when testing your interpretations.

Also note that the main Pre_Geology help describes the features available for picking surface elevations which are not directly at the sphere locations.

Write_UCD

Thisis a deprecated module that has been supplanted bySave_EVS_FieldandMaterial_to_cellsets

General Module Function

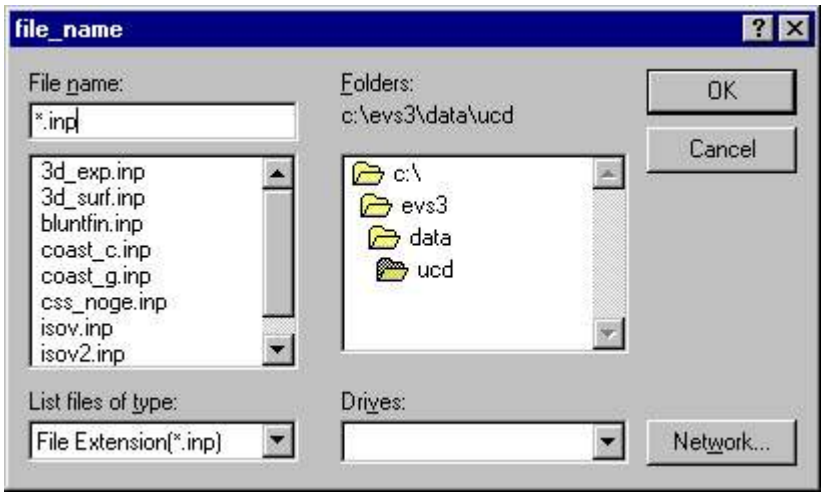
The Write UCD module creates an EVS Unstructured Cell Data (UCD) file containing all the mesh and nodal data component information sent to the input port. The data piped to the input port must be in UCD format. This module is useful for writing the output of modules which manipulate or interpolate data (Krig_3D , Krig_2D , etc.) to UCD files so that the data will not need to be processed in the future. The process data in the UCD can be read using Read_UCD, which is much faster than reprocessing the data.

Module Input Ports

The Write UCD module is shown above. The module has one input port which accepts any UCD format data.

Module Output Ports

Write UCD has no output ports.

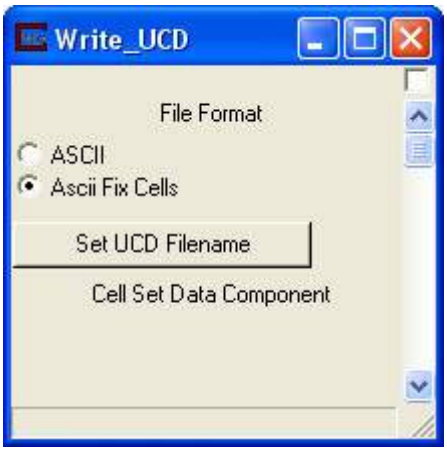


Write UCD Control Panel

The Write UCD control panel is shown above. Clicking on the Write UCD File push button opens a standard windows file browser. The filename and location can be specified in this browser. Clicking on the OK button tells Write UCD to create the UCD file. The user is not warned if the file they are creating currently exists. The file will be overwritten.

A modified version ofthis module is available in MVS, which is why it is not deprecated in MVS.

MVS has a modified version of Write_UCD that will fix cell sets corresponding to geologic layers when the field has been passed through modules such as plume_volume. Normally this causes a HEX and TET cell set to be created but merges the cell sets corresponding to geologic units. The *ASCII Fix Cells* option recreates the cell sets into Tet and Hex cell sets for each of the original geologic layers. This provides the ability to save large grids after subsetting with plume_volume, while preserving geologic layer information. The user interface is shown below.



Related Modules

-> Read_UCD

Write_DXF

General Module Function

This module has been deprecated and replaced by the write_CAD module.

Write_DXF provides EVS with vector output of graphical results. Write_DXF will output surface and line type objects created in EVS as a CAD standard DXF. Examples of compatible input are isolines, external_faces, Krig_2D, and isosurface. (Note: plume_shell does not output all of its objects through its' blue port.) Write_DXF will attempt to preserve a reasonable color range for the DXF objects output within the limitations imposed by AutoCAD. Surface output will be flat shaded (all triangles or quadrilaterals will be a single color vs. Gouraud shaded and the number of unique colors is limited). Since AutoCAD does not limit appending additional DXF files to an existing drawing, multiple objects can be output using Write_DXF several times, and all objects can be combined in a single drawing. This module preserves true user coordinates and all output is 3D. Special objects like isolines are grouped into layers by their level, but other objects are placed in a single layer.

Module Input Ports

Write_DXF has only one input port. This port accepts unstructured mesh data and nodal data. The first component of the data piped into this port must be scalar if colors are to be assigned to the output. If used with 3D data, then a slice plane or other subsetting module such as isosurface, or external_faces must be used or the output will contain the surfaces of every 3D element (cell) in the input.

Write_DXF supports POINT,LINE, POLYLINE, TRI(angle), POLYTRI(angle), QUAD(rilateral), TET(rahedron), PYR(amid), PRISM, and HEX(ahedron) cell types. Since spheres are rendered as points with radii (i.e. they have no real surfaces), spheres are not supported in this module. If 3D cell types are input the output will contain the surfaces of every 3D element (cell) in the input.

Module Control Panel

The control panel for Write_DXF is shown in the figure above. Color by Cell Type gives a unique color for each cell type. Color by Nodal Data reveals a Data Component selector if there is nodal data in the input. It should be used to color the data according to the selected Data Component. (Remember that the colors will be a fixed blue to red palette from the limited DXF and AutoCAD color palette.) The last option does not include any color data in the DXF file.

Related Modules

-> Read_DXF

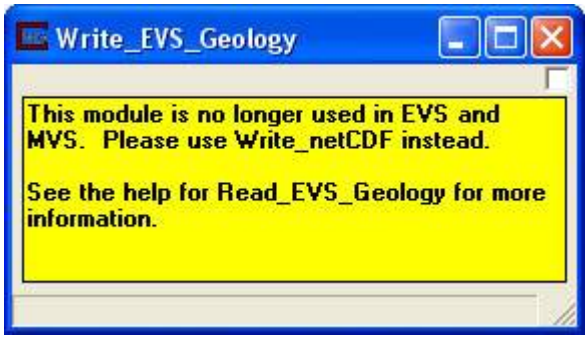
Write_EVS_Geology

This is a deprecated module that has been supplanted by Save_EVS_Field

General Module Function

Write_EVS_Geology is no longer a working module.

For more information see Read_EVS_Geology.



Related Modules

Read_netCDF

Write_netCDF

Generate Axes

This is a deprecated module that has been supplanted by axes

General Module Function

The Generate Axes module is used to place 3D axes in the viewer scaled by the model data and/or user defined limits. Generate Axes accepts data from many of the Subsetting and Processing modules and outputs directly to the viewer. Data passed to Generate Axes should come from modules which have scaled or transformed the mesh data, for example Explode_and_Scale. Axes generated by Generate Axes and displayed in the viewer are scaleable and transformable with other objects in the viewer.

The User interface to Generate Axes is very comprehensive. Each coordinate direction axis can be individually controlled. Axis labels and tick marks for each axes can be specified. The label font, label precision, label orientation, and other label parameters are all user specified. Many of the parameters do not have default values that will produce the desired results because many variables control how the axes should be defined.

Generate Axes requires a field input to position and size the axes. If you disconnect the (blue/black) field input port, you no longer lose the axes bounds values and your axes remain in place. This is useful when field data changes in an animation so that you don't constantly recreate the axes.

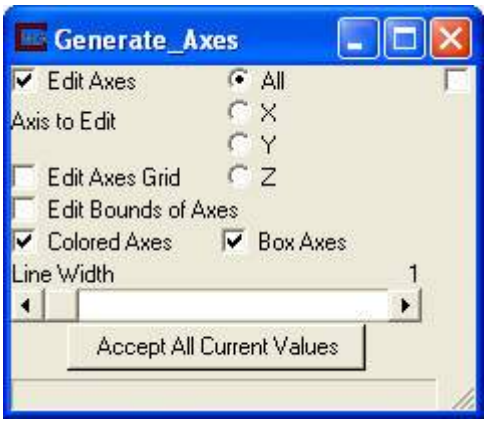
Also, the size of text and tick marks is based on a percentage of the x-y-z extent of the input field. This now allows you to set the extent of one or more axes to zero so you can have a scale of only one or two dimensions.

Module Input Ports

The Generate Axes module is shown above. The module has three input ports. The first port (closest to the left) accepts scaling data from Explode_and_Scale which contains information on the z exaggeration variable. The second port accepts explode distance data from Explode_and_Scale. If you have an explode distance set to anything but 0, Z tick labels are not printed, since it is impossible to do accurate labels for Z positioning in a general case. The third port accepts mesh data information specifying the model extents.

Module Output Ports

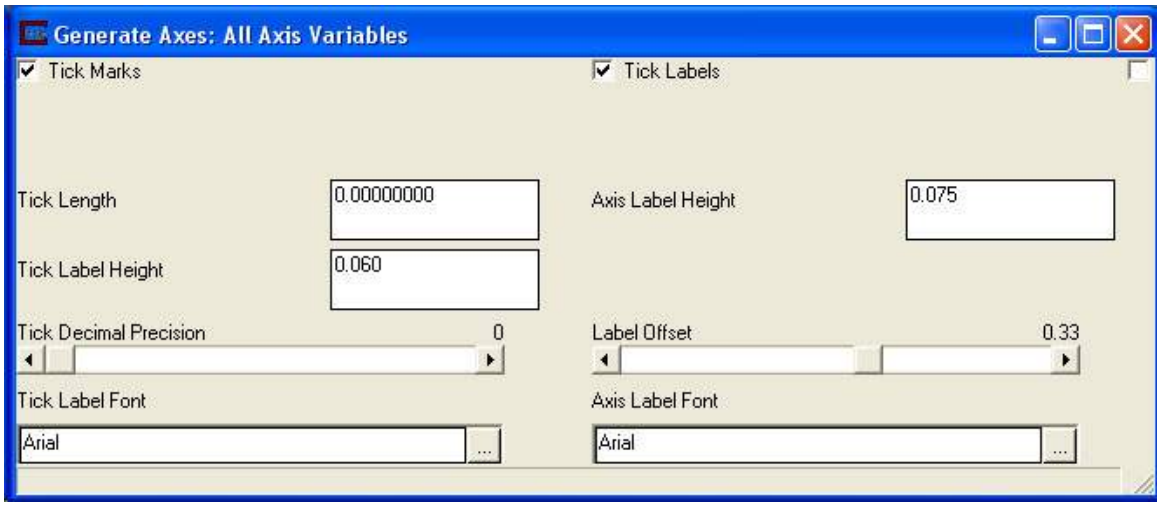
Generate Axes has one output port which sends the axes to the viewer for display.



Module Control Panel

The main control panel for Generate Axes is shown above. By default, the Edit Axes check box is not selected. By selecting this check box, four radio buttons appear on the right corresponding to all, X, Y, and Z. These radio buttons allow the user to choose which axis is currently being edited. The Edit Bounds of Axes check box bring up a control panel for specifying the bounds of the axes. The colored axes check box is used to specify if the axes should be colored depending on orientation (Red = X, Green = Y, Blue = Z) or monochromatic. The default is on (check in the box). Uncolored axes color can be edited using Objs.Props.Primary_Color in Viewer.

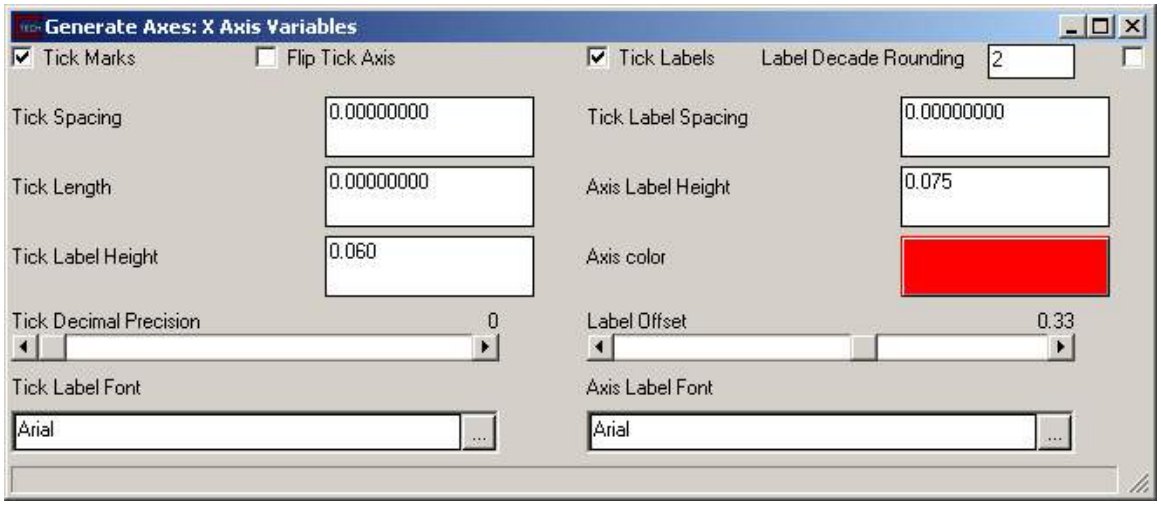
The Box Axes check box is used to specify if the axes should be drawn as a box around the data domain or as single X, Y, and Z lines. The Label Decade Rounding slider is used to specify how the data is rounded for use in labeling. For example, setting this to 2 tells Generate Axes to label axes using 2 decade rounding (100's). Therefore, a minimum "X" value of 2,378 would cause the first label to be placed at 2,400 on the axes label. If Label Decade Rounding was set to one, the label would be at 2,380. The default setting is 2. The Accept All Current Values push button is used to tell Generate Axes to send axes information to the output port using the current settings.



All Axis Variables

Selecting the All radio button, which appears when the Edit Axes check box is chosen, brings up the above control panel. A control panel identical to this will appear, along with another control panel, whenever the X, Y, or Z radio button is chosen. In the All Axis Variables axis variable control panel, the chosen parameters are defaults for each individual axis which can be overridden using the individual axis control panels. Changing an "all" variable will change the corresponding variable for all axes.

These axis variable control panels consist of check boxes, edit fields and sliders all used to control the display of axes and labels. The Tick Marks check box and Tick Labels check box are used to specify whether the tick marks and labels are displayed (on) or not (off). Both are on by default. The Tick Spacing determines the spacing interval for tick marks in model length units. Similarly, the tick label spacing determines the spacing interval tick mark labels in the same units. The default for each of these is 100.000 and the range is unbounded. By setting these two edit fields equal, every tick mark will be labeled. If Tick Label Spacing is twice the Tick Spacing, every other tick mark will be labeled. The Tick Length edit field is used to specify the length of the tick mark. The default is 10.000 and the range is unbounded. The Tick Label Height edit field is used to specify the height of the label text. The default is 0.050 and the range is unbounded. The Axis Label Height edit field is used to specify the height of the text string which labels the axis (X, Y or Z), and should not be confused with the above mentioned Tick Label Height. The default is 0.050 and the range is unbounded. The Label Font Slider is used to set the axis label font. The default is zero and the range is 0 to 6. Similarly, the Tick Label Font slider sets the font to be used by the tick labels. The default and range are the same as the Label Font slider. The Tick Decimal Precision slider sets the number of decimal places to be displayed in the tick labels. The default is 0 and the range is 0 to 5. The "Label Saturation" slider "Bright-Saturate" modifies the color saturation for values between 0 <= v <=1, and modifies the brightness of the axes (and labels) for -1 <= v < 0. Reduced brightness help axes to stand-out against a white background or to be less pronounced against a black background.



Choosing the X, Y or Z radio button in the main control panel brings up a panel similar to the one shown above. For example, if X is chosen as was the case in the figure above, the title of the axis variable control panel changes to Generate Axes: X Axis Variables and the above control panel box appears. In this control panel, the location and orientation of the labels are controlled. This control panel consist of all push button and one edit field.

The justification push buttons are used to set the location of the label relative to the end of the tick mark to be labeled. The layout of these push buttons corresponds to the location of the tick mark relative to the labels location. The label can exist in any of nine locations relative to the tick mark. For example:

- Selecting MC (middle center) the center of the label is placed directly over the end of the tick mark.

- Selecting UC (upper center) places the center of the label directly below the end of the tick mark (tick mark is above the center of the label).

- Selecting LL (lower left) places the left end of the label just above the end of the tick mark.

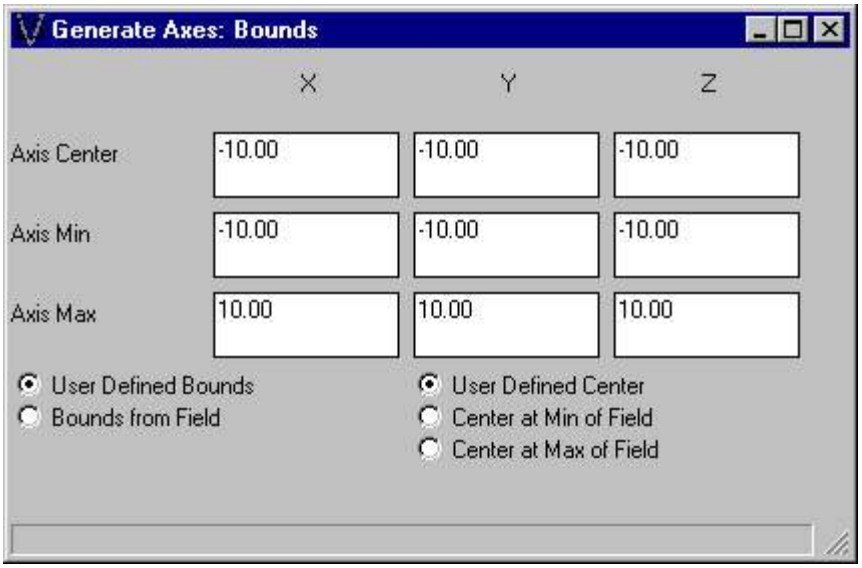
- Selecting MR (middle right) places the right end of the label directly to the left of the tick mark.

The other justification push buttons act in a similar manner.

The Orientation push buttons are used to specify which axis the labels should parallel. These push buttons are used in conjunction with the Label Plane push buttons. The Orientation push buttons make more sense once the Label Plane push buttons have been explained. The Label Plane push buttons define which plane the label is to reside in. There are 6 possible planes corresponding to the six faces of a cube. The planes are defined as parallel to the axes in the push button name with the order of the plane names defining the normal vector or positive side of the plane (for example, the XY label plane is parallel to the X and Y axes and the normal vector is in the positive Z direction). The normal vector or positive side of the plane determines whether a label is forward and right side up when viewed from above or below. Once the plane for the label is chosen, the orientation of the label can be defined. In any given label plane, there are four possible label orientations. For example, in the XZ plane the labels can be parallel to the +X, +Z, -X or -Z axis. +X will cause the labels to parallel the X axis in the XZ plane and read from left to right, right side up. -X create labels parallel to the -X axis which, when viewed from the Azimuth 180 - elevation 0, will appear upside down.

The orientation of the label also affects how the justification push buttons affect the label. The justification push buttons move the text around a label box relative to the end of the tick mark. The label box is oriented parallel to the orientation push button selection. Therefore, if +Z orientation is chosen, the UC justification location appear to move the label to the middle left of the tick mark in Cartesian coordinates, when viewed normal to the XZ plane. The LL will appear to move the label to the lower right of the tick mark when viewed normal to the XZ plane.

The Axis Label edit field is used to specify the text that will appear as the axis label for the axis being edited. The default is the Cartesian axis specifier for the axis being edited (For example, X for the X axis). This label can be any text string and can contain spaces and special characters.



Bounds Control Panel

The Edit Bounds of Axis control panel is shown above. This control panel is opened and closed by checking on the Edit Bounds of Axes check box on the main control panel. Radio buttons at the bottom of the control panel are used for automatically specifying the bounds of the axes. By default, the User Defined Bounds and User Defined Center radio buttons are chosen. The default values in the Axis Center edit fields are -10.00 for X , Y and Z. The default Axis Min values are -10.0 for X, Y and Z and the default Axis Max values are 10.0 for X, Y and Z.

Choosing the Bounds from Field radio button puts the minimum and maximum X, Y and Z coordinates in the data field passed to the input in the Axis Min and Axis Max edit fields, respectively. The Axis Center edit field will not be changed.

The radio buttons corresponding to Center at Min of Field and Center at Max of Field controls the location of the origin of the axes. Center on Min of Field places the origin of the axes at the minimum coordinates of the data field. Similarly, the Center at Max of Field places the origin at the maximum coordinates of the data field. The user can change any of these fields after they are updated.

Related Modules

-> Viewer

TrueScale

You should use Explode_and_Scale in all applications instead of TrueScale.

General Module Function

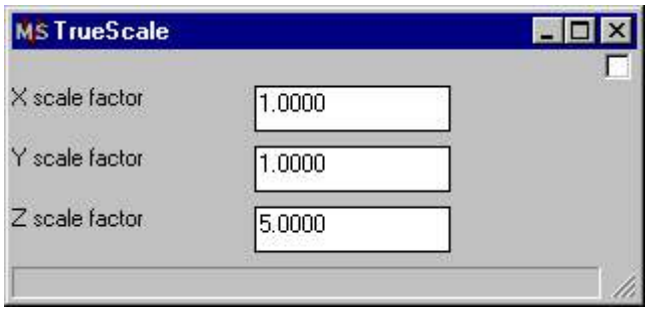
The TrueScale module is used to scale the coordinates in a mesh in the X, Y and Z directions independent of each other. Nodal data is not scaled by the TrueScale module. TrueScale does affect the nodal coordinates.

Module Input Ports

TrueScale has only one input port. Input to this port must contain mesh data of any type mesh. Nodal data can be input to scale.

Module Output Ports

TrueScale has two output ports. The first output port (closest to the left) outputs the same mesh data as the input mesh with the transformation matrix containing the scaling information. Nodal data piped to the input port passes through scale unaffected. The second output port is not used in EVS.



Module Control Panel

The control panel for TrueScale is shown in the figure above. Edit fields are available for adjusting the X scale factor, Y scale factor, and Z scale factor. The default for each of these edit fields is 1.0 and the available range is -1.0e+08 to 1.0e+08. The numbers entered into the edit fields represent a scale factor relative to the viewer's dimensions coordinate system.

Related Modules

Explode_and_Scale

Label_Probe

(This module has been deprecated. Its functionality has been surpassedby the interactive_labelsmodule.)

General Module Function

Label_Probe is an enhancement to the probe module which displays the data values (on the control panel only) for the data present at the probe object's location in a mesh. The Label_Probe enhancement is that it contains extra controls in the control panel for displaying coordinates (on/off), log transforming the data being probed, etc. The result is a text output in the viewer window of the probe value AND coordinates (optional). Like the probe module the basic function of Label_ Probe is that of clicking with the mouse (alt left mouse button) on any portion of an object and receiving a report of the objects scalar value. Note: The mouse action for activating a probe's

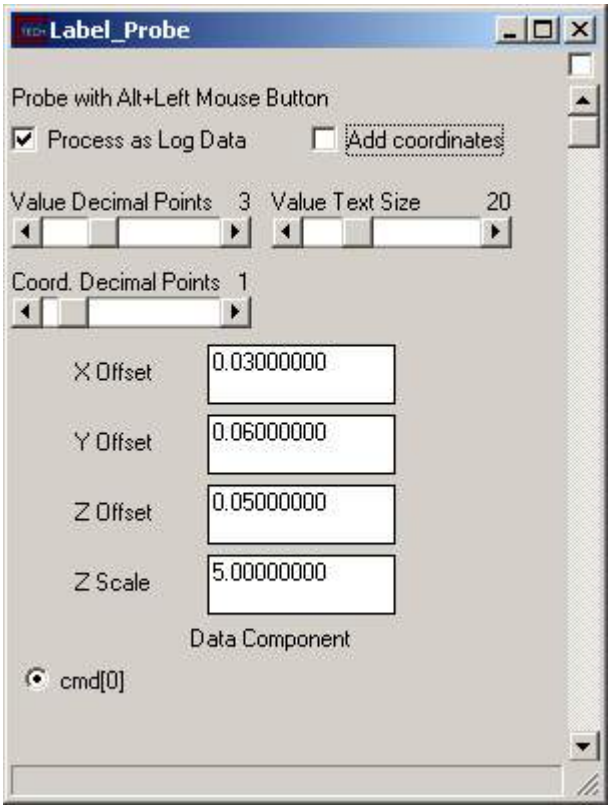
location is that of depressing the Alt key whilst depressing the left mouse button.

Module Input Ports

Label_Probe has three input ports. The first (leftmost) input port should be a field with any mesh type and Node_Data. The second input port must be attached to the red output of the object of interest. This red port connection is required since the probe's location comes from clicking (alt left mouse) within the rendered view. The last port is the z-exaggeration factor of the object being probed.

Module Output Ports

Label_Probe has two output ports. The left port sends a text output to the screen of the probe information. The last port is the z-exaggeration factor.



Module Control Panel

The user interface is shown in the figure above. Each parameter function is described below:

The Process as Log Data toggle activates the conversion of log-space values to normal units.

The Add Coordinates toggle activates the placement of text (beneath the value and in parentheses),

The Value Decimal Points slider allows a choice of decimal precision for the value only (default is 3).

The Coord. Decimal Points slider allows a choice of decimal precision for the coordinates only (default 1).

The Value Text Size slider allows sizing of the text (default 20).

X Offset, Y Offset, Z Offset: Specifies the offset distance from the probe location. A value of 0.5 in X or Y offsets the text half the screen length. For example, if the probe location is at the center of the viewer window, the text would start at the very edge of the viewer screen. The Z offset places the text nearer or farther from the camera view. The defaults are X Offset = 0.03, Y Offset = 0.06 and Z Offset 0.05.

The data component radio buttons allow for choosing which of the input field's components to probe. The component can be scalar or vector. The default is the first (0th) component. If node data labels are present, they are displayed.

Related Modules

-> probe

extract_component

This is a deprecated modulereplaced by select_data.

General Module Function

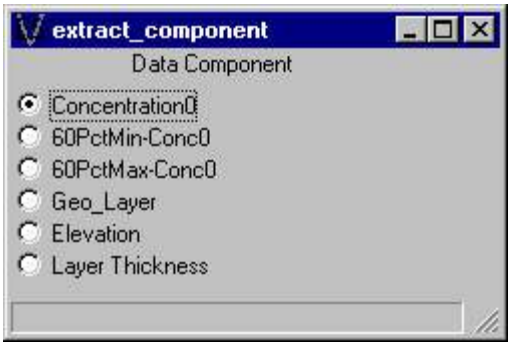
The extract component module extracts a single data component from a field. Extract component can extract scalar data components or vector components. Scalar components will be output as scalar components and vector components will be output as vector components.

Module Input Ports

Extract component has only one input port which accepts any nodal data. The input may contain mesh data. If mesh data is include in the input then it will pass through extract component unaffected.

Module Output Ports

Extract component has two output ports. The first port (closest to the left) outputs a new nodal data component containing only the extracted nodal data component. If mesh data was included in the input, it will be contained in the output. The second port will output a renderable object if the data are faces or lines.



Module Control Panel

The control panel for extract component is shown in the figure above. The data component radio button list contains all of the data components piped

into the input port. Only one component can be selected at a time and the default selection is the first (0th) data component. Any modules downstream of extract component will receive only the selected data component.

Related Modules

->extract_scalar

-> combine_comp

-> combine_vect

extract_scalar

This is a deprecated modulereplaced by select_data.

General Module Function

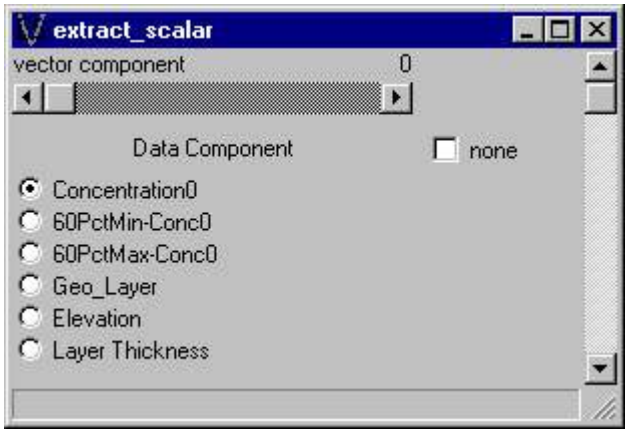
The extract scalar module extracts scalar nodal data components from a field's scalar and/or vector data components. If the input to extract scalar contains only scalar data, then the scalar data will be passed through for the data component selected. When only scalar data is present in the input, it is always preferable to use the extract_component module. Extract_scalar will cause a redetermination of the input data limits. This is usually not desirable. This module is used to select one scalar data component to be passed downstream in a network.

Module Input Ports

Extract scalar has only one input port which accepts mesh and nodal data. The mesh data in the input passes through extract scalar unaffected.

Module Output Ports

Extract scalar has two output ports. The first port (closest to the left) outputs scalar nodal data for the data component specified. Again, the mesh data (the coordinates of the mesh cells and vertices) are contained in the output. The second port (red) is a renderable object if the input is faces or lines (not volumetric elements). The rendered object from the second port will be colored by the scalar data component selected in Extract_Scalar.



Module Control Panel

The control panel for extract scalar is shown in the figure above. The slider for vector component is used to identify which component of the data vector

is to be extracted. The data component radio button list contains all of the data components piped into the input port. If a data component is a scalar (i.e., contains only one scalar value), the vector component slider will not allow the user to pick any value other than zero. Nodal data supplied to the output port consists of the nodal data for the selected data component only.

If the data component is a vector (i.e., it contains more than one data component), the slider will allow the user to pick any value from 0 to the i^{th} data component; where i=number of vector components minus 1. For example, with a 3 component vector (such as ground water velocity) [V_x, V_y , V_z]; V_x would be the 0th component, V_y the first component and V_z the second component. If the nodal data for V_x was selected for output, the data would consist of the scalar values of V_x.

Related Modules

->combine_comp

-> **combine_vect**

-> **extract_component**

Fence_Merge

General Module Function

The fence merge module is used to merge the output from multiple Krig_Fence modules into one data set (i.e., to merge cross sections into a fence diagram). This is useful for performing uniform data manipulation procedures on fence data from several Krig_Fence outputs. For example, if several Krig_Fence modules are used, they should all pass through a fence merge module before being passed to explode and scale. Therefore, all fences will be exploded and scaled the same amount and only one dialog box is needed to control all fences. Fence merge should always be used when more than one krig_fence module is used. There are no sub-windows in this module and it will not appear in the modules pull down menu.

Module Input Ports

Fence merge has one input port which can accept data from an unlimited number of krig_fence modules. All inputs are piped into the same input port.

Module Output Ports

Fence merge has one output port which contains a reference to all fence data sent to the input port. The fence data is unaltered in the output.

The user interface for this module is shown above.

Related Modules

->Krig_Fence

threshold

This is a deprecated modulereplaced bysubset_cells.

General Module Function

The threshold module is a subsetting module used to eliminate whole cells from the input mesh with nodal values that are above and/or below a given threshold value. If any nodal value of a cell is above and/or below the threshold criteria, then the whole cell is eliminated. This is in effect a cell based version of plume_volume, which outputs a blocky mesh of cells bounded by the threshold value. Nodes in cells which have been eliminated are set to a user defined value (the Null value) which allows the user to perform other operations with the excluded set of cells.

Threshold can be used in two ways:

1. It can look at the values in just one nodal data component, and set them to the null_value if they exceed the max check value or min check value (threshhold component = check component). This is its most common use. Or,

2. It can look at the values in one scalar component (check component) to see if they exceed the max check value or min check value. If they do exceed the threshold bounds, threshold sets the corresponding array index value in a second, *different* data component (threshold component) to the null_value. This allows the visualization of the distribution of one component within the threshold values of of another component, and is essentially the equivalent of mapping a second component within an isosurface of the first component.

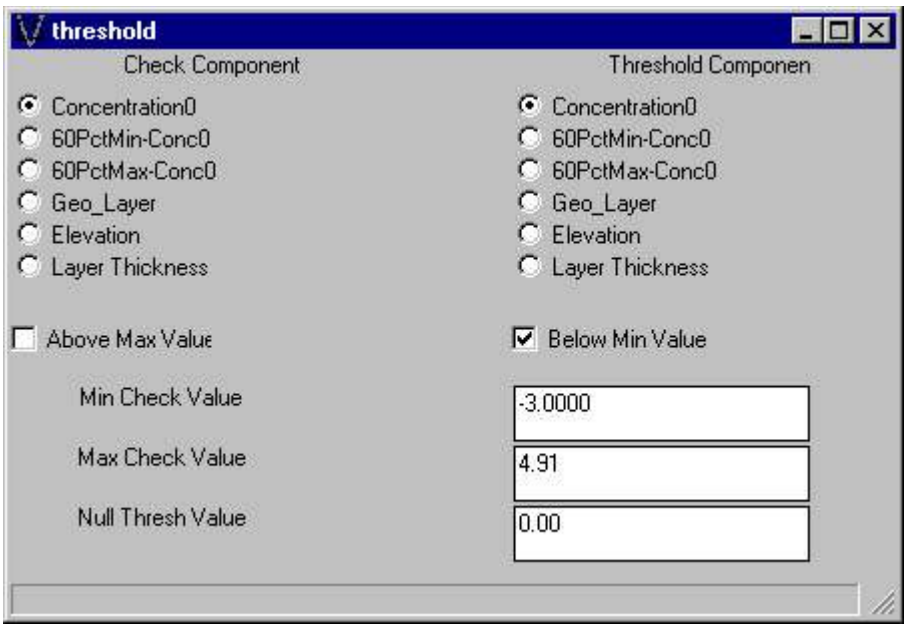
This second behavior is useful, for example, if you use one component to flag the visibility of a cell (for example, 0 = invisible, 1 = visible) within the threshold values of another component.

Module Input Ports

Threshold has only one input port. Input sent to this port must contain nodal data. If mesh data is piped to this port it will pass through threshold unchanged. The field's data component can be scalar or vector. If it is vector, only the first data component of the vector is considered.

Module Output Ports

Threshold has two output ports. The first output port (closest to the left) outputs only the selected threshold component with the nodal data values replaced with the null value where they exceeded the threshold criteria. If a mesh is passed to the threshold input it will pass to the output port unchanged. The second port outputs a renderable geometry if a mesh is present.



Module Control Panel

The control panel for threshold is shown in the figure above. The check component radio buttons determine which data component will be considered in threshold. The threshold component determines which data component will be affected by threshold. The Above Max value and Below Min Value check boxes determine if the data will be thresholded above the maximum value selected and/or below the minimum value selected. The Min Check Value edit field determines the minimum threshold check value, and the Max Check Value edit field determines the maximum threshold check value. The Null Thresh Value is the value assigned to any nodes that do not meet the threshold criteria.

The check component value determines which data component is to be considered by threshold. By default, the first (0th) component is selected. When a data component is selected under Check Component, it's nodal data values are checked against the Min Check Value and Max Check Value. Then, depending on if Above Max Value and/or Below Min Value are checked, the cells containing nodal data not meeting the threshold criteria are assigned the null value, and can be eliminated.

The Threshold Component radio button determines which data component is to be sent to the output ports. The default selection is the first (0th) data component. If this data component is different than the Check Component, then the Threshold Component is subsetted based on the Check Component and the selected criteria. If the Check Component and Threshold Component are the same data component, then the resulting output is the selected data component subject to the thresholding criteria.

The Above Max Value and Below Min value check boxes determine whether the data is to be thresholded above the selected Max Check value and/or below the Min Check Value. By default, the Above Max value is off (not checked) and the Below Min Value is on (checked). If both the Above Max Value and Below Min Value check boxes are selected, the resulting output is

the data component selected under Threshold Component subjected to both the Min Check Value and Max Check value criteria (similar to the output of the contour module, only cell based). If neither of the check boxes are selected, the resulting output is the data component selected under Threshold Component subjected to no thresholding.

The Null Thres Value is the value assigned to all nodes in cells containing any node that failed to meet the thresholding criteria. The default Null Thres value is 0.0.

Related Modules

->plume_volume

-> contour

Read_Field

This is a deprecated module that has been supplanted byLoad_EVS_Field

General Module Function

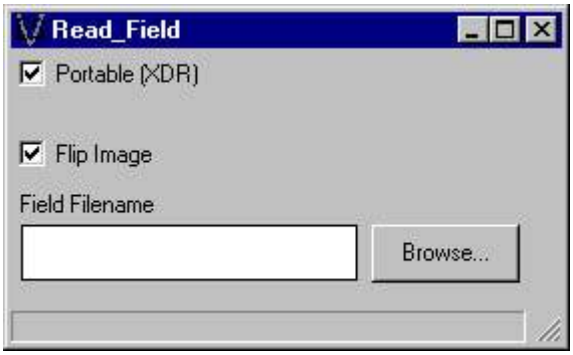
The Read Field module is used to read EVS field files, which generally have a *.fld* filename extension. An EVS field file contains structured cell geometries with implied connectivity and nodal or cell based data components. The cell geometries generally consist of rectilinear cells similar to a finite-difference grid. Field files are not currently created within EVS. The field file format is discussed in the section below called Field File Formats Because modules in the current version of EVS use UCD data types, the Read_Field module is not utilized often.

Module Input Ports

The Read Field module is shown above. Read Field has no input ports. It obtains the Field input data by reading a file with a file browser.

Module Output Ports

Read Field has two output ports. The first port (closest to the left) passes the mesh and data components to other modules which accept Field data types. Data passed from Read Field is usually sent to modules in the Subsetting and Processing Libraries. The second port is used to send Field file data directly to the viewer for rendering.



Module Control Panel

The user interface for Read Field consists of two toggles and a file browser shown above. The Portable XDR toggle (checked by default) specifies whether the data is in the machine-dependent XCR format. With this parameter enabled, and when binary data is being read in, the machine assumes that the data is in XDR format. Note that these types of fields are not created by EVS, but may be available through outside sources and are thus accommodated by this option. The Flip Image toggle specifies whether image data is inverted before it is output. Image data could be expressed as a field and is typically described with the upper-left corner as the starting point. As a result, when it is read in as field data, the image may appear flipped vertically. The default is "ON" which usually interprets EVS fields properly.

The file browser is used for selecting the Field file (Field files have a *.fld* extension). The default directory path is the \EVS\DATA\FIELD. Double clicking on a filename or selecting a file name and choosing OK reads the selected Field file into memory and closes the file browser. After a Field file has been read into memory, the mesh and data components are passed to any downstream modules in the network

Field File Formats

This is a description and some examples for the EVS field data format. The field format can be binary or ASCII files. These examples are in ASCII format. The standard field format is more memory efficient then the UCD format, but does not allow exploding geologic layers for visualization purposes. Otherwise it is probably better for Finite Difference code output.

Note: in EVS, coordinates apply to nodes (versus cells). If you write out the coordinates of cell centers (such as with block-centered flow data: MODFLOW), EVS will visualize these as nodes at the corners of hexahedral volume cells. To be most correct, MODFLOW models should be output with one extra node in all three axes. However, the compromise is generally acceptable. Note that the Groundwater Vistas Pre-Processor creates the 'correct nodal' corners of the s mesh and outputs a .fld and .dat file which can be read directly into EVS.

Generally, the .fld file describes the structure(s) of the data file(s) to be read and also describes how to read them. For example, if you want EVS to read a .dat file with coordinates you specify a combination of skip, offset and stride to read the x, y, then z values in the .fld file. Simply put, *'skip'* specifies the number of *lines* to skip before reading data; *'offset'* specifies *which value* in the line to read (e.g. offset of '0' means read 1st value, offset of '1' means read second value, etc.); *'stride'* specifies how many *values to jump* before starting to gather data again (e.g. for a file with x,y,z coords. the stride would be 3....if there is x,y,z and value the stride would be 4.

The files below are for reading a file which contains only coordinates AND a file which contains only data values. This is quite powerful in that you may generate only one coordinate file, then keep changing/adding new data files. A simpler yet less robust example would involve one .dat file which contains the coords AND the data component(s) in one file. The two .fld files are to

show that you may have multiple components....note that three of the components are velocity vector.

File list and descriptions:

mt3d_smp.fld Simple field header file to read only one data component (head). This file defines the files that contain coordinate data and nodal parameter data. In other words, this tells EVS how to read the data file.

The text below is from the file: mt3d_smp.fld

```
# AVS FIELD the string "# AVS/Express" must be the first five
# characters in the file.
ndim=3 # REQUIRED - the number of dimensions
dim1= 51 # REQUIRED - dimension of axis 1
dim2= 31 # REQUIRED - dimension of axis 2
dim3= 8 # REQUIRED - dimension of axis 3
nspace=3 # REQUIRED - coordinates per point
veclen=1 # REQUIRED - components at each point
data=float # REQUIRED - data type
field=irregular # REQUIRED - field type
label=head # OPTIONAL - label for variable 1
unit=feet # OPTIONAL - unit label for variable 1
#
# For each value in the vector: data reading instructions
#
variable 1 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 1 stride = 13
#
# For each coordinate X, Y, and Z: data reading instructions
#
coord 1 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 1 stride = 4
coord 2 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 2 stride = 4
coord 3 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 3 stride = 4
```

**********End of text**********

Note: Any characters following (and including) # in a header line are ignored.

mt3d_all.fld Simple field header file to read all 12 data components (last 3 are vel. vector). This file defines the files that contain coordinate data and nodal parameter data.

The text below is from the file: mt3d_all.fld

```
# AVS FIELD
ndim=3
dim1= 51
dim2= 31
dim3= 8
```

```
nspace=3
veclen=12
data=float
field=irregular
label=head drawdown thickness Geolayer Elevation Conductivity Storage Concentration
Change_in_Conc Velocity_x Velocity_y Velocity_z
unit=feet feet feet number feet feet/day 1/feet mg/kg mg/kg feet/day feet/day feet/day
variable 1 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 1 stride = 13
variable 2 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 2 stride = 13
variable 3 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 3 stride = 13
variable 4 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 4 stride = 13
variable 5 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 5 stride = 13
variable 6 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 6 stride = 13
variable 7 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 7 stride = 13
variable 8 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 8 stride = 13
variable 9 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 9 stride = 13
variable 10 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 10 stride = 13
variable 11 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 11 stride = 13
variable 12 file =mt3d_smp.dat filetype=ascii skip= 0 offset = 12 stride = 13
coord 1 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 1 stride = 4
coord 2 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 2 stride = 4
coord 3 file =mt3d_smp.crd filetype=ascii skip= 0 offset = 3 stride = 4
```

**********End of text**********

mt3d_smp.crd Field coordinate data file (first column (node number) is not needed so offset = 1, 2, or 3) Coordinates can/should be in user's original coordinate system. If model was rotated (to be run finite difference) and/or translated, it should be put back in real coordinates. This allows for incorporating simulation data with measured data results and incorporating CAD drawings drawn in real coordinates. In this example, the first column is the node number and is not needed in the file (so we offset to skip over it during reading).

Note: numerous lines were dropped from this example.

The text below is from the file: mt3d_smp.crd

```
1 9625.112 10786.380 6.275
2 9714.992 10742.550 6.367
3 9804.871 10698.710 6.581
4 9894.751 10654.870 6.939
5 9984.630 10611.030 7.581
6 10074.510 10567.200 8.159
7 10164.390 10523.360 8.808
8 10254.270 10479.520 9.420
9 10344.150 10435.690 9.922
10 10434.030 10391.850 10.080
.
.
.
.
12640 12084.940 6248.842 -36.863
12641 12174.820 6205.005 -36.843
12642 12264.690 6161.168 -36.822
```

```
12643 12354.570 6117.331 -36.803
12644 12444.450 6073.494 -36.562
12645 12534.330 6029.657 -36.317
12646 12624.210 5985.820 -37.157
12647 12714.090 5941.983 -38.017
12648 12803.970 5898.146 -38.020
```

**********End of text**********

mt3d_smp.dat Field nodal data file (first column is not needed so offset = 1, 2, etc.). This file contains the calculated data from the model. In this example, the first column is the node number and is not needed in the file (so we offset to skip over it during reading).

Note: numerous lines were dropped from this example file to keep it small.

The text below is from the file: mt3d_smp.dat

```
1 .890 .000 19.1 0 6.28 15.0 .250 .000 .000 .000 .535E-02 .000
2 .890 .000 19.6 0 6.37 15.0 .250 .000 .000 -.271E-02 .425E-02 .000
3 .895 .000 20.6 0 6.58 15.0 .250 .000 .000 -.271E-02 .817E-02 .000
4 .900 .000 21.2 0 6.94 15.0 .250 .000 .000 .000 .774E-02 .000
5 .900 .000 21.2 0 7.58 15.0 .250 .401E-41 .000 -.247E-02 .672E-02 .000
6 .905 .000 21.3 0 8.16 15.0 .250 .811E-37 .000 -.487E-02 .897E-02 .000
7 .915 .000 21.9 0 8.81 15.0 .250 .532E-33 .000 -.470E-02 .101E-01 .000
8 .925 .000 22.0 0 9.42 15.0 .250 .217E-29 .000 -.452E-02 .107E-01 .000
9 .935 .000 22.1 0 9.92 15.0 .250 .000 .000 -.445E-02 .108E-01 .000
10 .945 .000 22.6 0 10.1 15.0 .250 .122E-21 .000 -.718E-02 .116E-01 .000
.
.
.
.
12638 .264 .406 2.36 4 -40.9 1.00 .100E-05 .124E-20 .000 .223E-02 -.549E-03 .905E-03
12639 .218 .456 2.37 4 -41.2 1.00 .100E-05 .111E-18 .000 .248E-02 -.791E-03 .836E-03
12640 .165 .513 2.42 4 -41.2 1.00 .100E-05 .726E-15 .000 .275E-02 -.111E-02 .123E-02
12641 .107 .573 2.51 4 -41.0 1.00 .100E-05 .484E-12 .000 .304E-02 -.153E-02 .147E-02
12642 .409E-01 .643 2.57 4 -40.7 1.00 .100E-05 .129E-09 .000 .340E-02 -.208E-02 .165E-02
12643 -.354E-01 .724 2.80 4 -39.8 1.00 .100E-05 .103E-07 .000 .373E-02 -.281E-02 -.490E-03
12644 -.123 .818 2.92 4 -39.2 1.00 .100E-05 .325E-06 .000 .388E-02 -.376E-02 -.452E-02
12645 -.218 .919 2.85 4 -39.3 1.00 .100E-05 .282E-05 -.331E-04 .374E-02 -.490E-02 -.813E-02
12646 -.311 1.02 2.84 4 -38.9 1.00 .100E-05 .122E-04 -.325E-03 .323E-02 -.613E-02 -.112E-01
12647 -.397 1.11 2.79 4 -38.7 1.00 .100E-05 .221E-04 -.814E-03 .260E-02 -.744E-02 -.127E-01
12648 -.466 1.19 2.68 4 -38.4 1.00 .100E-05 .337E-04 -.132E-02 .152E-02 -.838E-02 -.177E-01
```

**********End of text**********

Related Modules

->Read_UCD

crop

Thedeprecated crop moduleis superceded by the newer crop_and_downsize module.

General Module Function

The crop module is used to subset an image, or structured 1D, 2D or 3D mesh (an EVS "field" data type). Similar to cropping a photograph, crop sets ranges of cells in the I, J and K directions which creates an orthogonal subset

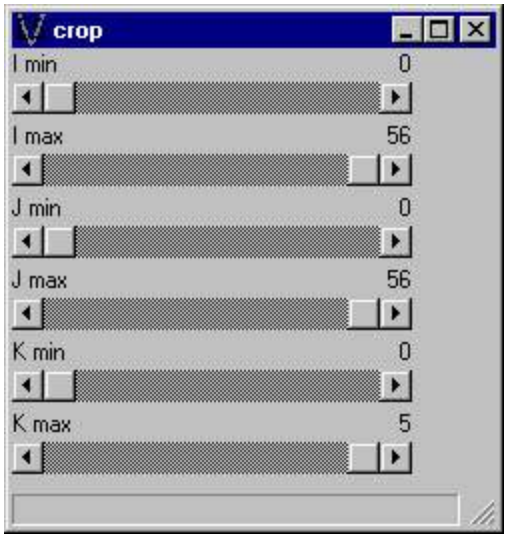
of the data. When used on an image (which only has two dimensions), crop removes pixels along the two edges of the image.

Module Input Ports

Crop has only one input port. Input to this port must contain structured mesh data and nodal data.

Module Output Ports

Crop has two output ports. The first output port (closest to the left) outputs a structured mesh data and nodal data of the cropped region. The second port outputs a renderable geometry.



Module Control Panel

The control panel for crop is shown in the figure above. Sliders are used to select the I, J, and K dimension crop ranges. Each dimension range has a slider to select the min value and a slider to select the maximum value. For example, to crop the structured mesh in the I coordinate direction from 10 to 22, set the I min slider to 10 and the I max slider to 22. The slider selections are inclusive. The default min and max for each coordinate direction reflect the min and max of the mesh data.

The ***Immediate*** toggle causes the display to update **as you move the slider(s).**

Related Modules

-> cut

-> downsize

combine_comp

This module has been deprecated and superceded bycombine_componentsandcombine_geology

General Module Function

The combine comp (combine components) module is used to create a new set of nodal data components by selecting components from two separate

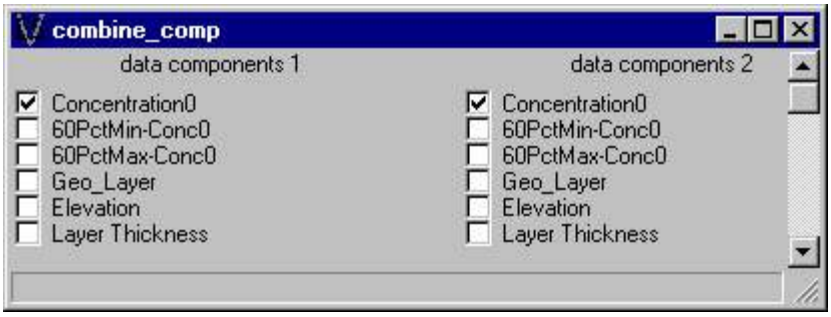
input data fields. Mesh data from the first input field, if supplied, will be the only mesh data retained in the output. However, the two data fields must have the same scale and origin, and/or number of nodes in the mesh, in order for the output data to have spatial meaning. This module is useful for assimilating data contained in two separate UCD files, or from two different Kriging modules.

Module Input Ports

Combine comp has two input ports. Both of these input ports must contain nodal data fields. If a mesh is present in the first input port data, it will be preserved in the output. Mesh data in the second input port data will be ignored.

Module Output Ports

Combine comp has two output ports. The first output port (closest to the left) outputs a new set of nodal data components reflecting the nodal data components combined from the two input data fields. If mesh data was supplied by the first input field, it will be preserved in the output. The second port (red) outputs a renderable object if the input is faces or lines.



Module Control Panel

The control panel for combine comp is shown in the figure above. Two columns of check boxes are displayed. The left column contains all of the data component fields passed to the first input port. Likewise, the right column contains all of the data components passed to the second input port. A check in the box next to any of these data components indicates a component to be included in the output. By default, the first (0th) data component in each column is selected and all other data components are not selected. Components having the same nodal component label will cause the output to have multiple nodal component labels that are the same. For example, if geo_layer is present and selected in both data component columns, the output will have two geo_layer data components.

Related Modules

->combine_vect

-> extract_component

-> extract_scalar

mirror

General Module Function

mirror generates a mesh that is the mirror image of the input mesh by reflecting the mesh about the X, Y, and/or Z plane. You can control each independently. Mirroring is useful for visualizing axisymmetric problems.

Module Input Ports

mirror has one input port which can contain any mesh. Any Node_Data that is present is ignored.

Module Output Ports

mirror has two output ports. The first output port (closest to the left) outputs a reference to the input field, but with a new, modified xform transformation matrix object. The second port outputs a renderable version of the output field.

Module Control Panel

The control panel for mirror is shown in the figure above. The toggles for mirror X axis, Y axis and Z axis are on by default. When on, their respective axes are mirrored.

tile_south

This module has been deprecated and replaced by the tile_wall module.

General Module Function

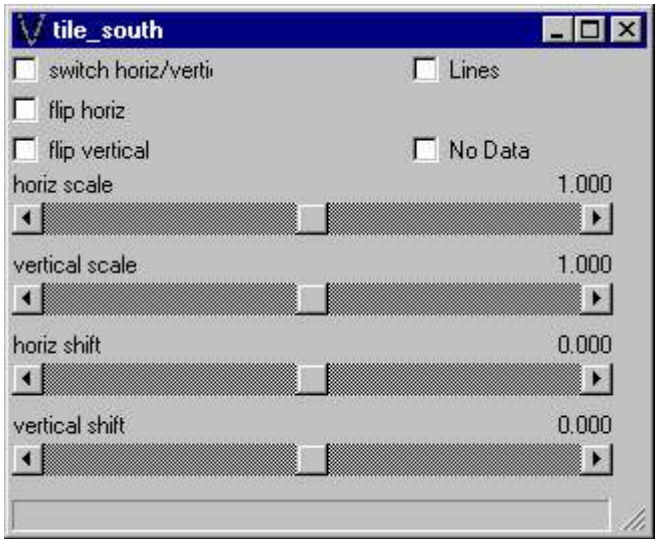
tile_south is a variation on the tile_wall module in that it provides a means to (texture map) project a single image onto north and south facing vertical surfaces such as walls of buildings to add more realism to your visualizations. This is different than tile_wall in that the tiling of a small bitmap many times over a surface is not performed. Instead, a single bitmap image of a building wall can be placed onto the surface of interest. Note that this image may actually be drawn or created in some other software (such as Microsoft Paint, Corel Draw, etc.) to create the proper look and feel (instead of photographing the building). Perhaps, if the building has a company logo, this icon could be placed onto the photo in Microsoft Paint, and then texture mapped onto the wall.

Module Input Ports

tile_south has two input ports. The left input port should be surface data (such as 3DFACES) from the Read_DXF module, and the right input port should be an image from the Read_TGA_BMP module. You should try to segregate your DXF files into one with east of west facing vertical walls for tile_south and one with roofs for texture_mesh.

Module Output Ports

The tile_wall module has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version which can hook directly to the Viewer.



Module Control Panel

The control panel for tile_south is shown in the figure above.

tile_west

This module has been deprecated and replaced by the tile_wall module.

General Module Function

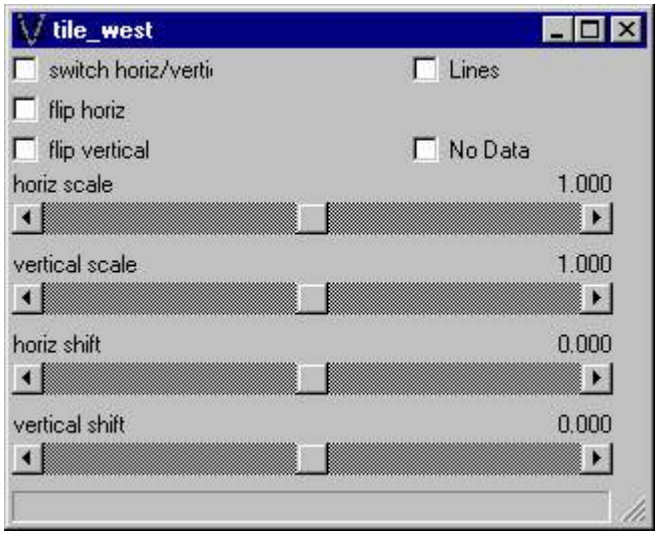
tile_west is a variation on the tile_wall module in that it provides a means to (texture map) project a single image onto east and west facing vertical surfaces such as walls of buildings to add more realism to your visualizations. This is different than tile_wall in that the tiling of a small bitmap many times over a surface is not performed. Instead, a single bitmap image of a building wall can be placed onto the surface of interest. Note that this image may actually be drawn or created in some other software (such as Microsoft Paint, Corel Draw, etc.) to create the proper look and feel (instead of photographing the building). Perhaps, if the building has a company logo, this icon could be placed onto the photo in Microsoft Paint, and then texture mapped onto the wall.

Module Input Ports

tile_west has two input ports. The left input port should be surface data (such as 3DFACES) from the Read_DXF module, and the right input port should be an image from the Read_TGA_BMP module. You should try to segregate your DXF files into one with east of west facing vertical walls for tile_west and one with roofs for texture_mesh.

Module Output Ports

The tile_wall module has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version which can hook directly to the Viewer.



Module Control Panel

The control panel for tile_west is shown in the figure above.

texture_dxf

General Module Function

This module is actually a macro module (e.g. a collection of three standard modules). texture_dxf combines Read_DXF, Read_TGA_BMP, and texture_mesh into a single module. It is equivalent to the following network fragment:

Module Input Ports

texture_dxf has no input ports.

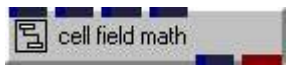
Module Output Ports

texture_dxf has one output port which outputs a renderable geometry directly to the Viewer.

Module Control Panel

The control panels for texture_dxf are the same as texture_mesh, Read_DXF and Read_TGA_BMP.

cell_field_math



This module has been deprecated as it has been superseded by cell_data_math.

General Module Function

The cell_field_math module (only in EVS PRO and MVS) is used to perform mathematical operations on cell data fields. Up to four data fields can be

input to cell field math. Mathematical expressions can involve any or all of these input fields. Data input to each of the four ports must be scalar. The output is also a scalar. If a data field contains more than one data component, you may select from any of them.

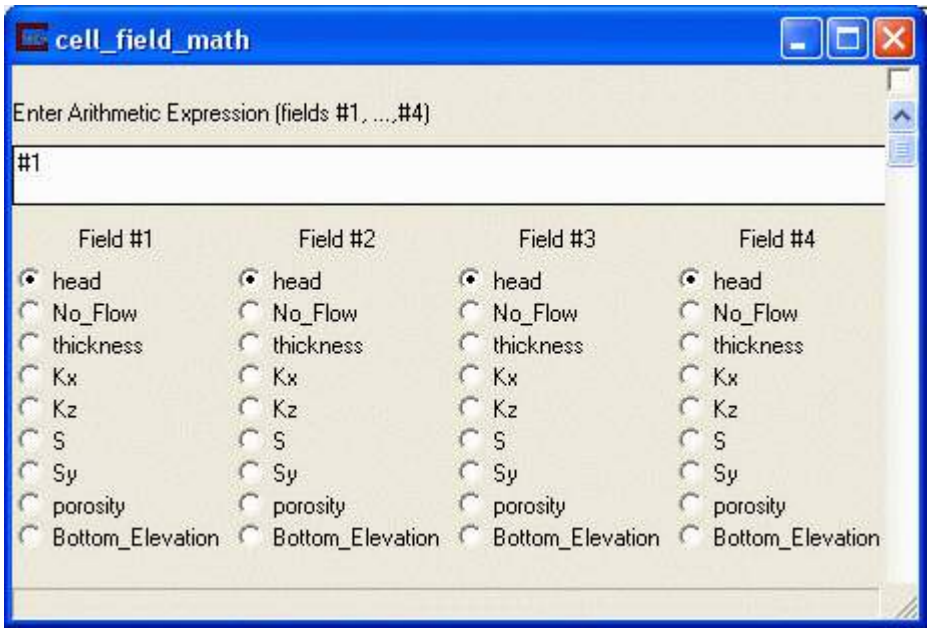
Module Input Ports

Cell_field_math has four identical input ports. Each port can accept any type of mesh with scalar nodal or cell data, but the meshes must have the same cell types and number of nodes (i.e., they must have identical geometries). At least one input port must be used and up to four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right. Nodal data passed to ports one, two, three and four are referred to as #1, #2, #3 and #4 in the mathematical expression. Again, the nodal data must be scalar.

IMPORTANT: The mesh of only the first port is used when passing onto downstream modules. Therefore, only the first input port must contain the mesh data and all input ports must contain data on the mesh of the first input port. Also note that Explode_and_Scale can NOT be used after field_math since it uses a function of the material color number and geo_layer (a data component) to do exploding operations.

Module Output Ports

cell_field_math has two output ports. The first output port (closest to the left) contains a nodal data component which is the result of the mathematical computation. Mesh data is referenced to the mesh in input field one. The second output port is renderable if the input is surfaces.



Module Control Panel

The control panel for cell_field_math is shown in the figure above. The edit field is used for entering the mathematical expression to be computed. Valid mathematical operators and functions are listed below. Refer to input data

components by #[input port number]. For example, to multiply the first field by the third field and then divide the result by the second field, enter:

#1 * #3 / #2

or, this operation can alternatively be completed using C style functions;

divide(multiply(#1,#3),#2).

An example of a less trivial equation to blend the values of (interpolate between) two different UCD files having cell data with a logarithmic distributions is:

log10(pow(10,#1)*0.75+pow(10,#2)*(1.0-0.75))

which performs interpolation of log processed data. To perform interpolation between non-log processed data use:

#1 * %lf + #2 * (1.0-%lf))

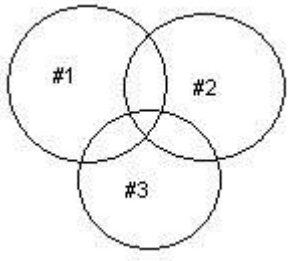
We can define a max function to find a bigger number between two numbers. In the example below, the parenthesis will return 1 if the conditional statement inside is right or return 0 if it is wrong.

(#1 > #2)*#1 + (#1 <= #2)*#2

we can define a min function in the same way.

(#1 < #2)*#1 + (#1 >= #2)*#2

Union of #1 > 5, #2>2.5 and #3 > 100



to define a union above, enter

(#1 > 5) | (#2 > 2.5) | (#3 > 100)

to get the intersection, replace | with &.

(#1 > 5) & (#2 > 2.5) & (#3 > 100)

Mathematical Operators

An expression can include the C-style numeric and logical operators shown in the following table:

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo

&	Logical and
|	Logical or
^	Logical xor
-value	Unary minus

Pop-upAvailable Mathematical Operators hereorJump to Available Mathematical Operators here

Order of precedence

Mathematical expressions follow C's order of precedence. Use parentheses to override the default order.

Functions

The following functions can be used to perform mathematical and logical operations:

Function	Description
abs acos add (+) asin atan cos cosh divide (/) exp log log10 modulo (%) multiply (*) pow sin sinh sqrt sub (-) tan tanh	Perform a mathematical operation. Several of these operations can be also be performed with the numeric operators shown in parentheses.
and (&) or (|) xor (^)	Logical operations can be performed with either functions or the operators shown in parentheses.

Related Modules

->coordinate_math

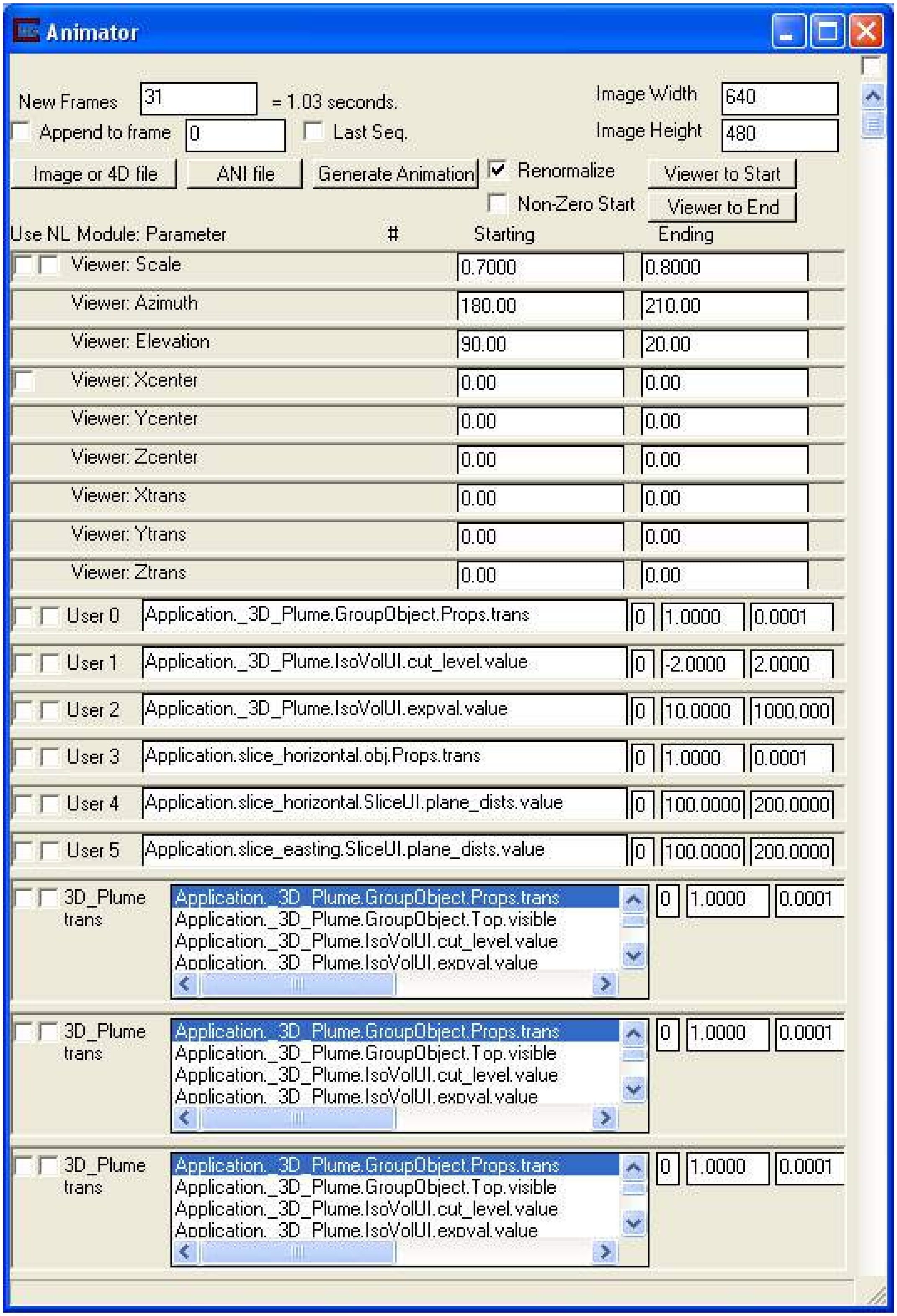
Animator (deprecated Pre-Version 7.0 module)

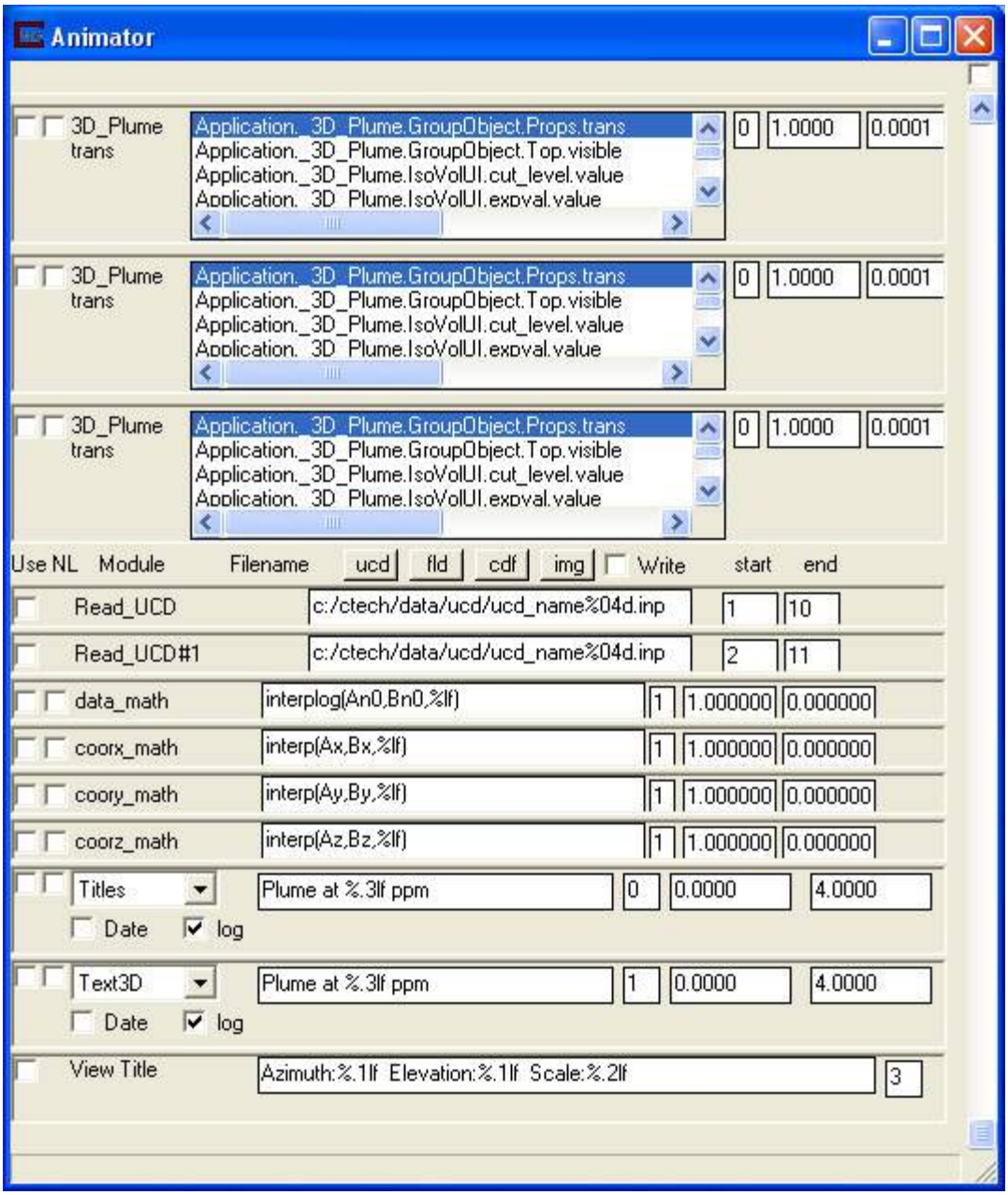
General Module Function

EVS has always had the ability to create sequences of bitmap images to produce animations. The Animator module makes this process very simple and provides powerful new capabilities to simultaneously modify the parameters of many modules in a network for each frame of the animation. The Animator creates a journal (.ani) file which is played back using Journal.Playback. The journal file will modify parameters of modules and optionally create a sequence of .bmp files which can be converted to animations. The user interface for Animator is one of the largest in EVS. However, because of the similarity of many of the options, Animator is very simple to use. Note that creation of avi files is also covered and a special section on making avi files is also included at the bottom of this help section.

Module Control Panel

The control panel for Animator is shown in the figure below.


Animator
New Frames
31
= 1.03 seconds.
Image Width
640
Append to frame
0
Last Seq.
Image Height
480
Image or 4D file
ANI file
Generate Animation
Renormalize
Viewer to Start
Non-Zero Start
Viewer to End
Use NL Module: Parameter
#
Starting
Ending
Viewer: Scale
0.7000
0.8000
Viewer: Azimuth
180.00
210.00
Viewer: Elevation
90.00
20.00
Viewer: Xcenter
0.00
0.00
Viewer: Ycenter
0.00
0.00
Viewer: Zcenter
0.00
0.00
Viewer: Xtrans
0.00
0.00
Viewer: Ytrans
0.00
0.00
Viewer: Ztrans
0.00
0.00
User 0
Application._3D_Plume.GroupObject.Props.trans
0
1.0000
0.0001
User 1
Application._3D_Plume.IsoVolUI.cut_level.value
0
-2.0000
2.0000
User 2
Application._3D_Plume.IsoVolUI.expval.value
0
10.0000
1000.000
User 3
Application.slice_horizontal.obj.Props.trans
0
1.0000
0.0001
User 4
Application.slice_horizontal.SliceUI.plane_dists.value
0
100.0000
200.0000
User 5
Application.slice_easting.SliceUI.plane_dists.value
0
100.0000
200.0000
3D_Plume trans
Application._3D_Plume.GroupObject.Props.trans
Application._3D_Plume.GroupObject.Top.visible
Application._3D_Plume.IsoVolUI.cut_level.value
Application._3D_Plume.IsoVolUI.expval.value
0
1.0000
0.0001
3D_Plume trans
Application._3D_Plume.GroupObject.Props.trans
Application._3D_Plume.GroupObject.Top.visible
Application._3D_Plume.IsoVolUI.cut_level.value
Application._3D_Plume.IsoVolUI.expval.value
0
1.0000
0.0001
3D_Plume trans
Application._3D_Plume.GroupObject.Props.trans
Application._3D_Plume.GroupObject.Top.visible
Application._3D_Plume.IsoVolUI.cut_level.value
Application._3D_Plume.IsoVolUI.expval.value
0
1.0000
0.0001



The user interface is shown below. Each of the user modified parameters is discussed below.

1) ***New Frames***: This type-in at the top left corner specifies the number of new-animation frames to create or append. If played back at 30 frames per second, the total number of seconds for this section is also displayed.

2) ***Image Width & Height***: These type-ins specify the size of the image file to create if an image type (not .4d) file name is specified with the "Image or 4D file" button.

3) ***Append to frame***: This line has two parameters, a toggle to specify that this new_frame section should be appended (added after the end) to a previously generated section. This is valid only if the frame specified here is already greater than 0. This number cannot be modified by the user and is

determined by the New_Frames type-in and the number of sequences appended.

4) ***Last Seq***.: This toggle must be set for the last sequence appended to a file. If only one sequence is being created, it should also be set.

5) ***Image or 4D file***: Push this button and specify an image "base" name if the journal file is to create a sequence of image files or a 4DIM file (.4d) name. Any supported image file type may be specified though we strongly recommend using PNG (.png). The name specified should be of the form base.png (or other suffix such as .bmp, .tif, .tga, etc.) and the files created will be base0001.png through base0101.png (for 101 total frames). Specifying a PNG file is optional, since the Animator can be used to create journal files for real-time playback without image file creation. If an image file name is specified the network to be used with the journal file must contain the Output_Images module.

If a 4DIM file is specified, each frame of the animation will be recorded as a 4DIM frame and the 4DIM file will be saved at the end of the animation. BE SURE TO CHECK "Last Seq." for your last scene of the animation to let the Animator know it is time to write your 4DIM file.

6) ***ANI file***: Push this button and specify an animation file name. The file name should have a suffix of .ani.

7) ***Generate Animation***: This button creates the journal file provided the .ani file is specified.

8) ***Renormalize*** and ***Non-Zero Start***: The renormalize toggle causes an additional Reset, Normalize, and Center to be performed every frame. This slows the creation of animation sequences but ensures that the objects stay centered if they change size (such as explode distance or scale). The Non-Zero Start toggle (if turned on) allows for creation of multiple .ani files, but with each journal using the same root file name and appending the numbering as specified by the user. This approach may be used instead of sequences if certain modules will be turned off midway through an animation.

Note: As of Version 7.0, this deprecated module has Viewer to Start and Viewer to End buttons that replace the Transfer to Start (and End) buttons in the advanced panel of the Az_El. These are provided to only for backwards compatibility. You are strongly encouraged to use the new Animator.

9) ***Use*** and ***NL*** Toggles: Each module listed can be selected for use for every frame. The Viewer's Scale, Azimuth, and Elevation are treated as a single module and only one Viewer can be specified.

Xcenter "Use" toggle NOW ALSO affects Viewer: Xtrans, Ytrans and Ztrans ... only one toggle is used to represent all 6, since you should ALWAYS do both at the same time.

The ***NL*** toggle Non-linear interpolation in the Animator provides "soft" acceleration when moving or rotating objects (or interpolating any other feature). Instead of linear interpolation, a SINE function is used so the slope (rate of change of the parameter) is zero at the beginning and end.

10) "**#**":The number before the start and end fields that was previously the number of decimal places has been changed to be the module number for that module name. For example if you have a plume_volume and plume_volume#1, you can select plume_volume in the selector lists (or in the User type-ins) and if the number (#) is set to 0 you get plume_volume, if it is 1 you get plume_volume#1. This works for all 6 User, the 6 selector blocks, and both Titles.

11) ***Starting:*** The Starting type-in is the initial value for the sequence.

12) ***Ending:*** The Ending type-in is the final value for the sequence.

13) ***User 0*** - ***User 5*** require you to input the actual syntax for changing the parameter of interest. An example would be: Application._plume_shell.GroupObject.Props.trans which controls the transparency of the group object of plume_shell. For example if you had two plume_shells, the second one's syntax would be:

Application._plume_shell#1.GroupObject.Props.trans

since the word between the "." dots following "Application" is the modulename. Now, you put the correct syntax directly into a type-in and it is used by the animator directly. This is also useful if you have renamed any modules but still want to animate their parameters. There are 6 of these "user" control items.

14) This selector block control item has a scrollable list of "EVERYTHING WE COULD THINK OF ANIMATING". In the scrollable list, you will have control over visibility and transparency for EVERY module. Additionally, every parameter that "makes sense" to animate in other modules is also included. Some interesting new examples are the x,y position of Titles (so you can animate scrolling or moving titles!),z-exaggeration for many modules (Map_Spheres, Explode_and_Scale), Geologic_Surface surface number, advector time, and much more. There are ~300 items in the scrollable list. We've tested many but not all. (However, remember that you can always use the user controls for anything not supported in the scrollable controls)

15) Modules ***Read_UCD and Read_UCD#1*** are treated in a special manner. the Read_UCD control items have buttons which switch between UCD, Fields, and NetCDF. All three types are fully supported.

The filename created will be determined by the base name and standard "C" convention for formatting. Therefore, pathnames must have forward slashes (Unix convention: /), or alternatively double back slashes (\\), instead of the commonly used DOS single backslash. Note that the path is not required if the files are placed in the \evs3\data\ucd\ directory. The default value of base%04d.inp will expect file names of base0001.inp through base0101.inp if there were 101 frames. If only one of these modules is used (Read_UCD) the file name is incremented from start to end in a simple fashion. If both modules are used with field_math, interpolation between files is activated. If both Read_UCD modules and field_math are used, the default values of Animator would use base0001.inp through base0011.inp in 31 steps such that each step would increment by 1/3 (e.g. the first frame would be all base0001.inp, the second would be 2/3 of base0001.inp and 1/3 of

base0002.inp. The 30th frame would be 1/3 of base0010.inp and 2/3 of base0011.inp).

16) ***data_math:*** data_math expressions can be created with %lf as the double precision variable name. The Vars field specifies how many %lf variables are used in the expression.

In the Animator module the default equation for data_math is:

interplog(An0,Bn0,%lf)

which performs interpolation of log processed data.

This expression exponentiates log concentrations, then adds the values and takes the log (base 10) of the result. Merely adding log concentrations would be equivalent to multiplying the actual concentrations. Note that the default expression has two %lf entries (variables) so the default Vars field is 2.

If your data component is not log processed use the following expression:

interp(An0,Bn0,%lf)

17) ***coordinate_math:*** field_math expressions can be created with %lf as the double precision variable name. The Vars field specifies how many %lf variables are used in the expression.

In the Animator module the default equations for coordinate_math (x, y, and z) are:

interp(Ax,Bx,%lf)

interp(Ay,By,%lf)

interp(Az,Bz,%lf)

which performs interpolation of coordinate data.

18) ***Titles:*** The Titles blocks support Text3D and "string". Those who have taken a recent training class will appreciate the value of controlling "string".

The titles field includes a log toggle to exponentiate the starting and ending values. The Title should include standard "C" syntax %lf syntax for including numbers. As with each animator function, be sure to instance and connect a titles module, then position the sample text in the viewer before creating an animation script. The values for the start and end boxes should represent the attribute range of interest. For example, if plume_shell is used and the range is -2.0 to 1.0 (log values) then the same values should be used in the titles start and end boxes. Also, we've added a second Titles and controls for coordinate_math (both x, y, and z coordinates).

The Title module option has the ability to output highly configurable date and time strings. This is accomplished using formatting codes.

Pop-up availableDateFormatshere or **Jump to available date formats here**.

The starting and ending dates (values) can be input in very standard (Excel like) windows format. Examples would be:

1. 03/14/89 (for March 14, 1989)
2. 1/2/75 (for January 2, 1975)

3. 01/23/03 (for January 23, 2003)

Note: If you enter an invalid date, as soon as you hit Enter or click-out (loose focus) on the type-in parameter, it will print a warning message to the Status Window telling you that the input is invalid.

19) ***View Title***: The view titles field provides text in the viewer which posts the current azimuth, elevation and scale. *This function works only with the animator module.* The variables box can be changed from the default of 3 (showing Az, El and Scale) to show only Az. and El (variable = 2) or only Azimuth (variable = 1). Note that the viewer toggle must be checked to activate and update the view title text, but the text may remain in the view after viewing animation is stopped (viewer toggle off), for the remainder of an animation. Of course the values of variables in this case will not be changed, but this may be useful for maintaining a posting of the viewer information.

Animation Creation using EVS

Once the sequence of images files is created, there are several options for creating animations. The highest quality *videotape* animation is produced by having:

a) A scan converter (which converts computer video RGB signals to NTSC composite or S-Video or RGB) connected to a single frame video recorder (either an animation video tape recorder or video disc recorder)... or

b) A high quality PC based video playback system capable of playing a sequence of image files (like Targa) stored on your computer's hard disc at 30 frames per second. This equipment usually has a built in scan converter to output a compatible NTSC video signal directly to a VCR.

Animation recorded onto videotape is not the only option. Sequences of image files can be converted to one of several standard computer animation file formats. These formats include .AVI and MPEG. Our experience is that MPEG usually plays back slower than AVI without special hardware. With special hardware, MPEG files can be played back at high frame rates like the systems described in b) above. Without special hardware, high quality animation sequences can be played back on your Windows computer system using the included Media Player by converting the image files to .AVI format.

Animate_4DIM

This is a deprecated module. This modules functionality has been surpassed by the create_4DV module and the bundle_4DV.

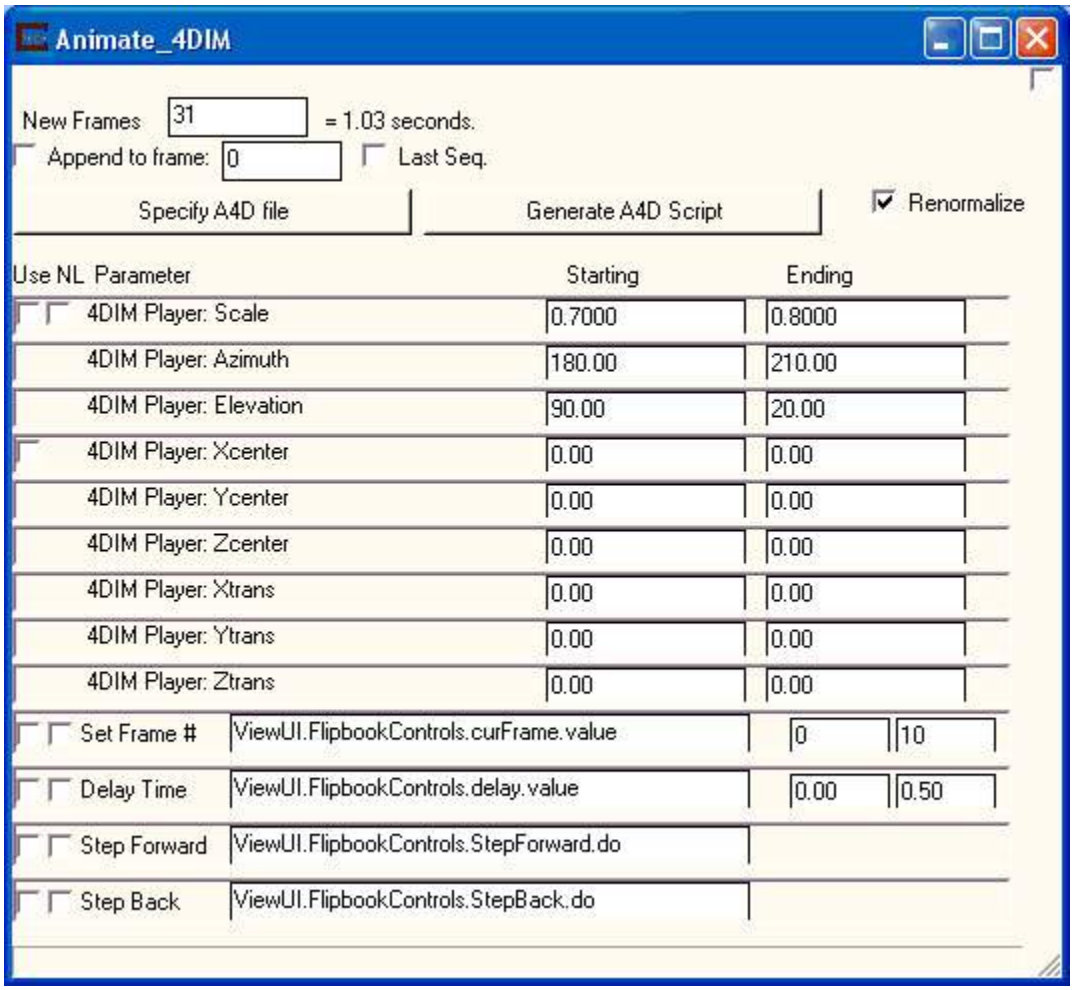
General Module Function

EVS-PRO and MVS have the ability to produce vector based 4DIM animations. The Animate_4DIM module makes it possible to script the behavior of 4DIM animations.

The Animate_4DIM creates a journal (.a4d) file which is played back from the 4DIM player.

Module Control Panel

The control panel for Animate_4DIM is shown in the figure below.



The user interface is shown below. Each of the user modified parameters is discussed below.

1) New Frames: This type-in at the top left corner specifies the number of new-animation frames to create or append. If played back at 30 frames per second, the total number of seconds for this section is also displayed.

2) Image Width & Height: These type-ins specify the size of the image file to create if an image type (not .4d) file name is specified with the "Image or 4D file" button.

3) Append to frame: This line has two parameters, a toggle to specify that this new_frame section should be appended (added after the end) to a previously generated section. This is valid only if the frame specified here is already greater than 0. This number cannot be modified by the user and is determined by the New_Frames type-in and the number of sequences appended.

4) Last Seq.: This toggle must be set for the last sequence appended to a file. If only one sequence is being created, it should also be set.

5) Specify A4D file: Push this button and specify an animation file name. The file name should have a suffix of .ani.

6) Generate A4D Script: This button creates the A4D file provided the . A4D file is specified.

7) Renormalize: The renormalize toggle causes an additional Reset, Normalize, and Center to be performed every frame. This slows the creation of animation sequences but ensures that the objects stay centered if they change size (such as explode distance or scale).

8) Use and NL Toggles: Each module listed can be selected for use for every frame. The Viewer's Scale, Azimuth, and Elevation are treated as a single module and only one Viewer can be specified.

Xcenter "Use" toggle NOW ALSO affects Viewer: Xtrans, Ytrans and Ztrans ... only one toggle is used to represent all 6, since you should ALWAYS do both at the same time.

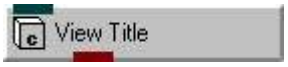
The "NL" toggle Non-linear interpolation in the Animate_4DIM provides "soft" acceleration when moving or rotating objects (or interpolating any other feature). Instead of linear interpolation, a SINE function is used so the slope (rate of change of the parameter) is zero at the beginning and end.

9) Starting: The Starting type-in is the initial value for the sequence.

10) Ending: The Ending type-in is the final value for the sequence.

The remaining commands are very similar to those in the Animator module which is discussed in Workbook 7.

View_Title



This is a deprecated module. The ability to output a text string from the Az_El panel which has all view information combined with the Titles modulesupersedesthis module's functionality with the old Animator.

General Module Function

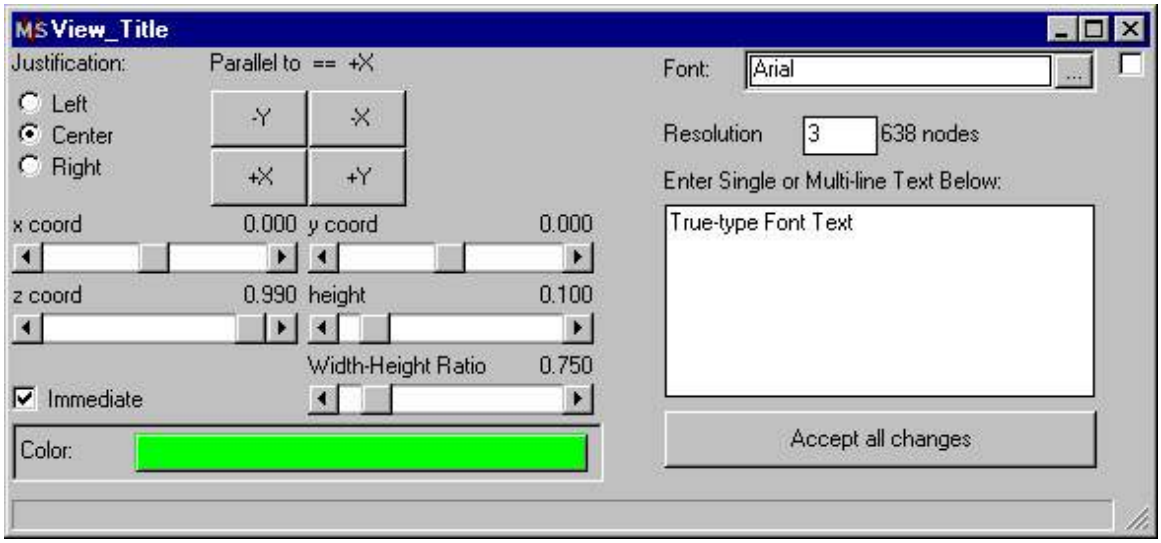
The View_Title module may only be activated through use of the animator module. Posting a title in the viewer when NOT using animator is accomplished with the Titles module. The View_Title module connects to the blue/red port on the Viewer (as does Titles and Color_Legend). The toggle for View_Title must be checked in the animator to activate this module. By using the blue/red port, the text is not moved or transformed by Viewer transformations and is positioned using sliders in the View_Title user interface. The main function of this module is to post a continuously updating Azimuth, Elevation and Scale value reflecting viewer transformations during an animation.

Module Input Ports

The string to be rendered.

Module Output Ports

Titles has only one output port. This port outputs labels to the Viewer.



Module Control Panel

The control panel for View_Title is identical to the Titles module interface. However, this module functions only as a result of animator activation. The positioning of the text is controlled by the control panel, but the text content is controlled by the type-in fields for the animator view titles box. See the animator help section for details of this function.

Control_Fly_Through

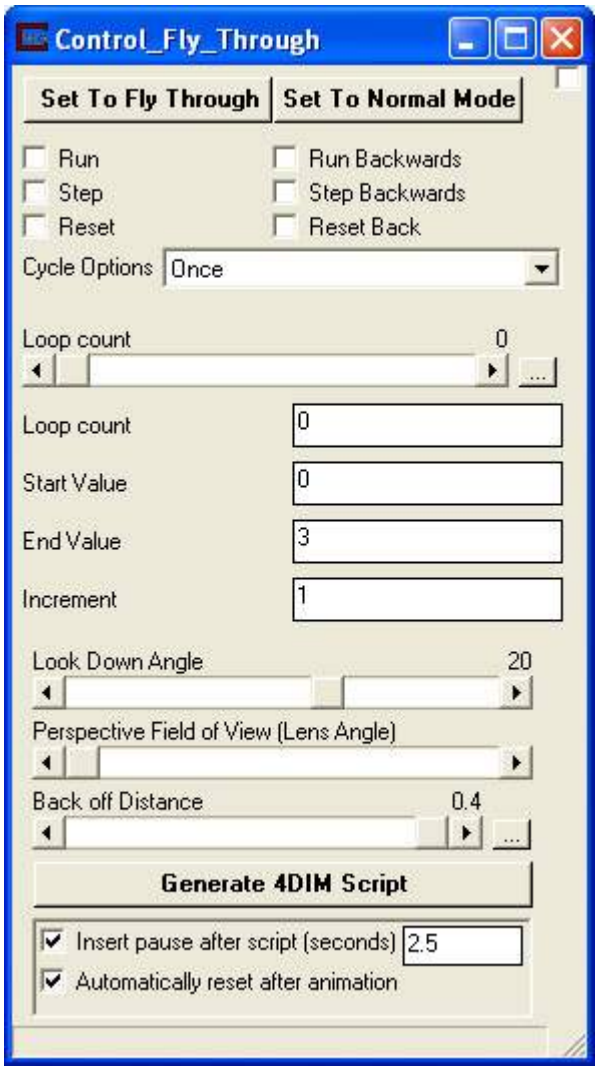
This module has been deprecated and supercededbyfly_through

General Module Function

Control_Fly_Through provides a means to export an ASCII file containing the coordinates (and optionally the data) of any object in EVS/MVS. The output contains a header line and one row for each node in the input field. Each row contains the x, y, & z coordinates and optionally node number and nodal data.

Module Input Ports

Control_Fly_Through has only one input port. This port accepts the output from any module with a blue/black (field) output port.



Module Control Panel

The control panel for Control_Fly_Through is shown in the figure above.

The ***Set To Fly Through*** button resets the viewer orientation to begin the fly through. This should be pressed after valid flying path input is supplied to the module.

The ***Set To Normal Mode*** button resets the viewer mode to the C Tech default paradigm. This is useful if you wish to use the Az-El panel or rotate the view with your mouse.

The ***Run*** toggle when set will kick off an iteration of the loop. It is reset to off when it is finished.

The ***Step*** toggle will make one increment of the loop as specified int the Increment type-in.

Reset, when toggled, resets the count subobject to the value of the start_val subobject. This value is reset to off right after it is changed.

The ***Run Backwards***, ***Step Backwards*** and ***Reset Back*** are self explanatory.

The ***Cycle Options*** allow for Once (one iteration), Cycle (keep running from beginning), and Bounce (run to end, then backwards, then forwards, etc.)

The **Loop count** is updated during running of the module, but the user may jump to any part of the specified iteration by typing in a valid value. It is available as a slider and a type-in. The slider is particularly useful as you can control the output of loop by moving it in real-time.

Start Value indicates the starting value of the iteration and refers to the slider value of the module to which loop is outputting.

End Value indicates the ending value of the iteration and refers to the slider value of the module to which loop is outputting.

Increment indicates the interval at which to count from the starting value to the ending value. For example, to move a slice plane in 10 steps from -50.0 (Start Value) to 40.0 (End Value) the user would choose an increment of 10.00. These increments may have non-integer values.

Look Down Angle is a slider that determines the angle (in degrees) of elevation of the viewer.

Perspective Field of View (Lens Angle) is a slider that sets the total included angle (in degrees) of the camera lens used to see the scene. Fly-Throughs always use perspective and perspective is based on the concept of a lens angle. Larger angles have a wider field of view.

Back off Distance is a slider that determines the distance between the surface and the camera. The units are the same as the units of your surface (feet or meters typically).

The ***Generate 4DIM Script*** button exports the fly-through path as an A4D script that can be used by the 4DIM player. If you create a 4DIM file of your model (in normal Viewer Mode), this script will execute the fly-through maneuvers inside the 4DIM Player!

The ***Insert pause after script*** toggle allows the user to insert a pause at the end of the 4DIM script.

The ***Automatically reset after animation*** toggle will reset the fly through at the end of the animation.

Sample Applications

A sample network that demonstrates the use of Control_Fly_Through is fly_over_geologic_surface.v in the Pro folder. The applications is shown below:

{bmc Control_Fly_Through_app.bmp)

In the application above, each module has an important function. In order to help users understand the basics of fly through animation, each modules function is summarized below.

Spline_Geology together with *Geologic_Surface* creates the surface for this flythrough.

Geologic_Surfaces_Flat (with input from *Spline_Geology*) creates a two-dimensional surface upon which we draw in the *TopView* window using *draw_2d_lines*.

Titles provides annotation for the *TopView* window.

Draw_2d_lines passes its two-dimensional polyline output to *polyline_spline* which segments and (optionally) smoothes the path drawn by hand with *ContinuousSketch*.

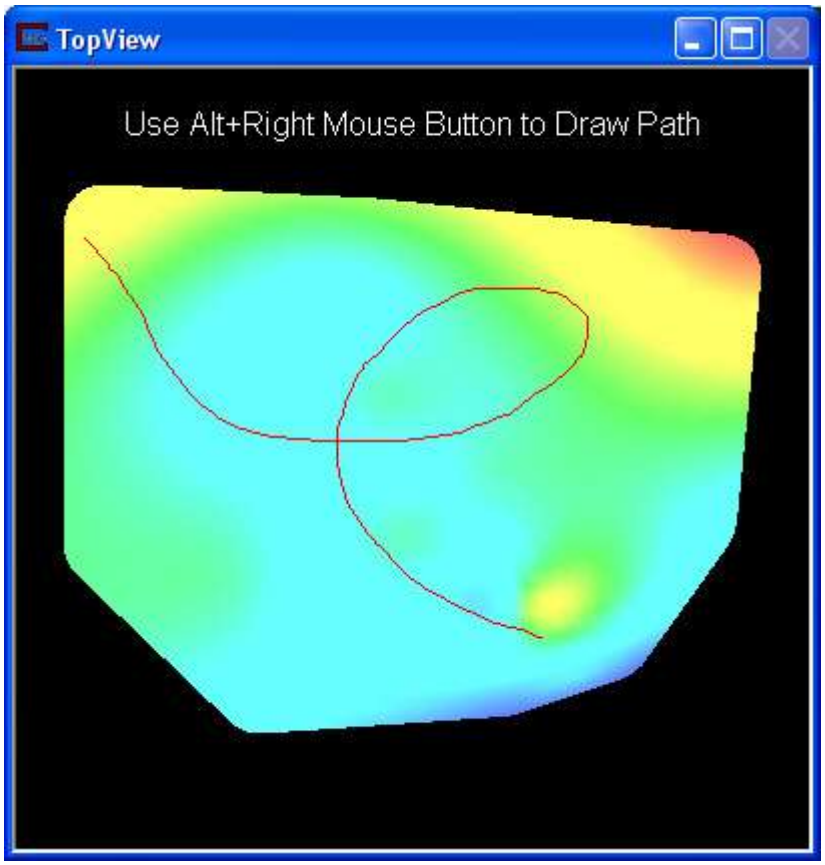
polyline_spline passes its two-dimensional polyline output to *interp_data* which also receives a two-dimensional surface with elevation data from *Geologic_Surface_Flat*. The key point here is that both the surface and polyline are flat (at Z=0.0). Therefore the line falls **inside** the surface. Because it is inside, interp_data can map the elevation data of the surface onto the line.

The output of *interp_data* is the two-dimensional polyline with data that corresponds to elevations on the surface we intend to fly through.

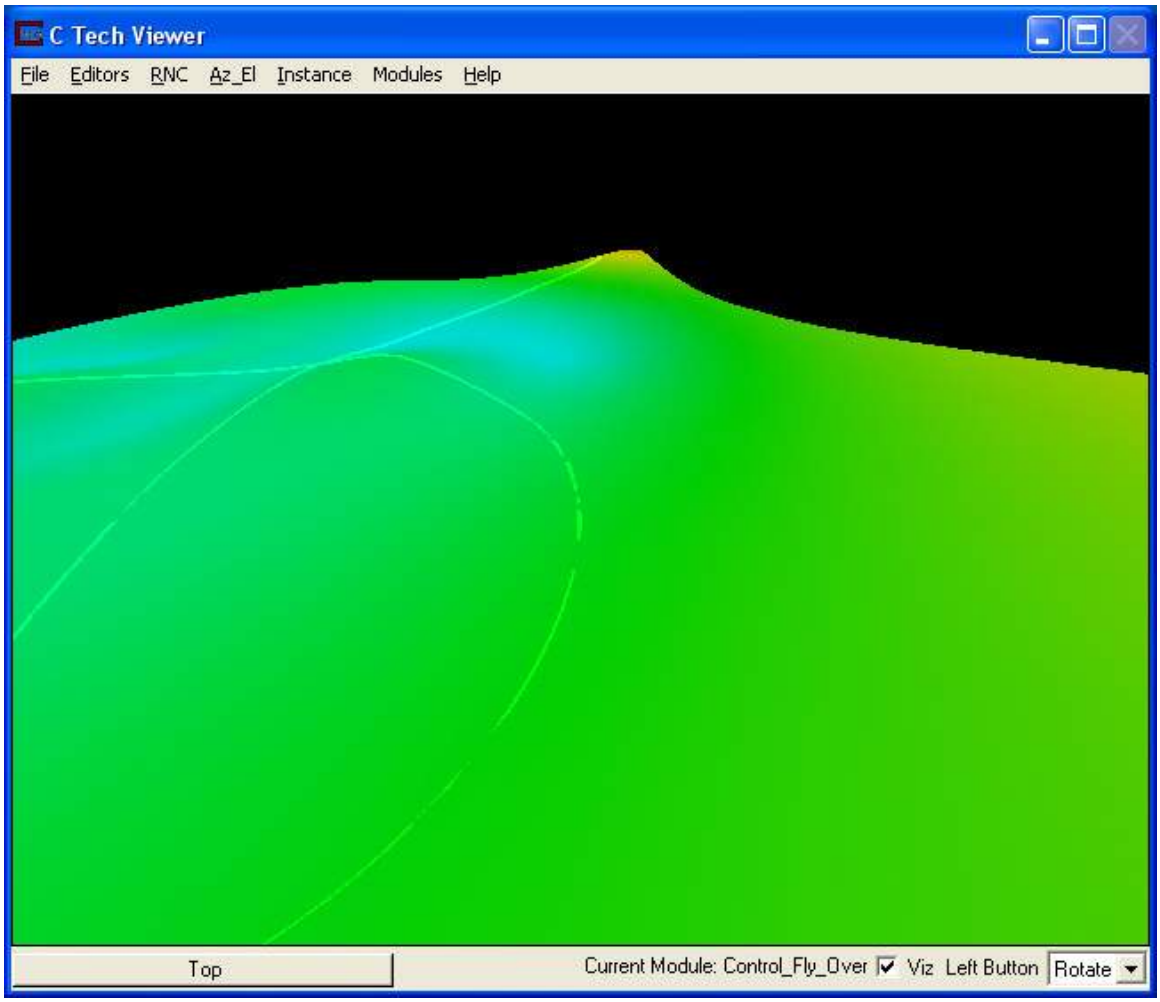
surf_plot takes the output from *interp_data* and maps it to the three-dimensional path which now lies on the three-dimensional output from *Geologic_Surface.* This is allowing us to create a 3D path having only drawn a 2D polyline.

Control_Fly_Through takes the output from *surf_plot* and controls *Viewer's* camera position. Please note that controlling the viewer's camera is a different paradigm from the normal Az-El controls which actually rotates the objects with respect to a static viewer camera.

The TopView window with a path drawn is shown below.



The Viewer window is shown below half way through the fly through.



ClickSketch

This is a deprecated module whose function has been superceded by thedraw_2D_linesmodule and thedraw_3D_linesmodule.

General Module Function

ClickSketch enables you to create a 2D drawing with individual clicks of the mouse. The mouse gesture for line creation is: depress the alt key and then click the right mouse button. The first click establishes the beginning point of the line segment and the second click establishes the end point. ClickSketch allows adding of points, erasing of region points, and closing a region. The lines, once drawn, will reside at an elevation of 0.0 with reference to the view plane of the current viewer. Therefore, it is recommended that drawing actions be made in a normalized view (from directly above the object) to ensure the line is drawn at a constant z-value. This module is commonly used to create 2D lines passed to the fence_cut module. Unlike most modules which create mesh data to used by other modules, the ClickSketch module receives input from the viewer, and also passes on field data to be

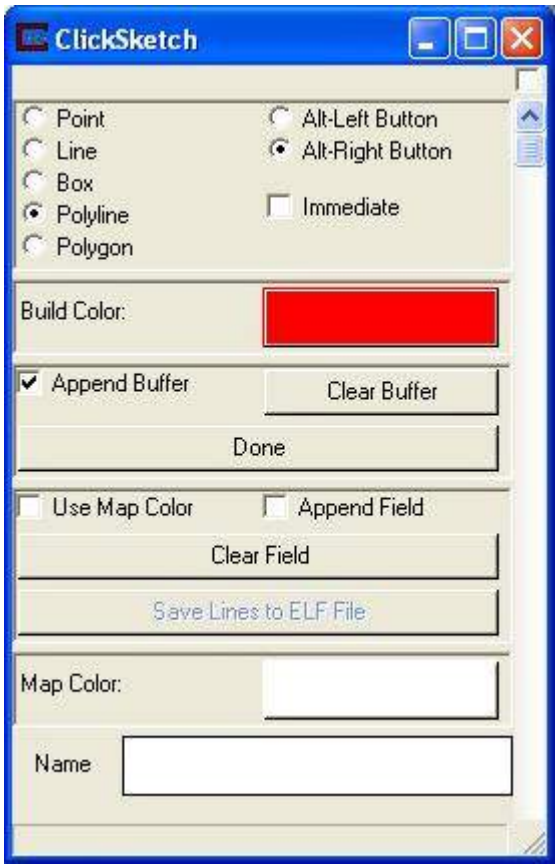
used by other modules. NOTE: Currently in EVS the line will not display in the viewer window unless software renderer is chosen in the viewer instead of OpenGL.

Module Input Ports

ClickSketch has two input ports. This first input port is typically connected to one the viewer's pink output. The second input port is connected to the red output of the object within which the line will be drawn. Without an object, ClickSketch converts the drawing to a field in the camera's world space. With an object, ClickSketch converts the drawing to a field in the object's world space.

Module Output Ports

ClickSketch has two output ports. This first output port sends mesh and nodal data to downstream modules. The second output port sends the line as a renderable object to the viewer to be rendered.



Module Control Panel

The ClickSketch user interface allows interactive creation of points or lines onto any connected object and then displays the points or lines in the viewer. One popular use for this functionality is for creation of line segments to create fence sections using the fence_cut module.

The first set of radio buttons allows picking which type of primitive to draw. Possible primitives are point, line. box, polyline, and polygon. The default is polyline because only polylines are supported with the fence_cut functionality

(box and polygon uses polylines). To the right of those you can select the keyboard and mouse button sequence that is used to draw. For Version 4.5 the new default is Alt+Right Mouse Button.

The immediate toggle controls whether the output buffer is immediately written to (data passed to other modules) upon completion of a line segment. When the toggle is off, you must press the Done button to cause the output buffer to be written.

The build color sliders control the color of the drawn line. Remember the line will not display unless software renderer is selected in the viewer instead of OpenGL.

The append buffer toggle determines whether each time a line segment is completed it overwrites any previous primitive. If append buffer toggle is on, each time a primitive is built, it is appended to a previous line segment(s). By using append mode, you can build multiple primitives before mapping them to a field.

The Clear Buffer button causes the build buffer to be cleared of all points or line segments. The Done button causes all existing line segments to be passed to other modules as a field.

The map color sliders enable passing of color values to be passed as field data to other modules. These color values are only passed if the Use Map Color toggle is on. If the toggle is off, no color information is written to the field. If on, color information is written to the output field as cell data. This gives you the ability to specify color on a per primitive basis.

The Append Field toggle is on by default. By using append mode, you can incrementally build the field (for example fence sections). If the Append Field toggle is off, each time points or lines are mapped, any previous field is overwritten.

The Clear Field button erases any field previously created.

The map name box allows creating a name for the cell sets passed upon pressing the Done button.

The Save Lines to ELF File allows the user to save all of the created lines in the EVS Line File format (*.elf).

ContinuousSketch

This is a deprecated module whose function has been superceded by thedraw_2D_linesmodule and thedraw_3D_linesmodule.

General Module Function

ContinuousSketch enables you to create a 2D drawing with one continuous motion of the mouse. The mouse gesture for line creation is: click and hold down the right mouse button (no keyboard actions). It combines all the capabilities of the ClickSketch module but has the characteristic of automatically creating points along a line as long as the mouse gesture is enacted. The density of points on the line depends on the speed at which the line is drawn (how long the right mouse button is depressed) and the speed

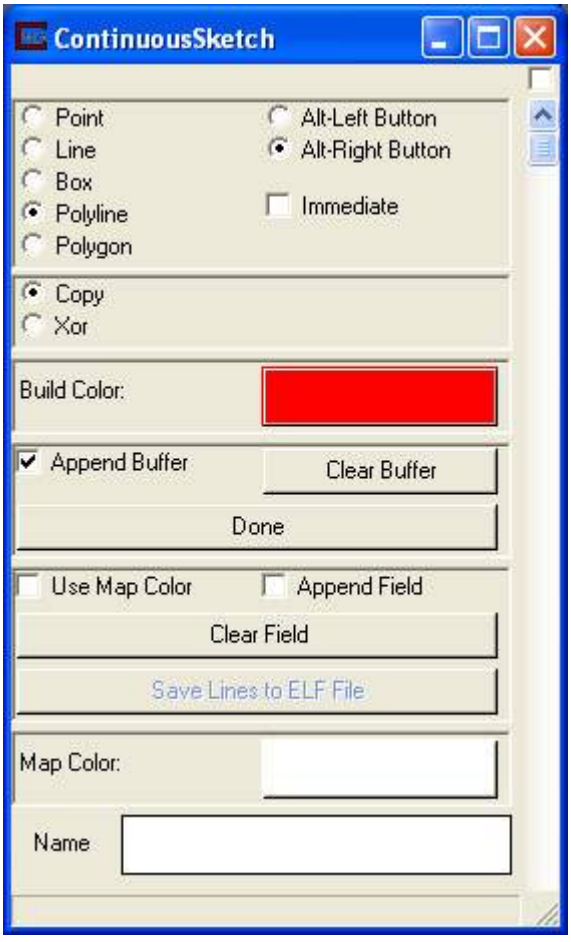
of the machine. The lines when drawn will reside at an elevation of 0.0 with reference to the view plane of the current viewer. Therefore, it is recommended that drawing actions be made in a normalized view (from directly above the object) to ensure the line is drawn at a constant z-value. Note that ContinuousSketch should be used in a single or muti-window application due competing mouse gesture functionality with the default EVS Viewer

Module Input Ports

ContinuousSketch has two input ports. This first input port is typically connected to one the viewer's pink output. The second input port is connected to the red output of the object within which the line will be drawn. Without an object, ClickSketch converts the drawing to a field in the camera's world space. With an object, ClickSketch converts the drawing to a field in the object's world space.

Module Output Ports

ContinuousSketch has two output ports. This first output port sends mesh and nodal data to downstream modules. The second output port sends the line as a renderable object to the viewer to be rendered.



Module Control Panel

The user interface is identical to that of the ClickSketch module with the exception of the Copy/Xor toggle which controls the line drawing color method. Xor simply takes the inverse of the background color while copy uses the referenced build color. Please refer to the control panel explanation in the ClickSketch module help for all other parameter information.

draw_Line

This is a deprecated module whose function has been superceded by thedraw_2D_linesmodule and thedraw_3D_linesmodule.

General Module Function

The draw_line module lets the user interactively draw polylines over an object in a viewer window. The mouse gesture for line creation is: click and hold down the right mouse button (no keyboard actions). Note that draw_line should be used in a single or muti-window application due competing mouse gesture functionality with the default EVS Viewer

Module Input Ports

The draw_line module has one input port. This port is connected to the red output of the object within which the line will be drawn. Without an object, draw_line converts the drawing to a field in the camera's world space. With an object, draw_line converts the drawing to a field in the object's world space.

Module Output Ports

The draw_line module has two output ports. This first output port sends mesh and nodal data to downstream modules. The second output port sends the line as a renderable object to the viewer to be rendered.

Module Control Panel

The Pick Point toggle allows the user to select a series of points on the object. Lines are drawn between each point. The default is on.

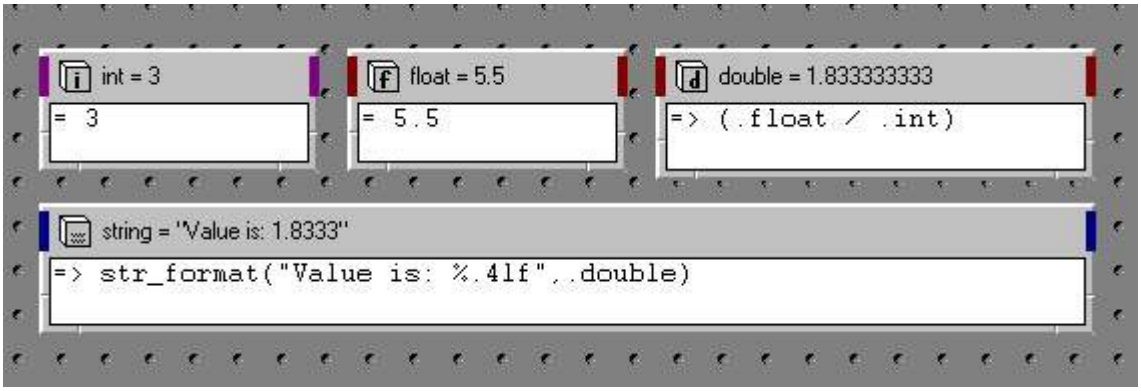
The New Line toggle allows the user to start a new series of lines When on, the user may select a new "first" point and begin drawing a new series of lines. When off, the user is in the middle of drawing a series of lines and any left mouse button click is a line endpoint. The default is off.

The Save Lines to ELF File allows the user to save all of the created lines in the EVS Line File format (*.elf).

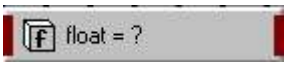
int

This is a deprecated module whose function has been superceded bystring_formatmodule.

int is a module that can contain a simple integer or an expression that represents an integer.

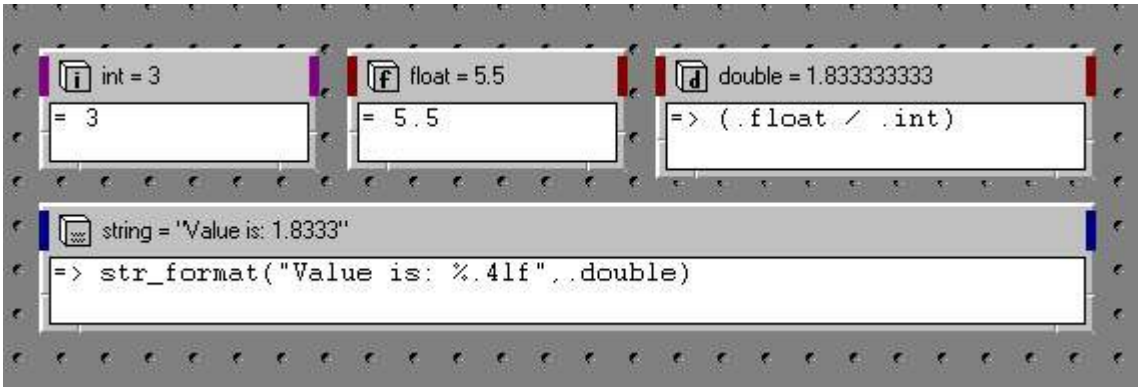


float

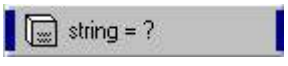


This is a deprecated module whose function has been superceded bystring_formatmodule.

float is a module that can contain a simple float or an expression that represents a float.



string



This is a deprecated module whose function has been superceded bystring_formatmodule.

string is a module that can contain a simple string or an expression that represents an string

The string module and the Animator both use standard "C" language formatting to construct text objects. The figure below shows a simple example. Additional formatting examples will all use the str_format command. This command allows for the construction of very complex strings that reference multiple modules (variables or parameters) in your application.

=> str_format("Unit %d",8) "Unit 8" (note the quotes will not be displayed)

=> str_format("Unit %4d",8) "Unit 8" (note the quotes will not be displayed)

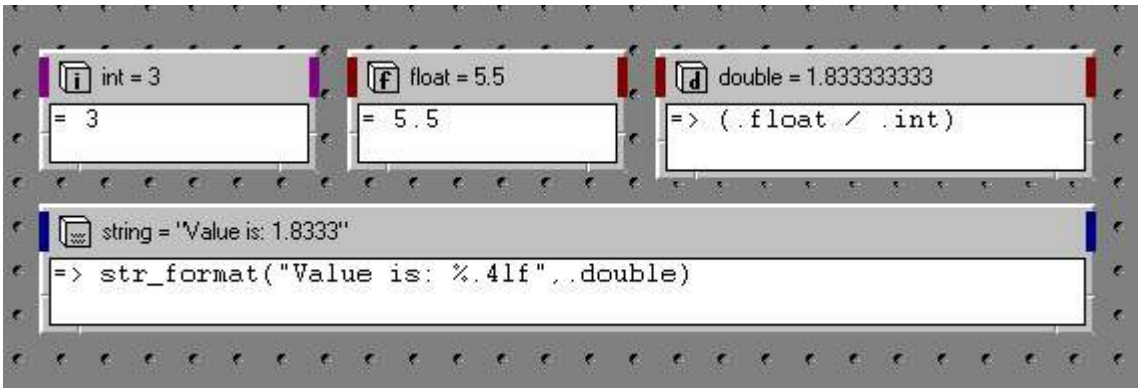
=> str_format("Unit %04d",8) "Unit 0008" (note the quotes will not be displayed)

=> str_format("%f",12567.98799) 12567.98799

=> str_format("%.3f",12567.98799) 12567.988 (note the rounding up of the number above)

=> str_format("%.8g",12567.98799) 12567.988

%lf safe double equivalent to %f



Volume_and_Mass

This module has been deprecated. Use volumetrics instead.

General Module Function

The Volume and Mass module is used to calculate the volumes and masses of soil, and chemicals in soils and ground water, within a user specified constant_shell(surface of constant concentration), and set of geologic layers. The user inputs the units for the nodal properties, model coordinates, and the type of processing that has been applied to the nodal data values, specifies the subsetting level and soil and chemical properties to be used in the calculation, and the module performs an integration of both the soil volumes and chemical masses that are within the specified isosurface. The

results of the integration are displayed in the EVS Status Window window, and in a popup window.

In general Volume_and_Mass will give more accurate results and should be used instead of the simpler volume_integrate module. The exception is when multiple serial subsetting operations are to be performed.

NOTE: The latest version of EVS/MVS can now accept output from any filter or mapper (containing non-hexahedral cell-types. Before this, volume_and_mass could only handle hexahedral cells, but that limitation no longer exists.

Module Input Ports

The Volume and Mass module has only five input ports

The first input port (the leftmost one) accepts a 3D data field.

The second input port is the z exaggeration port.

The third port accepts a string as input; this string is then written to the output file if the Output Results File toggle has been selected.

The fourth input port takes a float value representing the subsetting level.

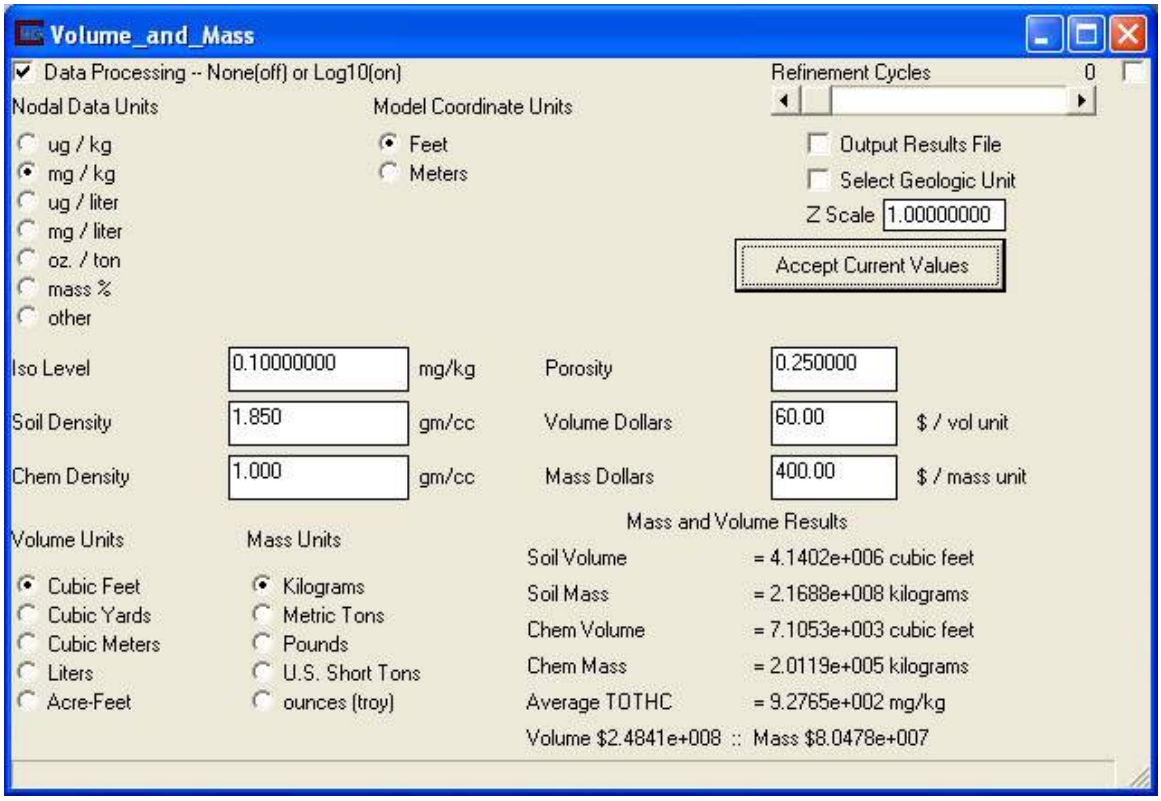
The fifth port takes an int as its input, this int causes the module to run when changed.

Module Output Ports

Volume and Mass has two output ports.

The first port(the leftmost) outputs a float representing the subsetting level.

The second port outputs the volume of the 3D field as a string.



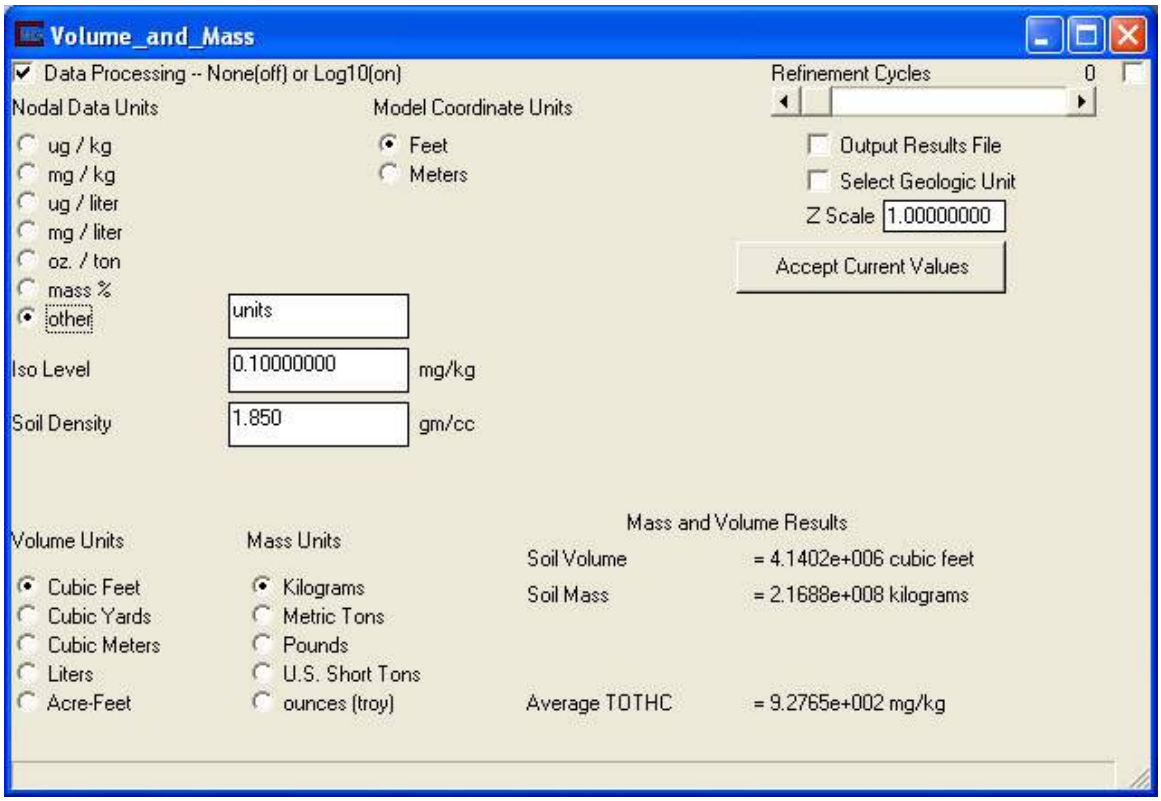
Module Control Panel

The control panel for Volume and Mass is shown above.

The Data Processing Check Box is used to specify whether the input data field has been processed to compute the Log 10 of the nodal values by Krig_3D or the module that is supplying the data to Volume and Mass. The default value is on, which means that Log 10 processing has been completed. Note that most analyte (e.g. chemistry) related data are log processed, while other types of data are not. If the user is reading a previously written UCD file into volume and mass, and is not sure whether processing has been completed, the Statistics module can be used to examine whether negative values that resemble log 10 data are present in the data set. When Log 10 processed data is passed to Volume and Mass, it exponentiates the nodal values in the data file, and completes the integration within the isosurface assuming that the data are exponentially distributed. This algorithm essentially returns the data to a normal distribution to provide accurate volume and mass estimates.

The "Output Results File" toggle causes Volume_and_Mass to write a file to the ctech folder (volume_and_mass_results.txt) that contains all Volume_and_Mass information in a format suitable for input to programs like Excel (tab delimited .txt file). This file is written to in an append mode. It never overwrites old data and it is not deleted (by us). It will grow in size as you use Volume_and_Mass. You should delete or move the file when you're done with it.

The Refinement Cycles slider specifies how finely each element will be subdivided to perform the integration. The default value is 0, which provides the fastest computation time, and resulting volume and mass estimates that are a good first approximation. Generally, the number of refinement cycles should be set to a value between 1 and 4, and the user should experiment with increasing the refinement cycles until subsequent calculations show a change that is insignificant to the problem at hand.



The Nodal Data Units radio buttons are used to specify the units of the input data field. Note that this data can be in mass/mass or mass/volume units to allow calculations of chemical mass in soil or groundwater to be completed. There is also an "other" selection (shown above) that provides a units type-in. When this option is selected, Soil Volume and Soil Mass can be computed, but chemical (analyte) masses cannot. The reason is that this option would require many other conversion factors to perform these calculations. The "other" option is most appropriate for units such as porosity, head, electrical conductivity or resistivity, etc. For these parameters the concept of mass is not appropriate.

The Model Coordinate Units radio button are used to select the unit of measurement for the X, Y, and Z coordinates of the model data. The user should be certain of the nodal and coordinate units selected, as obviously, the output of the calculations are significantly affected by the units of the data.

The Select Geologic Unit check box is used to bring up a dialog box that allows the user to select which of the geologic units the volumes and masses will be calculated in. The default value is All layers, which will provide volume and mass estimates for all of the geologic layers in the model. The user can select which layer they desire volume and mass estimates for by selecting that layer and clicking on the Apply button. The layer number for which the estimates are made is printed in the EVS Console Window, along without the results. If no geologic layer input is being provided or a calculation of the entire domain is required, the user should leave this value set to All.

The Z Scale input field specifies the value of z exaggeration and is used to unscale the model for the purpose of calculations.

The subsetting level input field specifies the value in user units of the isosurface within which the volume and mass estimates will be completed. Note that the units text following the subsetting level input field will change to be consistent with the Nodal Data Units radio button selected, to remind the user what data units are being used.

The Chem Density input field allows the user to input the density of the chemical constituent for which mass estimates are being completed. Note that this value is used to calculate the volume of chemical in the specified isosurface, as the mass units are calculated directly from the nodal data. If Logarithmic input is used, the units input should be actual units (e.g. .01 mg/kg = 0.01).

The Soil Density and Soil Porosity input fields allow the user to input the properties of the soil matrix in which the chemicals reside. Note that if the mass of chemicals in a combined soil and ground water plume are to be estimated, one of the geologic layers should be set up to have a boundary within it that corresponds to the water table position. In essence, this will create two layers out of one geologic unit that can be used to separate the soil domain from the ground water domain. The user can then choose the appropriate Nodal Data Units for each layer in the two domains, and obtain volume and mass estimates by summing the results in individual layers. There are several other alternative methods for completed volume and mass estimates in continuous soil and ground water plumes, which involve either

setting up separate soil and ground water models, or using the Field Math module to remove and include specified areas of the domains.

The Volume Units and Mass Units radio buttons allow the user to set the units in which the volume and mass calculations will be displayed. Note that conversions are made between the Nodal Data Units and the specified output units if the two are not consistent.

The Accept All Current Values button is used to execute the Volume and Mass module once the desired parameters have been input to the module. Until it is pushed, the results will all by "Not Yet Calculated". The user can quickly obtain volume and mass estimates for different combinations of geologic layers, soil and chemical properties, and output units by changing values and then pressing the Accept All Current Values button.

volume_integrate

(This module has been deprecated. Its functionality has been surpassed by the volumetrics module.)

General Module Function

The volume_integrate module is used to calculate the volumes of the entire field input. The input data to volume_integrate must be a three dimensional data field output from Krig 3-D (with or without geology) or any subsetting module which outputs three-dimensional data (such as plume_volume, threshold, cut, & cell_crop). The results of the integration are updated each time the input changes.

To calculate a plume volume: connect Krig_3D (or Read_UCD, etc.) to plume_volume. Set the plume level in plume_volume and connect plume_volume to volume_integrate. Without using plume_volume volume_integrate will give the volume of the entire field's domain. In general volumetrics will give more accurate results and should be used instead. The exception is when multiple serial subsetting operations are to be performed.

Module Input Ports

The volume_integrate module has only one input port, which accepts a 3D data field from Krig 3D, Read UCD, or any of the modules that output 3D fields. The user must remember that if the data have been scaled by any modules upstream of volume_integrate (i.e., in Explode and Scale), that the resulting calculations will reflect that scaling. One or more plume_volume module(s) (in series) can be used, for example, to subset the mesh based on kriging confidence levels (to provide mass estimates only for those regions in which nodal values have been modeled with a given confidence level) or any other parameter(s).

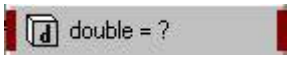
Module Output Ports

volume_integrate has one output port that exports the volume in your user units cubed.

Module Control Panel

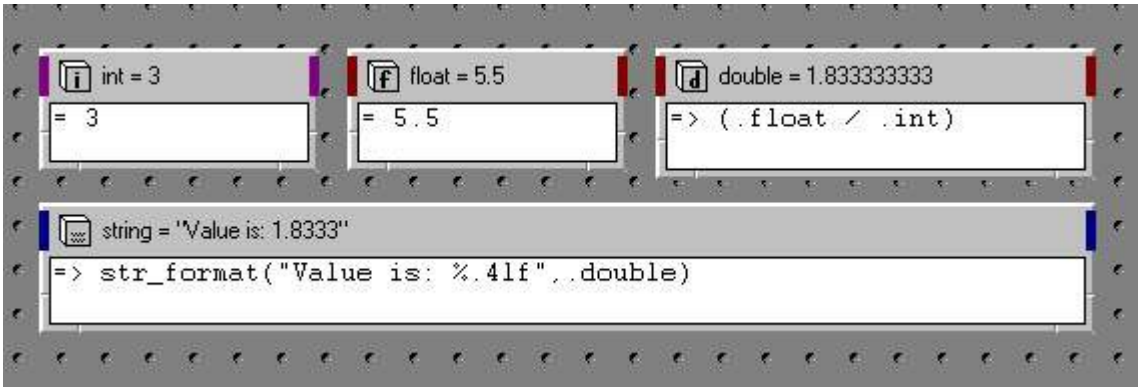
The control panel for volume_integrate is shown above. Note that if your input field has specified units (like feet) the results will show that.

double



This is a deprecated module whose function has been superceded bystring_formatmodule.

double is a module that can contain a simple double or an expression that represents a double



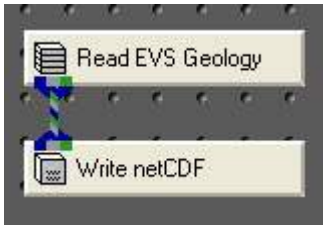
Read_EVS_Geology

This is a deprecated module that has been supplanted byLoad_EVS_Field

General Module Function

Read_EVS_Geology is a deprecated module provided only to allow customers to read archived EVS Geology Files (.egf files) which were saved prior to Version 6.0 as a means to save the results of the kriging by Krig_3D_Geology. There were 2 files that contained all required information.

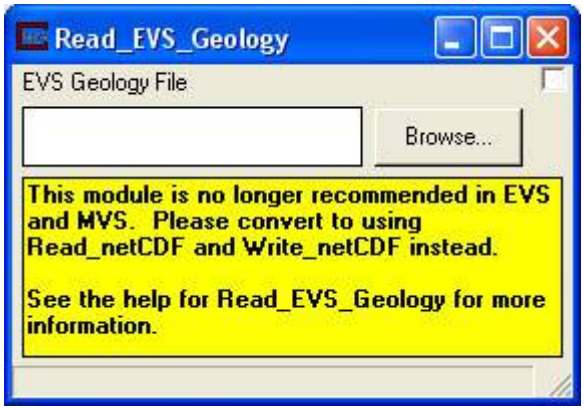
We strongly recommend that you convert older .egf files to the newer format using the following simple process:



As of Version 6.0 the output of Krig_3D_Geology and Spline_Geology can (and should be) saved as a netCDF file using Write_netCDF.

Module Output Ports

Read_EVS_Geology has only one output port. This port provides output that is equivalent to the Krig_3D_Geology module, but should be converted to a netCDF file for future compatibility.



Module Control Panel

The control panel for Read_EVS_Geology is shown in the figure above.

Related Modules

Related Modules

Read_netCDF

Write_netCDF

Color_Legend

This is a deprecated module whose function has been superceded by the Legend module.

General Module Function

The color legend module is used to place a color scale bar in the viewer window. The color legend will, by default, set the minimum value to blue and the maximum value to red. The minimum and maximum values are taken from the data input as is the color datamap. Labels can be placed at user defined intervals along the color scale bar. Labels can consist of numerical values and text or numerical values only. Color legend has an output port which sends data to a special input port on the viewer module. The color range displayed in the color legend scale bar can be changed using the datamap module decribed later in this section

Module Input Ports

Color legend has two input ports.

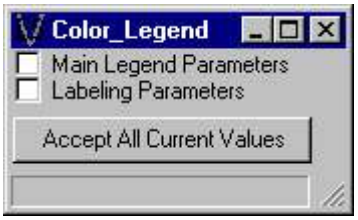
The first port (blue-red-blue-beige) is found on only a few modules in EVS/MVS. The modules with this port are Solid_3D_set, solid_contour_set, isolines, Datamap_Editor, and Color_Legend.

These modules pass data corresponding to one or more subsetting levels (or solid contour levels). When this data is passed to Color_Legend, it is used to set the precise break points in the legend where labeling will occur.

The second (red) port receives scaling data and a datamap from any single module. The module connected to this port should contain the data that the color legend references. Many modules contain output ports compatible with this input port.

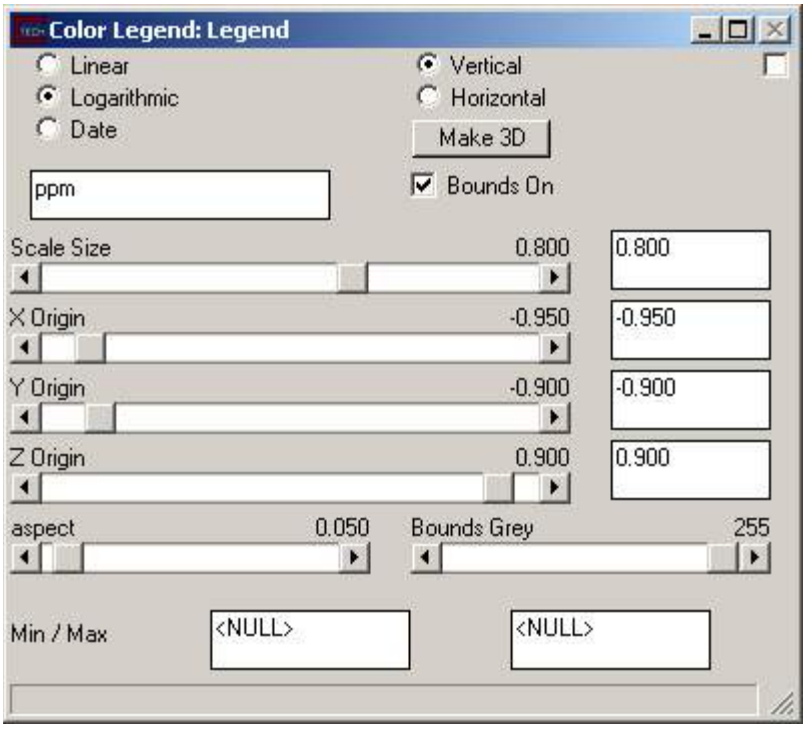
Module Output Ports

Color legend contains one special output port. This port sends color scale bar information directly to the viewer. The color scale bar is a special type of viewer object which is not affected by viewer settings. Also, the color scale bar in the viewer is not transformable with mouse actions.



Module Control Panel

The control panel for color legend is shown in the figure above. This panel is used to spawn two control subpanels. The Accept All Current Values push button is used to apply all the currently selected legend and label settings to the color bar image in the viewer. Affects of some of the selections made in the control subpanels appear immediately in the viewer, while other will not appear until the Accept All Current values push button is selected.

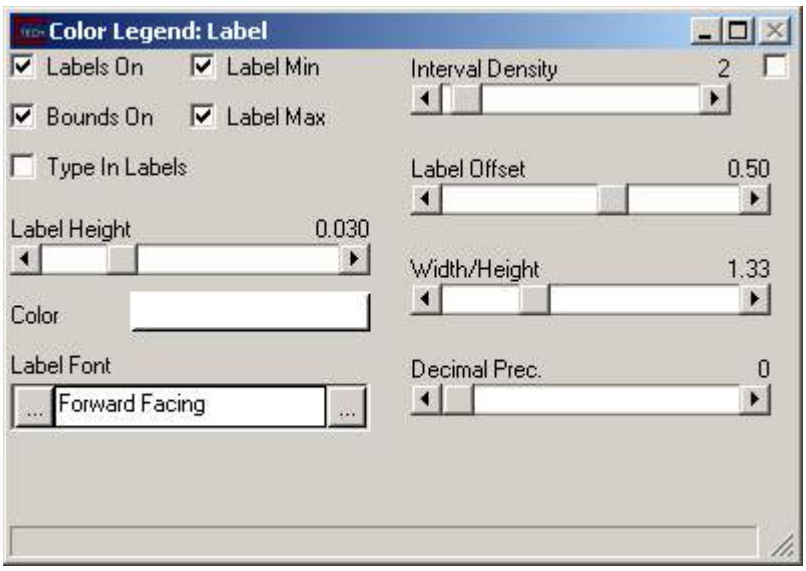


The Main Legend Parameters control subpanel is shown above. The radio buttons at the top are used to specify if the data is linear or logarithmically transformed. Logarithmic is the default. The other two radio buttons are used to tell the viewer that the legend is to be drawn vertically or horizontally. Vectical is the default. The edit field below the Logarithmic radio button is used to enter the label text to be displayed. This is generally the units (Feet, ppm, PSI, etc) that the color scale is representing. The text will appear after the label values on the scale bar.

The scale size slider sets the overall size of the scale bar in the viewer window. The range is from 0 to 10.0 and the default is 3.0. This scale size is in reference to the viewer window. The aspect ratio slider determines how wide the scale bar is with respect to its height. The default is 0.05 and the range is from .01 to 1.00. The X origin, Y origin, and Z origin determine where the lower left corner of the scale bar will be located in the viewer window. Setting the X origin and Y origin to 0 places lower left corner of the color legend bar in the center of the viewer. The Z origin allows the user to control wether the legend ia always in front (default) or behind 3-D objects in the viewer. The user should experiment with these values to get the optimum placement of the color legend.

The bounds grey slider is used to specify the greyscale setting of the boundary outlining the color scale bar. A value of 255, the default, sets the greyscale color to white. A value of 0 sets the boundary color to black. Any number between these two extremes will create a shade of grey. The shade of grey is lighter with larger numbers.

The Min / Max edit boxes are used to override the default range in the color scale. Caution should be used when changing these values because the color legend module will allow any numbers to be entered into these boxes. The minimum value will be placed at the bottom of the scale bar,the maximum at the top and the scale bar will be divided into equal pieces and labeled based on other settings in the module. Therefore, it is possible to have a data range from 0 to 10 and set the color legend to go from 100 to 500. In order to have the color scale bar labels exactly match the data, verify that the values type into the min and max values match the limits displayed below. If a more asthetically pleasing color scale is desired, either use an upstream clamp on the data before the module feeding color legend, or change the min and max value slightly to get even numbered increments. In the latter case, the color scale bar will not exactly match the data. However, if the data range is large and the changes to min and max are small, the differences should be negligible. The min and max of the input data range are displayed at the bottom of the control subpanel.

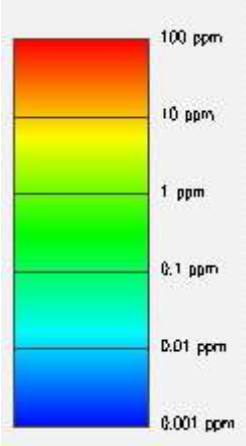


The Labeling Parameters control subpanel is shown above. The Labels On check box is used to specify whether or not the data labels are to be displayed with the color legend. A check in the box, the default, indicates that they are to be displayed. The Bounds On check box specifies whether or not a boundary box is to be drawn around the color legend scale bar. A check in the box, the default, indicates that the boundary is to be displayed.

The Label Density slider is used to specify the number of label increments to display in the color scale bar. The label density has different meanings depending on whether logarithmic or linear is chosen in the Main Legend Parameters control subpanel. If linear is chosen, the label density is the number of increments between the min value and the max value. The maximum label density is 31 and the number of label will be the label density plus 1. If logrithmic is chosen, the label density is the number of increments per decade. The maximum label density value allowed is 3 (however, any number larger than 3, up to 31, can be chosen and the display will display the same as if 3 was chosen) and the number of data labels displayed will be the number of decades times label density plus1.

The remaining sliders are all used to control the appearance of the color legend labels. The Label height slider sets the label height relative to the viewer units. The default is 0.030 and the range is from 0.001 to 0.060. The Label offset slider is used to specify how far away from the color scale bar the lables should appear. The value is in viewer units, has a default of 0.50 and a range from -3.00 to 3.00. Negative numbers place the label on the opposite side of the legend. The Width/Height slider determines the aspect ratio of the label text. A small width to height ratio creates narrow text and a large ratio creates wide text. The default is 0.90 and the range is from 0.20 to 1.50. The decimal Prec. slider is used to indicate the number of decimal places to display in the labels. The default in 1 and the range is from 0 to 6. Precision only applies to linear ranges. The Label font slider is used to determine which font style is used for the labels. The default is font 0 and the range is from 0 to 5. The font styles are not currently documented.

The Label Green, Label Red and Label Blue sliders are used for setting the color of the color legend labels. Each of these sliders range from 0 to 255 and the default for each is 255. The default setting produces white text. Setting each of these to 0 produces black text. Any other color can be used by varying these numbers.



Above is an example of a color scale bar produced by the color legend module.

Related Modules

-> Viewer

Legend_3D

This is a deprecated module whose function has been superceded by the Legend module.

General Module Function

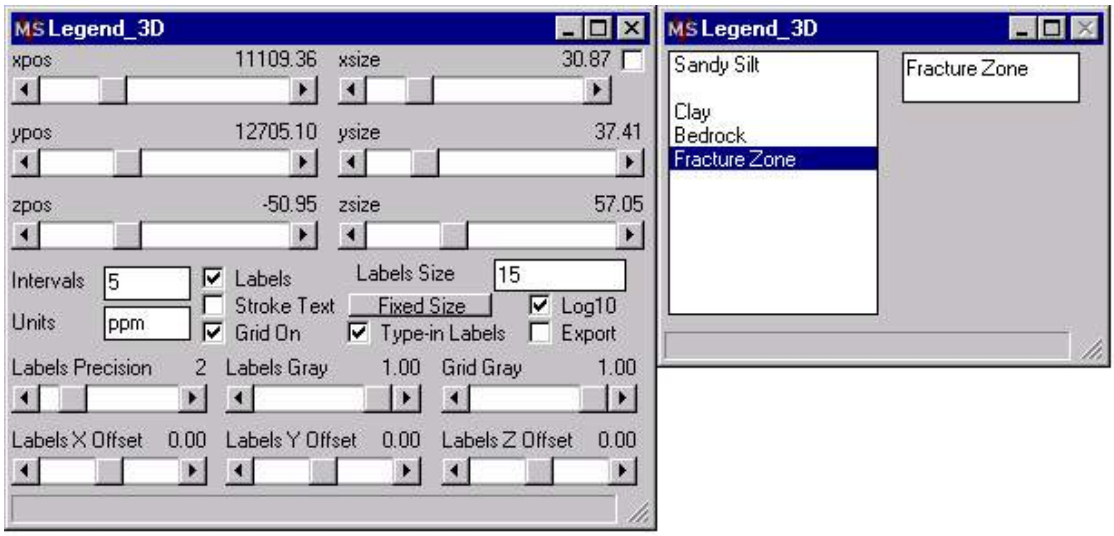
Legend3D takes a renderable object as input and produces a 3-dimensional color legend that, when rendered, displays that object's datamap within a 3D grid (box). This module is highly recommended if sending the Viewer output to a VRML file. Legend3D is rendered as a spatially attached object referenced to the input object. This module has capability for transforming Log10 data, displaying stroke text and type-in labels for customizing text descriptions.

Module Input Ports

Legend3D has only one input port. This port receives scaling data and a datamap from any single module. The module connected to this port should contain the data that the color scale is to reference.

Module Output Ports

Legend3D has two output ports. The first port sends the mesh and data of the 3D object for use by texture mesh, or Solid_3D_Set for special purpose rendering of the legend which is not possible with the provided controls. This second port sends color scale bar information directly to the viewer. Note that only stroke text will be supported if sending the View to VRML output.



Module Control Panel

The control panel for Legend3D is shown in the figure above. An unusual aspect of this module's controls is that sizing and placement of the legend box is simultaneously controlled by specifying the corners of the legend display area. Descriptions of the Legend3D controls are provided below:

xpos - The X coordinate for positioning the center of the 3D legend (in model units).

ypos - The Y coordinate for positioning the center of the 3D legend (in model units).

zpos - The Z coordinate for positioning the center of the 3D legend (in model units).

xsize - The X dimension of the 3D legend (in model units).

ysize - The Y dimension of the 3D legend (in model units).

zsize - The Z dimension (height) of the 3D legend (in model units).

Intervals - The number of labels and grid lines along the legend display.

Units - A type-in box for placing text for the variable's units. The default is ppm.

Labels - A toggle to display labels next to each interval. The default is on (checked).

Stroke Text - A toggle to switch label rendering to stroke text (which is limited to flat lying text in the x,y plane). This feature is required for sending text to VRML output. NOTE: Choosing this feature will change Label Size to stroke height.

Grid On - A toggle for placing gridlines at the specified intervals. The default is on.

Log10 - A toggle for transforming Log10 to real space. Use this feature if the data being passed to Legend3D is log10. The default is off.

Type-in Labels - A toggle which activate n number of type-in fields corresponding to the number of intervals. Choosing this toggle overrides the

labels and replaces with "default" which the user must overwrite with the desired text. Note that you may have blank fields if desired.

Export - A toggle to turn off rendering while leaving all labeling and gridlines thus allowing rendering by other modules. This feature is useful if special rendering tasks (such as texture mapping intervals, uneven intervals, etc.) are required. Once this toggle is set you may send Legend3D information to modules for special rendering tasks.

Labels Size - A type-in for the label font size. Note this type-in name changes to Stroke Height when Stroke Text is chosen.

Label Precision - A slider for setting the decimal precision of the labels.

Labels Gray - Controls the grayscale of the label text. A setting of 1.00 is white and 0.00 is black.

Grid Gray - Controls the grayscale of the gridlines. A setting of 1.00 is white and 0.00 is black.

Labels X, Y and Z Offset - Distance offset sliders in model coordinates for the legend labels. A negative value moves labels toward decreasing coordinate directions and a positive value moves the labels toward increasing coordinate directions.

Related Modules

-> LegendHoriz

-> LegendVert

-> Color_Legend

Geologic_Surface

This is a deprecated module that has been superceded by the geologic_surfaces and geologic_surface modules.

General Module Function

The Geologic Surface module extracts a single geologic surface from the set of surfaces output by Krig 3D Geology. This module allows visualization of the topology of a surface and\or the interaction of a set of individual surfaces, which can be exaggerated in the z axis according to the elevation of the surface at any given node point. Visualization of the interaction of a set of surfaces is accomplished by using separate Geologic Surface modules (with or without separate Krig_3D_Geology modules) that are connected to the same viewer. Geologic Surface also allows the user to either color the surface according to the surface z coordinate, or to produce an uncolored mesh that can be assigned a single color through the use of the **object selector** and **properties editor** in the Viewer module. This type of network could be used to visualize the intersection of a water table, colored all blue, and a geologic layer that is colored according to the surface elevation.

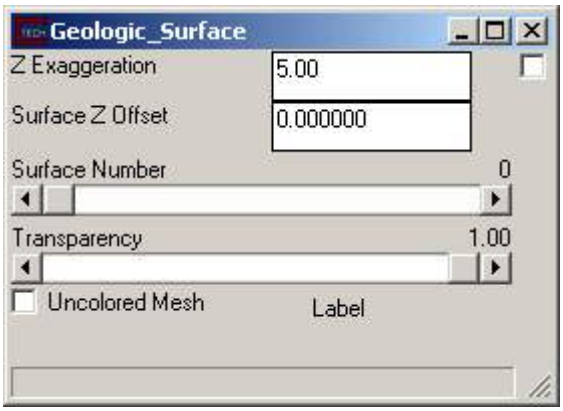
Module Input Ports

Geologic Surface has three input ports. The left port is for the integer of the surface number. The middle input port will accept a scale factor from the Explode and Scale module, if it is being used to set a master scale factor for

all of the modules in the network (that accept scale factors). Note using Explode and Scale with Geologic Surface is not necessary,.as Geologic Surface has its own scale factor parameter that can be used to set the surface scaling directly. The right input port accepts the 2-D surface data fields from Krig_3D_Geology. It does not accept data from any other module.

Module Output Ports

Geologic Surface has four output ports. The first port from the left is used by the geology to vistas module. (Geology to Vistas). The second port is used for Z-exaggeration. The third port, from the left, outputs the data field for the selected 2 dimensional surface. This port can be connected to any module that can process data fields, such as the isoline module, which can use the data field to display isolines on the geologic surface, the statistics module, which can report the characteristics of the suface data, or the generate axes module, which can generate a set of axes around the surface display. The right port outputs a renderable geometry of the surface that can be input to the viewer. This port can also be connected to the color legend module to set the limits of the displayed color scale.



Module Control Panel

The control panel for Geologic Surface is shown above. The Slider in the center of the panel is used to select which of the surfaces (that are being output by Krig 3d Geology) will be extracted and displayed by Geologic Surface. Note that the surfaces are numbered such that 0 is the top layer. The Z Exaggeration parameter is used to specify a factor by which all of the nodal depth or elevation values will be multiplied before display. The Surface Z Offset parameter specifies the distance that the surface willl be translated in the Z axis before display.. Note that these parameters affect the Z coordinates of the grid nodes, but not the values of the depth or elevation that are associated with those nodes by Krig 3D. Recall that in EVS each grid node has x,y,z, coordinates, and property values associated with the node. The property value assigned to the nodes by Krig_3D_Geology is the original (pre-scaling and translation) elevation or depth values. The Z Exaggeration and Surface Z Offset parameters affect the Z coordinates of the nodes of the quadrilateral elements in the mesh, but do not affect the property values associated with the nodes. This distinction is important for the user to remeber when passing the field data from Geologic Surface to other modules. As an example, if the field output of Geologic Surface is connected to the

Statistics module, the user will see that specifying different Z Exaggeration and Surface Z Offset parameters affects the *coordinate extents* reported by Statistics, but has no effect on the *parameter value* distribution reported (.ie., the min, max , mean, std deviation, and histogram bins).

scat_to_tet

General Module Function

The scat_to_tet module is used to convert scattered sample data into a three-dimensional tetrahedral unstructured mesh.

"Scattered sample data " means that there are discrete nodes in space. An example would be geology or analyte (e.g. chemistry) data where the x,y,z coordinates may also include a measured parameter. The data is "scattered" because there is not necessarily an implicit grid of data, either in the x-y plane or in 3D.

scat_to_tet uses a proprietary version of the Delaunay tessellation algorithm.

Module Input Ports

scat_to_tet has only one input port. Input to this port must contain scattered sample data. Currently, only the file_statistics module provides this output.

Module Output Ports

scat_to_tet contains one output port. The output port outputs the tetrahedral 3D elements having interpolated values based on the input data. Typically this port would be connected to external_faces or plume_volume.

Module Control Panel

This module has no control panel.

Related Modules

-> scat_to_tin

scat_to_unif

probe

This is a deprecated module. Use the probe functionality in the Az-El Advanced Panel.

General Module Function

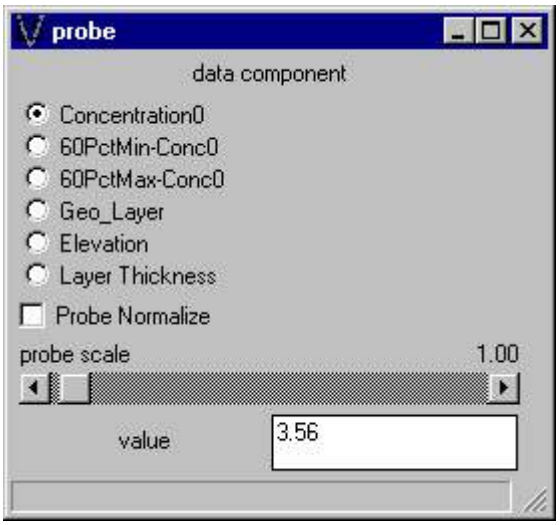
probe displays the data values present at the probe object's location in a mesh. The basic function of a probe is that of clicking with the mouse on any portion of an object and receiving a report of the objects scalar value. For example, connecting a probe to a slice plane would allow clicking anywhere on the plane and getting a report of the mapped value. When the probe object is not exactly on a node, the value shown is the result of an interpolation of the adjacent node data values. The probe can be attached to any 3D or 2D renderable object, and a glyph can be attached (and scaled as needed) to display at the probe's location.

Module Input Ports

probe has three input ports. The first (leftmost) input port should be a field with any mesh type and Node_Data. The second input port (optional) can be a glyph to represent the probe object. Any mesh can be used (that is, that of a teapot), but for convenience you can use the meshes defined in Geometries such as Arrow1, Cross2D, or Cross3D. The third input port must be attached to the red output of the object of interest. This red port connection is required since the probe's location comes from clicking within the rendered view.

Module Output Ports

probe has three output ports. The first output port (closest to the left) output is of marginal utility. It contains the mesh of the probe object and Node_Data that is the value of the data at the probe's current location. The second output port is a renderable version of the output field, usually the glyph object. The third sends a text output to the screen of the probe information. Currently this info is simply x,y,z data, but in the future this will be scalar data, vector data, or other data component information.



Module Control Panel

The user interface is shown in the figure above. Each parameter function is described below:

The data component radio buttons allow for choices which of the input field's components to probe. The component can be scalar or vector. The default is the first (0th) component. If node data labels are present, they are displayed.

The Probe Normalize toggle If off sizes of the probe proportional to the data component values at each node. If on, the probe is the probe scale size. The default is off.

The probe scale slider adjust the size of the probe. The default is 1.0. The range is -10.00 to 10.

The value field displays the value of the data at the probe's location. Where the probe does not fall exactly on a node, the value is interpolated from the

values of adjacent nodes. Note that the value displayed is accurate if the selected component was a scalar. If it was a vector, you see only the value of that component's first vector element.

The Probe Transform Editor is used to place the probe within the input field, as opposed to placing the probe with mouse clicks. This could be useful for moving the probe around within an unrendered mesh.

texture_mesh_set

This is a deprecated module.

General Module Function

texture_mesh_set(This module is available only in EVS-PRO),is an enhancement to texture_mesh in that it provides a means to project (texture map) images onto surfaces which are larger or smaller than the desired mapped extents. For example, the user may map a "postage stamp sized" aerial photo onto a very large topographic surface, or conversely, the user may project extents for an aerial photo that are outside the extents of the surface or object that the photo is projected onto.

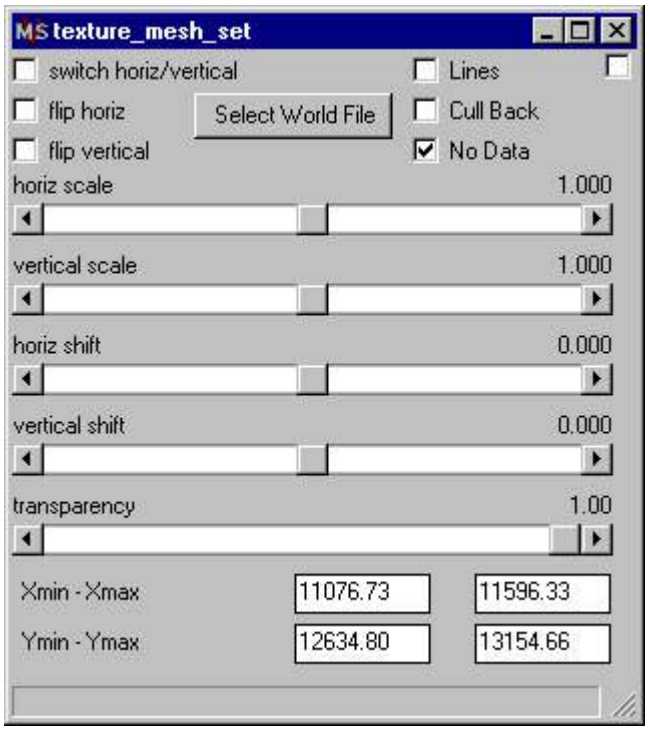
Additionally, this module can read world files (like TIF or BMP World Files) that provide georeferencing information for an image. This will set the extents for mapping the image for you to eliminate potential errors and speed processing.

Module Input Ports

texture_mesh_set has two input ports. The left input port should be surface data (such as 3DFACES) from the Read_DXF module or a geologic surface, and the right input port should be an image from the Read_TGA_BMP or Read_Image module.

Module Output Ports

texture_mesh_set has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version directly to the Viewer.



Module Control Panel

The control panel for texture_mesh_set is shown in the figure above.

The three toggles on the left provide a means to flip or mirror the image file for texture mapping. This is useful for quickly transforming an image file that has been produced using different scanning orders.

The Lines toggle determines whether lines are included in the output. The lines toggle will allow lines (such as a bounding box matching the image spatial extents) to be displayed.

Cull Back - A toggle for use with the transparency option. This feature may produce a better transparency when a surface is shown with other transparent surfaces, or if the same surface has high relief and is causing a confusing visualization. It's use will require experimentation to determine if it is appropriate.

The No Data toggle will remove the nodal data before texture mapping. If the input surfaces have nodal data, the texture map will be overlayed on colored surfaces. This is generally not desirable.

Horiz scale and vertical scale - Sliders for enlarging or shrinking the image. Horiz represents an East/West scaling, while vert represents North/South scaling

Horiz shift and vertical shift - Sliders for moving the image in East/West of North/South directions.

Transparency - A slider for adjusting the opacity of the texture mapped image. A value of 1.00 is completely opaque and a value of 0.00 is completely invisible.

Xmin - Xmax - Type-ins for the minimum and maximum X extents of the texture mapped image. As noted above, this can be anywhere inside or outside the extents of the surface being mapped onto.

Ymin - Ymax - Type-ins for the minimum and maximum Y extents of the texture mapped image.

tile_wall_dxf

General Module Function

This module is actually a macro module (e.g. a collection of three standard modules). tile_wall_dxf combines Read_DXF, Read_TGA_BMP, and tile_wall into a single module. It is equivalent to the following network fragment:

Module Input Ports

tile_wall _dxf has no input ports.

Module Output Ports

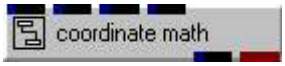
tile_wall _dxf has one output port which outputs a renderable geometry directly to the Viewer.

Module Control Panel

The control panels for tile_wall _dxf are the same as tile_wall, Read_DXF and Read_TGA_BMP.

coordinate_math

This is the olderdeprecatedversion of this module.



General Module Function

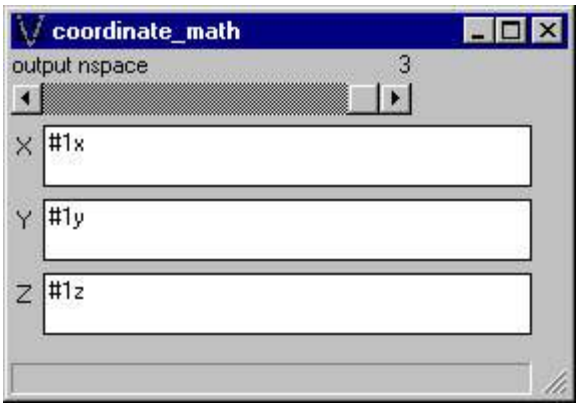
The coordinate math module is used to perform mathematical operations on any EVS mesh surface or 3D mesh. Up to four data fields can be input to coordinate_math. Mathematical expressions can involve any or all of these input fields. Some valuable uses of coordinate_math are for applying transformations of the z coordinates of a grid (such as a water table surface elevation), or for applying a mathematical operation on a surface (such as a fault surface displacement , rotation, etc.).

Module Input Ports

coordinate_math has four identical input ports. Each port can accept any type of mesh, but the meshes must have the same cell types and number of nodes (i.e., they must have identical geometries). At least one input port must be used and up to four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right. Coordinate data passed to ports one, two, three and four are referred to as #1x, #2x, #3x and #4x (y, or z) in the appropriate mathematical expression. You refer to individual coordinates in the expression as x, y or z. For example if you want to refer to X coordinate of the first input you would use #1x in the expression, if you want to refer to Z coordinate of the second input you should use #2z in the expression.

Module Output Ports

coordinate_math has two output ports. The first output port (closest to the left) contains the output mesh containing coordinates that are a result of the computation. The second output port is renderable version of the output mesh.



Module Control Panel

The control panel for field math is shown in the figure above. The types in boxes are for inputting mathematical expressions, which can be any valid C mathematical expression. Do not enclose in quotes or terminate with a semi-colon (that is, ";"). For example, the following expression converts input coordinates specified in the Polar coordinate system into output Cartesian coordinates:

X #1x*cos(#1y)

Y #1x*sin(#1y)

Related Modules

->field_math

explode_fields

General Module Function

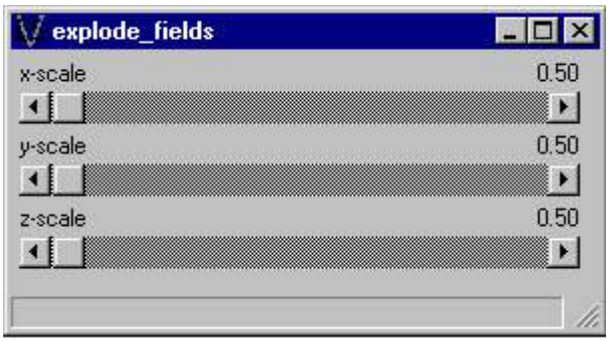
explode_fields takes each input field in the input array and translates it away from the common center of all the fields, which is computed by finding the midpoint of the bounding box. You can set the amount of translation via parameters. The grid and data for the input fields are not modified, only their transformation matrices.

Module Input Ports

explode_fields has one input port which can accept multiple inputs of any type of EVS field including fields (any type) or UCD mesh input.

Module Output Ports

explode_fields has two output ports. The first output port (closest to the left) outputs a mesh containing the transformed input fields. The second port outputs a renderable object.



Module Control Panel

The control panel for explode_fields is shown in the figure above. The x-scale, y-scale and z-scale sliders adjusts values specifying how much to scale the translation away from the center in the x,y or z directions. This value is in units of the bounding box of the original array of fields.

Related Modules

->Explode_and_Scale

explode_materials

General Module Function

explode_materials splits up a field into an array of fields based on the values of a particular element of the properties array. Each cell set in a field has an associated properties array. You may use this array to store anything related to the cell set, but it is commonly used to store material properties of the cells in the cell set.

The same array index is used to look up the value in the properties array of each cell set. Each cell set with a distinct value of that property goes into a separate output field. Whether the values are equal is all that is significant, the actual values are not significant.

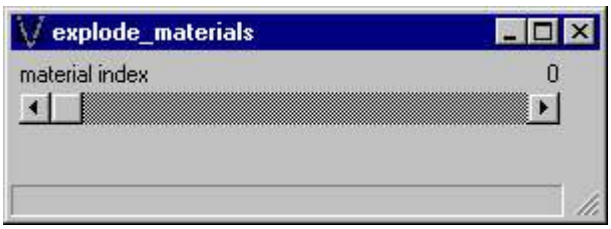
Note that this module compares floating point numbers for equality; you are responsible for ensuring that the values in the material property array are bit-for-bit equal or not equal, as desired.

Module Input Ports

explode_materials has one input port. Input to this port must contain the mesh to be split apart into multiple meshes on the output, according to the selected property value of each cell set.

Module Output Ports

explode_materials has two output ports. The first output port (closest to the left) outputs an array of meshes, with other data merged from the input field. The second output port outputs a group of renderable objects corresponding to the array of meshes.



Module Control Panel

The control panel for explode_materials is shown in the figure above. The material index slider selects which element of each cell set's material properties array to use to split the input. All cell sets' arrays use the same index.

Related Modules

->Explode_and_Scale

extract_layer

This is a deprecated modulewhose purpose was eliminated by changes to Krig_3D_Geology and Krig_3D.

General Module Function

Previously, there was no other parameter associated specifically with layers. Now, each layer can be assigned a material number. This allows for specifying layer material independent of layer number. Layer material is extracted as a data component using extract_layer. The material number replaces the layer number, so extract_layer should be after Explode_and_Scale in your network.

Module Input Ports

extract_layer has one input port that should be connected to or after Explode_and_Scale.

Module Output Ports

extract_layer has one output port which outputs a field with the Geo_Layer component modified.

Module Control Panel

The module has no control panel.

field_math

This is the olderdeprecatedversion of this module.

General Module Function

The field_math module (only in EVS PRO and MVS) is used to perform mathematical operations on nodal data fields. Up to four data fields can be input to field math. Mathematical expressions can involve any or all of these input fields. Data input to each of the four ports must be scalar. The output is also a scalar. If a data field contains more than one data component, you may select from any of them.

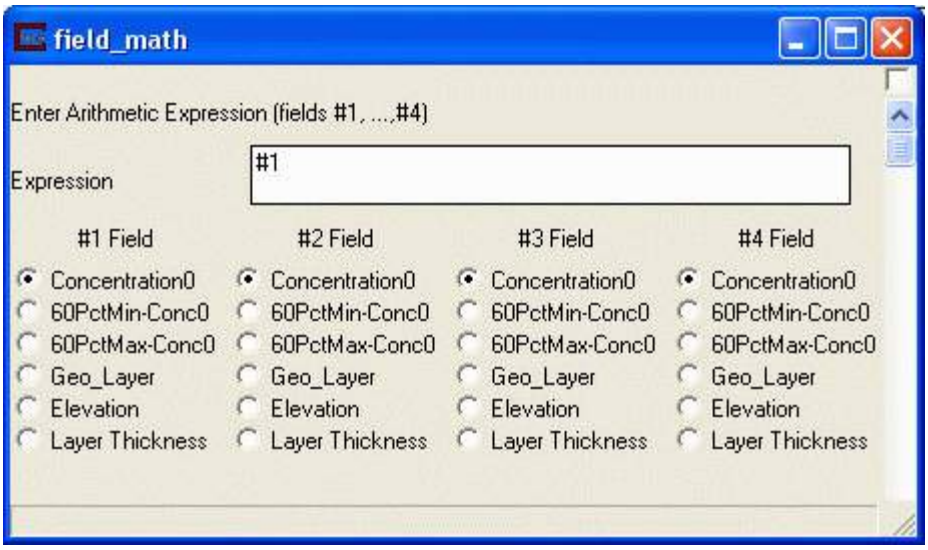
Module Input Ports

Field_math has four identical input ports and one special port. Each of the four ports can accept any type of mesh with scalar nodal or cell data, but the meshes must have the same cell types and number of nodes (i.e., they must have identical geometries). At least one input port must be used and up to four can be used. The first port is closest to the left and the ports are numbered sequentially in ascending order to the right. Nodal data passed to ports one, two, three and four are referred to as #1, #2, #3 and #4 in the mathematical expression. Again, the nodal data must be scalar. The rightmost port is the expression used to determine the mathematical operations. It is sometimes useful to create this expression with the string module driven by other parameters.

IMPORTANT: The mesh of only the first port is used when passing onto downstream modules. Therefore, only the first input port must contain the mesh data and all input ports must contain data on the mesh of the first input port. Also note that explode_and_scale can NOT be used after field_math since it uses a function of the material color number and geo_layer (a data component) to do exploding operations.

Module Output Ports

Field_math and cell_field_math each have three output ports. The first output port (closest to the left) contains a nodal data component which is the result of the mathematical computation. Mesh data is referenced to the mesh in input field one. The second output port is renderable object. The last port is the expression.



Module Control Panel

The control panel for field_math is shown in the figure above. The edit field is used for entering the mathematical expression to be computed. Valid mathematical operators and functions are listed below. Refer to input data components by #[input port number].

Pop-UpAvailable Mathematical Operators here.

Jump to a list of availableMathematical Operators

For example, to multiply the first field by the third field and then divide the result by the second field, enter:

#1 * #3 / #2

or, this operation can alternatively be completed using C style functions;

divide(multiply(#1,#3),#2).

An example of a less trivial equation to blend the values of (interpolate between) two different UCD files having nodal data with a logarithmic distributions is:

log10(pow(10,#1)*0.75+pow(10,#2)*(1.0-0.75))

In the Animator module the default equation for field_math is:

log10(pow(10.0,#1)*%lf+pow(10.0,#2)*(1.0-%lf))

which performs interpoloation of log processed data. To perform interpolation between non-log processed data use:

#1 * %lf + #2 * (1.0-%lf)

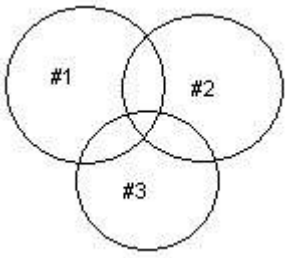
We can define a max function to find a bigger number between two numbers. In the example below, the parenthesis will return 1 if the conditional statement inside is right or return 0 if it is wrong.

(#1 > #2)*#1 + (#1 <= #2)*#2

we can define a min function in the same way.

(#1 < #2)*#1 + (#1 >= #2)*#2

Union of #1 > 5, #2>2.5 and #3 > 100



to define a union above, enter

(#1 > 5) | (#2 > 2.5) | (#3 > 100)

to get the intersection, replace | with &.

(#1 > 5) & (#2 > 2.5) & (#3 > 100)

Mathematical Operators

An expression can include the C-style numeric and logical operators shown in the following table:

Operator	Meaning
+	Addition
-	Subtraction

*	Multiplication
/	Division
%	Modulo
&	Logical and
|	Logical or
^	Logical xor
-value	Unary minus

Order of precedence

Mathematical expressions follow C's order of precedence. Use parentheses to override the default order.

Functions

The following functions can be used to perform mathematical and logical operations:

Function	Description
abs acos add (+) asin atan cos cosh divide (/) exp log log10 modulo (%) multiply (*) pow sin sinh sqrt sub (-) tan tanh	Perform a mathematical operation. Several of these operations can be also be performed with the numeric operators shown in parentheses.
and (&) or (|) xor (^)	Logical operations can be performed with either functions or the operators shown in parentheses.

Related Modules

->coordinate_math

rotate

General Module Function

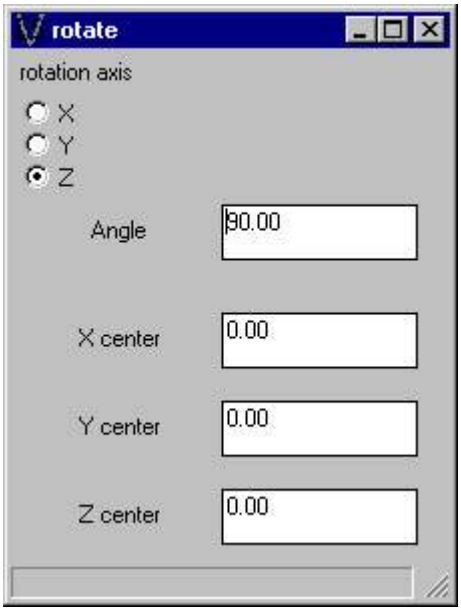
The rotate module performs a viewer transformation of any EVS object about any of the three principal axes and also allows rotation about a user defined center.

Module Input Ports

Rotate has only one input port which accepts a structured or unstructured mesh. There can be several data components, either vector or scalar, in the data field.

Module Output Ports

Rotate has two output ports. The first output port (closest to the left) outputs viewer transformation information which can be passed to downstream modules. The second output port sends renderable version of the transformed object to the viewer.



Module Control Panel

The control panel for rotate is shown in the figure above. The rotation axis toggles allow a choice of X, Y or Z for axis of rotation. The Angle type-in accepts positive or negative rotation angles with 0.00 relative to the principal Y axis (commonly north). The X center, Y center and Z center type-ins are provided to allow rotation about a user defined center. For example if the model centroid is x-5000, y-3000 and z-200, these values should be entered to rotate the model about it's centroid.

Related Modules

->Translate

surf_optimize

General Module Function

The surf_optimize module is used to optimize 2D uniform meshes with a scalar node data component. The node data is assumed to be an elevation (height) at each grid point. This type of mesh is typically used in GIS applications such as a Digital Elevation Model (DEM), or a x, y, attribute grid from another software package.

The module creates a mesh containing "optimum" triangles. In this usage, the term "optimum" means that the largest possible triangles that satisfy a specified tolerance criteria are met. The tolerance determines the maximum difference of elevations between any point in the output triangular mesh and the corresponding point of the input mesh. That is, the output surface of the elevation lies within the specified tolerance from original surface of elevation.

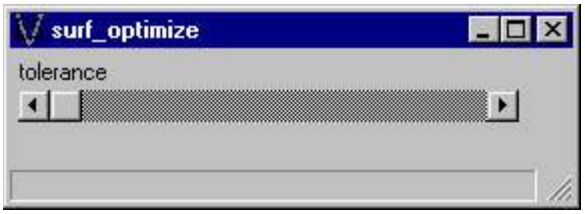
This module is used to create a coarse mesh in the regions where the surface of elevation changes gradually and a fine mesh in the regions where the surface changes rapidly. Increasing the tolerance results in a less accurate mesh containing fewer triangles and thus leads to a corresponding improvement in rendering performance. The surf_optimize module does not produce the elevation surface itself, it just creates a flat triangular mesh that can be extruded into a surface of elevation produced using the surf_plot module. The algorithm used in this module is described in: Automatic Generation of Triangular Irregular Networks using Greedy Cuts by Claudio T. Silva, Joseph S. B. Mitchell and Arie Kaufman, Visualization 95 Proceedings.

Module Input Ports

surf_optimize has two input ports. Input to the first port must contain a 2D uniform mesh with scalar node data.

Module Output Ports

surf_optimize has two output ports. The first output port (closest to the left) outputs a triangular mesh with a elevation node data. a renderable version of the output field. The second output port sends a renderable version of the output field.



Module Control Panel

The control panel for surf_optimize is shown in the figure above. The tolerance slider is provided to adjust the accuracy of the output mesh and the number of triangles generated. Increasing tolerance results in a less accurate mesh containing fewer triangles and thus leads to a corresponding improvement in rendering performance. The default is 10% of the difference between the max and min elevation values.

Related Modules

->surf_plot

translate

General Module Function

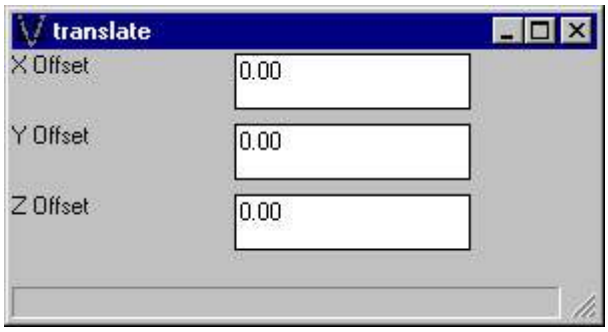
The translate module is used to translate or offset any three dimensional grid coordinate system to a new location. One possible use for this module would be to convert a modflow or mt3d grid (having a grid origin of 0,0,0) to the state plane coordinate system of the modeled area. Translate can offset one or all of the X, Y or Z values of a structured or unstructured mesh by a user specified value.

Module Input Ports

Translate has only one input port which accepts a structured or unstructured mesh. There can be several data components, either vector or scalar, in the data field.

Module Output Ports

Translate has two output ports. The first output port (closest to the left) outputs the same connectivity data as the input mesh with only the coordinate values adjusted to reflect the new translated coordinate information. The second output port sends a renderable object to the viewer.



Module Control Panel

The control panel for translate is shown in the figure above. Edit fields are available for adjusting the X offset, Y offset, and Z offset. The default for each of these edit fields is 0.0 and the available range is unrestricted.

Related Modules

->Rotate

ViewScale

General Module Function

You should useExplode_and_Scalein all applications instead of ViewScale.

The ViewScale module is used to scale a mesh in the X, Y and Z directions independent of each other. Nodal data is not scaled by the ViewScale module. ViewScale only affects the way the mesh will be displayed in the viewer. It does not affect the nodal data components or the actual mesh. NOTE: ViewScale was previously named "scale". All old applications will still

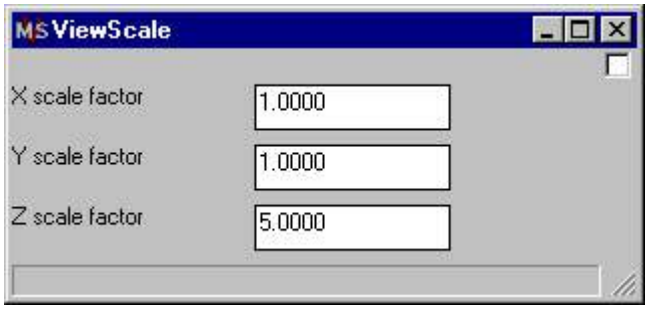
show the scale module in the network, but the functionality will be ViewScale.

Module Input Ports

ViewScale has only one input port. Input to this port must contain mesh data of any type mesh. Nodal data can be input to scale.

Module Output Ports

ViewScale has two output ports. The first output port (closest to the left) outputs the same mesh data as the input mesh with the transformation matrix containing the scaling information. Nodal data piped to the input port passes through scale unaffected. The second output port is not used in EVS.



Module Control Panel

The control panel for ViewScale is shown in the figure above. Edit fields are available for adjusting the X scale factor, Y scale factor, and Z scale factor. The default for each of these edit fields is 1.0 and the available range is -1.0e+08 to 1.0e+08. The numbers entered into the edit fields represent a scale factor relative to the viewer's dimensions coordinate system.

Related Modules

Explode_and_Scale

TrueScale TrueScale

vector_scale

This is a deprecated module.

General Module Function

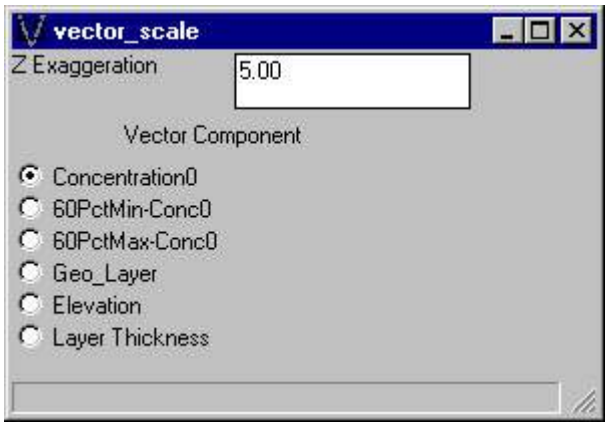
The vector_scale module is used to scale the Z component of vector data. It does not affect the actual mesh.

Module Input Ports

Vector_scale has only two input ports. Input to the blue port must contain a 3D vector data component. The white/brown port receives z exaggeration data from the Explode_and_Scale module.

Module Output Ports

Vector_scale has two output ports. The first output port (closest to the left) outputs z exaggeration data as does the Explode_and_Scale module. The second output port contains the scaled vectors



Module Control Panel

The control panel for vector_scale is shown in the figure above. The Z_Exaggeration input field adjusts the Z scale factor for the vectors. The radio buttons provide a means to directly select the vector component from among the data components in the input field.

Related Modules

->scale

LegendVert

General Module Function

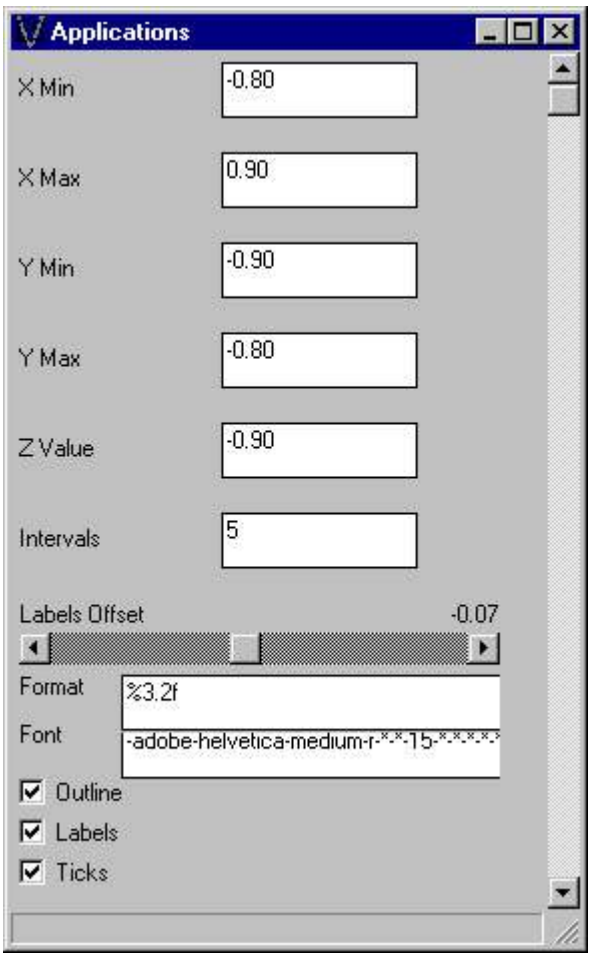
LegendVert is an obsolete module. Use Color_Legend. LegendVert takes a renderable object as input and produces a vertical color legend that when rendered displays that object's datamap. LegendVert is unlike Color_legend in that it *cannot* transform log10 data to normal space, but LegendHoriz *can* directly re-produce the datamap within the legend box.

Module Input Ports

LegendVert has only one input port. This port receives scaling data and a datamap from any single module. The module connected to this port should contain the data that the color scale is to reference.

Module Output Ports

LegendVert contains one output port. This port sends color scale bar information directly to the viewer. The color scale bar is a special type of viewer object which is not affected by viewer settings.



Module Control Panel

The control panel for LegendVert is shown in the figure above. An unusual aspect of this module's controls is that sizing and placement of the legend box is simultaneously controlled by specifying the corners of the legend display area. Descriptions of the LegendVert controls are provided below:

X Min The lower left screen coordinate for the legend display.

X Max The lower right screen coordinate for the legend display.

Y Min The upper left screen coordinate for the legend display.

Y Max The upper right screen coordinate for the legend display.

Z Value The Z screen coordinate for the legend display. Adjustment of this parameter allows placement of the legend either in front of or behind other viewer objects. Try changing the value to a negative number and notice the legend is rendered behind other objects in the viewer.

Intervals The number of labels along the legend display.

Labels Offset Distance offset in screen coordinates of the legend labels. A negative value places labels to the left of the display and a positive value places labels to the right.

Format Type-in for the label precision requiring C syntax type-in. For example: to change the decimal precision from 2 to 1 on the default format, the user would change the type-in from %3.2f to %3.1f.

Font Type in for adobe type fonts. The default is Helvetica medium, but this spec can be changed to any of the adobe level II fonts. Example: to change from Helvetica medium to times bold, the user would change the characters <Helvetica-medium> to <times-bold> in the supplied character string. Do not delete or replace any of the other characters in the string.

Related Modules

-> LegendHoriz

-> Color_Legend

FPlane

This is deprecated module superseded by create_grid.

General Module Function

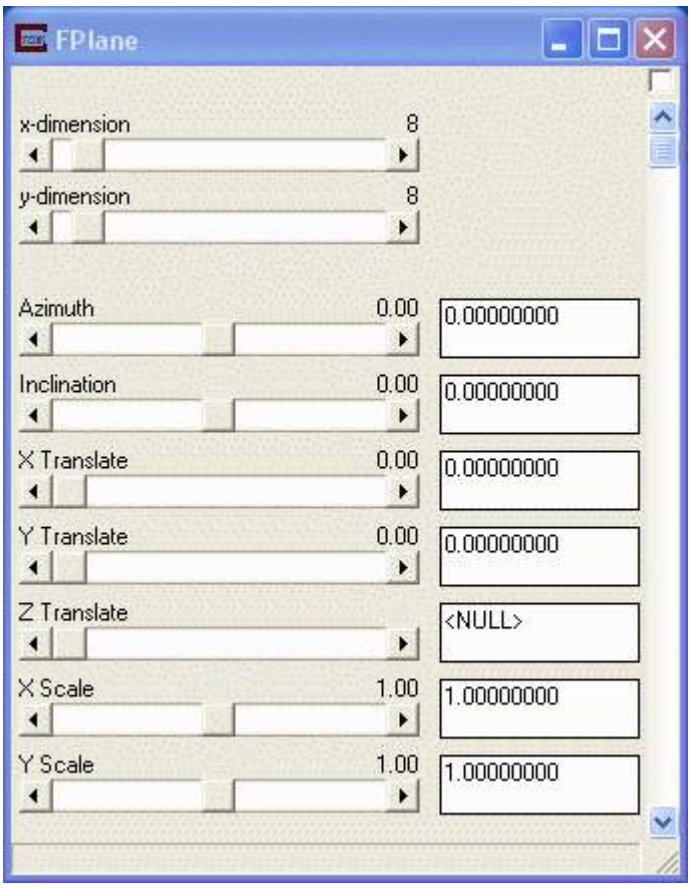
FPlane generates a 3D plane of variable size. This FPlane can be transformed in three dimensions. It will be located at the center of the input field's Z extents, and will be X and Y extents long. The plane is useful as a sampling or slicing object input to macros such as streamlines, advector, and cut.

Module Input Ports

FPlane has one input port which accepts any 3D mesh data.

Module Output Ports

FPlane has two output ports. The first port passes a 2D mesh representing the plane The second port passes the renderable geometry to the viewer.



Module Control Panel

The user interface for FPlane consists of numerous sliders for manipulation of the resulting plane. The x and y dimension sliders control the discretization of the nodes on the plane. The Azimuth and inclination sliders control the plane orientation. The translation sliders control the plane location and the scale sliders control the size of the plane. The FPlane is horizontal when originally instanced in EVS. All manipulations with the parameters described below are relative to an originally horizontal plane.

x and y-dimension Integer sliders to set the X and Y dimensions of the plane. The default for both is 8. Their range is from 2 to 100.

Transformation Editor An alternate XformEditor that controls the position, orientation, and scale of the object in 3D space. This panel is NOT as user friendly as the sliders on the initial panel (described below) so it's use is NOT recommended unless transformations cannot be achieved with the sliders below.

X , Y and Z Translate Locates the plane within the input field extents. The default extents represent the extents of the input field however the type-ins allow setting these values outside the original extents.

X and Y Scale Scales the plane relative the scale of the input field. The initial value is 1.00 representing 100% of the input field. Therefore, values between 0 and 1.00 will size the plane as percentages of the input x and y extents.

Type-ins allow setting these values greater than the x and y extents of the model.

Related Modules

->Advector

-> **Streamlines**

LegendHoriz

General Module Function

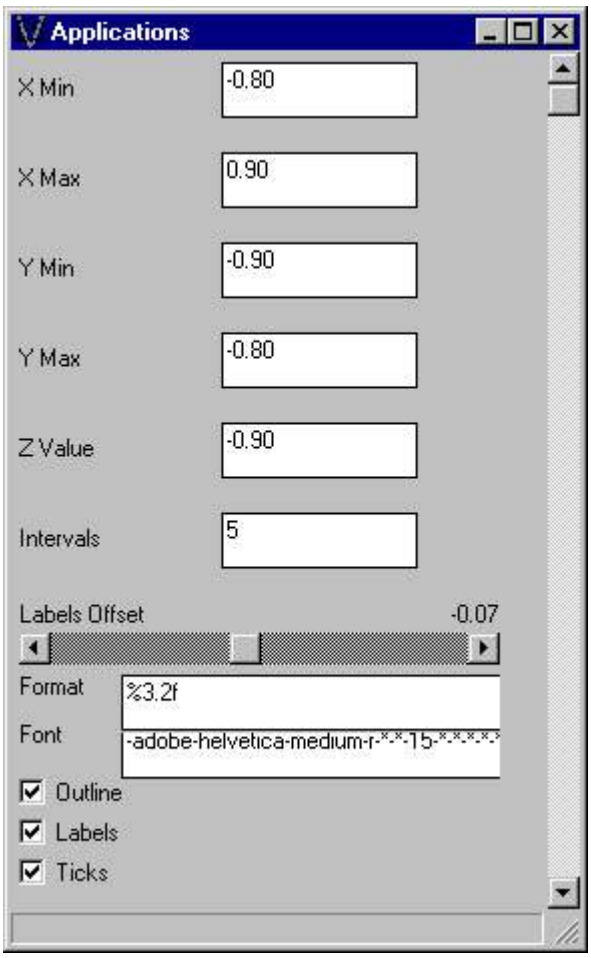
LegendHoriz is an obsolete module. Use Color_Legend. LegendHoriz takes a renderable object as input and produces a horizontal color legend that when rendered displays that object's datamap. LegendHoriz is unlike Color_legend in that it *cannot* transform log10 data to normal space, but LegendHoriz *can* directly re-produce the datamap within the legend box.

Module Input Ports

LegendHoriz has only one input port. This port receives scaling data and a datamap from any single module. The module connected to this port should contain the data that the color scale is to reference.

Module Output Ports

LegendHoriz contains one output port. This port sends color scale bar information directly to the viewer. The color scale bar is a special type of viewer object which is not affected by viewer settings.



Module Control Panel

The control panel for LegendHoriz is shown in the figure above. An unusual aspect of this module's controls is that sizing and placement of the legend box is simultaneously controlled by specifying the corners of the legend display area. Descriptions of the LegendHoriz controls are provided below:

X Min The lower left screen coordinate for the legend display.

X Max The lower right screen coordinate for the legend display.

Y Min The upper left screen coordinate for the legend display.

Y Max The upper right screen coordinate for the legend display.

Z Value The Z screen coordinate for the legend display. Adjustment of this parameter allows placement of the legend either in front of or behind other viewer objects. Try changing the value to a negative number and notice the legend is rendered behind other objects in the viewer.

Intervals The number of labels along the legend display.

Labels Offset Distance offset in screen coordinates of the legend labels. Negative values are below the display and positive are above.

Format Type-in for the label precision requiring C syntax type-in. For example: to change the decimal precision from 2 to 1 on the default format, the user would change the type-in from %3.2f to %3.1f.

Font Type in adobe type fonts. The default is Helvetica medium, but this spec can be changed to any of the adobe level II fonts. Example: to change from Helvetica medium to times bold, the user would change the characters 'Helvetica-medium' to 'times-bold' in the supplied character string. Do not delete or replace any of the other characters in the string.

Related Modules

-> LegendVert

-> Color_Legend

extract_cell_comp

General Module Function

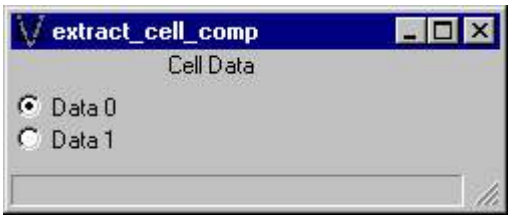
The extract cell comp module extracts a single cell based data component from a field containing cell based data. Extract cell component can extract cell based scalar data components or vector components. Scalar components will be output as scalar components and vector components will be output as vector components, one component for each cell.

Module Input Ports

Extract cell comp has only one input port which accepts any mesh type with cell based data.

Module Output Ports

Extract cell comp has two output ports. The first port (closest to the left) contains a reference to a merged object that contains the new cell based data, plus references to all other unchanged objects in the input Mesh. The second port is undocumented at this time.



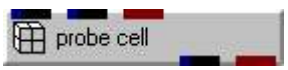
Module Control Panel

The control panel for extract cell component is shown in the figure above. The data component radio button list contains all of the cell data components piped into the input port. Only one component can be selected at a time and the default selection is the first (0th) data component. Any modules downstream of extract cell component will only receive the selected data component.

Related Modules

->cell_to_node

probe_cell



(This module has been deprecated. Its functionality has been surpassedby the interactive_labelsmodule.)

General Module Function

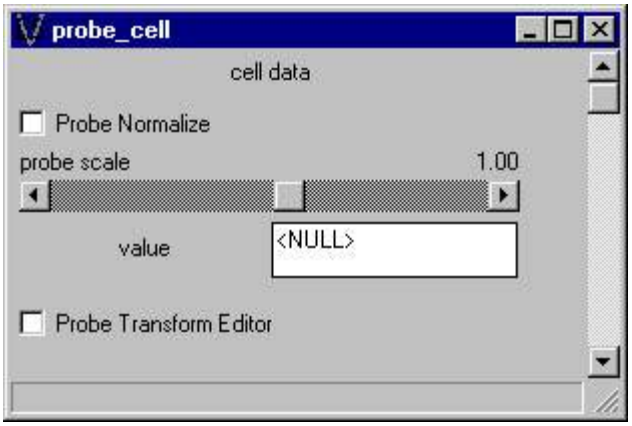
probe_cell has identical functionality to probe except it's use is intended for cell data input. probe_cell displays the data values present at the probe_cell object's location in a mesh. The basic function of a probe is that of clicking with the mouse on any portion of an object and receiving a report of the objects scalar value. For example, connecting probe_cell to a volume of cell data would allow clicking anywhere on the object and getting a report of the value of the cell the probe falls within. The probe_cell module can be attached to any 3D or 2D renderable object containing cell data (as opposed to nodal data), and a glyph can be attached (and scaled as needed) to display at the probe's location.

Module Input Ports

probe_cell has three input ports. The first (leftmost) input port should be a field with any mesh type and Cell_Data. The second input port (optional) can be a glyph to represent the probe_cell object. Any mesh can be used (that is, that of a teapot), but for convenience you can use the meshes defined in Geometries such as Arrow1, Cross2D, or Cross3D. The third input port must be attached to the red output of the object of interest. This red port connection is required since the probe's location comes from clicking within the rendered view.

Module Output Ports

probe_cell has three output ports. The first output port (closest to the left) output is of marginal utility. It contains the mesh of the probe_cell object and Nodal_Data that is the value of the data at the probe's current location. The second output port is a renderable version of the output field, usually the glyph object. The third sends a text output to the screen of the probe_cell information. Currently this info is simply x,y,z data, but in the future this will be scalar data, vector data, or other data component information.



Module Control Panel

The user interface is shown in the figure above. Each parameter function is described below:

The data component radio buttons allow for choices which of the input field's components to probe. The component can be scalar or vector. The default is the first (0th) component. If node data labels are present, they are displayed.

The Probe_cell Normalize toggle If off sizes of the probe proportional to the data component values at each node. If on, the glyph is the probe scale size. The default is off.

The probe scale slider adjusts the size of the probe. The default is 1.0. The range is -10.00 to 10.

The value field displays the value of the data at the probe's location. Note that the value displayed is accurate if the selected component was a scalar. If it was a vector, you see only the value of that component's first vector element.

The Probe Transform Editor is used to place the probe within the input field, as opposed to placing the probe with mouse clicks. This could be useful for moving the probe around within an unrendered mesh.

Fcircle3D

This is deprecated module superseded by create_spheroid.

General Module Function

Fcircle3D allows you to place a 2D circle in the 3D space of your model. By default the circle will be horizontal and centered in your model space. You may scale, rotate and translate the circle interactively with your mouse.

Module Input Ports

Fcircle3D has one input port which accepts any 3D mesh data.

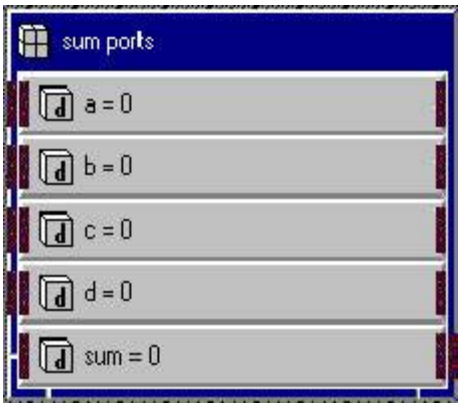
Module Output Ports

Fcircle3D has two output ports. The first port passes a 2D mesh representing the circle The second port passes the renderable geometry to the viewer.

Module Control Panel

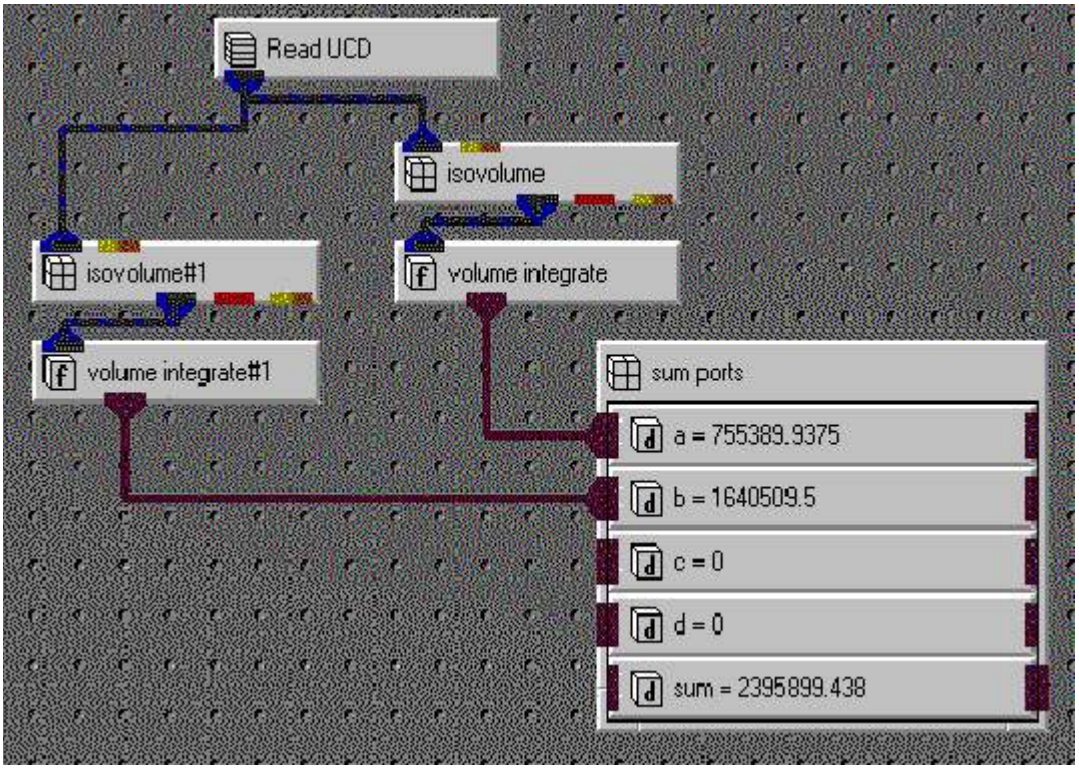
The user interface below allows you to scale the circle and set the number of sides (default is 12). The number of sides can be as low as 2, in which case the circle becomes a line segment.

Sum_Ports

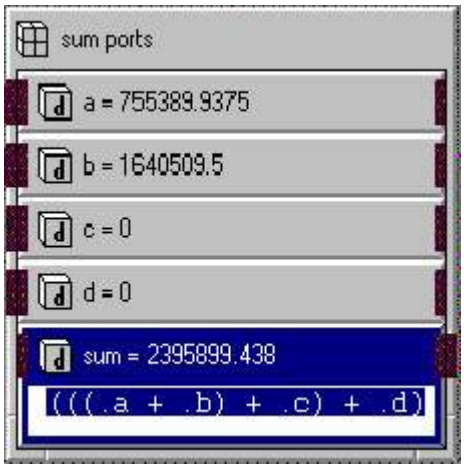


sum_ports is a deprecated module that has been superseded by float_math.

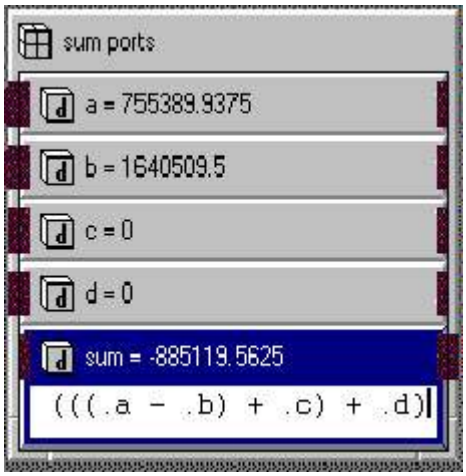
This module provides a simple means to add the numerical output from up to 4 input ports. By using multiple sum_ports modules, any number of values may be combined. Also, the module's expression can be easily edited to perform subtraction or other math operations.



If you double-click on the sum module, it opens to reveal the math expression inside.



The above expression merely adds the inputs from all four input ports. But we can change it!



The above example takes the first port minus the second port (a-b). The result is negative because b is greater than a.

Be careful: If you connect to one of the inputs and then delete the connection, that module will be undefined (?). This will make the sum undefined. You must reconnect something to that port, open the port and set it to zero, OR change the expression to not use it.

The output of sum (right output port) is the numeric value.

city_plot

General Module Function

This is a deprecated module.

city_plot creates a block on each node of a 2D input mesh (such as a slice plane). The color and height (out of the grid) of the blocks are determined independently by selectable input data components at that node. The lower left (min X and Y) corner of the block is at the node's coordinate, so the city

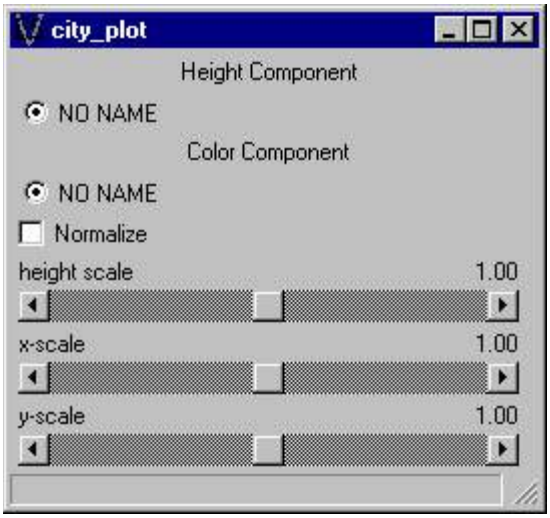
plot extends over the dimensions of the field by one block width (in X) and length (in Y).

Module Input Ports

city_plot has one input port. Data passed to this port should contain the 2D field to be plotted.

Module Output Ports

city_plot has two output ports. The leftmost output port creates an unstructured Mesh of cell type Quad containing the blocks. The second output port sends a renderable version of this mesh to the viewer.



Module Control Panel

The control panel for city_plot is shown in the figure above:

The Height Component selection determines which component of the input to map to the height of the block at that point of the grid.

The Color Component selection determines which component of the input to map to the color of the block at that point of the grid.

The Normalize toggle when turned on normalizes the heights of the blocks to a constant value set by the height scale, and the Height component is ignored.

The height scale slider scales the input height value by this amount. If normalize is on, this value becomes the height of the blocks. Height is measured in the coordinate system of the input field.

The x-scale slider scales the X size of each block relative to the size of that block's cell width. scale_x of 1 means the block takes up the full width of the cell and abuts the next block (in X); scale_x of 0.5 means the block takes up 1/2 the width of the cell.

The y-scale slider scales the Y size of each block relative to the size of that block's cell width. scale_y of 1 means the block takes up the full length of the

cell and abuts the next block (in Y); scale_y of 0.5 means the block takes up 1/2 the length of the cell.

isovolume2

This is a deprecated module whose function has been superceded by plume_volume and plume_area

General Module Function

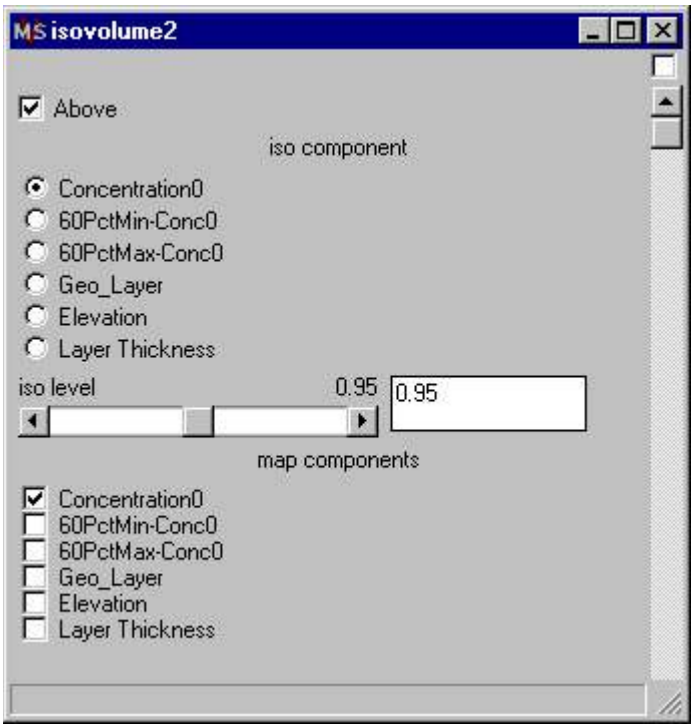
The isovolume2 module is a subsetting module, which produces a volume of the data inside or outside of an isosurface. Isovolume2 cuts the data at a specified subsetting level, and outputs those portions (volumes) of the input mesh that are below or above (default) the specified subsetting level. It can be used with 2D or 3D input data. If the input is 2D, the red port can be connected directly to the Viewer. If 3D, the output must be further processed (e.g. with external_faces) to yield surfaces for rendering. Multiple isovolume2 (subsetting) modules can be used in serial connection to subset an input field by more than one criteria. The isovolume of one component can be colored by the scalar value of another component (for example, the isovolume of concentration can be colored by uncertainty).

Module Input Ports

Isovolume2 has two input ports. The leftmost port accepts unstructured mesh data. The second port provides a means to share the subsetting level of other modules.

Module Output Ports

Isovolume2 has three output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains cells representing the external faces of the isovolume. Nodal data sent to this output port consists of the nodal data of the map component in the isovolume. The second port outputs a rendered geometry directly to the Viewer. The third output port provides a means to share the subsetting level of this module with others.



Module Control Panel

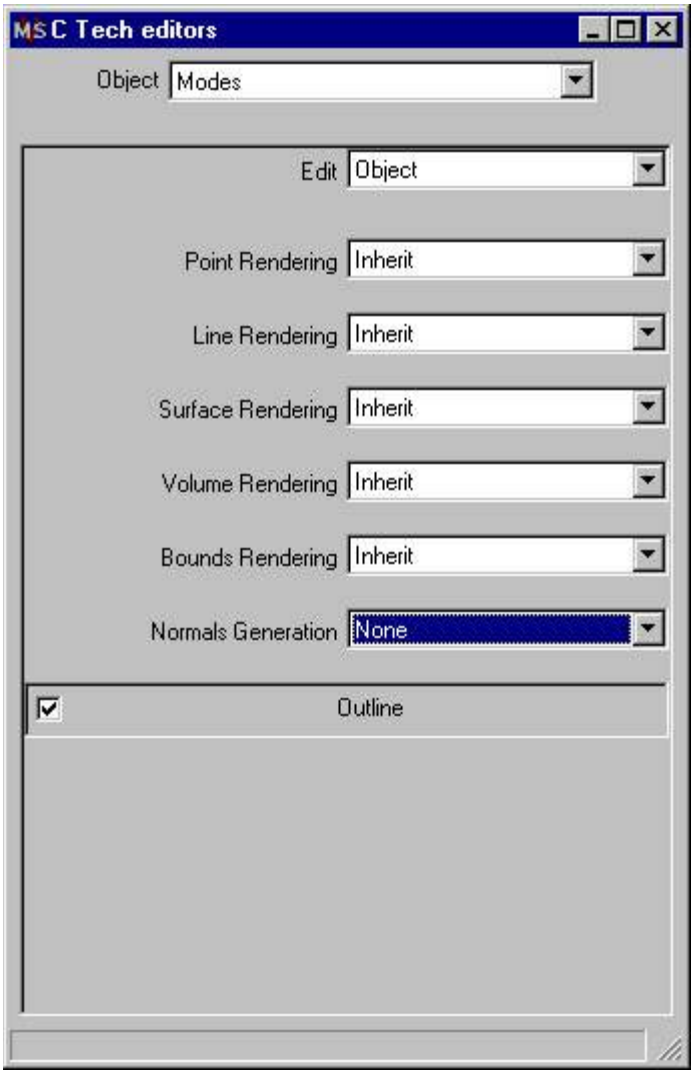
The control panel for isovolume is shown in the figure above. The left column, labeled iso component, consists of a series of radio buttons representing all available model data components. The right column, labeled map component, is a series of check boxes next to the same model data components. A user input field labeled iso level is centered at the bottom of the dialog box. To the right of this box is are two text lines displaying the minimum and maximum values for the data selected under iso component. Below the iso level input field is a check box labeled "Data Above(on) or Below(off) Iso Level"

Iso component refers to the model data component used to create the subset of the original input field. When an iso component is selected, the min and max values of the variable are displayed in the lower right corner of the dialog box. The default iso component is the first (0th) component in the column.

Map component determines which model data components in an unstructured mesh will be sent to the leftmost output port. Also, the first map component selected will be used to color the isovolume. For example, choosing iso component concentration and map component uncertainty will create a volumetric subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The iso level edit box is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default iso level value is the arithmetic average of the minimum and maximum values in the iso component.

The Data Above(on) or Below(off) Iso Level check box (Above Box) is used to display data above the iso level or below the iso level. For example, to display a volumetric subset of all concentrations greater than or equal to 1 ppm, set iso level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the isovolume of 1 ppm and below, simply turn the Above Box off (no check in box).



Normal Generations is set to none in Modes object of the view editor. This is the only difference of isovolume2 module with the isovolume module.

Related Modules

->isosurface

-> isolines

-> plume_shell

volume_render & volume_render_gl

General Module Function

volume_render and volume_render_gl are obsolete modules that have been replaced by the volume_renderer module. We strongly recommend that you replace all applications using either of these modules with the new volume_renderer module immediately.

These modules will be dropped in a future release.

Load_Datamap

General Module Function

THIS IS AN OBSOLETE MODULE. YOU SHOULD USE Datamap_Editor instead.

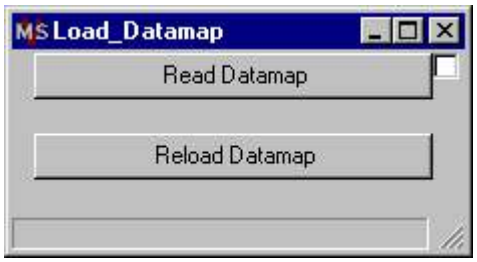
This simple module allows the user to apply one of 25 pre-made datamaps to any object(s) in the Viewer. These datamaps are mostly multi-range datamaps, but also include popular datamaps for coloring geologic materials. The user simply connects the Load_Datamap module to the object's red port and selects the datamap of interest. These datamaps are ASCII files which can be edited, but it is recommended that you rename any edited datamaps.

Module Input Ports

Load_Datamap has one input port (red) which should be connected to the module whose rendered object you would like to color.

Module Output Ports

None



Module Control Panel

The Load_Datamap control panel is shown above. Clicking on the Read Datamap push button opens a standard windows file browser. The filename (with *.dmp extension) and location can be specified in this browser. Selecting the file and clicking OK activates the loading of the datamap. The Reload datamap push button should be used in case any changes are made in the datamap editor which you want to override.

georeference_image

georeference_imagehas been superceded by a standalone utility programGeoreference Imagewhich can create world files or .gcp (ground control point) files for images.

General Module Function

The georeference_image module (only available in EVS-PRO and MVS), provides a means to create a world file or directly georeferenced an orthorectified image. The process is straightforward. The users sets thepositionof two crosshairs to specify pixels with known x-y locations. Those locations are typed into the modules window. This information is used to determine the overall spatial extent of the image and thereby properly georeferenced the image.

Module Input Ports

georeference_image has one input port which should be an image from the Read_Image module.

Module Output Ports

georeference_image has one output port that should be connected directly to the texture_map module.

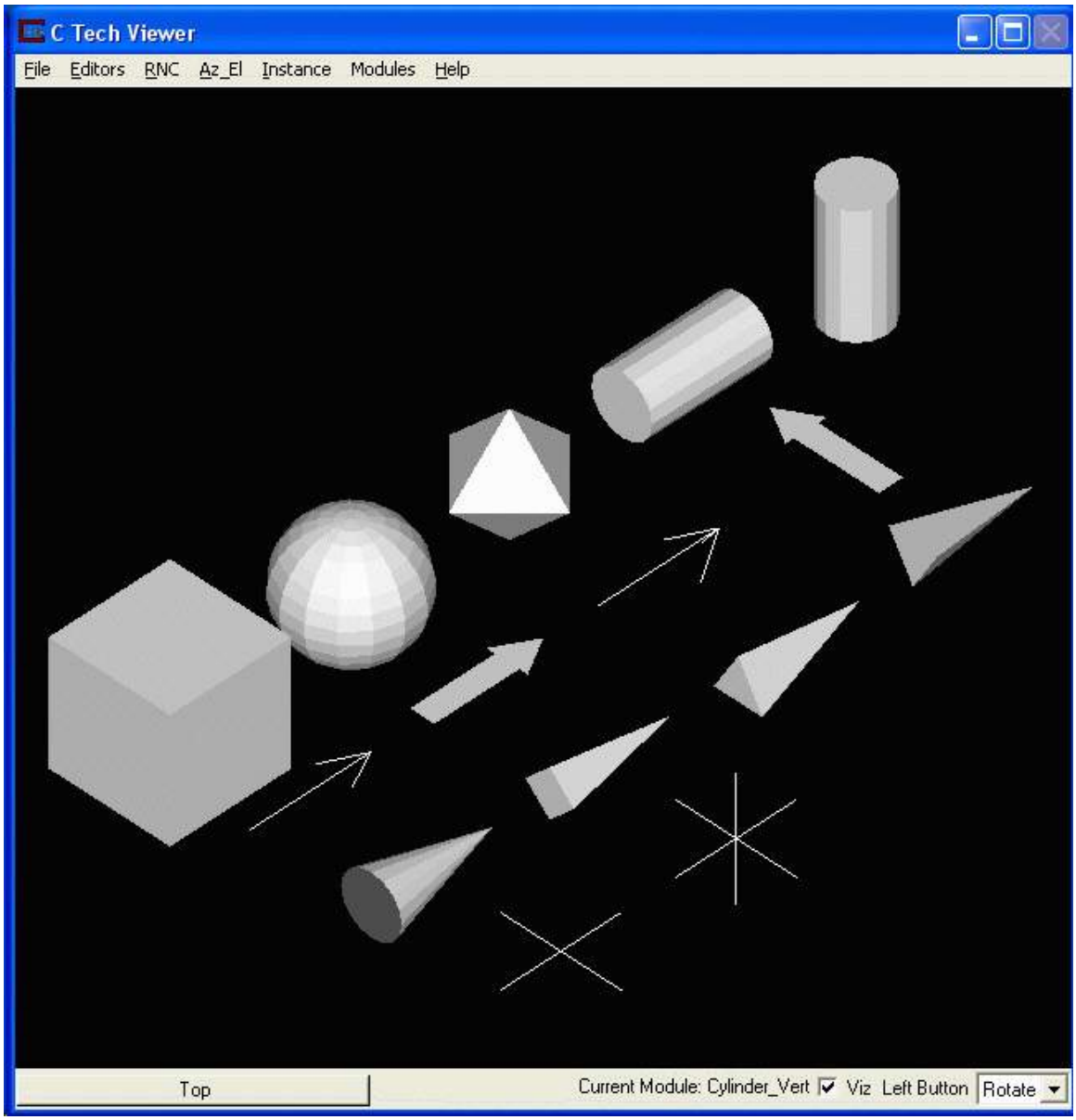
Module Control Panel

The control panel for georeference_image is shown in the figure above. There are identical controls for two independent sliders (nominally red and green). The *X (Pixel) Coord:* and *Y(Pixel) Coord:* sliders set the positions of

the crosshairs. The units here are in integer pixels. Color buttons allow the user to change the colors of the crosshairs if red or green is low contrast for your image.

Once the *X/Y Coordinate* values are typed in, the *Accept Current Values* button exports a georeferenced image from the output port.

Glyphs



Beginning in the top row (going left to right), the glyphs are:

box – sphere – diamond3D – cylinder – cylinder_vert

arrow1 – arrow2 – arrow3 – north_arrow

cone – pyrjet – jet – flipjet

cross2D – cross3D

Note: point3D, solid_cone and cylinder_vert_open are not shown. Solid_cone is identical to cone except it has a closed bottom. Cylinder_vert_open is identical to cylinder_vert except is doesn't have a closed bottom.

The panel for sphere is shown below. The only parameter is *Subdivision*, which determines the accuracy of the sphere by representing it as more polygons.

The panels for box, cone, solid_cone, cylinder, cylinder_vert and cylinder_vert_open are shown below.

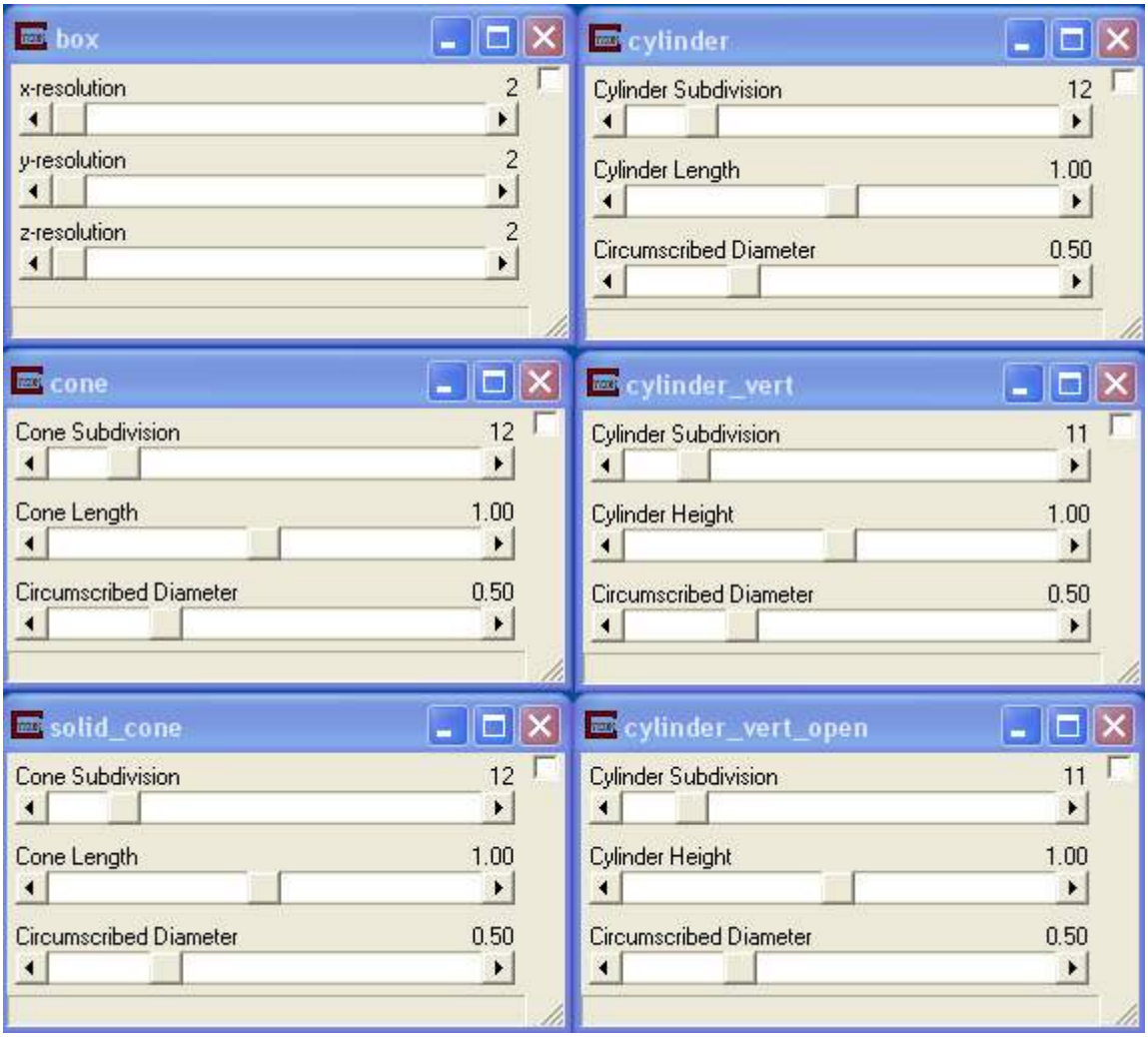
For Box, the *x, y, z resolution* is the number of nodes along each axis. When using as a simple glyph, keep it at two (2) to save memory.

For cone, solid_cone, cylinder, cylinder_vert and cylinder_vert_open:

The *Subdivision* parameters determine the number of sides.

The *Length or Height* parameters are the length in the x direction (or height in z for cylinder_vert)

The *Circumscribed Diameter* parameter determines the overall diameter.



Additional special glyphs are shown below. These are used to create specific geometric objects in your model space.

image_mask

General Module Function

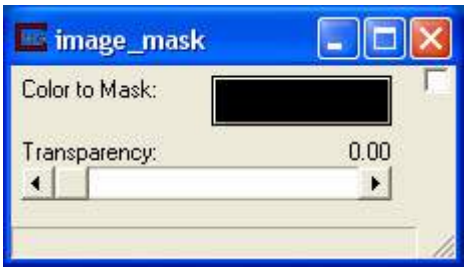
image_mask modifies an image so that a user specified color becomes transparent. This allows for texture mapping where parts of the surface are transparent or invisible.

Module Input Ports

image_mask has one input port which accepts the image from Read_Image.

Module Output Ports

image_mask has one output port. The output port outputs a new image that has an alpha channel corresponding to those pixels that match the selected color.



Module Control Panel

The control panel for image_mask is shown in the figure above.

The ***Color to Mask*** button sets the color that you wish to make transparent. This must be an exact color match. Usually white or black portions of an image are modified, but it can be any color.

The ***Transparency*** slider sets the degree of transparency for the masking color.

texture_map

This is a deprecated module that has been supplanted by overlay_aerial

General Module Function

Thetexture_mapmodule(available only in EVS-PRO and MVS),replaces the texture_mesh and texture_mesh_set modules which have now been deprecated as of version 6.5. Working in conjunction with the enhancements

to Read_Image or the new georeference_image module, it provides a much simpler, cleaner approach to texture mapping. It also supports transparency in the input images, allowing partially transparent textures.

texture_map is an enhancement to texture_mesh and texture_mesh_set in that it provides a means to project (texture map) images onto surfaces which are larger or smaller than the desired mapped extents. For example, the user may map a "postage stamp sized" aerial photo onto a very large topographic surface, or conversely, the user may project extents for an aerial photo that are outside the extents of the surface or object that the photo is projected onto.

It will handle ANY world file or GCP file (read automatically by Read_Image), including ones with arbitrary rotations and shears or higher order polynomial corrections (which can be created by the Georeference_Image utility).

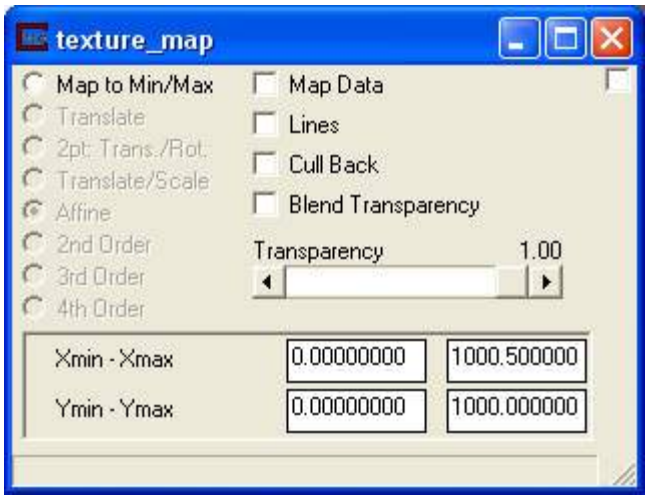
Additionally, this module can accept data from world files (like TIF or BMP World Files) that provide georeferencing information for an image. The world file data is passed automatically from Read_Image. This will set the extents for mapping the image for you to eliminate potential errors and speed processing.

Module Input Ports

texture_map has two input ports. The left input port should be surface data (such as 3DFACES) from the Read_DXF module or a geologic surface, and the right input port should be an image from the Read_Image module.

Module Output Ports

texture_map has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version directly to the Viewer.



Module Control Panel

The control panel for texture_map is shown in the figure above.

The ***Map Data*** toggle displays the underlying nodal data before texture mapping. If the input surfaces has nodal data, the texture map will be overlayed on colored surfaces. This is generally not desirable.

The ***Lines*** toggle determines whether lines are included in the output. The lines toggle will allow lines (such as a bounding box matching the image spatial extents) to be displayed.

Cull Back - A toggle for use with the transparency option. This feature may produce a better transparency when a surface is shown with other transparent surfaces, or if the same surface has high relief and is causing a confusing visualization. It's use will require experimentation to determine if it is appropriate.

BlendTransparency - A toggle to allow for transparent regions based on the alpha channel in your image.

Transparency - A slider for adjusting the opacity of the texture mapped image. A value of 1.00 is completely opaque and a value of 0.00 is completely invisible.

Modes: There are 8 different texture mapping modes as follows:

1) Map to Min/Max - Map image to the min/max extents of the input surface, or a user-defined value (can be typed into texture map directly).

2) Translate - Translate the image. Only requires a single GCP. No rotation or scaling is performed.

3) 2 pt: Trans./Rot. - Translate, Scale, and rotate the image. The image scaling is always the same in X&Y. Only a valid option if you have 2 GCPs. Good option if you only know 2 GCP points, and they are co-linear or near co-linear.

4) Translate/Scale - Translate and scale the image. Scale in X and Y are not the same. This keeps the image orthorectified. Can be used with 2 or more GCP points.

5) Affine - Perform a full affine transformation (1st order transformation) on the image. Requires a world file or 3 or more GCP points (from a gcp file). This is the default option which can be fully described with a World File.

6) 2nd Order - Perform a 2nd order polynomial transformation. This requires 6 or more GCP points (from a gcp file). It will map straight lines in the image into arcs. Allows an image that was georeferenced previously into LAT/LON coordinates to be "straightened" out and handled correctly. This can also be used to adjust for minor problems in the image due to topography. This option cannot be described with a World File because it uses a second order polynomial with more terms than are available in a world file. It requires the use of a GCP file.

7) 3rd Order - Perform a 3rd order polynomial transformation. Requires 10 or more GCP points. Allows you to adjust for drift in the image, "wedge" shaped photography, and more.

8) 4th Order - Perform a 4th order polynomial transformation. Requires 15 or more GCP points. Allows adjustments to be made

where different portions of the image move in opposite directions. Requires many GCP points to use effectively.

Xmin - Xmax - Type-ins for the minimum and maximum X extents of the texture mapped image. As noted above, this can be anywhere inside or outside the extents of the surface being mapped onto.

Ymin - Ymax - Type-ins for the minimum and maximum Y extents of the texture mapped image.

project_image

General Module Function

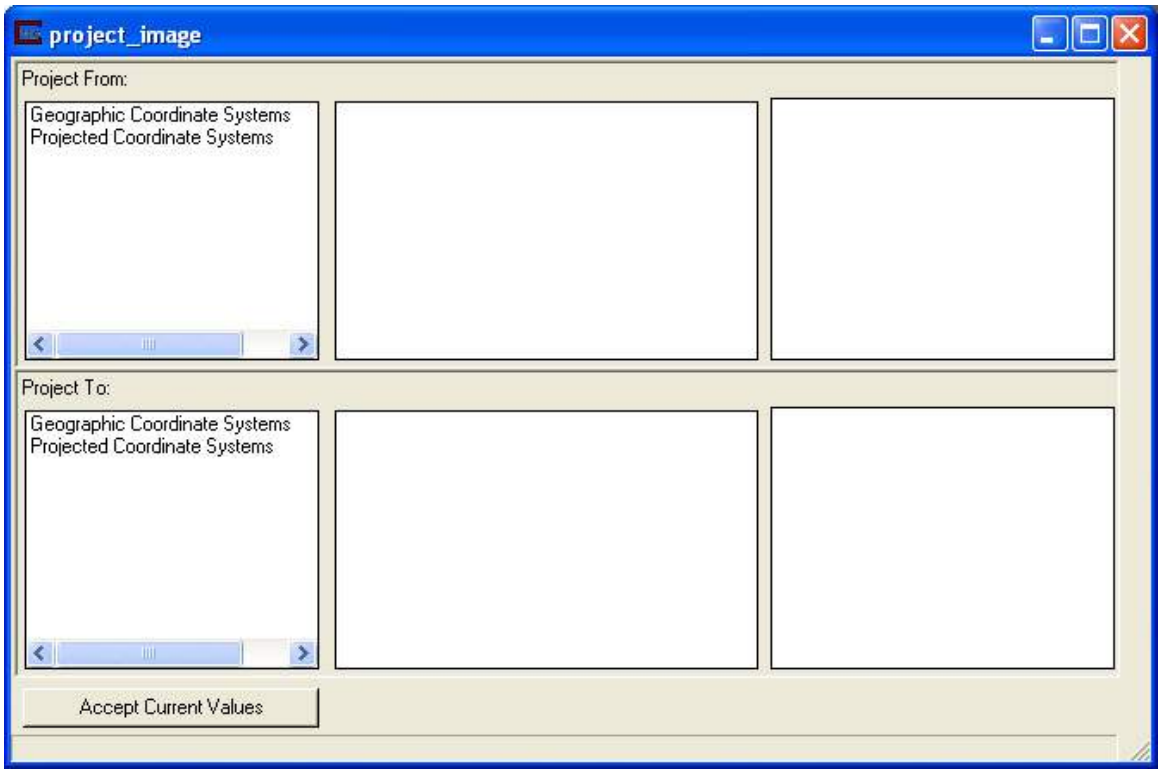
The project_image module is used to re-project the coordinate system for any image.

Module Input Ports

1) input_image (Blue/Green/Black): This port is used to import a previously created image to be projected.

Module Output Ports

1) projected_image (Blue/Green/Black): This port exports the projected image.

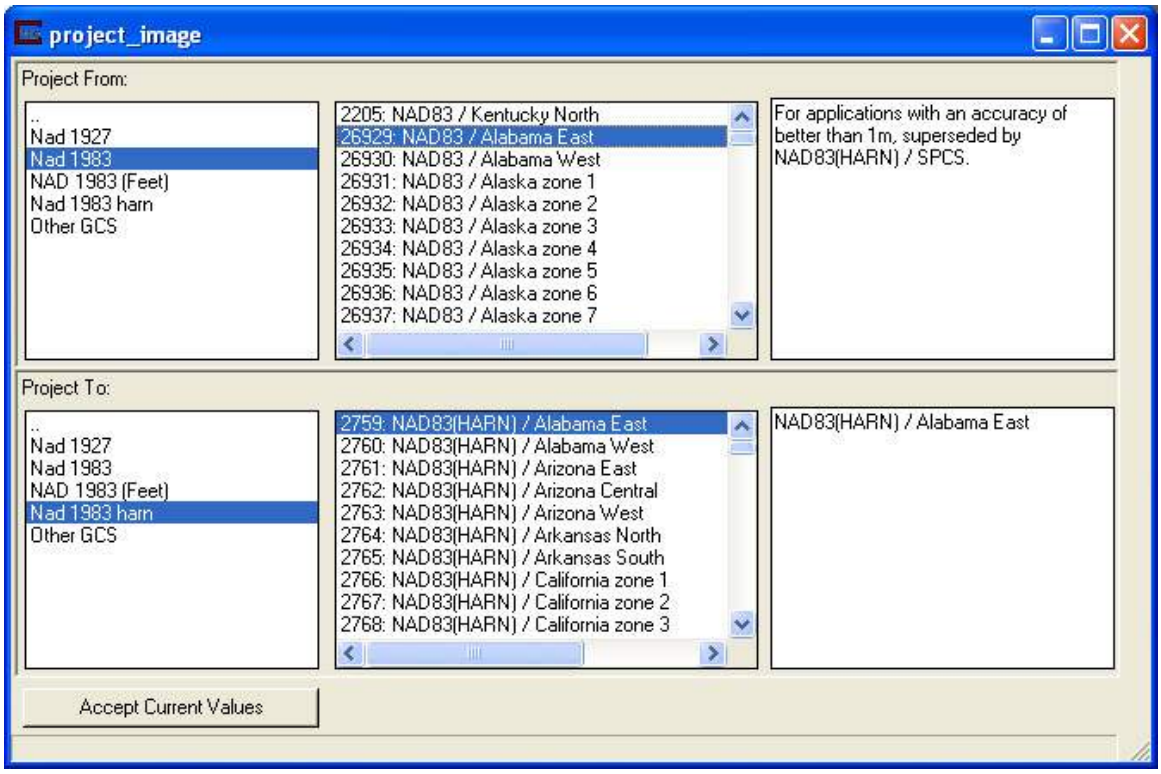


Module Control Panel

The control panel for project_image is shown in the figure above.

Each coordinate system is divided into either Geographic or Projected coordinate systems. The coordinate system types are navigated by selecting

the appropriate system type in the far left window. When a general coordinate system has been selected a specific coordinate system can be selected from the center window. If there are any details regarding the selected specific coordinate system, they will appear in the text window on the right. A specific coordinate system must be selected both to project from and to project to as in the picture below.



When both have been selected the Accept Current Values button must be pressed for the projection to occur.

tile_wall

This is a deprecated module that has been supplanted by texture_wall

General Module Function

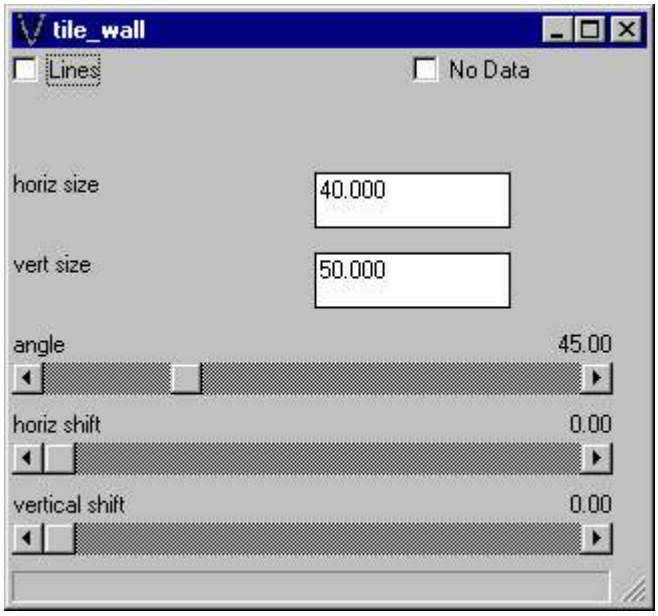
tile_wall provides a means to (texture map) project images onto vertical surfaces such as walls of buildings to apply texture maps such as brick or concrete to add more realism to you visualizations.

Module Input Ports

tile_wall has two input ports. The left input port should be surface data (such as 3DFACES) from the Read_DXF module, and the right input port should be an image from the Read_TGA_BMP module. You should try to segregate your DXF files into one with vertical walls for tile_wall and one with roofs for texture_mesh.

Module Output Ports

The tile_wall module has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version directly to the Viewer.



Module Control Panel

The control panel for tile_wall is shown in the figure to the right. The OpenGL renderer in EVS does not support this level of texture mapping. When tile_wall is connected to the Viewer, the OpenGL toggle will turn off the OpenGL renderer in the Viewer. The lines toggle will allow lines (such as a bounding box matching the image spatial extents) to allow lines to be displayed. If the input surfaces have nodal data, the texture map will be overlayed on colored surfaces. This is generally not desirable. The No Data toggle will remove the nodal data before texture mapping.

The horiz(ontal) size and vert(ical) type-ins specify the size of a rectangle in user units that the tile(d) image maps to. The angle slider determines the direction (counterclockwise from the "x" axis) from which the repeating pattern is projected. Since the pattern will be elongated in the horizontal direction, the vertical size should be larger (nominally by the 1.41) than the horizontal size to produce an undistorted texture. Larger sizes use less memory and run faster. Be careful not to use too small a size on a large set of walls.

It is not important that the image to be used for texture mapping in tile wall be properly registered with the surfaces upon which it is to be projected. With tile wall, we are applying repeating tiles of a small texture image. If the image and surfaces need to be shifted, both horizontal and vertical sliders are provided in the control panel.

A small collection of brick, rock and other textures is supplied as Targa files in the directory ctech\data\image\textures.

contour_texture

This is a deprecated module that has been supplanted by texture_geology

General Module Function

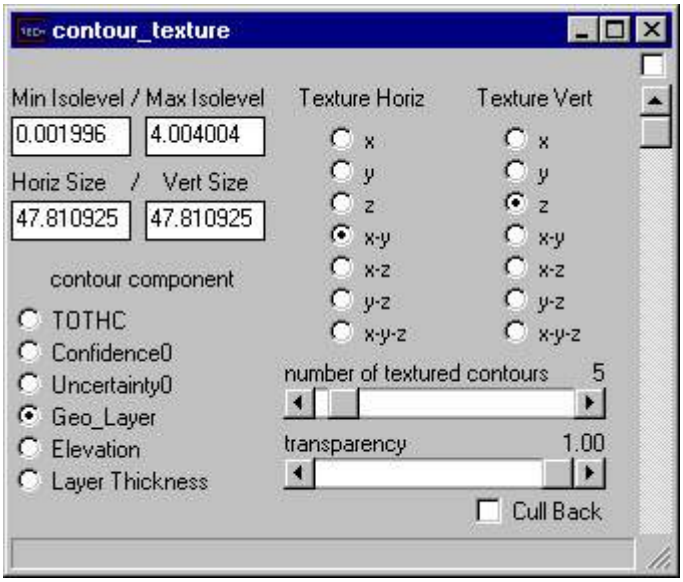
contour_texture provides a means to (texture map) project images such as geologic textures and hatch patterns onto 3D volumetric surfaces such as geologic layers and plume_shells. The module allows input of multiple bitmaps from Read_Images, and thus maps them onto a user-defined number of contours. These contours can a set of geologic layers or intervals of a plume distribution.

Module Input Ports

contour_texture has two input ports. The left input port 2D or 3D surface data (with nodal data), and the right input port should be images from the Read_Images module.

Module Output Ports

contour_texture has two output ports. The first output port (closest to the left) outputs an EVS field containing the texture mapped surfaces. The second port outputs a renderable version directly to the Viewer.



Module Control Panel

The control panel for contour_texture was substantially modified in Version 5.5 and is shown in the figure above.

Min & Max subsetting level - Type-in fields to control (bracket) the portion of the model you would like to texture map. These could be min/max layer numbers or concentrations. The default values are slightly lower and higher than the actual data min and max to avoid precision issues.

Horiz/Vert (Texture) Size - The size (in model units) that each bitmap (tile) will map to during the tiling process. For example, if your model is 1000 feet

wide and you use a value of 250, your bitmap texture will be repeated (tiled) four times across the model. Beware of small values. For example, for this site if you chose 25, the result would be 40^2 bitmaps across each surface, which could get extremely memory intensive.

Contour component provides radio buttons for selecting the desired data component for contouring.

Texture Horiz & Texture Vert are two radio boxes that are used to specify the direction of projection for the textures. There are 7 orientations that can be specified for both the horizontal and vertical textures axes. Horizontal and vertical do not correspond to your models directions, but rather to the bitmap image being mapped. You should not choose the same direction for both horizontal and vertical. Furthermore, some directions will not work well with others.

However the large number of possible projections directions (42) provide a great deal of control to obtain the look you desire.

Direction	Description
"x"	Perpendicular to the x axis
"y"	Perpendicular to the y axis
"z"	Perpendicular to the z axis
"x-y"	Project parallel to the x-y plane at 45 degrees to both axes
"x-z"	Project parallel to the x-z plane at 45 degrees to both axes
"y-z"	Project parallel to the y-z plane at 45 degrees to both axes
"x-y-z"	Project direction is the diagonal of a unit cube at 45 degrees to all axes

NOTE: The default projection direction is from the Southwest direction for horizontal and Perpendicular to the z axis for the vertical.

Number of textured contours - A slider which controls the desired number contour intervals (in the case of a plume or ore concentration volume), or according to the number material types in your geologic model. This number does NOT need to correspond to the number of layers, but should probably correspond to the number of images being input into contour_texture. The default value will be set according to how many images are input (usually Read_Images module).

Transparency controls the transparency of the output of contour_texture.

Cull Back toggle turns on culling of surfaces whose normal vector points away. Generally useful when doing transparency of complex CLOSED objects.

C Tech has provided several bit maps for use with this module, but users are encouraged to create their own textures, which are customized to their drafting and presentation standards. The dimensions of the provided images are generally quite small (e.g. 79 x 78 pixels).

Read_Images

This is a deprecated module that has been supplanted by texture_geology replacing contour_texture

General Module Function

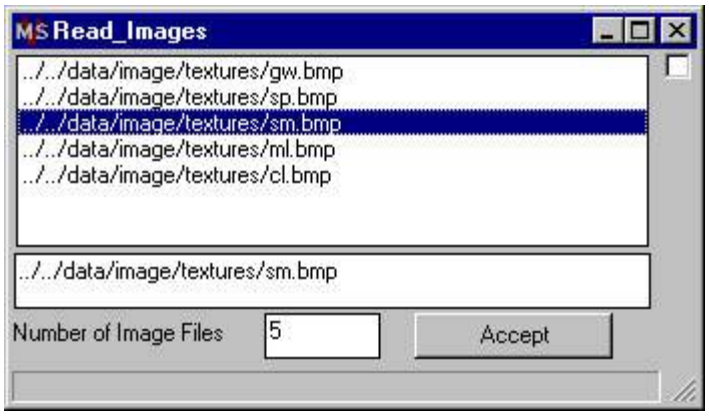
Read_Images is an enhancement to Read_Image designed to support reading of multiple bitmaps at once The output from Read_Images can then be passed to contour_texture for mapping different textures or hatch patterns onto multiple geologic units or concentration contours. Read_Images reads multiple image files of a specified type and converts them into 2D uniform fields.

Module Input Ports

Read Images has no input ports. It obtains the input data by reading a file with a file browser. The filename can be either the complete absolute pathname, or a relative pathname. If it is relative, it will be interpreted relative to the user's working directory when EVS was started.

Module Output

Read_Images has one output port. This port passes a uniform mesh and its data components to other modules, which accept uniform field data types. Data passed from Read_Images is usually sent to contour_texture. The x and y nodal dimensions of the field will equal the pixel dimensions of the image and the values assigned each node are obtained from the RGB value of each pixel.



Module Control Panel

The user interface (shown above) shows 5 filenames and their paths. The default number of images is 5. C Tech has populated these fields with sample images as a starter. The user may choose other images, and may also specify more images or less images by changing the value in the type-in box called Number of Image Files. Note that default text is placed once a number larger than 5 is chosen. Simply highlight the path you want to change, then edit the path in the lower box.

File types: The following image formats are currently supported:

* AVS .x

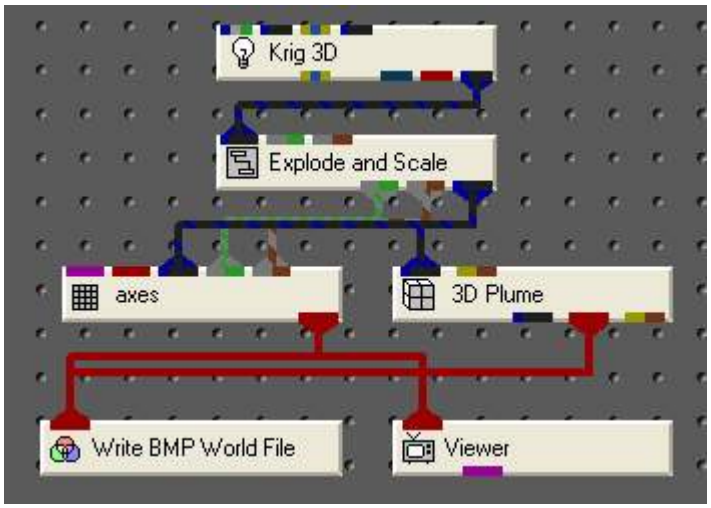
* JPEG
* PBM
* SGI Image
* Sun Raster
* GIF
* TIFF (uncompressed, No LZW compression)

Write_BMP_World_File

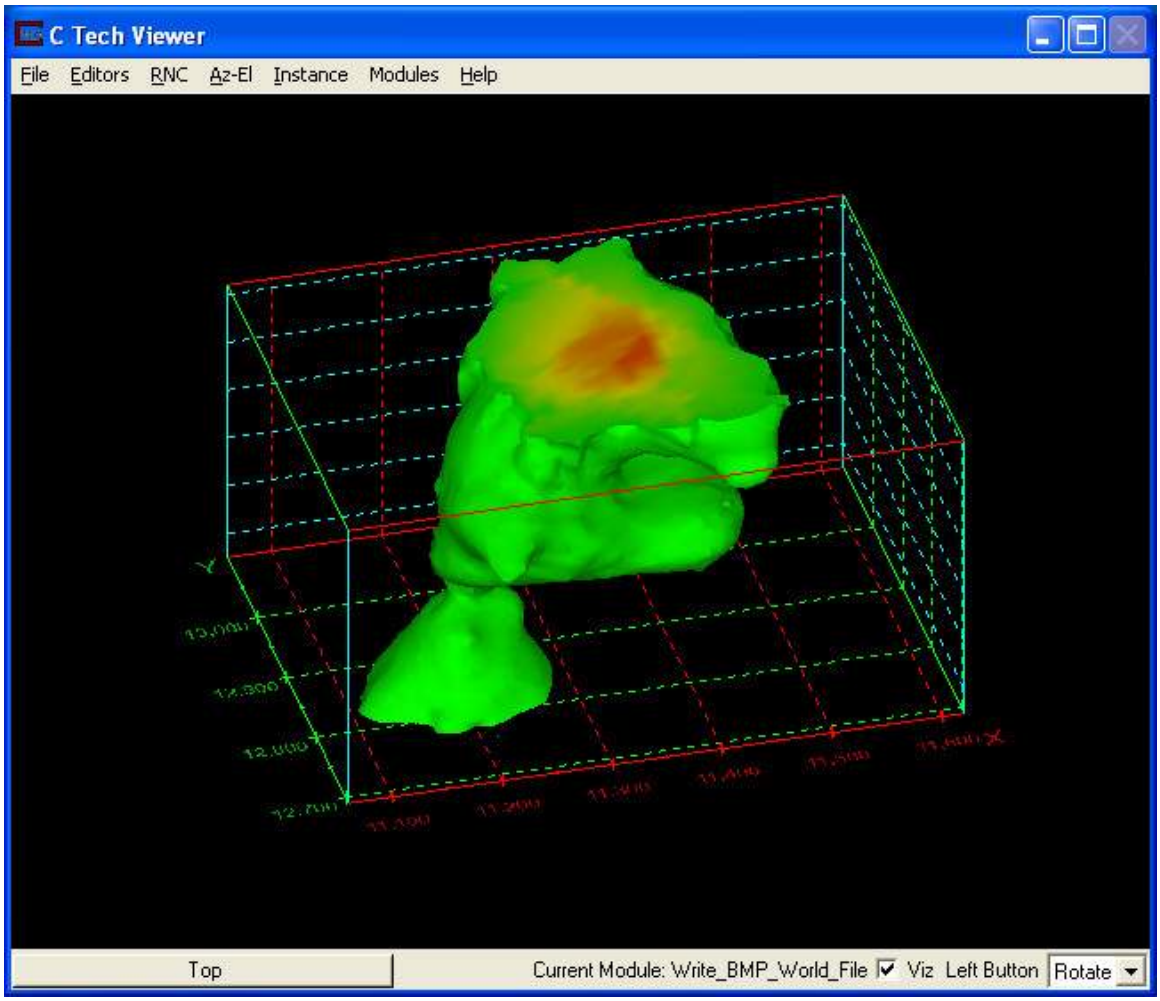
This is a deprecated module that has been supplanted by georeferenced_output

Write_BMP_World_File: This module writes a georeferenced BMP file (and associated .BPW file). When objects are connected to this module (like a Viewer) it renders a TOP view (cannot be zoomed, rotated or panned). This view will be written as a georeferenced image that can be imported by ArcView and other GIS applications.

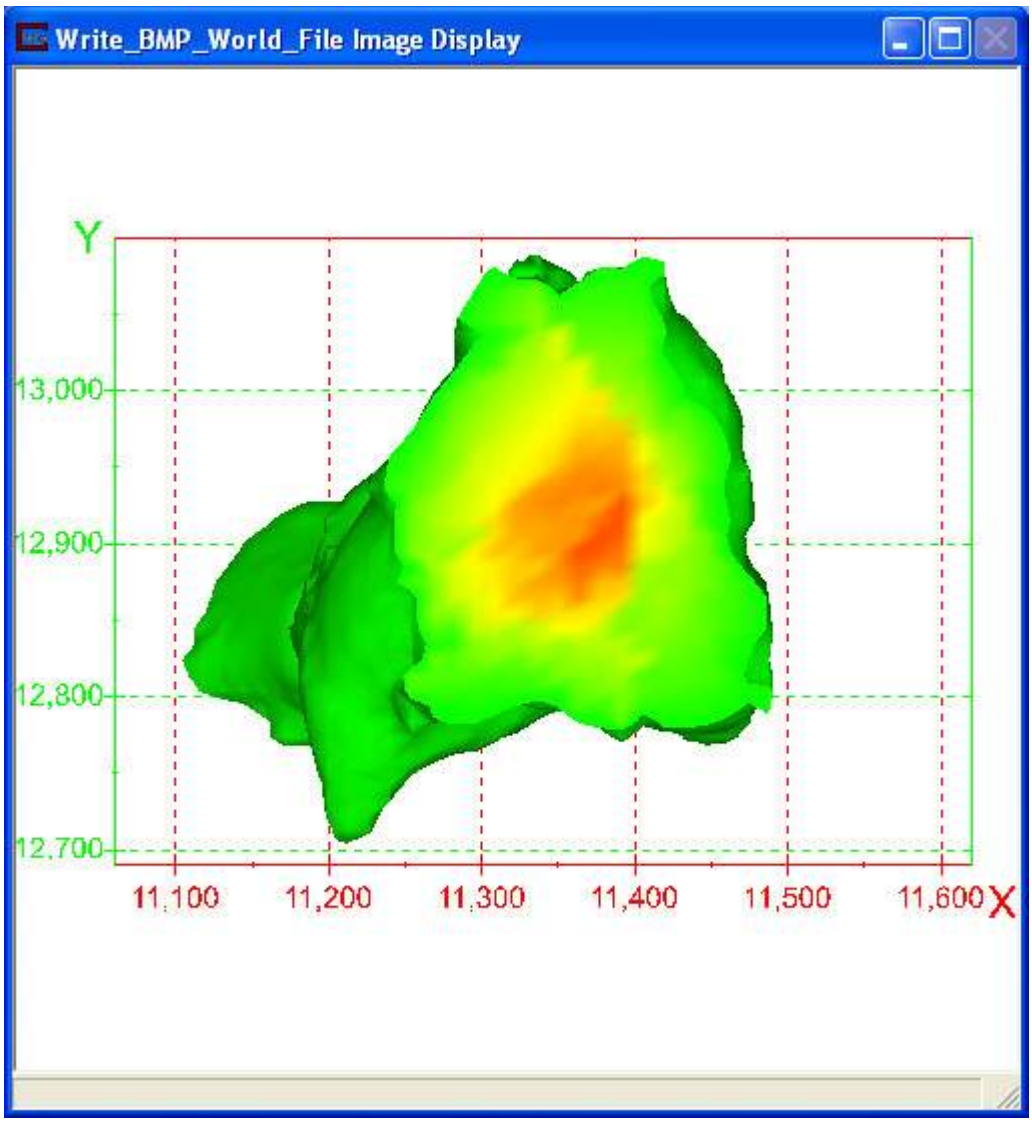
Set the Resolution to 500 x 500, and check the White background toggle.



A view from the Viewer



A view from the Write_BMP_World_File's view panel



surfer_to_geology

General Module Function

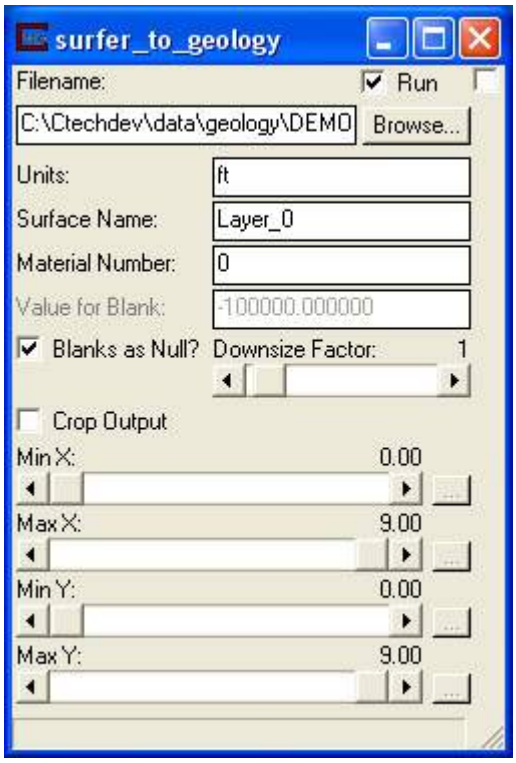
surfer_to_geology reads Surfer GRID files in EVS Geology format. This allows multiple surfer grid files to be combined with combine_comp into a 3D geologic model. Alternatively, a single surfer grid file can be displayed as a surface (with geologic_surfaces) or you can export its coordinates (with Write_Coordinates) to use the values in a GMF file.

Module Input Ports

surfer_to_geology has no input ports since it builds its output from a Surfer GRID file.

Module Output Ports

surfer_to_geology has one output port (blue-white-green) which is identical to the port on Krig_3D_Geology or Spline_Geology.



Module Control Panel

The control panel for surfer_to_geology is shown in the figure above.

The ***Units***, ***Surface Name***, and ***Material Number*** type-ins allow you to set the values for the surface you are reading. This provides equivalent information as to what is specified in a .Geo or .GMF file and passed through by Krig_3D_Geololgy.

Additional controls for cropping and downsizing the grid during reading are provided to dramatically reduce memory usage and increase speed of multi-layer 3D models.

The ***Value for Blank*** type in allows you to specify what the elevation should be for any blank cells in the surfer grid file. An EVS Geology surface cannot have blanks, however you can later simulate this effect using threshold to remove whole cells above or below a threshold level.

Text3D

This is a deprecated module whose function has been superceded by the place_text module.

General Module Function

Text3D defines 2D and 3D renderable text strings that have a user interface to control many of the text attributes.

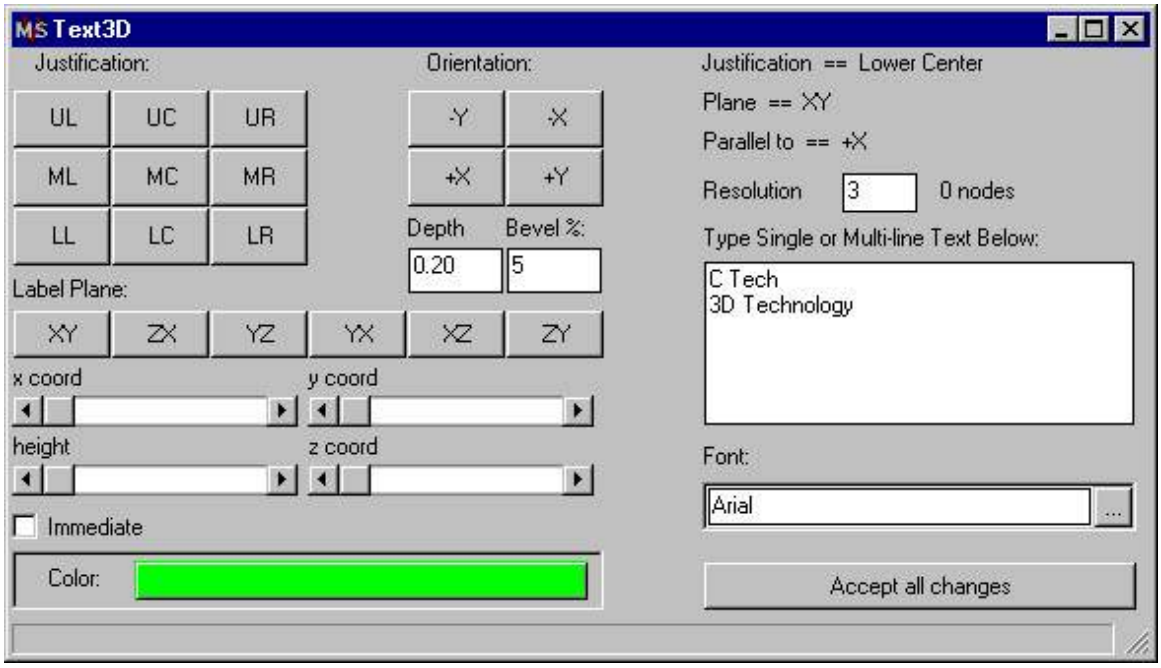
Module Input Ports

Text3D has two input ports. The first input port accepts a blue/black field input to provide a location from which the text object will reference. The

second input port is the string to be created. This can be input from a string module.

Module Output Ports

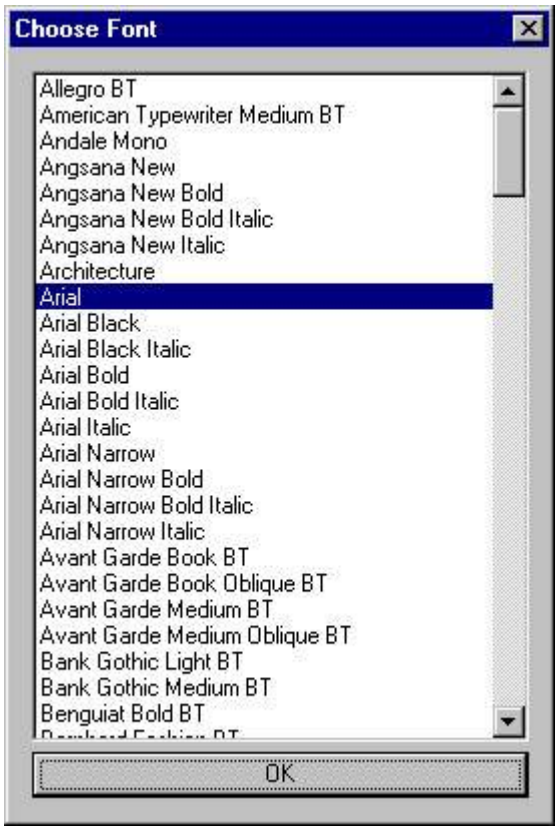
Text3D has two output ports. The blue_black field port can be used to texture map (or subset) the text and the red port which sends the renderable text string to the EVS viewer.



Module Control Panel

Type Single or Multi-line Text Below Contains the text-string to be rendered. The text may have carriage returns, thus allowing for multiple lines in the title-box.

Font Contains the font that the string is rendered with. The button on the right brings up the font selector window shown below. For more information on Font Selection Click Here.



Justification is identical to the controls in the Map_Spheres Labeling.

Orientation is identical to the controls in the Map_Spheres Labeling.

Label Plane Used in conjunction with the orient and angle subobjects, specifies the transformation for the character string: the choices are XY (analogous to horizontal/ground surface; etc. These are identical to the controls in the Map_Spheres Labeling.

X_coord, y_coord and z_coord are sliders that allow you to specify the 3D location of the string within extended bounds of your input field.

Immediate toggle causes the letters to move as you move the sliders.

The color button brings up a color selector window that allows you to specify any color for the text.

Depth: causes the text to be extruded as a truly 3D object. If set to zero (0.00) the text is flat.

Bevel: causes the 3D text to have beveled edges. Acceptable values are font dependent. If the value is too large the faces of some of the letters will appear to have holes and connectivity problems.

Resolution: is the smoothness of curves used to form the letters. Higher resolution requires more creation time, more memory and slower rendering, however the letters look better.

Orient Used in conjunction with the plane and angle subobjects, specifies the transformation for the character string: choices are left to right; right to left; down; and arbitrary (which requires using the angle slider).

The **Height** slider specifies the height of the character; the default character height is set based on the extents of the input field. It is used as a scale factor for both the height and width of the character.

The angle and path are adjusted by changing the orientation and plane. Spacing is determined by the font you choose. Each font's specific spacing is taken into account (this is actually part of the font itself).

Related Modules

-> Titles

MultiText3D

This is a deprecated module whose function has been superceded by the place_text module.

General Module Function

MultiText3D defines 2D and 3D renderable text strings that are specified and positioned using a new input file format with a .emt file extension. An example file is shown below. In .EMT files, there are the following command and/or data lines:

Lines beginning with "#" are comments

Lines beginning with "FONT" are font specification lines (more later)

Lines beginning with "END" specify the end of the file (this is optional, but if you want to have anything after the last command or data line, precede it with an "END" statement.

All other lines are DATA lines specifying the x-y-z coordinates of a string and the text for that string.

Blank lines are ignored.

The FONT specification lines contain the following information:

Size: The font size is the height of a typical Capitol letter in true user units

Justification: The justification options are the same as in Text_3D

Plane: The plane options are the same as in Text_3D

Orientation: The orientation options are the same as in Text_3D

Red, Green, Blue: These 3 numbers determine the font color.

Resolution: The resolution parameter is the same as in Text_3D

Depth: The parameter is the same as in Text_3D

Bevel%: The Bevel percentage is the same as in Text_3D

Font Face: The Font Face options are the same as in Text_3D

The DATA lines contain four columns of information:

X coordinate

Y coordinate

Z coordinate

Text: Everything on the line after the z coordinate (and trailing spaces) is the text to be placed at the above coordinate.

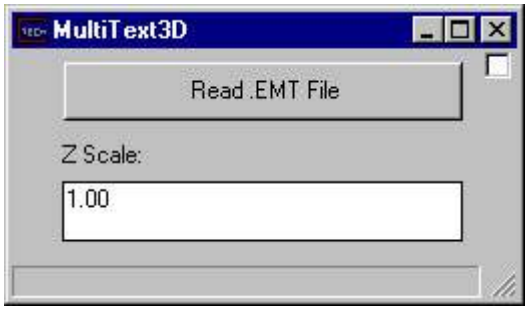
```
# FONT Size Just. Plane Orient R G B Resolution Depth Bevel% Font Face
FONT, 4, MC, XZ, +X, 0.8, 0.8, 0.8, 3, 0, 0, Arial
# X, Y, Z, Bore
11566.34, 12850.59, 8.5, B-30
11586.34, 13050.59, 12.5, B-31
11381.7, 12747.5, 2.5, B-33
11414.4, 12781.1, 3, B-34
11410.29, 12724.69, 4.5, B-4
11427, 12780.9, 7.5, B-42
11086.52, 12830.67, 5.5, B-49
11211.87, 12710.75, 3, B-50
11199.04, 12810.16, 5, B-51
11496.34, 12753.59, 2.5, B-53
11209.35, 12993.94, 3, B-57
11301.97, 13079.66, 5.5, B-58
11286.77, 13026.7, 3, B-59

# FONT Size Just. Plane Orient R G B Resolution Depth Bevel% Font Face
FONT, 6, MC, XZ, +X, 1, 0.5, 0.5, 3, 0.1, 0, Arial
11393.47, 12948.9, 4.5, B-60
11309.03, 12948.99, 5, B-56
11248.75, 12870.91, 4, B-48
11259.67, 12819.29, 3, B-46
11298, 12808.63, 4, B-52
11338, 12830.8, 5, B-38
11401.73, 12897.77, 5, B-45
11416.9, 12819.45, 3.5, B-44

# FONT, Size, Justification, Plane, Orientation, Red, Green, Blue, Resolution, Depth, Bevel%, Font Face
FONT, 8, MC, XZ, +X, 1, 0, 0, 3, .3, 0, Arial Bold
11340.49, 12892.61, 3.5, B-47
11251.3, 12929.27, 3, B-75
END
```

Module Input Ports

MultiText3D has one input port which accepts a scaling factor that specifies the multiplication factor (Z-Exaggeration) for the z-coordinates of the text strings.

Module Output Ports

MultiText3D has two output ports. The first is a pass-through for the Z-Exaggeration and the second is a red port which sends the renderable text strings to the EVS viewer.



Module Control Panel

The control panel has a browser button to specify the .emt file and the Z Scale type-in.

Related Modules

-> Text_3D

draw_2D_lines

This is a deprecated module that has been supplanted by draw_lines

General Module Function

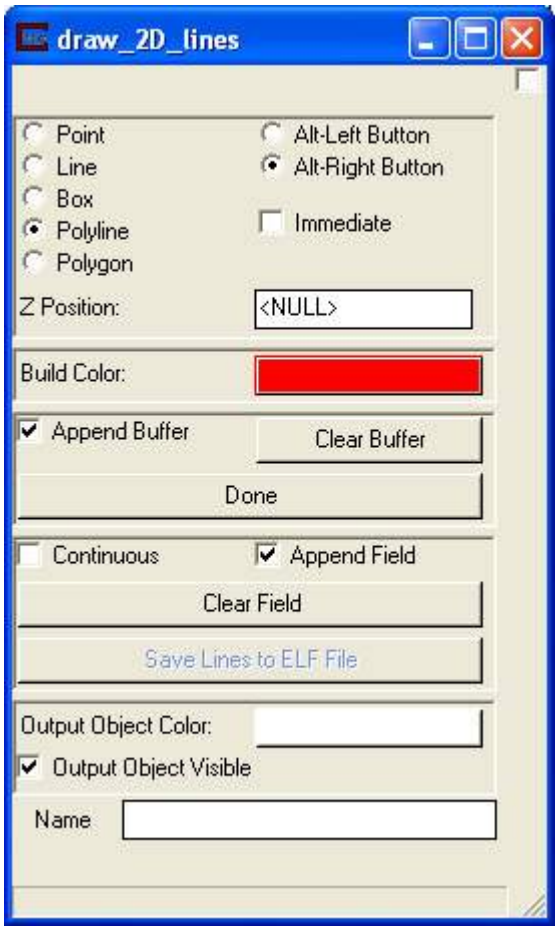
The draw_2D_lines module enables you to create a 2D drawing with individual clicks of the mouse. The default mouse gesture for line creation is: depress the alt key and then click the right mouse button. The first click establishes the beginning point of the line segment and the second click establishes the end point. draw_2D_lines allows adding of points, erasing of region points, and closing a region It is required that drawing actions be made in a normalized view(Top View) to ensure the line is drawn at a constant z-value. This module is commonly used to create 2D lines passed to the fence_cut module. Unlike most modules which create mesh data to used by other modules, the draw_2D_lines module receives input from the viewer, and also passes on field data to be used by other modules. NOTE: Currently in EVS the line may not display in the viewer window unless software renderer is chosen in the viewer instead of OpenGL.

Module Input Ports

The draw_2D_lines module has one input port. This port is connected to the output port of the viewer within which the line will be drawn.

Module Output Ports

The draw_2D_lines module has two output ports. This first output port sends a mesh to downstream modules. The second output port sends the line as a renderable object to the viewer to be rendered.



Module Control Panel

The draw_2D_lines user interface allows interactive creation of points or lines onto any connected object and then displays the points or lines in the viewer. One popular use for this functionality is for creation of line segments to create fence sections using the fence_cut module.

The first set of radio buttons allows picking which type of primitive to draw. Possible primitives are point, line. box, polyline, and polygon. The default is polyline because only polylines are supported with the fence_cut functionality (box and polygon uses polylines). To the right of those you can select the keyboard and mouse button sequence that is used to draw.

The immediate toggle controls whether the output buffer is immediately written to (data passed to other modules) upon completion of a line segment. When the toggle is off, you must press the Done button to cause the output buffer to be written.

The Z Position type in controls at what z position the drawing is placed at. By default this value is the midpoint of the Z extents of the objects connected to the Viewer.

The build color button controls the color of the drawn line. Remember the line may not display unless software renderer is selected in the viewer instead of OpenGL.

The append buffer toggle determines whether each time a line segment is completed it overwrites any previous primitive. If append buffer toggle is on, each time a primitive is built, it is appended to a previous line segment(s). By using append mode, you can build multiple primitives before mapping them to a field.

The Clear Buffer button causes the build buffer to be cleared of all points or line segments.

The Done button causes all existing line segments to be passed to other modules as a field.

The Continuous toggle allows the user to draw continuously while the Alt key and right mouse button are held down.

The Append Field toggle is on by default. By using append mode, you can incrementally build the field (for example fence sections). If the Append Field toggle is off, each time points or lines are mapped, any previous field is overwritten.

The Clear Field button erases any field previously created.

The Save Lines to ELF File allows the user to save all of the created lines in the EVS Line File format (*.elf).

The Output Object Color button controls the color of the entire set of finished lines.

The Output Object Visible toggle will turn the visibility of the lines on or off.

The Name box allows creating a name for the cell sets passed upon pressing the Done button.

set_minmax

This is a deprecated module that has been supplanted by change_minmax

General Module Function

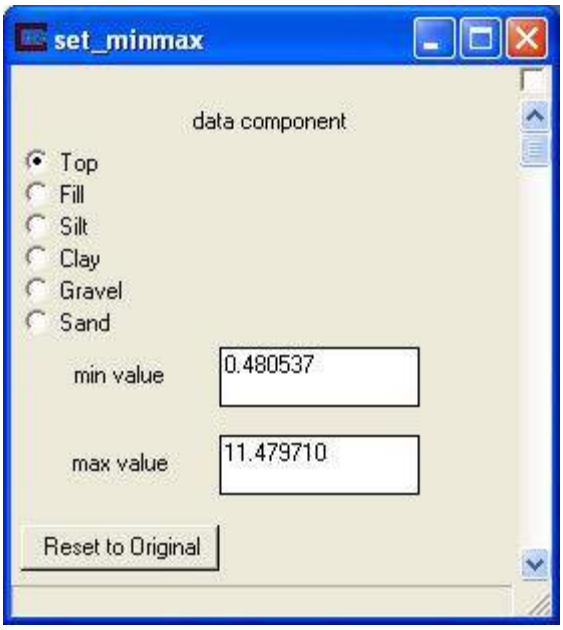
The set_minmax module extends the capabilities of the clamp module by allowing setting of max values above the true maximum data range and min values below the true minimum data range. This functionality is commonly needed for color mapping of time-series data. For example, the user can set the minmax values to bracket the widest range achieved for many datasets thus allowing consistent mapping from dataset to dataset during a time-series animation. This way 100 ppm would always be red throughout the animation, and if one dataset did not reach a maximum of 100 ppm, there would be no red color mapping for those time-steps.

Module Input Ports

Set_minmax has only one input port. Input piped to this port must contain nodal data. If mesh data is piped to this port it will pass through set_minmax unchanged. The nodal data components can be scalar or vector.

Module Output Ports

Set_minmax has two output ports. The first output port (closest to the left) outputs a new nodal data set containing only the selected data component subjected to the set_minmax criteria. If a mesh is passed to the set_minmax input it will pass to the output port unchanged. The second port outputs a renderable geometry if a mesh is present.



Module Control Panel

The control panel for set_minmax is shown in the figure above. The Min Value edit field determines the minimum set value and the Max Value edit field determines the maximum set value. The data range passed from this module will match the values in these type-in boxes. Note the min and max values may be higher (and/or lower) than the current data range.

Related Modules

->clamp

set_minmax_cell

This is a deprecated module that has been supplanted by change_minmax

General Module Function

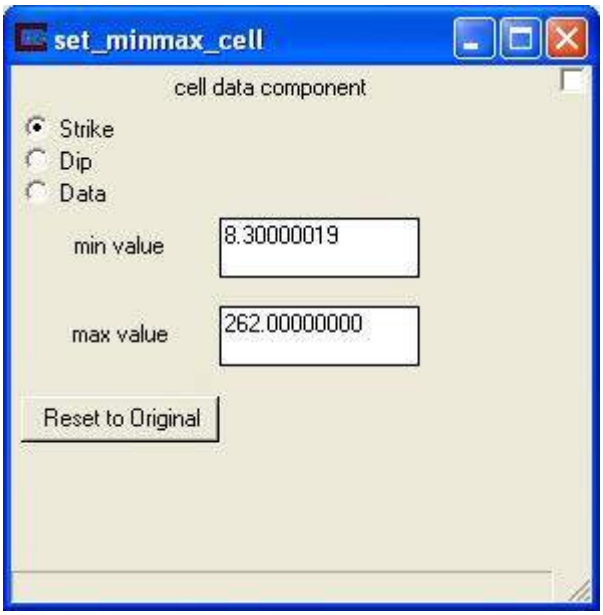
set_minmax_cell provides identical functionality to set_minmax, but requires cell data. Nearly all modules in EVS currently support only node based data.

Module Input Ports

Set_minmax_cell has only one input port. Input piped to this port must contain cell data. If mesh data is piped to this port it will pass through set_minmax unchanged. The cell data components can be scalar or vector.

Module Output Ports

Set_minmax_cell has two output ports. The first output port (closest to the left) outputs a new cell data set containing only the selected data component subjected to the set_minmax criteria. If a mesh is passed to the set_minmax_cell input it will pass to the output port unchanged. The second port outputs a renderable geometry if a mesh is present.



Module Control Panel

The control panel for set_minmax_cell is shown in the figure above. The Min Value edit field determines the minimum set value and the Max Value edit field determines the maximum set value. The data range passed from this module will match the values in these type-in boxes. Note the min and max values may be higher (and/or lower) than the current data range.

Related Modules

set_minmax

surf_plot

This module has been deprecated. Its functionality has been superceded by the field_math data_translate modules.

General Module Function

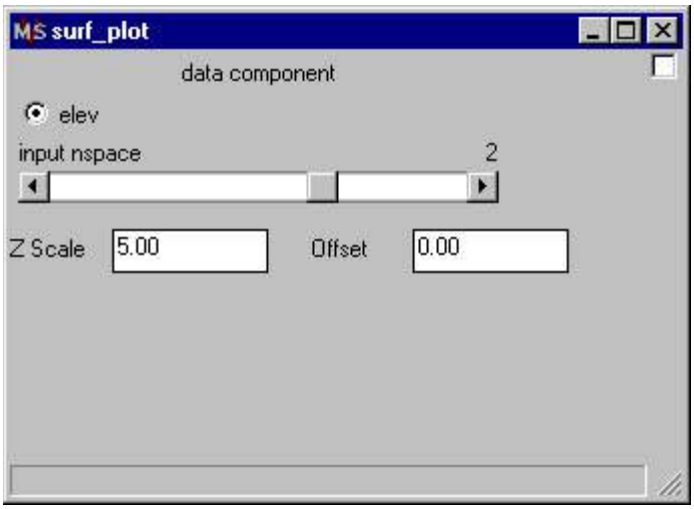
surf_plot creates a 3D or 2D mesh whose height at each point is proportional to the scalar data value at that node location in the original field. 2D input produces a 3D mesh. 1D input produces a 2D output.

Module Input Ports

surf_plot has two input ports. The first (blue/black) may contain any mesh type and any Node_Data. The field's mesh must have nspace=1 or 2. However, no checking is done to enforce this restriction. (Inputting an nspace=3 mesh will not generate a picture.) The Node_Data must contain at least one scalar component. The second input port is the gray/brown Z exaggeration port.

Module Output Ports

surf_plot has three output ports. The first output port (closest to the left) outputs a mesh of the same type as the input mesh, with nspace+1, whose coordinate values represent the projected, scaled, and offset surface. Its Node_Data contains the data in the selected component.. The second output port sends a renderable version of the output field. The rightmost port is the gray/brown Z exaggeration port.



Module Control Panel

The control panel for surf_plot is shown in the figure above. Each parameter is described below:

data component: Radio buttons to pick which of the input's data components to use to set the mesh height. The default is the first (0th) component. You must select a scalar component.

input nspace: A slider to indicate the dimensionality (nspace) of the input field. Remember the result will be the same mesh with one level of dimensionality added (the z dimension or height). This module generally converts images (2D fields) into 3D surfaces.

Z Scale: A float type_in to scale the height (scale*component values). The default is 5.0.

Offset: A float slider that raises or lowers the resulting mesh with respect to the input mesh. The default is 0.0.

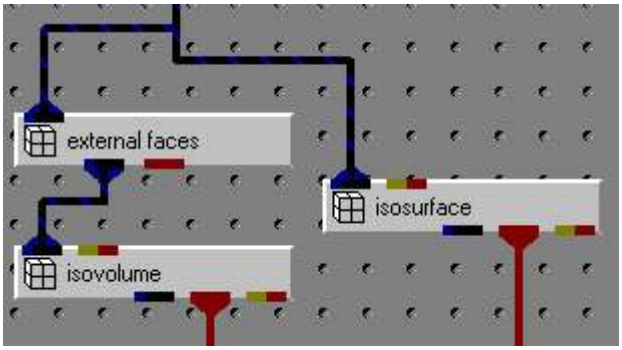
_3D_Plume

This is a deprecated module whose function has been superceded by plume_shell

General Module Function

The 3D_Plume module (previously named IsoVolume) is identical to a subnetwork containing the isovolume module with constant_shelland external_faces. This module creates a superior visualization of a plume that can be sent directly to the viewer for rendering. This is not a subsetting module (as is isovolume or contour). It is used exclusively for plume visualization of 3D fields (such as the output of Krig_3D).

The figure below shows a module network showing the three individual modules that make up the 3D_Plume module.

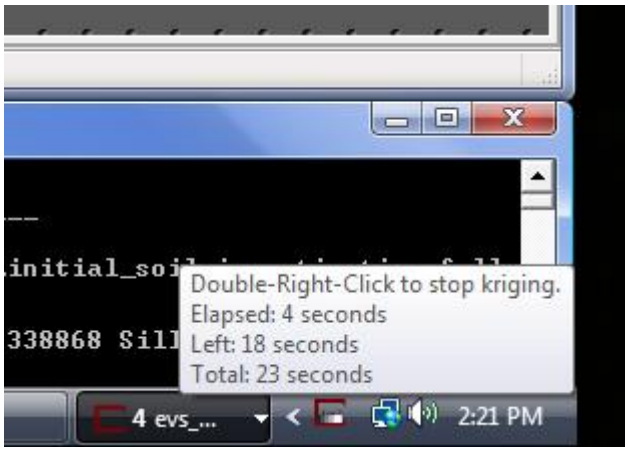


Module Input Ports

3D_Plume has two input ports. The leftmost port accepts unstructured mesh data. The second port provides a means to share the subsetting level of other modules.

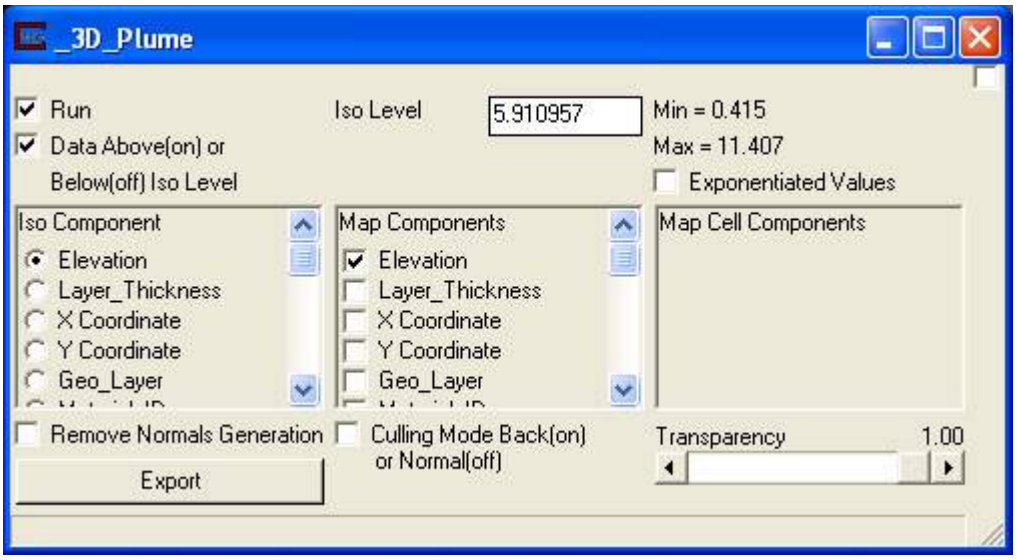
Module Status: Interruptible

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Output Ports

3D_Plume has three output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains cells representing the external faces portion of the 3D_Plume. Nodal data sent to this output port consists of the nodal data of the map component in the isovolume portion of the subnetwork. This blue output is used to input to isolines to place isolines on the portions of the plume that intersected the exterior surfaces of the input domain. The second port outputs a rendered geometry directly to the Viewer. The third output port provides a means to share the subsetting level of this module with others.



Module Control Panel

The control panel for 3D_Plume is shown in the figure above.

Iso Components refers to the model data component used to create the subset of the original input field. When an iso component is selected, the min and max values of the variable are displayed in the lower right corner of the dialog box. The default iso component is the first (0th) component in the column.

Map Components determines which model data components in an unstructured mesh will be sent to the leftmost output port. Also, the first map component selected will be used to color the output. For example, choosing iso component concentration and map component uncertainty will create a volumetric subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The ***Map Cell Components*** option box selector lets you map cell data (if any) to the surfaces output by 3D_Plume. This is markedly faster than using interp_cell_data. The cell data will only be visible if all nodal data is unselected (under *Map Components*).

The ***Iso Level*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum

value is placed in the box. The default iso level value is the arithmetic average of the minimum and maximum values in the iso component. If your input data has been kriged with log processing, the values here will be the Log of your input data.

The ***Exponentiated Values*** toggle makes another type-in field visible which convert real units to logarithmic units for you automatically.

The ***Run*** toggle when not selected will prevent the module from running.

The ***Data Above(on) or Below(off) Iso Level*** check box (Above Box) is used to display data above the iso level or below the iso level. For example, to display a volumetric subset of all concentrations greater than or equal to 1 ppm, set iso level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the 3D_Plume of 1 ppm and below, simply turn the Above Box off (no check in box).

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Culling Mode Back(on) or Normal(off)*** toggle is equivalent to setting the object surface property to cull back facing surfaces. This is recommended whenever Opacity is less than 1.00

The ***Opacity*** slider changes the opacity (opposite of transparency) of the entire 3D Plume Group.

The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. The additional ports represent:

a. Iso_Component name: This is a string with the name ot the selected data component.
b. Above/Below: This is a string containing the word "Above" or "Below" depending on the state of the toggle above.
c. Iso_Level: This is a real number representing the subsetting level.
d. Exponentiated Iso_Level: This is a real number representing the exponentiated subsetting level.

Related Modules

->isosurface

-> isolines

-> isovolume

isovolume

This is a deprecated module whose function has been superceded by plume_volume and plume_area

General Module Function

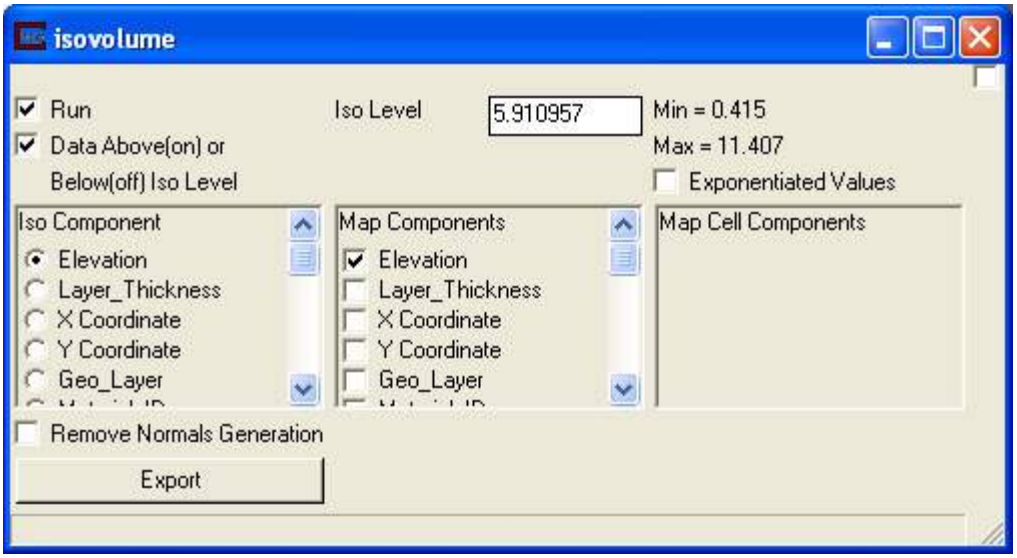
The isovolume module is a subsetting module, which produces a subset (volume, area or lines) of the data below or above an subsetting level (threshold). isovolume cuts the data at a specified subsetting level, and outputs those portions (volume, area or length) of the input mesh that is below or above (default) the specified subsetting level. It can be used with 1D, 2D or 3D input data. Multiple isovolume (subsetting) modules can be used in serial connection to subset an input field by more than one criteria. The isovolume of one component can be colored by the scalar value of another component (for example, the isovolume of concentration can be colored by uncertainty).

Module Input Ports

isovolume has two input ports. The leftmost port accepts unstructured mesh data. The second port provides a means to share the subsetting level of other modules.

Module Output Ports

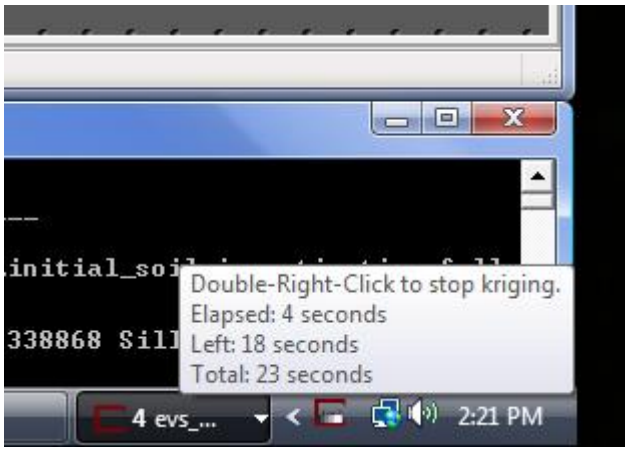
isovolume has three output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains cells representing the external faces of the isovolume. Nodal data sent to this output port consists of the nodal data of the map component in the isovolume. The second port outputs a rendered geometry directly to the Viewer. The third output port provides a means to share the subsetting level of this module with others.



Module Status: Interruptible

This module's computational processes can be terminated (interrupted) using the "C Tech" icon in the Windows Notification Area (aka System Tray) in the lower right corner of your desktop. If you hover over the icon, it will tell you the status of the module and expected completion time. Double-Right-

Clicking will terminate the process. Note that if you do stop any process, the output of the module is corrupted and any downstream module's results are not usable. You will need to re-run the module.



Module Control Panel

The control panel for isovolume is shown in the figure above.

Iso Components refers to the model data component used to create the subset of the original input field. When an iso component is selected, the min and max values of the variable are displayed in the lower right corner of the dialog box. The default iso component is the first (0th) component in the column.

Map Components determines which nodal data components in an unstructured mesh will be sent to the leftmost output port. Also, the first map component selected will be used to color the output. For example, choosing iso component concentration and map component uncertainty will create a volumetric subset of concentration colored by uncertainty. Initially, only the first map component is selected.

The ***Map Cell Components*** option box selector lets you map cell data (if any) to the surfaces output by isovolume. This is markedly faster than using interp_cell_data. The cell data will only be visible if all nodal data is unselected (under *Map Components*).

The ***Iso Level*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default iso level value is the arithmetic average of the minimum and maximum values in the iso component. If your input data has been kriged with log processing, the values here will be the Log of your input data.

The ***Exponentiated Values*** toggle makes another type-in field visible which convert real units to logarithmic units for you automatically.

The ***Data Above(on) or Below(off) Iso Level*** check box (Above Box) is used to display data above the iso level or below the iso level. For example, to display a volumetric subset of all concentrations greater than or equal to 1

ppm, set iso level to 1 (assuming concentration units are in ppm) and set the Above Box to on (check in the box). To see the plume_shell of 1 ppm and below, simply turn the Above Box off (no check in box).

The ***Remove Normals Generation*** toggle is equivalent to setting Normals Generation (in Object.Modes) to None. This changes the rendering of surfaces and is sometimes preferable.

The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. The additional ports represent:

Iso_Component name: This is a string with the name ot the selected data component.

Above/Below: This is a string containing the word "Above" or "Below" depending on the state of the toggle above.

Iso_Level: This is a real number representing the subsetting level.

Exponentiated Iso_Level: This is a real number representing the exponentiated subsetting level.

Related Modules

->isosurface

-> isolines

-> plume_shell

isosurface

This is a deprecated module whose function has been superceded by constant_shell

General Module Function

The constant_shellmodule produces a surface of constant value within the data domain. Isosurfaces are essentially three-dimensional contour lines. Each instance of isosurface can produce only a singular isosurface. However, multiple instances may be used to create more than one isosurface in the same data set. The isosurface of one component may be colored by the value of another component (for example, the isosurface of concentration colored by uncertainty).

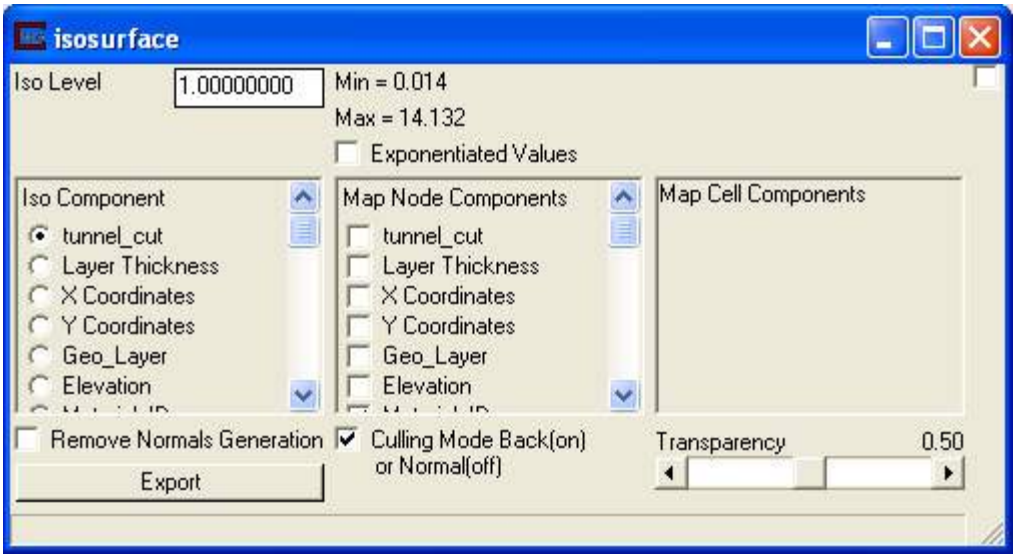
Module Input Ports

isosurface has two input ports. The leftmost port accepts unstructured mesh data. The second port provides a means to share the subsetting level of other modules.

Module Output Ports

isosurface has three output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains tri (polytriangle mesh) cell types representing the isosurface. Nodal data sent to this output port

consists of the interpolated values of the map component on the isosurface. The second port outputs a rendered geometry directly to the Viewer. The third output port provides a means to share the subsetting level of this module with others.



Module Control Panel

The control panel for isosurface is shown in the figure above.

The ***Iso Level*** type-in is used to set the level for subsetting the input field. If a value is chosen larger than the max value, the max value is placed in the edit box. Similarly, if a value less then the minimum is input, the minimum value is placed in the box. The default iso level value is the arithmetic average of the minimum and maximum values in the iso component. If your input data has been kriged with log processing, the values here will be the Log of your input data.

Selecting the **Exponentiated Values** will create a new field to appear that contains the exponentiated value of the Iso level.

Iso Components refers to the nodal data component used to create the subset of the original input field. When an iso component is selected, the min and max values of the variable are displayed next to the Iso Level type-in.

MapNodeComponents determines which nodal data components in an unstructured mesh will be sent to the leftmost output port. Also, the first map component selected will be used to color the output. If no component is selected the output will be colored by the first cell component mapped to the output. If no cell component is mapped the output will be a default white color.

The ***Map Cell Components*** option box selector lets you map cell data (if any) to the surfaces output by plume_volume. This is markedly faster than using interp_cell_data. The cell data will only be visible if all nodal data is unselected (under *MapNodeComponents*).

Related Modules

->plume_volume

-> isolines

Solid_3D_Set

This is a deprecated module whose function has been superceded by contour_data

General Module Function

Solid_3D_Set is an enhancement to the Solid_3D module. The enhancement involves type-ins for setting non-linear threshold values for any number of threshold cutoffs. For example, 5 cutoffs between 0 and 100 would *NOT* have to be 0, 20,40, 60,etc, but instead could be 0.5, 5, 10, 25, and 50. The subsetting level is set by a user type-in value and the desired number of contour levels is set with a slider. The levels can be reset to initial values with the reset button, then the contours levels are adjusted and set by the user as needed.

Module Input Ports

Solid_3D_Set has two input ports.

The first (blue-black) port accepts unstructured mesh and nodal data.

The second port (blue-red-blue-beige) is found on only a few modules in EVS/MVS. The modules with this port are Solid_3D_set, solid_contour_set, isolines, Datamap_Editor, and Color_Legend.

These modules pass data corresponding to one or more subsetting levels (or solid contour levels). When this data is passed to Solid_3D_Set, it is used to set the precise break points where contours will occur. When this data is passed from Solid_3D_Set to other modules, the levels set in Solid_3D_Set will be reflected in those modules.

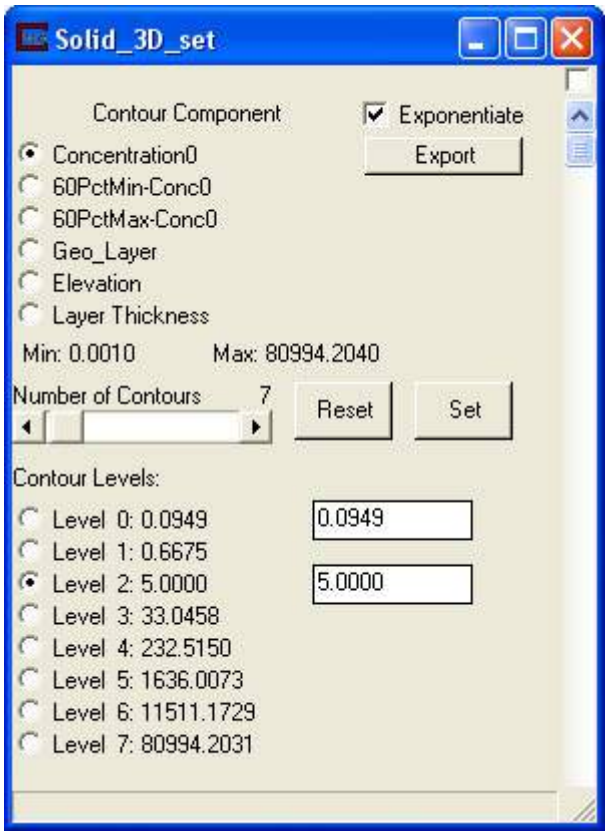
Module Output Ports

Solid_3D_Set has two output ports. The first output port (closest to the left) outputs a rendered geometry directly to the Viewer.

The second port (blue-red-blue-beige) is found on only a few modules in EVS/MVS. The modules with this port are Solid_3D_set, solid_contour_set, isolines, Datamap_Editor, and Color_Legend.

These modules pass data corresponding to one or more subsetting levels (or solid contour levels). When this data is passed to Solid_3D_Set, it is used to set the precise break points where contours will occur. When this data is passed from Solid_3D_Set to other modules, the levels set in Solid_3D_Set will be reflected in those modules.

The **Export** button causes additional output ports to appear as shown above.



Module Control Panel

The control panel for solid_3D_Set is shown in the figure above. The subsetting level of the selected component is changed by adjusting the Level Min type-in box. The user then adjusts the number of contours with the slider. By default, contour intervals are divided evenly between the Level Min and maximum value in the mesh. The process of setting user defined cutoffs requires the following order of steps:

first set the number of contours,

then set the Level Min

then set the contour level type-ins

then click the Set button.

If you wish to specify a different number of contours, all levels will be re-set to even increment contour spacing and the above process must be repeated. If you have set the contour levels but then want to shave away using level min, you simply type-in the new Level Min and click set. Solid_3D_Set will automatically shift all levels which are currently below this new Level Min to match it. The result is that all contour levels remain constant while shaving away to a higher concentration.

The ***Export*** button adds additional output ports to the module to facilitate passing text and numeric data to other modules. The result is shown above. The additional ports represent:

a. Iso_Component name: This is a string with the name ot the selected data component.
b. Above/Below: This is a string containing the word "Above" or "Below" depending on the state of the toggle above.
c. Iso_Level: This is a real number representing the subsetting level.
d. Exponentiated Iso_Level: This is a real number representing the exponentiated subsetting level.

Related Modules

-> _plume_shell

-> isolines

-> Solid_3D

Solid_3D

This is a deprecated module whose function has been superceded by contour_data

General Module Function

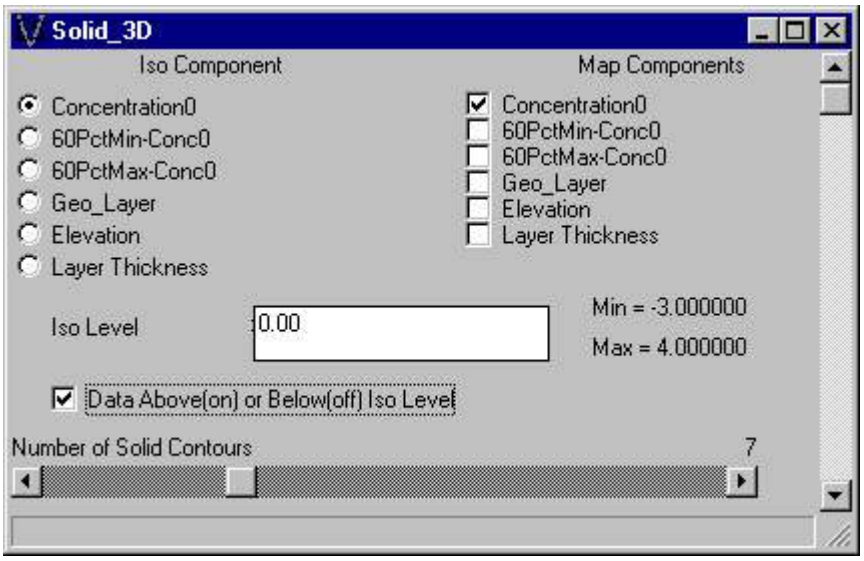
Solid_3D is identical to the plume_shell module, except it maps solid colors to a user specified set of cutoffs, as opposed to the Gouraud shading (linear) across the data range. Like the plume_shell module, Solid_3D is identical to a subnetwork containing the plume_volume module with constant_shell and external_faces. This module creates a superior visualization of a plume that can be sent directly to the viewer for rendering. This is not a subsetting module (as is plume_volume or contour). It is used exclusively for plume visualization of 3D fields (such as the output of Krig_3D).

Module Input Ports

Solid_3D has one input port which accepts unstructured mesh and nodal data.

Module Output Ports

Solid_3D_ has one output port which outputs a rendered geometry directly to the Viewer.



Module Control Panel

The control panel for solid_3D is shown in the figure above. The parameters are identical to the plume_shell parameters with an additional slider for setting the number of solid contour levels. The slider choice will result in n number of contours at even increments between the min and/or max subsetting level. If the even spacing is not desirable, the module Solid_3D_Set can be used to set un-evenly spaced contour levels.

Note that the coloring of each solid region is based on an average of the bordering contour levels. If this is objectionable try selecting the Solid_3D object in the Viewer, then choose datamap_editor-->Options-->Edit Range/Data-->then adjust Range Size slider to match the number of contours. This should solve any averaging issues with Solid_3D.

Related Modules

-> _plume_shell

-> isolines

-> plume_volume

solid_contour_set

This is a deprecated module whose function has been superceded by contour_data

General Module Function

The solid_contour_set module has identical functionality to Solid_3D and Solid_3D_Set This module is used for applying solid contours with user defined contour levels to objects such as slice planes, plume_volumes, geologic surfaces, etc.

Module Input Ports

solid_contour_set has two input ports.

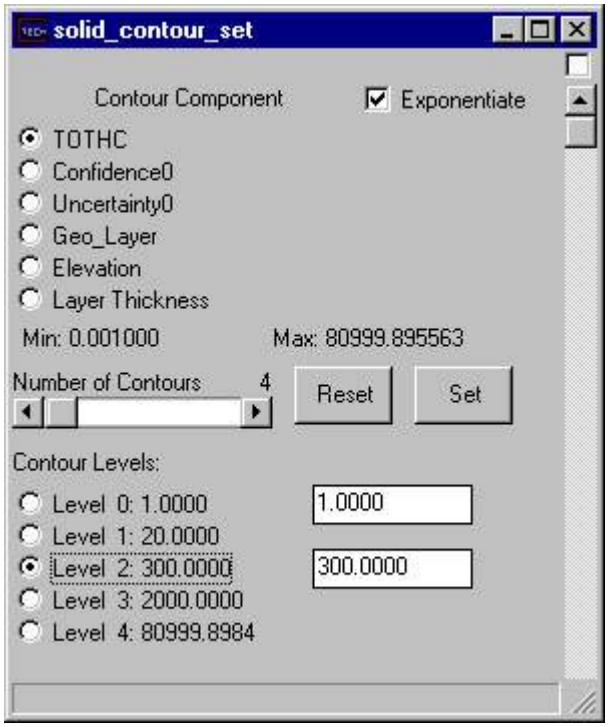
The first (blue-black) port accepts unstructured mesh and nodal data.

The second port (blue-red-blue-beige) is found on only a few modules in EVS/MVS. The modules with this port are Solid_3D_set, solid_contour_set, isolines, Datamap_Editor, and Color_Legend.

These modules pass data corresponding to one or more subsetting levels (or solid contour levels). When this data is passed to solid_contour_set, it is used to set the precise break points where contours will occur. When this data is passed from solid_contour_set to other modules, the levels set in solid_contour_set will be reflected in those modules.

Module Output Ports

solid_contour_set has two output ports. The first output port (closest to the left) outputs a new unstructured mesh which contains cells representing the external faces portion of solid_contour_set. Nodal data sent to this output port consists of the nodal data of the map component in the plume_volume portion of the subnetwork. This blue output is used to input to isolines to place isolines on the portions of the plume that intersected the exterior surfaces of the input domain. The second port outputs a rendered geometry directly to the Viewer.



Module Control Panel

The panel for solid_contour_set is shown above. The rules and functionality is described in Solid_3D_Set help.

Related Modules

-> isolines

-> Solid_3D_Set

Macro

General Module Function

The macro module is simply a container for a frequently used network fragment. Once saved with an appropriate name, the macro is instanced as an object just like any other EVS module. Any EVS modules within the macro are accessible once instanced and settings within the macro are savable. The purpose of this tool is to simplify your network by saving commonly used module configurations for use in other applications.

Module Input Ports

No input ports.

Module Output Ports

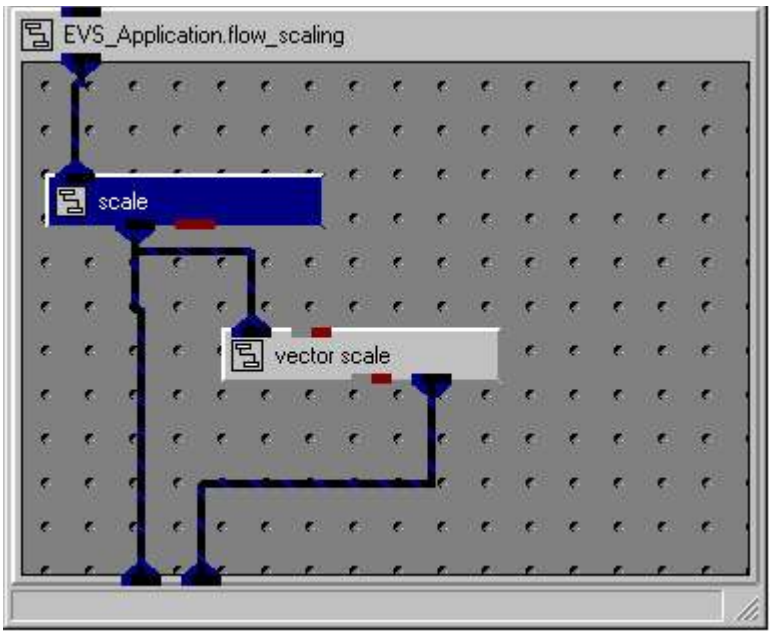
No output ports.

Module Control Panel

This module has no control panel but instead supplies a workspace into which modules are instanced for incorporating into the macro. The macro (workspace) must be "opened" by instancing the macro module, then double clicking with the left mouse (or right click and select open). This reveals the contents of the macro module workspace, which is initially empty. Into this new workspace the user must instance and connect the modules of the network fragment they wish to save. To create input and output ports to the macro, the user must make connections (which either receive data from upstream of the network fragment, or send data downstream from the network fragment) to the top or bottom of the macro workspace.

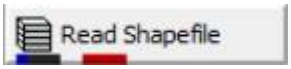
Once this macro is created the user should double click the top title bar of the macro, thus closing the workspace. Then, with the macro selected (blue shading), choose Object ->Rename from the EVS pull down menus, then name the macro appropriately. Then save the macro by choosing Object->SaveObjects and save it under its' new name with a .v extension in any directory. This module may be instanced into any application at a later time by first selecting the current EVS Application (single click top bar) choose Object ->LoadObjects, then select the *.v file representing your macro.

An example of a simple macro workspace called flow_scaling is shown below with one input port and two output ports.



This workspace results in a macro which when instanced looks like this.

Read_Shapefile



(This module has been deprecated. Its functionality has been surpassed by the read_vector_GIS module.)

General Module Function

The Read_Shapefile module will read an industry standard ESRI Shapefile (*.shp) format file and place the graphic entities into the EVS viewer. This module provides the user with the capability to integrate common street, parcel, water bodies and other ArcView feature data into the EVS visualization to provide a frame of reference for understanding the three dimensional relationships between the site features and the characteristics of geologic, hydrologic, and chemical features. ESRI 3D Shapefiles are supported as well, allowing the import of data directly from ArcView 3D Analyst into the EVS visualization.

Nearly all well formatted shapefiles are supported* and will be imported, and their data mapped into cell data. EVS will map all of the attributes in the shapefile's corresponding DXF file to cell data components. The output from the left output port can be used by the Cell_Data modules to affect the EVS visualization of the shapefile based on the cell data.

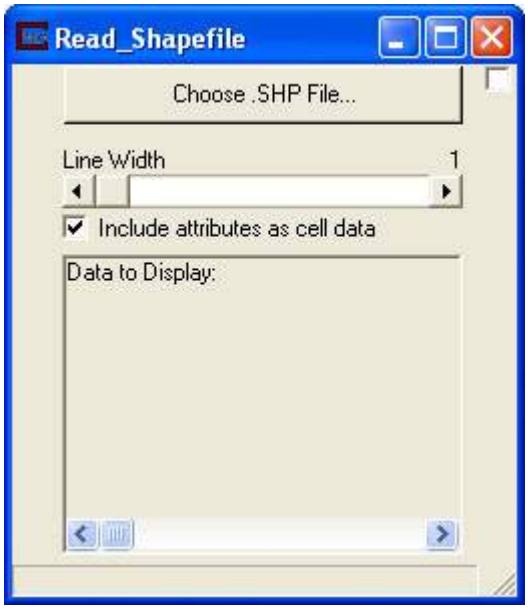
*NOTE: At this time, only Multipatch shapefiles are not supported. All point, arc, multipoint, and polygon shapefile (including ArcM, PolygonZ, etc.) types are fully supported.

Module Input Ports

The Read_Shapefile module is shown above. Read_Shapefile has no input ports. It obtains the shapefile input data by reading a file with a file browser.

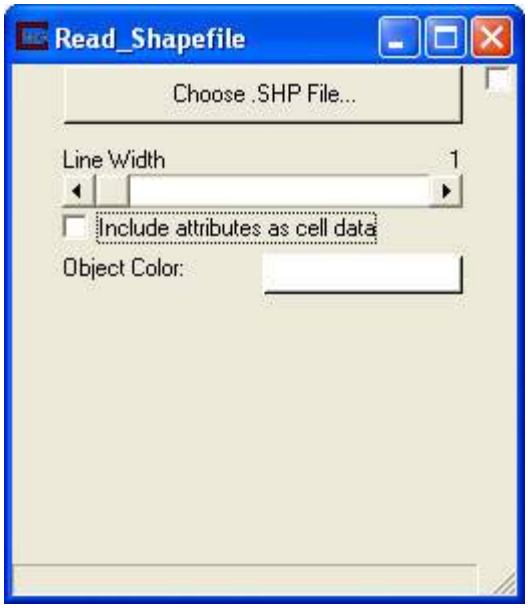
Module Output Ports

The first (left) port outputs a field containing cells and cell data. Read_Shapefile does not create nodal data. If nodal data is desired, use cell_to_node to convert. The second (right) port outputs a renderable geometry which is normally connected to the viewer or the Datamap_Editor module.



Module Control Panel

The user interface for Read_Shapefile is shown above. Clicking the Choose SHP File button opens a standard windows style file browser which allows the user to select a .shp file from a specified directory. There are many different types of entities, and some differences in formats, that can be present in shapefiles.



The "Include attributes as cell data" toggle causes the output to include the selected attributes present in the shapefile to be represented as cell data. If this is unchecked, a color button appears (as shown in the figure above) that allows you to set the color of the output mesh.

The Line Width slider sets the width of lines in pixels.

The "Accept Current Values" button causes the module to run.

Write_3D_Shapefile

(This module has been deprecated. Its functionality has been surpassed by the write_vector_GIS module.)

General Module Function

Write_3D_Shapefile provides EVS with vector output of graphical results. Write_3D_Shapefile will output surface and line type objects created in EVS as an ESRI formatted shapefile. Examples of compatible input are isolines, external_faces, Krig_2D, and isosurface. (Note: plume_shell does not output all of its objects through its' blue port.)

If you connect volumetric data (e.g. from Krig_3D or after plume_volume acting on volumetric data) Write_3D_Shapefile will output only the external faces of each cell set in your model.

Surface output will be flat shaded (all triangles or quadrilaterals will be a single color vs. Gouraud shaded and the number of unique colors is limited). Multiple objects can be output using Write_3D_Shapefile several times, and all objects can be combined in a single ESRI project. This module preserves true user coordinates and all output is 3D.

Write_3D_Shapefile supports POINT,LINE, POLYLINE, TRI(angle), POLYTRI(angle), QUAD(rilateral), TET(rahedron), PYR(amid), PRISM, and HEX(ahedron) cell types. Since spheres are rendered as points with radii (i.e. they have no real surfaces), spheres are not supported in this module. If 3D cell types are input the output will contain the surfaces of every 3D element (cell) in the input.

Write_3D_Shapefile writes three-dimensional ESRI (ARC/INFO or ArcView) shapefiles. The module supports virtually all cell types in EVS, but only one cell type at a time. Use select_cells to screen outputs with multiple cell types.

All data in the field is included as columns in the attribute table of the created shapefile. Cell data is mapped across, one column per component. Nodal data is also mapped across, but is computed for each cell as the average of the data at the nodes comprising that individual cell.

Module Input Ports

Write_3D_Shapefile has only one input port. This port accepts unstructured mesh data and nodal data. If used with 3D data, then a slice plane or other subsetting module such as isosurface, or external_faces should be used or the output will contain the surfaces of every 3D element (cell) in the input.

Module Control Panel

The control panel for Write_3D_Shapefile is shown in the figure above.

The toggle *Prepend Nodal Flag* causes all columns of nodal data to be prepended with "n_" (ie: "n_Concentration"), signifying that the original data was per node, whereas the shapefile's data was mapped to be per cell. This allows you to distinguish if attributes were interpolated from nodal data.

Related Modules

-> read_vector_GIS

-> read_CAD

-> write_CAD

GSLIB_KT3D

(This module is available only in MVS, but has been superceded by the external_krigig module)

The help for this module is not available in C Tech's help. All parameter names are identical to the Stanford GSLIB KT3D program. Complete instructions for its' use are in the book: GSLIB (Geostatistical Software Library and User's Guide) Second Edition by Clayton V. Deutsch and Andre G. Journel published by Oxford University Press 1998.

The only parameters not referenced in the book are those two major enhancements made by C Tech. These include:

1. Inclusion of the input port to receive geologic data from Krig_3D_Geology
2. Log10 data toggle for log processing

C Tech GMS Project File Converter

The GMS->EVS/MVS Converter takes as input a GMS project file (*.gpr) and parses it looking for parts that can be converted to an EVS/MVS file format. **This supports all version 4 GMS projects only.** To fully convert a MODFLOW, MT3DMS, or FEMWATER project make sure to run that simulation in GMS and save your project before attempting a conversion. There should only be one loaded solution for each model.

The following GMS Project and File types are convertible into their respective C Tech (EVS/MVS) file types:

1. **MODFLOW**: This converts into both EVS/MVS field files (*.eff, *.efz) and into an EVS/MVS TCF file (*.tcf) for animation.

2. **MT3DMS**: This converts into both EVS/MVS field files (*.eff, *.efz) and into an EVS/MVS TCF file (*.tcf) for animation.
3. **FEMWATER**: FEMWATER converts only into an EVS/MVS field file (*.eff, *.efz) at this time, because it is only handling steady state FEMWATER models. When this is rewritten to handle transient FEMWATER models, a TCF file (*.tcf) will also be added.
4. **IMAGE**: Image files (*.tif and *.jpg) are copied into the output directory and a georeferencing file (*.gcp) is created for image orientation.
5. **2D SCATTER**: These files are converted into EVS/MVS Geology Multi File format (*. gmf).
6. **3D SCATTER**: These files are converted into EVS/MVS 3D analyte (e.g. chemistry) files (*.apdv).
7. **2D Grid**: 2D Grid files are converted into the EVS/MVS Field file format (*. eff, *.efz) as well as EVS Geology Multi file format (*.gmf).
8. **TIN**: TIN files are converted into the EVS/MVS Field file format (*.eff, *.efz) as well as EVS Geology Multi file format (*.gmf).
9. **BOREHOLE**: BOREHOLE data is converted to the EVS Pre-Geology file format (*. pgf).

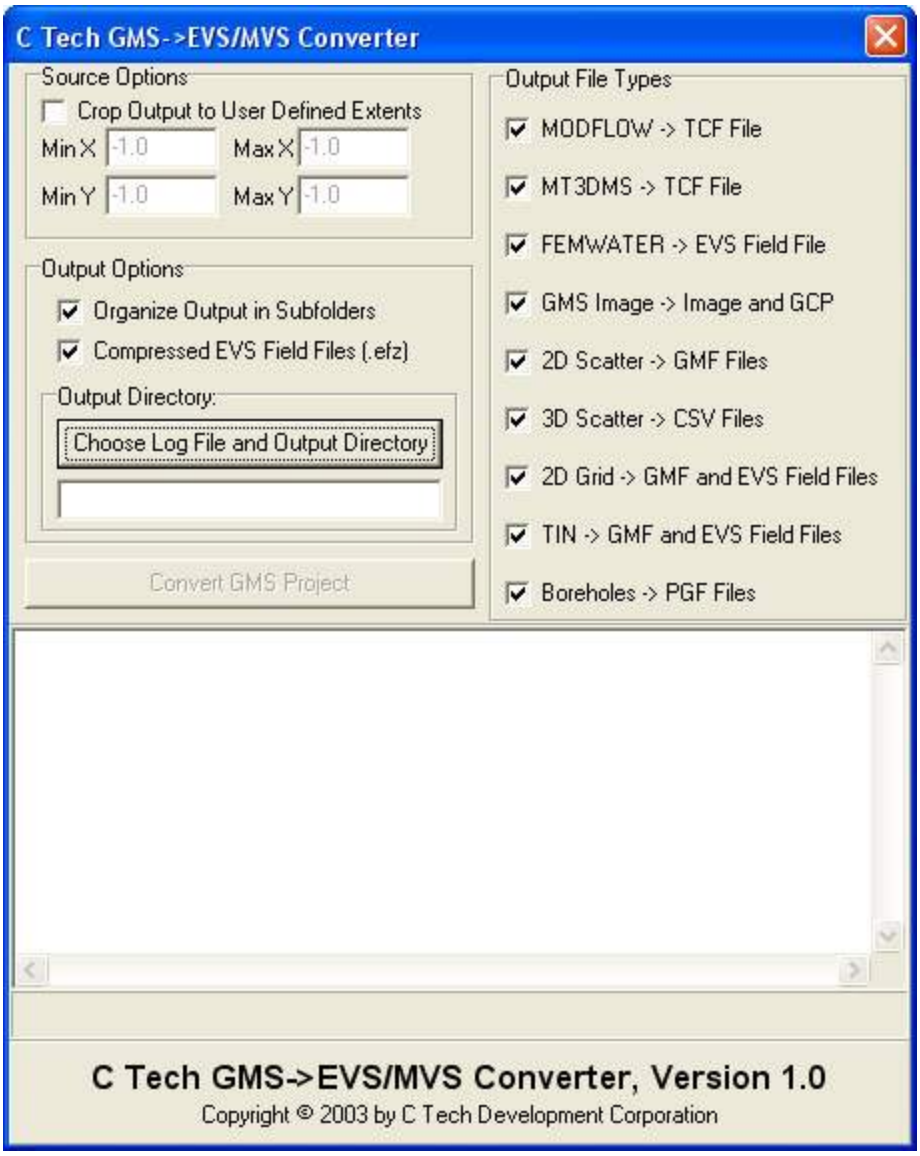
There is one source control option. By clicking on Crop Output to User Defined Extents the input from GMS can be focused on areas of interest to the user. These extents are assumed to be valid when entered by the user. No Error checking is done. This works only on MODFLOW, MT3DMS, and FEMWATER conversions.

There are two options for output:

1. **Organize output to subfolders:** This allows the user to keep track of which specific part of the project was converted by placing the appropriate converted files in new subfolders based on the data type (e.g. MODFLOW, IMAGE, etc.).
2. **Compressed EVS Field Files(*.efz):** This option changes all EVS/MVS field files (*.eff) into their compressed form (*.efz) for space saving purposes.

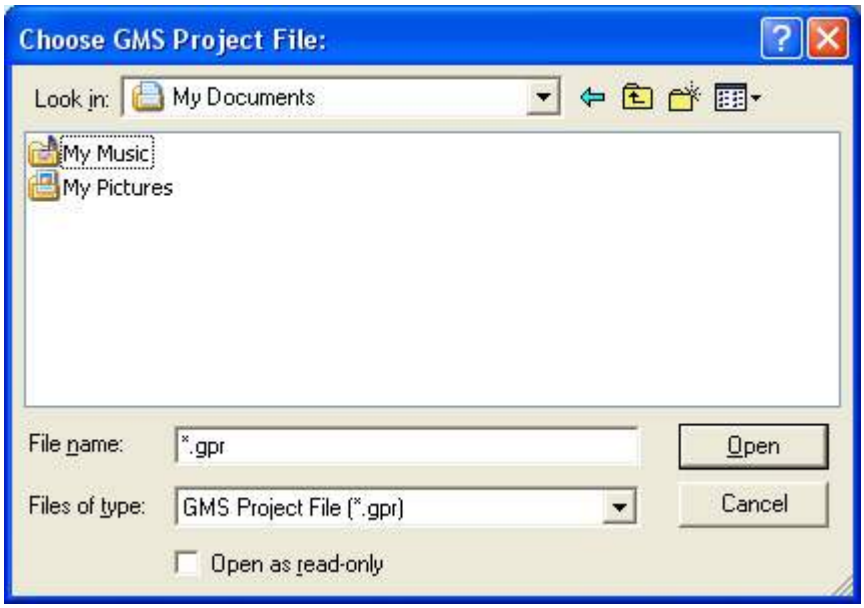
To run the converter

1. Select all appropriate options and file types to be converted.



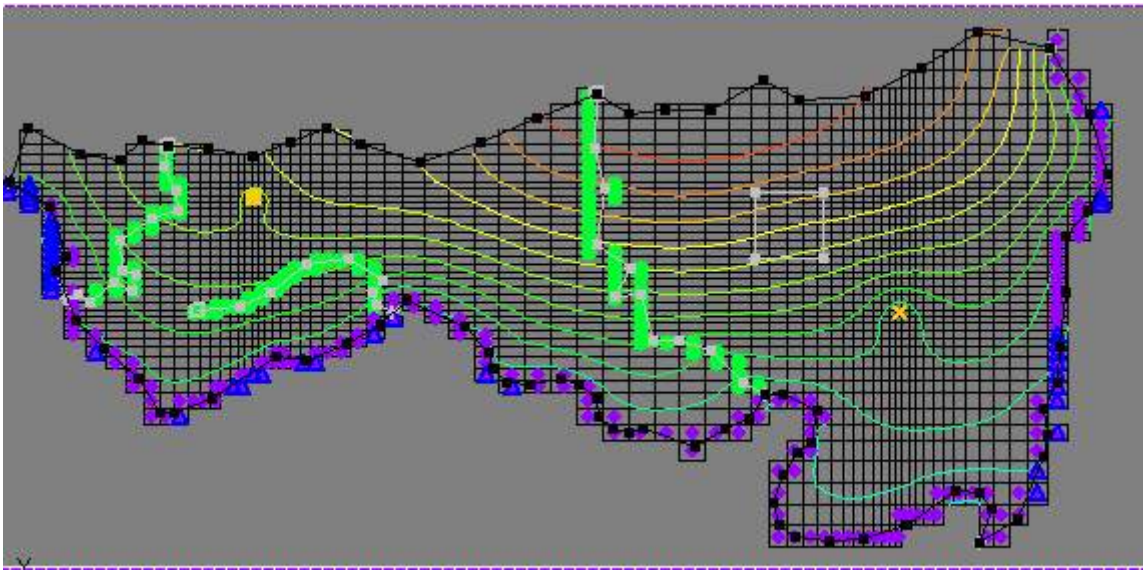
2. Click the Choose Log File and Output Directory button to choose the destination for the log file and converted files/folders. The log file is a status report of the converter which keeps track of all errors and files created.

3. Next click on the Convert GMS Project button and select the GMS project file (*.gpr) to be converted. All files should be created and shown in the Status window, click Done to finish.

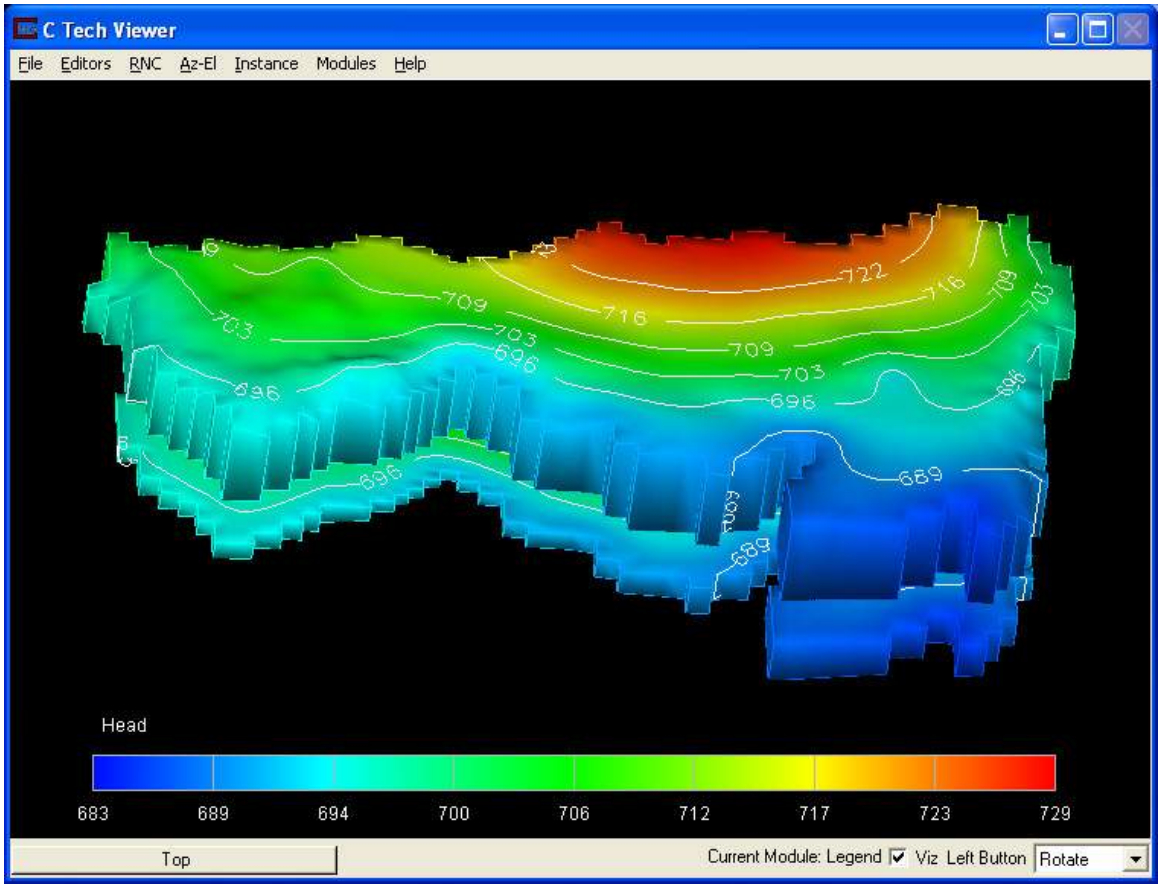


Example:

The following sample MODFLOW model in GMS was converted using the converter:



Upon running the GMS to EVS converter, a compressed EVS field file was created. This file can then be used in a variety of ways in EVS or MVS, as shown in the picture showing head values below:



C Tech GMS Project File Converter

The GMS->EVS/MVS Converter takes as input a GMS project file (*.gpr) and parses it looking for parts that can be converted to an EVS/MVS file format. **This supports all version 4 GMS projects only.** To fully convert a MODFLOW, MT3DMS, or FEMWATER project make sure to run that simulation in GMS and save your project before attempting a conversion. There should only be one loaded solution for each model.

The following GMS Project and File types are convertible into their respective C Tech (EVS/MVS) file types:

1. **MODFLOW**: This converts into both EVS/MVS field files (*.eff, *.efz) and into an EVS/MVS TCF file (*.tcf) for animation.
2. **MT3DMS**: This converts into both EVS/MVS field files (*.eff, *.efz) and into an EVS/MVS TCF file (*.tcf) for animation.
3. **FEMWATER**: FEMWATER converts only into an EVS/MVS field file (*.eff, *.efz) at this time, because it is only handling steady state FEMWATER models. When this is rewritten to handle transient FEMWATER models, a TCF file (*.tcf) will also be added.
4. **IMAGE**: Image files (*.tif and *.jpg) are copied into the output directory and a georeferencing file (*.gcp) is created for image orientation.
5. **2D SCATTER**: These files are converted into EVS/MVS Geology Multi File format (*. gmf).
6. **3D SCATTER**: These files are converted into EVS/MVS 3D analyte (e.g. chemistry) files (*.apdv).
7. **2D Grid**: 2D Grid files are converted into the EVS/MVS Field file format (*. eff, *.efz) as well as EVS Geology Multi file format (*.gmf).
8. **TIN**: TIN files are converted into the EVS/MVS Field file format (*.eff, *.efz) as well as EVS Geology Multi file format (*.gmf).
9. **BOREHOLE**: BOREHOLE data is converted to the EVS Pre-Geology file format (*. pgf).

There is one source control option. By clicking on Crop Output to User Defined Extents the input from GMS can be focused on areas of interest to the user. These extents are assumed to be valid when entered by the user. No Error checking is done. This works only on MODFLOW, MT3DMS, and FEMWATER conversions.

There are two options for output:

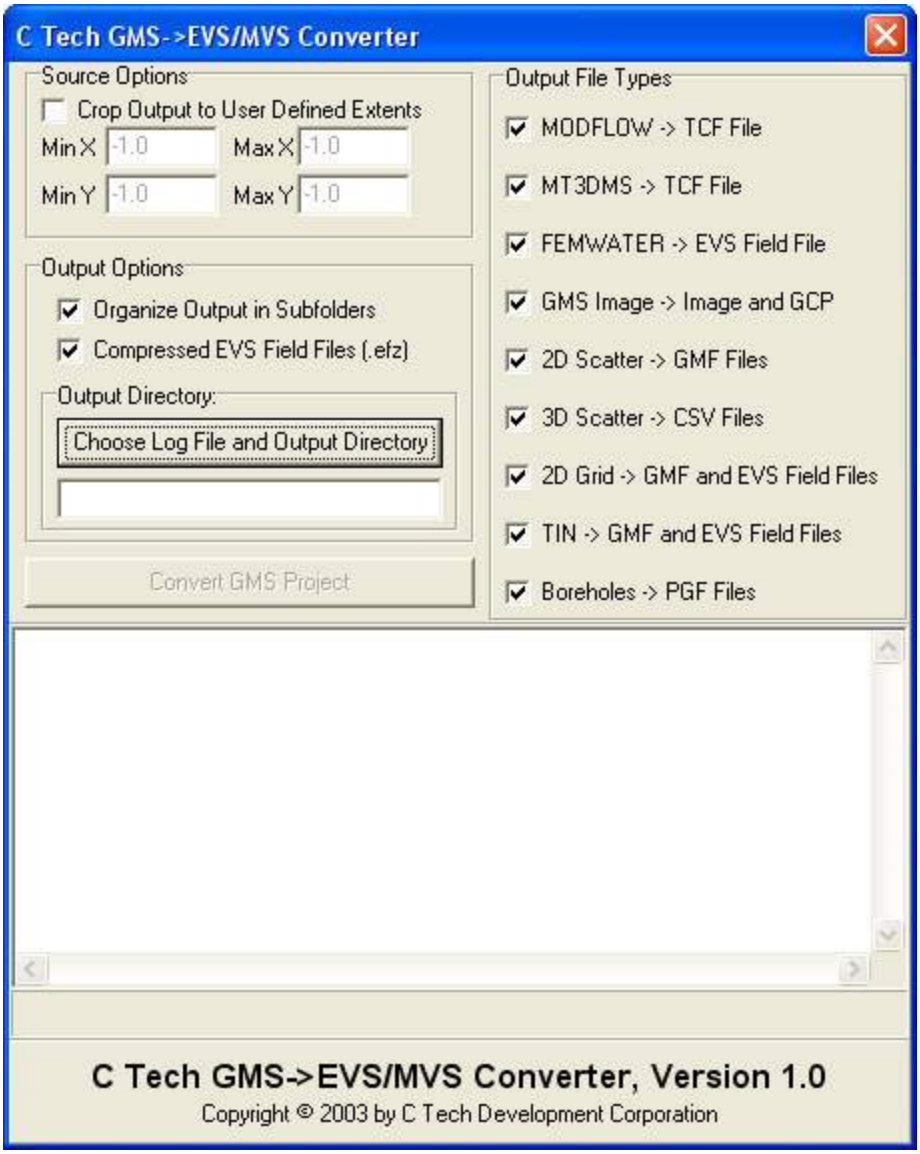
1. **Organize output to subfolders:** This allows the user to keep track of which specific part of the project was converted by placing the

appropriate converted files in new subfolders based on the data type (e.g. MODFLOW, IMAGE, etc.).

2. **Compressed EVS Field Files(*.efz):** This option changes all EVS/MVS field files (*.eff) into their compressed form (*.efz) for space saving purposes.

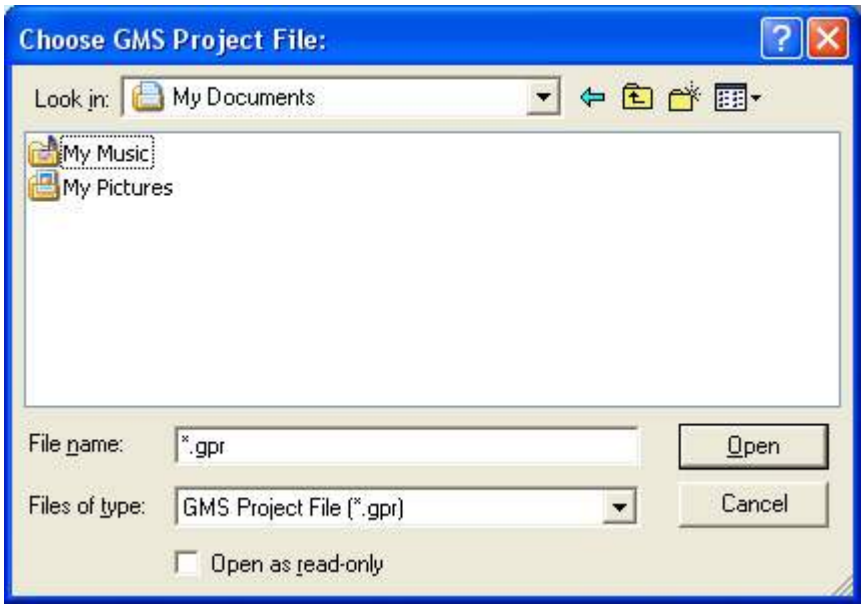
To run the converter

1. Select all appropriate options and file types to be converted.



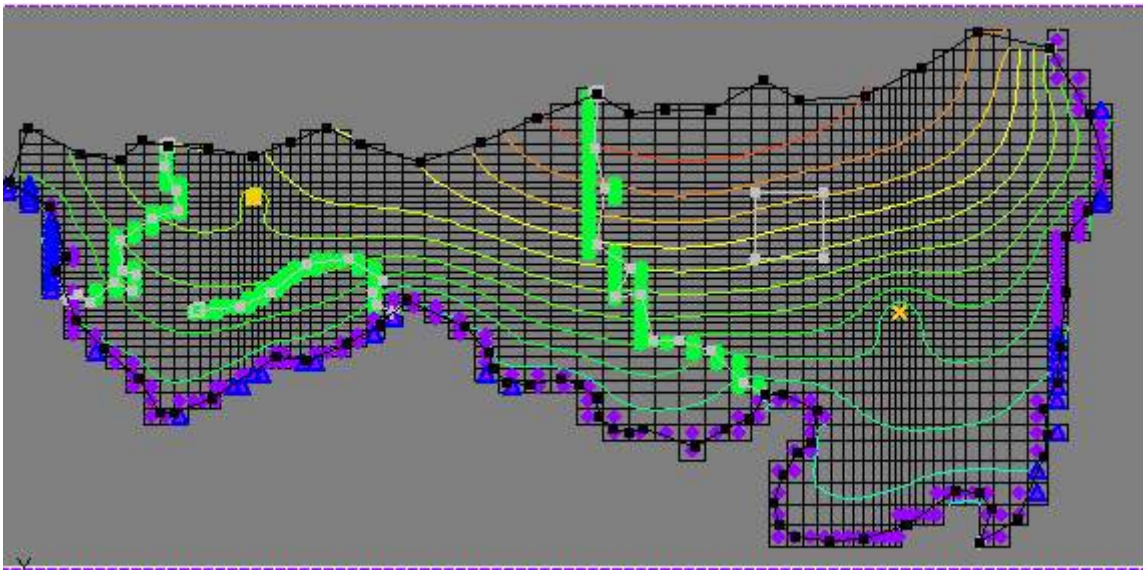
2. Click the Choose Log File and Output Directory button to choose the destination for the log file and converted files/folders. The log file is a status report of the converter which keeps track of all errors and files created.

3. Next click on the Convert GMS Project button and select the GMS project file (*.gpr) to be converted. All files should be created and shown in the Status window, click Done to finish.

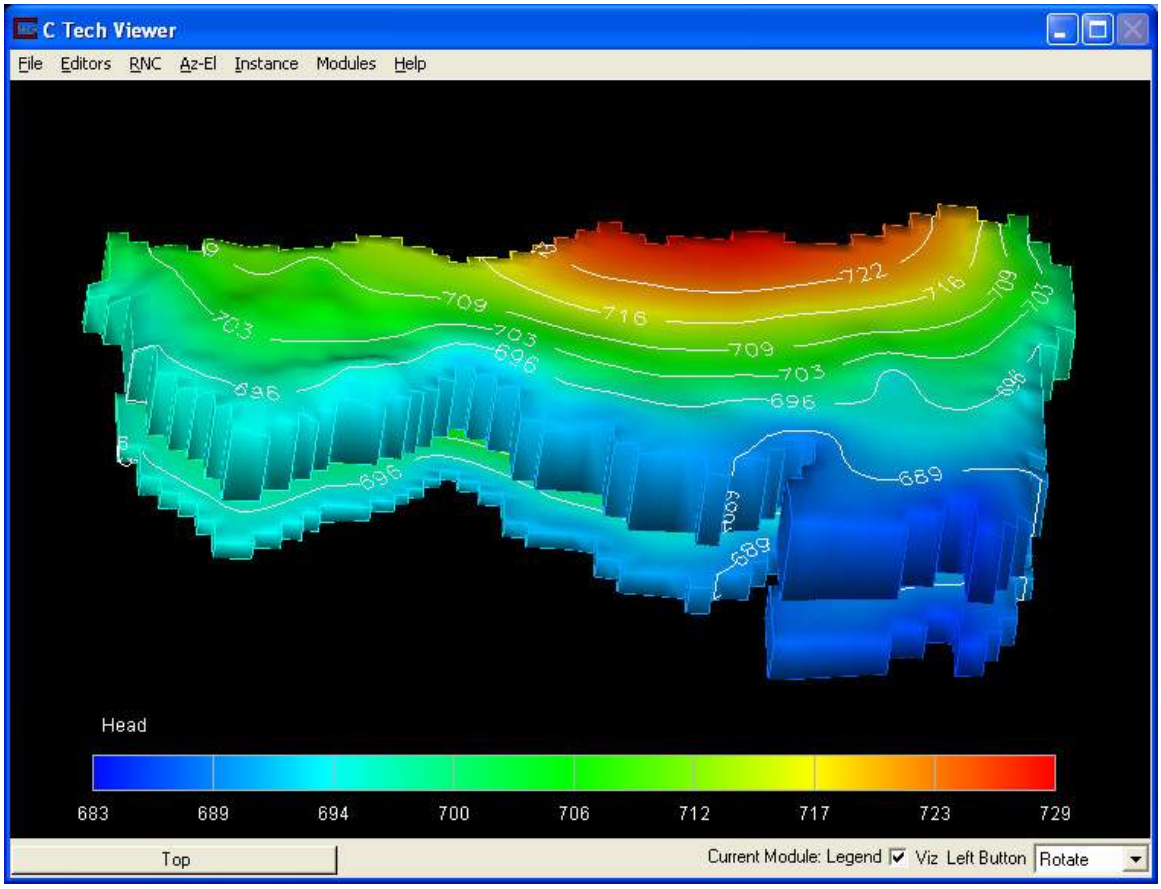


Example:

The following sample MODFLOW model in GMS was converted using the converter:



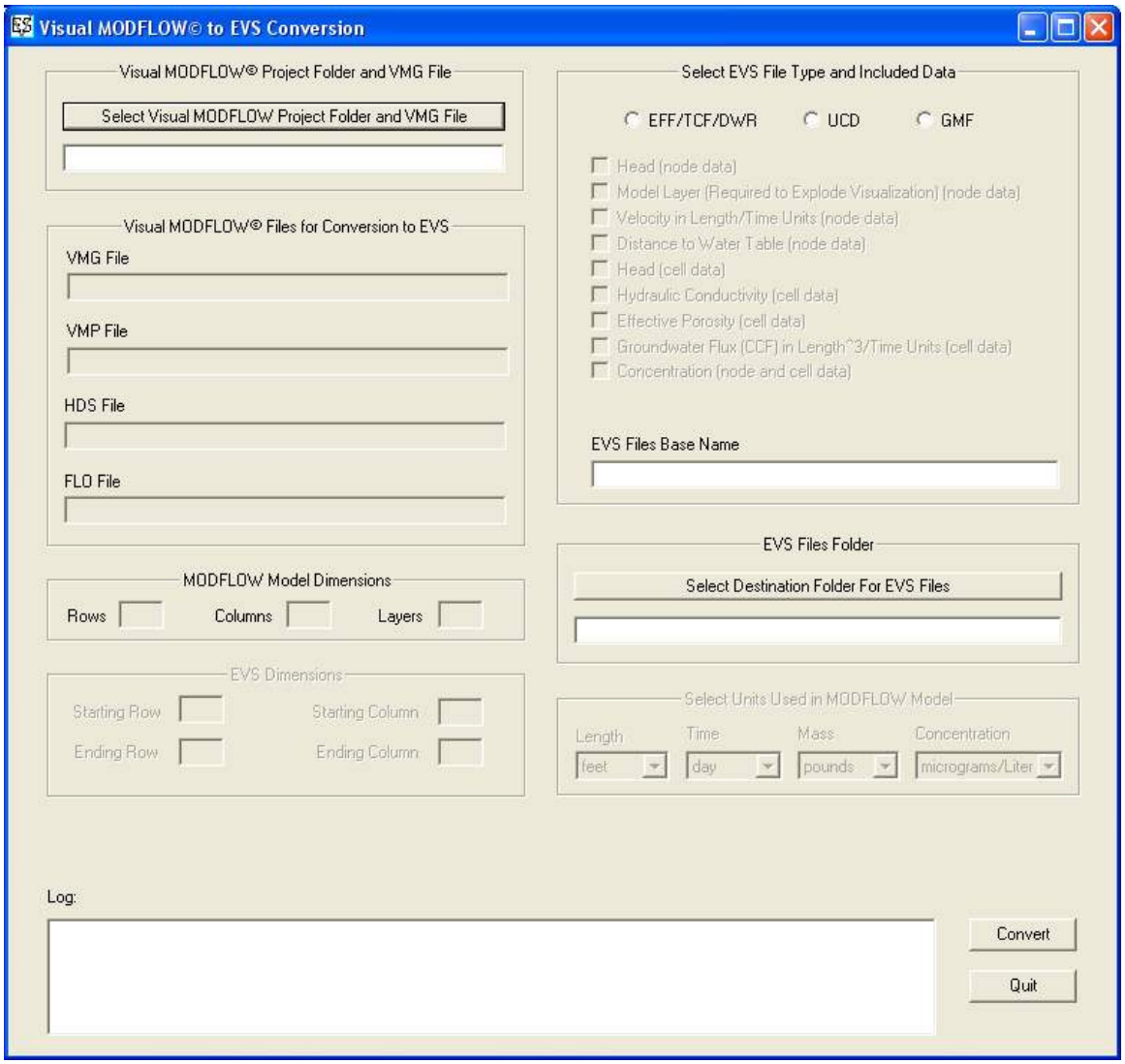
Upon running the GMS to EVS converter, a compressed EVS field file was created. This file can then be used in a variety of ways in EVS or MVS, as shown in the picture showing head values below:



VM_to_EVS

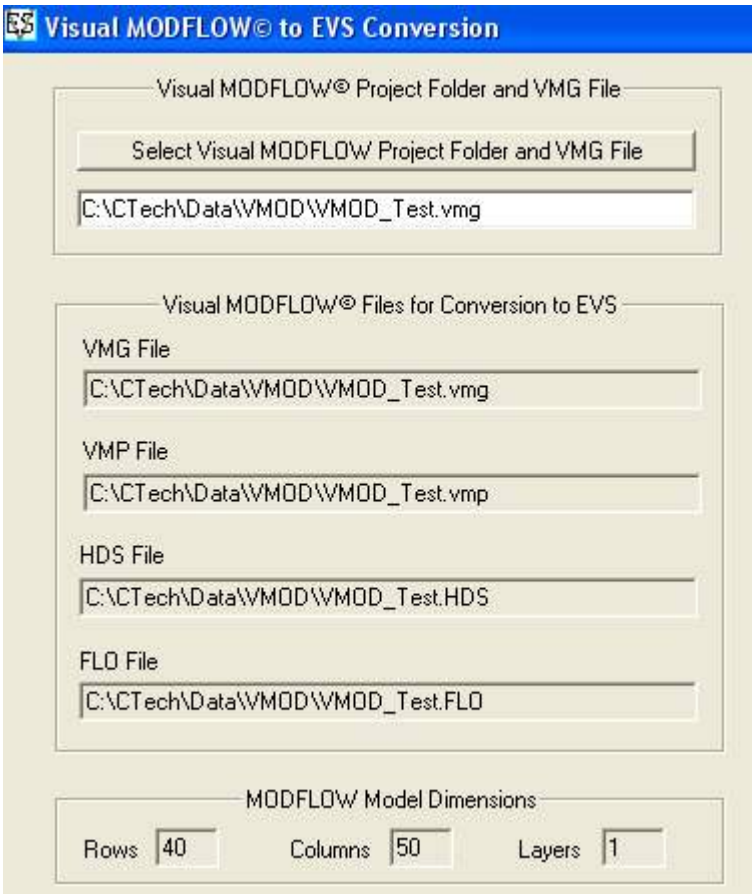
General ToolFunction

The VM_to_EVS tool can convert a Visual MODFLOW project to a several useful EVS file formats. This tool is not supported by C Tech Development Corporation, but is useful for visualizing VMOD projects.

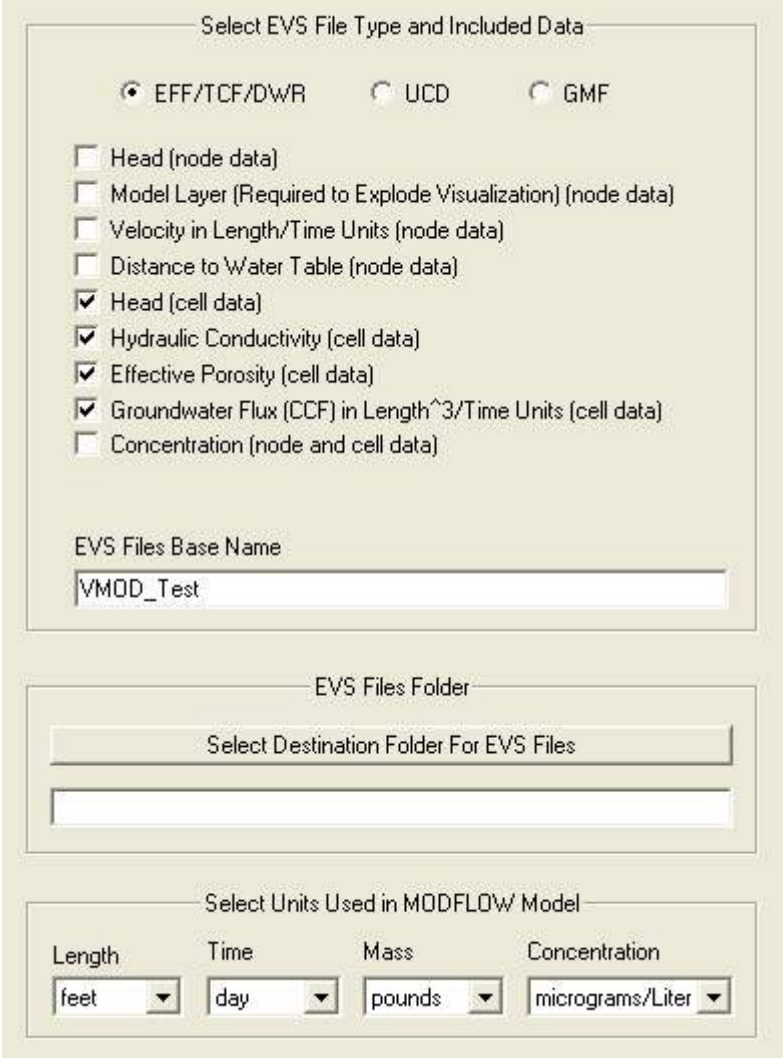


ToolControl Panel

The control panel for the VM_to_EVS converter can be seen above. The first step in converting a project is to select a Visual MODFLOW project (*.VMG) with the ***Select Visual MODFLOW Project Folder and VMG File button.*** The converter will then locate all additional VMOD project files (as shown below). The user can also crop the extents of the project that has been read in by changing the values in the Starting Row/Column and Ending Row/Column fields.

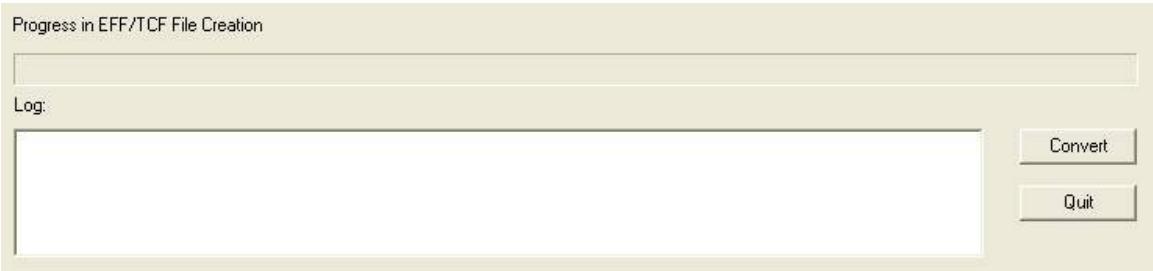


The next step is to select the type of output that is desired. The first (and recommended) option is to create a set of EFF/TCF/DWR files. To use the MODPATH (MVS Only) module the cell components Head, Hydraulic Conductivity, Effective Porosity, and Groundwater Flux must be selected as shown below.



To use the streamlines module the Velocity component must be selected. The MODFLOW units should be checked to ensure that the right coordinates are used.

The final step is select the Convert button (as seen below), when the progress bar has finished then select the Quit button to exit.

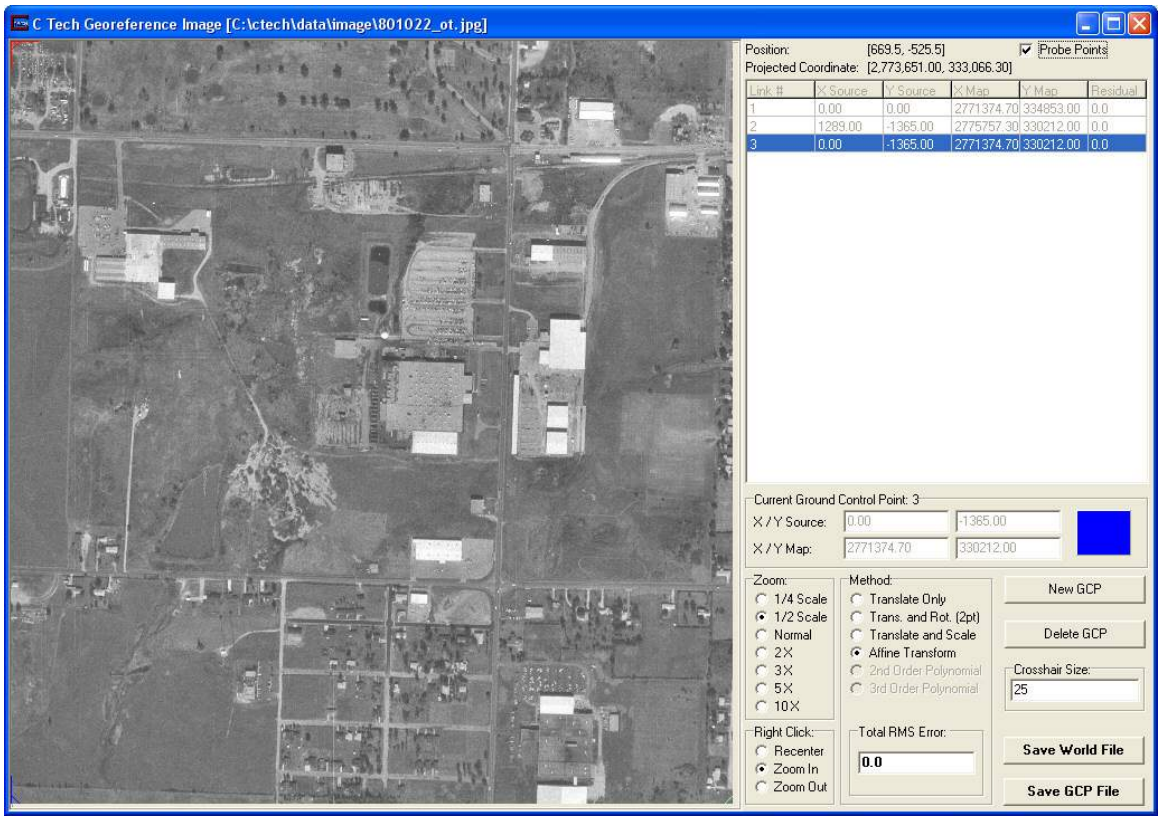


Georeference Image

GeoreferenceImage is a standalone utility program which can create world files or .gcp (ground control point) files for images. It supercedes the georeference_image module. The .gcp files are compatible with ArcGIS image link files, but use a .gcp extension (ArcGIS defaults to .txt extension

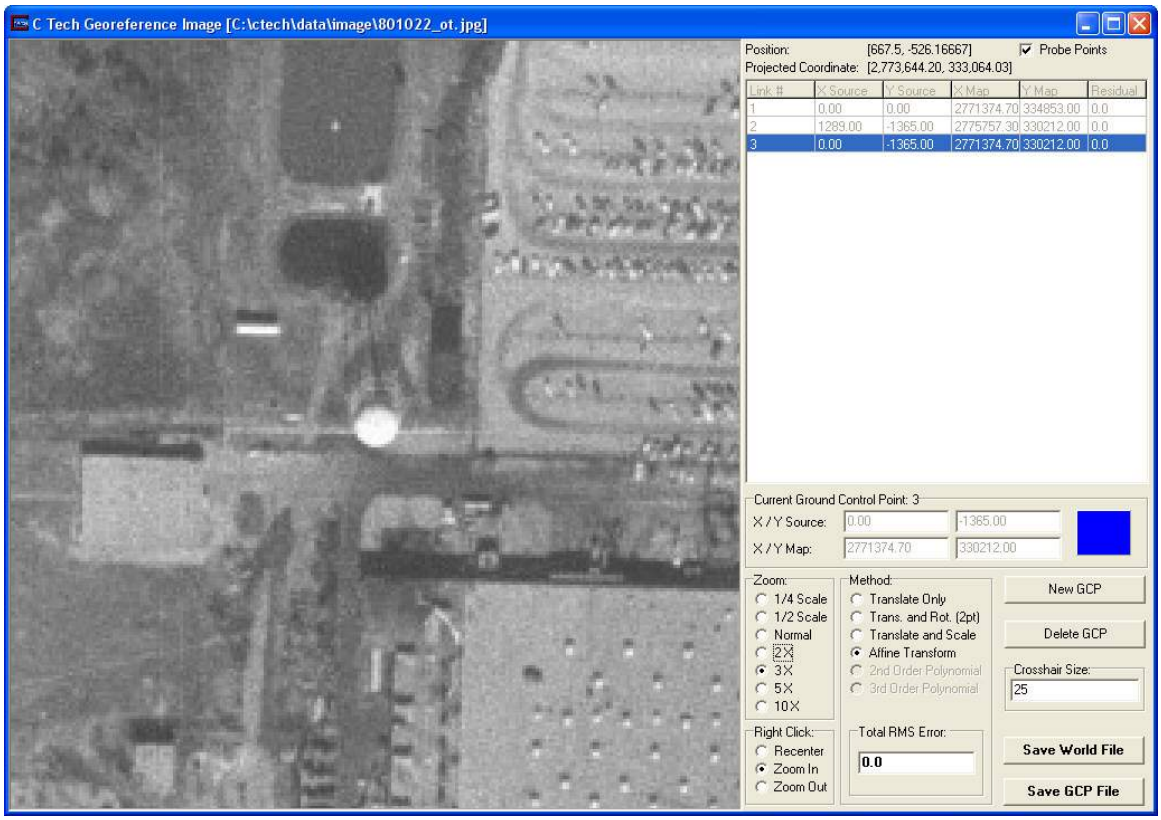
instead of .gcp, but these are compatible). **GeoreferenceImage** will allow you to view total root mean squared error given the number of Ground Control Points you have and the selected texture_map mapping option. This standalone utility can be run with a single button click from Read_Image or by running from the *tools* folder in the C Tech program group.

When you run Georeference Image from the *tools* folder in the C Tech program group it will initially prompt you for an image to open. If run from Read_Image, it will open the image already selected in Read_Image. The example shown below uses the image 801022_ot.jpg which has a corresponding world file 801022_ot.jpw. Georeference Image automatically reads the world file and creates three ground control points (GCPs) as shown in the figure below.



If you have additional ground truth data, you can add additional GCPs to enhance the accuracy of the image projection.

When initialized with a world file, you can use Georeference Image to determine the x-y coordinates of any point on the image. By zooming in on the image (as shown below) it is possible to get accurate coordinates of individual pixels.



C TECH EARTH SCIENCE SOFTWARE
TERMS AND CONDITIONS OF
END USER LICENSE AGREEMENT

(version 130225)

THIS AGREEMENT SHALL GOVERN ORDERING, RENTAL, LEASING, SUBSCRIPTION, PURCHASE, LICENSING, SUPPORT, AND WARRANTY OF ALL OF C TECH'S EARTH SCIENCE SOFTWARE AND ASSOCIATED SOFTWARE TOOLS, INCLUDING THE ENVIRONMENTAL VISUALIZATION SYSTEM "EVS-PRO", MINING VISUALIZATION SYSTEM "MVS", AND FOUR-DIMENSIONAL INTERACTIVE MODEL ANIMATION PLAYER "4DIM PLAYER", HEREAFTER REFERRED TO AS SOFTWARE. THIS AGREEMENT IS BETWEEN C TECH DEVELOPMENT CORP. ("C TECH") AND THE CUSTOMER. THE AGREEMENT CONSTITUTES THE ENTIRE CONTRACT BETWEEN THE PARTIES AND SHALL GOVERN ALL ORDERS COMPLETELY UNLESS AMENDED IN WRITING. SUCH AMENDMENTS MUST BE SIGNED BY BOTH THE CUSTOMER AND AN AUTHORIZED REPRESENTATIVE OF C TECH IN ORDER TO BE CONSIDERED VALID.

C Tech Development Corporation
2360 Corporate Circle, Suite 400, Henderson, NV 89074 USA
sales@ctech.com
Phone: (800) 669-4387 Fax: (714) 844-9255

1. ORDERING: The Customer may order any item or package listed on the C Tech Price List at www.ctech.com (”Product”). C TECH does not guarantee that prices will not change, nor does it honor out-of-date price lists. No order will be effective until made in writing and signed by the Customer and an authorized representative of C TECH.

2. PAYMENT AND TAXES: Payment terms are net 10 days except international orders that require advance payment. An authorized representative of C TECH must approve any revisions to C TECH’s terms or license agreement in writing. Customer is responsible for all shipping charges, taxes, fees, duties, and any or all government charges that are customarily the responsibility of the Customer.

3. SHIPMENT AND DELIVERY: C TECH will ship the Product F.O.B. Destination, Prepaid and Added by means of courier service (e.g. FedEx) requiring Customer’s signature for receipt of delivery. The shipping charges will be paid by the shipping party and added to the invoice to the purchaser over and above the items purchased costs. In the event of loss or damage during shipment, the Customer agrees to assist C TECH in filing freight claims for loss or damage.

4. ACCEPTANCE & VALIDATON: Upon receipt of the Product (software key for licensing), the Customer must notify C TECH within ten (10) calendar days that the Product is not performing correctly, or the Customer acknowledges acceptance of the Product. Once the customer notifies C Tech that they have received the software, and C Tech verifies receipt of full payment for either rentals or purchases, C Tech will e-mail a Key Update File (.KUF). Use of the Key Update File by Customer constitutes Acceptance of the software and Validation of the software key.

5. THIRTY DAY PURCHASE GUARANTEE: C TECH guarantees Customer’s satisfaction with the Product for a period of thirty (30) days, beginning on the day of delivery. At any time during this thirty-day period, the Customer may contact C TECH, obtain a return authorization, and return the Product. Upon receipt of the entire Product, C TECH will refund the net purchase prices paid less 10% and any shipping and handling fees on the original invoice. Prior to acceptance, the Product may be returned for a full refund less shipping and handling costs. However, THE PRODUCT MUST BE SHIPPED BEFORE THE END OF THE THIRTY-DAY PERIOD TO QUALIFY FOR A REFUND.

6. SOFTWARE WARRANTY: C TECH guarantees that the Product (Software), as delivered or updated by C TECH and properly installed and operated, will perform substantially as described in its then-current documentation for 90 days from the date of delivery. If the Software fails to so perform during such period, the Customer shall promptly notify C TECH of, and adequately describe, the failure, and Customer's exclusive remedy and C TECH’s sole obligation shall be to take corrective action as specified in Section 8.B (relating to Customer Support) at no charge to Customer.

A. LIMITATIONS. C TECH does not warrant that any item of the Software is error-free or that its use will be uninterrupted. C TECH shall not be obligated to remedy any Software defect which cannot be duplicated by C TECH. These warranties do not apply to any Software which has been (i) altered, except

by C TECH or in accordance with its instructions, or (ii) used in conjunction with another vendor's Product resulting in the defect, or (iii) damaged by improper electrical power or environment abuse, misuse, accident or negligence. Services which C TECH reasonably determines to be outside the scope of these warranties will, at C TECH's option, be furnished in accordance with this Agreement. Because the Software is inherently complex and may not be completely free from errors, Customer is advised to validate its work. THE FOREGOING WARRANTIES ARE EXCLUSIVE REMEDIES AND ARE IN LIEU OF ALL OTHER REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. No statement, including without limitation, representations regarding capacity, suitability for use or performance of Software, whether made by C TECH employees or otherwise, shall be deemed to be a warranty by C TECH for any purpose or give rise to any liability of C TECH unless expressly contained in this Agreement.

7. SOFTWARE LICENSE: C TECH grants the Customer a non-exclusive license to use the Software provided that the Customer does not (i) modify any portion of the Software without C TECH’s express, written permission, (ii) translate, decompile, reassemble, or reverse engineer the Software, (iii) create derivative works of the Software, or (iv) tamper with, bypass, or otherwise interfere with the floating license manager and/or software key copy protection system. You have acquired C TECH’s Earth Science Software EVS/MVS that includes software licensed by C TECH from C TECH and its software suppliers. Those installed software products from C TECH and its software suppliers’ origin, as well as any associated media, printed materials, and "online" or electronic documentation ("SOFTWARE") are protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold. The Customer shall not duplicate any portion of the Software except for backup or archival purposes, and shall duplicate C TECH’s proprietary rights and notices on each such copy.

A. MRSID SOFTWARE: MrSID software is protected by United States Copyright Law and International Treaty provisions and by U.S. Patent No. 5,710,835. Foreign patents are pending. Some of the MrSID technology was developed through a project at the Los Alamos National Laboratory (LANL) funded by the U.S. Government, managed under contract by the Regents of the University of California (University). The U.S. Government and the University have reserved rights in the Technology, including the following: (a) the U.S. Government has a non-exclusive, nontransferable, irrevocable, paid-up license to practice or have practiced throughout the world, for or on behalf of the United States, inventions covered by the University's Patent Rights, and has other rights under 35 U.S.C. § 200-212 and applicable implementing regulations and under the U.S. Department of Energy (DOE) Assignment and Confirmatory License through which the DOE's rights in the Technology were assigned to the University; (b) Under 35 U.S.C. § 203, the DOE has the right to require LizardTech to grant a non-exclusive, partially exclusive or exclusive license under U.S. Patent No. 5,710,835 in any field of use to a responsible applicant(s) upon terms reasonable under the

circumstances, if LizardTech does not adequately attempt to commercialize the MrSID Technology. See, 37 CFR 401.6; (c) The University makes no warranty or representation as to the validity or scope of Patent No. 5,710,835, and neither the Government nor the University have any obligation to furnish any know-how, technical assistance, or technical data in connection with MrSID software. For further information about these provisions, contact LizardTech, 1008 Western Ave., Suite 200, Seattle, WA 98104.

B: AGREEMENT TO EULA: If you do not agree to this End User License Agreement ("EULA"), do not use the PRODUCT. Promptly contact C TECH for instructions on return of the unused PRODUCT(S) for a refund if applicable. Any use of the SOFTWARE, including but not limited to use of the PRODUCT, will constitute your agreement to this EULA (or ratification of any previous consent).

C. GRANT OF LICENSE: You are granted a personal, nonsublicensable, nontransferable, nonexclusive license to use the SOFTWARE as integrated in the PRODUCT (as well as any associated documentation). You will not rent, sell, lease or otherwise distribute the SOFTWARE or any part of it.

D. LICENSE TYPES:

i. Fixed Licenses: Fixed Licenses may be transferred or sold, but may not be leased or rented. All License transfers must be accompanied by this License Agreement executed by the transferee and a letter from the transferor certifying the legitimacy of the transfer. University Licenses may not be used by any person in support of commercial projects not affiliated with the University.

ii. Floating Licenses: Floating Licenses may not be transferred, sold, leased, or rented, except in total as a single license to include all license seats. The Product (C Tech’s Software) may not be hosted on a publicly accessible server, nor used to offer (semi)-automated analysis or graphics services. The Licenses may not be used by any individuals who are not direct employees of the licensed organization. Use by consultants and temporary employees working outside of the licensed organization's facilities is expressly prohibited.

iii. Premier Customer Program: Premier Program licenses may not be transferred, sold, leased, or rented. The Product (C Tech’s Software) may not be hosted on a publicly accessible server, nor used to offer (semi)-automated analysis or graphics services. The Licenses may not be used by any individuals who are not direct employees or students (in the case of educational organizations) of the licensed organization. Use by consultants and temporary employees working outside of the subscribing organization's facilities is expressly prohibited. University Licenses may not be used by any person in support of commercial projects not affiliated with the University.

iv. 4DIM Player:

a. Key licensed 4DIM players are Fixed Licenses subject to the relevant terms above.

b. Customers licensed for MVS or the Premier Customer Program may freely redistribute C Tech's free unlicensed 4DIM Player to their clients.

8. CUSTOMER SUPPORT SERVICES: The Customer may order Technical Support and Software Maintenance to cover Software updates and technical support at the pricing shown on our website.

C Tech's Premium Maintenance provides unlimited software updates and unlimited technical support through C Tech's web Support Forums and the Submit Support Request form for issues which must be kept private. Customers may also e-mail requests to support@ctech.com and may telephone (808) 447-9751 for their support issues. Premium Maintenance also allows users to submit problematic data files for review without our imposing consulting fees. Data file review is limited to identification of file defects and does not include having C Tech consult on optimal settings for modules or applications to best employ the data.

NOTE: Premium maintenance is included for the first year with the purchase of all new products.

A. TERMS OF COVERAGE: Technical Support and Software Maintenance must be paid for and current before services will be rendered. Support will be provided Monday through Friday (holidays excluded) during C TECH's regular office hours and may be available at other times.

B. C TECH's DUTIES: C TECH will provide assistance in problem resolution for supported Products, and make reasonable efforts to provide work around and/or corrections for identified bugs in the Software for all customers whose support services contract is up to date.

Note: Work with Customer's data files (examination and review) will be performed under Standard Maintenance if the file follows the specifications in the Product documentation and is being mishandled by the Software. In all cases involving operator error or improperly formatted data files, such work shall be billed as consulting services at C Tech's then standard rates.

C. SOFTWARE MAINTENANCE, UPGRADES AND TECHNICAL SUPPORT:

i. Fixed and Floating licenses: Software updates and technical support (Standard Maintenance) for a period of 12 months from first delivery of the Product is included with purchase. Leases or rentals that are upgraded to purchase have an anniversary date that is the date of first delivery under the rental or lease. The cost for renewal of annual software maintenance, upgrades and technical support is published on the C Tech Price List at www.ctech.com.

ii. Premier Customer Program: Software updates and priority response technical support (Premium Maintenance) is included in the subscription fees.

D. REPLACEMENT OF SOFTWARE KEYS:

i. Fixed and Floating licenses: Prior to validation, lost software keys are subject to a $100 USD replacement charge. After validation, lost or stolen keys may be subject to a replacement charge of 20% of initial purchase price and require certification from Customer that the license key is lost or stolen. Malfunctioning keys handled in accordance with the installation instructions will be replaced at no cost during the first year and are subject to a $100

USD replacement charge thereafter. Replacement costs do not include applicable shipping charges which are extra.

ii. Premier Customer Program: Lost software keys are subject to a $200 USD replacement charge. Malfunctioning keys handled in accordance with the installation instructions will be replaced at no cost during the first year and are subject to a $100 USD replacement charge thereafter. Replacement costs do not include applicable shipping charges which are extra.

E. MAINTENANCE RENEWAL AND LAPSED LICENSES

i. Maintenance renewal: Prior to expiration of the software maintenance period, maintenance can be renewed at the current pricing. Maintenance payments not received within 120 days of the original due date will result in the associated license being classified as lapsed. Lapsed licenses can be reinstated at a cost of 50% of the current list price of the software provided that the license has not been lapsed more than 2 years. While any license is lapsed, the following risks are accepted by the client:

1. C Tech shall have no obligation to provide any technical support

2. C Tech shall have no obligation to provide access to older software version installation files or assistance.

3. If the software key is damaged or the software fails to function for any reason whatsoever, C Tech shall have no obligation to replace the key or software at less than the current full list price.

9. INTERNET-BASED SERVICES. C Tech incorporates Internet-based services into its software. C Tech may revise or cancel these services and features at any time.

A. Consent for Internet-Based Services. The software features described below connect to C Tech or its service provider's computer systems over the Internet. In general, you will not receive a notice when a connection is made. In some cases, you may switch off these features or not use them. For more information about these features, see the software documentation. By using these features, you consent to the transmission of this information. C Tech does not use the information to identify you.

i. Computer Information. The following features use Internet protocols, which send computer information, such as your Internet protocol address, the type of operating system, hardware details, browser and name and version of the software you are using. C Tech uses this information to make the Internet-based services available to you.

1. Customer Experience Improvement Program (CEIP). This software uses CEIP. CEIP automatically sends C Tech information about your hardware and how you use this software. We do not use this information to identify you.

2. Error Reports. This software automatically sends error reports to C Tech. These reports include information about problems that occur in the software. Sometimes reports contain information about other programs that interact with the software. C Tech does not use this information to identify you.

ii. Use of Information. We may use the computer information, error reports, and CEIP information, to improve our software and services. We may also

share it with others, such as hardware and software vendors. They may use the information to improve how their products run with C Tech software.

10. PROPRIETARY DATA. This Agreement does not constitute a Non-Disclosure Agreement (NDA) binding C TECH, or its employees. Although it is C Tech's standard policy to not disclose Customer data to a third party, if Customer intends to provide sensitive data files or other or proprietary data to C TECH during the course of technical support or consulting, Customer is advised to first obtain an executed NDA with C TECH. Neither this Agreement nor such NDA shall bind C Tech to exclusivity unless specifically agreed to in writing.

11. SOFTWARE UPDATES: As part of evolutionary development, C TECH may, at its sole discretion, provide new releases of the Software to those Customers with current software update agreements. C TECH reserves the right to set new fees for new functionality that might be offered as a part of such new releases of the Software. Software (license) transfer to a different computer architecture or operating system may not be considered an upgrade and may be subject to additional fees.

12. CLAIMS OF INFRINGEMENT:

A. INDEMNIFICATION. C TECH will defend or settle at its own expense any action brought against Customer, to the extent that it is based on a claim that the Software infringes a United States or Canadian patent or copyright, and will pay any costs and damages finally awarded against Customer in any such action which are attributable to any such claim. C TECH's obligation under the preceding sentence is subject to the conditions that (i) Customer promptly notifies C TECH in writing of any such claim and reasonably cooperate with C TECH in its defense (ii) C TECH will have sole authority and control of such defense and all negotiations for any settlement or compromise, and (iii) should the Software become, or in C TECH's opinion be likely to become, the subject of any such claim, Customer will permit C TECH, at C TECH's option and expense, to procure for Customer the right to continue using the Software, to replace or modify it so that it becomes non-infringing, or to grant Customer a credit for the Software as depreciated on a three-year, straight-line basis, and accept its return.

B. LIMITATIONS. C TECH shall have no liability to the Customer with respect to any claim of infringement which is based upon or results from (i) the combination of any Product with any machine, device, firmware or software not furnished by C TECH, (ii) any modification of the Software by a party other than C TECH, (iii) Customer's failure to install changes or updates as instructed by C TECH, or (iv) use of the Software as part of any infringing process. THIS SECTION STATES THE ENTIRE LIABILITY OF C TECH WITH RESPECT TO INFRINGEMENT CLAIMS CONCERNING THE SOFTWARE OR ITS USE OR OPERATION.

13. TITLE, PROPRIETARY RIGHTS. C TECH, on behalf of itself and its licensor, retains title to, and all copyright, patent, trade secret and proprietary rights in the Software, including without limitation (i) all designs, engineering details, and other data pertaining to the Software, and (ii) all original works, computer programs, discoveries, inventions, copyrights, trade secret rights,

patents, know-how and techniques pertaining to the Software or arising out of work done wholly or in part by C TECH in connection with this Agreement. Nothing in this Agreement constitutes a waiver of C TECH's rights under copyright or patent laws, or any other federal or state law.

14. TRAINING: C TECH's Software includes workbooks and tutorials sufficient for intermediate level self-taught training. All other training that may be offered by C TECH, including advanced training courses, User Conferences and on-line training shall be at C TECH's then-prevailing rates on the C Tech Price List at www.ctech.com. The time and location of training courses will be as specified by C TECH. The Customer is responsible for all expenses of the Customer's personnel while attending training courses.

15. LIMITATION OF LIABILITY. In no event will C TECH be liable for (i) the cost of substitute procurement, special, indirect, incidental, or consequential damages, (ii) any damages resulting from inaccurate or lost data or loss of use or profits arising out of or in connection with this Agreement or the use or performance of Software, or (iii) any defects or errors in the Software, except as provided in Section 6. In no event will C TECH's total liability for (i) any damages in any action based on or arising out of or in connection with this Agreement exceed the total amount paid to C TECH pursuant to this Agreement, or (ii) claims based upon C TECH's obligations under the customer support services provisions exceed the total amount paid to C TECH for customer support services in the previous year.

16. TERMINATION: This Agreement may be terminated under the following terms:

A. Either party may terminate this Agreement and any License granted hereunder at any time upon written notice if the other party fails to comply with any material term or condition. The party not in compliance shall have 30 days (seven days in the case of late payment) after written notice to cure any failure and avoid termination.

B. Either party may terminate this Agreement and any License granted hereunder if the other party becomes insolvent, or has filed against it a petition under any bankruptcy code (or any similar petition under any insolvency law of any jurisdiction), proposes any dissolution, liquidation, composition, financial reorganization or recapitalization with creditors, makes assignment or trust mortgage for the benefit of creditors, or if a receiver, trustee, custodian or similar agent is appointed or takes possession with respect to any property or business of such other party.

C. Upon termination of any license, Customer shall return all hardware to C TECH and return or destroy all copies of Software covered by such license and shall furnish to C TECH a certificate of compliance with this provision signed by an officer of Customer. Notwithstanding any termination, a party's pre-termination obligations shall not be affected by any termination of this Agreement, and the provision of Sections 1, 2, 6A, 10, 13, 15 through 19 shall survive termination of this Agreement.

17. U. S. GOVERNMENT RESTRICTED RIGHTS:

Use, duplication or disclosure of Software by the Government is subject to restrictions set forth in the Commercial Computer Software clause at DFARS 252.227-7202.3 and Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable..

18. CONTRACT INTEGRITY: If any provision of this Agreement or License is found void or unenforceable, the remaining provisions of this License shall remain in full force and effect. If any remedy provided is determined to have failed of its essential purpose, all limitations of liability and exclusions of damages set forth in the limited warranty shall remain in effect.

19. ORDER OF PRECEDENCE: If there is a conflict among the terms and conditions in the documents provided by Customer and C TECH, the descending order of precedence shall be as follows: (1) This License Agreement, (2) C TECH's Organizational Staff Level Certification for Premier Program License (if applicable), (3) C TECH's University Certification(if applicable), (4) Customer's Purchase Order and associated Terms and Conditions.

20. GENERAL: Any notices given under this Agreement shall be in writing and sent to the address set forth on the face page of this Agreement or to such other address as shall have been substituted by written notice. C TECH shall not be liable for failures or delays in the performance of its obligations under this Agreement due to strikes, wars, revolutions, fires, floods, explosions, earthquakes, government regulations, or other causes beyond C TECH’s control. This Agreement is made subject to all laws, regulations, orders or other restrictions on the export from the U.S.A., or re-export of Software or information derived from the Software, which may be imposed from time to time. Customer may not assign this Agreement without prior written permission from C TECH. Any attempt by Customer to assign any right, or delegate any duty or obligation which arises under this Agreement, without such permission will be void. All waivers, amendments or modifications of any right, remedy or other term under this Agreement will not be effective unless in writing and signed by the party against whom enforcement is sought. This Agreement is governed by the laws of the State of Nevada, excluding its conflict of laws rules. The parties acknowledge that unmodified agreements do not require execution by C TECH. This Agreement shall only be executed by Customer’s representatives who are authorized to legally bind the organization.

Customer Organization: _____

By: _____

(Signature of Authorized Individual)

Name: _____ Title: _____

Date: _____

C TECH EARTH SCIENCE SOFTWARE
TERMS AND CONDITIONS OF
END USER LICENSE AGREEMENT

(version 130225)

THIS AGREEMENT SHALL GOVERN ORDERING, RENTAL, LEASING, SUBSCRIPTION, PURCHASE, LICENSING, SUPPORT, AND WARRANTY OF ALL OF C TECH'S EARTH SCIENCE SOFTWARE AND ASSOCIATED SOFTWARE TOOLS, INCLUDING THE ENVIRONMENTAL VISUALIZATION SYSTEM "EVS-PRO", MINING VISUALIZATION SYSTEM "MVS", AND FOUR-DIMENSIONAL INTERACTIVE MODEL ANIMATION PLAYER "4DIM PLAYER", HEREAFTER REFERRED TO AS SOFTWARE. THIS AGREEMENT IS BETWEEN C TECH DEVELOPMENT CORP. ("C TECH") AND THE CUSTOMER. THE AGREEMENT CONSTITUTES THE ENTIRE CONTRACT BETWEEN THE PARTIES AND SHALL GOVERN ALL ORDERS COMPLETELY UNLESS AMENDED IN WRITING. SUCH AMENDMENTS MUST BE SIGNED BY BOTH THE CUSTOMER AND AN AUTHORIZED REPRESENTATIVE OF C TECH IN ORDER TO BE CONSIDERED VALID.

C Tech Development Corporation

2360 Corporate Circle, Suite 400, Henderson, NV 89074 USA

sales@ctech.com

Phone: (800) 669-4387 Fax: (714) 844-9255

1. ORDERING: The Customer may order any item or package listed on the C Tech Price List at www.ctech.com ("Product"). C TECH does not guarantee that prices will not change, nor does it honor out-of-date price lists. No order will be effective until made in writing and signed by the Customer and an authorized representative of C TECH.

2. PAYMENT AND TAXES: Payment terms are net 10 days except international orders that require advance payment. An authorized representative of C TECH must approve any revisions to C TECH's terms or license agreement in writing. Customer is responsible for all shipping charges, taxes, fees, duties, and any or all government charges that are customarily the responsibility of the Customer.

3. SHIPMENT AND DELIVERY: C TECH will ship the Product F.O.B. Destination, Prepaid and Added by means of courier service (e.g. FedEx) requiring Customer's signature for receipt of delivery. The shipping charges will be paid by the shipping party and added to the invoice to the purchaser over and above the items purchased costs. In the event of loss or damage during shipment, the Customer agrees to assist C TECH in filing freight claims for loss or damage.

4. ACCEPTANCE & VALIDATON: Upon receipt of the Product (software key for licensing), the Customer must notify C TECH within ten (10) calendar days that the Product is not performing correctly, or the Customer acknowledges acceptance of the Product. Once the customer notifies C Tech that they have received the software, and C Tech verifies receipt of full payment for either

rentals or purchases, C Tech will e-mail a Key Update File (.KUF). Use of the Key Update File by Customer constitutes Acceptance of the software and Validation of the software key.

5. THIRTY DAY PURCHASE GUARANTEE: C TECH guarantees Customer's satisfaction with the Product for a period of thirty (30) days, beginning on the day of delivery. At any time during this thirty-day period, the Customer may contact C TECH, obtain a return authorization, and return the Product. Upon receipt of the entire Product, C TECH will refund the net purchase prices paid less 10% and any shipping and handling fees on the original invoice. Prior to acceptance, the Product may be returned for a full refund less shipping and handling costs. However, THE PRODUCT MUST BE SHIPPED BEFORE THE END OF THE THIRTY-DAY PERIOD TO QUALIFY FOR A REFUND.

6. SOFTWARE WARRANTY: C TECH guarantees that the Product (Software), as delivered or updated by C TECH and properly installed and operated, will perform substantially as described in its then-current documentation for 90 days from the date of delivery. If the Software fails to so perform during such period, the Customer shall promptly notify C TECH of, and adequately describe, the failure, and Customer's exclusive remedy and C TECH's sole obligation shall be to take corrective action as specified in Section 8.B (relating to Customer Support) at no charge to Customer.

A. LIMITATIONS. C TECH does not warrant that any item of the Software is error-free or that its use will be uninterrupted. C TECH shall not be obligated to remedy any Software defect which cannot be duplicated by C TECH. These warranties do not apply to any Software which has been (i) altered, except by C TECH or in accordance with its instructions, or (ii) used in conjunction with another vendor's Product resulting in the defect, or (iii) damaged by improper electrical power or environment abuse, misuse, accident or negligence. Services which C TECH reasonably determines to be outside the scope of these warranties will, at C TECH's option, be furnished in accordance with this Agreement. Because the Software is inherently complex and may not be completely free from errors, Customer is advised to validate its work. THE FOREGOING WARRANTIES ARE EXCLUSIVE REMEDIES AND ARE IN LIEU OF ALL OTHER REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. No statement, including without limitation, representations regarding capacity, suitability for use or performance of Software, whether made by C TECH employees or otherwise, shall be deemed to be a warranty by C TECH for any purpose or give rise to any liability of C TECH unless expressly contained in this Agreement.

7. SOFTWARE LICENSE: C TECH grants the Customer a non-exclusive license to use the Software provided that the Customer does not (i) modify any portion of the Software without C TECH's express, written permission, (ii) translate, decompile, reassemble, or reverse engineer the Software, (iii) create derivative works of the Software, or (iv) tamper with, bypass, or otherwise interfere with the floating license manager and/or software key copy protection system. You have acquired C TECH's Earth Science Software EVS/MVS that includes software licensed by C TECH from C TECH and its

software suppliers. Those installed software products from C TECH and its software suppliers' origin, as well as any associated media, printed materials, and "online" or electronic documentation ("SOFTWARE") are protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold. The Customer shall not duplicate any portion of the Software except for backup or archival purposes, and shall duplicate C TECH's proprietary rights and notices on each such copy.

A. MRSID SOFTWARE: MrSID software is protected by United States Copyright Law and International Treaty provisions and by U.S. Patent No. 5,710,835. Foreign patents are pending. Some of the MrSID technology was developed through a project at the Los Alamos National Laboratory (LANL) funded by the U.S. Government, managed under contract by the Regents of the University of California (University). The U.S. Government and the University have reserved rights in the Technology, including the following: (a) the U.S. Government has a non-exclusive, nontransferable, irrevocable, paid-up license to practice or have practiced throughout the world, for or on behalf of the United States, inventions covered by the University's Patent Rights, and has other rights under 35 U.S.C. § 200-212 and applicable implementing regulations and under the U.S. Department of Energy (DOE) Assignment and Confirmatory License through which the DOE's rights in the Technology were assigned to the University; (b) Under 35 U.S.C. § 203, the DOE has the right to require LizardTech to grant a non-exclusive, partially exclusive or exclusive license under U.S. Patent No. 5,710,835 in any field of use to a responsible applicant(s) upon terms reasonable under the circumstances, if LizardTech does not adequately attempt to commercialize the MrSID Technology. See, 37 CFR 401.6; (c) The University makes no warranty or representation as to the validity or scope of Patent No. 5,710,835, and neither the Government nor the University have any obligation to furnish any know-how, technical assistance, or technical data in connection with MrSID software. For further information about these provisions, contact LizardTech, 1008 Western Ave., Suite 200, Seattle, WA 98104.

B: AGREEMENT TO EULA: If you do not agree to this End User License Agreement ("EULA"), do not use the PRODUCT. Promptly contact C TECH for instructions on return of the unused PRODUCT(S) for a refund if applicable. Any use of the SOFTWARE, including but not limited to use of the PRODUCT, will constitute your agreement to this EULA (or ratification of any previous consent).

C. GRANT OF LICENSE: You are granted a personal, nonsublicensable, nontransferable, nonexclusive license to use the SOFTWARE as integrated in the PRODUCT (as well as any associated documentation). You will not rent, sell, lease or otherwise distribute the SOFTWARE or any part of it.

D. LICENSE TYPES:

i. Fixed Licenses: Fixed Licenses may be transferred or sold, but may not be leased or rented. All License transfers must be accompanied by this License Agreement executed by the transferee and a letter from the transferor

certifying the legitimacy of the transfer. University Licenses may not be used by any person in support of commercial projects not affiliated with the University.

ii. Floating Licenses: Floating Licenses may not be transferred, sold, leased, or rented, except in total as a single license to include all license seats. The Product (C Tech's Software) may not be hosted on a publicly accessible server, nor used to offer (semi)-automated analysis or graphics services. The Licenses may not be used by any individuals who are not direct employees of the licensed organization. Use by consultants and temporary employees working outside of the licensed organization's facilities is expressly prohibited.

iii. Premier Customer Program: Premier Program licenses may not be transferred, sold, leased, or rented. The Product (C Tech's Software) may not be hosted on a publicly accessible server, nor used to offer (semi)-automated analysis or graphics services. The Licenses may not be used by any individuals who are not direct employees or students (in the case of educational organizations) of the licensed organization. Use by consultants and temporary employees working outside of the subscribing organization's facilities is expressly prohibited. University Licenses may not be used by any person in support of commercial projects not affiliated with the University.

iv. 4DIM Player:

a. Key licensed 4DIM players are Fixed Licenses subject to the relevant terms above.

b. Customers licensed for MVS or the Premier Customer Program may freely redistribute C Tech's free unlicensed 4DIM Player to their clients.

8. CUSTOMER SUPPORT SERVICES: The Customer may order Technical Support and Software Maintenance to cover Software updates and technical support at the pricing shown on our website.

C Tech's Premium Maintenance provides unlimited software updates and unlimited technical support through C Tech's web Support Forums and the Submit Support Request form for issues which must be kept private. Customers may also e-mail requests to support@ctech.com and may telephone (808) 447-9751 for their support issues. Premium Maintenance also allows users to submit problematic data files for review without our imposing consulting fees. Data file review is limited to identification of file defects and does not include having C Tech consult on optimal settings for modules or applications to best employ the data.

NOTE: Premium maintenance is included for the first year with the purchase of all new products.

A. TERMS OF COVERAGE: Technical Support and Software Maintenance must be paid for and current before services will be rendered. Support will be provided Monday through Friday (holidays excluded) during C TECH's regular office hours and may be available at other times.

B. C TECH's DUTIES: C TECH will provide assistance in problem resolution for supported Products, and make reasonable efforts to provide work around

and/or corrections for identified bugs in the Software for all customers whose support services contract is up to date.

Note: Work with Customer's data files (examination and review) will be performed under Standard Maintenance if the file follows the specifications in the Product documentation and is being mishandled by the Software. In all cases involving operator error or improperly formatted data files, such work shall be billed as consulting services at C Tech's then standard rates.

C. SOFTWARE MAINTENANCE, UPGRADES AND TECHNICAL SUPPORT:

i. Fixed and Floating licenses: Software updates and technical support (Standard Maintenance) for a period of 12 months from first delivery of the Product is included with purchase. Leases or rentals that are upgraded to purchase have an anniversary date that is the date of first delivery under the rental or lease. The cost for renewal of annual software maintenance, upgrades and technical support is published on the C Tech Price List at www.ctech.com.

ii. Premier Customer Program: Software updates and priority response technical support (Premium Maintenance) is included in the subscription fees.

D. REPLACEMENT OF SOFTWARE KEYS:

i. Fixed and Floating licenses: Prior to validation, lost software keys are subject to a $100 USD replacement charge. After validation, lost or stolen keys may be subject to a replacement charge of 20% of initial purchase price and require certification from Customer that the license key is lost or stolen. Malfunctioning keys handled in accordance with the installation instructions will be replaced at no cost during the first year and are subject to a $100 USD replacement charge thereafter. Replacement costs do not include applicable shipping charges which are extra.

ii. Premier Customer Program: Lost software keys are subject to a $200 USD replacement charge. Malfunctioning keys handled in accordance with the installation instructions will be replaced at no cost during the first year and are subject to a $100 USD replacement charge thereafter. Replacement costs do not include applicable shipping charges which are extra.

E. MAINTENANCE RENEWAL AND LAPSED LICENSES

i. Maintenance renewal: Prior to expiration of the software maintenance period, maintenance can be renewed at the current pricing. Maintenance payments not received within 120 days of the original due date will result in the associated license being classified as lapsed. Lapsed licenses can be reinstated at a cost of 50% of the current list price of the software provided that the license has not been lapsed more than 2 years. While any license is lapsed, the following risks are accepted by the client:

1. C Tech shall have no obligation to provide any technical support

2. C Tech shall have no obligation to provide access to older software version installation files or assistance.

3. If the software key is damaged or the software fails to function for any reason whatsoever, C Tech shall have no obligation to replace the key or software at less than the current full list price.

9. INTERNET-BASED SERVICES. C Tech incorporates Internet-based services into its software. C Tech may revise or cancel these services and features at any time.

A. Consent for Internet-Based Services. The software features described below connect to C Tech or its service provider's computer systems over the Internet. In general, you will not receive a notice when a connection is made. In some cases, you may switch off these features or not use them. For more information about these features, see the software documentation. By using these features, you consent to the transmission of this information. C Tech does not use the information to identify you.

i. Computer Information. The following features use Internet protocols, which send computer information, such as your Internet protocol address, the type of operating system, hardware details, browser and name and version of the software you are using. C Tech uses this information to make the Internet-based services available to you.

1. Customer Experience Improvement Program (CEIP). This software uses CEIP. CEIP automatically sends C Tech information about your hardware and how you use this software. We do not use this information to identify you.

2. Error Reports. This software automatically sends error reports to C Tech. These reports include information about problems that occur in the software. Sometimes reports contain information about other programs that interact with the software. C Tech does not use this information to identify you.

ii. Use of Information. We may use the computer information, error reports, and CEIP information, to improve our software and services. We may also share it with others, such as hardware and software vendors. They may use the information to improve how their products run with C Tech software.

10. PROPRIETARY DATA. This Agreement does not constitute a Non-Disclosure Agreement (NDA) binding C TECH, or its employees. Although it is C Tech’s standard policy to not disclose Customer data to a third party, if Customer intends to provide sensitive data files or other or proprietary data to C TECH during the course of technical support or consulting, Customer is advised to first obtain an executed NDA with C TECH. Neither this Agreement nor such NDA shall bind C Tech to exclusivity unless specifically agreed to in writing.

11. SOFTWARE UPDATES: As part of evolutionary development, C TECH may, at its sole discretion, provide new releases of the Software to those Customers with current software update agreements. C TECH reserves the right to set new fees for new functionality that might be offered as a part of such new releases of the Software. Software (license) transfer to a different computer architecture or operating system may not be considered an upgrade and may be subject to additional fees.

12. CLAIMS OF INFRINGEMENT:

A. INDEMNIFICATION. C TECH will defend or settle at its own expense any action brought against Customer, to the extent that it is based on a claim that the Software infringes a United States or Canadian patent or copyright, and will pay any costs and damages finally awarded against Customer in any

such action which are attributable to any such claim. C TECH's obligation under the preceding sentence is subject to the conditions that (i) Customer promptly notifies C TECH in writing of any such claim and reasonably cooperate with C TECH in its defense (ii) C TECH will have sole authority and control of such defense and all negotiations for any settlement or compromise, and (iii) should the Software become, or in C TECH's opinion be likely to become, the subject of any such claim, Customer will permit C TECH, at C TECH's option and expense, to procure for Customer the right to continue using the Software, to replace or modify it so that it becomes non-infringing, or to grant Customer a credit for the Software as depreciated on a three-year, straight-line basis, and accept its return.

B. LIMITATIONS. C TECH shall have no liability to the Customer with respect to any claim of infringement which is based upon or results from (i) the combination of any Product with any machine, device, firmware or software not furnished by C TECH, (ii) any modification of the Software by a party other than C TECH, (iii) Customer's failure to install changes or updates as instructed by C TECH, or (iv) use of the Software as part of any infringing process. THIS SECTION STATES THE ENTIRE LIABILITY OF C TECH WITH RESPECT TO INFRINGEMENT CLAIMS CONCERNING THE SOFTWARE OR ITS USE OR OPERATION.

13. TITLE, PROPRIETARY RIGHTS. C TECH, on behalf of itself and its licensor, retains title to, and all copyright, patent, trade secret and proprietary rights in the Software, including without limitation (i) all designs, engineering details, and other data pertaining to the Software, and (ii) all original works, computer programs, discoveries, inventions, copyrights, trade secret rights, patents, know-how and techniques pertaining to the Software or arising out of work done wholly or in part by C TECH in connection with this Agreement. Nothing in this Agreement constitutes a waiver of C TECH's rights under copyright or patent laws, or any other federal or state law.

14. TRAINING: C TECH's Software includes workbooks and tutorials sufficient for intermediate level self-taught training. All other training that may be offered by C TECH, including advanced training courses, User Conferences and on-line training shall be at C TECH's then-prevailing rates on the C Tech Price List at www.ctech.com. The time and location of training courses will be as specified by C TECH. The Customer is responsible for all expenses of the Customer's personnel while attending training courses.

15. LIMITATION OF LIABILITY. In no event will C TECH be liable for (i) the cost of substitute procurement, special, indirect, incidental, or consequential damages, (ii) any damages resulting from inaccurate or lost data or loss of use or profits arising out of or in connection with this Agreement or the use or performance of Software, or (iii) any defects or errors in the Software, except as provided in Section 6. In no event will C TECH's total liability for (i) any damages in any action based on or arising out of or in connection with this Agreement exceed the total amount paid to C TECH pursuant to this Agreement, or (ii) claims based upon C TECH's obligations under the customer support services provisions exceed the total amount paid to C TECH for customer support services in the previous year.

16. TERMINATION: This Agreement may be terminated under the following terms:

A. Either party may terminate this Agreement and any License granted hereunder at any time upon written notice if the other party fails to comply with any material term or condition. The party not in compliance shall have 30 days (seven days in the case of late payment) after written notice to cure any failure and avoid termination.

B. Either party may terminate this Agreement and any License granted hereunder if the other party becomes insolvent, or has filed against it a petition under any bankruptcy code (or any similar petition under any insolvency law of any jurisdiction), proposes any dissolution, liquidation, composition, financial reorganization or recapitalization with creditors, makes assignment or trust mortgage for the benefit of creditors, or if a receiver, trustee, custodian or similar agent is appointed or takes possession with respect to any property or business of such other party.

C. Upon termination of any license, Customer shall return all hardware to C TECH and return or destroy all copies of Software covered by such license and shall furnish to C TECH a certificate of compliance with this provision signed by an officer of Customer. Notwithstanding any termination, a party's pre-termination obligations shall not be affected by any termination of this Agreement, and the provision of Sections 1, 2, 6A, 10, 13, 15 through 19 shall survive termination of this Agreement.

17. U. S. GOVERNMENT RESTRICTED RIGHTS:

Use, duplication or disclosure of Software by the Government is subject to restrictions set forth in the Commercial Computer Software clause at DFARS 252.227-7202.3 and Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable..

18. CONTRACT INTEGRITY: If any provision of this Agreement or License is found void or unenforceable, the remaining provisions of this License shall remain in full force and effect. If any remedy provided is determined to have failed of its essential purpose, all limitations of liability and exclusions of damages set forth in the limited warranty shall remain in effect.

19. ORDER OF PRECEDENCE: If there is a conflict among the terms and conditions in the documents provided by Customer and C TECH, the descending order of precedence shall be as follows: (1) This License Agreement, (2) C TECH's Organizational Staff Level Certification for Premier Program License (if applicable), (3) C TECH's University Certification(if applicable), (4) Customer's Purchase Order and associated Terms and Conditions.

20. GENERAL: Any notices given under this Agreement shall be in writing and sent to the address set forth on the face page of this Agreement or to such other address as shall have been substituted by written notice. C TECH shall not be liable for failures or delays in the performance of its obligations under this Agreement due to strikes, wars, revolutions, fires, floods, explosions, earthquakes, government regulations, or other causes beyond C TECH’s

control. This Agreement is made subject to all laws, regulations, orders or other restrictions on the export from the U.S.A., or re-export of Software or information derived from the Software, which may be imposed from time to time. Customer may not assign this Agreement without prior written permission from C TECH. Any attempt by Customer to assign any right, or delegate any duty or obligation which arises under this Agreement, without such permission will be void. All waivers, amendments or modifications of any right, remedy or other term under this Agreement will not be effective unless in writing and signed by the party against whom enforcement is sought. This Agreement is governed by the laws of the State of Nevada, excluding its conflict of laws rules. The parties acknowledge that unmodified agreements do not require execution by C TECH. This Agreement shall only be executed by Customer's representatives who are authorized to legally bind the organization.

Customer Organization: _____

By: _____
(Signature of Authorized Individual)
Name: _____ Title: _____

Date: _____

Mathematical Operators

Built In Operators:

The following is a list of all built in operators available in mathematical expressions. All words (e.g. mod or AND) are case insensitive.

Symbol	Description	Example Expression	Result
()	Prioritizes an expression	5*(1+1)	10
!	Factorial	5!	120
%	Percentage	35%	0.35
^	Raised to the power of	4^5	1024
**	Raised to the power of	4.00E+05	1024
*	Multiply by	3 * 6	18
/	Divide by	2 / 9	4.5
\	Integer divide by	9 \ 2	4

mod	Modulo (remainder)	7 mod 4	3
+	Add	1+1	2
-	Subtract	9 - 4	5
>	Greater than * see note	9 < 2	1
<	Less than	9 < 2	0
==	Equality test	5 == 4	4
>=	Greater or equal	3 >= 3	1
<=	Less than or equal	4 <= 3	0
<>	Not equal	5 <> 4	1
NOT	Logical (Bitwise) NOT	NOT(15)	-16
AND	Logical AND	13 AND 6	4
&	Logical AND	13 & 6	4
OR	Logical OR	13 OR 6	15
|	Logical OR	13 | 6	15
XOR	Logical Exclusive OR	9 XOR 3	10
EQV	Logical Equivalence	6 EQV 9	-16
IMP	Logical Implication	1 IMP 5	-1

* NOTE: All relational operators (>, <, ==, >=, <=, <>) return 0 if false and 1 if true

In general, all the modules using mathematical operations follow the general syntax rules of Visual Basic (left to right evaluation, case insensitive syntax), with the following enhancements:

Implicit Multiplication:

When multiplication is implied, the times symbol (*) can often be omitted, as in the following examples:

x y = x*y

3pi + 10 = 3*pi + 10

5(4+8) = 5*(4+8)

(5+5)(3+9) = (5+5)*(3+9)

(3+2)8 = (3+2)*8

Note: Implicit multiplication has the same priority as regular multiplication. For instance '1/2q' is translated as '1/2*q' not '1/(2q)'. This is subject to change in future versions. To avoid such ambiguity, the multiplication symbol (*) should be used explicitly as much as possible.

Numeric Bases:

Binary, octal, and hexadecimal numbers can be used in all math expressions. These numbers must be preceded by the character # followed by b, o, or h for binary, octal, or hexadecimal. Numbers may include a floating point. For example:

452 = #h1C4

452 = #o704

452 = #b111000100

#o704 = #h1C4

Order of Precedence:

When multiple operators and functions occur in a single expression, each individual part is evaluated in the following order:

Anything inside parenthesis is performed first

Factorial, percentage !, %

Exponentiation ^

Negation (unary) -

Multiplication, division *, /

Integer division \

Modulo (remainder) MOD

Addition, subtraction +, -

Relational operators <, >, >=, <=, =, <>

AND operator

OR, XOR (exclusive or)

EQV (equivalence)

IMP (implication)

When consecutive operators have the same priority, they are evaluated from left to right. This means that an expression such as "a-b-c" is evaluated as "(a-b)-c".

Functions:

The following functions can be used to perform mathematical and logical operations:

Function	Description	Example of Use	Result
IIF	If condition	IIf(1+1=2,4,5)	4
MIN(a, b, c, etc) *	Minimum value	min(10,3,27,15)	3
MAX(a, b, c, etc) *	Maximum value	max(1,9)	9
CLAMP	Clamp between two values	clamp(1,4,100)	4
INTERP *	Interpolate between values	Interp(0,10,0.3)	3
INTERPLOG *	Interpolate between log values	Interplog(0,2,.5)	1.7
SIN	Sine	sin(pi)	0
COS	Cosine	cos(pi)	-1
TAN	Tangent	tan(pi)	0
ASIN	Arc sine	asin(1)	1.570
ACOS	Arc cosine	acos(-1)	3.14159
ATAN or ATN	Arc tangent	atan(0)	0
SEC	Secant	sec(0)	1
CSC	Cosecant	csc(1)	1.18
COT	Cotangent	cot(1)	0.642
SINH	Hyperbolic sine	sinh(3)	10.01
COSH	Hyperbolic cosine	cosh(2)	3.76
TANH	Hyperbolic tangent	tanh(1)	0.76
COTH	Hyperbolic cotangent	coth(1)	1.31
SECH	Hyperbolic secant	sech(0)	1
CSCH	Hyperbolic cosecant	csch(1)	0.85
ASINH	Hyperbolic arc sine	asinh(2)	1.44
ACOSH	Hyperbolic arc cosine	acosh(9)	2.89

ATANH	**Hyperbolic arc tangent**	**atanh(.1)**	**0.10**
ACOTH	**Hyperbolic arc cotangent**	**acoth(7)**	**0.14**
ASECH	**Hyperbolic arc secant**	**asech(.3)**	**1.87**
ACSCH	**Hyperbolic arc cosecant**	**acsch(2)**	**0.48**
ABS	**Absolute value**	**abs(-8)**	**8**
POW	**Raise base to exp**	**pow(10,3)**	**1000**
EXP	**e to the power of**	**exp(3)**	**20.08**
EXP2	**2 to the power of**	**exp2(3)**	**8**
EXP10	**10 to the power of**	**exp10(3)**	**1000**
LOG or LN	**Natural log**	**log(16)**	**2.77**
LOG2	**Log base 2**	**log2(8)**	**3**
LOG10	**Log base 10**	**log10(100)**	**2**
CEIL	**Round up**	**ceil(6.2)**	**7**
ROUND	**Rounding function**	**Round(588026.07,-4)**	**590000**
		Round(588026.07,-2)	**588000**
		Round(26.07623,2)	**26.08**
RND	**Random number**	**rnd(1)**	**0.969**
INT	**Truncate to an integer**	**int(6.8)**	**6**
SGN or SIGN	**Sign of expression(-1, 0, or 1)**	**sgn(-9)**	**-1**
SQR or SQRT	**Square Root**	**sqr(64)**	**8**

NOTES:

- The MIN and MAX functions can take any number of parameters
- Interp and Interplog each take 3 parameters. The first are the 2 values to interpolate between, the third is the percentage of the first value to use. Interplog should be used if the values are log processed, since it will exponentiate the values, then interpolate, then take the base 10 logarithm of the result.

Variables:

The following variables can be used to perform mathematical and logical operations:

Pi = 3.14159265358979323

E = 2.718281828

User defined variables and functions:

The end user now can define their own functions and variables to use in mathematical expressions. C Tech has defined many conversion functions as examples, which can be found in the file **data\special\CTechFunctions.math**. You can define your own macro functions and variables to use in the file **data\special\UserFunctions.math**. See these files for examples of use. All samples provided will be available for use in any mathematical expression.

Date Operators

Date and Time Formatting Codes	Replaced by
c	Displays the date using the format given by the Windows standard Short Date Format, followed by the time using the format given by the Windows standard Long Time Format. The time is not displayed if the fractional part of the Date Time value is zero.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat).
dddd	Displays the day as a full name (Sunday-Saturday)
ddddd	Displays the date using the format given by the Short Date Format.
dddddd	Displays the date using the format given by the Long Date Format.
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec).
mmmm	Displays the month as a full name (January-December).
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).

h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (00-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
z	Displays the millisecond without a leading zero (0-999).
zzz	Displays the millisecond with a leading zero (000-999).
t	Displays the time using the format given by the Short Time Format.
tt	Displays the time using the format given by the Long Time Format.
am/pm	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
ampm	Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.
/	Displays the date separator character given by the DateSeparator.
:	Displays the time separator character given by the TimeSeparator global variable.
'xx'/"xx"	Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

Notes

Format specifiers may be written in upper case as well as in lower case letters--both produce the same result.

Examples

Code	Example Result created by the Titles module and displayed in the Viewer
"Date is "ddd mmm dd yyyy "at"	Date is Fri Dec 31 2000 at 11:00:00 PM

hh:nn:ss AM/PM	
c	12/31/2010 10:00:00 AM
dddddd	Friday, December 31, 2000

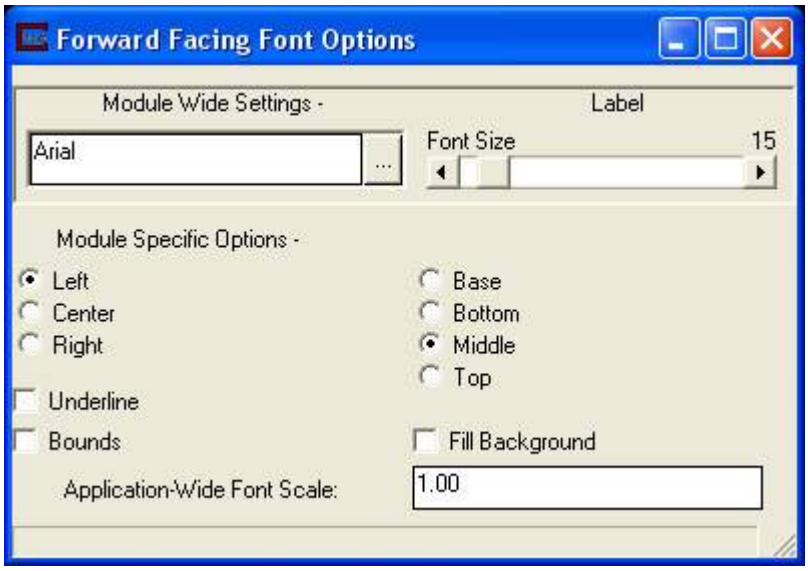
Font Selection

Many modules in EVS/MVS allow you to choose from among all Windows fonts installed on your computer. When you select the font in those modules the following window appears.

{choose-font.BMP}

Most fonts are rendered as polygonal faces. These fonts create objects that are either drawn in the 2D overlay (are not affected by zooming or rotations in the Viewer) or as 3D objects that move with your model. The exception is Forward Facing.

When Forward Facing is selected as shown above, an additional button appears to the left of "Forward Facing". Clicking on it opens another window as shown below:



Within Forward Facing Font Options you may select:

1. The Windows font.
2. The size of the Font, specified in PIXELS
3. Centering options (left, center, right, top, bottom etc.)
4. Whether the font is underlined or boxed in (Bounds)
5. Whether to "Fill Background"
6. AND and Application-Wide Font Scale.

This last parameter is very useful since forward facing fonts are drawn in pixels and are therefore of constant pixel size regardless of the viewer size (in pixels). The apparent size (relative to your screen dimensions) therefore

does change and can be compensated IN ALL MODULES by changing this parameter in any module's Forward Facing Font Options window.

Software References & Acknowledgements

1. Some routines for reading shapefiles were based on works by Frank Warmerdam.
2. Some routines for reading PNG images were based on works by Gustavo Daud (gubadaud@terra.com.br) using his "TPNGImage component".
3. The MVS module GSLIB_KT3D was based on the Stanford GSLIB KT3D program. Complete instructions for its use are in the book: GSLIB (Geostatistical Software Library and User's Guide) Second Edition by Clayton V. Deutsch and Andre G. Journel published by Oxford University Press 1998.
4. The Thin Plate Spline algorithm implementation was derived from work by David Eberly at Magic Software - http://www.magic-software.com/