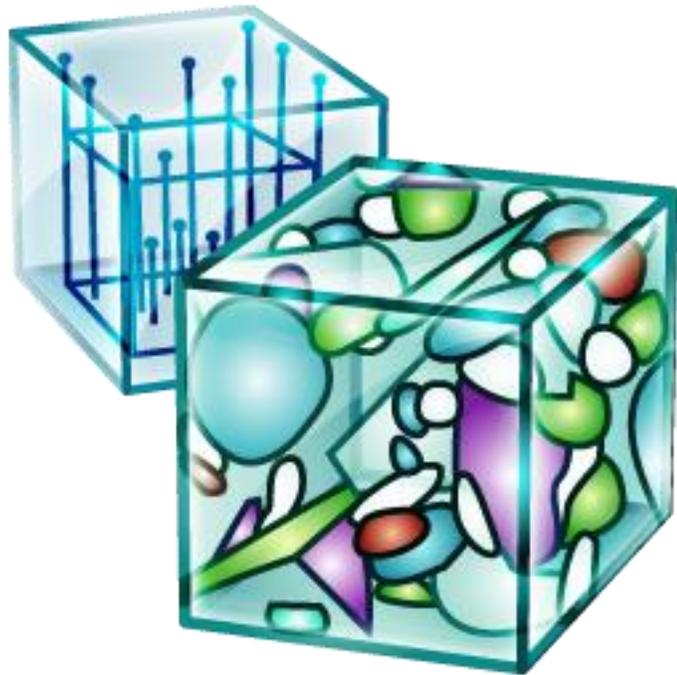


FracMan7

Interactive Discrete Feature
Data Analysis, Geometric Modeling
and Exploration Simulation



GOLDER

March 28, 2018

Table of Contents

TABLE OF CONTENTS	1
ABOUT THIS DOCUMENT	1
WELCOME TO THE WORLD OF DISCRETE FRACTURE NETWORKS	1
INSTALLING FRACMAN	2
SYSTEM REQUIREMENTS	2
INSTALLING AND INITIALIZING FRACMAN.....	3
LICENSE ACTIVATION.....	3
UNINSTALLING FRACMAN.....	5
VERSION CHECK	5
GETTING STARTED	5
COMPONENTS OF A RESERVOIR MODEL	7
POLES, DIP VECTORS, TRENDS, AND PLUNGES.....	8
USER INTERFACE	8
CONTROL WINDOWS	12
TOOL BAR	20
MENU BAR.....	22
FILE MENU.....	22
EDIT MENU	29
VIEW MENU.....	43
FRACTURE MENU	47
INSERT MENU.....	62
DATA ANALYSIS MENU.....	79
STATIC ANALYSIS MENU	79
DYNAMIC ANALYSIS MENU	81
TOOLS MENU	81
THE WINDOW MENU	100
THE HELP MENU.....	100
BUILDING YOUR MODEL	101
DISTRIBUTIONS	101
INTERACTING WITH OBJECTS	104
OBJECT PROPERTIES.....	109
REGIONS.....	127
SURFACES	127
GRIDS.....	141
WELLS.....	159
WELL LOGS	173

TRACEMAPS	185
POINT DATA OBJECT	191
CONVERTING OBJECT UNITS.....	196
HANDLING UTM COORDINATES	196
GENERATING FRACTURES.....	197
THE THEORY BEHIND FRACTURE GENERATION.....	197
GEOMETRIC.....	219
GEOCELLULAR.....	227
GEOLOGIC.....	240
TRACEMAP FRACTURE SET	252
STRATIGRAPHIC.....	255
LAYERED FRACTURE SPACING MODEL	259
STRESS MODEL	268
CREATE FROM POINT DATA.....	270
CONDITION TO TRACES	273
CREATE PSEUDO MATRIX FRACTURE.....	273
WORKING WITH CHARTS.....	273
HISTOGRAM	273
CUMULATIVE AND COMPLEMENTARY DISTRIBUTION FUNCTION PLOT	277
POWER LAW PLOT.....	278
CUMULATIVE FRACTURE INTENSITY PLOT	279
STEREONET	282
PROPERTY STEREONET	296
EXPORTING YOUR CHART.....	296
DATA ANALYSIS	297
SET IDENTIFICATION.....	297
FLOW CAPACITY (OXFILET) ANALYSIS	305
SPATIAL ANALYSIS	311
TRACE LENGTH FRACTURE SIZE	315
COMPUTING LINEAR INTENSITY (P10)	319
COMPUTING TRACE INTENSITY (P21/P31).....	321
INTENSITY CONVERSION (C31) COMPUTATION	322
WELL LOG REPORT	327
STATISTICS	328
STATIC ANALYSIS	329
FRACTURE-BASED ANALYSES	329
GRID-BASED ANALYSES	344
WELL-BASED ANALYSES	365

SURFACE-BASED ANALYSES	370
VOLUMETRIC-BASED ANALYSES	372
GEOMECHANICS.....	377
STATISTICS	467
STEREONET	467
PROPERTY STEREONET	467
CREATE MULTIPLE STEREONETS	468
PROPERTY CORRELATION GRAPH	469
BASIC GEOSTATISTICS MAPPING	470
DYNAMIC FRACTURE ANALYSIS	471
TRANSIENT WELL TEST	472
MAFIC FLOW SIMULATION	493
FRACTURE UPSCALING.....	516
HYDROMECHANICAL COUPLING.....	516
USING MACROS	519
INTRODUCTION TO MACROS	519
RECORDING A MACRO.....	520
EDITING A MACRO.....	520
LOOPS AND MONTE CARLO SIMULATION	522
PLAYBACK A MACRO	522
IGNORE MACRO ERRORS	523
MACRO LANGUAGE SYNTAX.....	523
APPENDIX A FRACMAN SUPPORTED FILE FORMATS	525
MAFIC MESH FILE	527
LEGACY MAFICOIL FRACTURES	527
GENERIC FRACTURE DATA.....	527
TOMOGRAPHIC FRACTURE IMAGES – TFI	527
PETREL WELL TRACE	527
GOCAD ASCII FILES	528
LOG ASCII STANDARD (LAS).....	529
ORS FILES.....	529
WELL INTERVAL (TOPS) FILES.....	530
WELL INTERVALS TABLE FILES	530
WELL COLLAR / DEVIATION FILE	530
ASCII WELL FRACTURE FILE	531
ECLIPSE ASCII GRID.....	531
STRESS FIELD FILE.....	533
3DMOVE GRID.....	533
CENTERPOINT SIZE GRID.....	533

ELFEN GRID	533
FLAC3D GRID	533
GSLIB PROPERTY FILE	534
RESCUE GRID	534
MODFLOW DISCRETIZED GRID	534
MODFLOW LAYER PROPERTY FILE.....	535
MODFLOW ZONE NUMBERS FILE	535
FEFLOW GRID.....	535
Z-MAP PLUS	535
DXF SURFACE.....	535
RMS SURFACE.....	535
CPS-3 SURFACE.....	536
EARTH VISION SURFACE	537
PETREL SURFACE	537
CHARISMA FAULT STICK.....	537
POLYGON FILE	538
WAVEFRONT OBJ	538
GiD ASCII MESH.....	539
ESRI SHAPEFILE	540
MAPTEK VULCAN	540
SURPAC DIGITAL TERRAIN / STRING FILES.....	540
FRACMAN TRACEDATA FILES.....	541
CPS-3 LINES.....	541
IRAP CLASSIC LINES.....	541
MAFIC PARTICLE TRACK	541
SEGY FORMAT DATA.....	541
APPENDIX B TROUBLESHOOTING USB SECURITY KEY ISSUES	542
APPENDIX C FAB FRACTURE FILE FORMAT	542
APPENDIX D FEFLOW INTERFACE.....	548
FEM IMPORT FROM FEFLOW	548
PROPERTY EXPORT WITH FEFLOW PLUGIN	548
APPENDIX E STRESS FIELD AND FAULT GRID FILE FORMAT.....	549
BASICS.....	549
INTENSITY PARAMETERS	550
LOCAL COORDINATE SYSTEMS	550
GRID STRUCTURE	551
STRESS REGION DEFINITION	552

APPENDIX F FILE FORMATS FOR WELL TEST SIMULATION.....	554
APPENDIX G FILE FORMATS FOR FLOW SIMULATION	554
PARAMETER CONTROL FILE (WELLTEST SIMULATION)	554
PARAMETER CONTROL FILE (PERMEABILITY CALCULATION)	554
WELL CONTROL FILE	555
PERMEABILITY CALCULATION CONTROL FILE	555
UNITS CONTROL FILE	556
MESH FILE	557
WELL PRESSURE RESULTS FILE	558
WELL FRACTURE FLOW RESULTS FILE	558
APPENDIX H THIRD PARTY SOFTWARE.....	559
GEOMETRIC TOOLS (WILD MAGIC)	559
HARU	559
HDF5 (HIERARCHICAL DATA FORMAT 5)	560
MUPARSER.....	562
PCRE - PERL COMPATIBLE REGULAR EXPRESSIONS	562
PROJ.4	563
RESCUE.....	564
SCINTILLA.....	564
SHAPELIB.....	565
TINYXML.....	565
THE OPENGL EXTENSION WRANGLER LIBRARY (GLEW).....	566
VEGAFEM	566
ZLIB	567

About this Document

FracMan® User's Manual Release 7.7

© 1999-2018 Golder Associates Inc.

This manual contains detailed background on the features of FracMan. While every effort has been made to keep the information accurate, changes may have occurred to the program after this document was finalized. The latest information about the program can be found at the FracMan Website.

IMPORTANT NOTE:

Due to contractual restrictions, your version of FracMan may not contain all the functionality described in this manual. Please consult your licensing agreement for specific details.

Contact and Support:

www.fracman.com

Welcome to the World of Discrete Fracture Networks

A DFN, or discrete fracture network, is a population of individual fractures that work together as a single hydrological and/or mechanical system. DFNs have a huge effect on fluid flow, the structural integrity of mine drifts and tunnels, the production of fractured reservoirs - you name it. Whether transporting oil, caving boulders onto a highway, or carrying water into a cavern where nuclear waste is being stored, fractures are important. And that's where FracMan comes in.

FracMan was developed by Golder Associates Inc. to model the geometry of discrete features, including faults, fractures, paleochannels, karsts and stratigraphic contacts. For convenience, all discrete features will be frequently referred to as fractures in this manual, although the tools provided in FracMan are equally useful for all types of discrete features.

FracMan is designed to provide geologists and engineers with an easy-to-use tool for modeling fractured rock masses, rock mechanics and hydrologic applications in hazardous and nuclear waste management, underground construction, mining and petroleum reservoir engineering.

FracMan provides an integrated environment for the entire process of discrete feature data analysis and modeling.

FracMan provides:

- Data analysis features to allow transformation of raw data into the formats needed for discrete fracture modeling,
- Stochastic simulation of fracture patterns to facilitate three-dimensional visualization,
- Exploration simulation, to improve the design and interpretation of site characterization programs for collection of fracture data,
- Finite element mesh generation and output post-processing, to facilitate flow and transport modeling in networks of fractures and macro support, to facilitate Monte Carlo stochastic simulation.

FracMan's data analysis capabilities include innovative techniques for analyzing fracture orientation, size, intensity and transmissivity. These techniques provide the data needed for fracture geometric modeling, using raw data which are frequently *collected* as part of comprehensive site investigations, but are rarely *utilized* in hydrologic or mechanical modeling. The FracMan package provides a "thinking tool" for discrete fracture modeling. It frees geologists and engineers from the tedium of the modeling process and allows them more time to think about the physical relevance of their models. FracMan accomplishes this with a graphical user interface (GUI), which makes each step in the modeling process immediately comprehensible. The user controls the package by choosing options from a system of drop-down menus, while three-dimensional and two dimensional graphical displays show the user each step in the modeling process, from fracture genesis and sampling to formation of finite element meshes. These graphical displays afford insight into the system of fractures being modeled. By reducing the drudgery of modeling and by giving a clear picture of the results, FracMan frees the user to explore the meaning of those results and to ponder the aptness of the modeling strategy itself.

If you've bought this software, you're already ahead of your competition. Realizing the importance of your own DFNs is just the first step; this manual will show you how you can convert that knowledge into real results. The tools FracMan offers will turn your mysterious geophysical data into the information you need to accomplish what you want. But first you need to install your software.

Installing FracMan

System Requirements

- A computer running Windows 7, or newer
- A minimum of 2 GB of RAM (4 GB recommended)
- A minimum of 4 GB of free disk space.

- Open/GL compatible graphics card.

In order for FracMan to function effectively on your computer, your system must meet the following requirements:

- Display Color Settings must be set to True Color or High Color
- Physical or virtual memory must be at least 2GB. For well-testing or block-K simulation, physical + virtual memory must be at least 4 GB.
- Video memory must be at least 512 MB.

Installing and Initializing FracMan

A simple step-by-step interface is used for installing FracMan onto your system.

Installing FracMan from a CD:

- Insert the CD into your computer's CD-ROM drive.
- Open the Windows Explorer and open the folder that contains the FracMan CD.
- Double-click on the Installer executable file.
- Follow the installation instructions.

Installing FracMan from the web page:

- On the web page, download the Installer file to your hard drive.
- Double-click on the Installer file.
- Follow the installation instructions.

To start the program, open the group folder named "FracMan7" in your Start menu and select the shortcut for FracMan.

License Activation

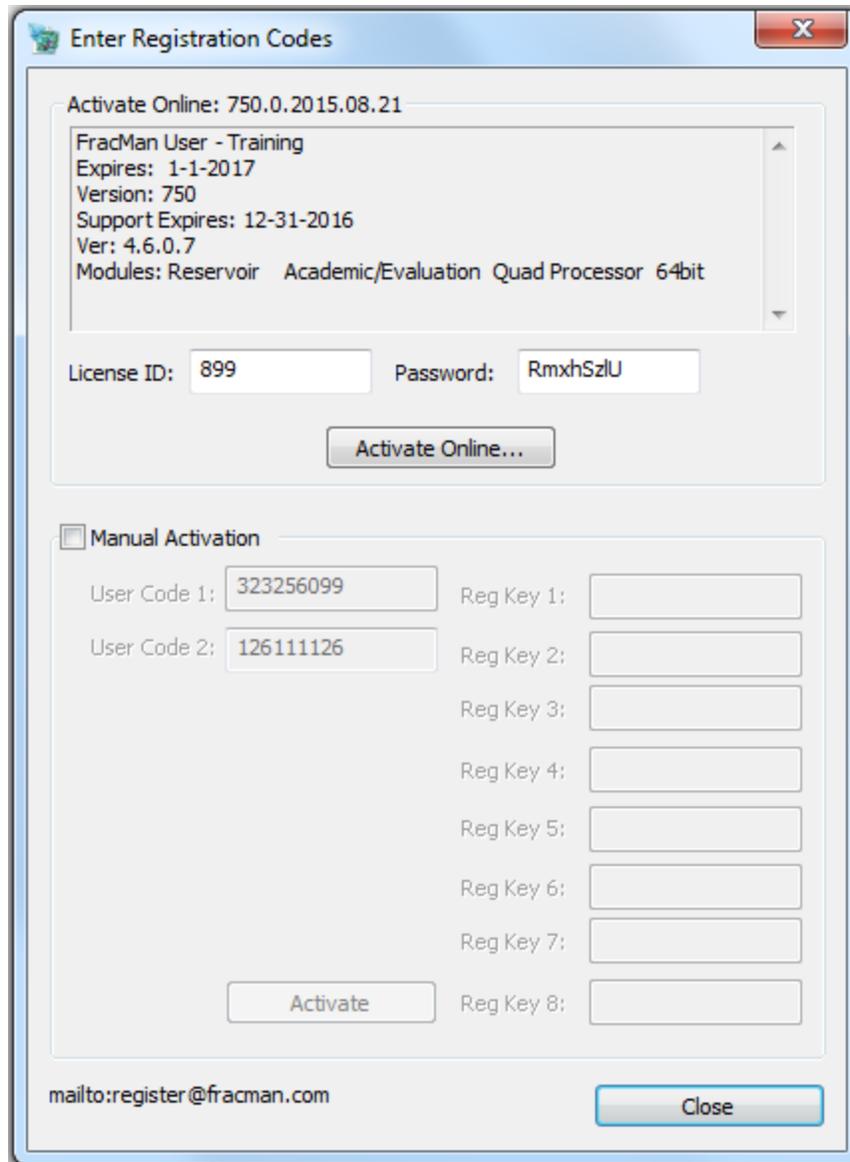
Until you activate your FracMan license, a dialog box will appear each time you open the program telling you how many days remain until your evaluation license expires.

If you click the **Continue** button, the dialog box will close and the program will open normally.

The dialog will reappear the next time you open the program. If you do not register your copy of FracMan, your license will expire after 30 days, and you will no longer be able to use the software. To avoid this, you must register with Golder via the internet.

If the License Activation dialog box is displayed, click the "Register FracMan" button. If the program is already open, go to the "Help" menu and click "Enter Registration Codes..."

A dialog box will appear:



This box tells which version and edition of FracMan you are using and the name and organization under which it is registered. It also shows when your support and registration codes expire. If this is the first time you have installed FracMan you can activate your license by entering a valid License ID and Password and clicking the **Activate Online** button next to the Password box. To ensure that the process was successful, check that the expiration information at the top of the dialog box has changed to match your purchased registration period.

You may also be provided with a USB security Key containing your license. The FracMan installation program will install the necessary drivers – just insert the security key into an available USB port and your license will be active. See [Troubleshooting USB Security Key](#) for tips on fixing problems with your license key.

When your current registration and/or support period is about to expire, FracMan will prompt you to renew. Once your license/support period has been extended, the renewal prompt will stop appearing until your next expiration date approaches.

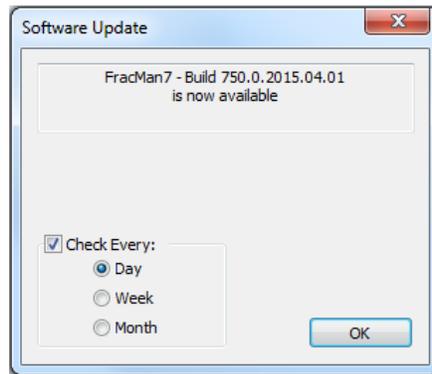
Uninstalling FracMan

Should you ever wish to uninstall, you may do so using Windows Control Panel (Windows 7).

- Click the **Start** button and open the **Control Panel** from the **Settings** menu.
- Click the **Programs and Features** icon
- From the list in the **Programs and Features** window, select FracMan7 and click **Uninstall**.
- Follow any further directions that appear.

Version Check

When your computer is connected to the Internet, FracMan can automatically check for new version releases via the Web. When you open the program, depending on the update settings you have specified, FracMan will show you a dialog box telling you its version status. In order to look for updates manually and to change update settings, select *Help > Check for Updates* from the file menu.



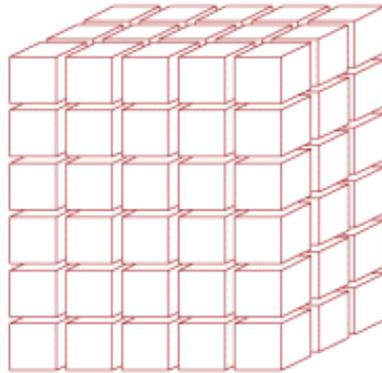
You can set FracMan to look for updates every day, week, or month by checking the appropriate box. If the **Check Every** box is not checked, FracMan will not automatically look for updates, but will only look when you instruct it to. When a new version is available, the dialog box will have a message saying the build date of the version and a blue hyperlink saying “Get the update...” In order to install the new version, just click the hyperlink. If there are no version updates, this hyperlink will not be available. Clicking the “Version Information” hyperlink will bring you to a webpage detailing how the new FracMan build improves upon previous builds.

Getting Started

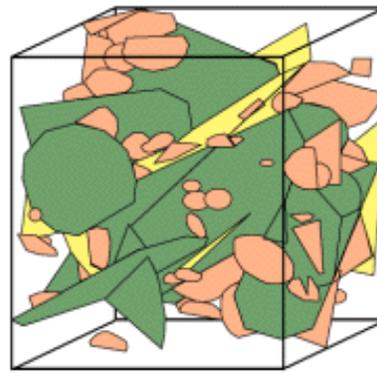
Now that you’ve installed and registered your software, you’re ready to begin modeling. As mentioned in the preface, FracMan uses the DFN approach. DFN models offer numerous advantages over conventional dual-porosity models, including:

- More realistic representation of geology and fracture network geometry,

- Direct simulation of reservoir connectivity, and
- Consistent, comprehensive integration of geological, geophysical, production, and well test data.



Dual Porosity



Discrete Fracture Network

The DFN approach models the geometry of a fracture network explicitly and provides a realistic way of modeling fractured reservoir performance. The approach consists of three general steps:

- Fracture Data Analysis
- Generation of Discrete Fracture Networks
- Discrete Fracture Network Analysis

Step one involves analyzing the information from a variety of data sources to derive the parameters needed for step two. These parameters are fracture locations (spatial model), size, shape, orientation, flow properties, and number of distinct fracture sets.

Step two involves generating multiple discrete fracture networks based on the results of the data analysis.

In step three, these networks are analyzed to derive engineering information. These include simple geometric analysis like the computation of fracture densities, as well as complex multi-well flow simulations.

Typically, a forward modeling approach is used to iterate between steps two and three during the model calibration process. For example, the calculated fracture density or the calculated response from a transient well test is compared to field measurements. If there is not an acceptable match, the input parameters for the fracture network are changed and the network analysis task is repeated with the updated model.

Components of a Reservoir Model

FracMan uses a hierarchical object model for all the objects you need to model a fractured reservoir. The highest object in the FracMan hierarchy is called a *Scene*. The parent *Scene* object has children objects. Objects currently available are:

Fractures and Fracture Sets. Fractures in FracMan are planar polygons with 3 or more sides with no restrictions on location or orientation. They are typically generated in groups (called *fracture sets*) from stochastic descriptions of their properties or from relations to geology. It is also possible to create individual fractures using deterministic values. If fractures are generated in sets, the individual fracture objects are children of the fracture set object. If individual fractures are generated deterministically, they are placed under a *Single Fractures* fracture set object.

Surfaces. Surfaces in FracMan consist of a set of triangles with shared vertices (commonly known as *triangulated irregular networks*, or TIN's) which must be created in structural surface modeling programs, and then imported into FracMan. These surfaces can then be used to define model regions (for example the top and bottom of a reservoir layer), create grids, and define faults. In order for FracMan to treat a surface as part of the actual model, it can be converted to a fracture and assigned properties.

Regions. Regions are user-defined sub-volumes of the model in which fractures are generated. By defining multiple model regions and fracture sets it is possible to create reservoir models that have different fracture properties in different sections of the model. Regions can be simple geometric objects like a box or a cylinder, or they can be defined by more complex geometry such as a geocellular layer, a fault, or a fold.

Wells. FracMan supports deviated wells. Data associated with wells includes fracture intersections from image logs, P10 sampling intervals, flow logs, perforation intervals, pressure histories, and so on. This data can be used in conjunction with other information to derive statistical parameters for the size, shape, location, orientation, aperture, and permeability distributions of a particular fracture set.

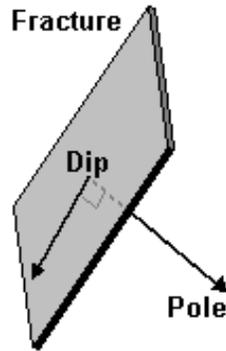
Meshes. Meshes are networks of triangles generated at the same locations as the fractures used for fluid flow simulation. A complete mesh (containing all the triangles representing a complete fracture set) is referred to as a *MAFIC group*. A group can have children called *MAFIC sets*. These are sets of triangles representing individual fractures. See [MAFIC Flow Simulation](#) for more information on MAFIC meshes.

Grids. Grids are three-dimensional volumetric elements with polygonal sides. A grid can be generated outside FracMan by programs like reservoir simulators or geo-cellular models or inside FracMan, where they are used to represent geological features like folds or faults.

FracMan can be used to calculate fracture network parameters (e.g. directional permeability) for grids.

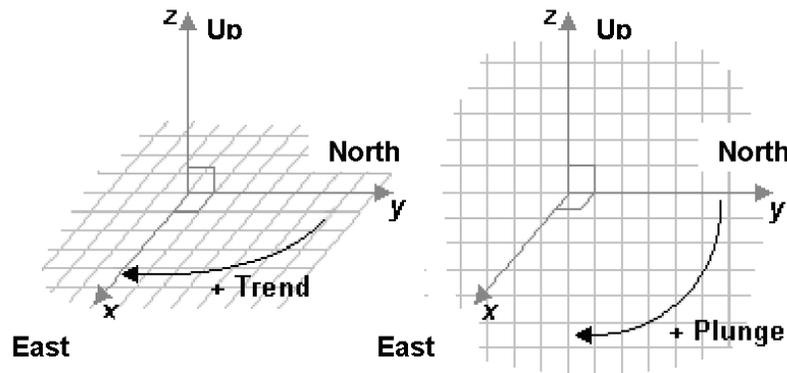
Poles, Dip Vectors, Trends, and Plunges

This section is a review of some terms used in fracture modeling. It may be skipped if you are comfortable with these terms.



In general, *poles* and *dip vectors* are terms used to define the orientation of a plane in space. Fractures in FracMan are modeled as planar polygons and use poles or dips to describe their orientation. The *pole* is a vector normal to the fracture plane and usually (by convention) points downward. The *dip vector* is a vector normal to the pole and lies in the plane pointing in the direction of the maximum slope gradient.

Trend and *plunge* describe the orientation of a vector. The *trend* is the horizontal angle in the *x-y* plane away from the North (positive *y*-axis). The *plunge* is the vertical angle in space away from horizontal (*x-y* plane). The orientation of a dip or pole vector is completely specified by its trend and plunge.

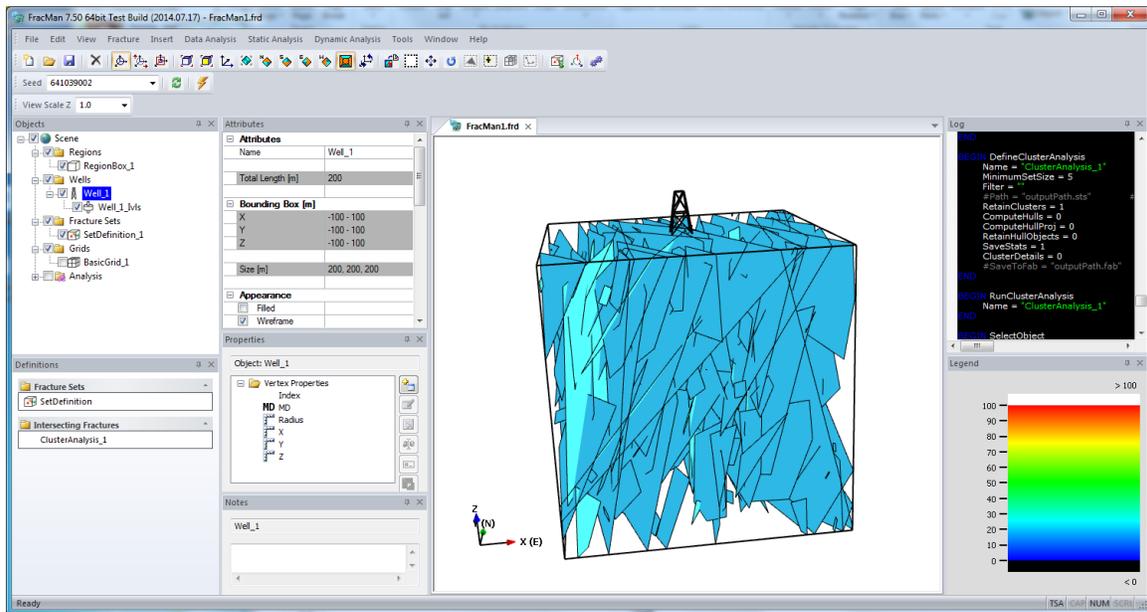


User Interface

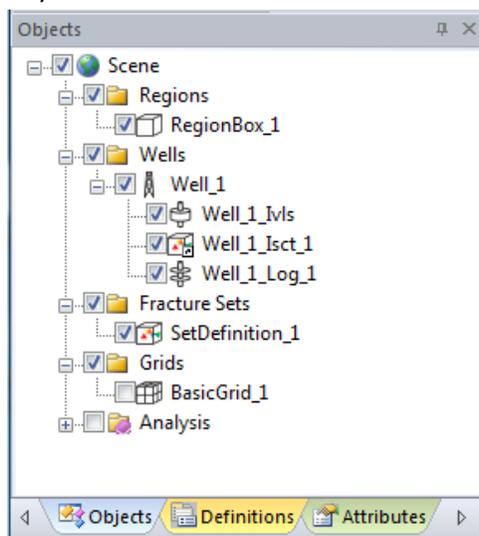
The FracMan program window contains several different sub-windows or bars that make up the user interface and model visualization. Throughout this manual, these different sections will be referred to by the names designated below and shown in the following figure:

- Application Menu
- Application Tool Bar
- Objects
- Definitions

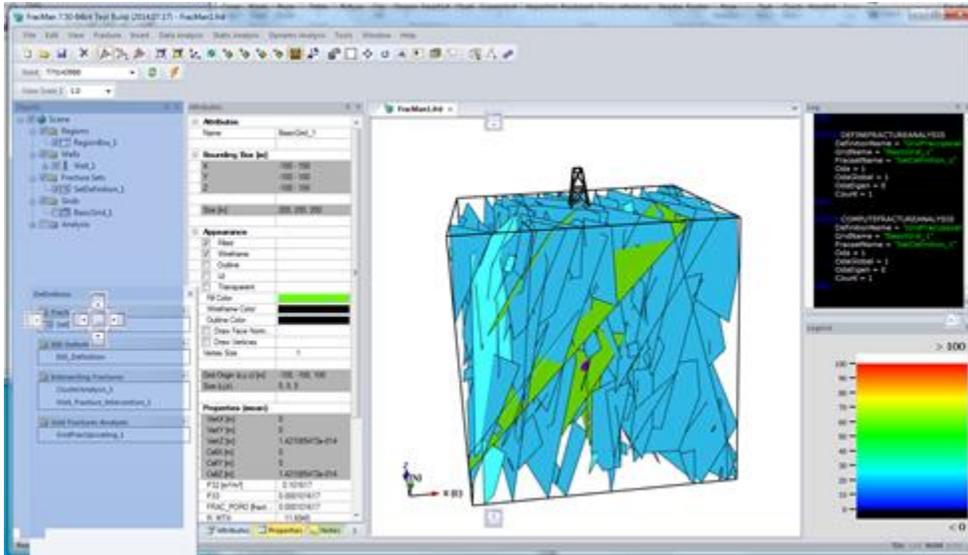
- Attributes
- Properties
- Notes
- Visualization window
- Log
- Legend



FracMan allows a great deal of flexibility in how you arrange the docking bars which surround the main visualization. Each window can be docked to either side above, below or alongside other docked windows. Additionally the bars can be docked on top of another bar, forming tabs along the bottom which allow you to select between the different windows.



The three window panels on the side of the screen can be detached from the main FracMan window and moved around the screen if desired. Their vertical order can also be changed by dragging and dropping.



Simply drag the window over the location you'd like the window to attach to. You should see several icons representing the different possible dock locations. Drag the cursor to one of these icons and the window will dock to that point:



Dock as a tab on the window beneath the icon



Dock to the left of the window



Dock below the window

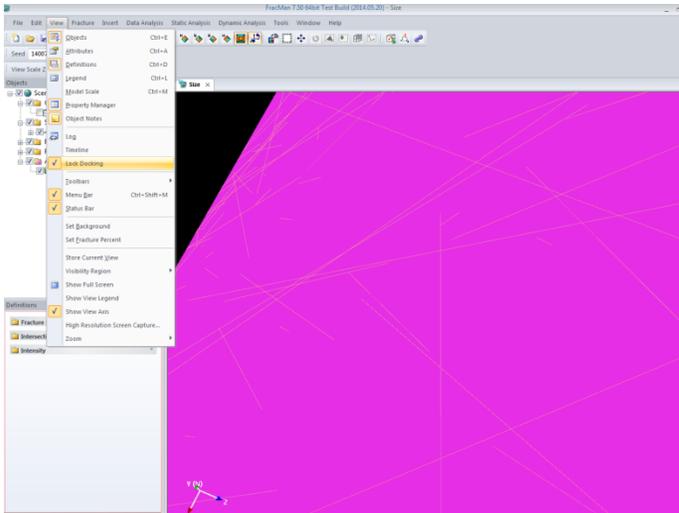


Dock above the window

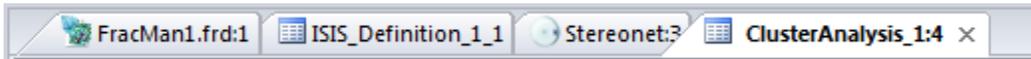


Dock to the right of the window or on the right side of the application

Docking windows can be locked in place to prevent inadvertent moving around.

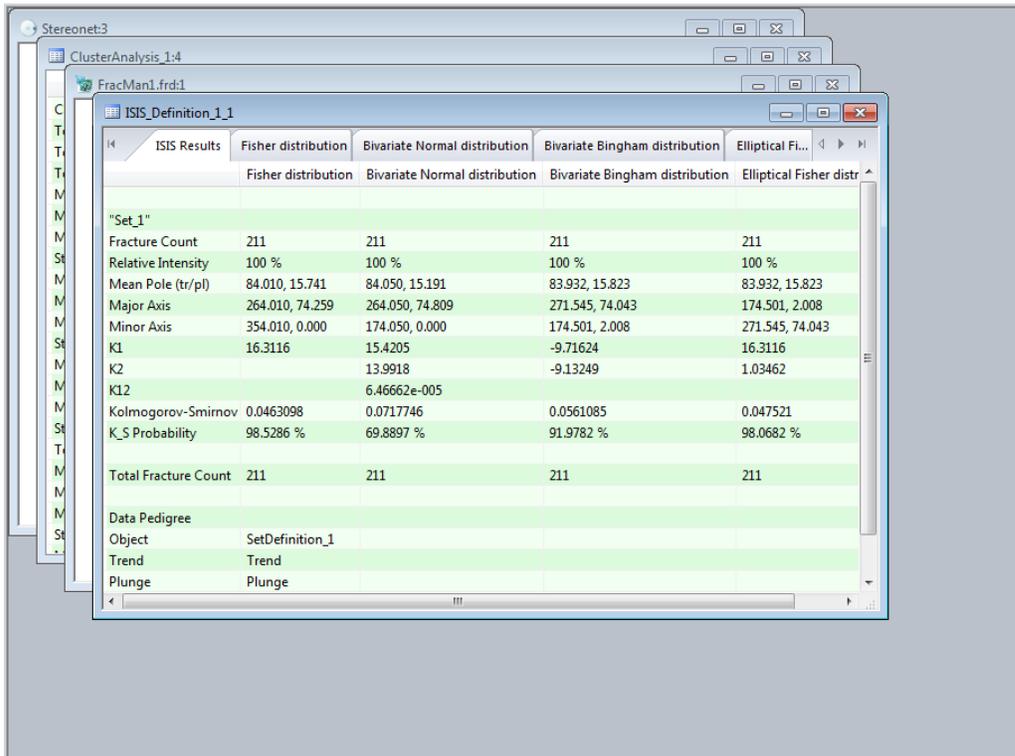


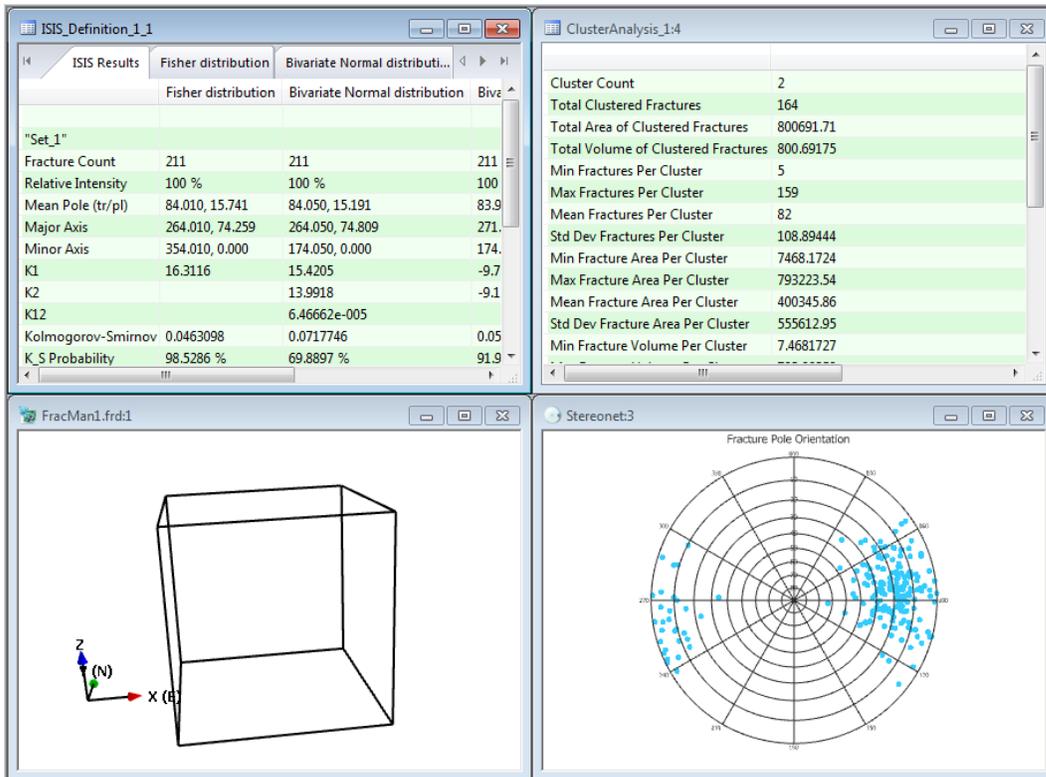
At the top of the visualization window is a row of tabs. As you run your analyses, additional windows will appear in this row



Click on any tab to bring that window to the front.

You can also get rid of the tabbed windows by Un-selecting "Tabbed Windows" from the "Window" menu. This will return FracMan to the old mode of having independent windows. Selecting "Cascade" or "Tile" from the "Window" menu will provide additional display options.

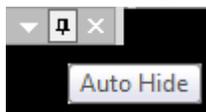




Control Windows

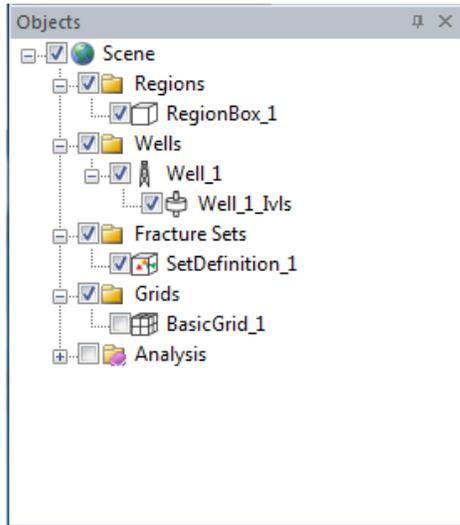
There are eight main windows/panels used in FracMan: Objects, Definitions, Attributes, Properties, Notes, the Visualization window, Log and Legend. Each window displays a different portion of the file's contents, or displays the data in a different way.

All the docking bars can be set to auto hide by selecting the "Pin" icon



When autohide is enabled, the bars will minimize to the edge of the main window and appear when you hover the mouse over their icon.

Objects Bar



The **Objects Bar** displays the contents of your FracMan model in a tree chart format. Left-clicking on the minus/plus box expands or compresses the tree.

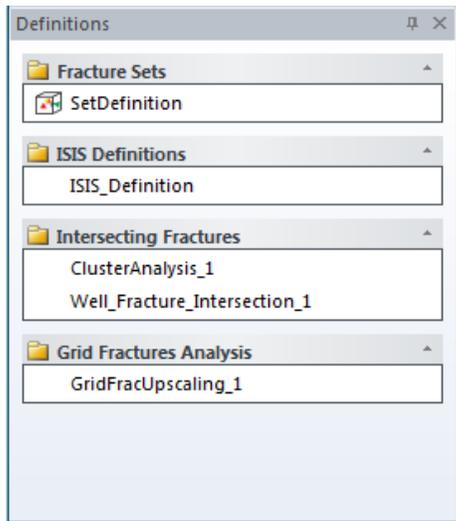
Objects can be hidden from the Visualization window (they are not removed from the actual FracMan model) by un-checking the box next to the object name.

The Objects window can also be used to select objects. Simply click on the object name, and the object will be highlighted. Once an object is selected you can edit or control the display properties of the object in the **Attributes** and **Properties** windows.

The **Objects Bar** can be shown and hidden by selecting “Objects” from the “View” menu or by pressing Ctrl-E. Right-clicking in the Objects window away from any objects will bring up a menu of display options.



Definitions Bar

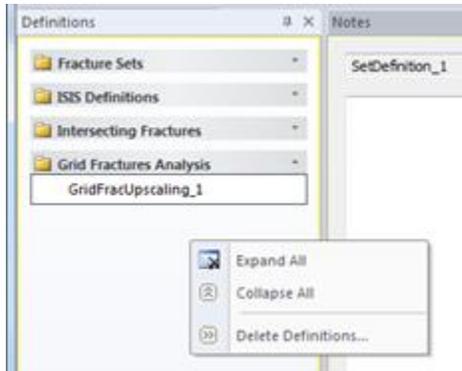


The **Definitions Bar** allows you to change the parameters of any definition previously created in the current FracMan model. Definitions in FracMan are sets of parameters used to perform complex tasks.

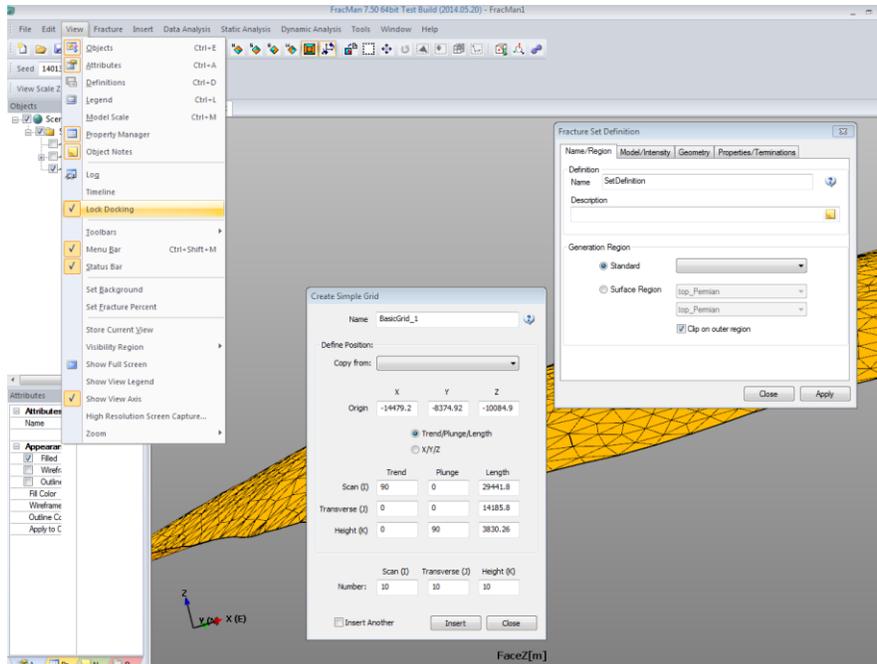
Clicking on a **Definition** launches a small menu that lets you either edit the definition, or start the fracture generation or the export of flow simulation files.

The Definitions Window can be shown and hidden by selecting “Definitions” from the “View” menu or by pressing Ctrl-D.

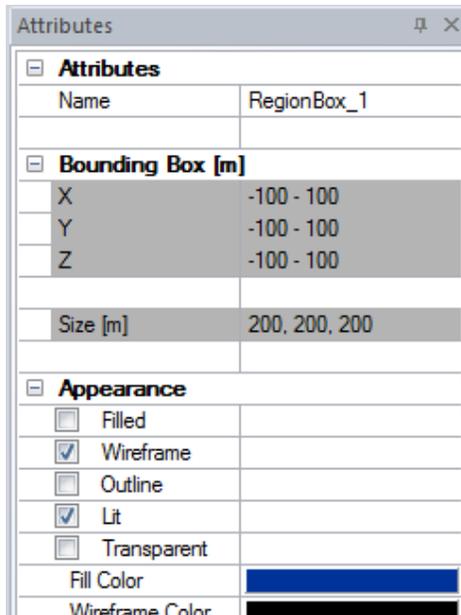
Right-clicking in the Definitions window away from any definitions will bring up a short menu of options including Expand All, Collapse All and Delete Definitions.



Multiple definition dialogs can be viewed simultaneously.



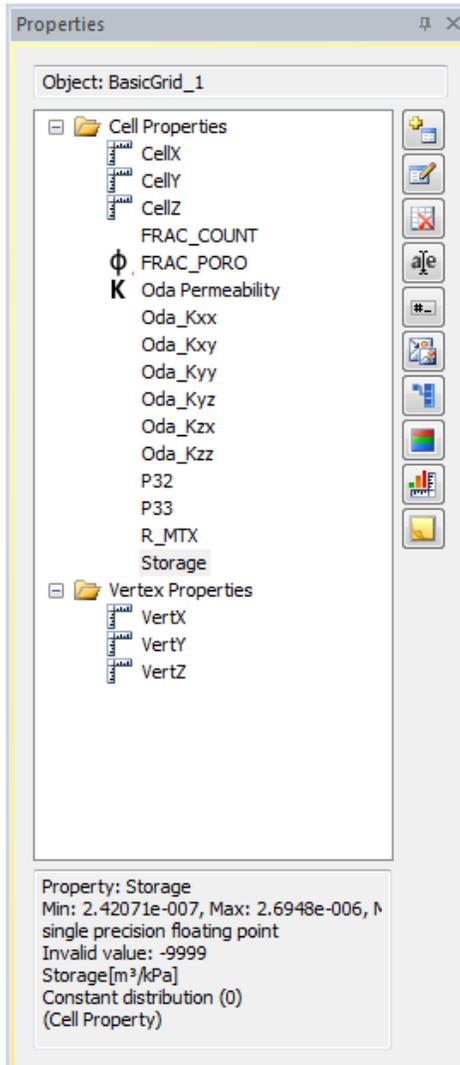
Attributes Bar



The **Attributes Bar** allows you to edit many of the parameters and properties of the object or objects currently selected in the **Objects** window. When a graph (such as a histogram or stereonet) is active, the Attributes window allows you to edit properties of that graph.

The Attributes window can be shown and hidden by selecting “Attributes” from the “View” menu or by pressing Ctrl-A.

Properties Bar



The **Properties Bar** allows you to add or edit properties of the object or objects currently selected in the **Objects** window.

The Properties window can be shown and hidden by selecting “Properties” from the “View” menu.

The object’s name is shown at the top and existing properties will be listed in the left sub-window. Clicking on one of these properties will bring up a list along the bottom of the box giving the property name, its minimum and maximum values, mean and size as well as the data type, invalid value token, units and the distribution or correlation that defines it. *Note:* if the property is defined by an equation, the equation will not be shown; similarly, the program will not tell you if the property was copied from another property.

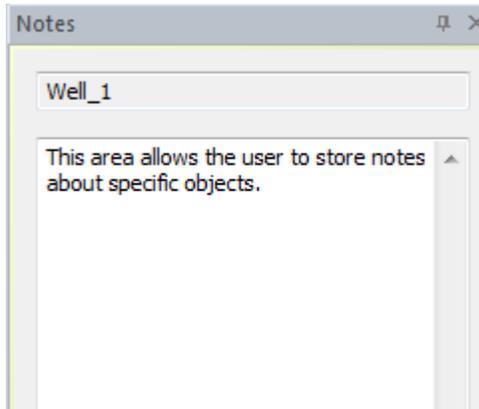
The buttons on the right side of the panel allow these functions:

- [Add Property](#)
- [Edit Property](#)
- [Delete Property](#)
- [Rename Property](#)
- [Set Invalid Value](#)
- [Offset/Scale Values](#)
- [Replace Values](#)
- [Visualize](#)
- [Set Units](#)
- [Set Property Notes](#)

If you choose a property and right-click, you get a menu of options including those on the tool bar and additionally:

- [Duplicate Property](#)
- [Define Orientation Properties](#)
- [Object Groups](#)

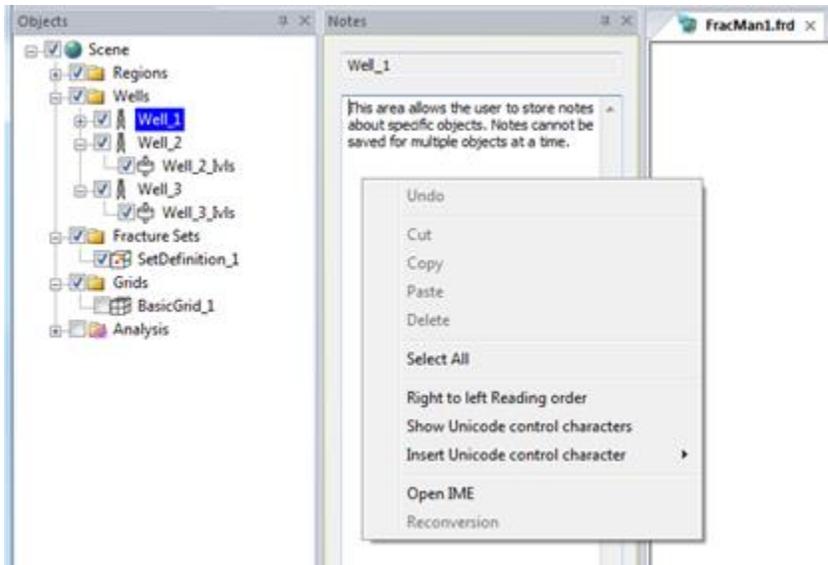
Notes Bar



The **Notes Bar** allows you save comments about the object currently selected in the **Objects** window.

The Notes window can be shown and hidden by selecting “Notes” from the “View” menu.

Right-clicking in the Notes window will bring up a menu of options.

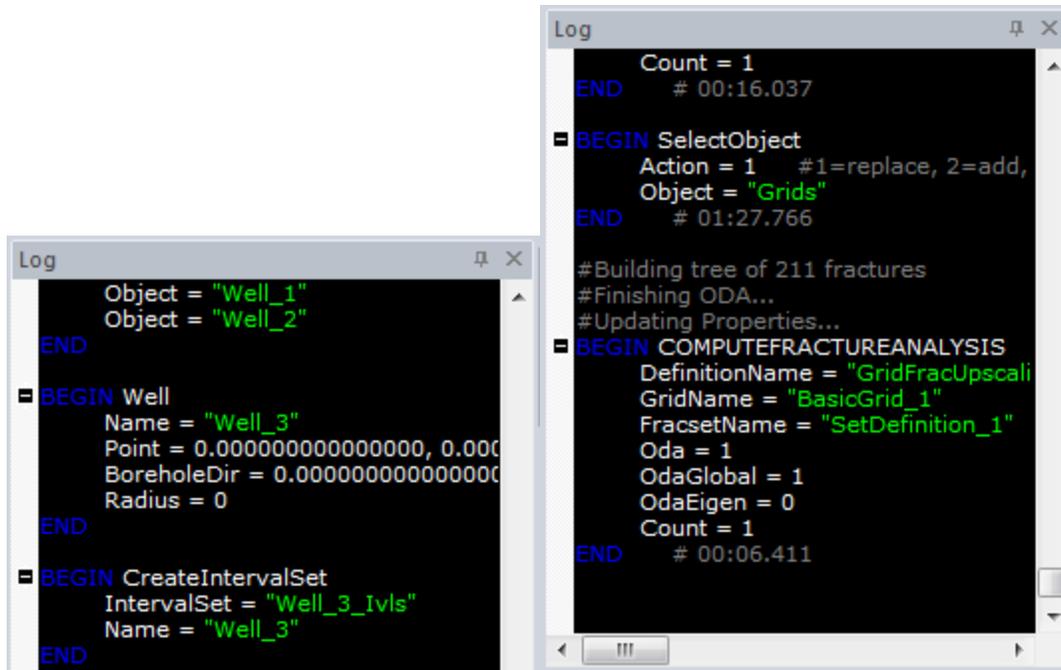


Visualization Window

The **Visualization** window displays all visible objects in a FracMan model. By using your mouse, you can study your model from any direction or distance. There are three methods to investigate your model in the Visualization window: rotate, translate, and zoom in or out on the model. The desired method can be selected by clicking on the respective button on the Application Tool bar, or, if you are in the default rotate mode then you can use the keyboard shortcuts indicated in the [Tool Bar](#) section to translate and zoom.

Log Bar

The log bar continuously displays user operations in the form of Macro commands. See the section [Using Macros](#) for a complete description of the commands that are used by the macro system.



Left: Regular Log output. Right: Log output with Display Timing and Display Progress checked.

The contents of the log may be saved as an ASCII text file by right-clicking in the log window and selecting “Export”. You may also copy and paste from the log or clear its contents via the right-click menu. In the right-click menu there are two display options as well: “Display Timing” and “Display Progress”. When the former is checked, the system time at function completion will appear in the log next to “END”; when the latter is checked, processes that contain multiple operations will log each operation as it is begun.

All user operations can be logged to an ASCII text file for later reference by right-clicking on the log window and selecting “Log to File”. This function can also be used as a log of macro progress and for macro debugging.

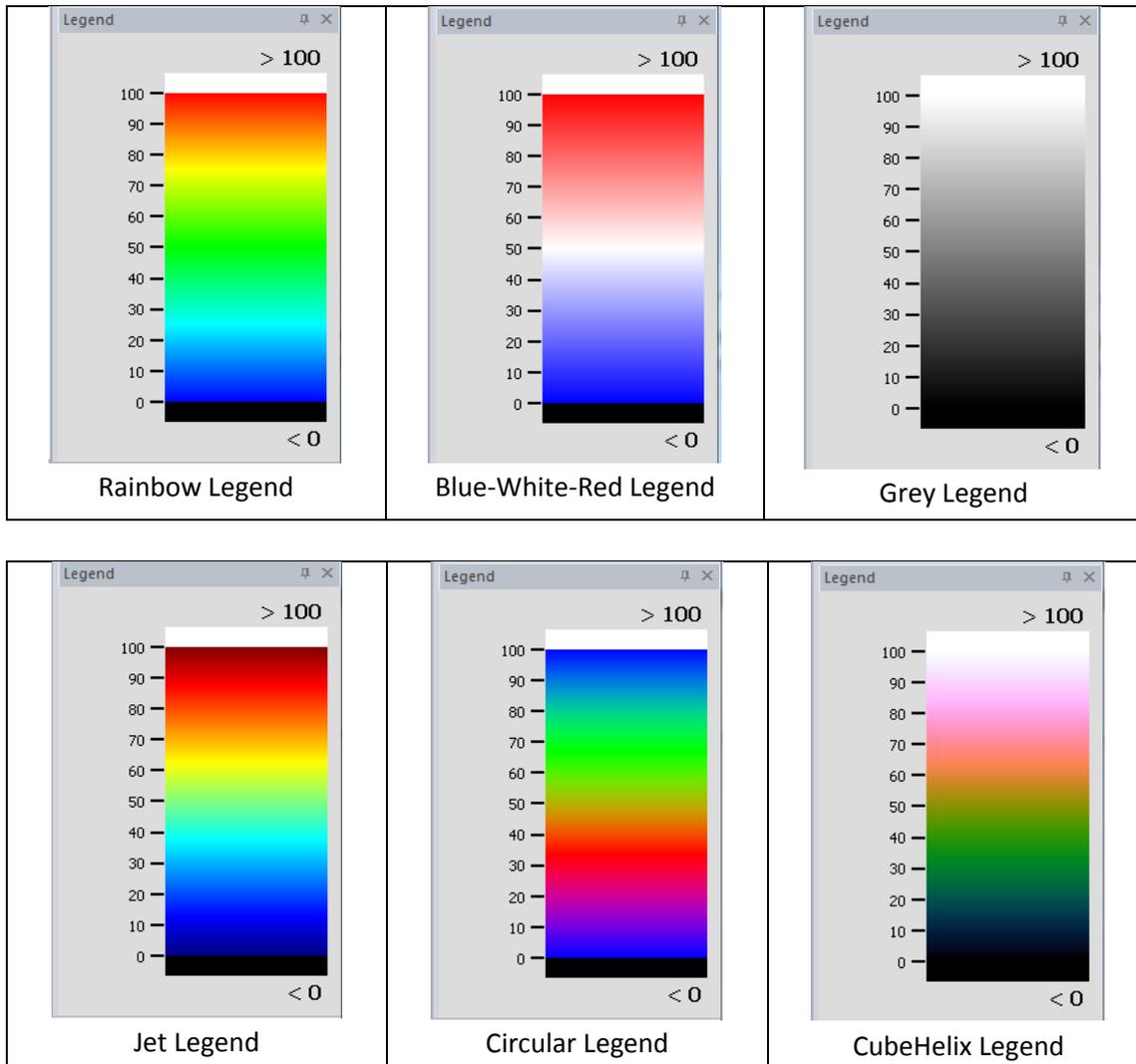
The Legend Bar

You can display a legend in FracMan by selecting **Legend** from the **View** menu. The legend can be moved around the screen as its own window or joined to the main FracMan window’s sidebar just like one of the other panels. The legend is helpful when visualizing different grid and fracture properties.

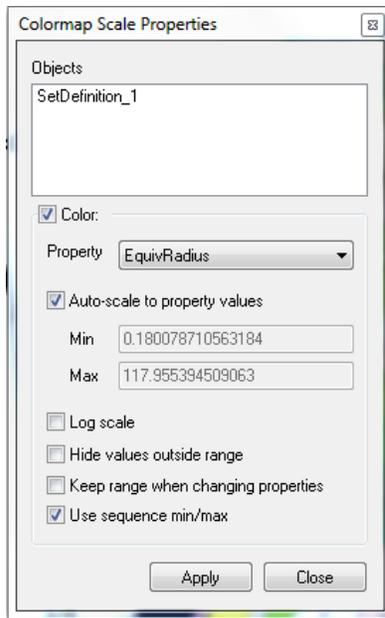
If you wish to adjust the color position of the legend – so that blue signifies a value other than 0, for instance – click anywhere on the color scale and drag it vertically until the values and colors match up as you desire.

Right-clicking on the legend allows you to alter its properties. There are six different available color schemes: Rainbow, Blue-White-Red, Grey, Jet, Circular and CubeHelix. Jet provides a little more contrast than the standard “Rainbow”. The Circular scheme is specially designed for visualization of orientation data (trend/plunge etc) where the colors map around allowing you

to more easily compare values. CubeHelix presents a uniform saturation change that allows it to print equally good on color or black and white and also looks correct on projectors etc.



The right-click menu also has three other options. **Reverse** flips the legend, so that, for instance, low values will be red and high values blue. **Reset Position** undoes any changes you have made to the legend, returning it to the default position. **Properties** allows you to modify the scale as well as change what aspects of the object are visualized and how. A window will pop up that allows you to set the minimum and maximum values of the color map. As indicated by the legend, elements with values greater than the maximum value will be drawn in white. Elements with values smaller than the minimum value will be shown in black. The properties window also allows you to switch between linear and log scale.



The dropdown menu next to “Color: Property” includes all properties of the selected object; the selected one will be visualized.

If no boxes are checked then FracMan uses the minimum and maximum values specified by the user. If the **Auto-scale to property values** box is checked, FracMan will scale the color map automatically. This automatic color map scaling can be done either using the values for the currently displayed property (e.g. fracture apertures or pressures for the current time step) or using the values for a property sequence (e.g. pressures over multiple time steps). For the second option the **Use sequence min/max** box has to be checked. The second option is particularly useful when visualizing transient pressure changes in a series of pictures.

If the **Keep range when changing properties** box is checked, the minimum and maximum values will not change when you select different properties to visualize, otherwise the minimum and maximum values will reset for the new property.

Tool Bar

The tool bar contains several buttons which serve as shortcuts to common operations FracMan performs. The buttons and their functions are as follows:

	New	Creates new FracMan file.
	Open	Opens an existing FracMan file.
	Save	Saves the current FracMan file.
	Delete Selected	Deletes the selected object.

	Free Rotate (Left-click and hold)	Rotates model in the direction the mouse is moved. Axis of rotation is centered in model and perpendicular to the direction the mouse is moved.
	Free Translate (Left-click and hold + Shift, or hold left and right mouse buttons)	Translates model in the direction the mouse is moved.
	Free Zoom (Left-click and hold + Control, or use mouse wheel)	Zooms in on model when mouse is dragged down. Zooms out on model when mouse is dragged up.
	Zoom Extents	Zooms in or out in order to fit the entire model (visible or not) within the boundaries of the Visualization Window.
	Zoom Extents Selected	Zooms in or out in order to fit the selected object(s) within the boundaries of the Visualization Window.
	Align	Aligns the view to the nearest orthogonal axis.
	Plan View	Changes to top view and orthogonal projection.
	North	Changes to North view.
	South	Changes to South view.
	East	Changes to East view.
	West	Changes to West view.
	Perspective/Orthographic	Toggles between perspective and orthogonal projection. When the button is depressed, the model view is given the appearance of depth.
	Enable Spin	Allows the user to turn the spin feature on or off. When the button is depressed, the model view can be set to rotate continually.
	Copy to Clipboard	Copies an image of the view window to the clipboard.
	Area Select	Selects all objects in the enclosed field of the Visualization Window.
	Translate Objects	Moves the selected object.
	Rotate Objects	Rotates the selected object.
	Select Faces	While this is active, the object editing tools (Select, Translate Object, Rotate Object) will manipulate only object faces.
	Select Vertices	While this is active, the object editing tools will manipulate only object vertices.
	Select Cells	Select grid cells from the current object.

	Select Traces	Select Traces from a tracemap object.
	Generate Fracture Set	Creates a new geometric fracture set definition.
	Generate Single Fracture	Creates a new single fracture.
	Record Macro	This toggles whether a macro is recording. When the button is depressed, macro recording is on.

Before you can start using the program like a pro, you need to know where the commands are and what they do.

Menu Bar

There are eleven drop-down menus on the menu bar. These are:

- The File menu: program and file options, as well as import and export functions.
- The Edit menu: object selection and modification tools.
- The View menu: view attribute controls.
- The Fracture menu: DFN and single fracture generation and manipulation.
- The Insert menu: object generation, including wells, regions, grids and surfaces.
- The Data, Static, and Dynamic Analysis menus: tools for analyzing well, grid, surface, and fracture data; the meat of the program. These will be covered in greater detail later in the manual.
- The Tools menu: program options, macro control, and Monte Carlo analysis.
- The Window menu: multiple window manipulation.
- The Help menu: program information and assistance, online registration.

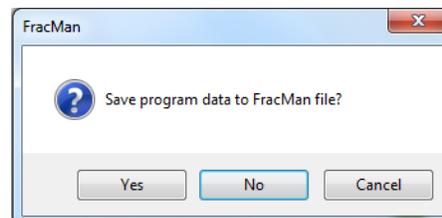
Many of FracMan's major functions can be accessed from these menus.

File Menu

The **File** menu has eight commands under it: **New**, **Open**, **Close**, **Save**, **Save As**, **Import**, **Export**, and **Exit**. It also lists the most recent project files that have been saved.

New

This creates a new, blank FracMan file. If you have a FracMan file open when you click New, the program will automatically close it. If the open document is not empty, FracMan will prompt you to save it before closing:



This dialog also appears when you quit FracMan. If you click **Cancel**, the program will return to the open file without generating a new file. If you click **No**, the program will not save the open file before creating a new one. If you click **Yes**, the program will open a dialog asking you to specify the file name and directory you wish to save to. Once you click **Save**, FracMan will save your current file, close it, and make a new file.

Open

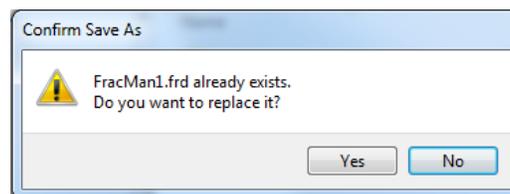
This allows you to load a preexisting FracMan file. It brings up a browser dialog in which you can find and select the FracMan file you wish to open. Once you have selected the file you want, click the **Open** button. This will load the file into FracMan. If another file is already open, the program will prompt you to save it and close it, just as when you close the program. Clicking **Cancel** will return you to the program as it was, without opening another file.

Close

This closes an open FracMan file without quitting the program. It will prompt you to save the file just as when you create a new file, but instead of displaying another file in place of the closed one, the program remains open but has no workspace. In order to use any of FracMan's toolset, you must have a FracMan file open.

Save

This saves any changes you have made to the current FracMan file. If you have not already saved the file, the program will bring up a browser dialog, allowing you to select which directory you wish to save the file to. You may also specify a name for the saved file. If you do not specify a name, FracMan will save it as "FracMan1", the default filename. Click the **Save** button. If you have selected a file name and directory that are identical to those of a preexisting FracMan file, the program will warn you:



If you click **Yes**, your current save will overwrite the old one. If you click **No**, FracMan will return you to the previous **Save As** dialog box, allowing you to select a different directory or file name and save again. Clicking **Cancel** on the **Save As** dialog will return you to your file without saving any changes.

Save As

If you wish to save your changes but do not wish to overwrite the current FracMan file, this option allows you to specify a different file name and/or directory under which to save the current version of your file as a separate file from any previous saves.

Import

This allows you to bring many different types of data into FracMan: volumetric grids, triangulated surfaces, point data, tracemaps, well logs, seismic data and data exported from the FracMan program suite.

- **File Format Assistant** is useful when you aren't sure what file format you have. Once you select a file, the format may be automatically detected from the extension and the identified type will be shown in the "File Format" selection list. If the format is not automatically detected, identify what type of file it is with the "Primary Data Type" menu. A list of recognized file formats will then appear in the selection list for the "File

Format”. When the “Preview Selected File” button is highlighted, the window below shows several lines from the selected file. When the “Preview Selected Format” button is clicked, the view changes to show an example of this type of file format so the user can compare them and verify that the file format is the same.

- **Well** data describing one well can now be imported from several files including one for the well trajectory and separate ones for the well collar, deviation and well log.

Supported file formats include:

- [Petrel well trace data](#) (.txt, .asc, or .dat)
- [GoCAD well files](#) (.wl, .gp)
- [Log ASCII Standard](#) (.las)
- [FracMan object files](#) (.sab)
- [ASCII well fracture files](#) (.txt, .ors)
- [generic borehole definition files](#) (.txt, .csv, or .dat)
- [well interval \(tops\) file](#) (.txt)
- [well deviation file](#) (.txt)
- **Fracture File** brings in:
 - FracMan fracture files (extension .ffb, .fab and .fabgz – See [Appendix E FAB Fracture File Format](#))
 - [legacy MaficOil fracture files](#) (.itm_fra)
 - [generic fracture data](#) (.txt)
 - [Tomographic fracture images](#) – TFI (.csv).
- **Grid** accepts:
 - [ECLIPSE ASCII grid files](#) (.grid, .grd or .grdecl)
 - [stress field files](#) (.str)
 - [3DMove grid files](#) (.sdb)
 - [RESCUE grids](#) (.txt or .bin)
 - [GoCAD SGrids](#) (.sg, .gp)
 - [Modflow discretized grids](#) (.dis)
 - [FeFlow grids](#) (.fpd, .fem)
 - [ASCII grids](#) (.fgr)
 - [FEHM mesh](#) (.geo)
- **Surface** allows surfaces to be imported from:
 - [DXF format](#) (.dxf)
 - [RMS format](#) (.txt or .rms)
 - [ZMap Surface](#) (.zcor, .dat)
 - [CPS-3 Surface](#) (.cps)
 - [Earth Vision Surface](#) (.txt, .dat, .ev)
 - [GoCAD Surface](#) (.ts, .gp, .tsurf)
 - [Petrel Surface](#) (.txt)
 - [Charisma Fault Stick](#) (“.”)
 - [Polygon File Format](#) (.ply)
 - [ESRI Shapefile](#) (.shp)

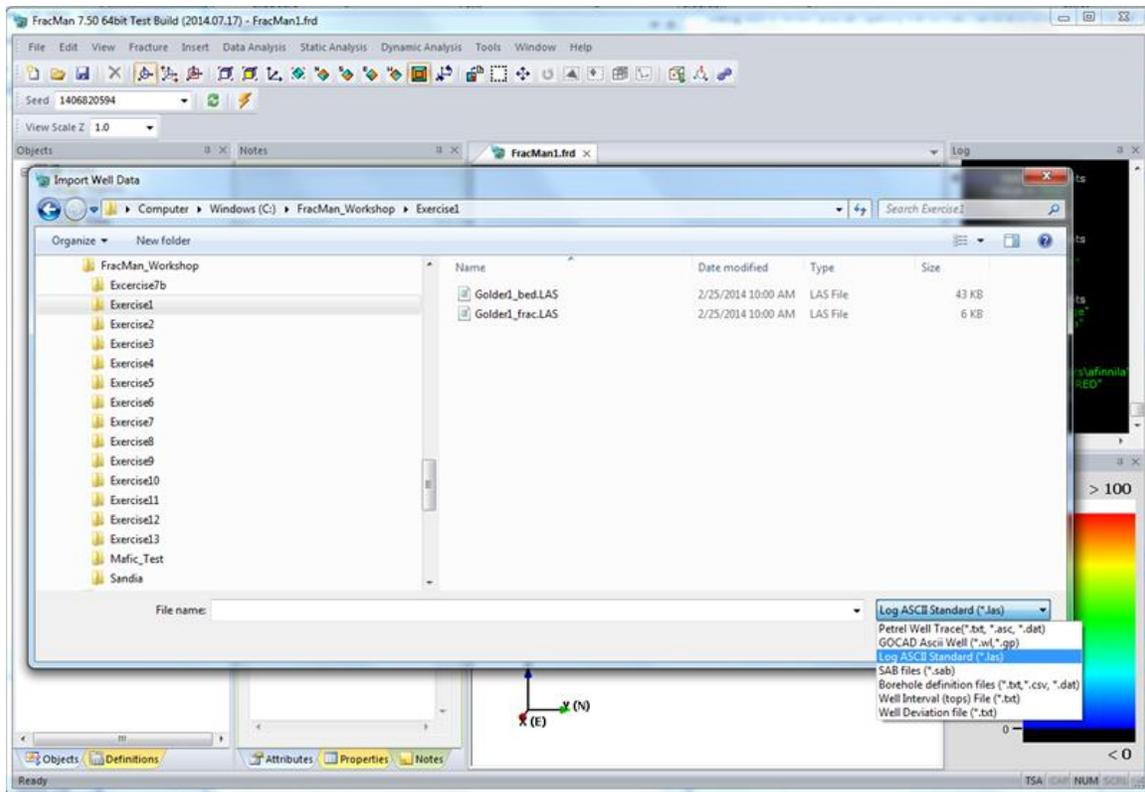
- [Maptek Vulcan](#) (.00t_asc)
- [Surpac Digital Terrain Model](#) (.dtm)

See [Surfaces](#) for more information about surface objects.

- **Tracemap** brings in trace data from:
 - [FracMan Trace Data Files](#) (.f2d)
 - [DXF Trace Data](#) (.dxf)
 - [GOCAD PLine](#) (.pl, .gp)
 - [Polygon File Format](#) (.ply)
 - [ESRI Shapefile](#) (.shp)
 - [CPS-3 lines](#) (.dat)
 - [Irap classic lines](#) (.irap)
 - [Zmap+ lines](#) (.zmap)
 - [MAFIC Particle Track](#) (.pcl)

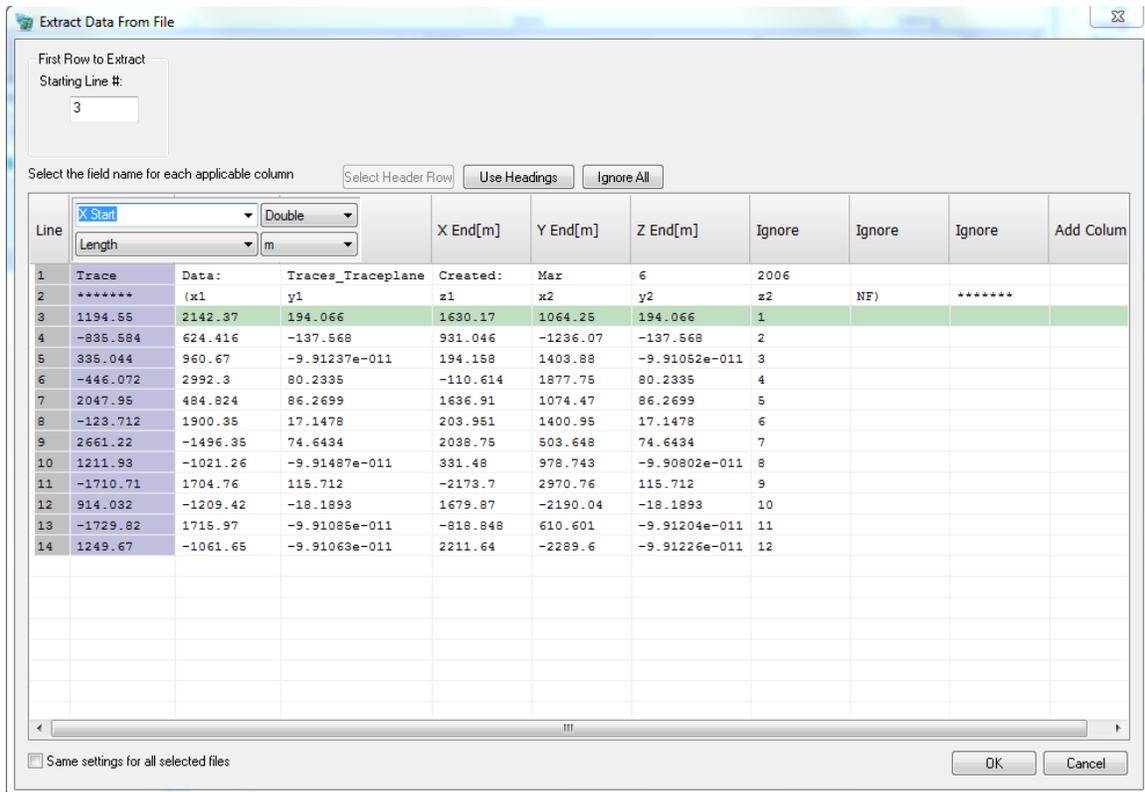
See [Tracemaps](#) for more information about tracemap objects.
- **Point Data** accepts text files containing point coordinates in three dimensions without regard for file extension. See [Point Data Object](#) for more information about Point data objects. Supported file formats:
 - [GoCAD VSet](#) (.vs, .gp)
 - [Polygon File Format](#) (.ply)
 - [ESRI Shapefile](#) (.shp)
 - [MAFIC Particle Track](#) (.pcl)
 - [Surpac String File](#) (.str)
 - [SEG Y format data](#) (.segy, .sgy)
- **Mafic Definition File** imports MAFIC model files (extension .sab). See [MAFIC Flow Simulation](#) for more information about MAFIC files.
- **MAFIC Mesh** brings in MAFIC mesh files (.mafic_input, .maf, .mff, .mafic, .mft or .rst)
- **Seismic Data** import seismic cube data or microseismic point data (.segy).
- **Fluid Database** import fluid database used to store hydraulic fracturing fluids (*.hff or *.txt).
- **Merge from FracMan File** merges objects from other FracMan project files (.frd).
- **FracMan Object File** imports objects from FracMan object files (.fro).

When you have selected the type of file you wish to import, FracMan will bring up a browser dialog box in which you can specify the format, name and directory of the file which you wish to import.



Only files with applicable extensions will be shown as available to import. For instance, if you have selected Log ASCII Standard as the file format you wish to import, files with extension .txt will not appear in the browser window even if there are such files in the shown directory. In order to change the file format you wish to import, you must click on the drop-down menu next to “Files of type:”. This will bring up a list of all possible formats under the type of data you have selected. If your file extension is not in the list and you know that it is in one of the available formats, you may type * into the drop-down menu next to “File name:” and click on **Open**. This will make all files in the shown directory visible, no matter what file extensions they have. Then you may select the appropriate file and open it. *Note:* if your file is not in the appropriate format, it will not load.

When you have the desired file selected – the name of the file should appear in the box next to “File name:” – click the **Open** button. Many file formats will load automatically at this point, but some types, such as point data, will bring up a dialog box asking you to identify the columns in your data.



FracMan automatically guesses what each column is based on the header for each column. Clicking the **Guess Columns** button will also check the column headers for clues as to what the columns contain. When manually selecting columns, the user can specify the unit and precision as well as the data type and name. In order to specify what type of data each column contains, you must click on the title of the column. Make sure that the columns have been correctly identified before clicking the **OK** button. If the type of data contained in that column is not in the list, you may select “Ignore” and the program will not take the column’s data into account. Doing this keeps FracMan from trying to read the data and becoming confused. Clicking on the **Ignore All** button will automatically set all column content labels to “Ignore”. Once you imported your file, FracMan will remember your column specifications for all future imported files of the same format.

When you have finished labeling your columns, enter the line number on which data begins in the box in the upper left-hand corner of the dialog (“First Row to Extract Starting Line #.”). This will be the first line in the file after the header information. If there is no header, just raw data, enter 1 in this box.

Once all of this information has been specified, you may click the **OK** button, and FracMan will load your data file. You can check to see whether the file was successfully imported by checking the Objects window. If the import was successful, a new object will have appeared here, even if it does not show up in the Visualization window.

Imported objects in the Object window are grouped under blanket categories based on their type. For example, well traces will be grouped under “Wells”, while image logs and ORS data will be grouped under “Well Logs”. All categories are grouped under “Scene”. In order to display children for each parent category, simply click on the + next to the category name. The children will appear and the + will change to a -. To hide the children again, click on the – and they will disappear.

Export

This menu item allows you to send data out of a FracMan file into its own separate file. This makes it possible to import FracMan data into other programs. FracMan can export several different types of file.

- **Fracture File** allows you to export all fractures and fracture sets in your file as FracMan fracture files (extension .fab), GoCAD surface files describing your fractures (.ts), and MaficOil compatible fracture data (.itm_fra).
- **Well** exports a FracMan model file (.sab) containing all wells in your file.
- **Grid** takes the selected grid and exports it as an ECLIPSE grid (.grd or .grdecl), a stress field grid (.str), a FracMan fracture file (.fab), or a RESCUE grid (.bin or .txt). At least one grid must be selected before this function can be called.
- **Surface** allows the selected surfaces to be exported to GoCad (.ts).
- **Object Properties** export an object’s properties to a generic property file (.txt) or a GSLib simplified GEOEAS file (.txt).
- **FracMan Object(s)** select objects from the current project and save them in the FracMan Object file format (.fro). These can be imported into other FracMan projects.
- **Mafic Definition File** creates a MAFIC-compatible well data file (.sab).
- **FracMan Definitions** select definitions from the current project and save them in the FracMan Object file format (.fro). These can be imported into other FracMan projects.
- **Screen Capture** save the current contents of the Visualization window in any of the following formats:
 - Encapsulated Post-Script (.eps)
 - Portable Document format (.pdf)
 - Scalable Vector Graphics (.svg)
 - BMP (.BMP, .DIB, .RLE)
 - JPEG (.JPG, .JPE, .JRLE)
 - GIF (.GIF)
 - TIFF (.TIF, .TIFF)
 - PNG (.PNG)
- **Fluid Database** Export the database of hydraulic fracturing fluids (*.hff).
- **3D PDF** Export a visualization of fractures, wells, well logs, point data and surface objects to a 3D PDF file.

When you select one of these file types to export, the program will bring up a browser dialog that allows you to specify the format, name, and directory of the export file you wish to create. The available formats will be displayed when you click on the drop-down menu next to “Save as type:”. Select which of these you wish to create. Type the name for your file in the box next to “File name:”. When you have done these things, press the **Save** button, and the exported file will be created in your chosen directory.

Recent Files

FracMan displays a list of quick links to the last six FracMan files that you have saved. This list is found between the **Export** and **Exit** tabs on the **File** menu. The files are numbered and listed from most recent to least. You may open the file directly from here by clicking on its name. This avoids the hassle of browsing through many directories looking for the file you were last working on. When a FracMan file not on the list is opened or saved, the last file on the list will be dropped and the most recent one added to the top of the list.

Exit

This item quits the FracMan program entirely. If you have a FracMan file open, the program will prompt you to save. If you click **Cancel**, the program will return to its previous state without quitting. If you click **No**, the program will quit without saving and any unsaved changes will be lost. If you click **Yes**, FracMan will ask you to specify what to save the file as and where. If you click **Save** without changing the file name or directory, the file will be saved and overwrite the previous saved version.

Edit Menu

The Edit menu allows you to modify object properties, select and delete specific objects, filter objects into subsets, organize how the objects are displayed, and edit object geometry.

Undo

To undo an operation, click *Edit > Undo* on the menu bar. The Undo function will be enabled only if the previous operation can be undone. To prevent memory leakage, not all processes support the Undo function. In general, operations such as object creation or deletion cannot be undone, while operations that transform an object, such as moving or scaling, can.

Redo

The Redo function is only enabled after the Undo function is used. It reverses the previous action of the “Undo” function.

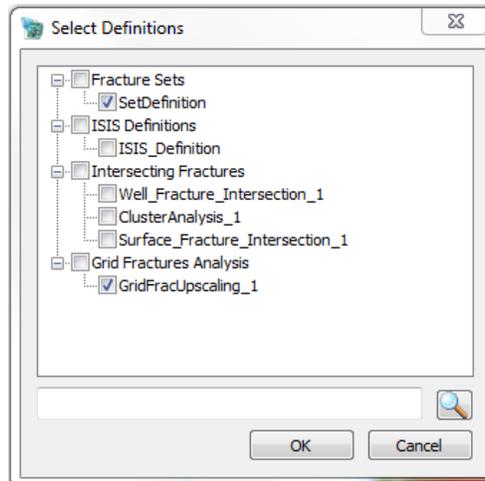
Delete Selected

This deletes the selected object or objects from the file. **Once an object is deleted, it cannot be recovered.** Objects can also be deleted by selecting them in the Objects panel and either hitting

Delete on the keyboard or clicking the  button on the toolbar.

Delete Definitions

You can delete multiple definitions at once by selecting “Delete Definitions” and selecting the definitions you want deleted.



Checking the root objects will select all objects of that type for deletion.

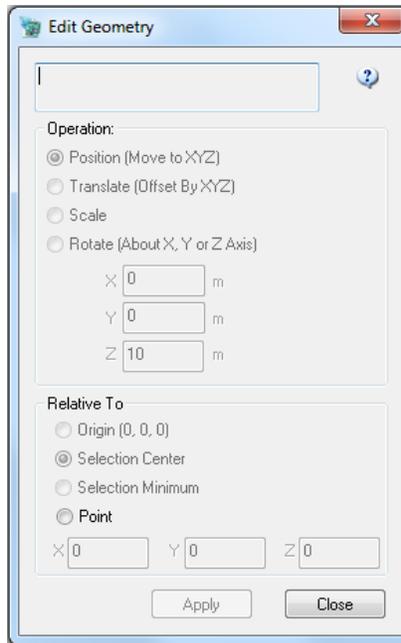
Delete Definitions cannot be undone.

Modify Properties

This activates the Properties window where you can edit, add, or remove properties from an object. This window is activated also by right-clicking on an object in the Objects window and selecting **Edit Properties**. See [Properties Bar](#) for more information.

Translate/Scale Object Geometry

This menu option allows you to edit the position, orientation, and size of any object in the project. Select the object whose geometry you wish to modify in the Objects window and select “Translate/Scale Object Geometry” under the Edit tab on the file menu. On some objects, such as wells, you may also right-click on the object name and select “Translate/Scale Geometry”. You may select multiple objects to edit at once; the change in scale, position, or angle will be applied to all selected objects. A dialog box will appear asking you to specify how the object is to be altered. You must first select the radio button next to the operation you wish to perform.



- *Position (Move to XYZ)*: You enter the new x-, y-, and z-coordinates of the object's center or minimum.
- *Translate (Offset by XYZ)*: Here you may specify how many units you wish to move your object. The entered value will be added to the object center's coordinates; negative values are acceptable.
- *Scale*: This allows you to change the units of the object from feet to meters or meters to feet using the Selection Center option, or to change the scale of the object in the given direction by the specified number for a more general function. FracMan takes the absolute value of the entered number as the scaling factor; thus, negative values are treated the same as positive.
- *Rotate*: Enter here the number of degrees about the given axis by which you wish to rotate the object. Rotation is clockwise and pivots around the point specified in the *Relative To* section. Positive or negative values are accepted.

Duplicate Object

This function allows you to select one or more objects and make an exact, independent duplicate of each object. Simply select the object(s) you wish to copy in the Objects window and click *Edit > Duplicate Object* from the menu bar. Duplicate objects will appear in the Objects window, with the same name as the original object plus the suffix "_Copy_1". The second copy of the same object will have the suffix "_Copy_2" etc. **Warning: large objects may take a long time to duplicate and may use a great deal of memory.**

Copy to Spreadsheet

This operation copies data from histograms, stereonet and result views to the clipboard in list format. With the histogram or stereonet windows open in the Visualization panel, select Copy to

Spreadsheet from the Edit menu. The result object may be selected in the Objects window and this option will be activated in the Edit menu.

Well

The Well menu item on the Edit menu provides the following sub-menu functions for [Wells](#) and [Well Logs](#):

- [Edit Well](#)
- [Resample Well Path](#)
- [Offset/Scale Intervals](#)
- [Add Interval Set](#)
- [Create Intervals From Objects](#)
- [Create Regions Around Wells](#)
- [Merge Well Logs](#)
- [Create Data Subset](#)
- [Create Log Subsets from Intervals](#)
- [Create Subsets from Categories](#)

Grid

The Grid menu item on the Edit menu provides the following sub-menu functions for [Grids](#):

- [Compute](#)
 - Compute Grid Cell Volume
 - Compute Grid Cell Height
 - Compute Grid Cell IJ Dimensions
 - Compute Grid Cell XY Area
 - [Convert Tensor To Principal Vector](#)
 - [Dipole Sonic To Elastic Material Properties](#)
 - [Compute Coordinate Frame](#)
 - [Create Brittleness Properties](#)
 - Compute Grid Mean Layer Height
- [Extract Surface](#)
- [Create Sub-Grid](#)
- [Merge Layers](#)
- [Mark Cells](#)
- [Flip Cell IJK Order](#)
- [Create Stress Properties](#)
- [Create Brittleness Properties](#)
- [Draw Slices / Subsets](#)
- [Refine Grid](#)

Surface

The Surface menu item on the Edit menu provides the following sub-menu functions for [Surfaces](#):

- [Edit Surface Normals](#)
- [Apply Texture Map](#)

- [Decimate Surfaces](#)
- [Weld Fracture Faces](#)
- [Clip Surface](#)
- [Convert to Fracture](#)
- [Compute Coordinate Frame](#)
- [Compute Surface Curvature](#)
- [Simplify Surface](#)
- [Merge Surfaces](#)
- [Trim Surface Outline](#)
- [Extract Surface From Selection](#)
- [Triangulate Selection](#)
- [Convert Region To Surface](#)
- [Convert Fracture To Surface](#)
- [Split Object By Property Values](#)

Tracemap

The TraceMap menu item on the Edit menu provides the following sub-menu functions for [TraceMaps](#):

- [Simplify Tracemap](#)
- [Merge Tracemaps](#)
- [Create Data Subset](#)
- [Create Subset From Selection](#)
- [Compute Segment Orientation](#)
- [Disconnect Traces](#)

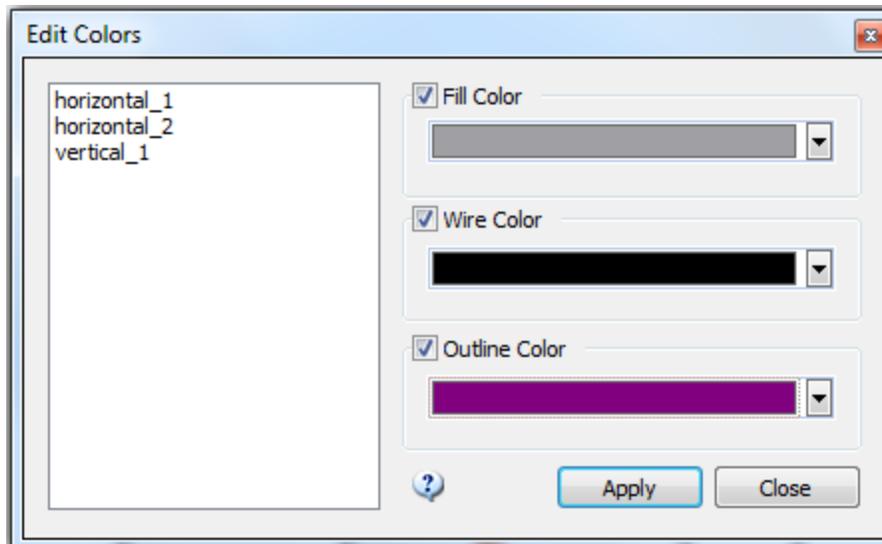
Point Data

The Point Data menu item on the Edit menu provides the following sub-menu for [Point Data](#):

- [Convert Object to Points](#)
- [Convert Selection to Points](#)
- [Merge Point Data Objects](#)
- [Delete Duplicate Points](#)
- [Apply Coordinate Operation](#)
- [Create Surface from Point Data](#)

Change Colors

You can change the colors of multiple selected objects at once, either setting them to a single color or setting each object to a different color.



The selected objects are shown in the list on the left. Select Fill, Wire or Outline color to change and either select a color, or choose “Cycle Colors” to set each object to a different color.

Organize Scene

As you get more objects and your file becomes increasingly complex, you may need help keeping track of things. To this end, FracMan has a set of handy tools to aid object organization.

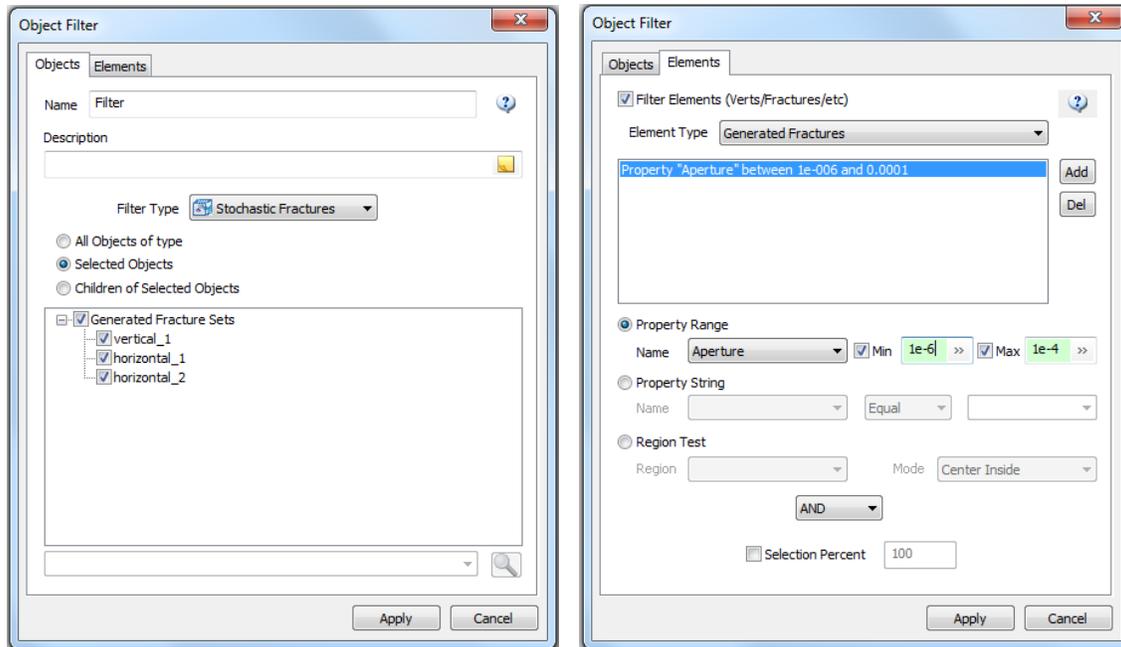
- *Insert Object Group*: creates a new, empty category. The group appears as a child of whatever object the user had selected upon execution of the command. The object’s name can be changed in the attributes window.
- *Sort Children A-Z*: puts all children of the selected parent object into alphabetical order
- *Sort Children Z-A*: puts all children of selected parent into reverse alphabetical order
- *Reparent Object(s)*: switches selected object’s parent to the specified object. Objects can also be reparented by dragging and dropping. Simply drag the object onto the name of the desired parent and release it. Multiple objects can be reparented at the same time.
- *Sanitize Model*:
- *Set FracWorld (P32) Region*: Set the FracWorld Region (same as in the Tools menu).

Select

This allows you to specifically select only objects of a certain type or fitting a certain set of parameters.

Create Object Filter

This allows you to create object subsets or reference sets based on one or more different properties. You may also filter the selected set(s) to a certain percentage of the original number. Available objects that can be filtered include Grids, Surfaces, Point Data, Tracemaps, Stochastic Fractures, Tessellated Fractures, Intersecting Fractures, Connected Fractures and Clusters.



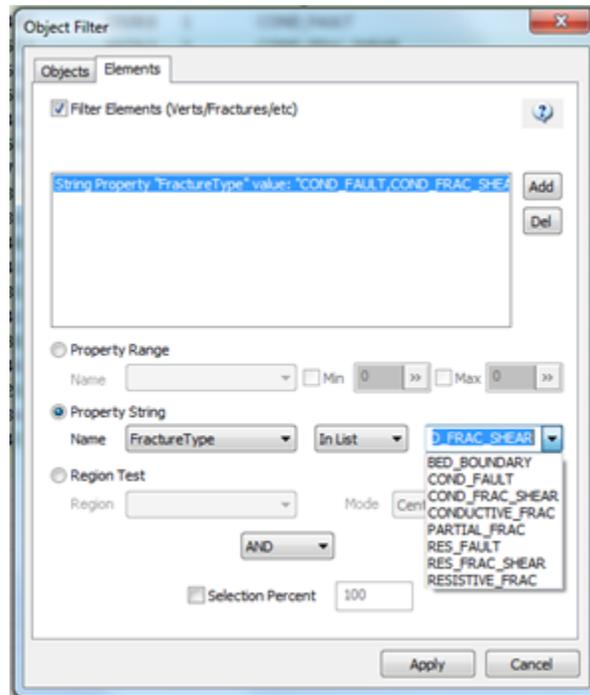
In the **Objects tab**, enter a **Name** for the Filter if you wish to save it to use again. Saved filters will appear in the **Definitions** window. Select one or more objects to include in your filter. Often, you will use a filter to group several fracture sets together, e.g. all the natural fracture sets in your model. In this case, just select the objects that you want included and click **Apply** and **Reference** to create a reference set including just these sets. If you want to select a subset of your objects using a Filter, then complete the **Elements tab** before clicking on **Subset** or **Reference**.

You build your list of selection criteria in the **Elements tab**. You may create a list of selection criteria or simply choose a **Selection Percent** to apply to the objects selected in the **Objects tab**. Check the box next to **Filter Elements (Verts/Fractures/etc)** to activate the fields on this tab.

Property Range: Use for filtering on numeric properties using minimum and/or maximum values. For example, the **Elements tab** shown above filters the “Aperture” property so that only fractures having an aperture between 1e-6 and 1e-4 are selected.

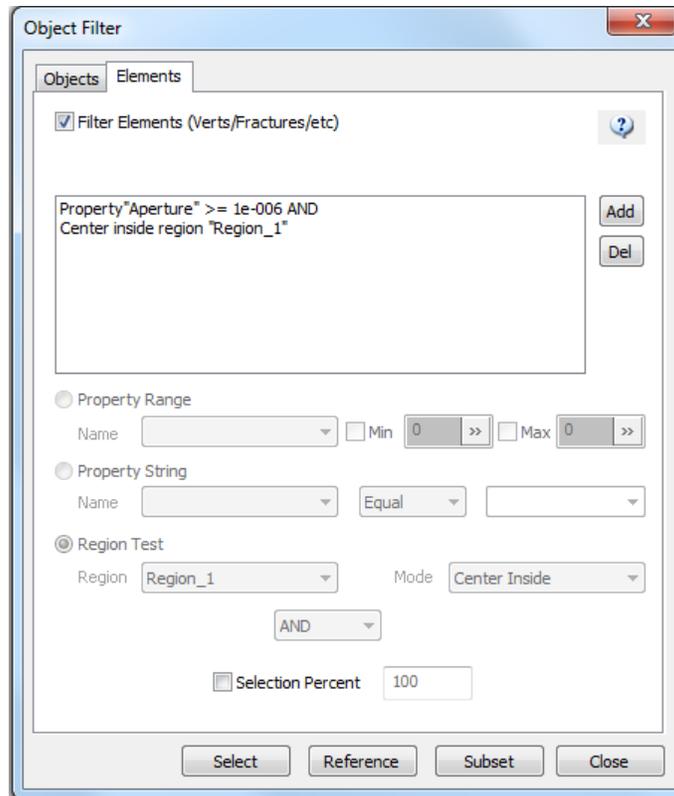
Property String: Use for filtering on string/text properties. For example, if a well log has the property “FractureType” with values like “BED_BOUNDARY, COND_FAULT, COND_FRAC_SHEAR”, you can use a filter to create subsets of the well log with fractures that only contain particular values of “Fracture Type”. Click on the radio button next to **Property String**. Select the property that you want matched from the dropdown list next to **Name** (this is “FractureType” in the example shown). Then select what type of string matching you would like to do, the choices are “Equal”, “Not Equal”, “In List” and “Not in List”. For the first two options,

“Equal” and “Not Equal”, just select one of the property values provided in the drop-down list on the right. For the two list options, type in a comma-separated list into that drop-down list containing the values that should be included:



Click **Apply** to add that filter criteria to the list shown in the text area on this tab.

Region Test: Use a Region Test criterion to select object with either their “Center Inside” or “Center Outside” of the selected region.



Applying Multiple Criteria: To use more than one filter criteria, first create the first filter using one of the three options, “Property Range”, “Property String” or “Region Test”. Select either “AND” or “OR” from the drop-down list below “Region Test” and click on Apply. This will add the first criteria to the filter and will be displayed in the text area on this tab. Then click on the Add button to the right of the text area to create a second filter criteria. You will see either “AND” or “OR” appear after the first filter criteria. Create your second criteria similar to how the first was created.

Selection Percent: Whether or not you have selected a property to filter on, you can choose a smaller subset to select by checking Selection Percent and entering a number between 0 and 100. For example, you may want to create a subset of 10% of your fractures, randomly selected. For this case, you would select the fracture sets on the **Objects tab**, check the **Filter elements** box at the top of the **Elements tab** and check **the Selection Percent** checkbox entering the value 10 into the field on the right. Click **Apply** to save the definition and Subset to create a new Fracture set containing only 10% of the original fracture set. If you have also applied a filter, the filter will first be used, and then the selection percent will be applied to that selection. This function works a bit differently for the Reference or Subset buttons. A Reference object will have exactly the entered Selection Percent while the Subset will apply this Selection Percent as a probability associated with each element, so will have a resulting object that is close to having the specified Selection Percent, but not necessarily the exact number.

Changing the Criteria Definition: To edit or delete a criterion, select it from the text area box. Once it is selected, you can change the values and click on **Apply**, or click on the **Del** button to remove it. Click **Apply** to save the definition when you are done editing.

Apply: Click on this button to save the current definition. It will appear in the **Definitions window** under “Element Filters” with an icon showing what type of object it is defined for.

Select: This will select the filtered objects in the **Objects window** and change their display in the **Visualization window**. It does not create a new subset or reference set. This is very handy for previewing your filter.

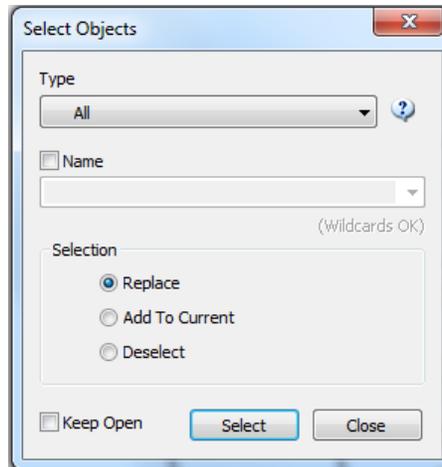
Reference: Click on this if you want to create a new Reference Set based on your filter. See the Manual section [Create Reference From Selection](#) for more information about characteristics of Reference Sets. This will create a new top-level reference object, it is not a child of the original set.

Subset: This label is somewhat misleading as it also creates reference sets. This will create one or more new reference sets that are children of the original objects selected with the filter. It will appear as a child of the original set (if only one was used). If multiple objects were included in the filter, there will be a child object for each of them, each containing a subset of the parent set. If you want to convert this fracture reference set child object into an independent object, use the Fracture->[Convert to Fracture Set](#) functionality.

Close and Cancel: Close the Object Filter dialog without saving. This is the same as clicking on the  icon at the top right corner of the dialog.

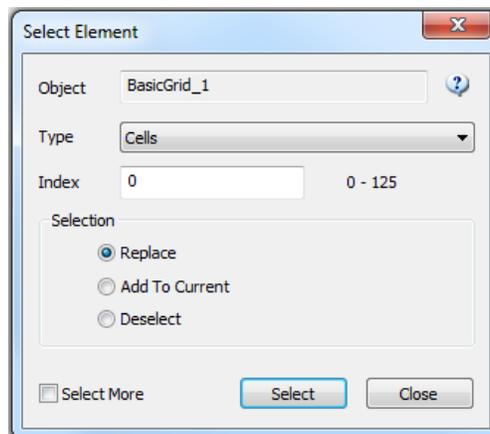
Select Objects

Select Objects brings up a dialog allowing you to select in the Objects window only a certain type of object or objects with a specified name. “Wildcards OK” means that you can also search for objects with a specified string in the name with a placeholder for other characters. An asterisk means any set of characters may fit into the space occupied therewith; a question mark means any single character may occupy that slot.



Select Element By Index

Select Element By Index brings up a dialog that allows you to select an element (vertex, face, cell etc) by its numeric index and inspect its properties location etc in the attributes window.



Copy Selected Object Names

Select Faces, Vertices, and Cells

Select Faces, Vertices, and Cells use the same mechanism as the Object Filter to select certain portions of an object, thus allowing you to move or modify only the selected elements rather than the entire object.

Select Lines

Select Lines is used to select traces from a tracemap object. These are also available from the Tool Bar.

Area Select Mode

Select all objects within a user-specified area. Drag the mouse to choose the area. All objects that partially or fully reside in that projected area in 3D space are selected.

Translate/Rotate Selection Mode

Switch between translate and rotate modes to aid in object selection.

Create Reference From Selection

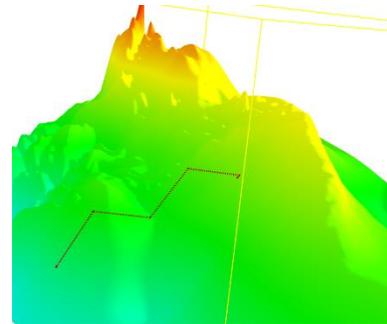
After selecting some objects in the scene, save them as a reference object in the project.

Draw Traces Mode

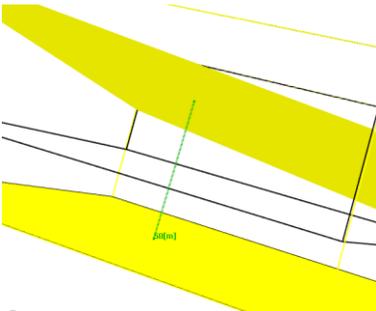
It is often convenient to be able to draw traces onto a surface.

This menu item will enter trace drawing mode on the currently selected surface. Press 'TAB' to start a new trace, press 'ESC' to cancel the current trace. When in this mode you can click/drag on a surface to define trace segments.

Select 'Save Current Tracemap' to store the defined traces as a new object.



Measure



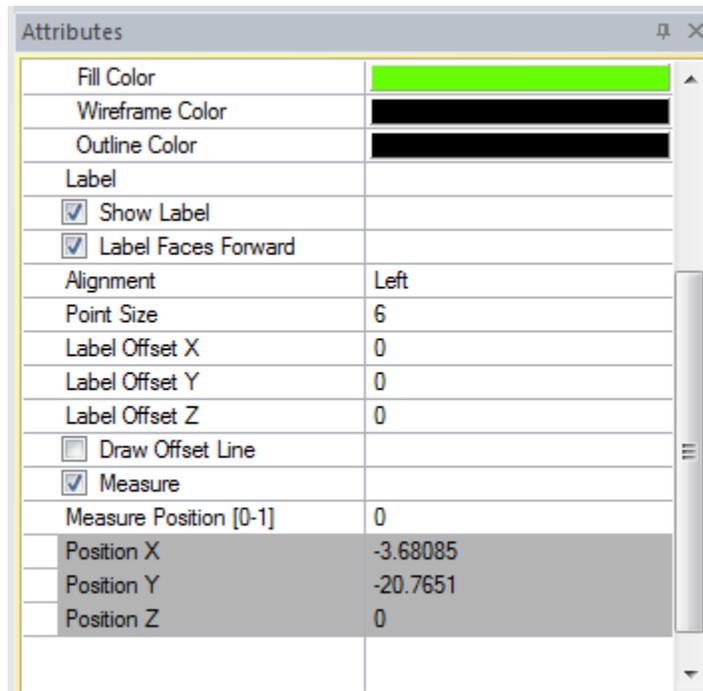
Measure distances between points on a single surface or between multiple surfaces.

Select one or more surface objects and select 'Measure', left click anywhere on any of the surfaces and drag, the measurement will remain until you change modes or click. Press the 'TAB' key will store the measurement into your model as a pair of point markers.

You can change the way the measurement is displayed by selecting the child point marker and setting options in the attributes bar.

Measure: Check this box to make this point a measurement and to display the distance to the parent point marker.

Measure Position [0-1]: Use this to move the displayed distance from one end of the line to the other (or more likely somewhere in between).



Set Model Origin

FracMan has an internal “model origin”, allowing it to treat the objects in the model as if they were centered around (0,0,0). This is only an internal function and will not affect displayed or exported model coordinates. If you wish to change the model origin, you may do so by selecting *Edit > Set Model Origin...* on the menu bar. This can be especially useful when the model is stored using UTM coordinates (see section on [Handling UTM Coordinates](#) for details).

You may change the origin to:

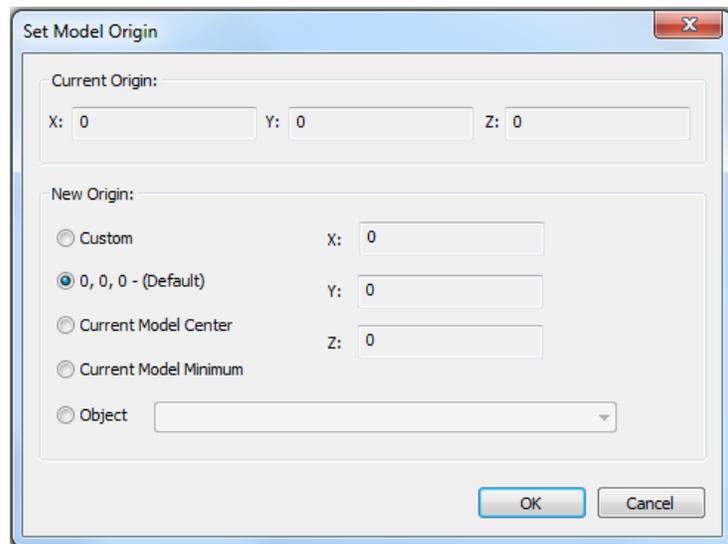
A custom point, whose coordinates you must specify in the X, Y, and Z boxes on the right side of the dialog box,

The point with coordinates (0,0,0), which is FracMan’s default origin,

The center point of all the objects currently contained in your model, which FracMan will automatically calculate and input for you,

The point in your model whose coordinates are smallest, or

- The coordinates of a given point marker object or well.



When you have chosen your desired relative origin, hit **OK**. There will be no visible changes; however, when you center your “scene” it will center on your selected origin. This is particularly handy when translating objects great distances, as FracMan does not automatically update the model origin to match the model center.

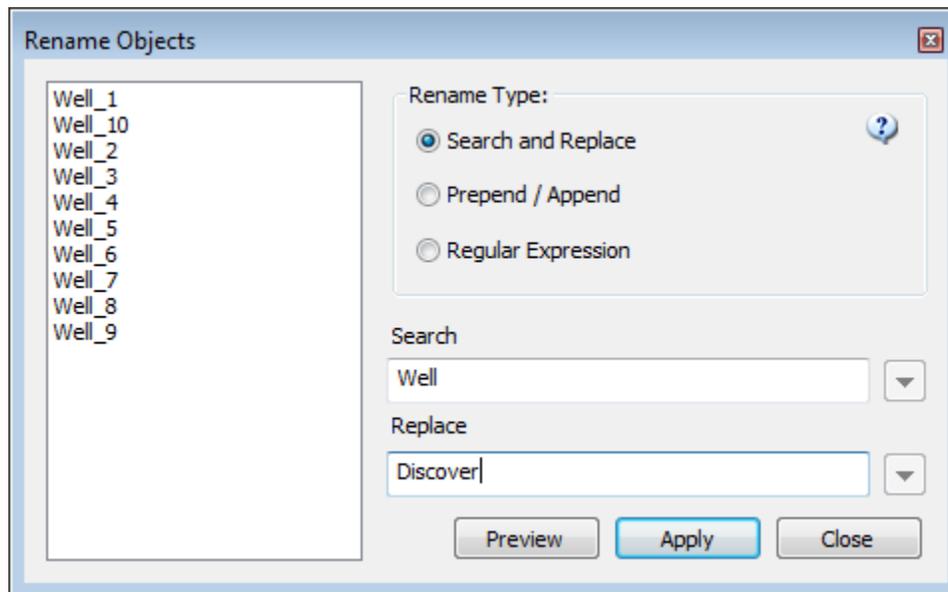
The set model origin operation moves the global origin and adjusts the position of all objects accordingly. If you want to change the model position, you can move the model to a new global origin by selecting the **Scene** object in the objects bar and selecting “Translate / Scale Object Geometry” from the “Edit” menu. This will adjust the global origin without updating the position of objects relative to that point.

Rename Objects

Select the objects that you want to rename in the Object window, then select “Rename Objects” from the Edit menu.

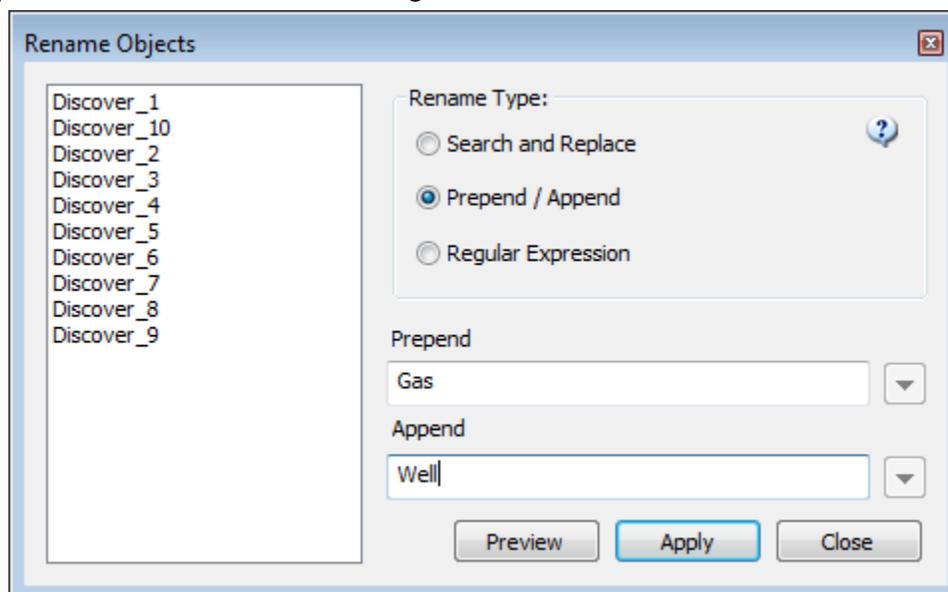
Search and Replace

The Search and Replace feature of the Rename Objects dialog can be used to replace part of the name of an existing object with a new name.



Prepend/Append

The Prepend/Append will rename the object with a specified name in front of the original name and a specified name at the end of the original name.



Regular Expression

The Regular Expression box allows the User to use more complex naming conventions.

View Menu

This menu allows you to select which panels and panes FracMan displays. Some of the options show available panels for display while others affect details of what gets displayed in each panel. An orange outline around the icon or a check mark next to a panel name signifies that it is already being displayed. In order to show a panel that is hidden, just click on its name in this list.

- **Objects:** panel that lists all objects in your FracMan file.
- **Attributes:** panel that allows you to see and modify the properties of a selected object.
- **Definitions:** panel that lists all definitions you have generated in your file, including fracture set definitions, critical stress analyses, stored views, and filters.
- **Properties:** panel that allows you to manage the selected object properties not listed as attributes.
- **Legend:** panel containing a colored bar keyed to number values that allows you to visualize properties graphically at a glance.
- **Model Scale:** a box with scale markers that appear in the Visualization window behind the entire scene to give a scale. See Manual section on [Model Scale](#) for more detail.
- **Notes:** panel that allows you save comments about the object currently selected in the Objects window.
- **Timeline:** panel that allows you to create an animation for visualizing a series of properties on an object (like well test results) or for visualizing a single property where the values represent time (like hydraulic fracturing).
- **Log:** panel that tracks the user's actions in FracMan, recording commands as they are enacted in macro-language format.
- **Lock Docking:** panels can be locked in place to prevent inadvertent moving around.
- **Toolbars**
 - **Toolbar:** contains pictorial icons that serve as shortcuts for common commands; usually shown at top of screen just below menu bar.
 - **Random Seed Toolbar:** allows easy access to view/set random seed.
 - **View Toolbar:** allows easy access to change the Z scale shown in the scene.
- **Menu Bar:** uncheck to hide the menu bar; to get the menu bar visible again, use the shortcut Ctrl-Shift-M or right-click on the top border of the application and select Menu Bar.
- **Status Bar:** bar at bottom of FracMan main window that gives a brief description of what a command does when the user mouses over that command.
- **Colormap:** opens same options for color visualization as from right-clicking on the Legend panel.
- **Set Background:** opens a dialog window to set the background color of the Visualization window. See Manual section on [Set Background](#) for more detail.
- **Set Fracture Percent:** use to choose what percentage of fractures to display in the visualization window. See Manual section on [Set Fracture Percent](#) for more detail.

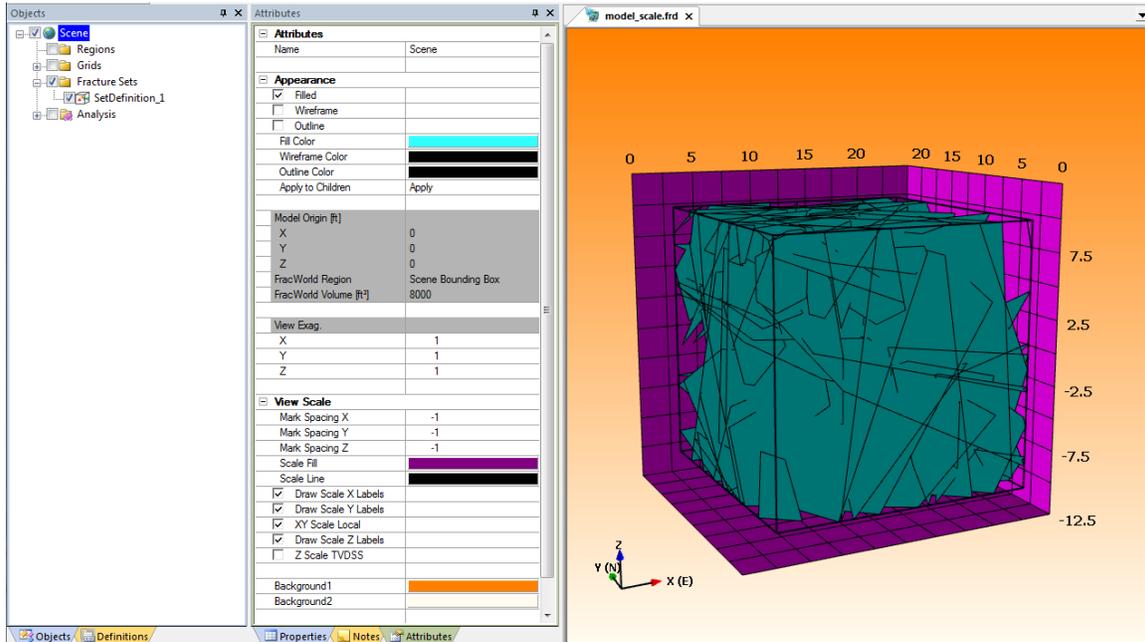
- **Store Current View:** saves your current view of the Visualization window's contents as a Definition. See Manual section on [Store Current View](#) for more detail.
- **Visibility Region**
 - Set [Visibility Region](#)
 - Clear Visibility Region
- **Show Full Screen:** use this option to have the Visualization window fill the monitor screen.
- **Show View Legend:** toggle this option on/off to show or hide the legend in the Visualization window.
- **Show View Axis:** toggle this option on/off to show the orientation axis in the Visualization window.
- **High Resolution Screen Capture:** take screen shots at higher than screen resolution. See Manual section [High Resolution Screen Capture](#) for more detail.
- **Zoom:** The sub-menu options under Zoom manipulate the view in the Visualization window. These options are all available as Toolbar buttons.
 - Zoom To Scene Extents
 - Zoom To Selection
 - Align View
 - Perspective / Orthographic
 - Plan View
 - North View
 - South View
 - East View
 - West View
 - Enable Spin
- **Enable Transitions:** show intermediate views in the object window when changing from one view to another, thus allowing a smooth visual transition from one viewpoint to another as when zooming in or out or rotating the model. Disable this option if the rendering speed is slow to improve performance.
- **Draw Vectors:** opens the [Visualize Arrows](#) dialog so that vectors can be displayed.
- **Hide Vectors:** Hides any currently visible vectors, this is the same as unchecking the Visible property in the [Visualize Arrows](#) dialog.

Model Scale

Parameters controlling the model scale can be accessed from the Attributes panel of the **Scene** object in the **View Scale** section.

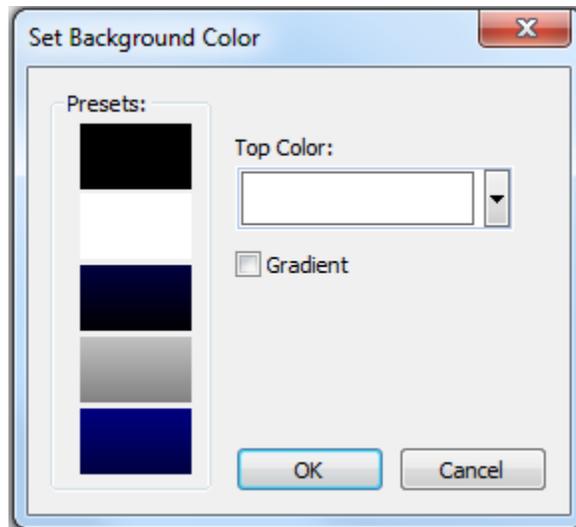
- **Mark Spacing X, Y, Z:** control the grid spacing on the model scale box in three directions. The value entered is the space between grid marks in the project unit of length. A value of -1 lets the program decide on an appropriate spacing.
- **Scale Fill:** color of scale box
- **Scale Line:** color of grid lines and axis labels
- **Draw Scale X,Y,Z Labels:** toggle display of scale labels

- **XY Scale Local:** use local coordinates (especially useful when data is in UTM)
- **X Scale TVDSS:** use total vertical depth instead of Z values



Set Background

This command allows you to select the Visualization window’s background color. Choose a top and bottom color or select one of the presets from the list on the left.



Set Fracture Percent

When you generate an extremely large fracture set, FracMan has a hard time displaying every single fracture. Instead, it will show only a percentage of the fractures. When less than 100% of fractures are displayed, a notification will appear at the bottom of the Visualization window:

12% of Fractures Displayed

You can choose what percentage of fractures FracMan shows. Enter any integer value from 0 to 100 in the blank next to “Percentage of Fractures Displayed:”. Then hit OK. The fewer fractures

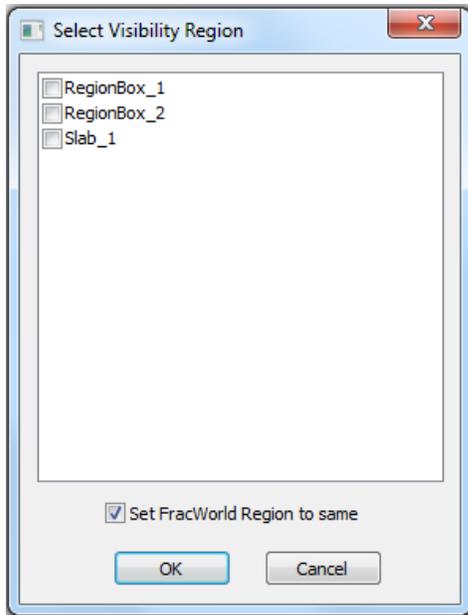
you have displayed, the faster FracMan’s graphics will be able to draw them as you rotate or translate the fracture set.

Store Current View

This saves your current view of the Visualization window’s contents as a definition in the Definitions pane. When you wish to return to the saved view, right click on the name of your saved view in the Definitions pane and select “Execute”. This will return the Visualization window’s display to the saved perspective. You may rename your saved view by right clicking on it and selecting “Rename” or you may copy it by selecting “Duplicate Definition”. “Store” updates the saved view to the current view. Once you have saved a view, you may return to it at any time.

Note: the stored view only saves the angle, zoom, and perspective of your current Visualization view. It does not save the current state of the objects in the window. A restored view will not restore objects that have since undergone change to the state they were in when you saved the view.

Visibility Region



This sets the selected region as the “visibility region”, hiding everything outside of that region so that only the contents of the chosen region are displayed. For instance, if a fracture intersects the visibility region, only the portion of the fracture inside the region will be shown. **Set Visibility Region** allows you to select which region you wish to be the visibility region from a list of all available regions. Simply check the box next to the region you wish to be your visibility region and hit **OK**.

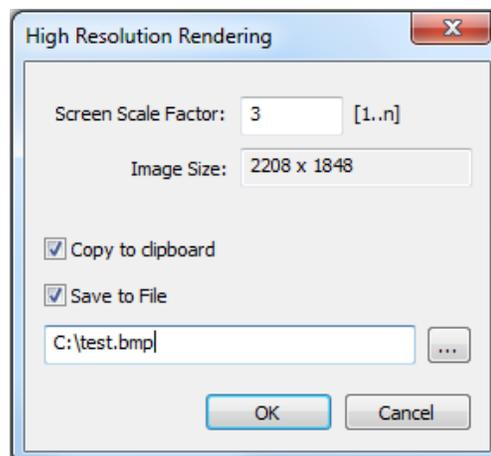
The “Set FracWorld Region to same” option ensures that fracture intensities are calculated based on the selected region as well. (The FracWorld Region is discussed in greater detail below under the [Tools Menu](#).) **Clear Visibility Region** resets the visibility

region to the entire scene so that all displayed objects are again visible, no matter their location.

High Resolution Screen Capture

You can now take screen shots at higher than screen resolution.

These images are useful for publication where high dpi images are required. Select “High Resolution Screen Capture” and enter a scale factor – the resulting image will be captured at this many times the size of the current window. You can save the image to the clipboard or save it directly to file in a number of different image formats. For best results you should also select “Enable Multisampling” from the “Display Options” page of the options dialog.

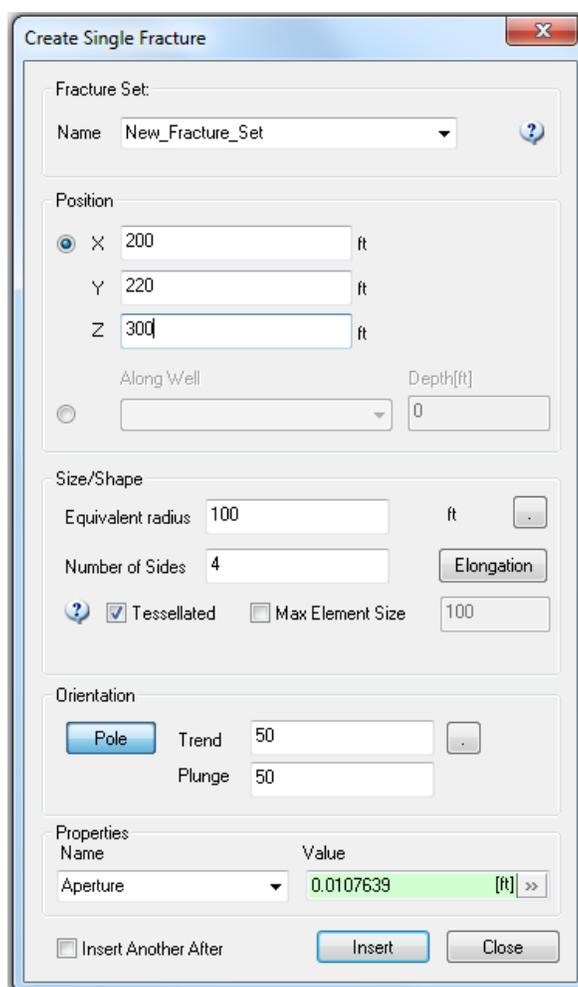


Fracture Menu

Fracture allows you to create and edit fracture objects within your project.

Insert Single Fracture

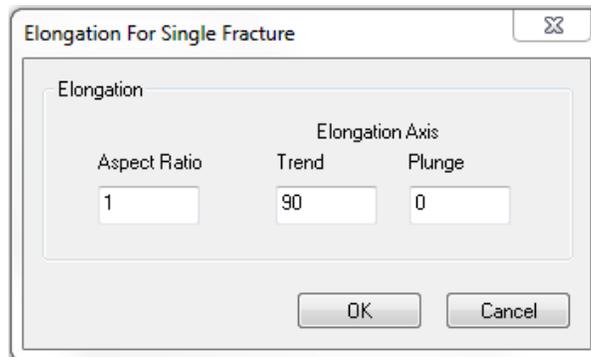
This allows you to place a single fracture of a specific size and orientation at a specific spot in the scene. When you select this command, FracMan brings up a dialog box asking you to define certain properties for the fracture.



Fracture Set: You can select an existing fracture set to add this fracture to or enter a new name and a new set will be created.

Position: You can either specify X-, Y-, and Z-coordinates for the fracture center, or you can locate your fracture along a given well at a certain measured depth. The well will intersect the fracture at the fracture's center.

Size/Shape: Here you must specify the equivalent radius of the fracture and the number of sides you wish it to have. Because the fracture is not circular, the radius specified is that of a circle with the same area and center as the fracture. Pressing the '.' Buttons next to radius or Trend will compute a random size or orientation value for quickly creating a number of fractures. If you click the **Elongation** button, you are given another dialog:



The **Aspect Ratio** specifies how much longer you wish the fracture to be than its width. For instance, a fracture twice as long as it is wide has an aspect ratio of two. The **Elongation Axis** is the axis along which you wish the fracture to be stretched.

Choosing the **Tessellated** option will create a single fracture composed of triangular elements whose size is defined by the **Max Element Size** option box. Each tessellated fracture is a separate Surface object child of the parent fracture set and can be interacted with. Un-checking the Tessellated option will create a single planar polygon fracture, similar to those created by Geocellular or Geometric fracture set generation options. The fracture child objects generated by this method **cannot** be interacted with directly, though they can be selected and copied to subsets using fracture filters. Tessellated fractures occupy significantly more memory than planar polygon fractures. They should only be used in the situation when the fracture generated needs to be able to be selected, moved, or edited.

Orientation: You can specify either a pole trend and plunge or a dip and strike for your fracture, which will determine its tilt relative to horizontal. To toggle between pole trend and plunge and dip and strike, click the button in the left center of the dialog box. When the button is raised, it says **Pole**, indicating that orientation is to be given in pole trend and plunge format.

When the button is depressed, it reads **Dip**, indicating orientation is to be given in dip trend, or strike, and dip plunge, or dip, format. Click on the “.” Button to generate a random orientation.

Properties: FracMan allows you to specify the Aperture, Transmissivity, and Storativity of your fracture. Click the drop-down menu under “Name” to select which property you wish to change. Then enter the desired value in the box under “Value”. Toggling between properties does not reset their assigned values.

Once you have defined your fracture’s characteristics, hit **Insert**. This will generate the fracture. It will appear in the Visualization window and in the Objects panel under the parent group “Fracture Sets”, under “Single Fractures”. Single fractures are discussed in greater detail in the [The Theory Behind Fracture Generation](#) section of this manual.

Define New Set

This allows you to create a set of one or more stochastic fractures based on a given suite of parameters. There are six different methods of defining your set:

[Geometric](#) – generated within a region

[Geocellular](#) – generated based on a grid

[Geologic](#) – generated based on a fold model grid or a fault model grid

[Tracemap](#) – generated along traces on a tracemap

[Stratigraphic](#) – generated between surfaces

[Layered Fracture Spacing Model](#) – generated based on layers of a grid

[Stress](#) – generated based on stress properties of a grid

Lava Flow Field --

[Insert Single Fracture](#) – same as Fracture -> Insert Single Fracture

[Create From Point Data](#) – generated from point data sets

Fracture Set Definition Summary – displays a summary screen of all fracture set definitions. At a glance, this allows the user to look at all input parameters for all fracture set definitions that are being used. If the user clicks on one of the fracture set names in the report, the actual fracture set definition will be brought up so it can be directly edited.

[Condition To Traces](#) --

[Create Pseudo Matrix Fracture](#) --

The **Define New Set** functions are discussed in much greater detail in the “[Fractures and Fracture Sets](#)” section of this manual. Once you have defined your fracture set and clicked **OK**, a new entry will appear in the Definitions panel under the parent “Fracture Sets”. No fracture sets will actually be created until you generate a fracture set based on the definition.

Edit Selected Set

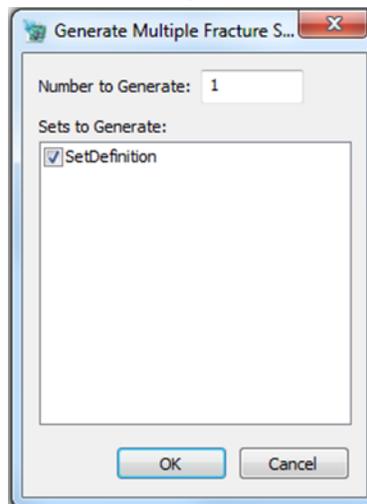
This allows you to edit a given fracture definition. Select the set definition in the Definitions panel; then hit this command and FracMan will reopen the definition specification dialog, allowing you to alter it. You may also reopen fracture definitions by right-clicking on the set name and selecting **Edit Definition**.

Generate Selected Set

This generates a fracture set based off of a given definition. Select the set definition you wish to generate in the Definitions panel; then choose this command from the file menu and a new fracture set will be generated stochastically, matching your specifications. A new object will appear in the Objects panel under the parent “Fracture Sets”, and the fracture set will appear in the Visualization window. You can also create a new set by right-clicking on the definition name and selecting **Generate Set**. If you select an existing fracture set in the Objects panel and execute this command, a new fracture set will be generated to the same specifications as the existing set.

Generate Multiple Sets

Select “Generate Multiple” from the fracture set definition left-click menu and you can generate multiple iterations / multiple definitions in a single pass.

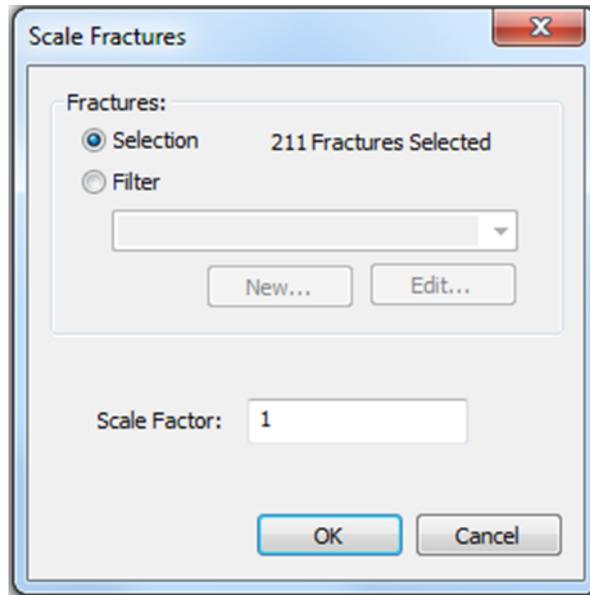


Just specify the number of sets you wish to generate and specify which definition(s) you wish to use.

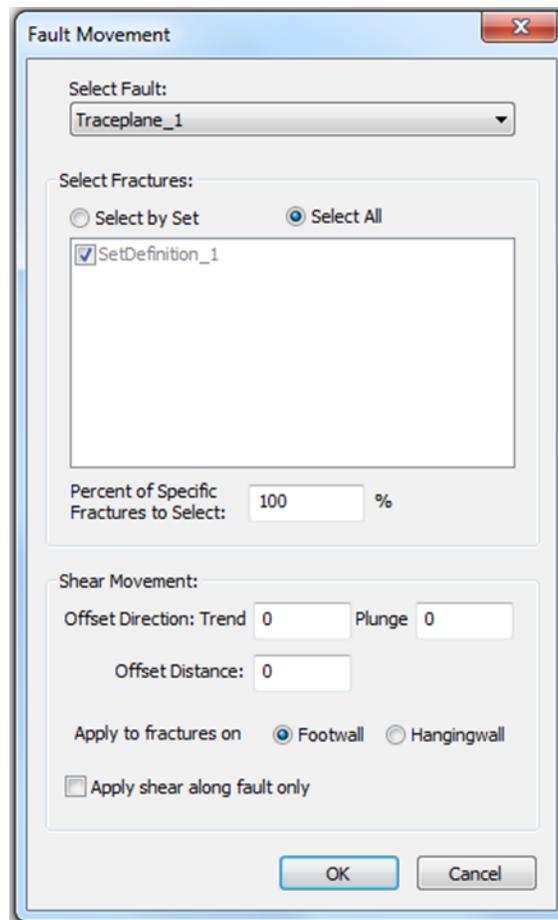
Edit

Scale Fractures

You can quickly make fractures bigger or smaller with the scale fractures operation. Select the fracture sets or create a fracture filter, then specify the amount you want to scale by. The scale operation does not change the location of the fractures so it should not be used for converting between units (feet to meters for instance). This function also does not affect any of the fracture’s property values.



Fault Movement



The Fault Movement operation provides the ability to simulate discontinuous deformation of a fractured rock mass by introducing a linear shear displacement across the fault surface. The fractures lying on the moving side of the fault are displaced.

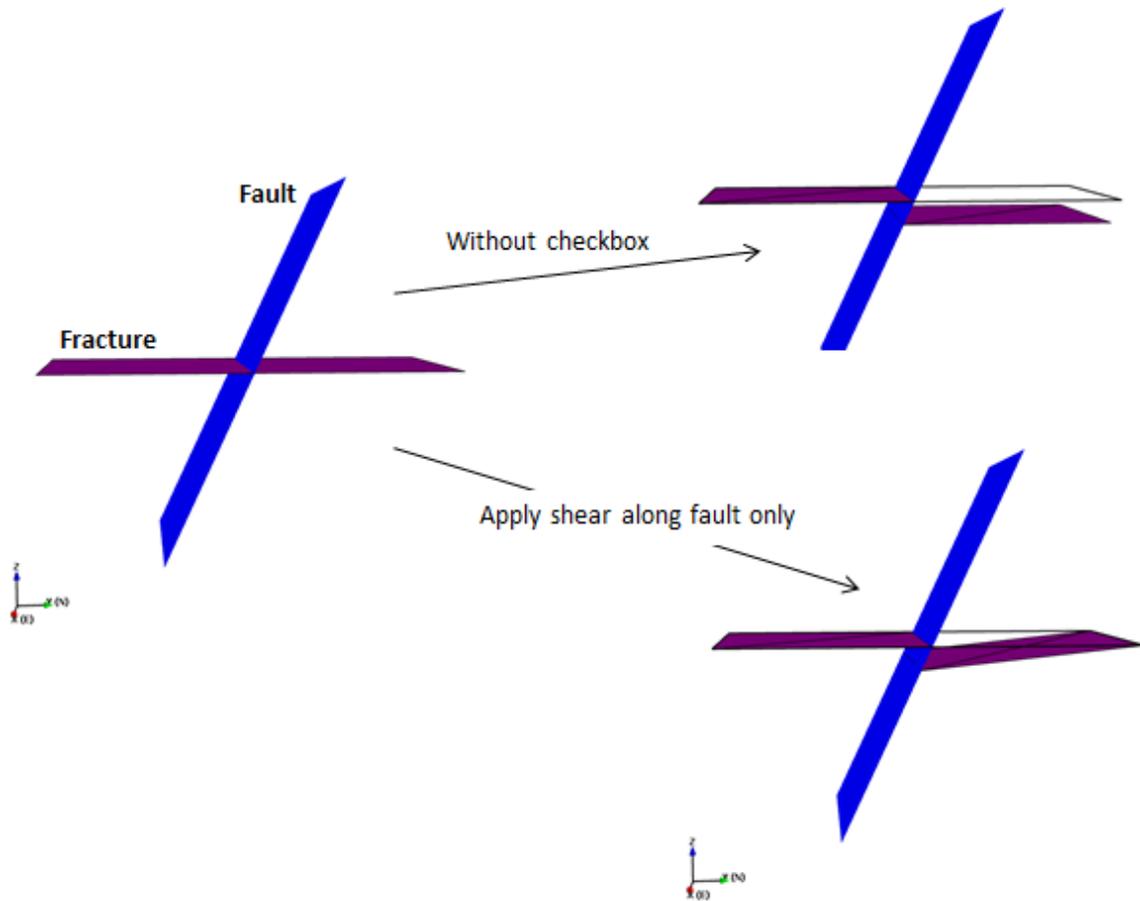
In order to use the function you must have a surface in your file that will be used to simulate the fault plane. You may access the feature by selecting **Fracture > Fault Movement** from the file menu. A dialog box will appear asking you to define some parameters for the simulation.

The dropdown menu beneath “Select Fault:” lists all surfaces in your file. Choose which one you wish to use as the fault along which movement is to occur. If you wish to offset all fractures that intersect the fault, check “Select All”; otherwise, check “Select by Set” and pick the fracture sets you wish to offset from the list provided. You may also offset only a select percentage of the given fracture set(s) by entering the desired percentage in the appropriate box.

Then specify the shear direction, that is, the trend and plunge you wish to offset the fractures along, with respect to the fault surface. In the box next to “Offset Distance:” specify how far you wish to shift your fractures.

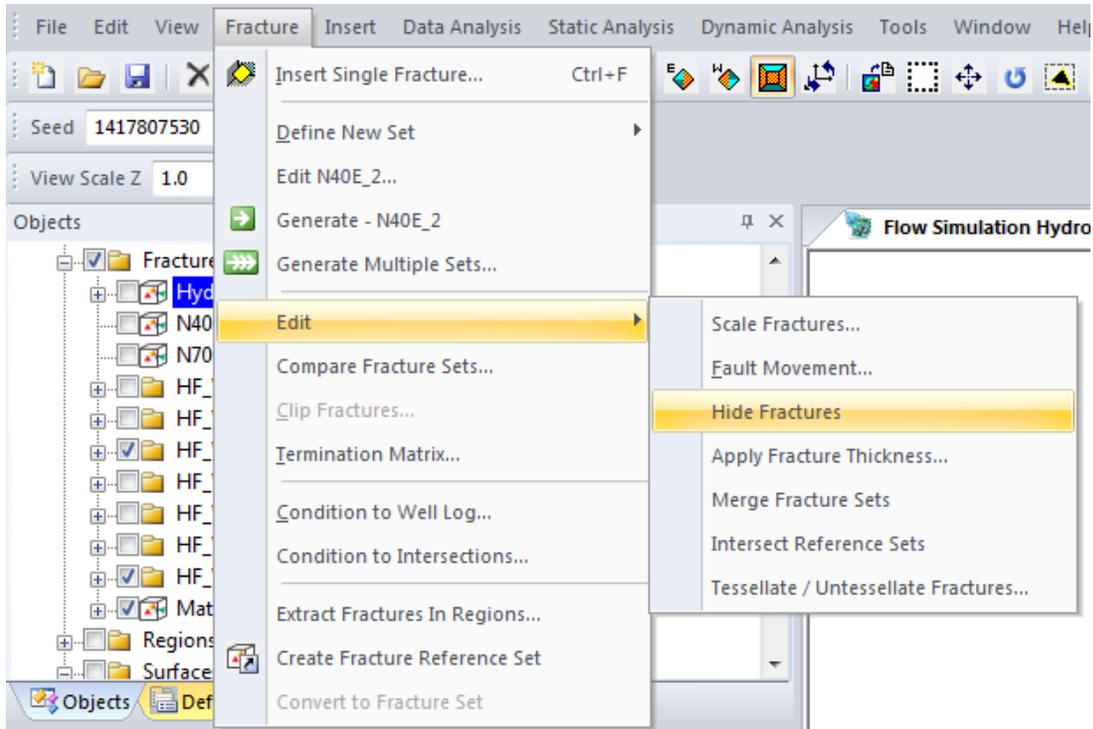
Choose which side of the fault you wish to show offset. Select “Footwall” to show movement on the portion of the fractures below the fault surface, or if you wish the portion of the fractures overhanging the surface to show movement, select “Hangingwall”.

Apply shear along fault only: if checked, only the intersection of the fracture with the fault gets moved.



Hide Fractures

You can hide/unhide fractures from visualization. Select the fracture sets and then tick/untick Hide Fractures. This function does not affect any of the fracture's property values.



Merge Fracture Sets

Multiple fracture sets can be merged together by selecting them and selecting “Merge Fracture Sets” from the “Fracture” menu. All the fractures from the selected sets will be combined into a single set.

Intersect Reference Sets

It can be useful to know which fractures from one set intersect another set. FracMan provides the Intersect Reference Sets function available from the Function/Edit menu to create just such customized reference sets. The “Compute Fracture Boolean Operation” dialog has fields to select the fracture sets of interest and the intersection criteria.

Name: Enter the definition name, a default will be provided.

Description: Optional comment field.

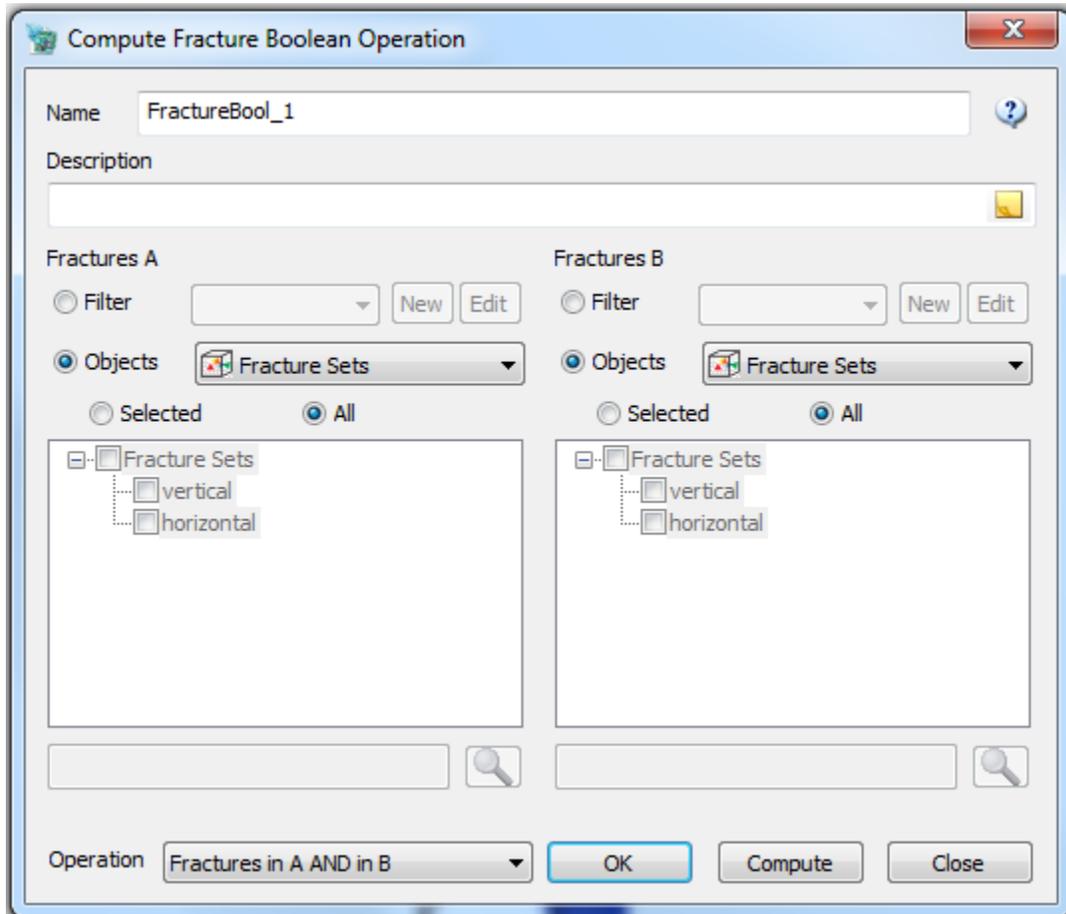
Fractures A and Fractures B: Select one or more fracture sets. These can be already present in the model or newly created as subsets using a Filter. If selecting fracture sets already present in the model, you can select the type of fracture set from the categories:

- Fracture Sets
- Stochastic Fracture Sets
- Tessellated Fracture Sets
- Intersecting Fracture Sets
- Connected Fracture Sets
- Clusters
- **Search Fields** (below Fractures A and Fractures B): Use these fields to help locate fracture sets when there are a lot present in the model. Enter a search string and click

on the magnifying glass icon. Any matches will be highlighted in the list, but not checked. You must check the fracture set to select it for use in the comparison.

Operation: four different comparisons are available

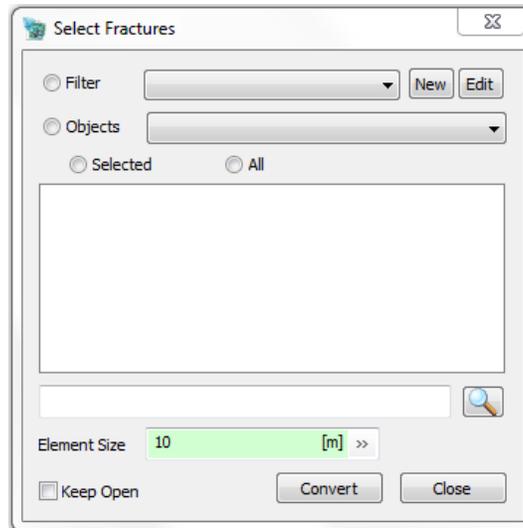
- Fractures in A AND in B
- Fractures in A OR in B
- Fractures in A or B but not both (XOR)
- Fractures in A but not in B (NOT)



Note: the Intersecting Fractures sets are not independent objects; if any of the fracture sets from which it was derived are deleted, the “reference sets” will be deleted as well. Intersecting Fractures are stored as references to the original objects rather than objects themselves. This uses less memory than full fracture sets and allows you to modify property values of subsets of the original fractures easily. If you want to save the results as a distinct object, use the [Convert to Fracture Set](#) function from the Fracture menu bar.

Tessellate / Untessellate Fractures

To Tessellate a Stochastic Fracture Set, select the object either by using a Filter or the Object selection pane, enter an approximate maximum **Element Size** and click on the **Convert** button.

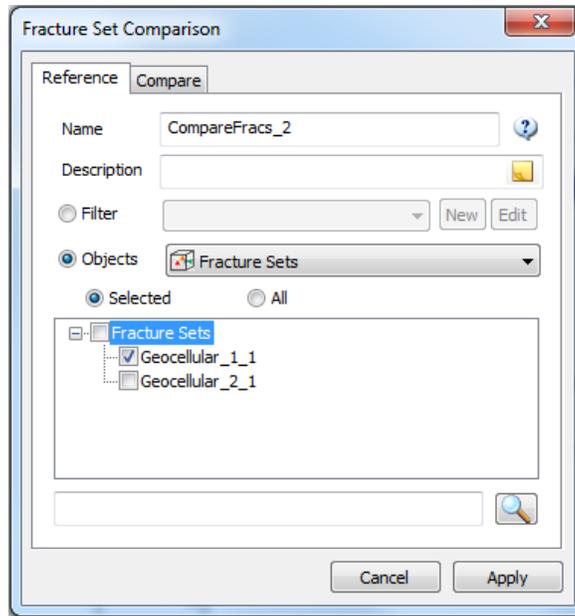


To untessellate a Tessellated Fracture set, select the object either by using a Filter or the Object selection pane and click on the **Convert** button. The **Element Size** field is not used. This function can be useful in reducing the complexity of a mostly flat, but highly tessellated fracture.

If you want to keep the Select Fractures dialog box open to run this function multiple times, check the **Keep Open** box before clicking on the Convert button.

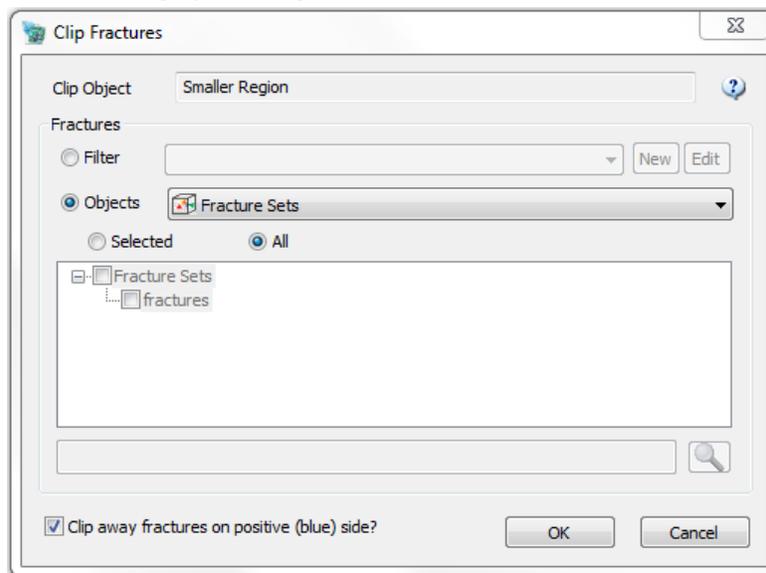
Compare Fracture Sets

This function is available from the Fracture Menu. Select one or more fracture sets to use as a reference set on the first screen and one or more fracture sets to compare on the second tab. Click Apply to save the Comparison definition (this will appear on the Definitions panel under “Intersecting Fractures”). Click Run to run the comparison and generate a report which shows all the properties in a side-by-side comparison of their sum (Total), min, max, mean, and deviation when appropriate. This report is saved as an analysis object and can be viewed at a later time or saved to file.



Clip Fractures

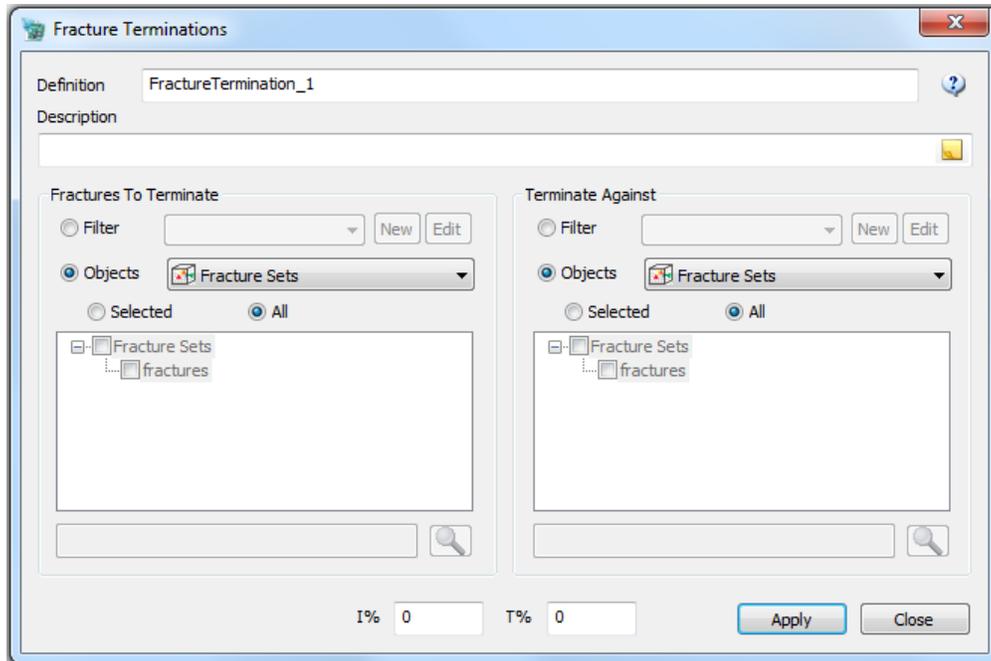
This feature allows you to trim a fracture set so that it does not extend beyond the boundaries of a given grid or region. Select which grid you wish to clip your fractures to in the Objects panel. You may access this command as well by right clicking on the desired grid and selecting **Clip Fractures**. This will bring up a dialog box:



If you do not want all fractures in the file to be clipped, you may create an Object Filter to discriminate between fracture sets or to establish subsets to be clipped based on some property of the fracture. [Object Filters](#) are explained in greater detail in the Edit Menu section. Select the button next to “Filter:” and either select a preexisting filter from the dropdown menu to the right, or click **New**. This will open an Object Filter creation dialog box, allowing you to specify which fractures you wish to clip.

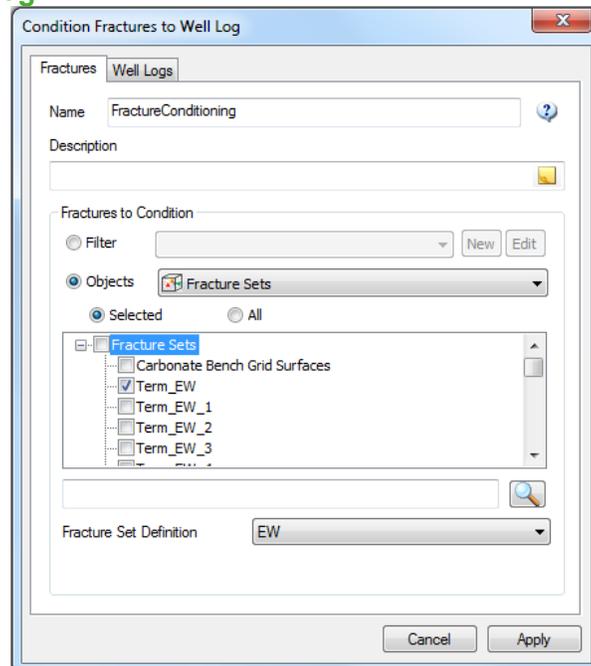
Termination Matrix

This function will terminate one fracture set to one or more other fracture sets. The Fracture Terminations dialog box will appear:



Enter a percentage value for "I" and "T" intersections and the code will attempt to clip fractures so that the resulting fracture set will match intersection percentages. The definition will be saved under the Definitions window by the name that you specify here. You may also apply the matrix to a filtered set.

Condition to Well Log



This function requires one or more Wells with associated Fracture Logs, and a Fracture Set which intersects the Well to proceed. The Fracture Set should also have a saved Definition in the project file. The function compares the intersecting fractures in the set to the FMI fractures in the well log(s) and alters the intersecting fractures to match the orientations and depths of the well log fractures. Fracture size generally remains unchanged (can be modified if new fractures need to be generated, see algorithm below). In this way the fracture set's connections to the well will match those seen in the original data. Warning: It is not usually desirable in a modelling project to force this type of perfect match between a stochastic model and measured data, so use this functionality with caution.

In order to run the analysis, first select the fracture set whose well intersection you wish to condition under the Objects window. Then select **Fracture > Condition to Well Log** from the top menu. Select the Fracture Set that you want to use and its associated Fracture Set Definition on the first tab. On the second tab, select the Wells and/or Intervals and the Well Logs that should be matched. Performance of this function degrades as more wells are added, so it's best to keep the number of wells used to three or less.

Clicking Apply saves the definition in the category "Intersecting Fractures". Clicking Run will generate the new conditioned set. The analysis will create a new fracture set whose well intersections match the FMI logs of those wells. Wells without well log data will have intersecting fractures removed (assumes a well log with zero fracture intersections). The selected Fracture Set Definition is used if more fractures need to be created in order to match the number of fractures in the well log(s). If this occurs, you will be warned with the message: "Fracture number in the model is smaller than well log number. New fractures will be generated." If your well log is not in same region (or grid for Geocellular) as is defined for the fracture set definition, you will get the error message: "Point is not in GeolGrid" for each fracture in the well log.

Algorithm used to create this new, conditioned fracture set

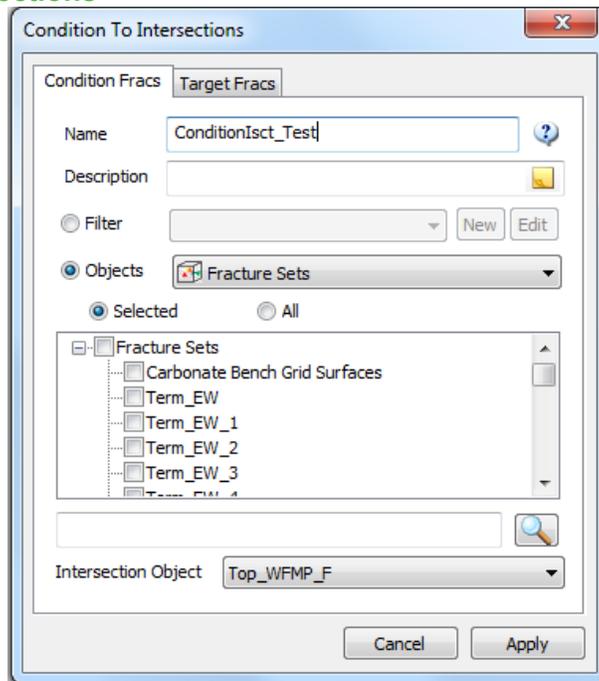
Copy the original fracture set to use as the new set and modify it as follows:

For each well log fracture,

- a) Pick the nearest unselected intersecting fracture from the working set or the nearest non-intersecting fracture (if no intersecting fractures are available) and delete it from the working set.
- b) Create a new fracture using the selected fracture set definition, locating it at the well log point with the associated orientation from the well log (if available) and add it to this working copy.

Finally, delete any intersecting fractures from the working set that were not matched and already deleted.

Condition to Intersections



In addition to being able to condition a fracture set to a well log as described in the previous section, you can also condition a fracture set to a tracemap, that is, the intersection of a fracture set with a surface. The analysis will create a new fracture set whose intersections with the selected Intersection Object match the intersections of that Intersecting Object with the conditioning fracture set. This functionality is available from the Fracture menu. As with the [Condition to Well Log](#) functionality, is not usually desirable in a modelling project to force this type of perfect match between a stochastic model and measured data, so use this feature with caution.

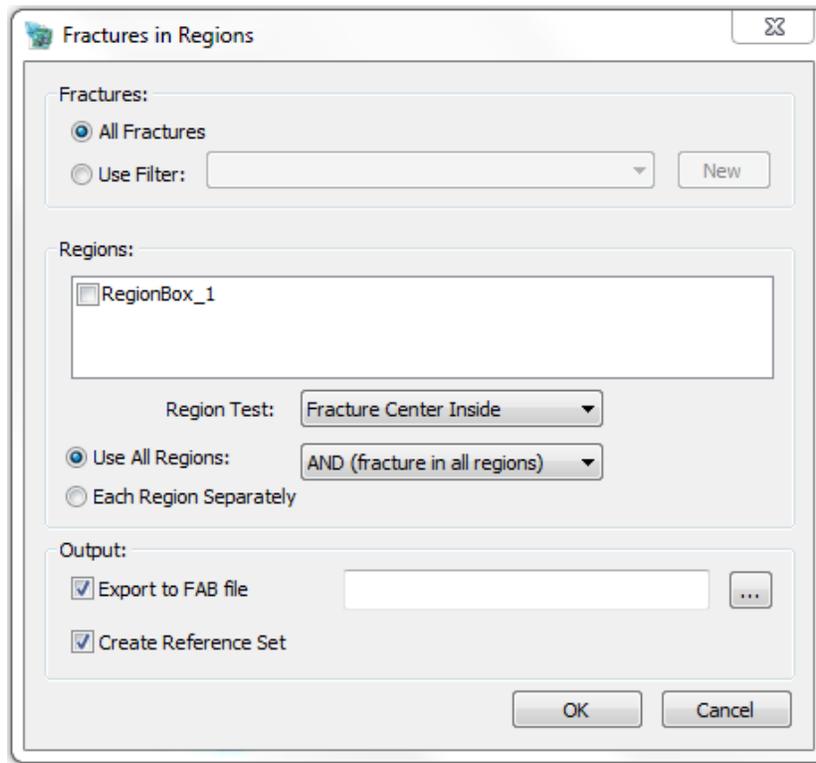
To make the tracemap that you will condition against, select the fracture set(s) and Intersection Object (usually a surface or region) from the first tab. On the second tab, choose the fracture set that you want to alter to match the tracemap. Clicking Apply saves the definition in the category “Intersecting Fractures”. Clicking Run will generate the new conditioned set.

Extract Fractures in Regions

One way to quickly gather the fractures in one or more regions is to use the Extract Fractures in Regions feature. This operation works similarly to the Object Filter, except that it runs in a more efficient manner better suited to large fracture sets.

Select a filter if desired, then select the regions you want to use and whether you want the output to a FAB file on disk or a reference set in the model. The resulting fractures can then be used in various analyses more efficiently than extracting the fractures repeatedly for multiple analyses.

You can extract sets of fractures that are inside or outside of regions. This operation works similarly to the standard fracture filter operation but is optimized for large numbers of fractures and regions. Select Fracture > Extract Fractures in Regions from the menu bar.



Here you can select which fractures for the analysis to select and check all the regions you wish to use. Then specify whether you wish to gather only the fractures within or those outside the regions. If “Use All Regions” is checked, the resulting fracture set will contain all applicable fractures; if “Each Region Separately” is selected, a separate fracture set will appear for each region in the analysis. You may also export the resulting fracture sets to a FAB file. If neither “Export the FAB file” nor “Create Reference Set” is selected, the analysis will do nothing. One must be selected; both may be as well.

Create Fracture Reference Set

Fracture Reference Sets are stored as references to the original objects rather than objects themselves. This uses less memory than full fracture sets and allows you to modify property values of subsets of the original fractures easily. To create a fracture reference set, select one or more fracture sets in the Object window and then select “Create Fracture Reference Set” from the Fracture menu. The new reference set will be added to the Objects window under Fracture Sets with the  icon (notice the small arrow in the bottom right corner). Intersecting and connected fractures are automatically created as reference sets. This functionality is also available from other functions such as [Extract Fractures in Regions](#), [Surface/Region Intersecting Fractures](#) and [Cluster Analysis](#).

Convert to Fracture Set

When you run an analysis on a fracture set that results in the creation of a fracture subset, the subset will appear as a child object but will not be an independent object from the original fracture set. That is, if you delete the original fracture set, even if you have reparented the subset, it will still be deleted as well. Additionally, some analyses can only be performed on fracture sets, not reference sets. In order to work around these restrictions, you must select the object you wish to alter and convert it to a fracture set of its own. This command makes that change. The icon next to the set will change from  to . Please note that henceforth any changes applied to the original set will not transfer to the liberated set.

This operation is similar to exporting the reference set as a FAB file and then re-importing. Once done, the original objects may be modified or deleted without affecting the subsets. Just click on the subset you wish to convert and click *Fracture > Convert to Fracture Set* from the menu bar.

Insert Menu

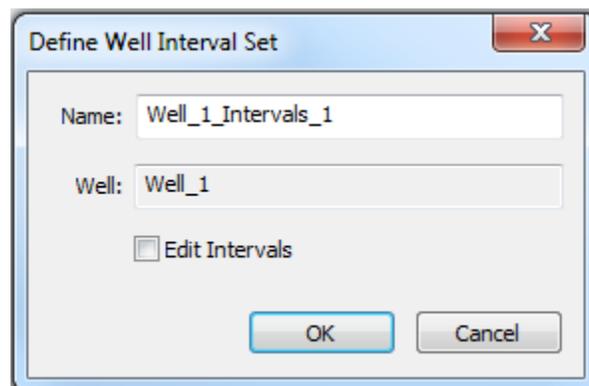
Insert allows you to create new non-fracture objects within your file. Objects will appear in both the Objects panel and the Visualization window.

Insert Well

This creates a new well object in the scene. A dialog will pop up showing some options for creating a Vertical, Inclined, or Deviated Well. See [Wells](#) for more information on using well objects.

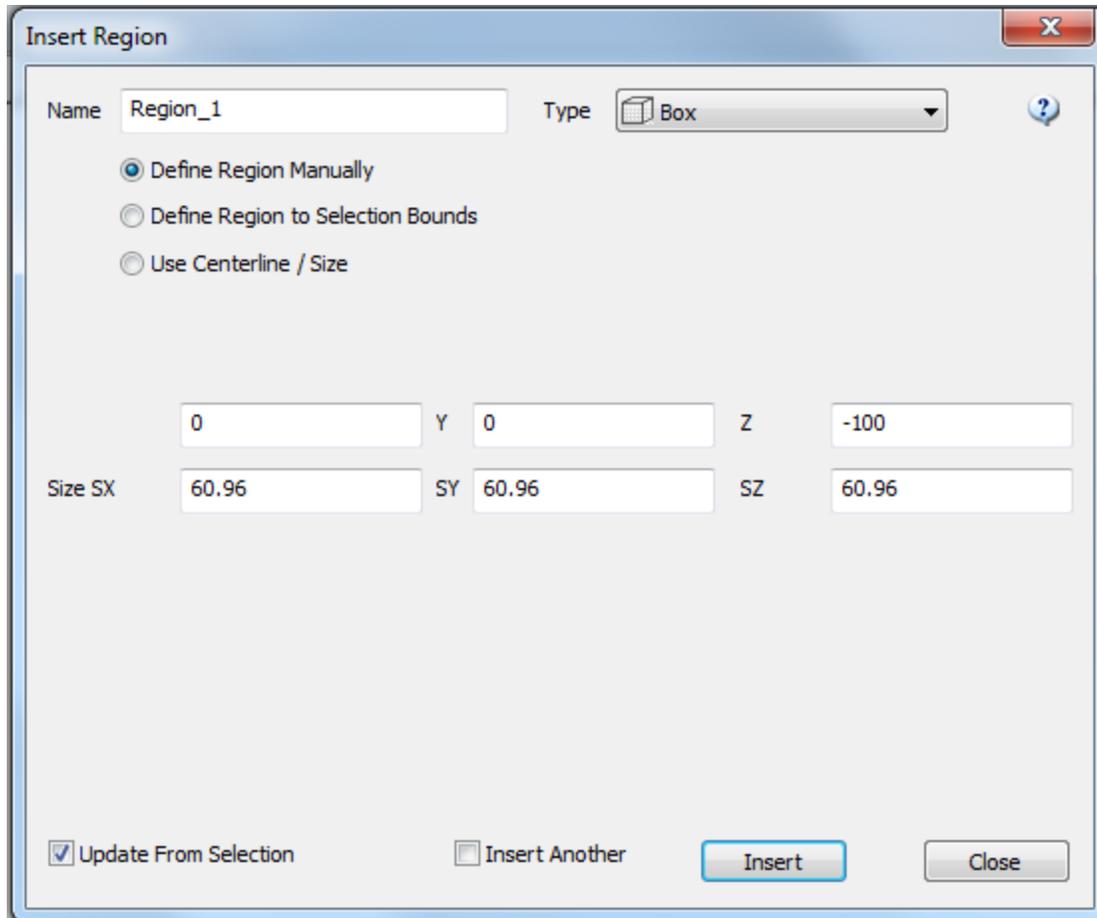
Insert Well Interval Set

If you have a well selected in your Objects window, you can add a well interval set using the Insert Well Interval Set function from the Insert menu. This will bring up the [Define Well Interval Set](#) dialog.



Region Object

Allows finer control over creation of region objects.



Define Region Manually

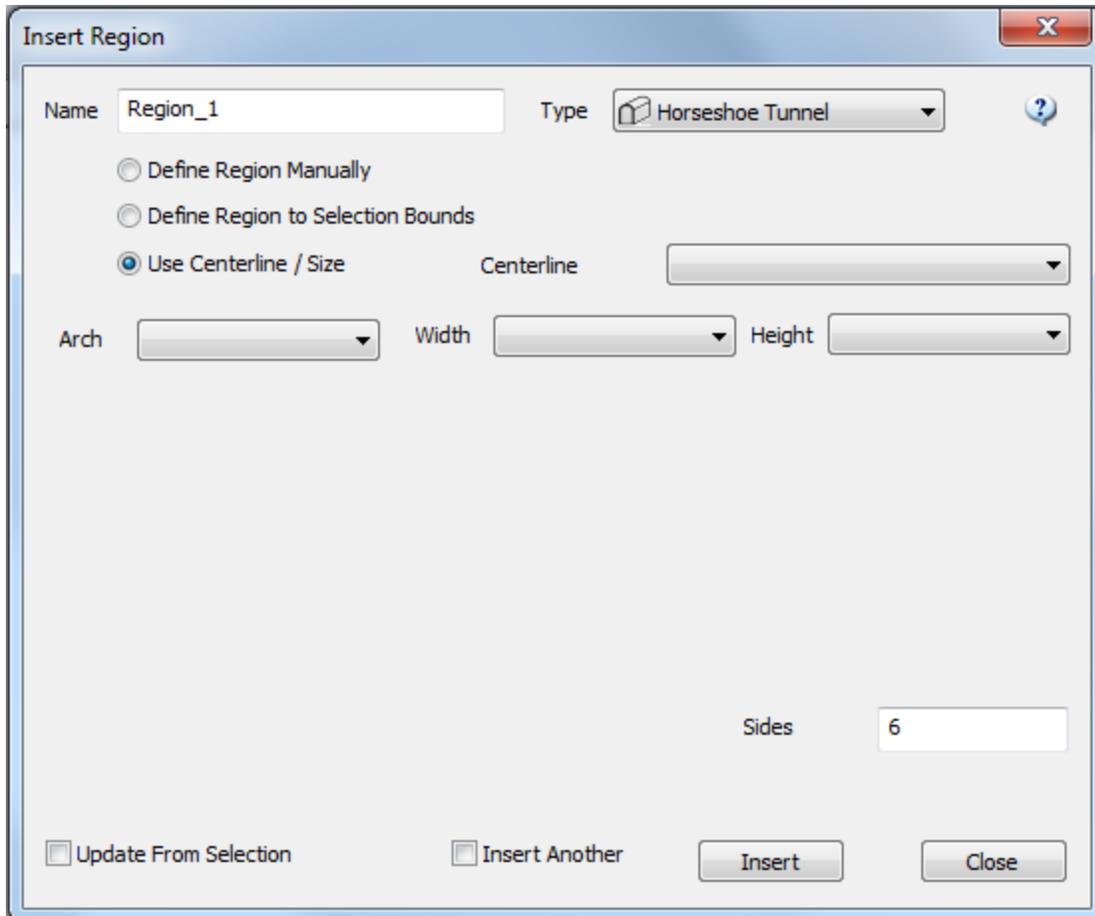
The Define Region Manually option may be used to define Box, Cylinder, Horseshoe Tunnel and Slab regions. This interface allows the user to specify at the outset the basic region parameters such as region center and size (for the Box region) without using the Attributes window. The checkbox **Update From Selection** can be used to populate the parameter values using the object selected in the Object window. If the checkbox **Insert Another** is checked the dialog will remain open after **Insert** is clicked.

Define Region to Selection Bounds

Works the same as selecting one or more objects and choosing **Insert Box Region, Cylinder Region, Slab Region** or **Horseshoe Tunnel**. Creates an object that encloses all the selected objects. Dimensions may be modified using the **Attributes** window.

Use Centerline / Size

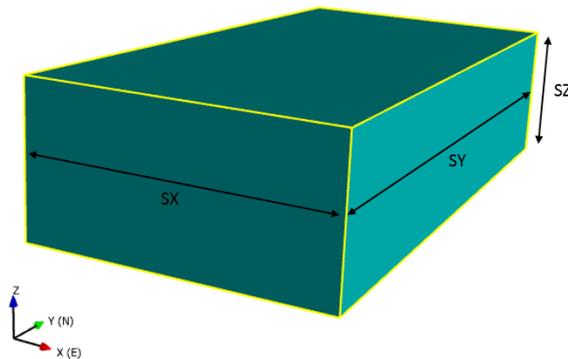
This can be a very useful function for creating complex tunnel regions when you have a trace with the tunnel centerline.



Box Region

This creates a new region shaped like a rectangular prism. The default region encompasses all objects in the scene. In order to change the size, aspect, and position of the box, you must go into the Attributes panel. Region attributes are:

- Center: X, Y, Z
- Size: SX, SY, SZ

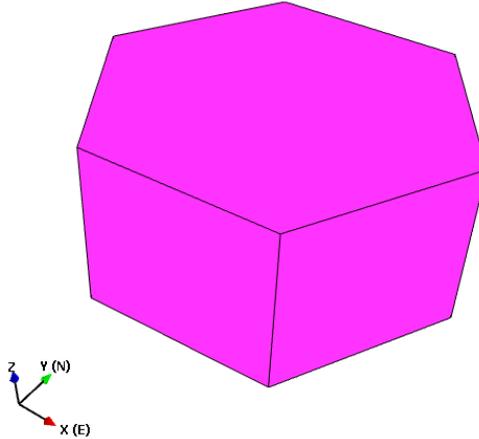


Cylinder Region

This generates a cylindrical region. The default is six-sided; in order to change this, as well as the size and position of the region, you must edit the region's properties in the Attributes panel.

Region Attributes are:

- Origin: X, Y, Z
- Axis: Trend, Plunge, Length, Radius, Sides



Prism Region

This creates a prism region which is similar to a cylinder except that you can define the shape of the cross section. To create a prism region, select *Insert > Prism Region* from the menu bar.

Define Prism Region

Name:

Prism Origin

X: m

Y: m

Z: m

Prism Axis

Trend:

Plunge:

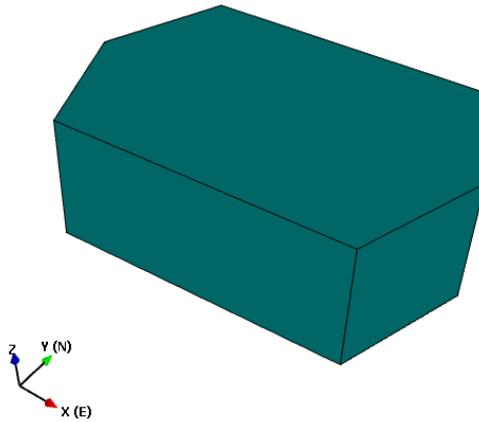
Length: m

Cross Section Points (x, y)

- 100.000000, 100.000000
- 100.000000, -100.000000
- 100.000000, -100.000000
- 100.000000, 100.000000

Specify the coordinates of the region's center point in the X, Y, and Z boxes. Then specify the trend and plunge of the region's long axis and the length of that axis in the appropriate boxes.

Finally, specify the vertices of your prism, as if you were looking along the long axis. You may specify as many points as you wish. Hit OK. The result will be a prismatic region with a cross-section whose vertices match your specified points and whose long axis is of the given orientation and length, centered on your specified (X, Y, Z).



Slab Region

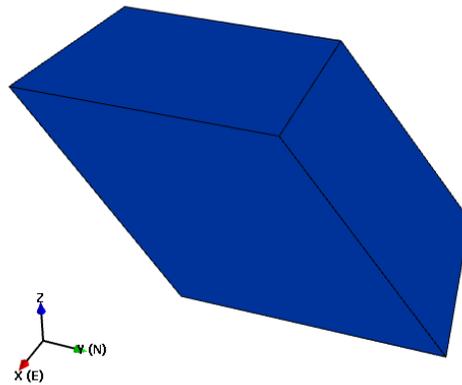
A slab is a prismatic region defined by its dimensions, a point at one end of the slab axis, and the orientations of the prism sides, expressed as trend and plunge.

By default:

Scan = x

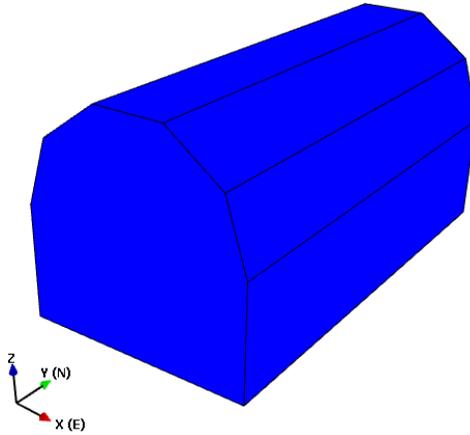
Transverse = y

Height = z



Horseshoe Tunnel

This generates a horseshoe-shaped prism region, like a tunnel that has been filled in. The Attributes panel allows you to change its size and orientation, the coordinates of one endpoint of the long axis of the region's base, and the number of sides comprising the arched portion of the tunnel.



Tunnel objects can also be defined using one or more polylines (loaded as a tracemap) via the Create Tunnel From Centerline / Height feature.

Traceplane

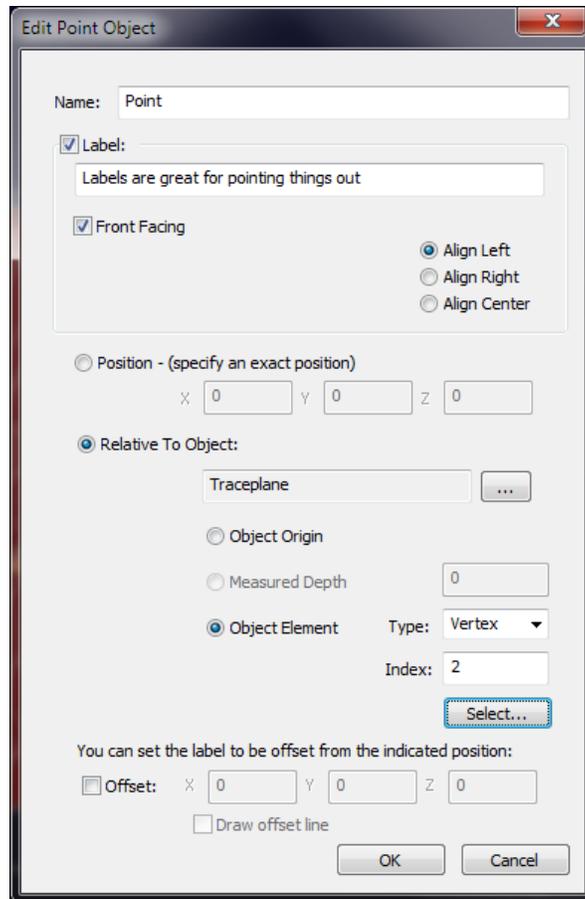
This creates a simple plane. Its orientation, size, and location can be altered in the Attributes panel.

Point Marker

Point marker objects are useful for locating important features in your model. Each marker can include a label which will be displayed in the visualization window.

You can define a point marker by specifying an x, y, z location in your model or by making the point relative to some object, measured depth (along a well) or off of an object element (vertex, cell etc.).

This is an object type intended for labeling objects or points of interest in your model. The object has a name as well as a label that can be drawn in the scene. You can specify an absolute location of the point marker or select it to be relative to another object – either as a measured depth along a well or at the location of the object's sub element (cell/vertex/segment etc).



You can also specify to draw the point marker's label offset from the points location in order to get it out of the way of other objects in the scene.

Slope

Simplifies creation of open pit bench slopes.

The origin is the XYZ coordinates of the top of the crest (left side when facing the slope). The slope itself is defined by the direction and length of the crest (top), the length of the toe (bottom) as well as the width of both the crest and base. Benches are defined by the angle of their face from horizontal (a value of 90 would be vertical), their width, height and the total number of benches. Ramps are treated as a special case of bench with a different width. Specify the positions of the ramps by a list of their locations (1, 3, 5) would replace the 1st, 3rd and 5th benches with ramps.

Basic Grid

This creates a simple grid without any properties except cell volume. You may specify a box region from which the grid will be mapped; regions of other shapes do not work here. If you do not specify a region, you must give an origin, dimensions, and skew by which to define the grid. A grid need not be rectangular.

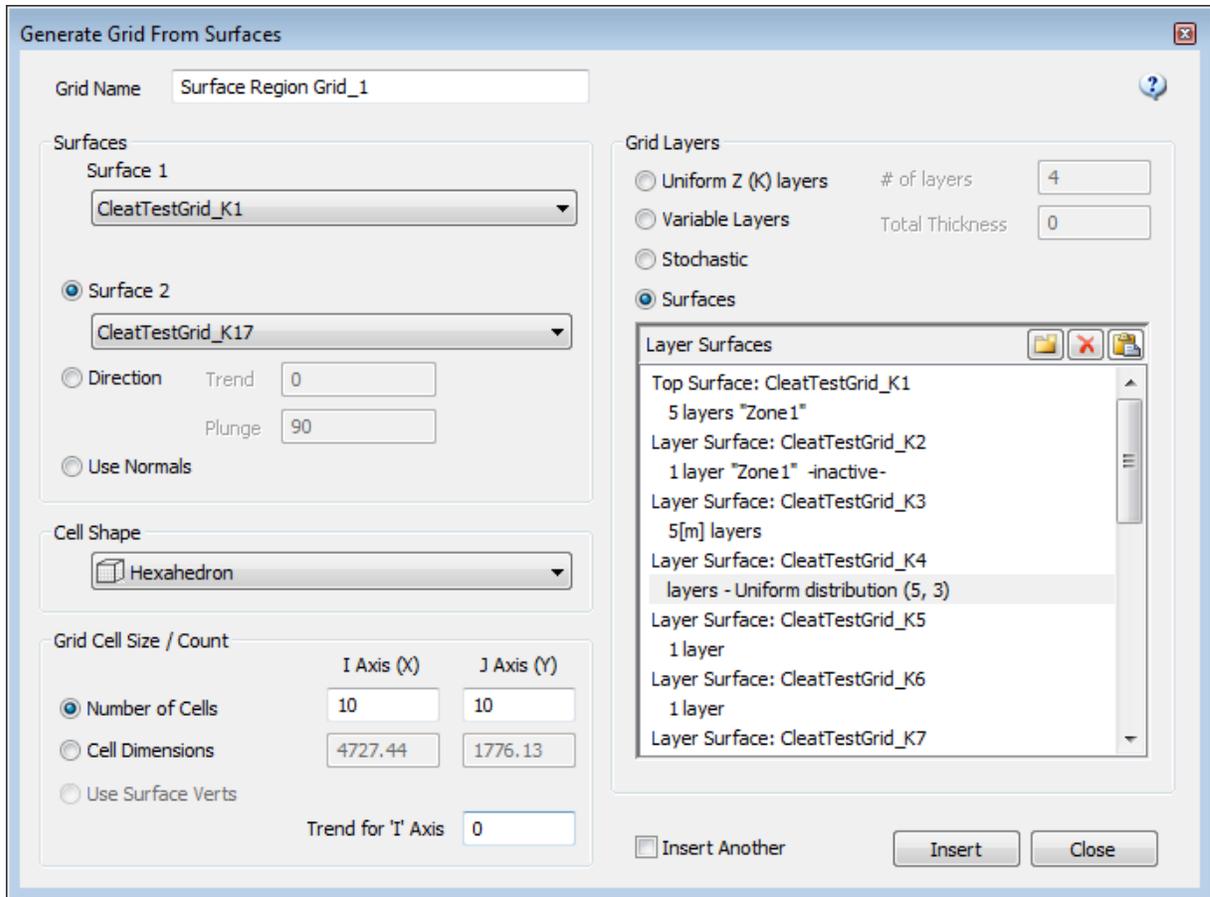
If you wish to use an existing box region to define the grid boundaries, select the desired region from the dropdown menu next to “Copy from:”, otherwise, specify the X, Y, and Z coordinates where you wish the grid corner where i, j, and k = 0 to be in the appropriate boxes next to “Origin”. The specified Scan, Transverse, and Height lines will extend from there, thus defining the grid boundaries.

When you have finished defining your grid, click **Insert**. A new object will appear in the Objects window under “Grids”.

Grid From Surfaces

When defining a grid from one or more surface objects you can define the grid based on the surface normal, a specified direction or a second surface. If you are defining a grid between two surfaces, you must have at least two distinct surfaces defined in your file before you may proceed. The surfaces can touch or intersect, but if their positions are identical FracMan will reject them. *Note:* Because the grid is mapped between the surface points along the Z-direction, surfaces oriented perpendicular to the Z-axis will result in empty grids.

Specify the grid name in the box next to “Grid Name”. Please note that if you choose the same name as some pre-existing grid in your file, the existing grid will be overwritten when this new grid is created.



The dropdown menus under “Surface 1” and “Surface 2” will each give a comprehensive list of all available surfaces in your file. Specify “Surface 1” and then choose to define the surface by either selecting a second surface, prescribing a specified direction, or checking the “Use Normals” to base the grid on the normals of “Surface 1”.

Specify the shape of the grid cells from the “Cell Shape” drop-down menu as Hexahedron or Triangular Prism.

The Grid Cell size can be determined from cell number, cell dimensions, or by using the surface normals. The “Trend for ‘I’ Axis” box allows you to specify how the final grid is oriented relative to the x-axis. It is important to note that the Trend for the “I” Axis will affect the numbers of cells/cell size.

Use the “Grid Layers” section to specify the thickness of the grid layers. You can specify uniform z layers or variable by specifying their thickness based on a fraction of the total width. The “# of layers” box specifies how many cells high the grid should be – that is, how many cells should be mapped between the surfaces. If you are defining the grid using one surface and either a direction or the surface normals, you will also need to specify the “Total Thickness” of the grid.

When generating a grid between two surfaces you can specify non-uniform layer thicknesses. Select “Uneven layer thicknesses” and specify the fractional layer width of each layer (0 – 1). The total must add up to 1.0.

Select “Surfaces” to define intermediate grid layers based on additional surfaces from the model. Press the insert button  to select a new layer surface. Double click on the “layer” line to define additional layers between the surfaces. Use the paste button  to enter multiple layers at once with text in the form:

```
Surface1Name NN  
Surface2Name NN
```

NN is the number of layers to be inserted between the surfaces.

When you have completed the wizard, hit **Insert**. The new grid will appear in the Objects window under “Grids”.

Fault Model Grid

FracMan describes faults as grids that consist of one or more triangular or quadrilateral elements of specified thickness. The intensity of fracturing in each element is then controlled by a quantity termed the fracture potential. This fracture potential is a function of slip magnitude, distance from the fault tip, distance from the fault midplane, and up to two user-specified scalar intensity parameters.

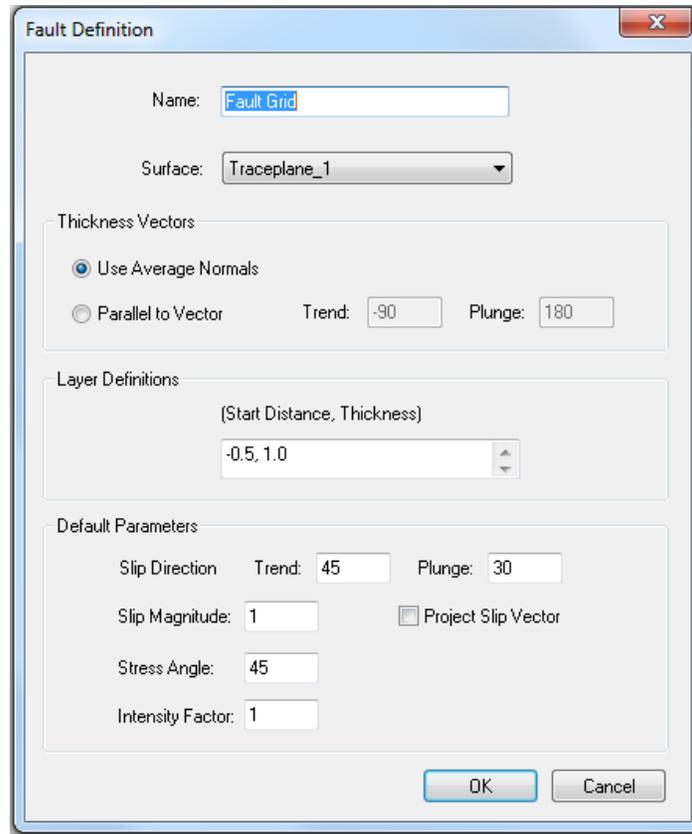
The fracture orientation is controlled by local fault plane orientation, the slip direction, and the inclination of principal stresses to the slip plane.

The Thickness Vectors section allows you to choose the direction of the layer thickness.

- **Use Average Normals** will create the Fault Grid parallel to the surface’s normal vector field (to visualize the surface’s normal check Draw Face Normal in the surface’s Attributes window).
- **Parallel to Vector** function will create the Grid parallel to the vector designated.

The **Layer Definitions** section allows you to choose the amount, starting distance from the surface, and thickness of the grid layers. In order to enter multiple layers, press Enter after each Start Distance and Thickness entry.

The **Default Parameters** section is used to define the slip direction and magnitude of the grid due to the fault.



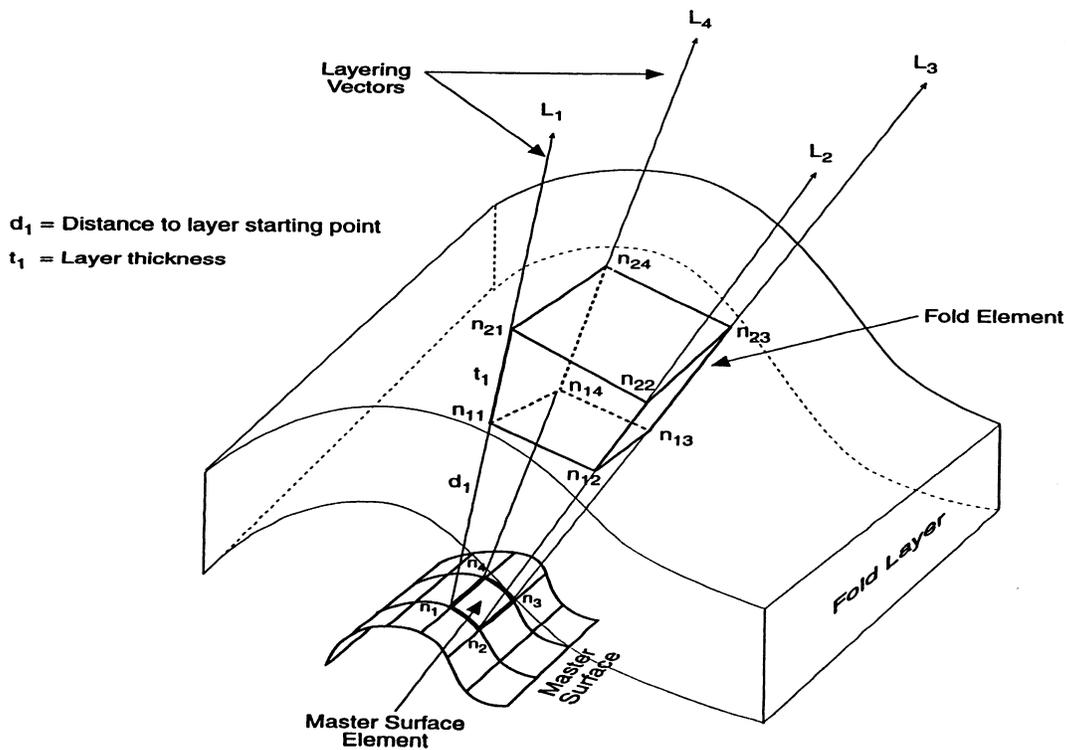
Note: Slip Magnitude under the *Default Parameters* section is used in the *Geologic Fracture Generation Definition* as a parameter to define fracture intensity.

Fold Model Grid

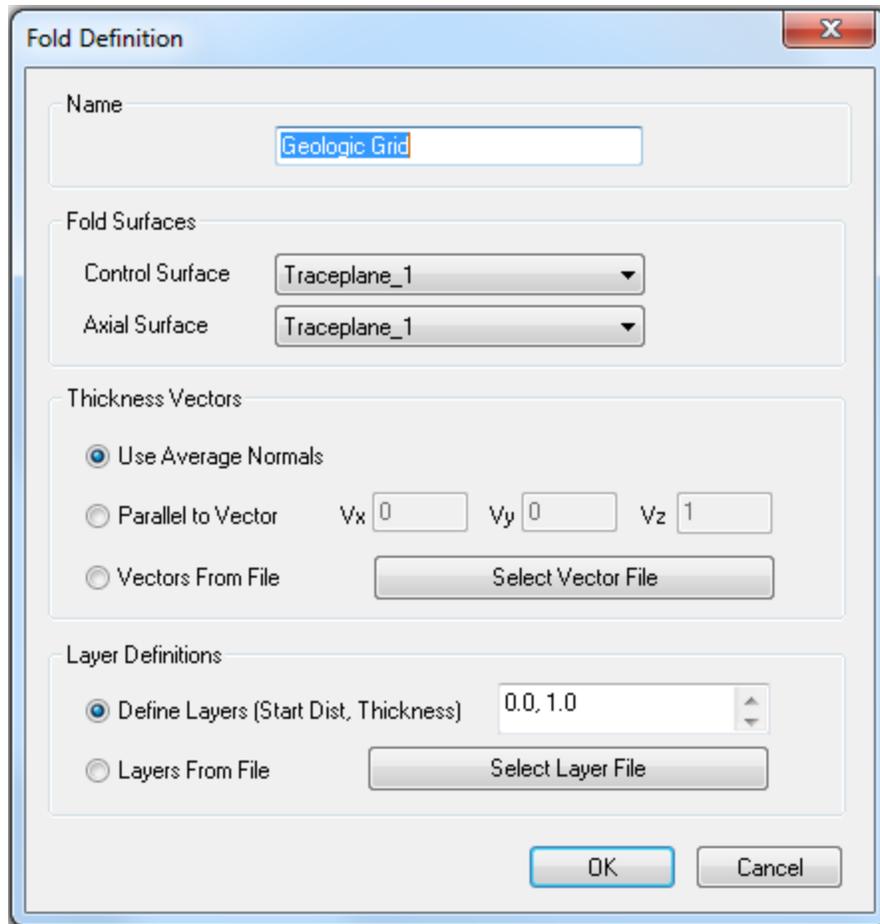
Using the Fold model, fracture geometry can be related to the local orientation of the bedding or to the axial surface of the fold. If the fractures in a stratigraphic formation arose from bending of the rock, then they should show certain orientations and intensity patterns that relate to the local bedding orientation, thickness, and distance relative to a “neutral” surface.

FracMan uses a control surface and a plane representing the fold axial plane to define the geometry of folds. A fold is described by first defining a control surface (usually a bedding surface) and then specifying how adjacent rock layers conform to or deviate from this surface. In general, the control surface will be non-planar. Therefore, FracMan uses imported surfaces to describe its geometry.

Each fold is comprised of one or more layers. Each layer, in turn, consists of one or more triangular or quadrilateral elements of specified thickness. The location of these elements is defined by the projection of the elements on the control surface along layering vectors.



The axial surface of a fold is a surface that divides the fold as symmetrically as possible. The intersection between the axial surface and the bedding surface defines the fold axis. In non-cylindrical folds or in refolded rock, this surface may not be perfectly planar. Therefore, the axial plane is also specified as a surface in FracMan.



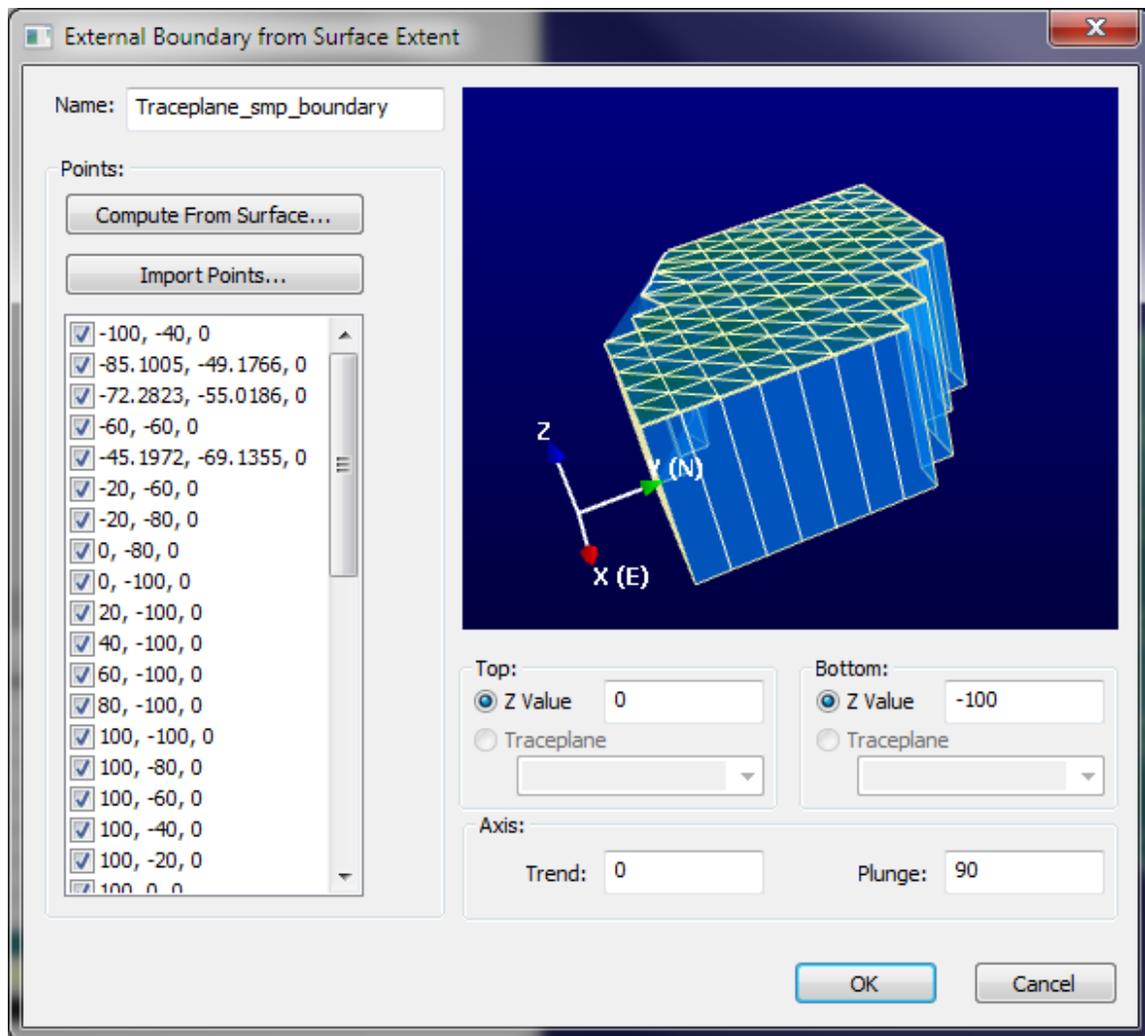
The **Thickness Vectors** section allows you to choose three different ways of defining the direction of the layer definitions. The **Layer Definitions** allows you to define the thickness of the grid cells per layer. In order to enter multiple layers, press **Enter** in the dialog box after each entry. You can select a layer from a previously created data file.

Boundary Region from Surface Extent

A surface boundary region is composed of a surface object with an irregular edge. This type of region object is generally used only for dynamic flow simulations.

First you need to define the outline. You can import this from a file or compute it based on a surface in your model.

Once the outline is defined, specify the location and direction of the top and bottom of the region.



Map Object

If you have a surface map available for your model, you can display it within FracMan by inserting a Map Object. This object will be a specially formatted rectangular surface that has your image applied as a texture map across it.

Select the image file you want to use. The file will be automatically resized when loaded to be able to fit in video memory.

Provide the coordinates of the top left corner of the map.

Specify either the coordinates of the top right and bottom left of the image, or specify the width, height and direction of the map (in world coordinates).

Insert Map

Object Name:

Map Image File:
Select the image file containing the map:
 ...

Resolution:

Top Left Corner:
X: Y: Z:
(this is generally the northwest corner of the map - X being the E-W and Y being N-S)

Size:

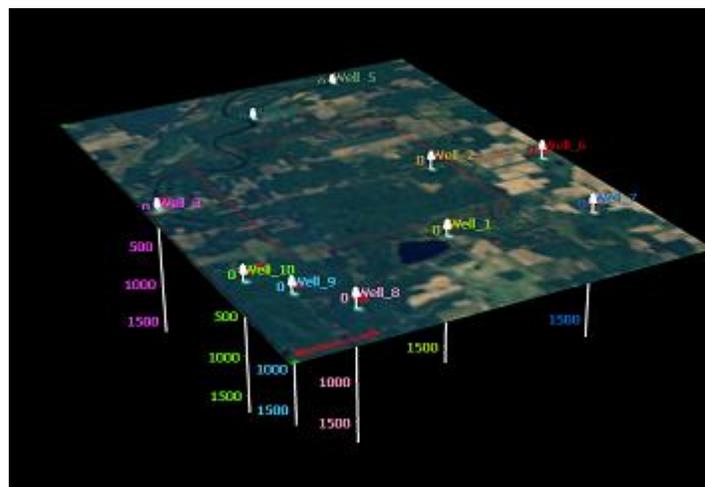
Corner Points:

Top Right X: Y: Z:
Bot Left X: Y: Z:

Size:

Map Width: Height:
Trend:

The resulting map can be seen in the visualization window:



Convex Region Around Objects

If you select an object in the Objects window and choose the function **Convex Region Around Objects** in the **Insert** menu, a new Polyhedron-type Region object will be created that surrounds the selected object.

Insert Object Group

The **Insert Object Groups** function in the **Insert** menu allows the user to create custom folders for more organization of objects. Select the parent level for the Object Group in the Objects window and the new Object Group folder will be created with the user-specified name as a child of that object. Other objects can then be added to this group.

Region from closed surface

If you have a closed surface object selected in your project, you can select **Region from closed surface** from the **Insert** menu to create a new region object which will be automatically added to your model based on the selected surface object.

Data Analysis Menu

The functions available under the Data Analysis menu manipulate data that has been imported into the project in order to define the parameters that can be used to generate a discrete fracture model. Detailed description of each function can be found in the Manual section on "[Data Analysis](#)".

- [Spatial](#)
 - Set Spatial Parameters
 - Run Spatial Analysis
- Orientation
 - [Set Identification](#)
 - Terzaghi/Mauldon Correction
- Intensity
 - Cumulative Fracture Intensity Plot
 - OxFilet Flow Capacity Analysis
 - Compute P10
 - Intensity Conversion (C31) Calculation
 - Compute P21 / P31
- Size
 - Trace Length Fracture Size
- Spatial Patterns
 - Power Law Plot
- Well Log Report
- Object Statistics

Static Analysis Menu

The Static Analysis menu includes unique tools to answer practical flow, transport and geomechanical questions by determining fracture network connectivity, pathways, stress and rock blocks. These tools provide geometric answers to geometric questions such as where the

oil is going or how big are the rock blocks. A full description of each operation is included in the Manual section "[Fracture Model Analysis](#)".

- Fracture
 - Cluster
 - Pathway
 - Critical Stress
 - Fracture / Fracture Traces
 - Compute Height / Width
- Grid
 - Export Cell Info
 - Fracture Upscaling
 - Layered Upscaling
 - Intersecting Grid Layers
 - Grid Layer Property Report
 - Create Stress Properties
 - Compute Grid Cell Volume
 - Compute Grid Cell Height
 - Compute Grid Cell IJ Dimension
 - Compute Grid Cell XY Area
- Well
 - Intersecting / Connected Fractures
 - Conditioning Statistics
 - Intersecting Grid Cells
 - Correlate P10 To Grid Property
- Surface
 - Intersecting Fractures
- Volumetric
 - Tributary Drainage Volume
 - Estimated Ultimate Recovery
- Geomechanics
 - Rock Wedge Analysis
 - Block Size
 - Subsidence
 - Hydraulic Fracture
 - [FracStress Geomechanics](#)
 - Grouting
 - See My Frac
- Statistics
- Stereonet
- Property Stereonet
- Property Correlation Graph

- Basic Geostatistics Mapping

Dynamic Analysis Menu

The Dynamic Analysis menu houses the functions available for flow and transport simulation. See the Manual section “[Dynamic Fracture Analysis](#)” for a detailed description of each operation.

- Well Test
- Mafic
 - Define Mafic Simulation
 - Define Boundary Conditions
 - Define Mafic EPM/DFN Model
- Grid Upscaling
- Dynamic Upscaling
- HydroMech

Tools Menu

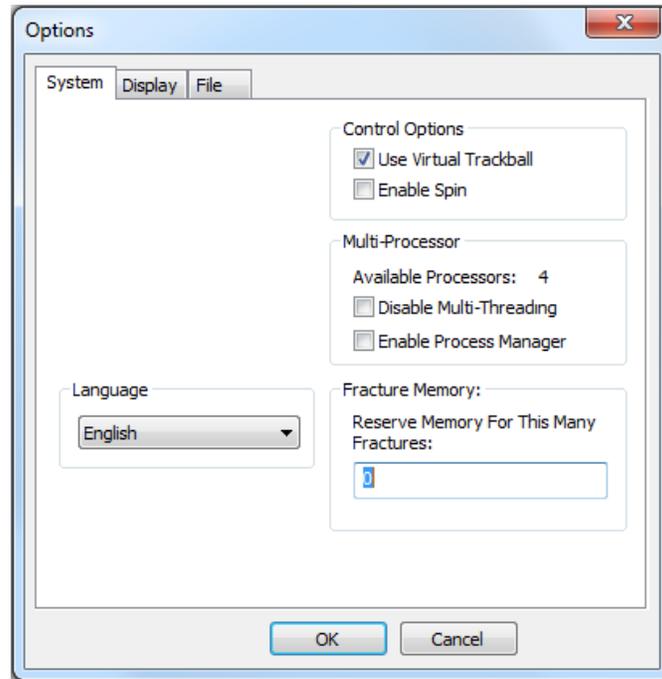
Select FracWorld Region

This allows you to choose which region in your file the program uses to calculate fracture intensities. When you click on it, FracMan brings up a list of all available regions. The default FracWorld region is the entire scene – that is, the total area covered by your file. To change this, just click on the name of the region you prefer. The name will be highlighted. Click OK. Your base region is now the one you have chosen.

Options

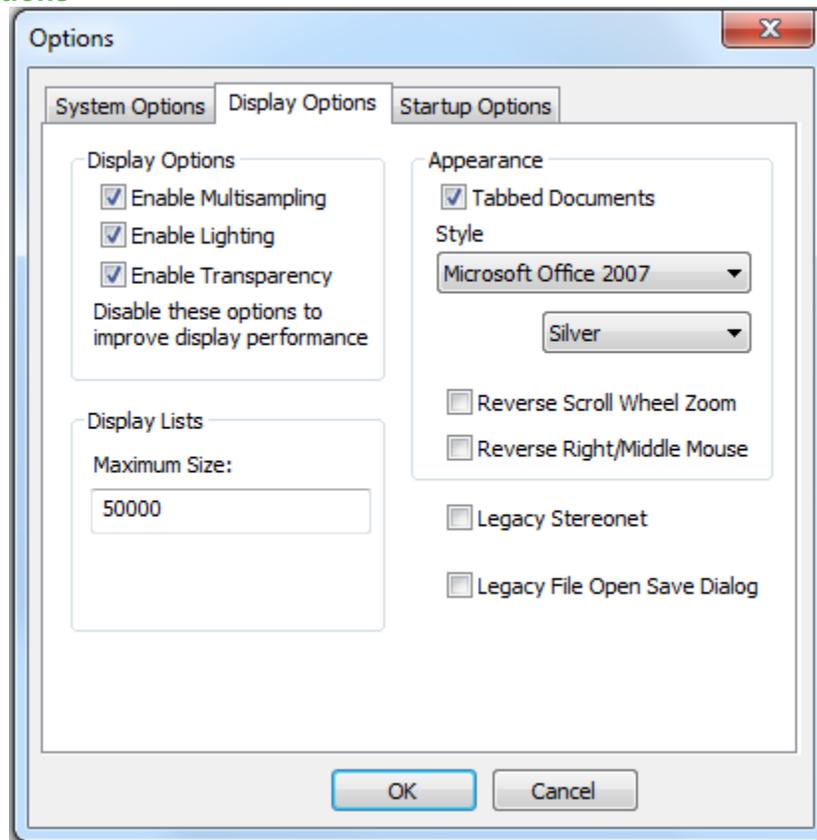
This brings up a set of three tabs containing options for how FracMan runs which may be turned on and off. If there is a check mark in the box next to the option, the option is on. Clicking in the box will remove the check and turn the option off. Likewise, clicking in an empty box will activate the adjacent option.

System Options



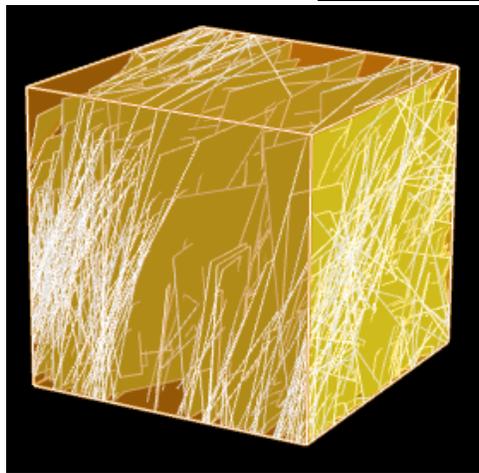
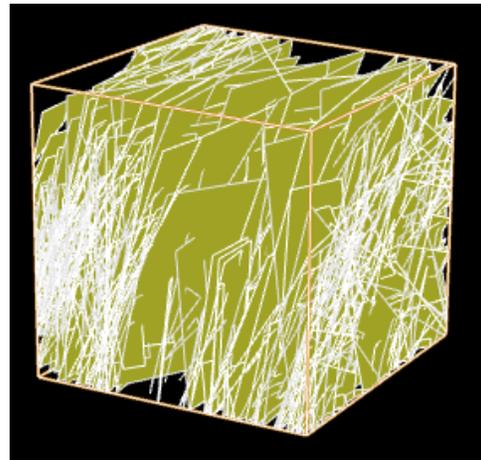
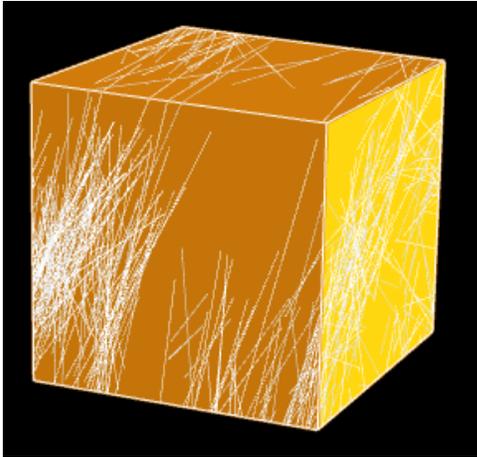
- *Language*: Select which language you wish the FracMan menus and dialogs to appear in. Currently, FracMan supports three languages: English, Chinese and Russian. This option can also be changed directly from the Tools menu option [Language](#).
- *Use Virtual Trackball*: There are two standard ways of handling rotating the scene on is the more common mechanism.
- *Enable Spin*: When this is selected, the user can make the image in the Visualization window rotate continually without using the mouse to maintain rotation. To do this, simply hold down the left mouse button and rotate the image normally. Then, with the mouse still moving, release the button. The faster the mouse is moving on release, the faster the image will spin. To stop spin, click anywhere in the Visualization window, and the image will freeze.
- *Disable Multi-Threading*: Many FracMan operations will use multiple processors in order to improve performance, however this can cause slight differences in results between machines and even between runs. Checking this will force FracMan to use a single thread to produce exactly reproducible results.
- *Enable Process Manager*: Function currently available for internal Golder use only.
- *Reserve Memory For This Many Fractures*: When you generate very large fracture sets, FracMan searches for and reserves the memory for the fractures as it generates them. This causes memory fragmentation, which means that the program uses more memory than it should. This number pre-allocates storage for the specified number of fractures when generation starts.

Display Options



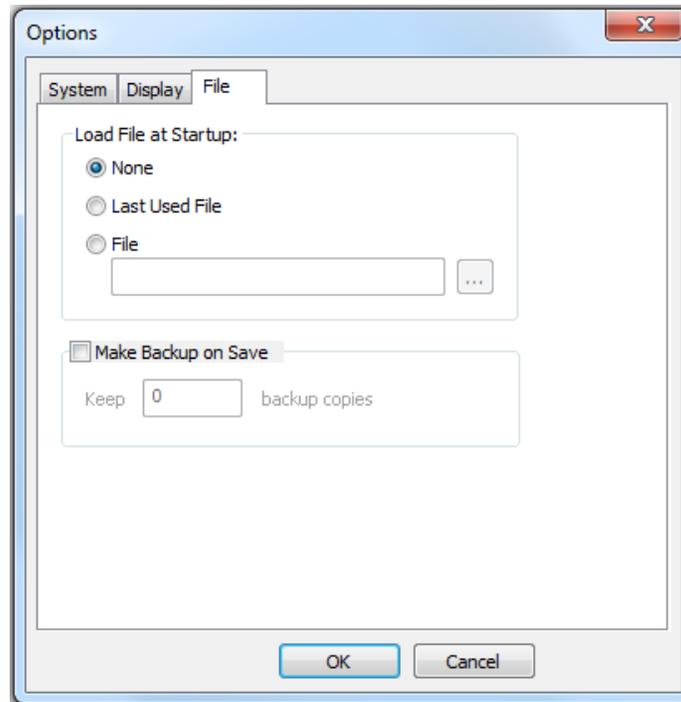
- *Enable Multisampling:* Some video cards support a feature called Full Screen Antialiasing, which can improve the appearance of the visualization display. If supported, this feature can be enabled by checking the **Enable Multisampling** display option in the options dialog.
- *Enable Lighting:* This changes how certain objects are shaded so that it appears a light is shining on them from a certain direction, adding depth and realism to the image. Older video cards can see improved performance by turning lighting off.
- *Enable Transparency:* Some objects like surfaces and grids have a “Transparent” option which will make them appear semi-transparent in the visualization window. This option only works when no property is being visualized and when the **Enable Transparency** display option is enabled in the options dialog. When enabled, this will allow you to see through the transparent object to the objects behind or inside.
- *Maximum Size:* Rendering performance can be increased by setting a number here that corresponds to the amount of video memory your card has. Experiment with various settings to between 1000 and 500000 for the best performance.
- *Appearance:* You can select between a number of different appearance styles as well as choosing whether to use tabbed documents or traditional document subwindows.
- *Reverse Scroll Wheel Zoom:* Causes zoom to work in opposite direction during scrolling.

- *Reverse Right/Middle Mouse:* Pan when the left and middle mouse buttons are pressed at the same time instead of zoom.
- *Legacy Stereonet:* Use FracMan v 7.4 stereonet options.
- *Legacy File Open Save dialog:* Use older version of the file open/save dialog. Setting this option can prevent crashes occurring with some specific combinations of the Windows operating system and MS Office software.



Clockwise from top left: Filled box region, Wireframe box region, Transparent box region. In order to make an object Transparent, you must check that option in the Display Options panel. Some objects, such as fracture sets, and some graphics drivers do not support transparency.

Startup Options

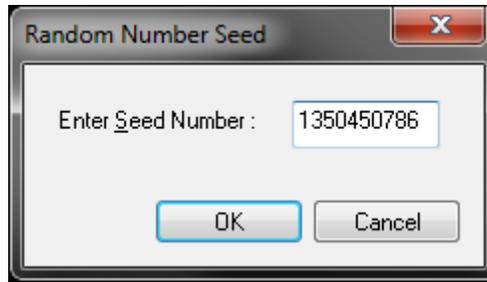


Load File at Startup: This option allows you to tell FracMan to automatically open the specified FracMan file when you open the program. This is especially helpful when you are working consistently on the same file. Click the circle next to the option you prefer. If you select “File”, you must specify the file by clicking on the “...” button to the right of the option. This will allow you to browse for and select the file. Click “OK” or “Apply” to commit the changes. **You can override this feature at startup by holding the Shift key while the application starts.**

Make Backup on Save: When you click on Save, the previously saved project file (if any) will get copied to a backup file having a “frb1” extension in the same directory. If you have specified more than 1 backup, the backup filename extension names will all be shifted so that the numbers on the extension indicate how far back the changes go. The “frd” file will be the current file, the “frb1” file will be the previously saved version, the “frb2” file will be the version saved two iterations back, etc.

Set Random Seed

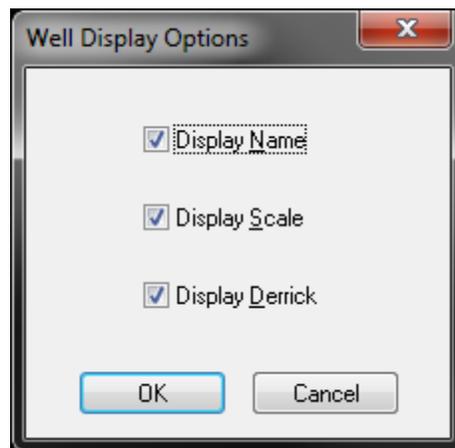
This allows you to specify what number to base your random number generation off of. Since most of FracMan’s capability lies in stochastic processes, this number is frequently used. It regenerates randomly every time you open the program.



If you wish to use a different seed than that supplied, simply enter your preferred number in the box next to “Enter Seed Number:” and hit **OK**. Your seed has now been changed.

Set Well Display

This controls how wells are shown in the Visualization window. To display a particular feature, make sure the box next to the feature name is checked. Click on a box to check or uncheck it.

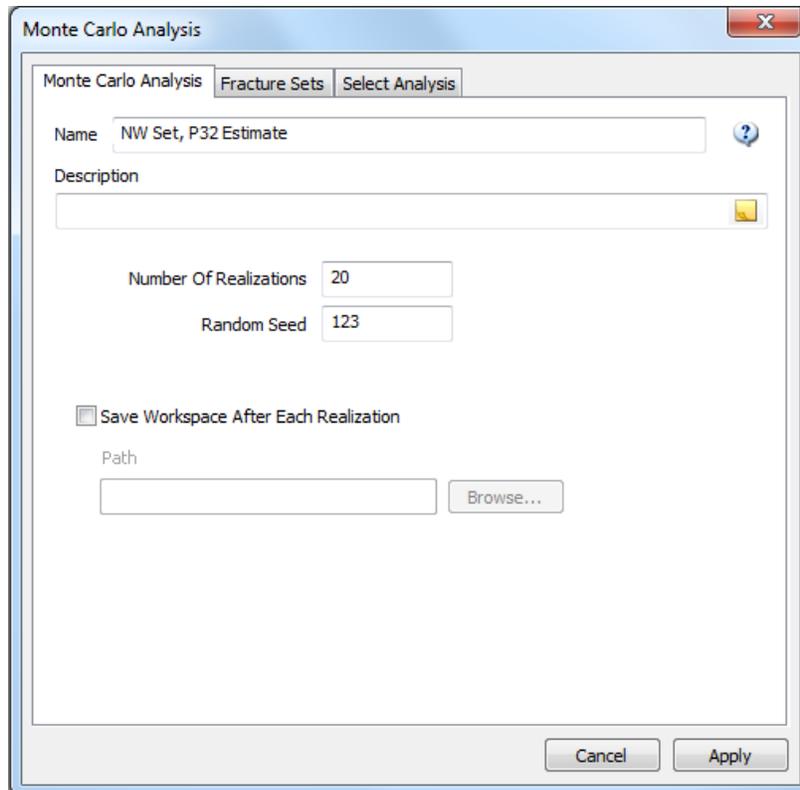


- *Display Name*: Shows the well’s full name at the well head.
- *Display Scale*: Shows scale markers along the length of the well. Scale interval can be changed in the Attributes panel.
- *Display Derrick*: Shows a simple schematic well rig at the head of the well.

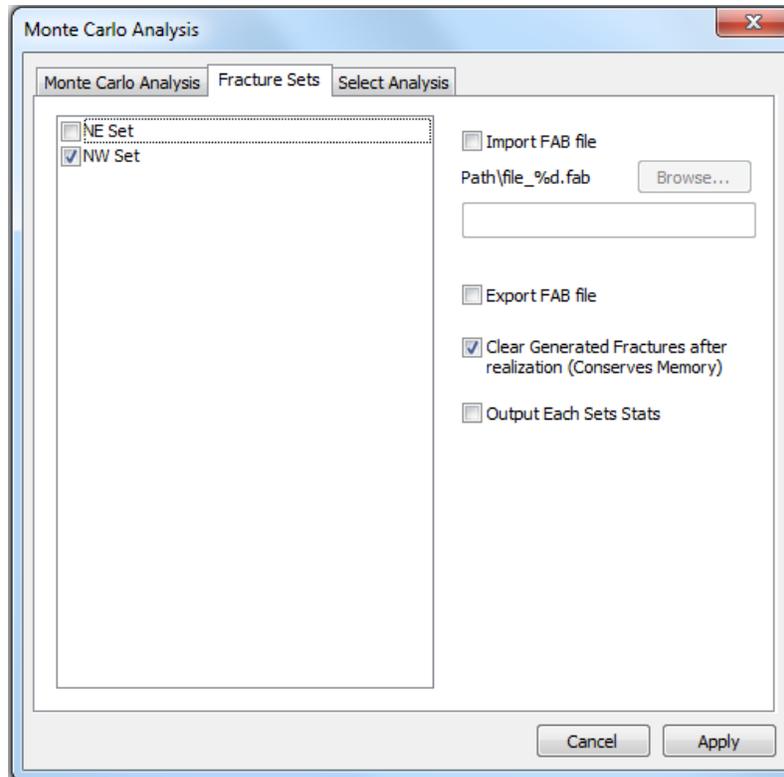
Monte Carlo / Optimization

Define Monte Carlo Analysis

This allows you to generate and analyze multiple fracture sets in order to create a large sample size.



The **Monte Carlo Analysis** tab allows you to specify the basics of your analysis: how it will be saved and where, how many times it will run, and so on. The analysis will be saved under the definitions window by the specified name. An optional description may be added. If **Save Workspace After Each Realization** is checked, a FracMan file corresponding to each iteration of the Monte Carlo analysis will be saved to the selected directory (in the box under **Path**). The saved files will have the same name as the definition, with a number corresponding to which iteration resulted in each. The test will be run for as many iterations as you enter next to **Number of Realizations**. The option to specify a **Random Seed** is so that an existing Monte Carlo analysis can be reproduced exactly, as the seed controls the results of stochastic fracture generation.

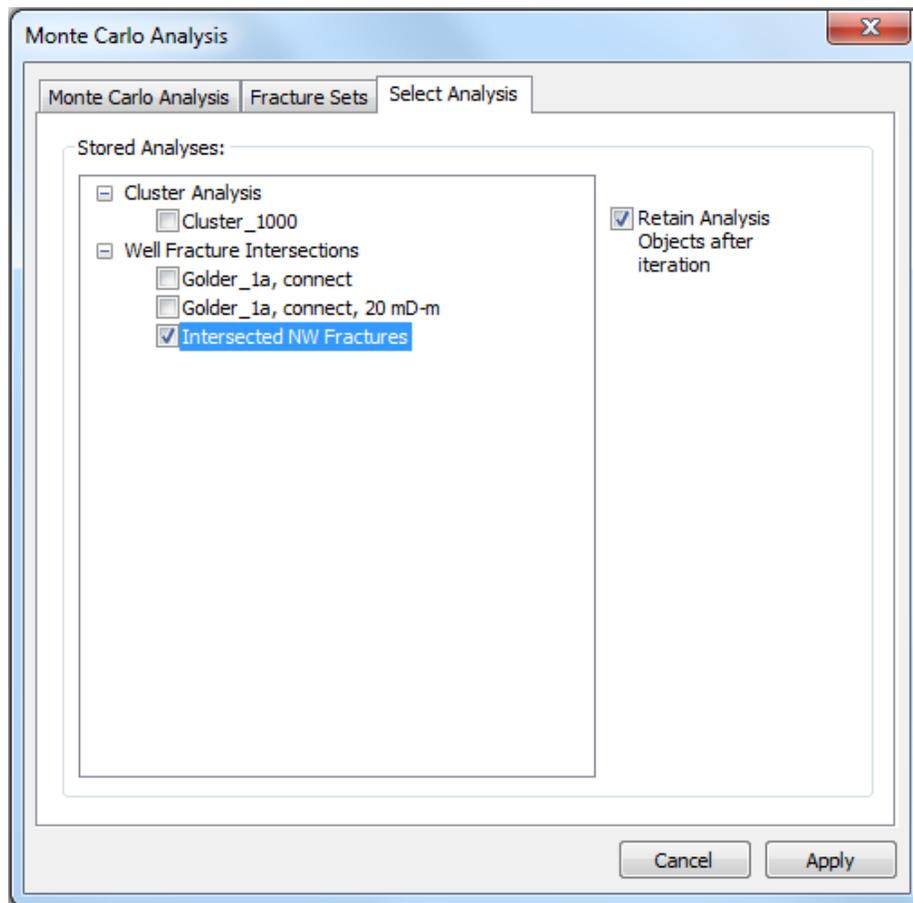


The **Fracture Sets** tab is where you define which fracture sets the analysis will run on and whether they will be saved or not. From the list down the left side of the dialog box, select which fracture set definitions you wish each iteration of the test to create. If you wish instead to use preexisting fracture sets, check **Import FAB file** and specify the directory where the desired .fab files for the analysis may be found. In order for this to work, your fracture sets must have similar names, so that a “wildcard” character (* or ?) can be used to differentiate them.

If you check **Export FAB file**, each iteration’s fracture set will be exported to the same directory specified in the **Path** textbox on the preceding tab. Even if you do not save the FracMan workspace file resulting from each iteration, you must specify a directory in that box if you wish to save the FAB files from each iteration. The files will be saved using the same filename structure as the FracMan files, with “Fractures” appended to the definition name.

If **Clear Generated Fractures after each realization** is checked, the fracture set created for each step of the Monte Carlo simulation will be deleted once it has been analyzed, before the next fracture set is generated unless the fracture set is needed in order to save the results (see **Retain Objects** option on following tab). Otherwise (if the option is not checked) you will end up with a fracture set corresponding to each iteration of the Monte Carlo analysis in your file.

Output Each Set Stats will cause FracMan to output a set of statistics for each generated set at every iteration, otherwise only the aggregate statistics will be computed.



The **Select Analysis** tab is where you specify which analyses you wish to run on the selected fracture sets for each iteration of the simulation. Analyses which may be used include:

- Rock Wedge
- Well Fracture Intersection
- Cluster
- Pathway
- Surface Fracture Intersection
- Fragmentation
- Subsidence
- TDV
- HydroFrac
- Well Conditioning Statistics

Check whichever analyses you wish to run from the list of saved analysis definitions on the left side of the dialog box. If you check **Retain Analysis Objects after iteration**, any objects resulting from the selected analyses, such as rock wedges or pathways, will be saved in the file for every

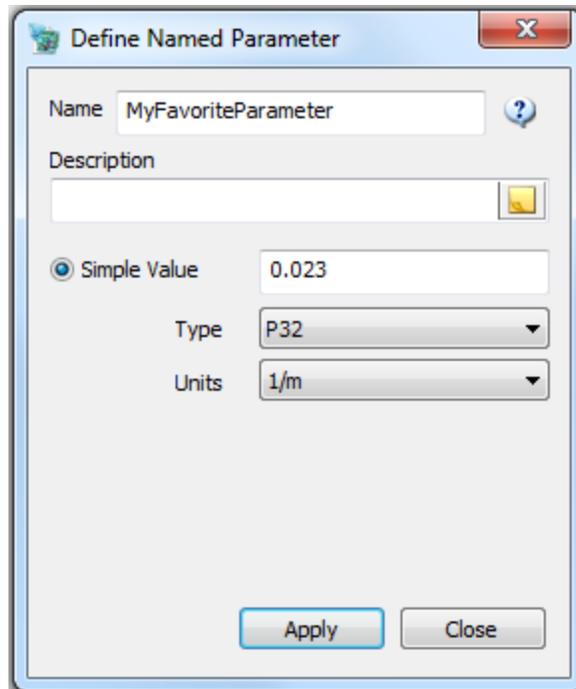
realization of the Monte Carlo simulation, leaving you with as many of these objects as there are iterations of the simulation. Otherwise, the objects will be deleted after each run.

Note that the **Retain Analysis Objects after interaction** option is saved for *each* analysis...first check the analysis so that it is selected, and then check the **Retain Analysis Objects after iteration** checkbox if desired for that particular analysis. You may have this option checked for none/some/all of the selected analyses. The **Clear Generated Fractures** setting on the previous tab will be overridden if an analysis has been selected with the **Retain Analysis Objects after iteration** that needs the fractures (such as the intersecting fractures result object which is a reference set). If it needs the generated fractures in order to maintain the results, the **Clear Generated** setting is ignored and the fractures are kept.

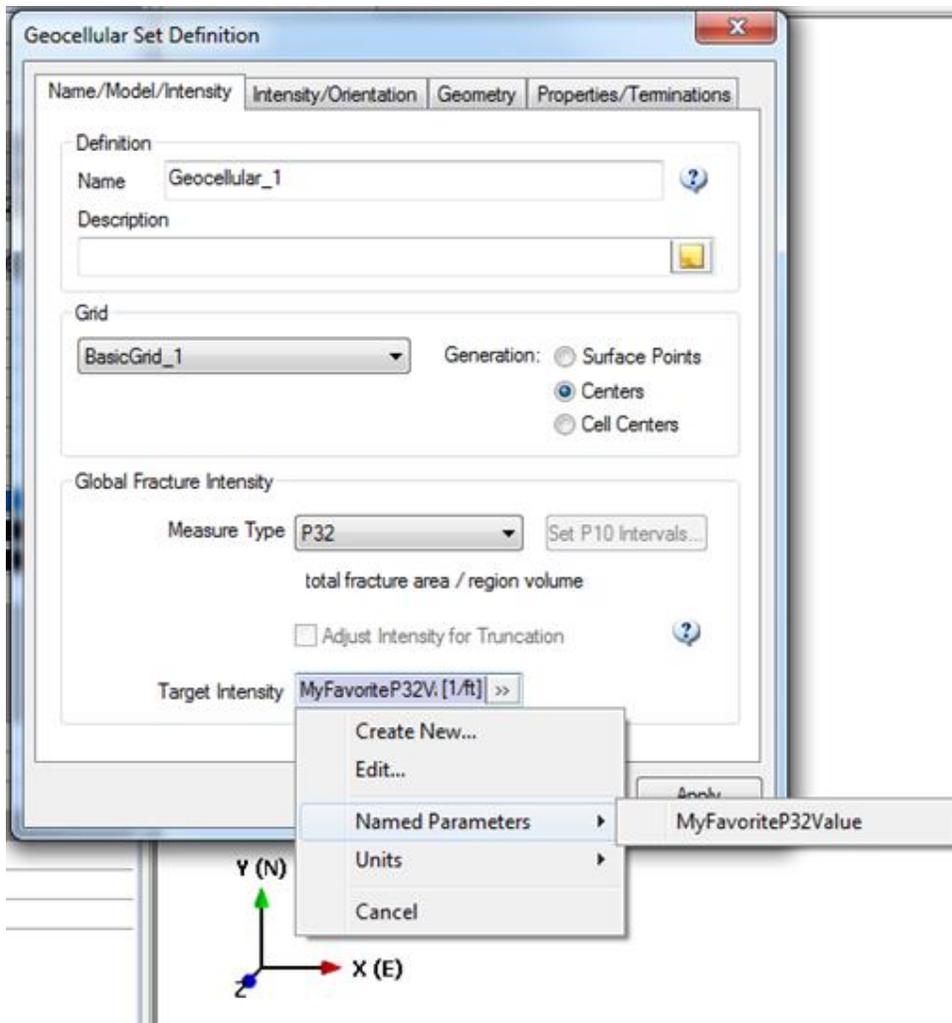
Once you have finished defining your Monte Carlo simulation, click **Apply**. A new definition will appear in the Definitions window under “Monte Carlo Definitions”. In order to run the simulation, right-click on the definition name and select **Run Monte Carlo**. When the simulation is complete a data printout will appear on the screen listing the statistics for the generated fracture sets and the results of the analyses run on these sets for every iteration of the simulation, as well as a mean and standard deviation for these values considering all iteration results collectively. Like all FracMan’s data displays, you may save the information to its own file or copy and paste it into another application, such as Excel, by right-clicking in the spreadsheet and selecting the appropriate option. You can also create a histogram for any row in the spreadsheet in by right-clicking on that row and hitting **Histogram**.

Define Named Parameters

Named parameters are global variables for your project. They are usually a constant value (**Simple Value**). Once defined using this dialog, they appear in the Definitions window under their own category of “Named Parameters”. They can be edited by right-clicking on their definition and choosing “Edit Named Parameter” which will bring up the Define Named Parameter dialog.



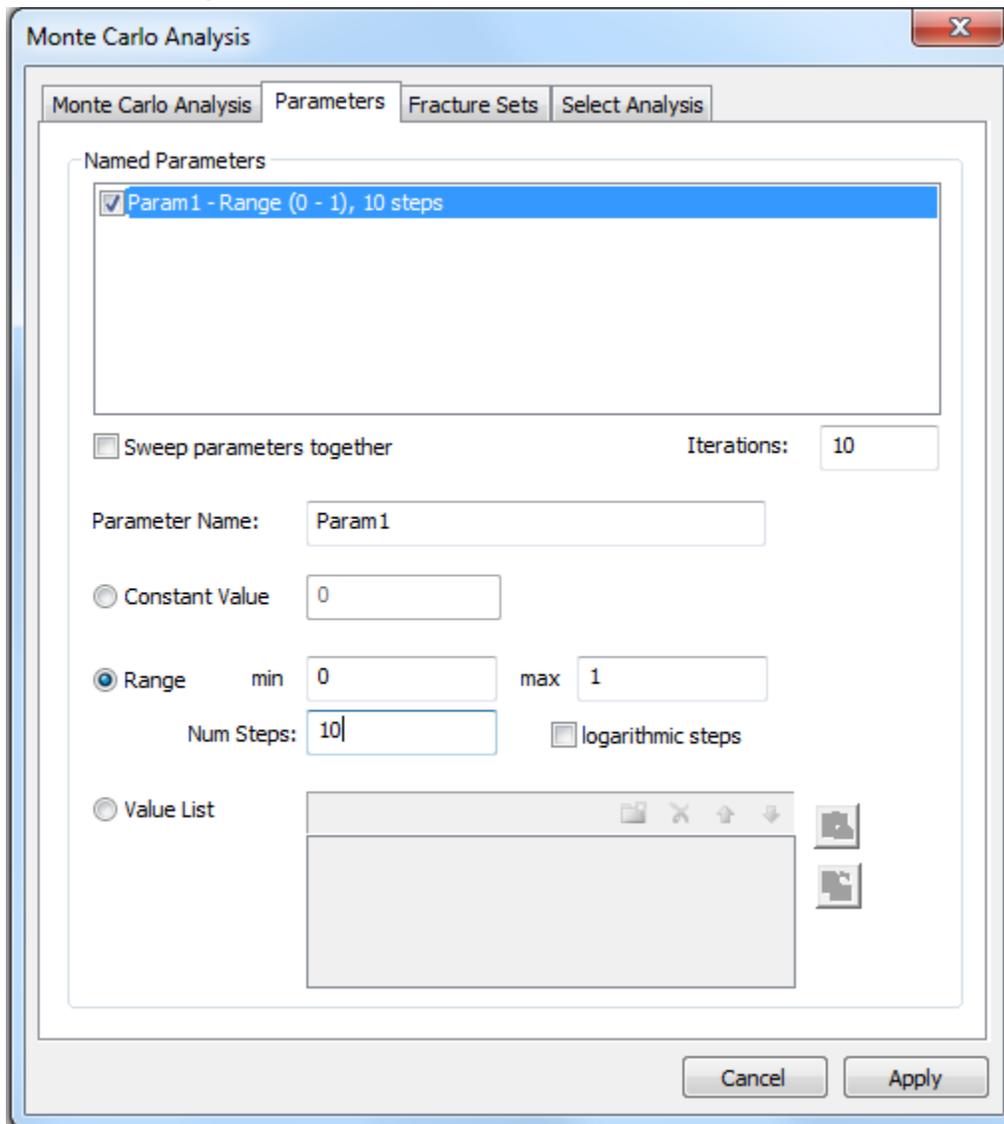
Named Parameters may be used to populate many fields in FracMan dialogs. Fields having the  icon button on the far right can use the Named Parameters (just click on the icon to see a list of Named Parameters having the appropriate units. You can also define a new Named Parameter by clicking on this icon and selecting “Create New...” Named Parameters are also used in the Monte Carlo module [Define Parameter Sweep / Optimization](#).



Define Parameter Sweep / Optimization

The Define Parameter Sweep / Optimization function allows you to run a [Monte Carlo Analysis](#) multiple times using different values for one or more Named Parameters. For optimization purposes, you can then look at the results to choose what value(s) to use in your project for the Named Parameter(s).

Multiple Named Parameters may be selected for the Analysis. To set their values, select them one-by-one in the Named Parameters box and for each, select either Constant Value, Range or Value List to assign their values for use in each iteration.



- *Sweep parameters together:* If you are using more than one parameter, the default behavior is to provide one iteration for every possible combination of parameter values; for instance, three parameters each having three values would result in nine iterations. However, if you check the “Sweep parameter together” option, then only three iterations will be run using 1) the first value for each parameter, 2) the second value for each parameter and 3) the third value for each parameter. If the number of values for each parameter is not the same, then the number of iterations will match the smallest number of values.
- *Iterations:* If the “Sweep parameters together” flag is not checked, this will be a combination of how many different values are run for each Named Parameter. One parameter with 10 values will result in 10 iterations while two parameters, each having

10 values will result in 100 (10x10) iterations. If the “Sweep parameters together” flag is checked, it should equal the size of the smallest value list of all the Parameters. If you are selecting Range for the parameter values, then the Iterations will be set automatically from the combination of min, max and Num Steps.

- *Constant Value:* For convenience, if you want to include a Parameter but not vary its value across iterations, you can set its value here and it will temporarily override what is saved in its Definition.
- *Range:* Use min, max and Num Steps to define what values to use (and optionally the logarithmic steps checkbox).
- *Logarithmic steps:* Used for calculating values defined by Range.
- *Value List:* Click on the Add/Insert icon  to directly enter values, one per line. Click once to get an input box and then, after entering your number, click it again to insert it. Values may also be loaded into the list from a file having the values saved one per line or saved to a file using the  icon located on the far right.

Macros

A macro is a simple program that does the same or a very similar operation several times in a row. FracMan allows you to record and edit macros of all program functions, thus enabling you to run large numbers of simulations without having to actually be there while they run. Macros and macro control are discussed in greater detail in the [“Using Macros”](#) section of this manual. The menu items include five Macro Functions as well as a list of the most recently created Macro files.

- Run Macro
- Edit Macro
- Record Macro
- Stop Recording
- Ignore Macro Errors

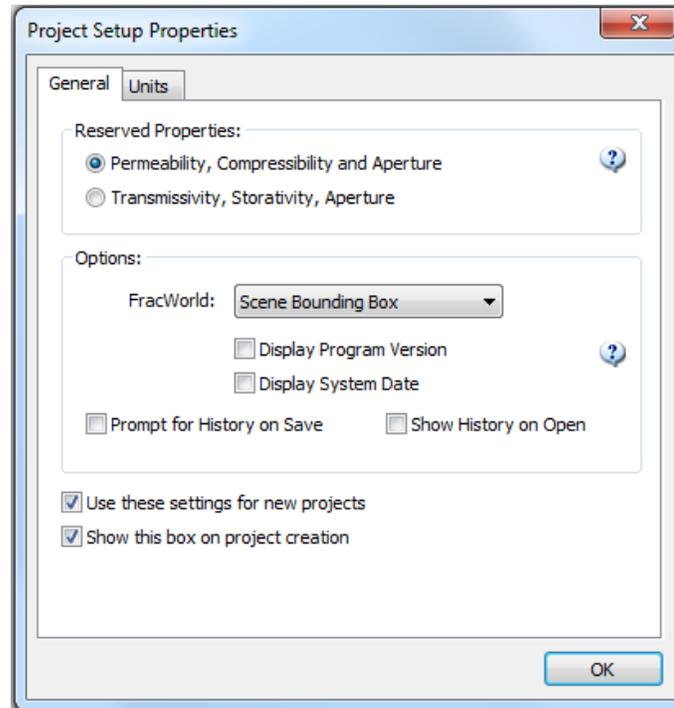
Language

This allows you to select which language you wish the FracMan menus and dialogs to appear in. Currently, FracMan supports three languages: English, Chinese and Russian. In order to toggle between languages, just mouse over the language button and click on the language you prefer. You may change this at any time.

Project Setup

This allows you to set project options such as units. While you can use different units throughout the project, it is very helpful to set these as appropriately as possible before beginning your model. Also be sure to set your fluid properties appropriately whether you are working with water, oil or gas.

Reserved Properties



Many of FracMan’s analyses require one or more of the “Reserved” Properties to be present, thus these properties will always be available on any fracture set. Depending on your situation, you may prefer to work in **Permeability/Compressibility/Aperture** (PCA) which is more common in the oil and gas industry or **Transmissivity/Storativity/Aperture** (TSA) which is more common for hydrological projects. In either case, if FracMan needs to convert from one to the other it will use the Fluid Density, Compressibility and Viscosity specified for **the Fluid Properties** in the Units tab.

Conversion between PCA and TSA

Fracture Transmissivity/Permeability:

$$T_{frac} = \frac{P_{frac} \rho g A_{frac}}{\nu}$$

Fracture Storativity/Compressibility:

$$S_{frac} = (C_{fluid} + C_{frac}) \rho g A_{frac}$$

Matrix Transmissivity/Permeability:

$$T_m = \frac{P_m \rho g}{\nu}$$

Matrix Storativity/Compressibility:

$$S_m = (C_{fluid} + C_m) \rho g \varphi$$

Where,

T_{frac} = fracture transmissivity [m²/s]

T_m = matrix transmissivity [m/s]

S_{frac} = fracture storativity [unitless]

S_m = matrix storativity [1/m]

A_{frac} = fracture aperture [m]

P_{frac} = fracture permeability [m^2]

C_{frac} = fracture compressibility [1/Pa]

C_{fluid} = fluid compressibility [1/Pa]

C_m = matrix compressibility [1/Pa]

ρ = fluid density [kg/m^3]

g = acceleration of gravity [m/s^2]

ν = fluid viscosity [$kg/(ms)$]

ϕ = matrix porosity [unitless]

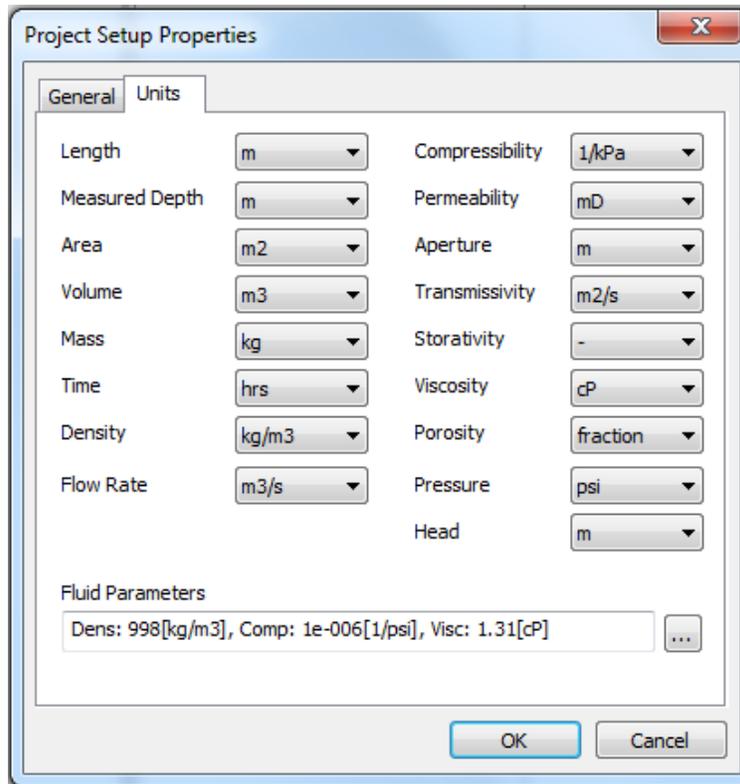
Project Options

Here you can specify a “FracWorld” object (see [Select FracWorld Region](#)), and specify whether the program version and/or system dates are shown on the Visualization window. If you check **Prompt for History on Save**, you will be shown the [Project History](#) dialog when saving the project where you can enter a note. Press the Save icon button after entering a note to continue saving the project. If you check **Show History on Open** then you will be presented with the Project History dialog each time the project file is opened.

Use these settings for new project – check this box to make the values entered the defaults for new projects.

Show this box on project creation – check this box to show this dialog every time you start a new file.

Project Units

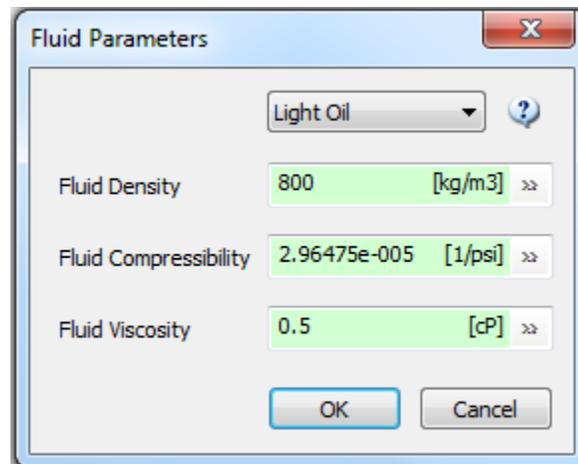


Units are displayed in each window and dialog. There is great flexibility in setting the units appropriately throughout the program. Specification of units is an important step as part of data import and export. Below is a table of supported units.

Length	cm, ft, m, km
Measured Depth	ft, m
Area	ft ² , m ²
Volume	ft ³ , m ³ , usgal
Mass	kg, lbs, mt
Time	sec, min, hrs, day
Density	g/cm ³ , kg/m ³ , lbs/ft ³ , ppg
Flow Rate	bbbl/d, bpm, fpd, fps, gpm, lpm, m ³ /s, m ³ /d
Compressibility	1/psi, 1/atm, 1/bar, 1/kPa, 1/Pa
Permeability	mD, D, m ²
Aperture	mm, cm, in, ft, m
Transmissivity	ft ² /s, m ² /s
Storativity	unitless
Viscosity	cP, Pa-s
Porosity	fraction, %
Pressure	psi, kpsi, Mpsi, atm, bar, Pa, kPa, MPa, GPa, mHg
Head	ft, m

Note that if you change the units of your model after you have added objects or definitions, FracMan will not automatically convert values – you will need to check through all the objects of your model and make conversions using [Translate/Scale Geometry](#) (object coordinates) or [Converting Well Log Units](#).

Fluid Parameters – Enter average values for the density, compressibility and dynamic fluid viscosity of the fluid in your model by clicking on the button located on the right of the text box. You may select default values for Water, Light Oil, Heavy Oil and Gas or enter your own values. These parameters are used to convert properties between the PCA and TSA systems as requested by the user or required by the FracMan functions utilized in the project. The default units are those specified in the Units tab but can be changed by clicking on the “>>” buttons on the right side of the text boxes and selecting from the Units options.

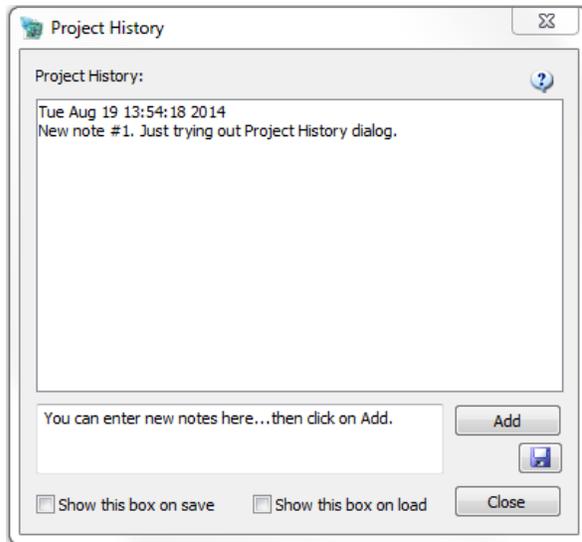


Export Project Notes / History

Creates a text file containing the project file pathname, project notes, object names in a tree structure as well as any object notes.

Project History

You can enter notes about your project at any time using the Tools Menu -> Project History. This window can also be set to show up when you load or save your project by setting the appropriate checkboxes in the Project Setup dialog on the General Tab. Project notes along with any object notes can be exported to a report using the Tools Menu -> Export Notes / History function.



The Window Menu

This tab has commands to help you control how FracMan's internal windows are displayed.

There are five commands:

- **New Window.** This creates a duplicate of the window you currently have open.
- **Close All Report Windows.** This closes out all open windows except for the Visualization window.
- **Close All Dialogs.** This closes out all open dialog windows.
- **Tabbed Windows.** This will toggle between tabbed window mode and classic mode.
- **High Resolution Screen Capture.** Same function as listed in the [View Menu](#).

Beneath these commands will be a list of all open windows. A check marks the one currently displayed. To display a hidden window, just click on its name in this list, and it will appear on top of the other windows. If multiple windows are visible, the check marks the active window – the one currently selected.

The Help Menu

This offers resources to aid you in better using and understanding FracMan.

- **Contents** opens the FracMan Manual, allowing you to search for and find information on a specific command.
- **About FracMan** gives program information: version, installation date, user name and company, license expiration date, and video card information. It also gives links to contact the FracMan group directly.
- **FracMan on the Web** takes you directly to the FracMan website. This will only work if you have an active internet connection.
- **Enter Registration Codes...** allows you to register your software. This is discussed in greater detail in the "Installing FracMan" section of this manual under [License Activation](#).

- **Check For Updates** checks for updates to FracMan online. See description under [Version Check](#).

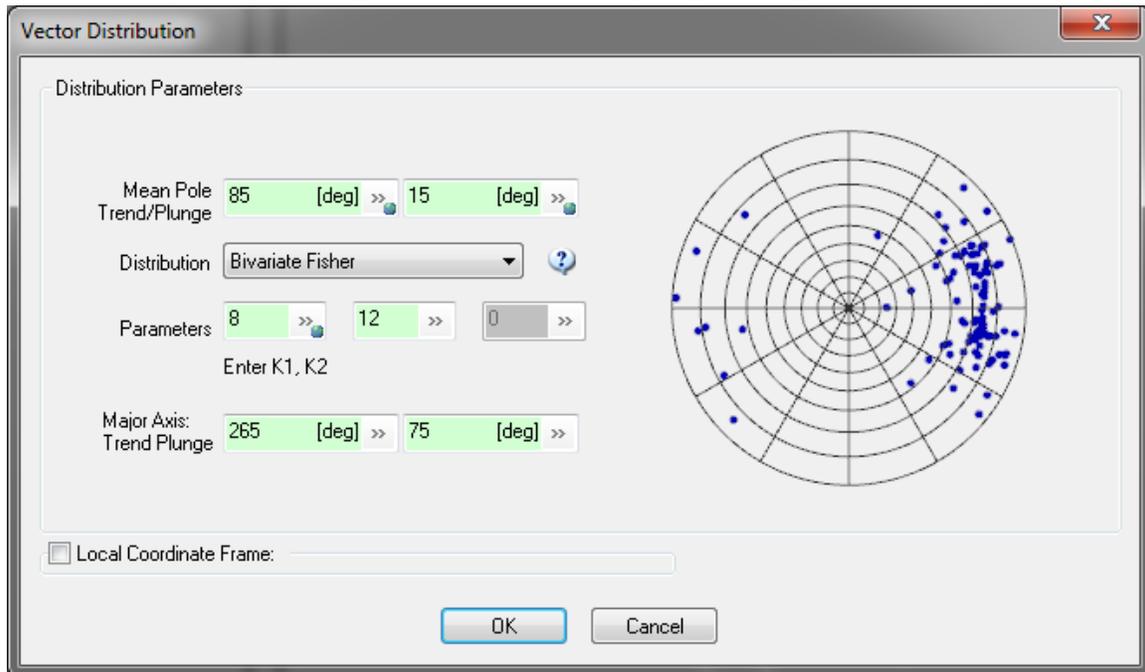
Building Your Model

Distributions

FracMan has two different kinds of distributions whose parameters you can set: Vector and Scalar. The same dialog boxes for these two distributions will appear frequently in fracture generation and elsewhere.

Vector Distribution

The vector distribution allows you to define some distribution with multiple related parameters, such as orientation (which includes both a trend and plunge).



The parameters you must enter here depend on what distribution you choose from the drop-down menu. The “K” for a Fisher distribution, for instance, is a concentration constant showing how wide or narrow the range of values is in the distribution. The smaller the K, the more disperse the values are, with a K of zero resulting in a random distribution. The “Concentration” parameter for a Bootstrap distribution behaves in a similar same way except the dispersion acts around each individual bootstrap value rather than a single Fisher pole”.

For the Fisher and Bingham distributions, provide values of the Fisher/Bingham concentration coefficients k_1 (and k_2 if bivariate). Specify k between 1 and 5 for high orientation variability, and between 20 and 50 for low orientation variability. Use one or two negative values in the Bingham distribution for unimodal or bimodal distributions, and positive values for girdle distributions. For the Bivariate normal, provide standard deviation values for trend and plunge, and the correlation coefficient, which can be either 1 (perfect correlation) or 0 (no correlation).

For the Bootstrap distribution, enter the Fisher distribution concentration coefficient to be applied around each point from the bootstrap data. A stereo plot is generated in the dialog box based on the chosen distribution and parameters.

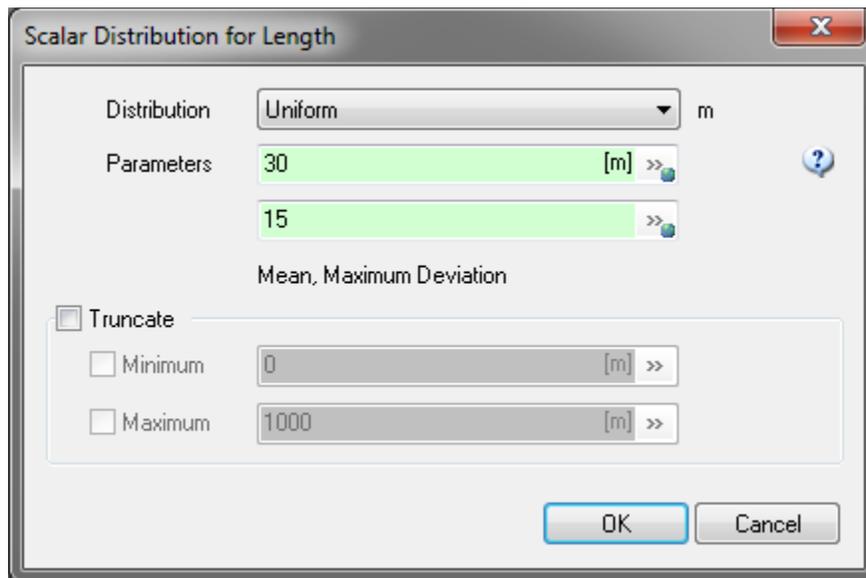
See [Orientation](#) for a detailed description of these distributions.

Local Coordinate Frame – allows you to provide a local coordinate system as defined by a grid, so that vectors are distributed relative to grid – orientations will then be generated relative to this local reference frame.

Scalar Distribution

A scalar distribution creates a distribution of single values, for something such as fracture length (which has only one value unrelated to other properties).

The parameters you must enter here depend on what distribution you choose from the drop-down menu. See [Probability Distributions for Scalar Values](#) for a detailed description of the available probability distributions.



The image shows a dialog box titled "Scalar Distribution for Length". It has a standard Windows-style title bar with a close button (X) in the top right corner. The dialog contains the following elements:

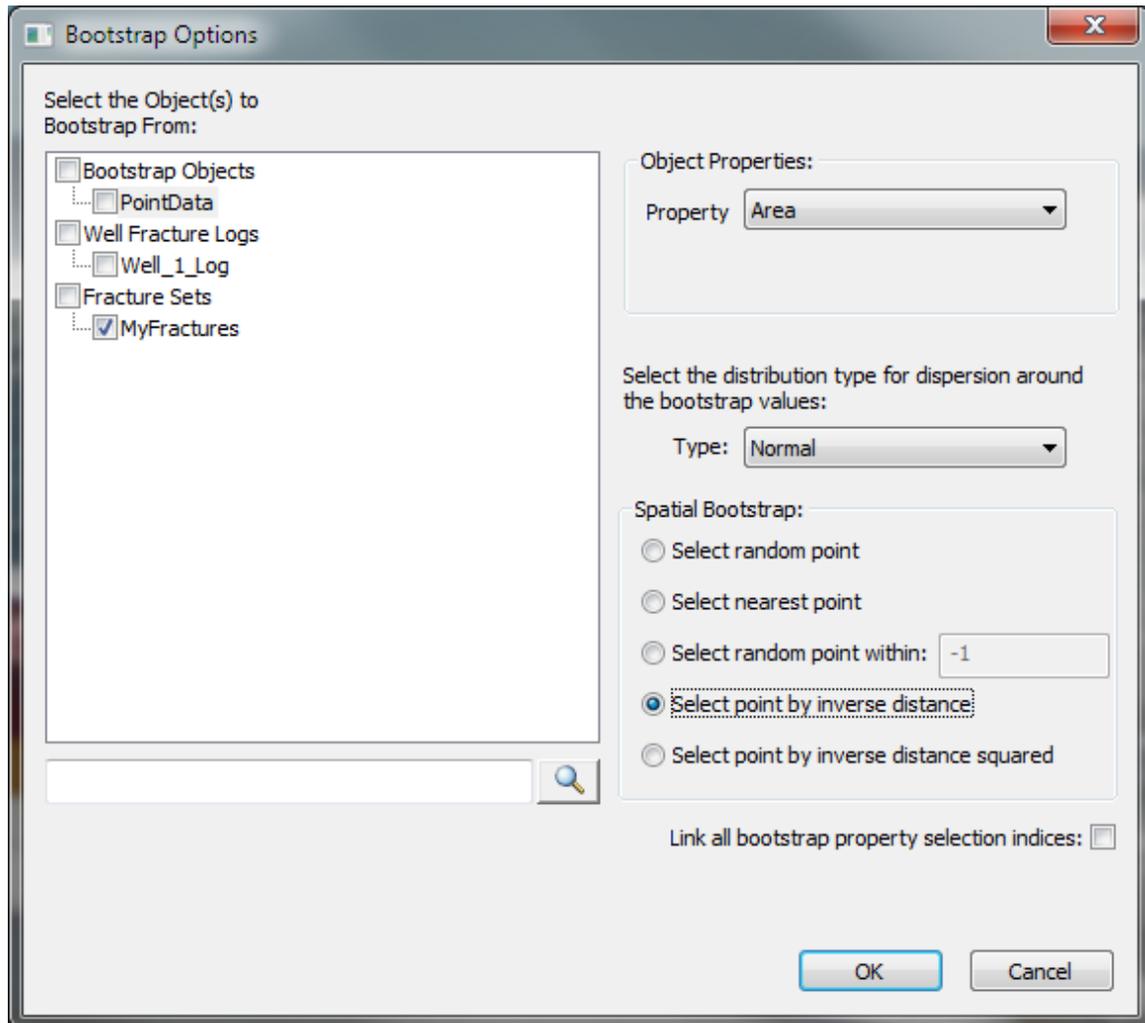
- Distribution:** A dropdown menu set to "Uniform" with a small "m" unit indicator to its right.
- Parameters:** Two input fields. The first contains "30" and the second contains "15". Both fields are highlighted in light green. Each field has a "[m]" unit indicator and a right-pointing arrow button. A blue question mark icon is located to the right of these fields.
- Mean, Maximum Deviation:** Text centered below the parameter fields.
- Truncate:** A checkbox that is currently unchecked.
- Minimum:** A sub-section containing an unchecked checkbox and an input field with "0" and a "[m]" unit indicator and arrow button.
- Maximum:** A sub-section containing an unchecked checkbox and an input field with "1000" and a "[m]" unit indicator and arrow button.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

If you check “Truncate”, FracMan will filter out the largest and/or smallest values of the distribution, depending on the user’s specifications. Values below the given minimum or above the given maximum will be discounted and replaced with numbers more amenable to the calculation. You must check “Minimum” and/or “Maximum”, dependent on which truncation bounds you wish to define. Then enter the bounds in the given boxes.

If you select a “Bootstrap” distribution, you must click on the **Bootstrap** button next to the distribution parameters to set your bootstrap options.

Scalar Bootstrapping

You can use data in your model to guide your fracture generation by selecting a “Bootstrap” distribution. Bootstrapping draws its values from existing data rather than from a random distribution.



First check all of the objects you wish to bootstrap this property from.

- **Object Properties:** Select the property containing the values you want to choose from.
- **Type:** This allows the user to specify which distribution is used for bootstrapping. Select “Constant” to use the exact values from the bootstrap data, or select a scalar distribution and enter the corresponding parameter on the Scalar Distribution dialog to get values near your bootstrap values. For example if you select “Normal” and enter a deviation parameter of 10, each bootstrap value will be used as the mean of a normal distribution with deviation of 10, so you will get values normally distributed within 10 of your bootstrap values. This option is quite useful when bootstrapping to sparse data.

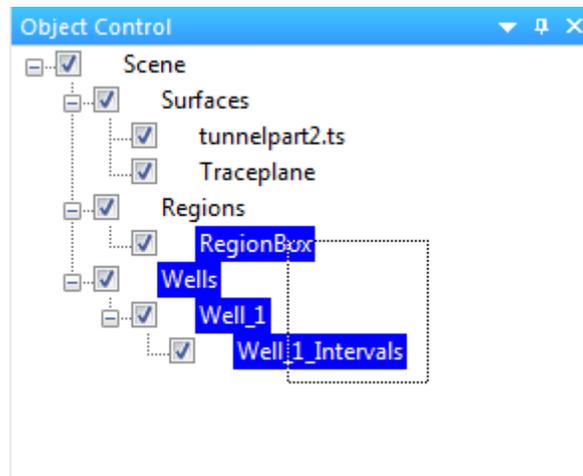
- **Spatial Bootstrap:** This allows you to specify (for objects that support it) whether to pick a point at random, select the nearest point, select a point at random within a radius, or to select a point at random weighted by either inverse distance or inverse distance squared.
- **Link all bootstrap property selection indices:** When this is checked, FracMan links all bootstrapped properties for the same fracture, such that each individual fracture takes all of its bootstrapped properties from the same source (i.e., another individual fracture).

Click the **OK** button to complete the Bootstrap definition and return to the Scalar Distribution dialog.

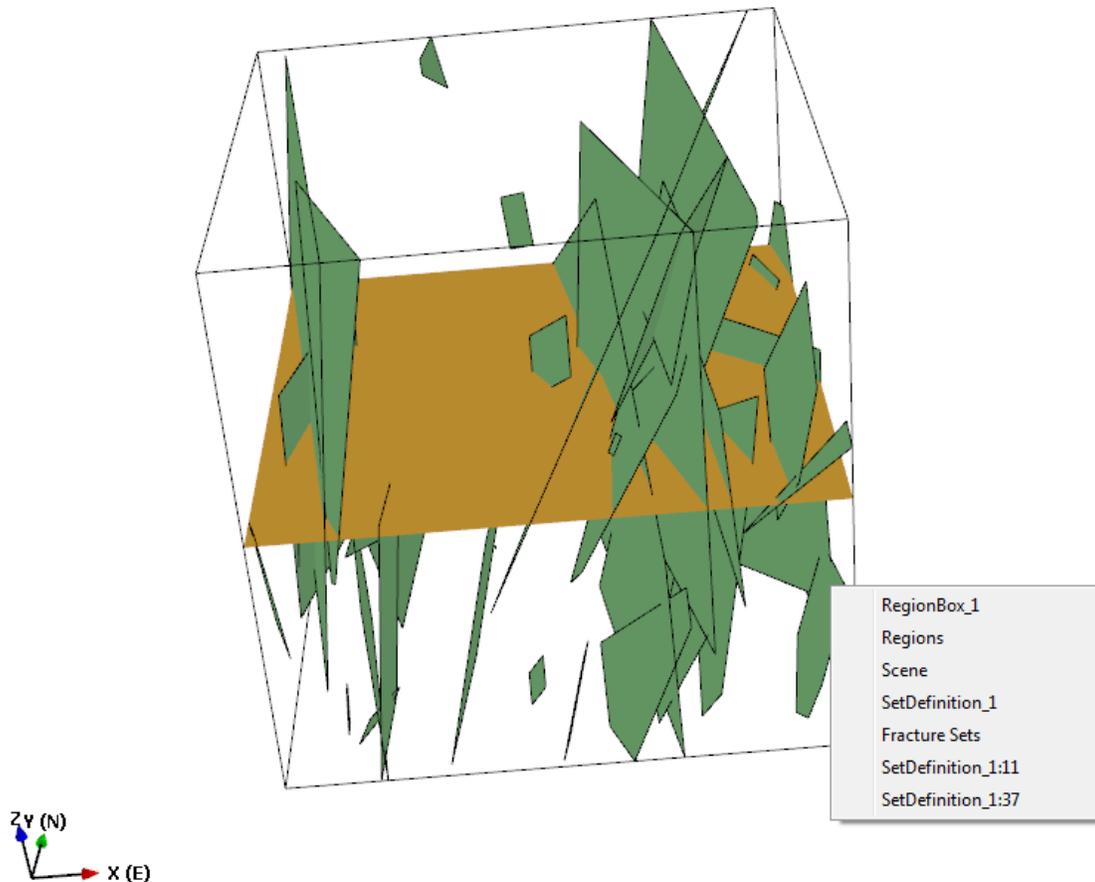
Interacting with Objects

Selecting Objects

FracMan operations are generally performed on one or more selected objects. The attributes bar shows the aggregate attributes of all selected objects. If all selected objects are of the same type you will have access to edit all of these objects at once. You can select one or more objects by clicking on them in the object bar. You can also drag select to select multiple objects at one time.



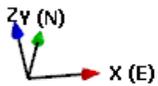
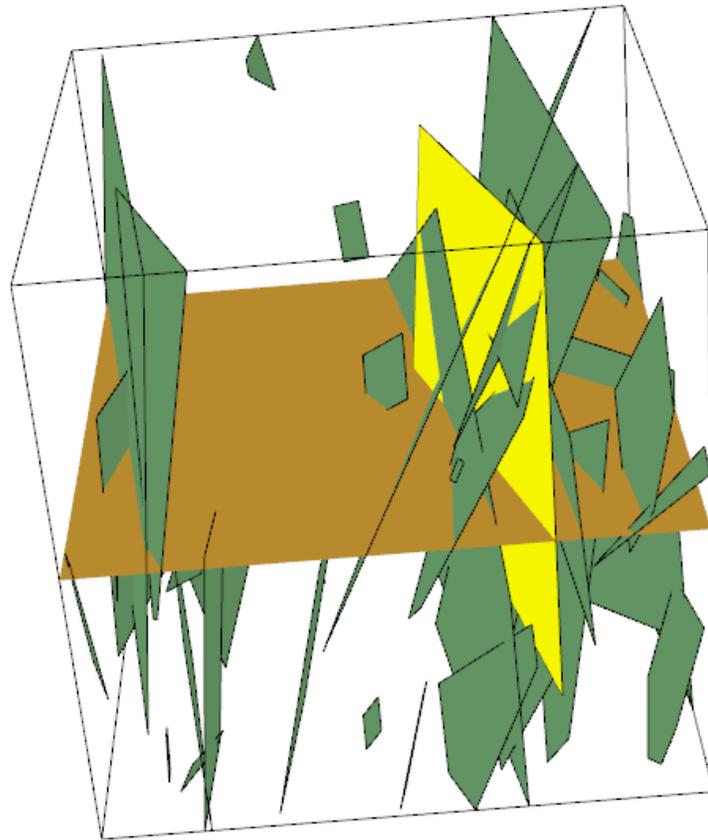
Another way to select objects is to click on them in the visualization window.



When you do, you will get a list of all the objects under the cursor. Note that in addition to objects you will also get any stochastic fractures in the list. Selecting an item from the list will cause that object to be selected in the Objects window.

Selecting Object Elements

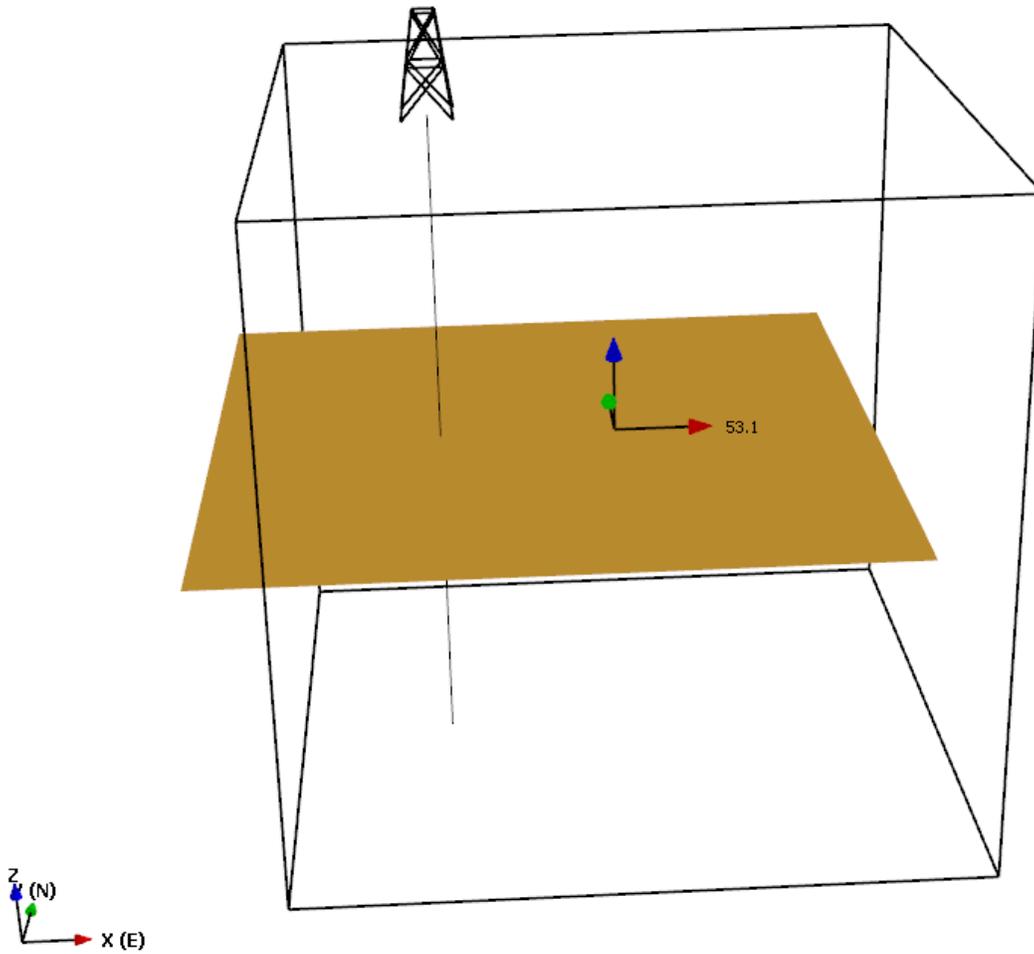
Objects in FracMan often are made up of elements such as Faces, Vertices and Cells. You can select one or more sub elements of an object by selecting the object and then selecting either the vertex  or face  selection mode buttons in the toolbar. Now you can click on the visualization window to select the desired element(s). Once in face or vertex selection mode, you can select multiple elements using the drag select mode button . Selected elements will appear highlighted in the visualization window and the attributes of that element will appear in the Attributes bar.



If two vertex elements are selected, the attributes window will display the distance between them.

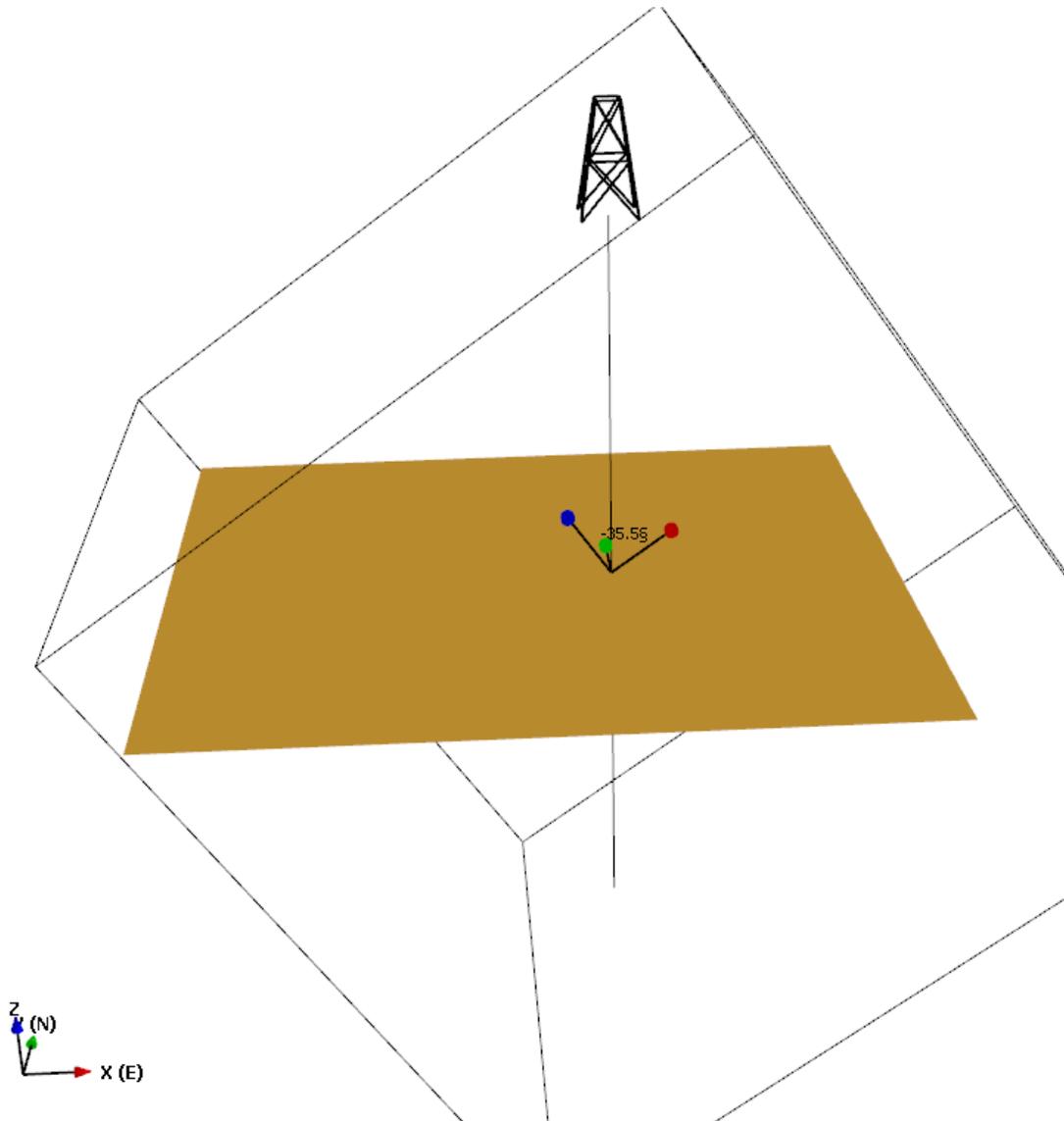
Mouse Translate / Rotate

To translate an object, select the object / objects and press the “Translate Objects” button  in the toolbar. The translate axes will appear in the center of the selected object. Click and drag any of the colored arrow points to drag the object along that axis.



A small black number will appear next to the axis along which you are translating your object telling you by how many units you are moving it.

The rotate object function works similarly. Select the  button from the toolbar. Then click on the circular axis indicators and drag to rotate the object about that axis.



A number will appear next to the axis about which you are rotating your object, telling you by precisely how many degrees the object has been rotated.

Rotation and translation operations can be undone by selecting *Edit > Undo* from the menu bar.

Translate Object Elements

Some object elements can be moved directly in the visualization window. Surface, well, grid and tracemap vertices, well log entries, and well intervals can all be moved in the scene.

Delete Object Elements

Some object elements (Surface faces, tracemap segments, well log entries, fractures) can be deleted by selecting them and selecting “Delete Selected” from the Edit menu or by pressing the Delete Key. Element deletion can not be undone.

Object Properties

Many Fracman objects such as fracture sets, surfaces and grids can have element properties. These properties assign discrete values to each face, vertex or cell in the object.

Visualize Properties

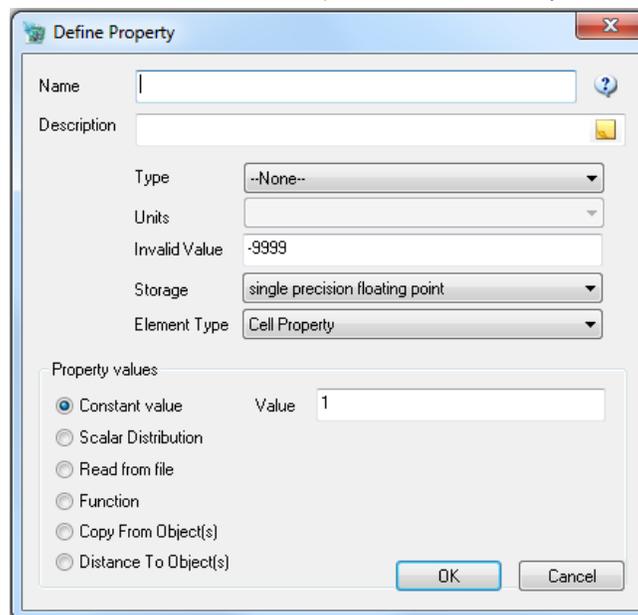
You can visualize property values by right clicking on the object(s) and selecting the property name from the “Visualize” menu. See [The Legend Bar](#) for more information about visualizing properties. You can do the same thing by selecting the property in the Property bar and clicking on the “Visualize” icon located on the right side of the bar. Also see the section on [Timeline](#) to visualize how properties are changing over time.

Edit Property Values

Properties can be manipulated with many functions that are available either from active buttons on the right side of the Properties window once the property is selected, or from the menu available by right-clicking on the property itself.

Add / Edit Property

To make a new scalar property, hit **Add Property**. This will open a new dialog asking you to define the property. **Edit Property** opens the same dialog box. If you want to create a new vector property, use the [Define Orientation Properties](#) functionality.



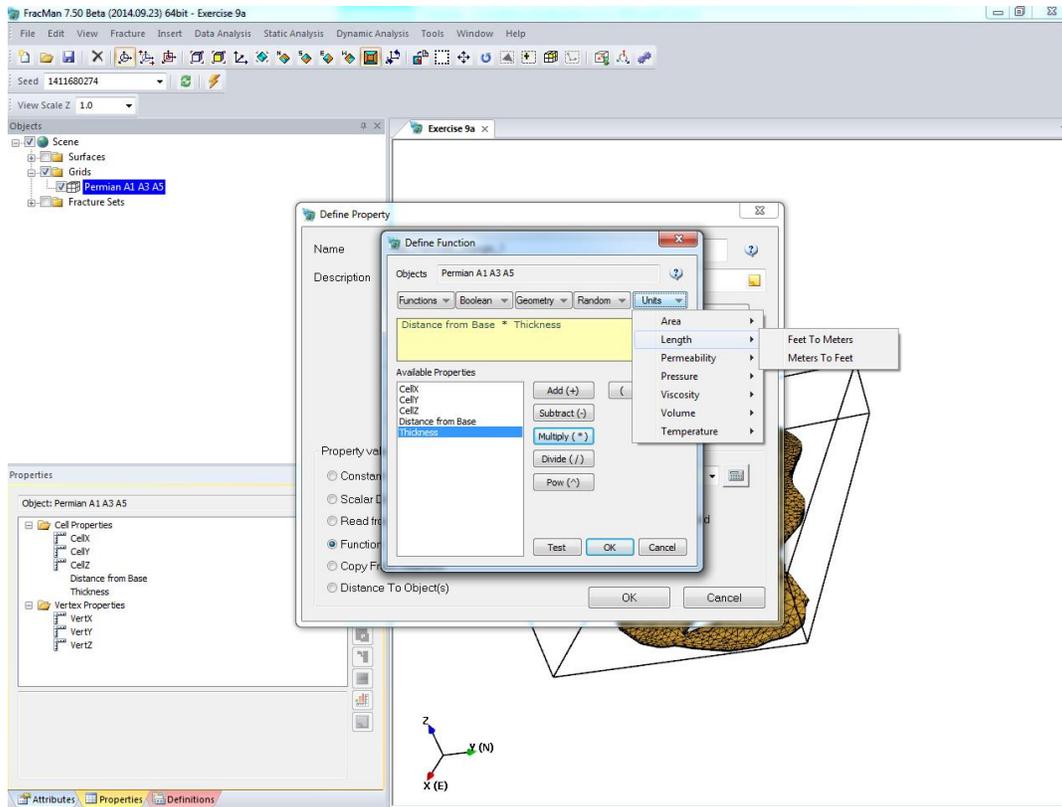
When adding a property, first specify the property’s *Name*. Do not change the name if you are editing the property, instead, use the **Rename Property** function. A description of the property can be added. Click on the note icon on the right side of the **Description** input box to enter a longer comment. The Description field can be reached directly from the Properties bar by selecting the property and clicking on the Property Notes icon located on the right side of the Properties bar.

Optionally, you can specify the parameter *Type*. Available types include:

- Angle

- Aperture
- Area
- Compressibility
- Custom
- Density
- Depth
- FlowRate
- Force
- Head
- Length
- Mass
- MatrixStorativity
- MatrixTransmissivity
- --None-- (default)
- P10
- P32
- Permeability
- Porosity
- Pressure
- Stiffness
- Storage
- Storativity
- Temperature
- Time
- Transmissivity
- UnitStrength
- Velocity
- Viscosity
- Volume

Once you specify the parameter type, the units for that type will be displayed. The units are set according to those specified in the [Project Setup](#) Properties. If you want to use different units, you can convert using the Define Function dialog (see figure below) or use the [Set Property Units](#) function.



You must also specify what *Invalid Value* to use for the property. An invalid value is a flag that signifies that the property value is out of bounds, indicating to FracMan that it should ignore these values during analysis. Be sure that you pick an invalid value appropriate to the property type and definition – that is, choose a number that would never occur for this property unless something were very wrong. For instance, -999 would be a good invalid value for a length property, but a poor invalid value for a property defined by a normal distribution with a mean of -900. The value -9999 is provided as a default. Boolean property types do not use an invalid value.

Next, you must choose the *Storage* type for your property. The value for each cell is stored on your computer as a specific type of data, and this allows you to specify exactly what type you want this property to be. There are four options. These are, in order from smallest disk space occupied to largest:

- 1-bit Boolean – either a 1 or 0, essentially specifying true or false
- 32-bit signed integer – integral values used primarily for flags
- Single precision floating point – (default) standard values
- Double precision floating point – used for higher precision values but requires double the amount of memory

Be sure that you choose an appropriate type for your property. A Boolean, for instance, may be good for the grid “Active” cell property, but it wouldn’t be much use for storing fracture

apertures. If used properly, this specification can radically decrease the amount of space your object takes up in memory.

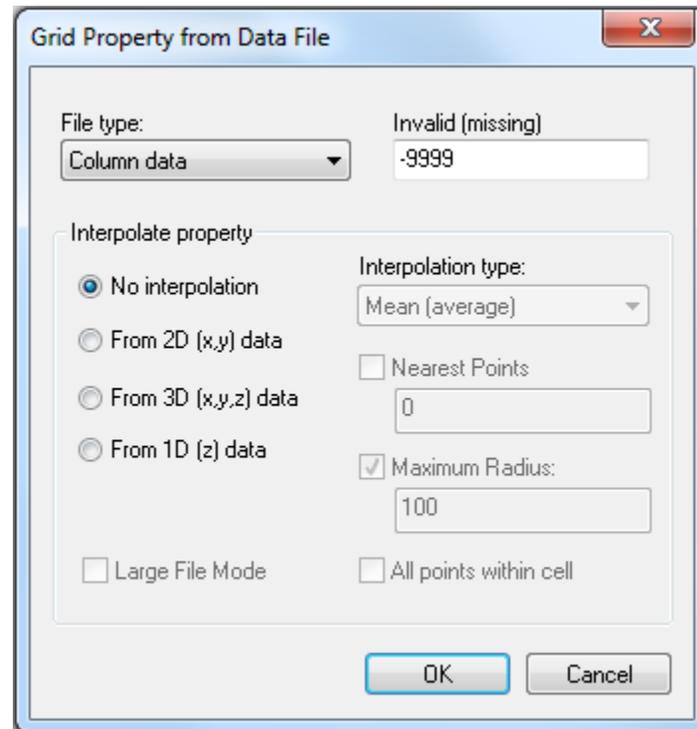
Next you must tell FracMan what values to use to define the property. There are six different ways to do this.

- **Constant value:** select this if the property is the same at every point in the object.
- **Scalar Distribution:** a probabilistic distribution i.e. Normal, Log Normal etc. See [Probability Distributions for Scalar Values](#) for the details of the different distributions available.
- **Read from file:** You can import property values from a number of different data file formats. The values will be interpolated and applied to the current object. See extended discussion in section below.
- **Function:** This allows you to base the property on a function using other properties as variables, thus correlating this property to another. All properties defined using a function are forced to be of type double precision floating point.
- **Copy from Object(s):** Allows you to interpolate property values for this object from another object's properties.
- **Distance to Object(s):** This will compute the distance of elements of this object to elements of another. The objects are resolved into points (generally using vertices) and you can select from the min/max/mean distance value. *This function operates on object elements only (faces, vertices etc) it does not compute object intersections.*

When you are done, hit **OK** and your property or the changes you have made to it will be saved and applied to the object.

A few of the ways to define a property deserve a more in depth treatment. They are detailed below.

Read from File

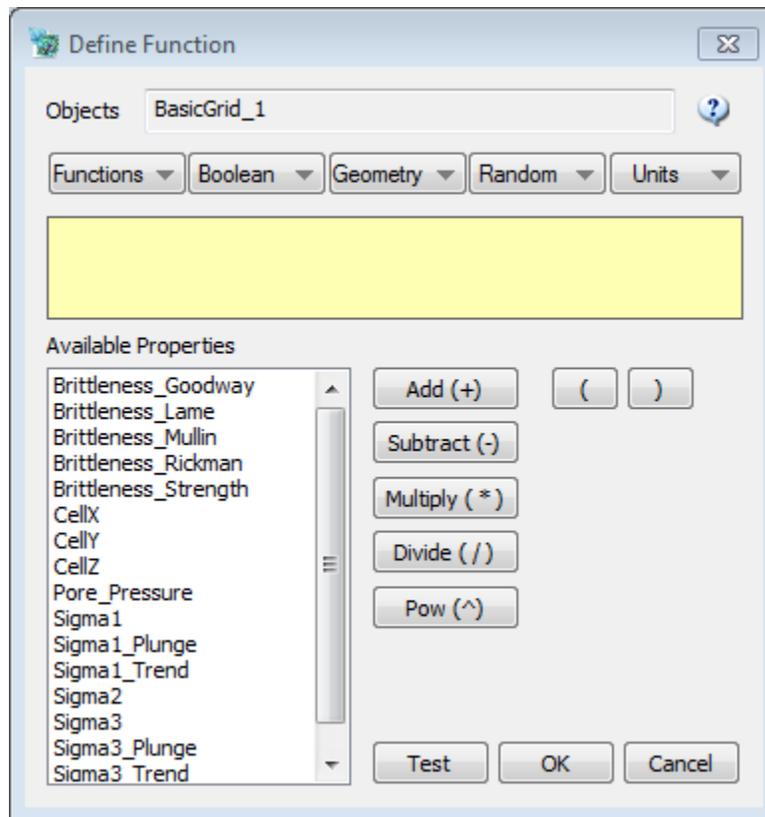


If you are adding a new property from an external file, click on the “Browse” button to select the file on your computer using the **Property Data File** dialog. Specify whether the type is a generic Property Data File (*.*) or a SEGY Seismic file (*.segy). Click **Open** to get the import dialog specific to the object type. For instance, when loading a grid cell property the **Grid Property from Data File** dialog will appear. Here you specify again the file type. Various options are available for the different file types:

- **Column Data:** this is a generic file type where you need to specify which columns contain the property value and, if interpolating, what which columns contain the point coordinates. Simple point-to-point interpolation is available from 2D, 3D or 1D data sets where the interpolation type can be by Mean, Median, Min, Max, Sum or Count using **Nearest Points**, points within a **Maximum Radius** or **All Points within cell**. Checking the **Large File Mode** will maximize the interpolation performance for large file sizes (recommended for file sizes larger than 1GB).
- **Eclipse Keyword Property:** the property file must match the grid exactly with the same number of property values as grid cells. The only available option is to enter the (Invalid missing) value. The property name will be taken from the first line of the data file. See [Appendix A](#) for details on the input file format.
- **SEGY Seismic Data:** selecting this format will possibly load your seismic data without any further input. Due to wide variations in industry interpretations of this standard, if you are trying to import a SEGY file and this option does not work for you, try the more versatile file import tool [File Format Assistant](#) accessible from the File->Import menu. If that fails, contact FracMan support and we will assist you.

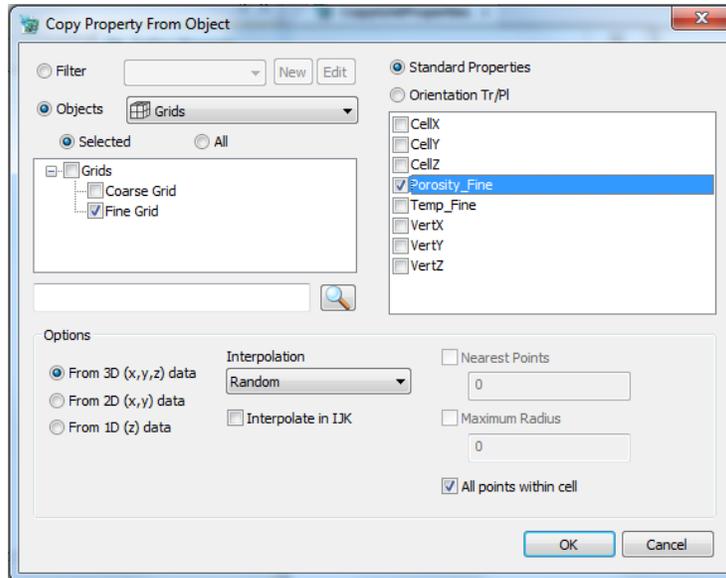
Function

A number of functions have been added to the equation processor. The mean/min/max/stddev/sum/random/random inverse distance/rand inverse distance squared of any property can be referenced. There are predefined functions for converting between dip and pole as well as for computing dot and cross products. “Trend/Plunge” interpolation is also available.



Copy from Object(s)

Property values can be copied from one object to another, with interpolation options for matching objects having different data resolution. Simple point-to-point interpolation is available from 2D, 3D or 1D data sets where the interpolation type can be by Mean, Median, Min, Max, Sum, Count, Random, Random (Inv Dist), or Random (Inv Dist Sq) using **Nearest Points**, points within a **Maximum Radius** or **All Points within cell**.



Random: Select a random point.

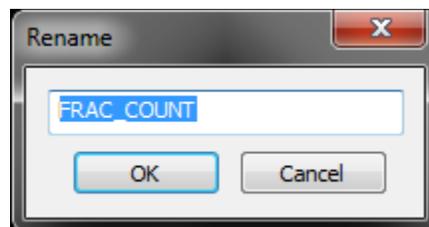
Random (Inv Dist): Select a random point weighted by distance.

Random (Inv Dist Sq): Select a random point weighted by distance squared.

Interpolate in IJK: If you are copying properties to a Grid object from another Object contained in that Grid, then you can check the interpolate in IJK to use the grid coordinates instead of the XYZ coordinated during the interpolation. For example, you may have a grid based on inclined bedding surfaces and a well contained within that grid that you use to calculate P10 values for each formation. You may want to assign P10 values to planes in the grid corresponding to the well/K-layer grid intersection depths.

Delete Property

If you wish to remove a property, select it in the “Current Properties:” list and hit **Delete Property**. If you wish to change a property’s definition, select it and hit **Edit Property**. If you only wish to alter its name, however, **Rename Property** opens a small dialog asking you to specify a new name for the selected property, nothing else.



Duplicate Property

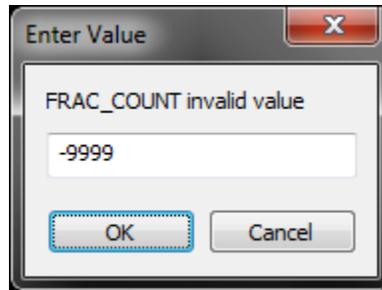
Access the Duplicate Property functionality by right-clicking on the Property you wish to duplicate. A new property will be created having the same name with the string “_copy” added to the end.

Rename Property

Use this function to rename the property, it will not be done correctly if you try to do this using Edit Property.

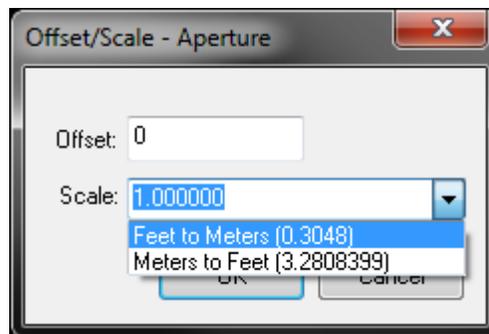
Set Invalid Value

Change Invalid Value will only allow you to modify what value serves as a flag to FracMan that the selected property has gone out of bounds and should be discounted. Boolean property types do not use an invalid value.



Offset/Scale Property Values

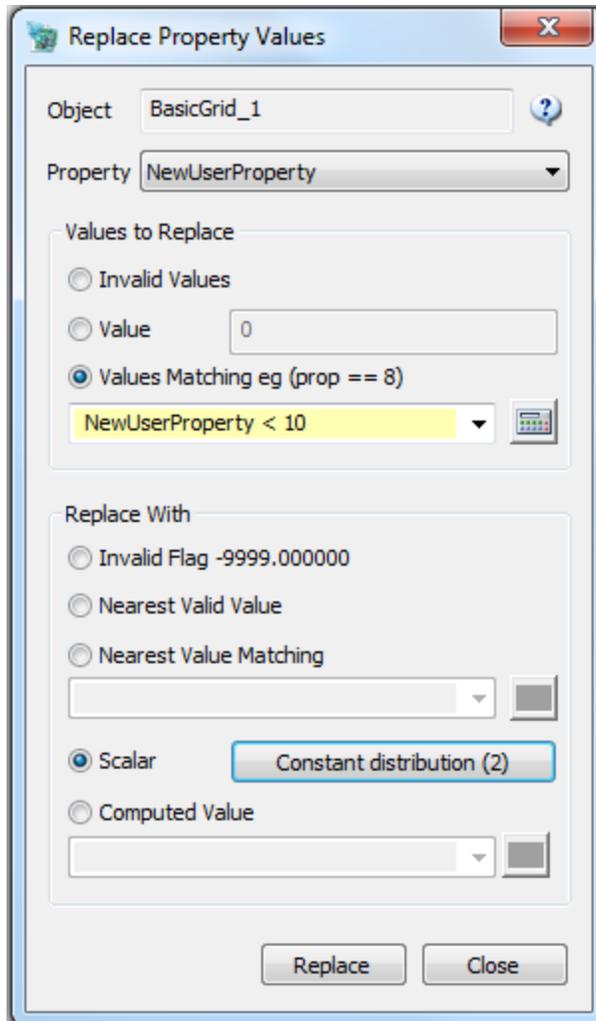
Offset/Scale Values allows you to change the selected property at every location by the same simple transformation. For instance, you may select a unit conversion under the “Scale:” drop-down menu; or if you found an error in your original calculations, you might add some compensatory modifier in the “Offset:” box.



Replace Property Values

The user has the option to replace any property value in most objects in FracMan. The Replace Property Values dialog can be opened by right-clicking on any property in the property window (or using the respective button in the property window) and selecting the “Replace Values”.

The user has the option of not only replacing Invalid Values but Values matching any statement the user specifies using Boolean Functions with the ‘Values Matching’ fields.



Values to Replace

- Invalid Values – the property will be scanned for invalid flagged values and replaced.
- Value – Look for an exact match (appropriate for Integer values).
- Values Matching – The entered Boolean function will be evaluated at each element and non-zero values of the expression will be replaced.

Replace With

- Invalid Flag – Replace with the Invalid Value.
- Nearest Valid Value – Flagged values will be replaced with the nearest non-invalid value from the object.
- Nearest Value Matching – The entered function will be evaluated at each element and flagged values will be replaced by the nearest value where the evaluated function is non-zero.
- Scalar – Flagged values will be replaced either by a constant or a random number as specified by the Scalar Distribution selected.
- Computed Value – Flagged values will be replaced by the value of the entered expression.

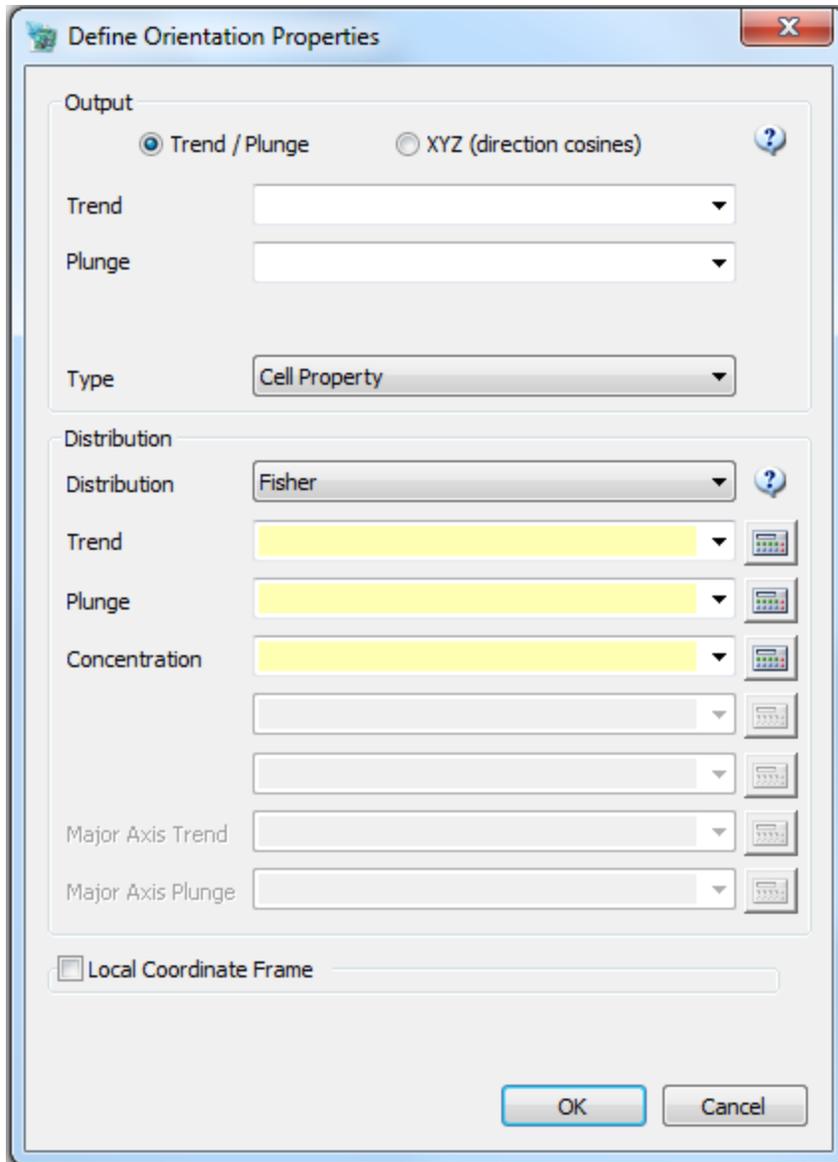
Visualize

Show the object in the Model View with this property colored according to the Legend. This is the same function as selecting the object, right-clicking, selecting Visualize and selecting this property.

Define Orientation Properties

There may be times when you want to define vector properties (i.e. those having an orientation). This could occur for instance with a variable stress field that you would like to correlate with a grid property including some type of random angular dispersion. Or perhaps you want to fit some microseismic data to a complex stress orientation model. This is an alternative way of creating a new property beyond what is available using the Add or Edit Property functionality. There is in fact not one new vector property created, but a set of scalar properties which together define the vector property, two angular properties for trend/plunge or three xyz direction cosine properties.

First create a new property using the Add Property icon which contains the vector property magnitude. Then in the Properties window, select this new property and right-click to select the Define Orientation Properties to create or assign the angular properties needed to define the orientation of the vector.



Output

- Select either Trend/Plunge or XYZ (direction cosines) to specify the output desired.
- Enter new names or select existing property names for the resulting Trend/Plunge or XYZ (direction cosines). The names should reflect a common origin, say StressTrend and StressPlunge to remind you of their association.
- If this is a grid property, select the Type: Cell Property or Vertex Property.

Distribution

Select the type of orientation distribution desired. The required parameters for the selected distribution will appear as input fields below. You can use constants or functions (which can be correlated to other parameters). See the Manual section on [Orientation](#) for details about the different distributions available.

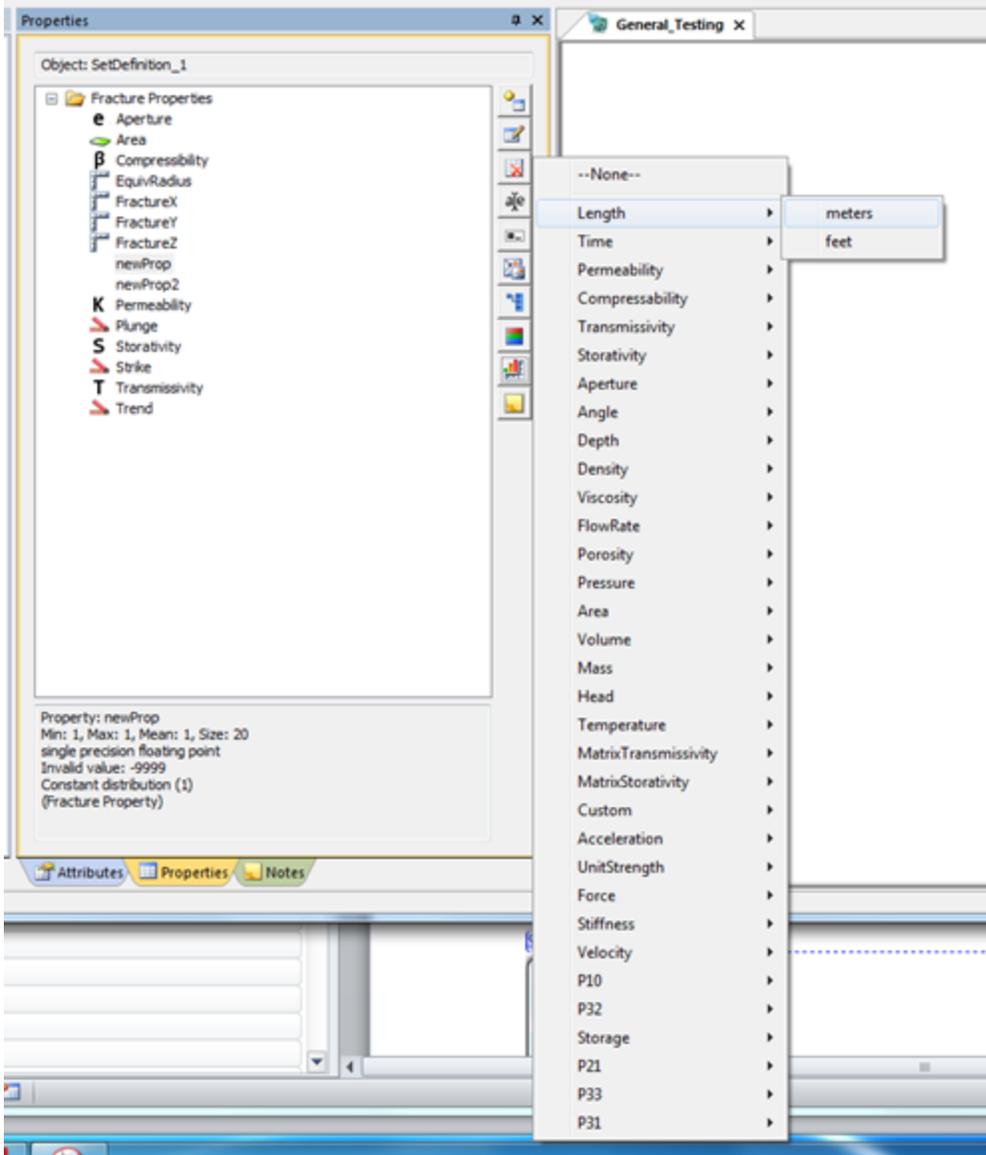
Local Coordinate Frame

Checking this box allows you to provide a local coordinate system as defined by a grid, so that vectors are distributed relative to the grid – orientations will then be generated relative to this local reference frame.

Set Property Units

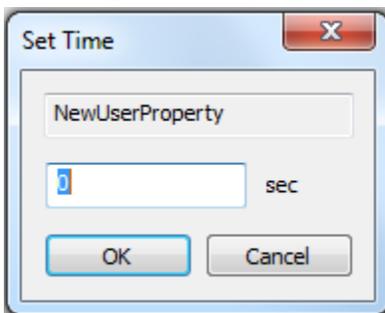
Property units can be assigned to a property in the Add or Edit Property dialogs, or directly using the Project Units icon from the Properties bar. While user-defined properties are not required to have a unit associated with them, it is strongly recommended that one be assigned in order to allow FracMan to make unit conversions when necessary, and warn you if you try to use an inappropriate property for an input.

Warning: If you want to convert your property from one unit to another, do so using a function call in the Edit Property dialog, not the Set Property Units interface. Setting the property unit tells FacMan what the units are of the current values. Changing the units here will not trigger any conversions to the current values.



Set Property Time

Property Time is used for visualizations using the View ->Timeline menu option. These visualizations can be used to create an animation of a series of properties on an object (like well test results) or for visualizing a single property where the values represent time (like hydraulic fracturing). Generally the Time property is automatically generated within FracMan.

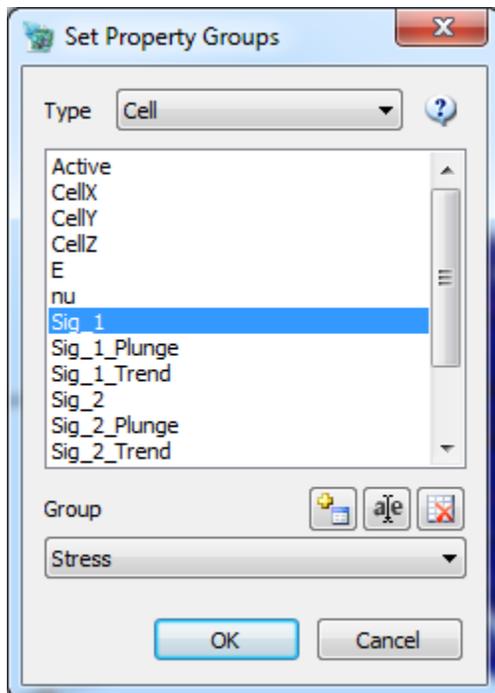


Set Property Notes

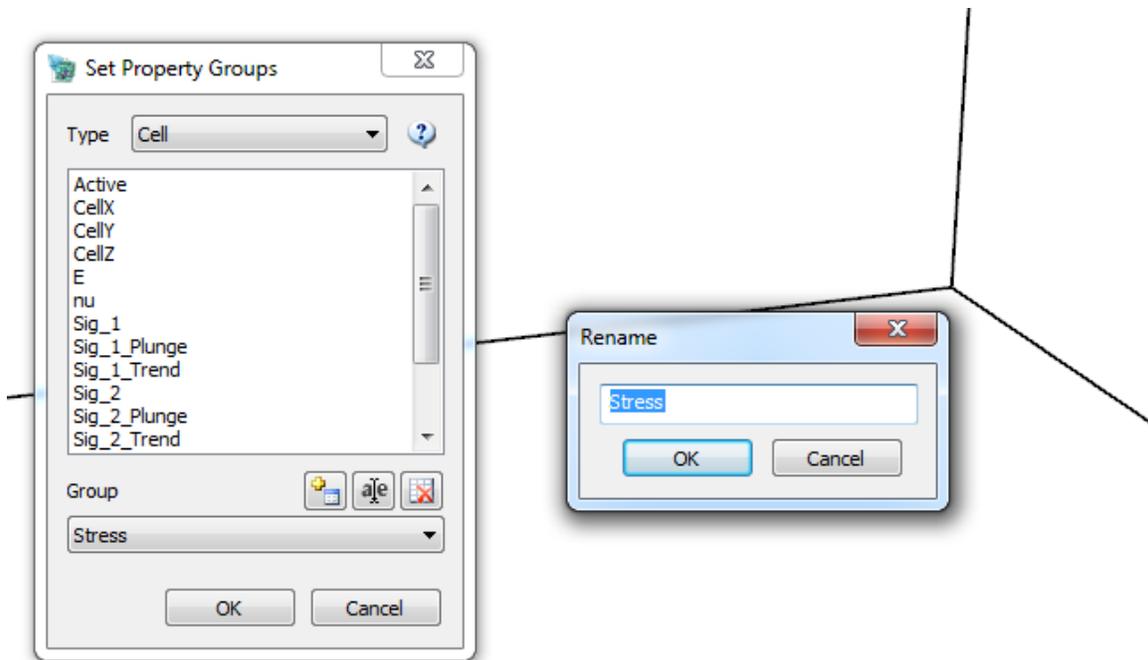
Click on this icon to access the selected property's Description field. Values can be edited. This field can also be modified from the Define Properties dialog using Edit Property.

Property Groups

Properties can be organized into groups using the **Set Property Groups** dialog. Right-click on the property and select Property Groups-> Add Property Group to add the selected property to a new group.



The new group will have a default name which can be modified using the Rename Property function. To add another property to this group, just drag it to the new group in the Properties window.



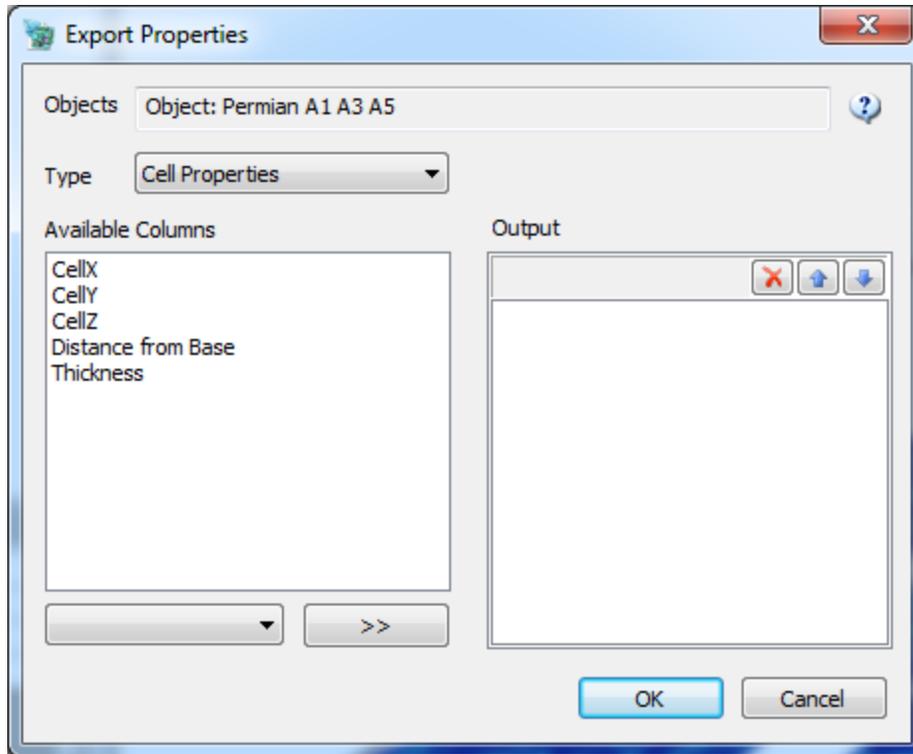
Attributes

Some objects when selected will display their properties in the Attributes Bar. The values displayed are the mean of the selected object(s) properties. Changing the values in the attributes bar sets the values of all selected objects or object elements to the entered value. Properties drawn with a gray background cannot be edited.

Attributes	
Properties (mean)	
Aperture	1e-006
Permeability	1000
Compressibility	1e-006
Size	9.30577
Trend	222.695
Plunge	76.8485
Area	449.485
Strike	148.347

Export Property Values

All object properties can be exported to an ascii file. Right click on one or more objects and select "Export" followed by the property you want to export. You can also export property values by selecting "Object Properties" from the "File->Export" menu. This mechanism will export multiple properties into a single tab delimited file.



Several formats are supported depending on the object type:

- **Generic Property** – Property values are listed in a single column.
- **Cell Indexed (Regular Grids Only)** – Property values are listed along with the cell index (I, J, K).
- **Cell Centered (Grids Only)** – Property values are listed along with the X, Y, Z values of the cell center.

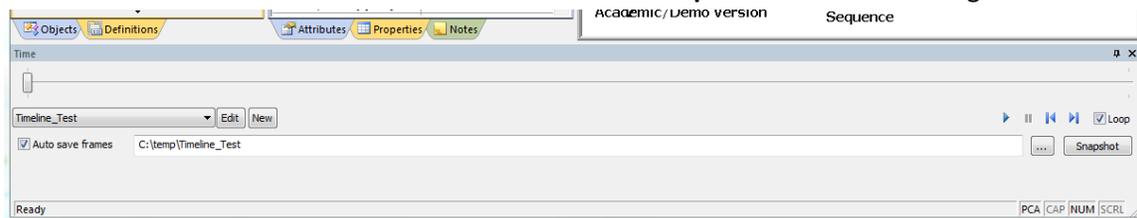
Histogram

Property values can be displayed as a histogram. See the section [Histogram](#) for details on how to use histogram charts

Timeline

The Timeline panel is used to display an animation of a series of properties on an object (like well test results) or for visualizing a single property where the values represent time (like hydraulic fracturing).

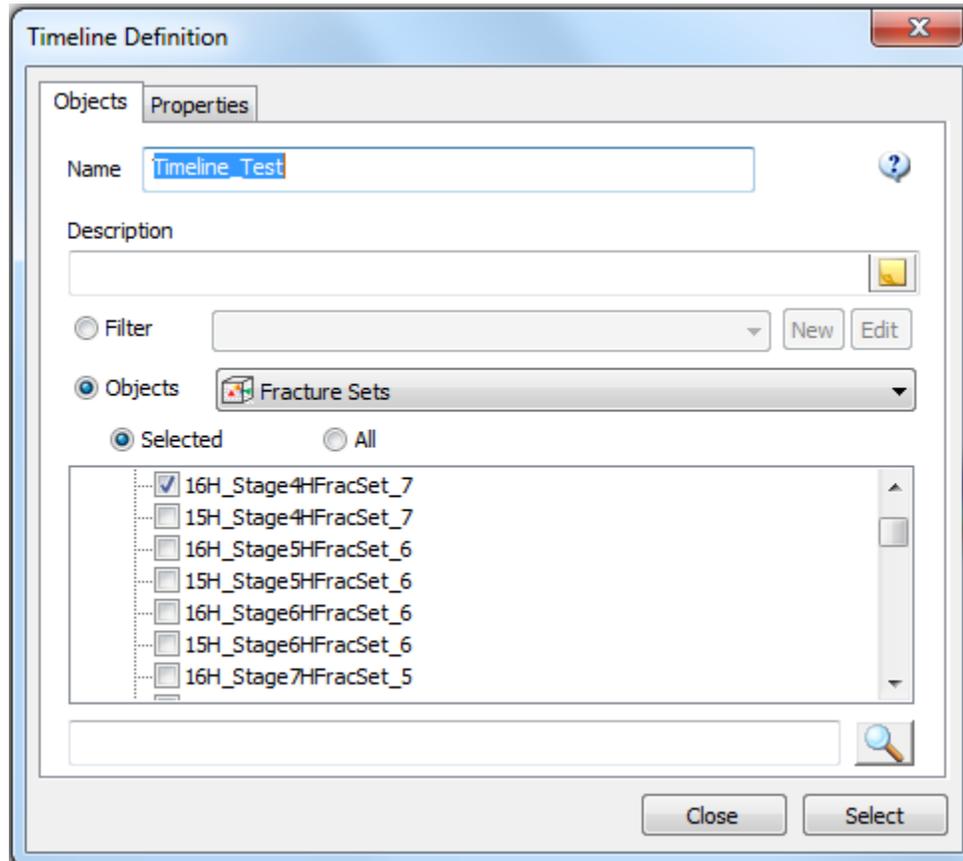
To define a new Timeline, make sure that the Timeline panel is active in the **View** menu. [Re-size](#) your FracMan panels if necessary to ensure that the Timeline panel is large enough to show the checkbox for **Auto save frames** at the bottom and the **Loop** checkbox at the far right.



From the **Objects** panel, select one or more objects to display in the timeline sequence. It can be helpful to [Hide](#) any objects that are not of interest in the timeline.

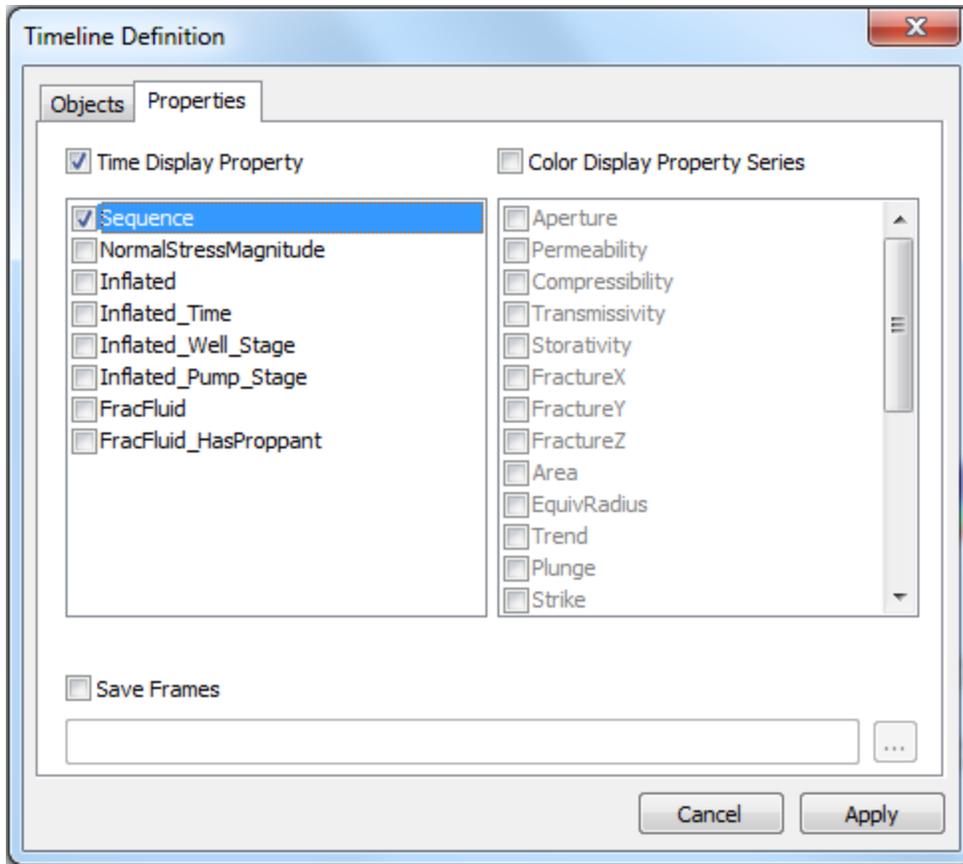
Creating a New Timeline Definition

Select **New** from the Timeline panel to display the Timeline Definition dialog.



The objects that were selected in the Objects panel should be pre-selected on the **Objects** tab. If not, you can also choose them here by using the Object checkboxes or Filter. The search field at the bottom of the tab can be used to find your object if there are a large number.

On the **Properties** tab, usually you will just want to check one property to visualize through time, either a Time Display Property or a Color Display Property. If you select more than one, they will be cycled through one at a time. For example, to show the changes to an induced fracture during hydrofracturing, select the **Time Display Property** 'Sequence'.



If you want to save a picture of each time step, check **Save Frames** and choose a directory and enter a new filename prefix in the **File name** field for the exported pictures using the **Export Frames To** dialog opened by clicking on the '...' button.

Click on **Apply** to save your definition and **Select** to get it ready to display. Click on **Close** to close the dialog.

Playing a Property Timeline

Once you have a timeline defined, select it from the drop-down menu on the Timeline panel. Double check that your object is visible and correctly oriented in the Visualization panel. Use the **Play Animation**, **Stop Animation**, **Rewind** and **End** icons at the top right corner of the Timeline panel to control the Visualization. If you want the animation to restart from the beginning once it has reached the end, check the **Loop** checkbox. If you want to grab a picture of one of the timeline steps, click on the Snapshot button while the time step of interest is showing (you can click on **Stop Animation** to make sure you get the view you want). The picture will be saved in the directory specified in the definition with a filename created from the entered **File name**. The directory and file name prefix may alternatively be specified directly in the Timeline panel in the field next to **Auto save frames**. Checking **Auto save frames** will save a picture of each time step in the Visualization into the directory specified with a file name created using the entered **File name**. This is the same field as the one below **Save Frames** in the **Timeline Definition** dialog.

Regions

Regions in FracMan are usually convex polyhedron and are used to define where fractures should be generated and also provide borders where surfaces and fractures should be clipped. They define volumes for calculating P_{32} fracture intensities and P_{33} porosities. They can be used to define the visible areas of a project and can be transformed into surfaces.

Regions can be created using the **Insert** menu options [Region Object](#), [Box Region](#), [Cylinder Region](#), [Prism Region](#), [Slab Region](#) and [Horseshoe Tunnel](#). See those Manual sections for details on region creation. Once created, regions can be modified by selecting them in the **Objects** window and then adjusting values in the **Attributes** panel. Individual vertices can be relocated by selecting them in the **Visualization** window using tools such as **Area Select** and **Translate Objects**. Care should be taken with this latter approach as customized, non-regular polyhedron may not give the desired results.

Operations available to regions may be accessed by right-clicking on a region object. The following menu options appear:

- Zoom to Selection: same as [Zoom Extents Selected](#) in the Toolbar.
- [Clip Fractures](#)
- [Intersecting Fractures](#)
- Move to Object(s) Bounding Box
- [Convert Region to Surface](#)
- [Convert Region to Panels](#)
- Set FracWorld Region: sets the **Frac World** region to the selected region object. Use the **Tools -> Select Frac World Region** dialog to undo the change.
- [Set Visibility Region](#): only appears as an option if it will cause the visibility region to change.
- Export

Move to Object(s) Bounding Box

Convert Region to Panels

This function creates new **Surface** objects for each of the polygons comprising the **Region** object polyhedron. The original **Region** object is retained.

Export Region

Surfaces

Surfaces are used in FracMan to help define model regions and to create non-planar fractures. While box and prism regions work well for local models, most reservoirs cannot be modeled adequately using simple geometries. FracMan provides three ways to use surfaces to define a fracture generation region:

- Generate fractures between two surfaces (see [Creating a Fracture Set using the Geometric Model](#))

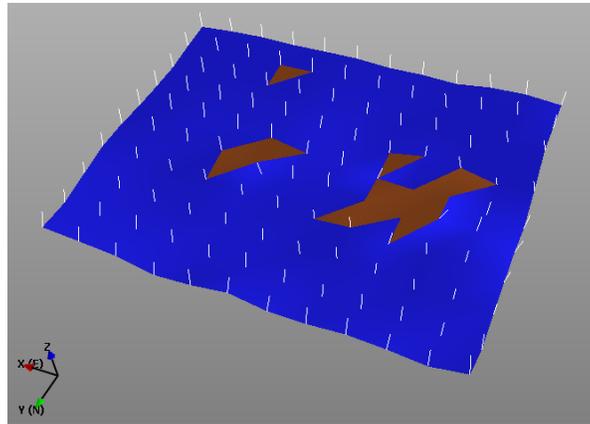
- Create a geologic region from surfaces and create fractures within this region (see [Geologically Conditioned Fracture Models](#))
- Create fractures in a geometric or geocellular region and use surfaces to clip fracture networks (see [Fracture Set Clipping](#))

See [Appendix A FracMan Supported File Formats](#) for a complete list of the file formats supported for surfaces. FracMan does not interpret all the information contained in any particular file format; only the necessary geometric information is read.

Surface Creation

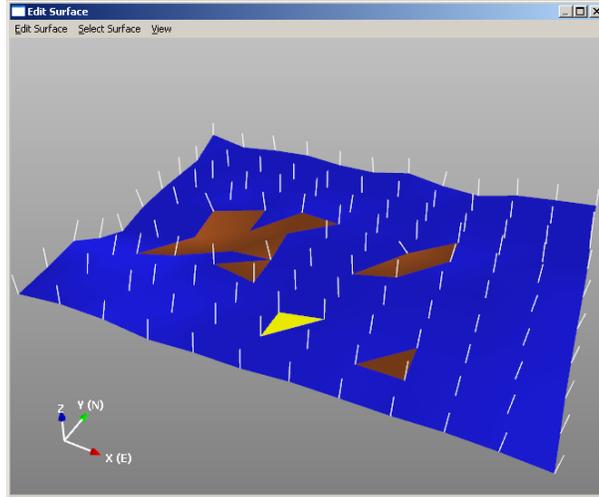
Edit Surface Normals

Some surface when imported from other software will have irregularities in their surface orientation which can cause problems in analysis. You can correct these issues by reversing the normals of individual faces so that all faces have a uniform normal direction

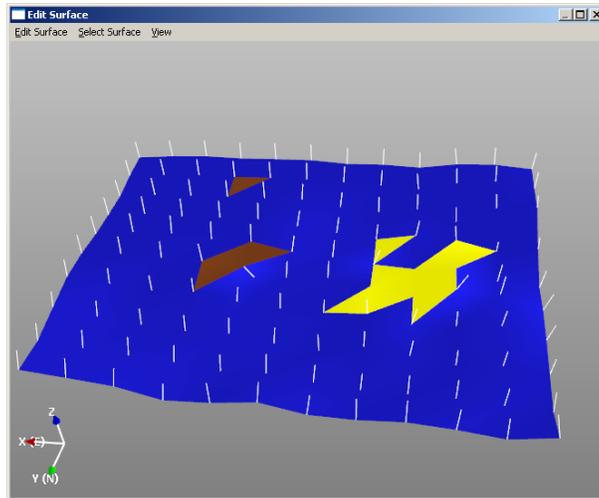


Bad normals can cause problems with clipping, decimation, fracture generation between surfaces, and rock wedge analysis, among other things. Thus it is important that your surface normals match up.

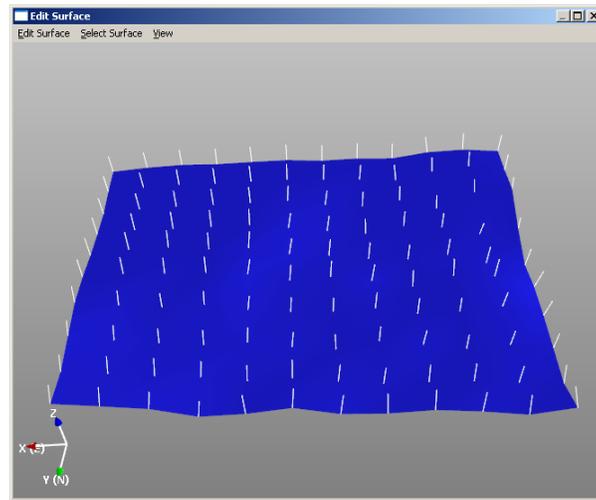
Normal vectors appear as rods sticking out perpendicular to a face, in a region, or a triangle vertex, on a surface. The normal points “into the rock” for the purposes of the analysis. On a surface, triangles whose normals point away from the viewer are shown in brown; triangles whose normals point towards the viewer are shown in blue. In order to flip a normal, you must first select the face/triangle whose normal you wish to change. You can select a single face/triangle by clicking that face in the image. Highlighted surface triangles appear yellow; selected region surfaces will appear another shade of the region color.



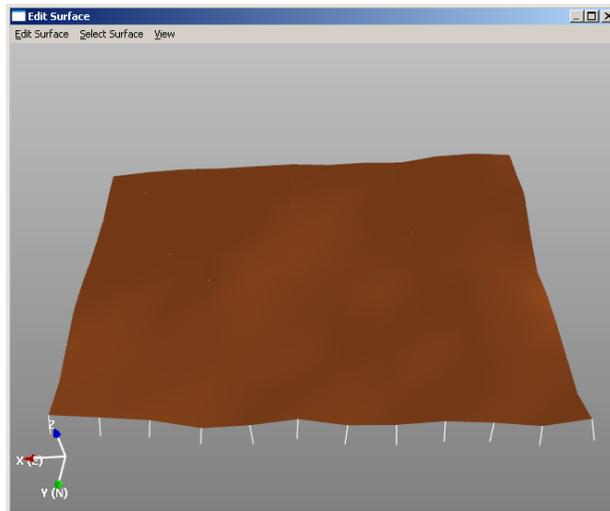
You may highlight multiple faces/triangles at once by holding the ctrl key while clicking, or by selecting *View > Select Rectangle* from the window's file menu and dragging the rectangle over the triangles you wish to highlight.



Once you have selected all the faces/triangles whose normals you wish to change, click Edit Surface > Reverse Selected Faces.



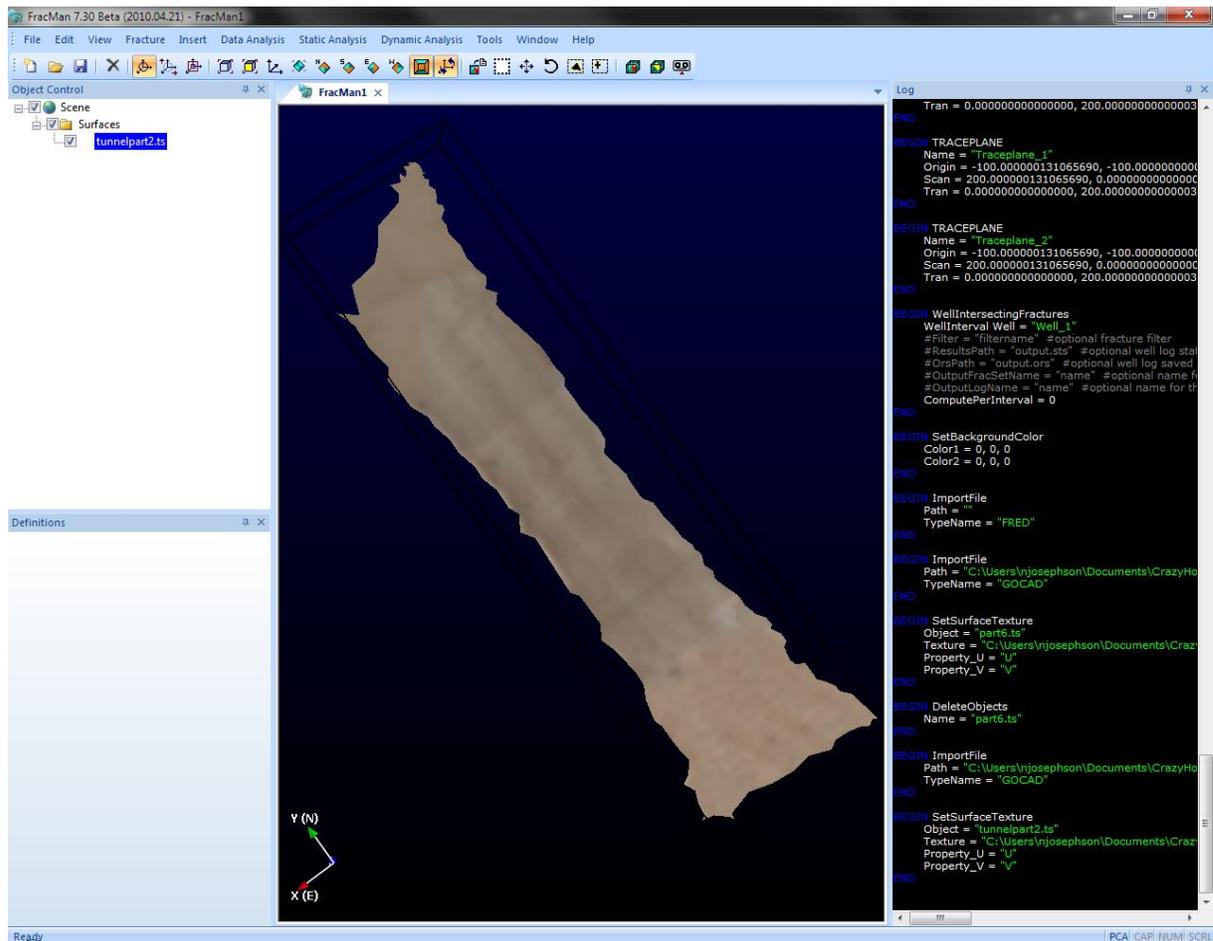
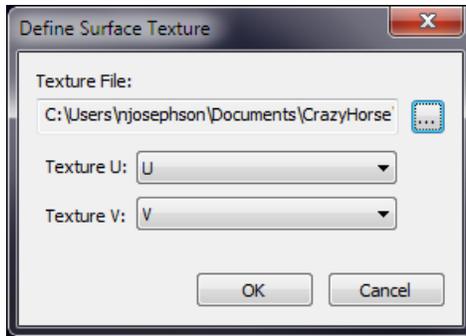
You can also flip a surface or invert a tunnel by selecting Edit Surface > Reverse All Normals.



When you have set up the normal vectors to your satisfaction, select **Edit Surface > Save Changes and Return**. If you do not wish to save your normal changes, select **Edit Surface > Revert Changes** and select **No** when FracMan asks you if you wish to save changes (closes dialog).

Apply Texture Map

With FracMan's new texture map feature you can display texture mapped surfaces from other applications. These consist of specially built surfaces that have U and V coordinates as vertex properties. Simply load the surface, then select "Apply Texture Map". Select the image file containing the texture and specify the U and V property names.

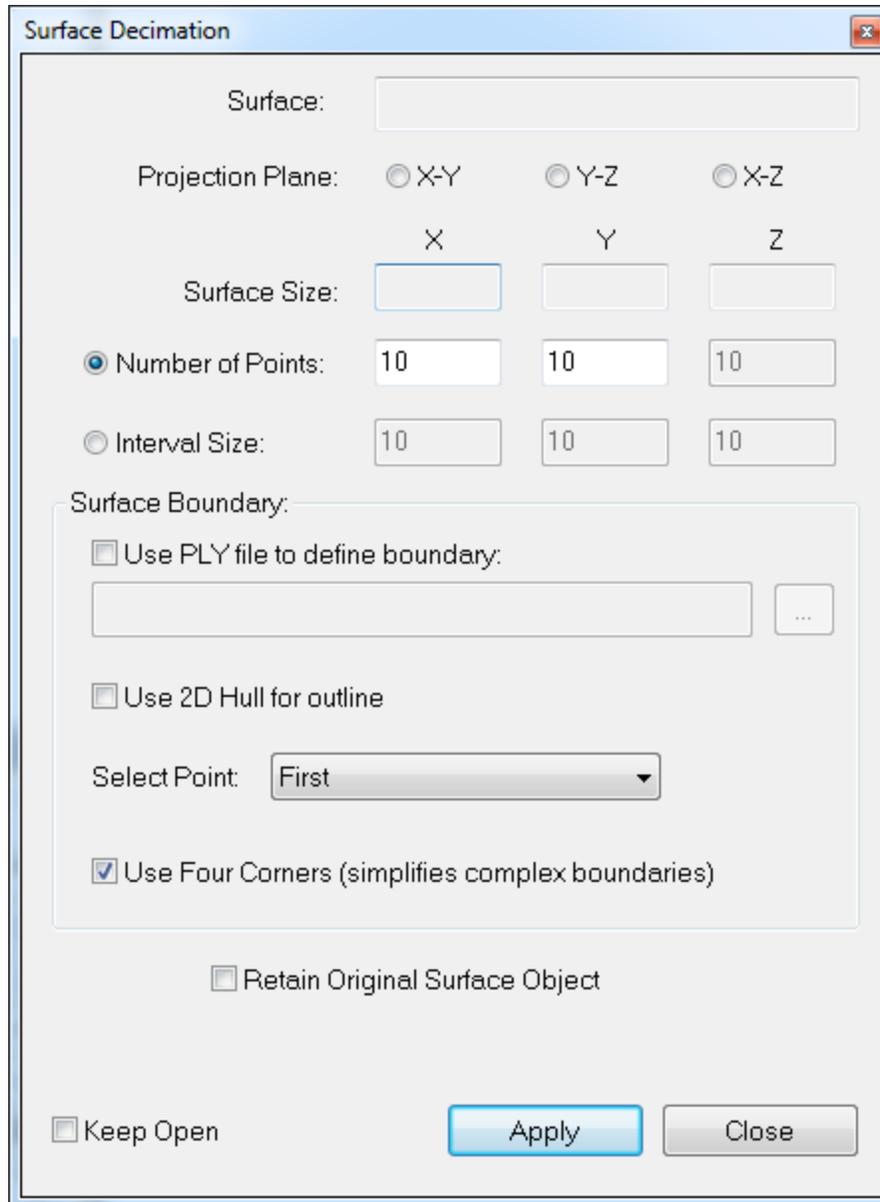


Decimate Surfaces

In many cases surfaces are too large to be useful for significant calculations. Decimating the surface will reduce the number of points in the surface while retaining the basic shape and features, making it small enough to be used for calculations. This function simplifies a surface by dividing into triangles. You must first select the surface you wish to decimate. Surfaces with extreme irregularities may not decimate well. It is important to carefully examine the output of surface decimation in order to verify that it retains necessary features.

First select the surface you wish to decimate. You may either right-click on the surface object name in the Objects window and hit “Decimate Surface” or select “Decimate Surfaces...” under the Edit tab on the file menu. You may select more than one surface to decimate at a time. A dialog box will open, giving you a set of options by which to define the decimation. The options are:

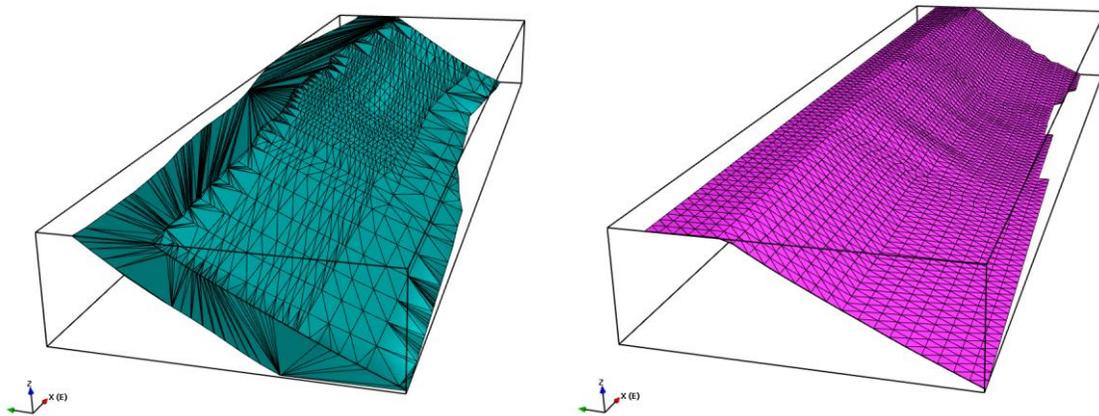
- **Projection Plane:** Specifies which directional plane FracMan uses for decimation - X-Y, Y-Z, or X-Z.
- **Surface Size:** Displays the X, Y, and Z dimensions of a hypothetical bounding box enclosing the surface.
- **Number of Points:** Allows the user to specify how many points to use for decimation in the given directions. The available directions are determined by which projection plane has been selected.
- **Interval Size:** Allows the user to specify what spacing between points the program uses for decimation in the given directions. Again, the available directions are determined by which projection plane the user selects.



- **Surface Boundary:** You can specify a EarthVision format (PLY) file which defines the outline of the surface. Decimation of complex surfaces can be dramatically improved by providing a PLY file. If a file is not specified, FracMan will compute the surface outline itself.
- **Use PLY file to define boundary:** A PLY file is an ASCII formatted text file that defines the vertices of a surface boundary. An example PLY file for a simple surface outline looks like:

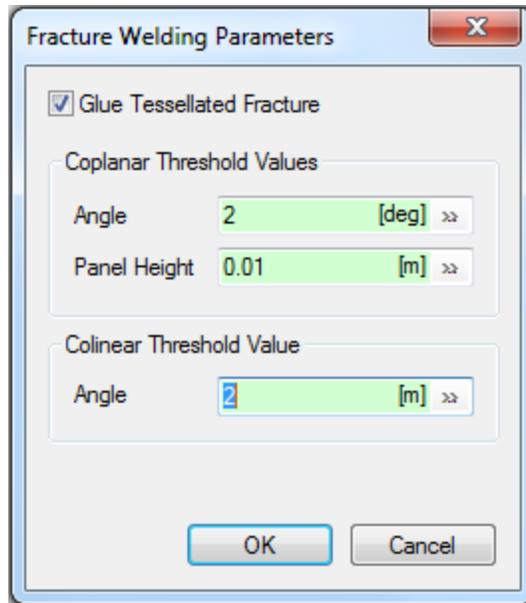
```
POLYGON
0 0
6000 0
6000 4000
0 4000
```

- **Use Four Corners:** Selecting this option instructs the decimation routine to pick four points within the boundary (either computed or from the PLY file) to define the new surface, rather than using the entire surface outline.
- **Retain Original Surface Object:** If you do not select this, the decimated surface will overwrite the original surface, replacing it, and the surface from which the decimation was extrapolated will be deleted.
- **Closed Surfaces:** Closed surfaces don't have outer edges that we use to bound the surface decimation. Checking the "Use 2D Hull for outline" option will compute the 2D convex hull of the object (projected in the selected direction) to enable surface decimation of these objects. The "Select Point" function allows you to use the first point, minimum distance, maximum distance or mean distance in resolving multiple decimation intersections.



Weld Fracture Faces

The Weld Fracture Faces operation can be used to clean up tessellated fractures or surfaces by combining nearly coplanar faces into singular elements. This reduces the number of elements necessary to specify the object in 3-space (which in turn increases performance and reduces the memory footprint) while still maintaining geometrical accuracy. In other words, this is a bit like reverse decimation.



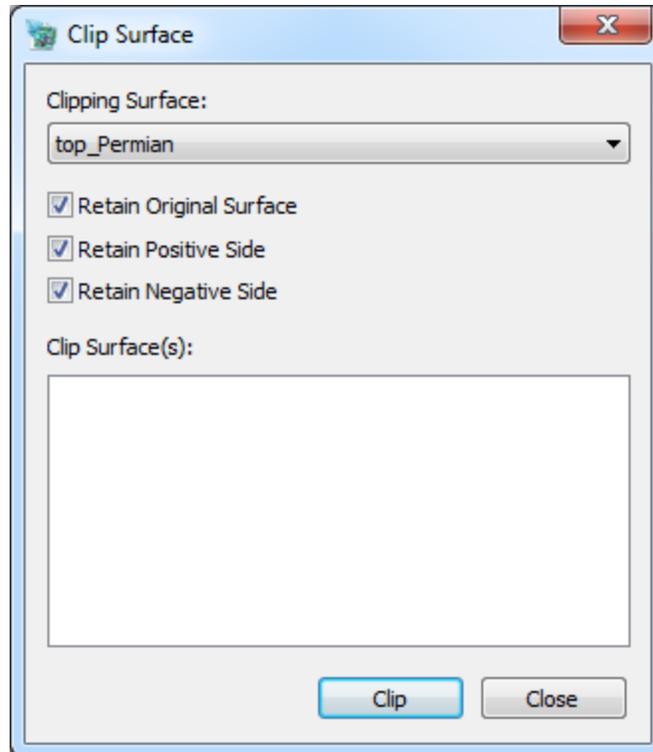
Coplanar Threshold Values are used to combine adjacent faces. Adjacent faces having angular differences less than the specified **Angle** and shifted center distances which are less than the **Panel Height** will be combined.

Colinear Threshold Value is used to remove nodes between adjacent edges having similar orientations. Edges sharing a node having a difference in trend less than the specified **Angle** will be combined and their adjacent faces will be combined.

Clip Surface

A handy feature not found in earlier versions of FracMan, surfaces can now be clipped to surfaces directly, rather than having to convert them to fractures first.

To clip a surface (or surfaces) select *Edit > Surface > Clip Surface* from the menu bar.

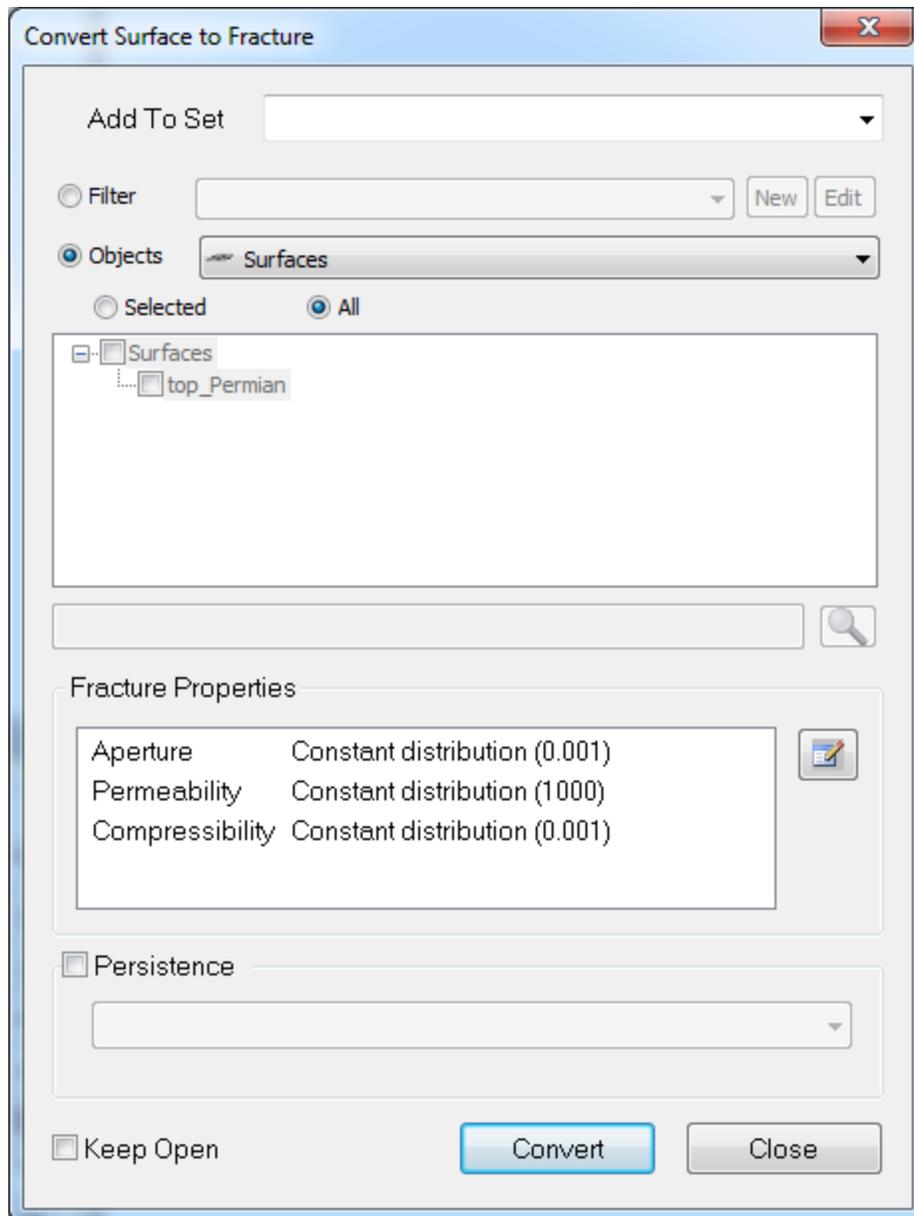


Select the surface that you wish to clip to from the dropdown menu under “Clipping Surface:”. Then check all the surfaces you wish to clip from the list beneath “Clip Surface(s)”. If you check “Retain Original Surface”, FracMan will keep the unclipped surface in the model along with the results of the clipping. “Retain Positive Side” and “Retain Negative Side” determine which half of the clipped surface you clip away and which half remains in the model. If both are checked the surface will be split into two halves. When you are ready to clip your surface, hit **OK**. The new surfaces will appear in the Objects window, with the same name as the original surface plus the suffix “_pos” or “_neg”, depending on which options you checked in your clipping definition.

Convert To Fracture

You can convert a surface object to a fracture. The object can be added to an existing fracture set or created as a new deterministic fracture set by leaving the drop-down box ‘Add to Set’ blank. Triangle/face properties of the surface become face properties of the fracture.

If the reserved fracture properties of Permeability, Compressibility, Aperture (PCA) or Transmissivity, Storativity and Aperture (TSA) were defined as vertex properties of the surface, they will be translated to face properties of the fracture. While the reserved properties PCA/TSA can be edited to have constant values, these will only be used in the new Fracture object if these property values are not already defined on the Surface object.



Compute Coordinate Frame

This will create a set of coordinate reference frame properties from surface face or grid cell orientations. It was designed to simplify the use of coordinate reference frames by allowing you to quickly compute the necessary properties based on the normal vectors of a controlling surface. You can then interpolate these property values to a grid in order to have the necessary coordinate reference frame. A simple smoothing factor is also included which averages the normals of neighboring triangles / cells.

This algorithm is designed along standard structural geology conventions and assumes a right-handed coordinate system. The algorithm will not return the correct results for a completely horizontal surface or fracture, because in this case the pole trend and/or fracture strike are not uniquely defined. The algorithm can be used to initialize the correct properties on the fracture

or surface, but the end user is expected to manually input correct values for the coordinate reference frame based on their needs.

Compute Surface Curvature

This will compute the curvature of the surface. Gaussian and mean curvature is calculated as well as the K1 and K2 principal curvature vectors.

Simplify Surface

Convert to Quad: this will replace the selected surface(s) with a planar rectangle.

Merge Nodes: this will merge vertices that are closer than the indicated distance, removing tiny or degenerate triangles and improving connectivity.

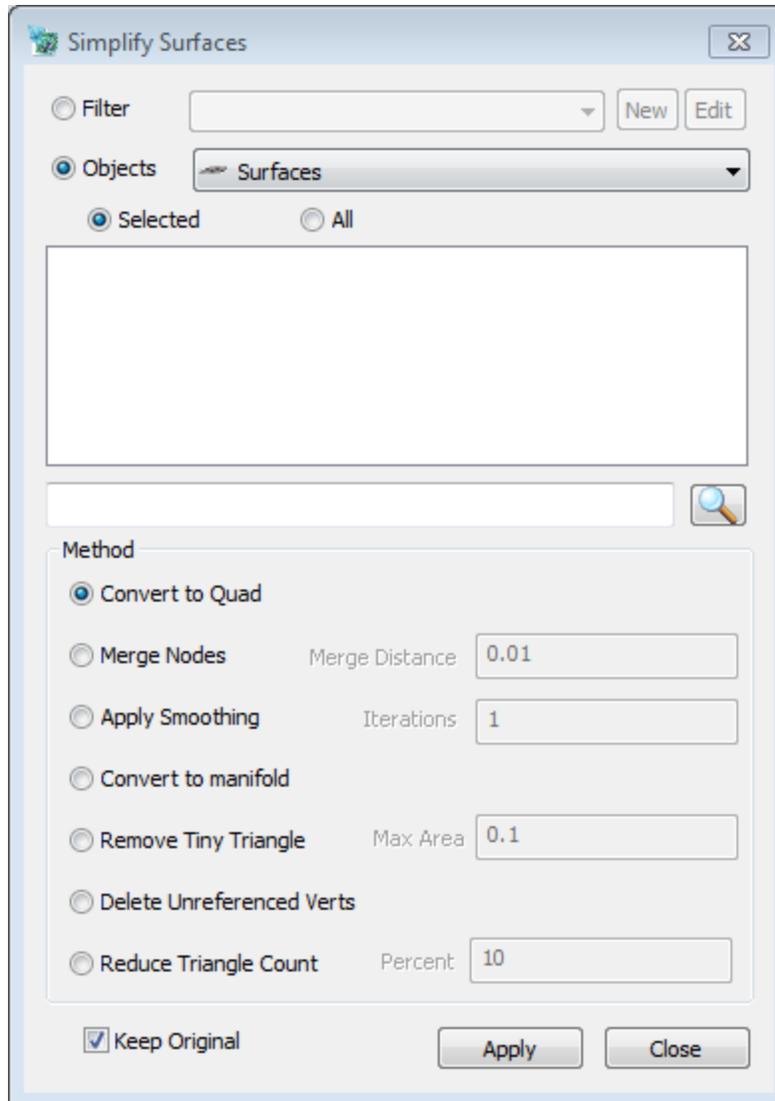
Apply Smoothing: this will adjust vertices to smooth out sharp points in the surface.

Convert to manifold: this will adjust the triangulation so that no triangle edges intersect a vertex. It's a good idea to run **Delete Unreferenced Verts** after this function.

Remove Tiny Triangle: this will remove small triangles and adjust the mesh to eliminate holes.

Delete Unreferenced Verts: Some file formats allow unused vertices – selecting this option will eliminate them from the object.

Reduce Triangle Count: This will simplify the surface by reducing the number of triangles in the surface.

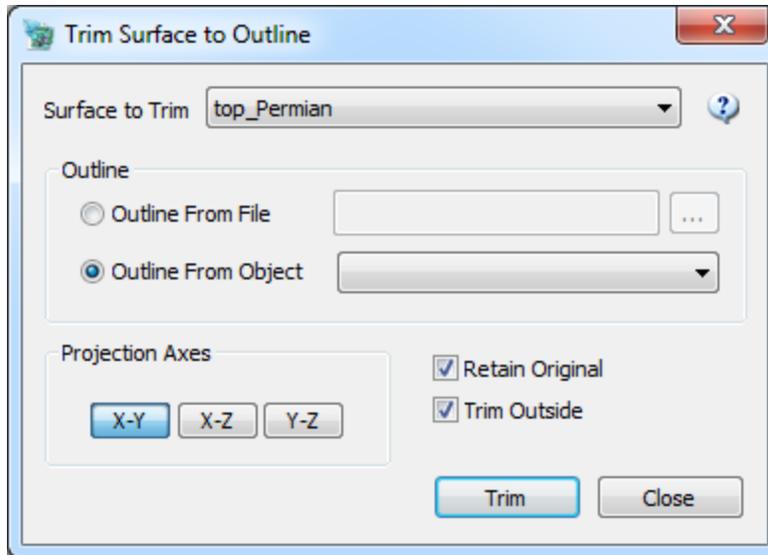


Merge Surfaces

In some cases multiple surfaces will in fact make up a single surface object – select the surfaces you want to merge together and a new object will be created that is the union of both originals.

Trim Surface Outline

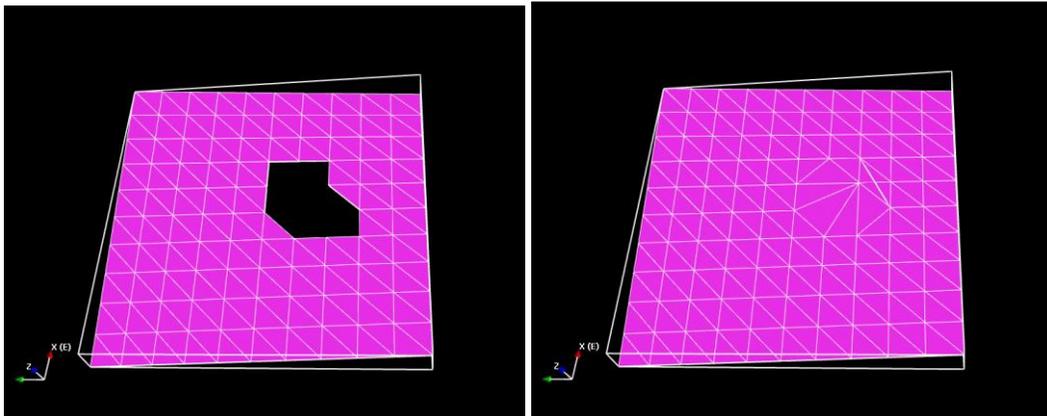
Trims triangles from a surface that are outside of a traceline defined in a file or stored as a tracemap.



Extract Surface From Selection

Triangulate Selection

Select the vertices of a surface around the hole and select “Triangulate Selection” from the Edit->Surface menu. Allows user to fill holes in a surface.



Convert Region To Surface

There is some functionality in FracMan that is only available to operate on surface objects. The **Convert Region To Surface** function creates a new single **Surface** object out of the outer surface of the region polyhedron. The original **Region** object is retained and a new surface object is added to the **Surfaces** folder in the Objects window.

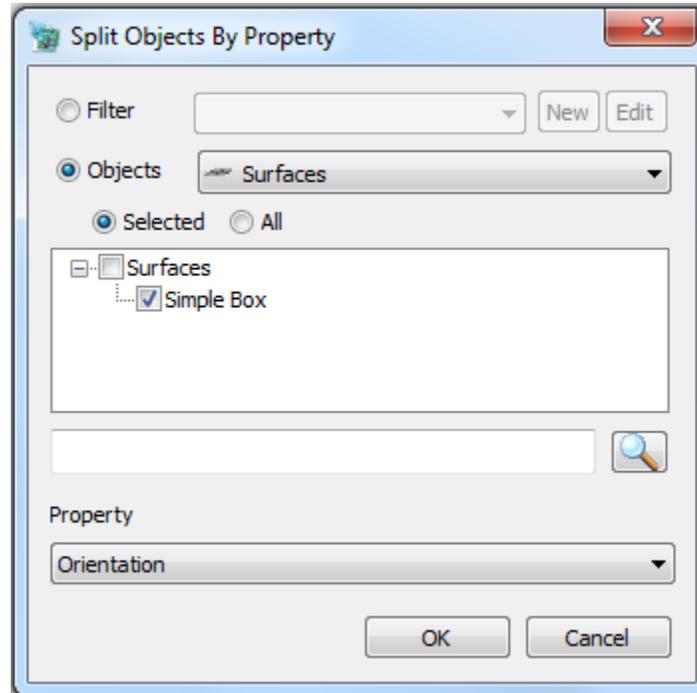
Convert Fracture To Surface

There is some functionality in FracMan that is only available to operate on surface objects. It is sometimes desirable to run this functionality on fractures as well. A simple way to achieve this is to convert the fracture to a surface using the **Edit -> Surface -> Convert Fracture To Surface** function. Select the fracture in the Objects window and then run the operation using the top

Edit menu. All the fracture properties are maintained as surface properties. The inverse functionality is Edit -> Surface -> [Convert To Fracture](#).

Split Object By Property Values

Surfaces, point data and well logs objects can be split into subsets by property if the property is stored as a string or integer. For example, if you have a surface that has a face property "Orientation" with 2 different integer values ("1" and "2"), the **Split Objects By Property Values** will create two surface subsets, each set containing only those surface faces having the same Orientation value. The **Split Objects By Property** dialog is reached from the Edit -> Surfaces menu.



Grids

Some fracture generation models use irregular grids to control fracture generation. These grids are either created inside FracMan (fold model) or imported (fault and stress field models). Together with predefined and user-defined relationship functions, the grid cell properties determine the resulting fracture properties such as orientation, intensity, size, etc.

Stratigraphic and lithological information is routinely stored in geocellular models. FracMan can generate discrete fracture models based on attributes in grids exported from a geocellular modeling package. Fracture parameters, such as fracture orientations, intensity, size, and fluid flow properties can be conditioned based upon the parameter values in each grid cell.

The reverse procedure allows you to sample fracture system properties and to return the results as grid values. The program will determine the location of each element and assign the element a value based on the underlying fracture network. Properties that can be sampled this way may

include fracture density, fracture system porosity, connected fracture volume. The results can then be read back into the geo-cellular modeling package for further analysis.

Discrete fracture models can be used to calculate equivalent continuum parameters (equivalent porosity, equivalent directional permeabilities, and matrix-fracture interaction factor). For example, FracMan can be used to import an ECLIPSE® reservoir simulator grid and calculate the respective continuum parameters for each grid cell.

See [Appendix A FracMan Supported File Formats](#) for a complete list of the file formats supported for grid objects.

Grid Creation

You can create a simple grid from the Insert menu.

	X	Y	Z
Origin	-100	-100	100

Trend/Plunge/Length
 X/Y/Z

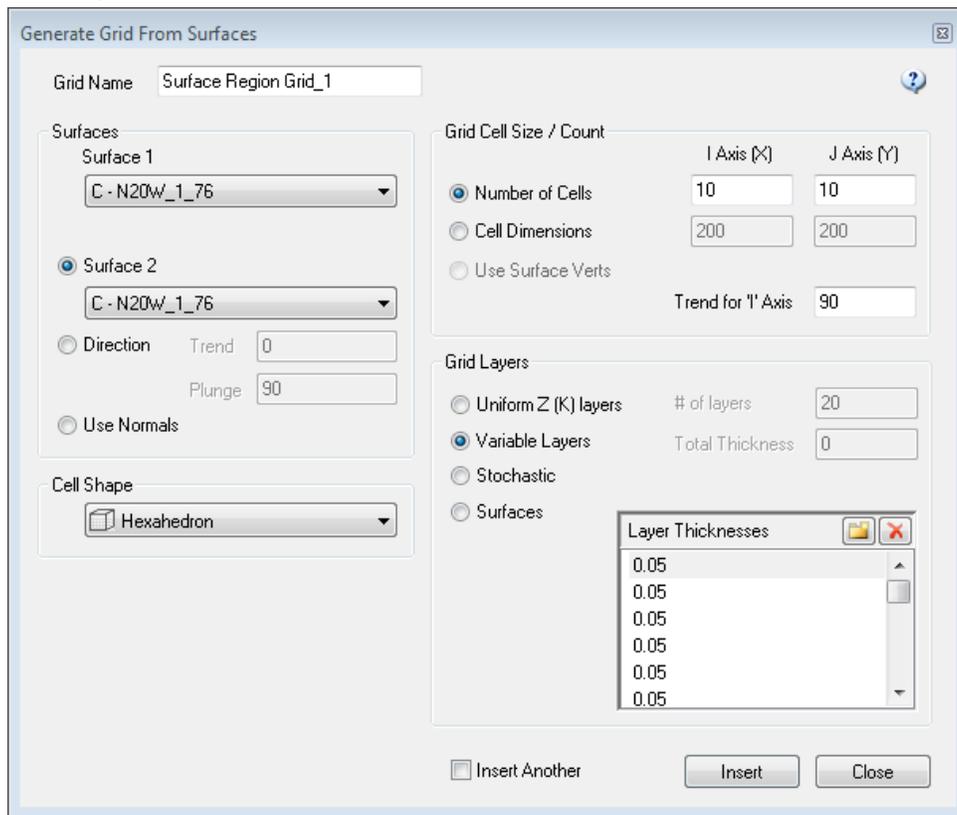
	Trend	Plunge	Length
Scan (I)	90	0	200
Transverse (J)	0	0	200
Height (K)	0	90	200

	Scan (I)	Transverse (J)	Height (K)
Number:	5	5	5

You can also create a grid defined by two surfaces using Insert->Grid from Surfaces. When defining a grid from one or more surface objects you can define the grid based on the surface normal, a specified direction or a second surface.

The user can also specify the shape of the grid cells as Hexahedron or Triangular Prism. The Grid Cell size can be determined from cell number, cell dimensions, or by using the surface normals. It is important to note that the Trend for the "I" Axis will affect the numbers of cells/cell size.

You can define uniform z layers, specify their thickness based on a fraction of the total width, specify their relative thicknesses based on a random distribution or by selecting the layer surfaces directly.

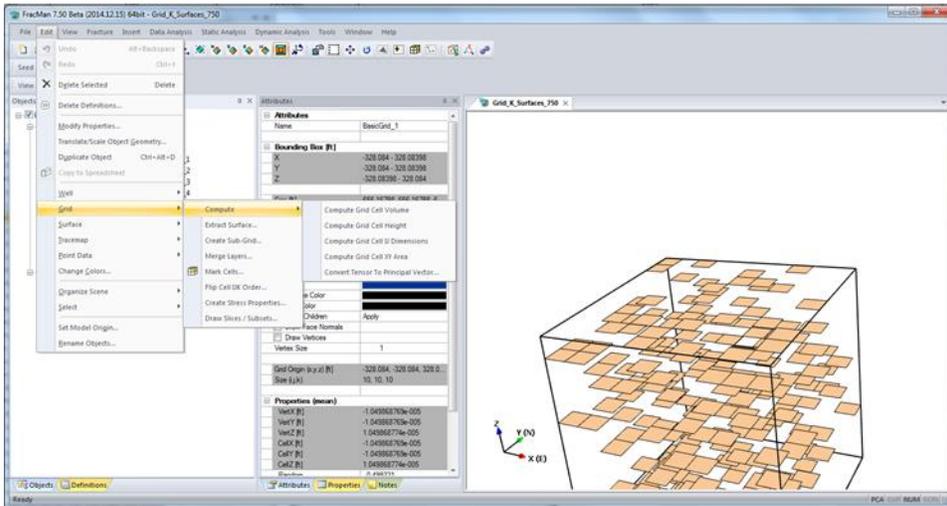
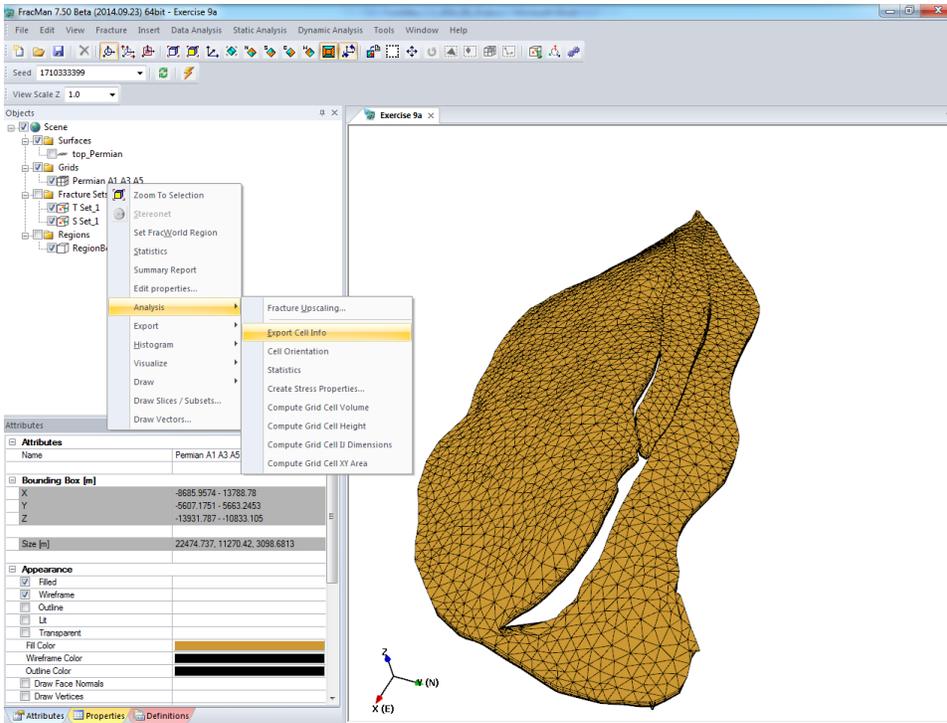


Compute

A number of grid calculations are available including:

- Grid Compute Cell Height
- Grid Compute Cell Volume
- Grid Compute Cell IJ size
- Grid Compute XY Area
- Convert Tensor to Principal Vector

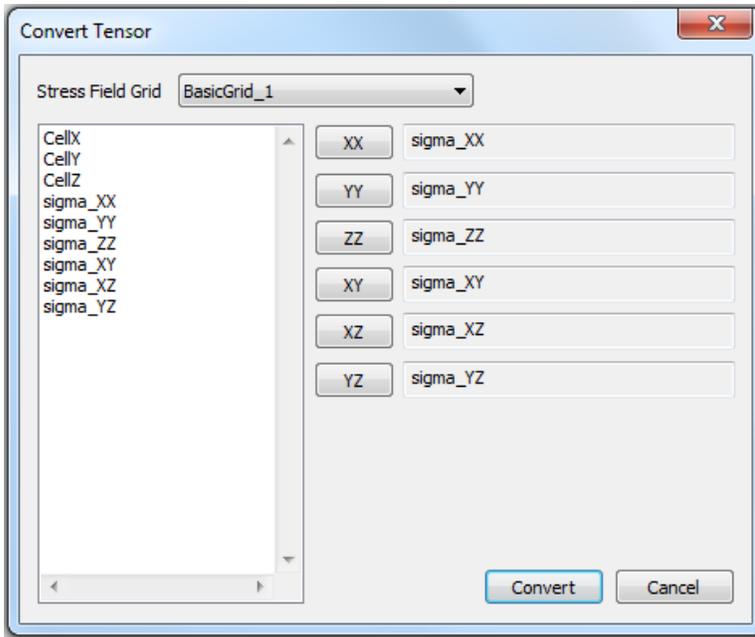
These can be accessed either from the Analysis functionality available after right-clicking on the grid object, from the Edit -> Grid -> Compute menu option, or from the Static Analysis -> Grid menu option.



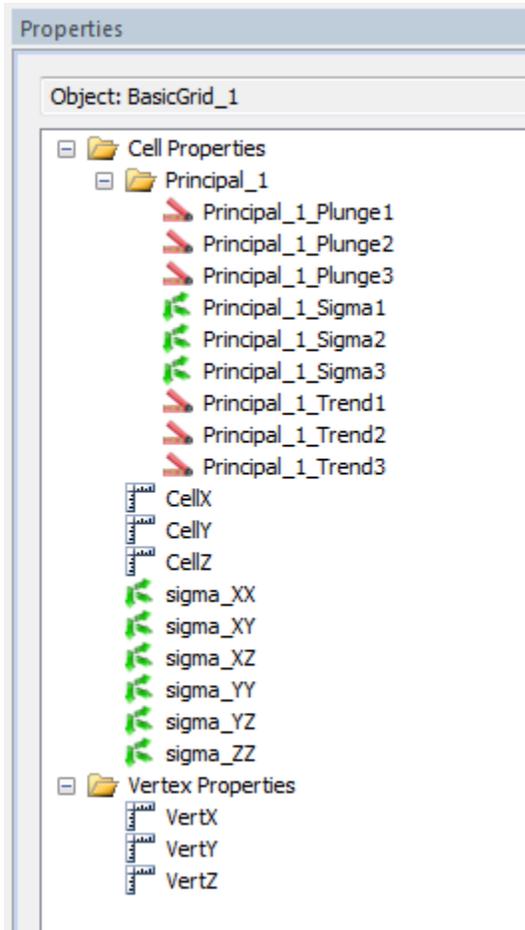
Convert Tensor to Principal Vector

This function allows the user to convert six tensor values into nine principal vector components. Note that compressive stress is positive and tensile is negative.

Assign Grid properties for tensor components:

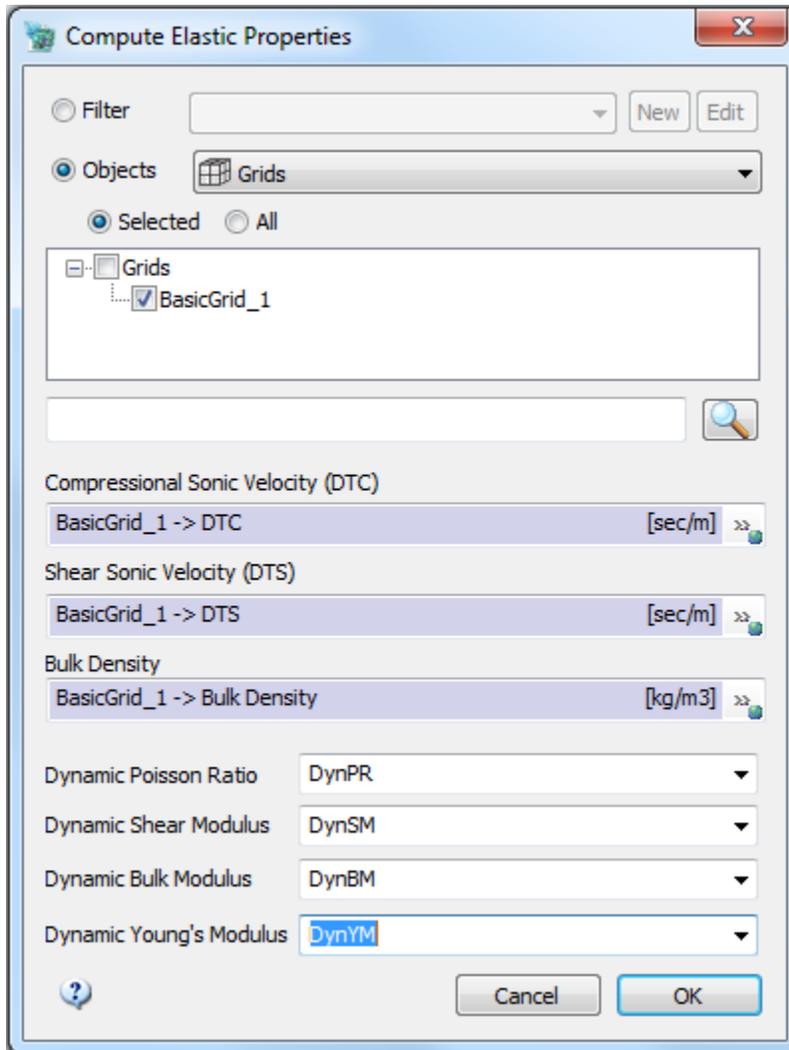


Click **Convert** to obtain new grid properties for principal stress values:



Compute Material Properties from Sonic Log Data

In order to simplify the process of computing material properties from well log sonic velocities we've automated the process of setting up the material properties using the **Compute Elastic Properties** dialog.



Select the object(s) that you want to compute the properties for (grids / well logs) containing properties representing DTC and DTS (both have units of time/length) as well as DENS (units mass/vol).

- DTC = Compressional Sonic Velocity
- DTS = Shear Sonic Velocity
- DENS = Bulk Density

Assign these three properties of the well log to the appropriate text fields in the dialog using the buttons on the right side of the text boxes. The new calculated properties will be saved into the same object as new properties if a unique name is typed into the selection box, or saved into an existing property field selected from the dropdown list. Below are their formulas:

Dynamic Poisson Ratio from Dipole Sonic

- $PR_{dyn} = (0.5 * R^2 - 1) / (R^2 - 1)$ where $R = DTS / DTC$
- Check limits of $0.00 < PR_{dyn} < 0.50$
- Units are dimensionless

Dynamic Shear Modulus from Dipole Sonic

- $G_{dyn} = DENS / (DTS^2)$
- Units of pressure

Dynamic Bulk Modulus from Dipole Sonic

- $K_{dyn} = KS5 * DENS * (1 / (DTC^2) - 4/3 * (1 / (DTS^2)))$
- Units of pressure

Dynamic Young's Modulus from Dipole Sonic

- $E_{dyn} = 2 * G_{dyn} * (1 + PR_{dyn})$
- Units of pressure

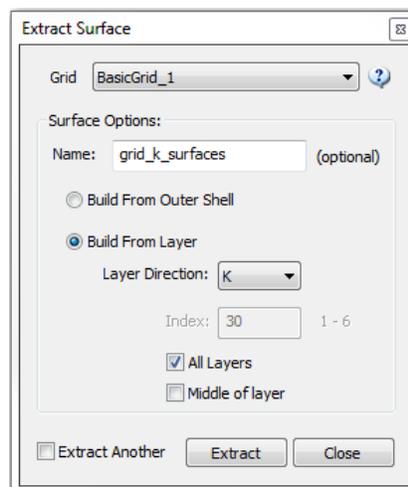
Click **OK** to compute the properties.

Compute Grid Coordinate Frame

See Manual section for Surface [Compute Coordinate Frame](#).

Extract Surface

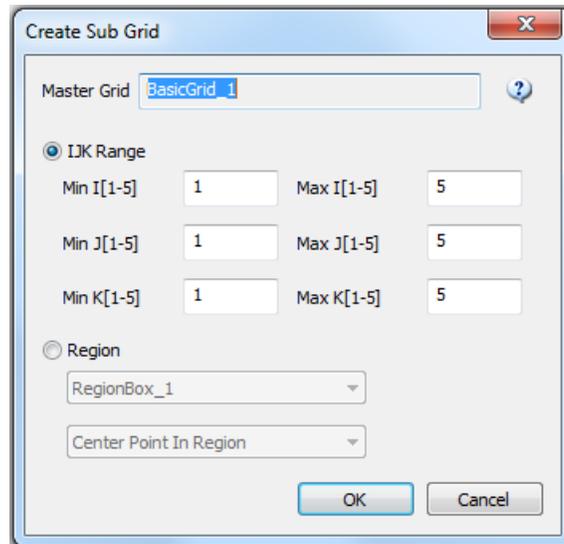
You can extract surfaces from regular grids using the Edit -> Grid -> Extract Surface function. The outer shell of the grid is available as well as any layer or range of layers in the I, J, or K directions.



Create Sub-Grid

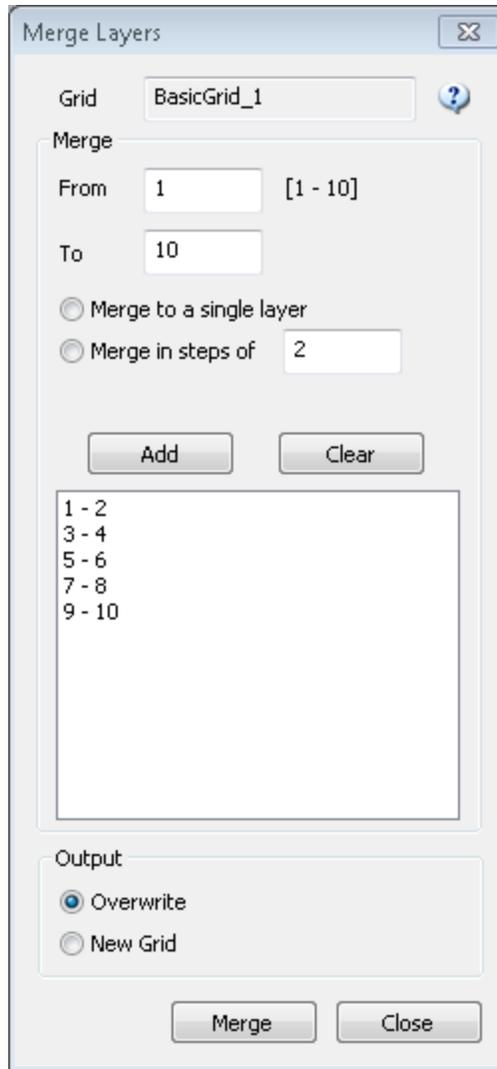
Build a new grid out of a subset of an existing grid. The new grid will have all the same properties of the original grid. You can either define an IJK range or select a region object.

Note: if you create a subgrid via a region object, the resulting subgrid will be a non-regular (no IJK axes) grid.



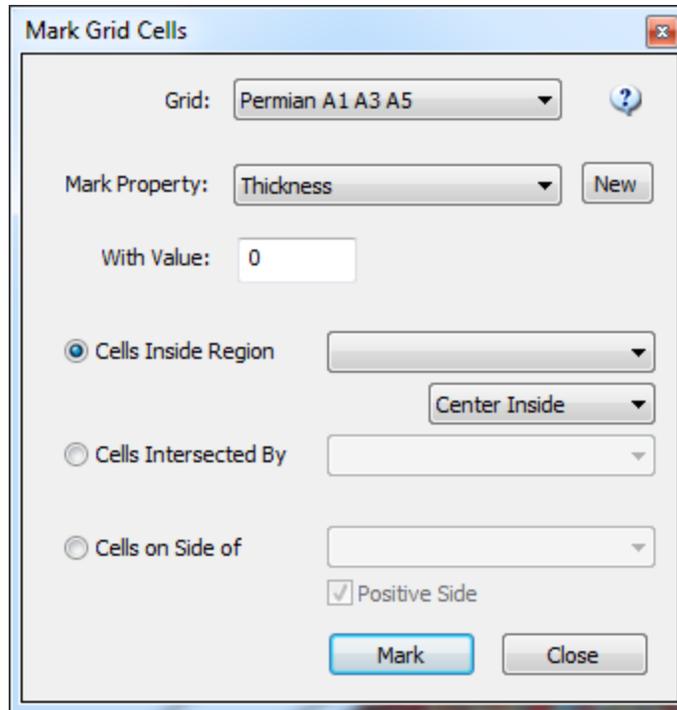
Merge Layers

This allows you to simplify your grid object by combining multiple K layers together. Property values will be averaged in the combined cells.



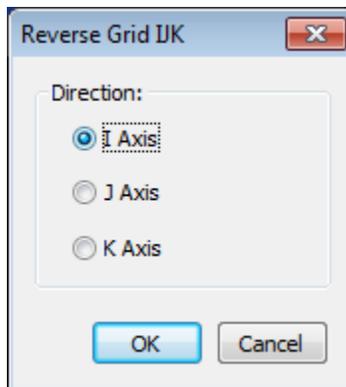
Mark Cells

You can quickly mark grid cells that are contained inside a region or intersected by a surface or well. Marked cells are indicated by setting a chosen property to a value you specify.



Flip Cell IJK Order

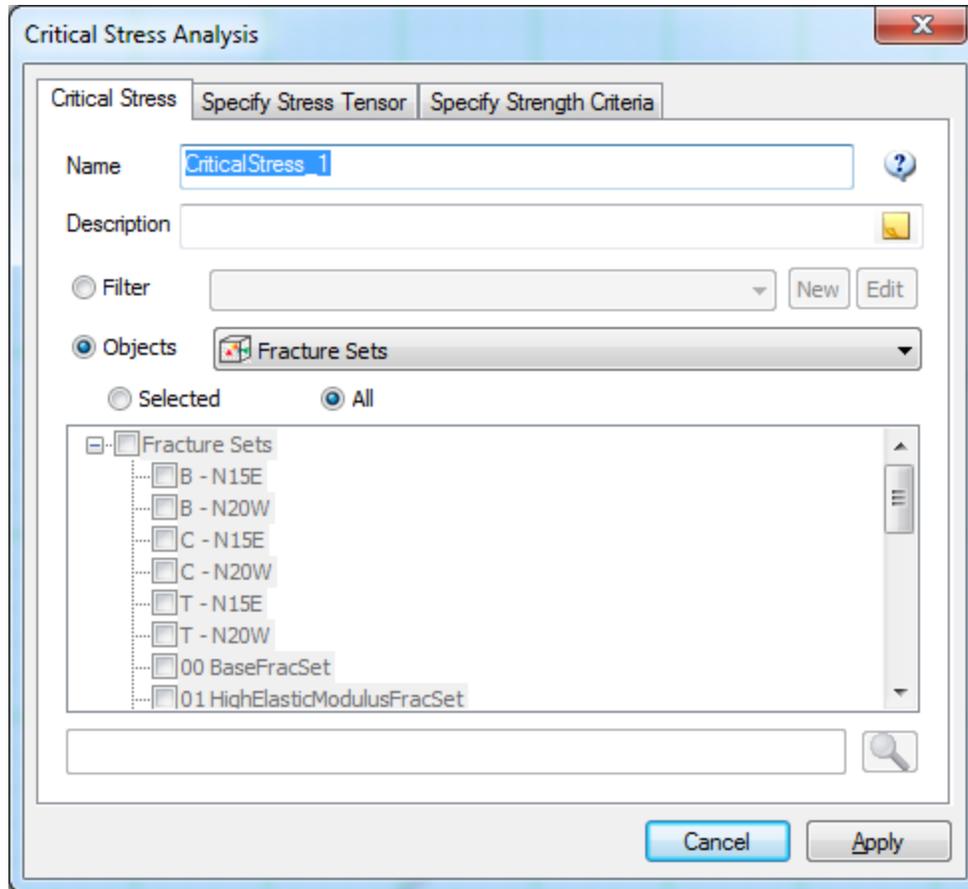
This is useful for correcting compatibility issues with Petrel or other simulators that have imposed limitations on the directions of the I, J or K axes.



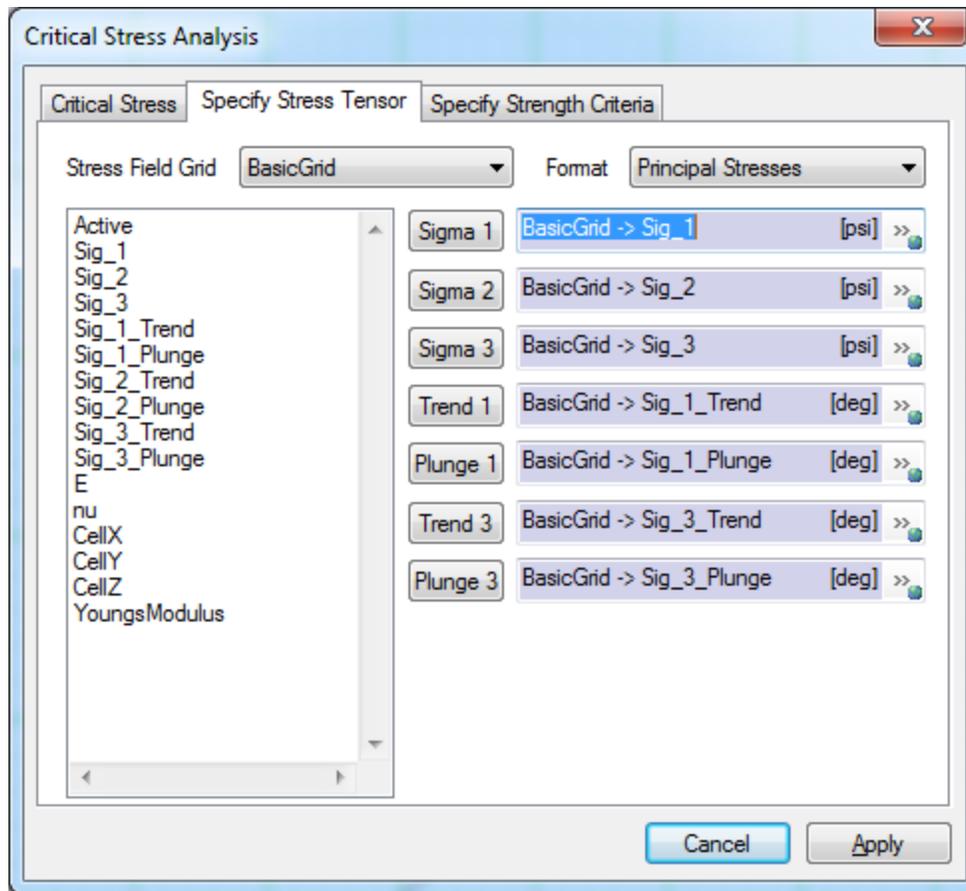
Define Stress Properties/Critical Stress

This feature simplifies the definition of the stress field properties used by the Critical Stress Analysis and the Stress Propagation fracture generation. You can select existing property names for each of the principal values or enter your own as well as specifying a constant property value.

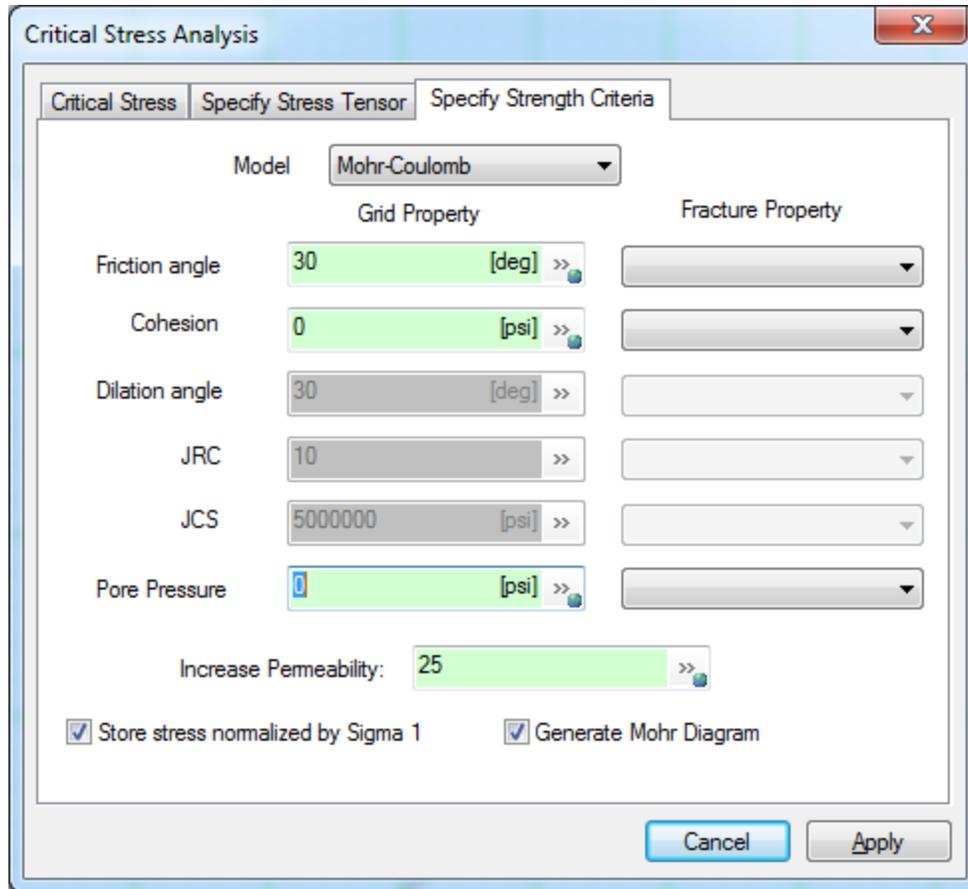
The first tab of the critical stress definition asks the user to specify the name and which fractures to analyze for critical stress



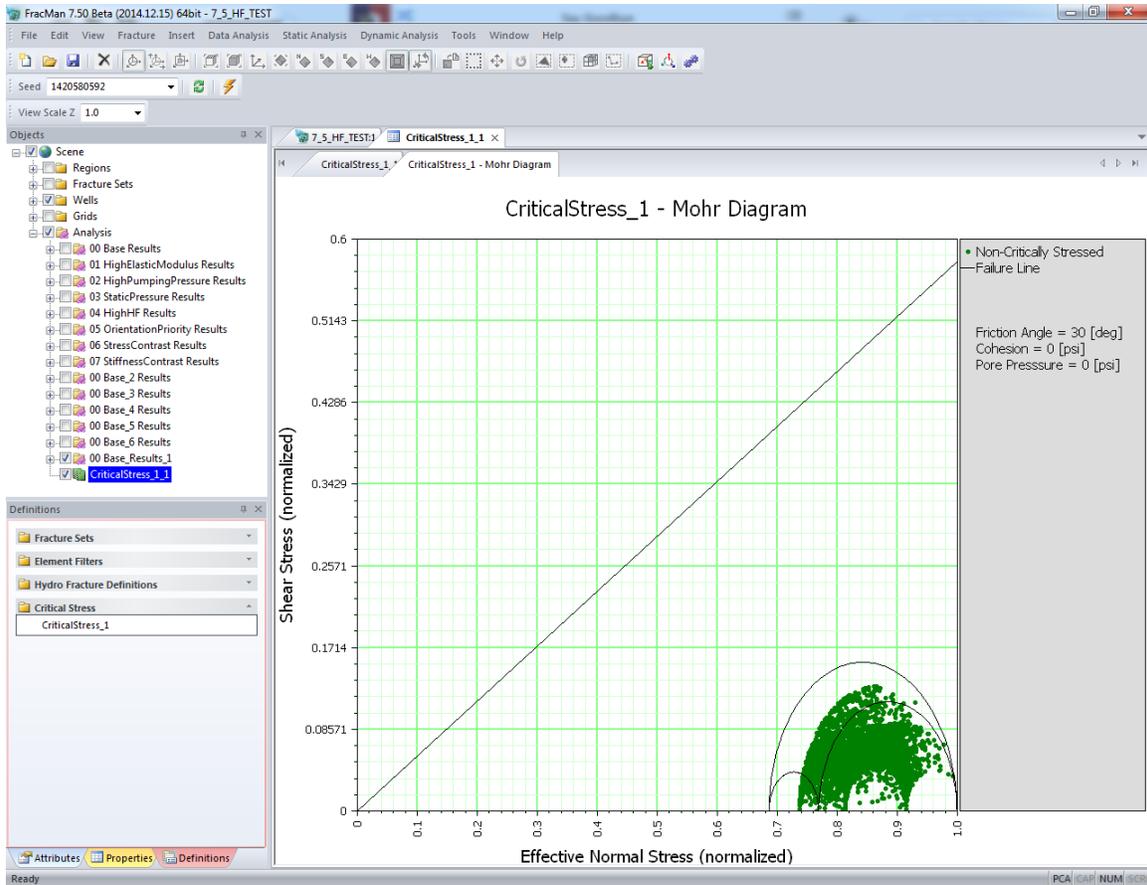
To specify the stress tensors, the user must have an existing grid populated with stress magnitudes and stress orientations (Trend and Plunge) for all 3 stress tensors (Sigma 1, Sigma 2, and Sigma 3).



On the last tab the user specifies the strength criteria (i.e. Pore Pressure, etc.) and can give a constant value or use a current grid property.



Once the user has run the critical stress definition a statistics file along with a Mohr Diagram will appear, which report on the number of critically stressed fractures in any given set along with the total number of critically stressed fractures. The Mohr Diagram (a new feature in FracMan 7.5 will show the current 3D stress regime and where the fractures are in relation to being critically stressed.



Create Brittleness Properties

On a grid object we may want to calculate a number of Brittleness Indexes. Brittleness is the measurement of stored energy before failure and can be a function of

- Rock strength
- Lithology
- Texture
- Effective stress
- Temperature
- Fluid type
- Diagenesis
- TOC

However, most currently in use are a function of Young's Modulus, Poisson Ratio and/or material strength. These can provide a good indicator of the level of natural fracturing or susceptibility to fracturing. FracMan currently supports five of the many different brittleness indexes.

Type I: Strength

$$BI_{\text{Strength}} = UCS / UTS$$

Type II: Lane

Based on Lamé's constant, L (aka λ) is a measure of a rock's brittleness, which is a function of both Young's Modulus and Poisson's Ratio.

$$L = BI_{\text{Lame}} = YM * PR / ((1 + PR) * (1 - 2 * PR))$$

Type III: Mullin

Mullin proposed a practical model that is widely used as a brittleness indicator (SPE 115258, 2007). *If this index is selected, the Young's Modulus input must be in units of Mpsi.*

$$Y_{\text{Mullin}} = ((YM - 1) / (8 - 1)) * 100$$

$$PR_{\text{Mullin}} = ((PR - 0.40) / (0.15 - 0.40)) * 100$$

$$BI_{\text{Mullin}} = (Y_{\text{Mullin}} + PR_{\text{Mullin}}) / 2$$

Type IV: Goodway

Goodway (2004) calculated a brittleness indicator based on the Lamé constants of the rock, both of which are functions of Young's Modulus and Poisson's Ratio.

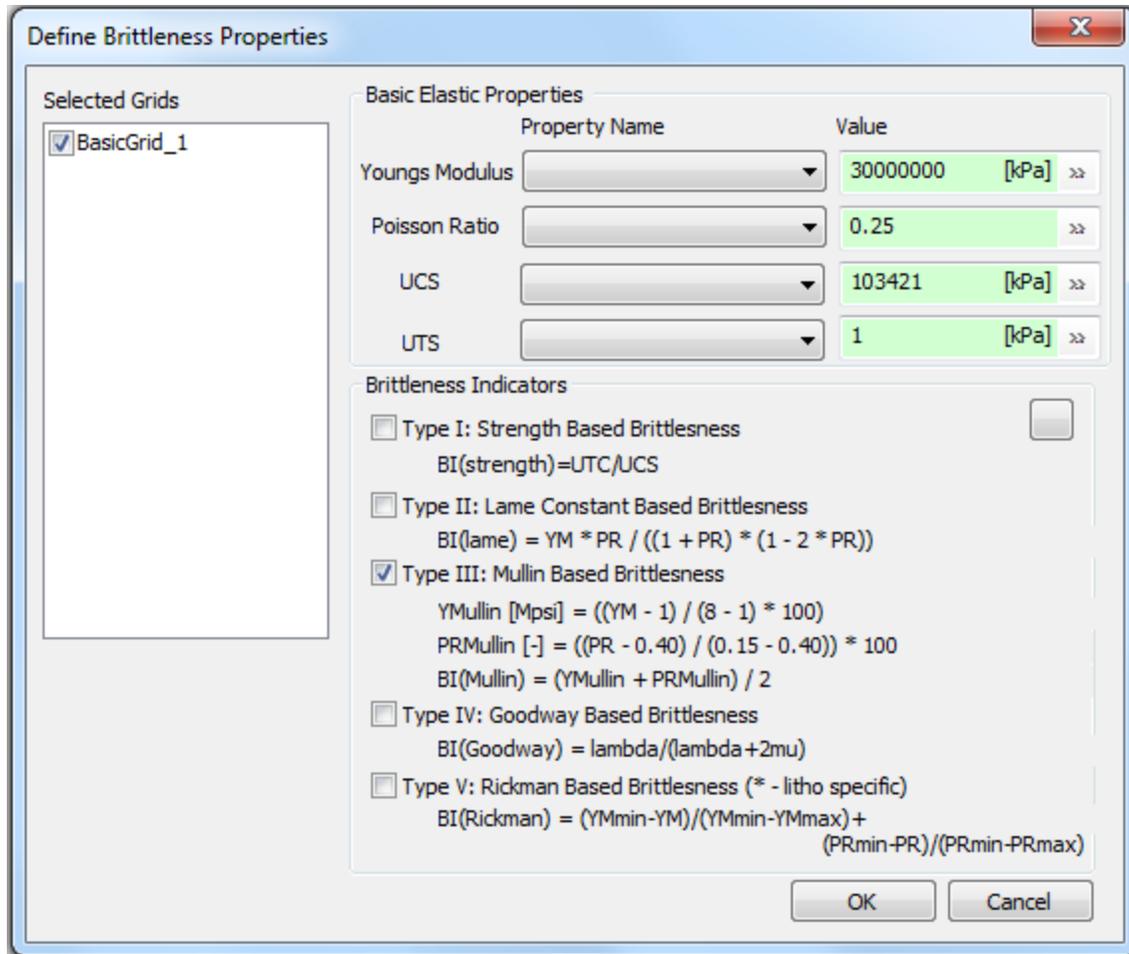
$$BI_{\text{Goodway}} = \lambda / (\lambda + 2 * \mu)$$

Type V: Rickman

This index should only be used on a single lithology at a time and not wholesale on the grid (SPE 115258, 2008).

$$BI_{\text{Rickman}} = (YM_{\text{min}} - YM) / (YM_{\text{min}} - YM_{\text{max}}) + (PR_{\text{min}} + PR) / (PR_{\text{min}} + PR_{\text{max}})$$

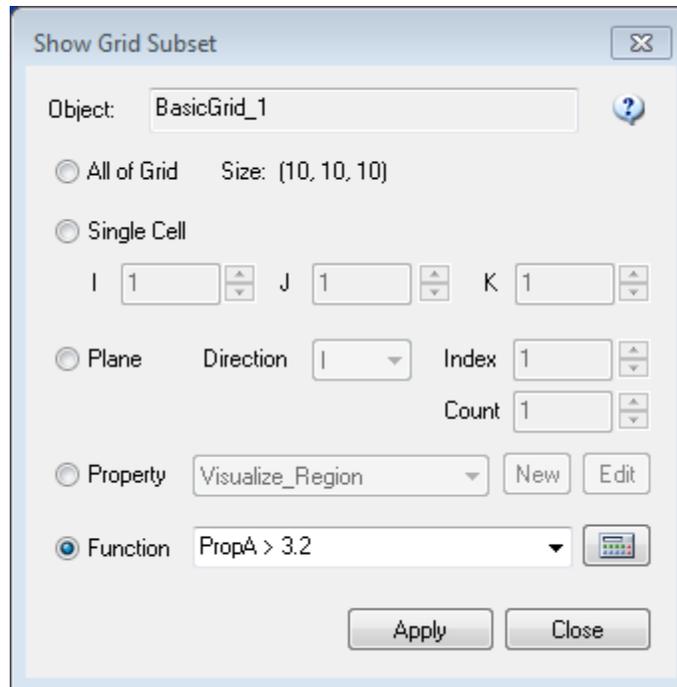
These five indexes may be calculated for a grid using the **Edit -> Grid -> Create Brittleness Properties** (also found at **Edit -> Grid -> Compute -> Create Brittleness Properties**) and also available by right-clicking on a grid object and selecting **Compute -> Create Brittleness Properties**. Selecting this menu option opens the **Define Brittleness Properties** dialog.



Here the user selects one or more grids where the output properties will be added and from which, optionally, some or all of the input properties can be used. Input parameters can be grid properties or constants. If more than one grid is selected, property name for the input variables must be the same in each grid. One or more brittleness indexes can be selected for calculation. The output grid properties are all prefixed 'Brittleness_'.

Draw Slices / Subsets

This function is available either from the Edit -> Grid -> Draw Slices / Subsets menu option or by right-clicking on a Grid object. The section of grid displayed as a result of this function is the only active part of the grid, meaning that applying other functions that call on the grid will only affect the part of the grid that is shown. For example, a geocellular fracture set generated on a grid that has only one K layer shown, will only generate fractures within that one particular K layer. Also, extracting a surface from this grid will only extract from the part of the grid that is displayed.



Draw Vectors

The Draw Vectors visualization shows grid properties as vectors. This can be combined with a color visualization based on another property. Select the grid and right-click on **Draw Vectors**. The **Visualize Arrows** dialog will open.

Visible: check to make arrows visible (after clicking on the Apply button).

Turn off Filled Attribute: if the Filled attribute on the grid object had been selected, checking this will de-select that for the entire grid.

Draw Arrowheads: un-check to have vectors not show an arrowhead.

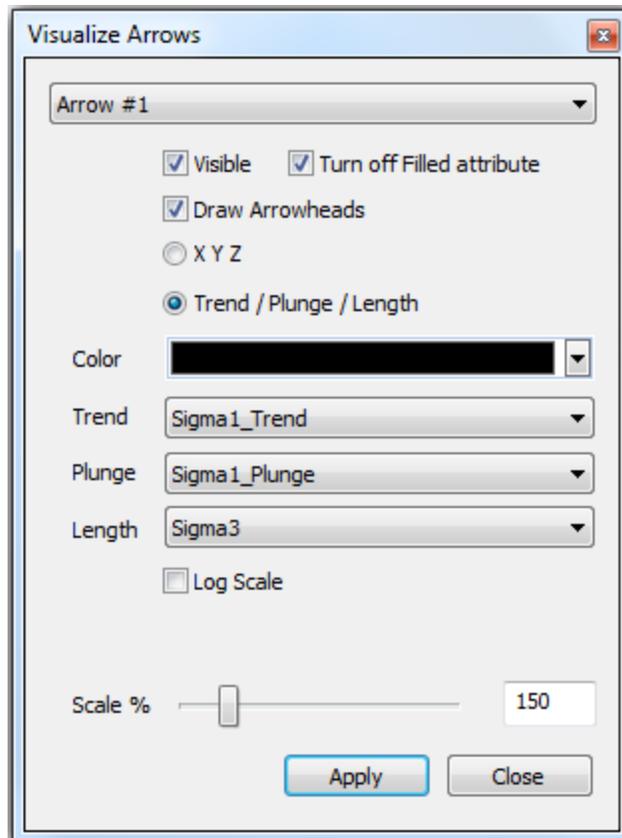
XYZ: if selected, you may select which property controls each directional component (ie. i, j, k, respectively) of the arrow in three dimensions.

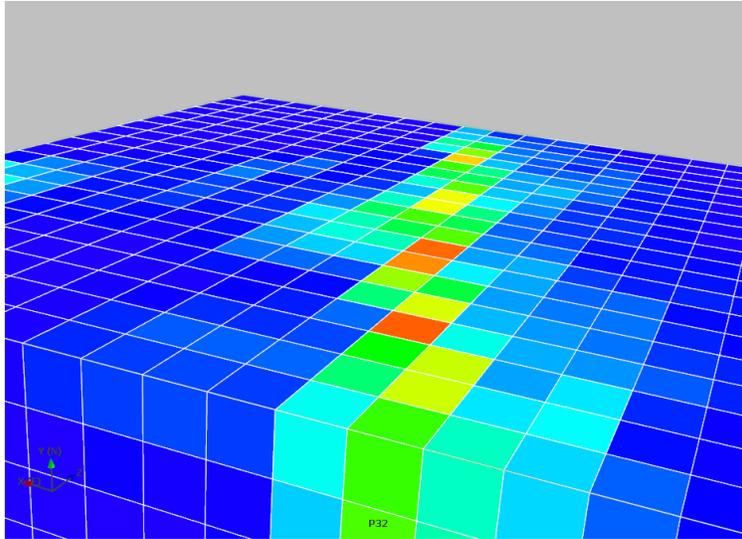
Trend / Plunge / Length: if selected you may select which properties of the grid control the trend, plunge, and length of the vector arrows. Selecting a **Length** control property is optional.

Color: Will only be used if another property has not already been selected for Visualization before applying the **Draw Vectors** functionality. Select on the grid **Visualize -> None** to reset if you want the vector color to be set by the **Visualize Arrows** dialog.

Log Scale: Only affects the **Length** of the vector.

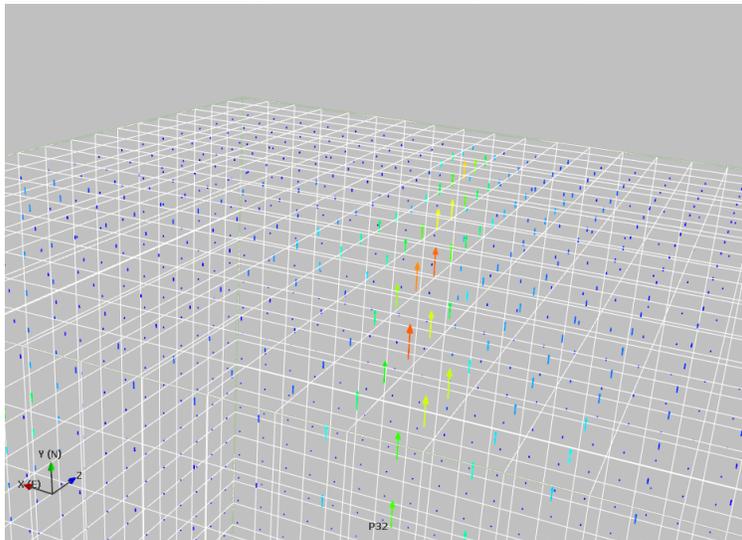
Scale %: Only affects the **Length** of the vector. Can use the sliding scale or enter a percentage directly.





Different visualizations of the same grid property.

Top: color visualization determined by grid cell P32 only.



Bottom: color and arrows visualizations, with color and magnitude determined by P32, trend and plunge constant.

Wells

Wells can be used in FracMan for geometric analysis (e.g., calculating the fracture density for different drilling directions) or as source/sink terms in well test simulations. Wells are represented as piecewise-linear trajectories. Borehole and perforation radii are represented by properties of the well trajectory segments.

To Create a Well Within FracMan

On the Insert menu, select Insert Well. A dialog will pop up showing some options for creating a Vertical, Inclined, or Deviated Well. Measured Depth, Z, or Segment Length can be specified.

Insert Well

Name: Edit Well

Collar: X: Y: Z: m

Type:
 Vertical
 Inclined
 Deviated

Well Depth:
 Measured Depth
 Z (TVDS)
 Segment Length

Upper Segment:
Deviation Trend:
Plunge:
Segment Len: m

Lower Segment:
Deviation Trend:
Plunge:
Segment Len: m

Add Interval Set
 Full Length Edit Intervals

Define Radius
Borehole Radius: m

Insert Another After

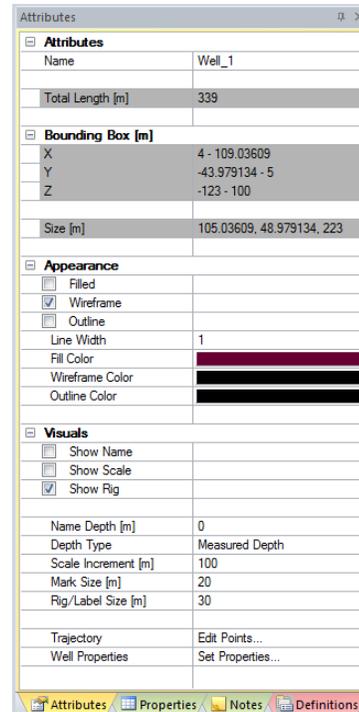
The Collar indicates the top of the well, however that is defined by the user, with measured depth calculated from this point. Changing the Type of well also changes the parameters required for its definition. Checking the “Edit Well” box opens the Well Definition dialog immediately after well creation where the well trajectory and segment properties can be defined.

To modify the properties of the well, left-click on its label in the Object Control Bar, and then select the Attribute bar.

By default wells are displayed in wireframe mode with a line width of 1. In wireframe mode the line width is a screen (pixel) size of the line representing the well trajectory. If “Filled” is selected, the well is drawn as a cylinder with a radius of “Line Width” in world units.

Name Depth determines the measured depth where the well name is displayed.

Show Scale determines whether or not to display the well depth labels along the well trajectory and Scale Increment determines the spacing between depth labels.



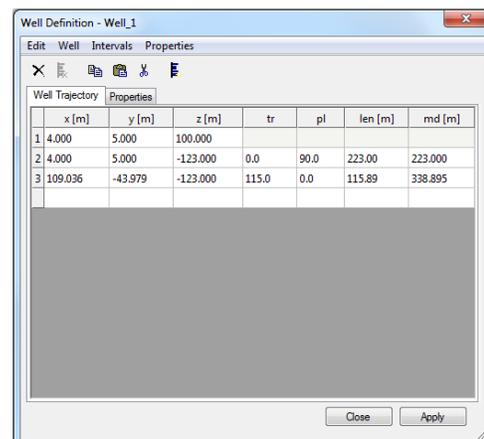
To Load A Well From A File

On the File menu, select Import | Well Data. Select the file type corresponding with the well file you are about to load (see [Appendix A FracMan Supported File Formats](#) for details on the types of files supported). Select one or more files and select “Open.” All wells defined in the files will be loaded and displayed in the viewing window.

Editing the Well Trajectory

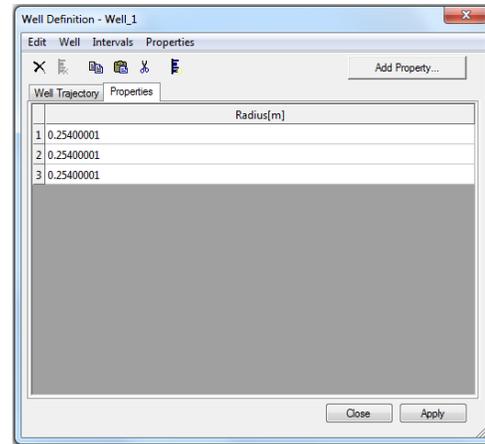
When the properties of the well are displayed in the Attributes window, left-click the Edit Points... button to invoke the Well Definition dialog or right-click on the well in the objects bar and select Edit Well.

On the Well Trajectory tab, enter the x, y and z coordinates of the points of the trajectory. Data can be pasted into the grid by pressing Ctrl+v.



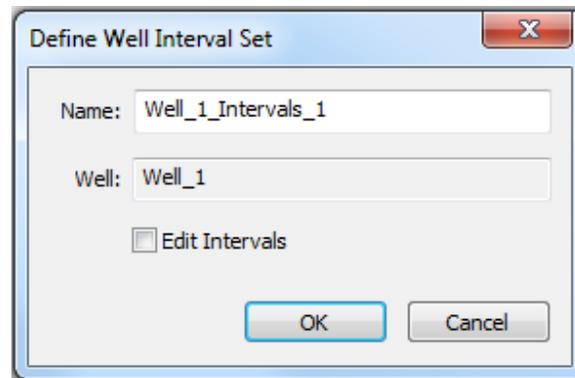
On the “Properties” tab you will see a column for each well property that has been defined. Initially only “Radius” will be displayed.

You can edit the values of the property for each segment of the well’s trajectory. You can add additional properties by pressing the “Add Property” button or by selecting “Add Property” from the “Properties” menu. The full capabilities of FracMan’s property editing system are available by selecting “Edit” from the “Properties” menu.



Defining Well Intervals

Wells are generally divided into sub regions called intervals. These intervals are generally defined by measured depth. FracMan groups these intervals into “Well Interval Set” objects. Each well can have multiple interval sets and each interval set can have multiple well intervals. You can add a new interval set object to a well by right clicking on the well and selecting “Add Interval Set”.



Enter a name for the interval set and press OK. If you check the “Edit Intervals” box you will be taken directly to a dialog where you can define the intervals.

Each interval set for the well will appear as a tab in the Well Definition Dialog or you can edit them individually by right clicking the interval set in the objects bar and selecting “Edit Intervals”

You can import well intervals from a column based text file by selecting “Well Data” from the “File”->“Import” menu and then selecting “Well Interval (tops) file”.

You can export well intervals to a text file by right clicking an interval set objects and selecting “Export Intervals” from the “Export” menu.

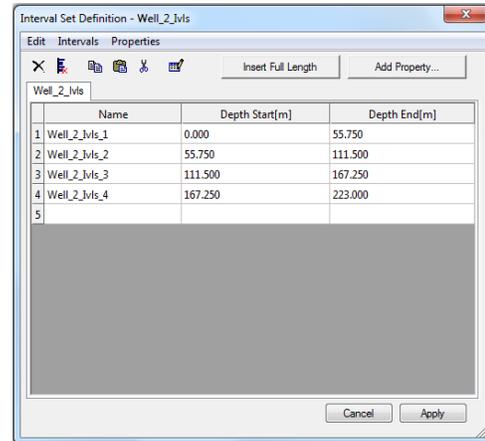
You can duplicate well interval sets by selecting “Duplicate Object” from the “Edit” menu.

You can move a well interval set from one well to another by dragging the object in the object bar and dropping it on the destination well or by selecting the object and selecting “Reparent Object” from the Edit -> Organize Scene menu.

Editing Well Intervals

You can edit well intervals by right clicking on an interval set object and selecting “Edit Intervals” or by right clicking on a well and selecting “Edit Well”. If you’ve selected the well you will get a tab for each interval set owned by the well object.

In the interval set tab, you will have a row for each interval defined. Columns for the interval name, start and end measured depths as well as any properties that are defined for the interval sets

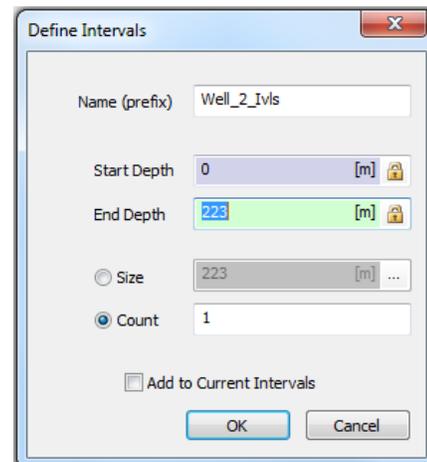


	Name	Depth Start(m)	Depth End(m)
1	Well_2_Ivls_1	0.000	55.750
2	Well_2_Ivls_2	55.750	111.500
3	Well_2_Ivls_3	111.500	167.250
4	Well_2_Ivls_4	167.250	223.000
5			

You can define multiple intervals at once by selecting “Define Multiple” from the “Intervals” menu. This dialog allows you to define intervals a fixed number or constant sized intervals from a start to an end depth.

If you do not check “Add to Current Intervals” the new intervals will replace any intervals you have currently defined.

Intervals are allowed to overlap but this is not recommended. If you want multiple overlapping intervals it is recommended that you group them into separate interval set objects.



Name (prefix) Well_2_Ivls

Start Depth 0 [m]

End Depth 223 [m]

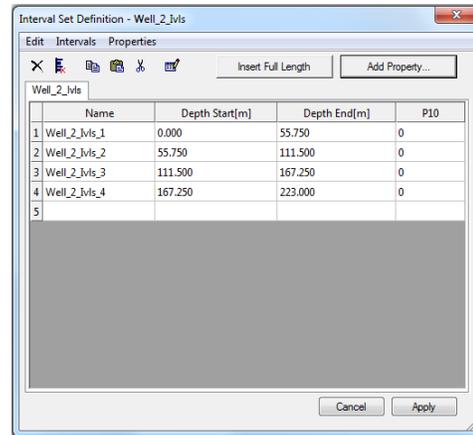
Size 223 [m] ...

Count 1

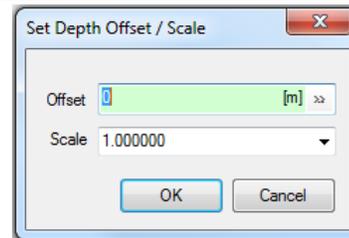
Add to Current Intervals

OK Cancel

You can edit the property values of an interval set by manually entering the values for each interval. You can also import or define the property values using the Edit Properties feature.

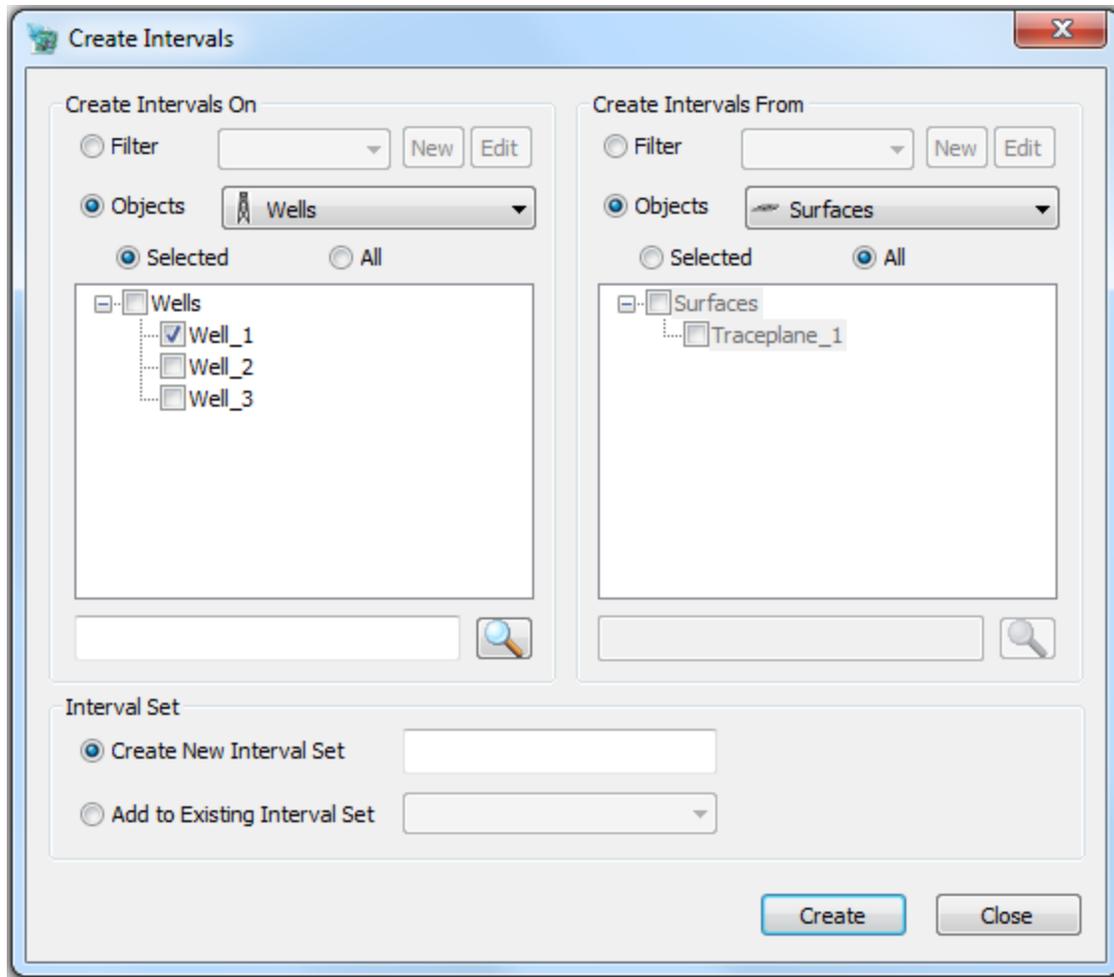


You can also adjust the depths of the well, either by adding an offset or scaling them, perhaps to convert feet to meters, by using the Offset/Scale Intervals function available from the Edit menu->Well or by right-clicking on a Well.



Creating Intervals from Objects

You can define by the intersections between the wells and region/surface objects or by well logs.



- **Wells** - Select the well(s) you want to add intervals to.
- **Interval Set** – Select whether to add to an existing interval set or a new one
- **Intervals** – “Define From Well Logs” will define the intervals based on the min and max measured depth from the selected well logs. “ Define from Object Intersections” will attempt to compute the intervals based on the intersection of the well trajectory with the indicated selected region/surface object(s).

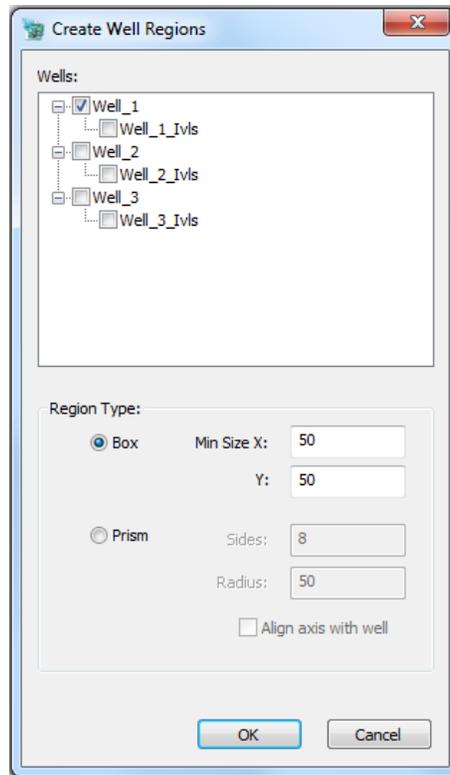
Create Regions around Wells/Intervals

You can quickly create region objects around wells/well intervals by selecting “Create Regions around Wells”.

Select the well, interval set or intervals you want regions around.

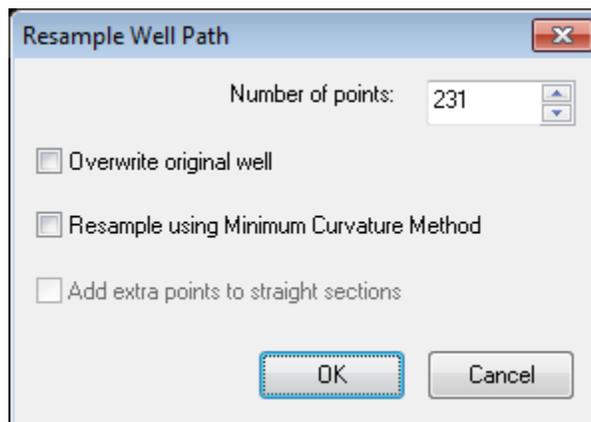
Select the type of region (Box / Cylinder) and the size of the region.

If you check “Align Axis with Well” (cylinder regions only) the cylinders will be aligned with the well trajectory rather than with the bounding box of the object.



Resample Well Path

This changes the path of the well. For complex well traces it can be used to reduce the number of points on the path thus 'simplifying' it. However it can also be used to interpret each line segment in the path as a representation of the inclination, azimuth and change in measured depth. And it will then recalculate the path of the well from these values using the Minimum Curvature method. As part of this recalculation of the path it can optionally add extra points to give a more detailed path. It is also possible to add these extra points to existing paths that were created using the Minimum Curvature method.

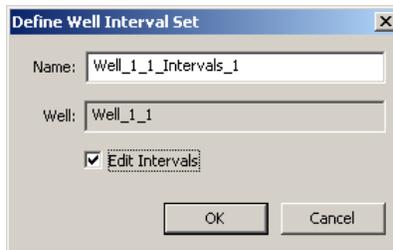


Choose the number of points you wish your new well trajectory to have by pressing the up and down arrows next to “Number of points” or by entering the number manually into the white box. If you wish to replace the original well trajectory with the simplified one, check the small box next to “Overwrite original well”. This will delete the original well trace. If you wish to keep the original well trace or are uncertain if this simplification is the one you want, you should not check this box as the process is irreversible.

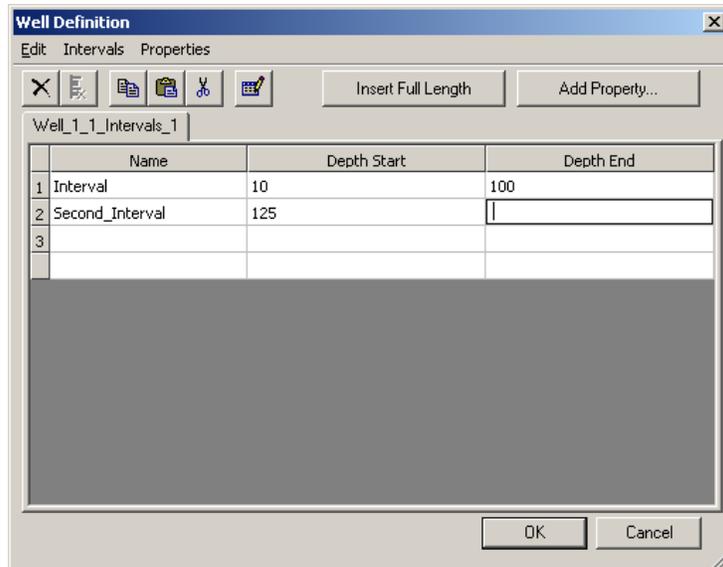
The “Resample using Minimum Curvature Method” option is only available if you are not reducing the number of points. Check this if your existing path’s segments are straight line representations of inclination, azimuth and change in measured depth and you wish to recalculate the path using the Minimum Curvature method. If you don’t check this option then the path will be assumed to have already been calculated using the Minimum Curvature method, and extra points will be added between the existing ones. These will generally be on arcs of circles which correspond to the path assumed by the Minimum Curvature method.

The final option is only available if you are increasing the number of points. It controls the behavior if an additional point would lie on the straight line between the adjacent points. Such points do not change the path or make it any more detailed, so they are normally not created. However if you want to include these points (for example for use in interpolation) you can check the ‘Add extra points to straight sections’ option.

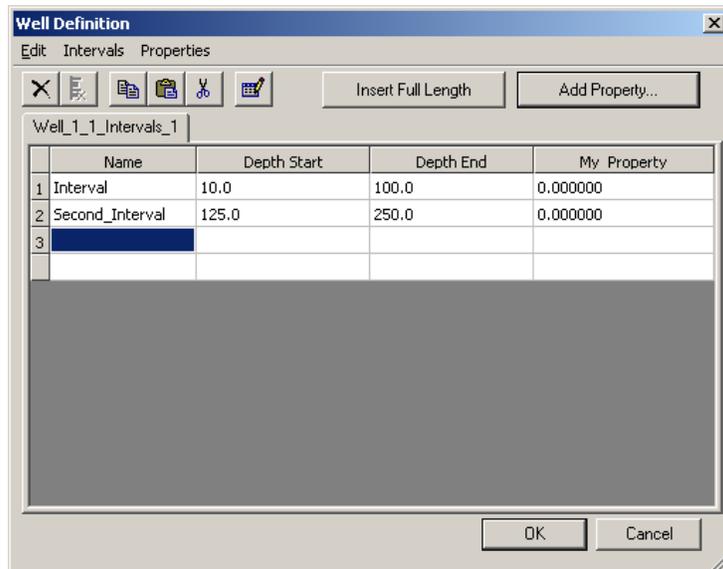
In order to create a well interval, right-click on the well name in the Objects window and hit **Add Interval Set**.



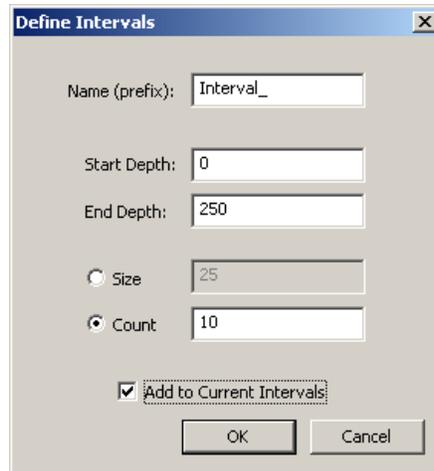
Here you can give your interval set a name, such as P10_Intervals, or Perfs. If “Edit Intervals” is checked, clicking **OK** will take you to another dialog box where you can define your intervals.



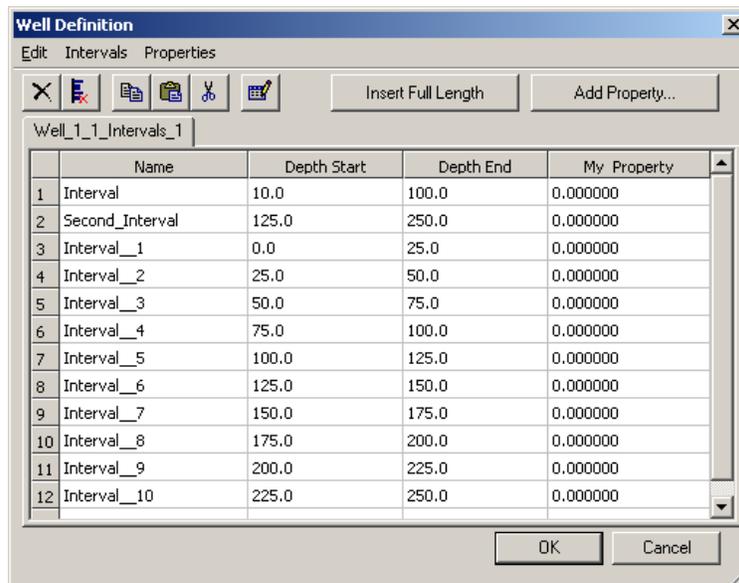
Input the name, start and end depths in the appropriate columns here. You can also add a property, such as P10 or Radius, by clicking Add Property. Enter the property name, then hit **OK**.



If you wish to make several intervals of even size without manually inputting them all, you can click the  symbol or select "Add New" from the "Properties menu".



Specify a start and end depth along the well within which the intervals will be created. If you wish to create intervals of a given length along the well, click “Size” and specify that length; FracMan will create as many intervals of this size that it can fit between the specified start and end depths. If, instead, you wish to create a set number of intervals in the space, click “Count” and the space between the start and end depths will be divided into the given number of intervals. All intervals will be the same size, without separation between them. If “Add to Current Intervals” is not checked, any intervals you have already defined in this interval set will be deleted. All the intervals will have the specified name followed by a number (with 1 being the uppermost interval).

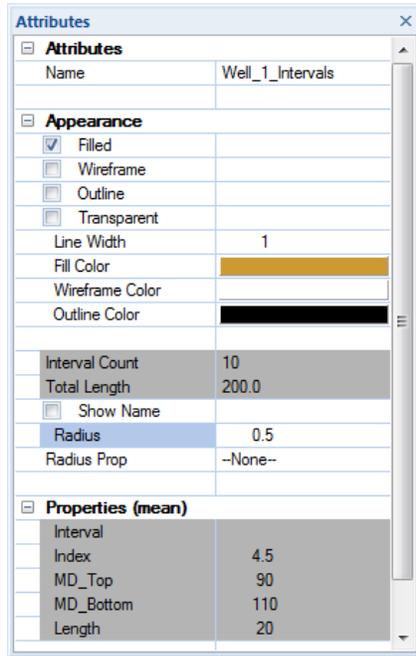


When you are done defining your well intervals, hit **OK**.

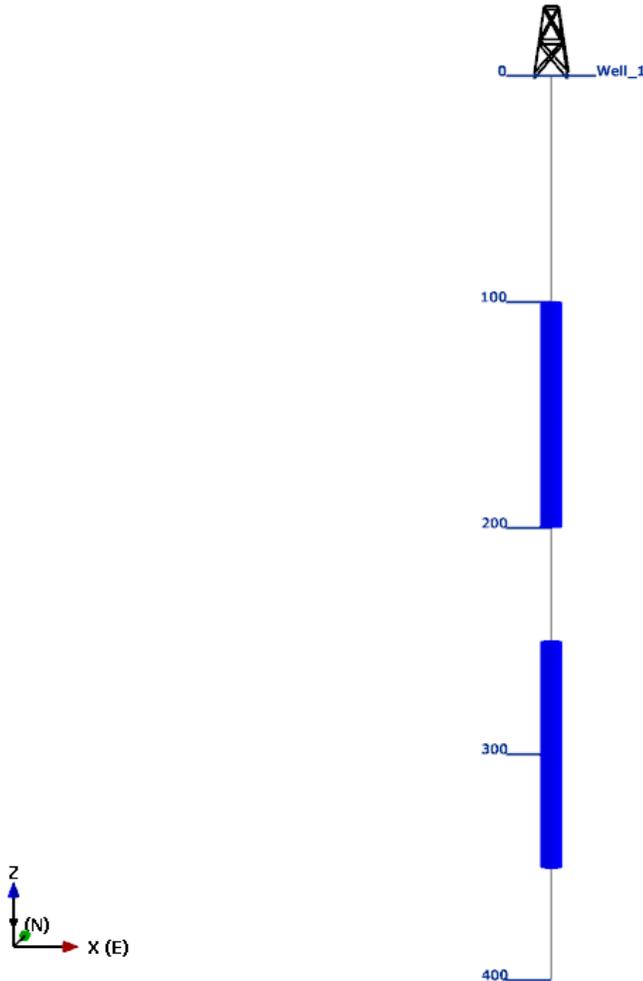
Visualization

You can control the display of individual interval set objects by checking or unchecking the box in the objects bar. Properties can be visualized by right clicking and selecting the desired property name from the “Visualize” menu. You can specify the displayed radius of an interval set by setting the value in the “Radius” box in the Attributes bar

You can visualize the radius of well intervals as property values by selecting the property name in the “Radius Prop” box. Clear this selection by selecting “—None—”.



Below: An example well, showing intervals in blue.



Import / Export Wells

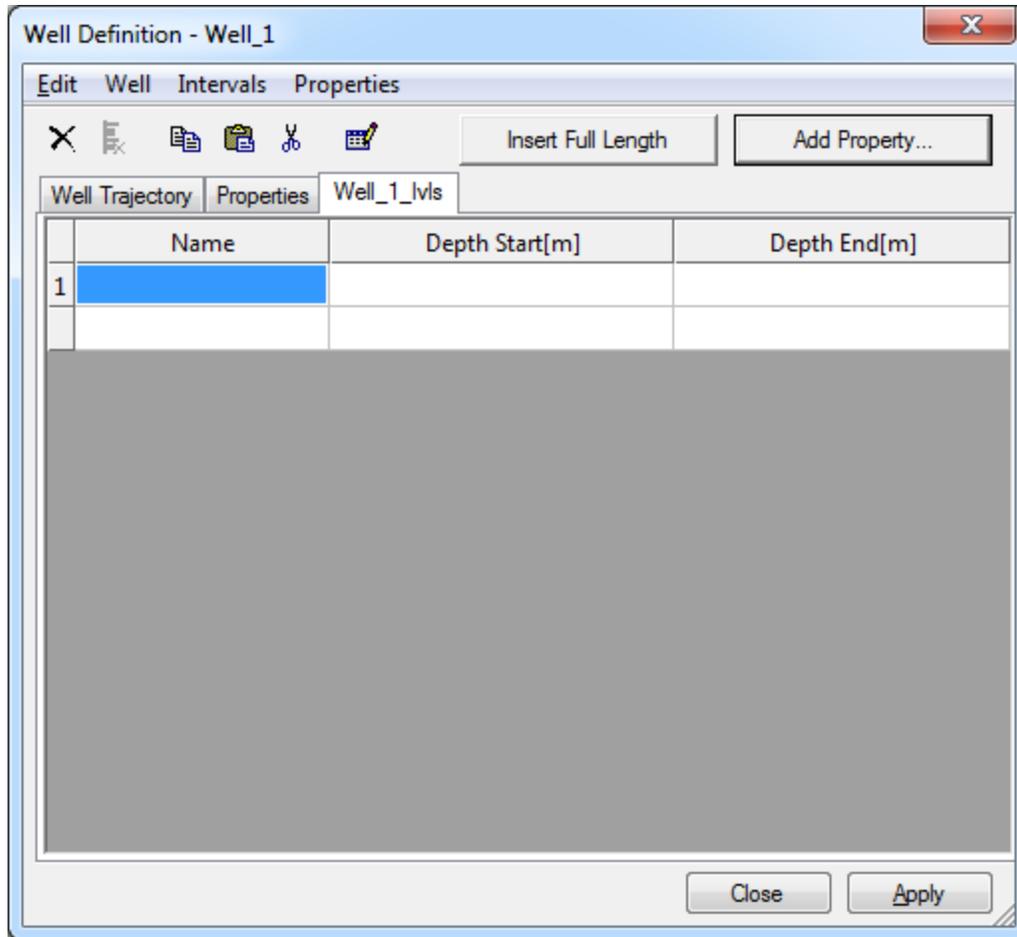
Well Interval Sets can be exported to a file by selecting one or more objects in the objects bar and selecting “Well Intervals” from the “Export” menu.

Well	IntervalSet	Interval	Index	MD_Top	MD_Bottom	Length
Well_1	Well_Intervals	Interval_1	0	0	20	20
Well_1	Well_Intervals	Interval_2	1	20	40	20

You can also import well intervals from a similar formatted file. If measured depth bottoms are not available it will assume that the next top is the current interval bottom.

Property Definition / Import / Export

You can create well interval properties by pressing the “Add Property” button in the well interval edit dialog.

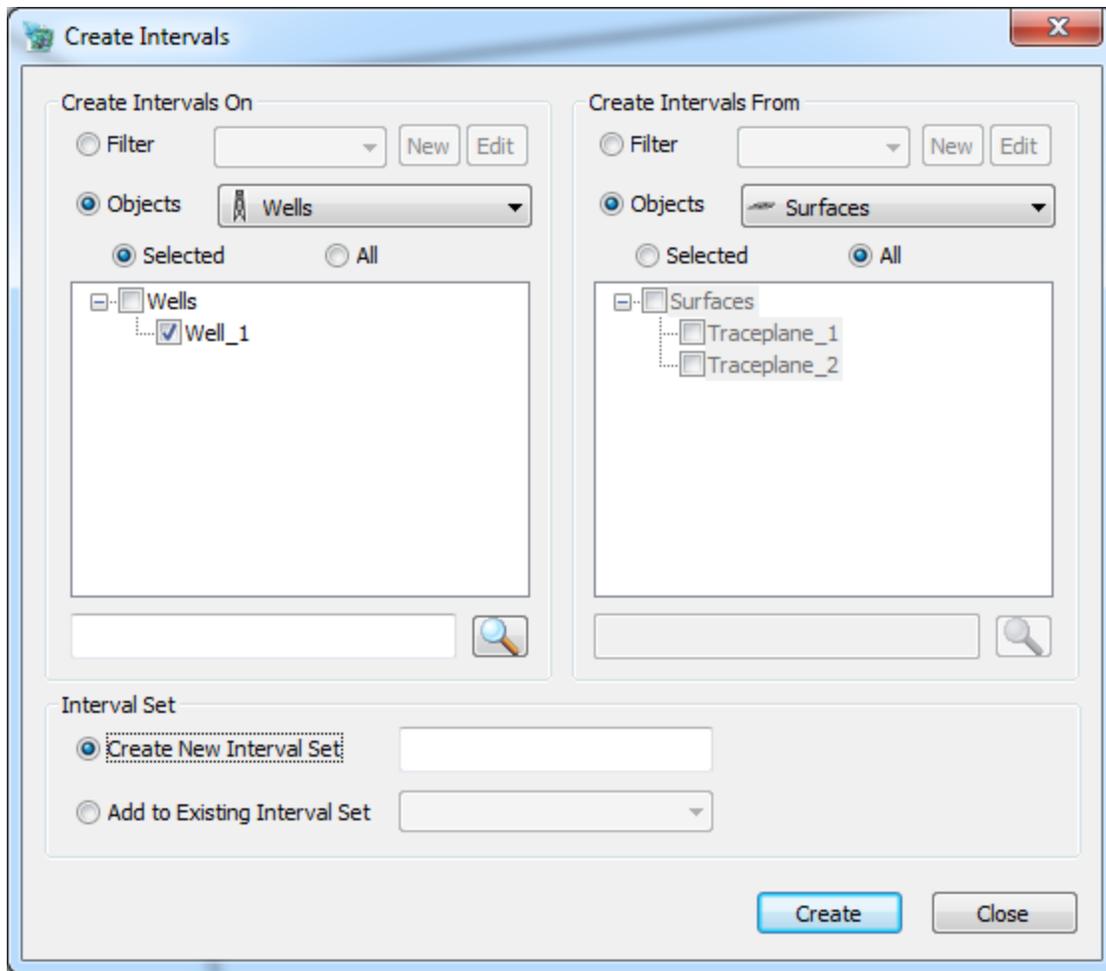


You can also create well interval properties by selecting one or more interval set objects and selecting “Modify Properties” from the “Edit” menu. This will give you all the property capabilities available to other object types.

Create Well Intervals from Well/Object Intersections

You can define well intervals from the intersections of the well with certain types of objects or as the extent of well log objects. You can add these new intervals to an existing set or in a new interval set.

First select the well(s) to which you wish to affix the interval set. Then select *Edit > Well > Create Intervals From Objects* from the menu bar.



If you click the button under “Wells:” you can check the names of all the wells you wish to create intersection intervals for. You can either create a new interval set, which will have whatever name you specify in the box next to that option, or you can add the intervals to an existing set selected from the drop-down menu. Finally, choose whether you wish to create your intervals based on an existing well log, or from the intersections between an object and the well trace. Check the well log(s) or object(s) you wish to use. Then hit **OK**. The interval set will appear as a child of the select well(s).

Well Logs

In the field, valuable information about fracture locations, orientations and flow properties can be derived from wells.

Loading a Well Log

FracMan can extract any of this data from whitespace-delimited ASCII files. When the data is loaded into the program you can specify which portion of the file is read and which column(s) is used.

FracMan can also read well fracture log information from an ORS file, along with trajectory info described in an associated SAB file. It can also read fracture information from an LAS file.

Creating an Empty Well Log

Select “Blank Well Log” from the “Insert Menu”. Pick one or more wells, start depth, stop depth and the depth increment.

The screenshot shows the "Create Well Log(s)" dialog box. It features a "Wells / Intervals" section with a "Filter" dropdown menu and "New" and "Edit" buttons. The "Objects" dropdown menu is set to "Wells", and the "Selected" and "All" radio buttons are visible. Below this is a large empty rectangular area. Underneath is a search bar with a magnifying glass icon. The "Starting Depth" section has "Top" selected and a value of "0 [m] >>". The "Ending Depth" section has "Bottom" selected and a value of "0 [m] >>". The "Increment" field has a value of "5 [m] >>". There is a "Name Base" checkbox and an empty text field. At the bottom, there is a "Create More" checkbox, a "Create" button, and a "Close" button.

Creating a Well Log from Fracture data

See the section [Well Intersecting Fractures](#).

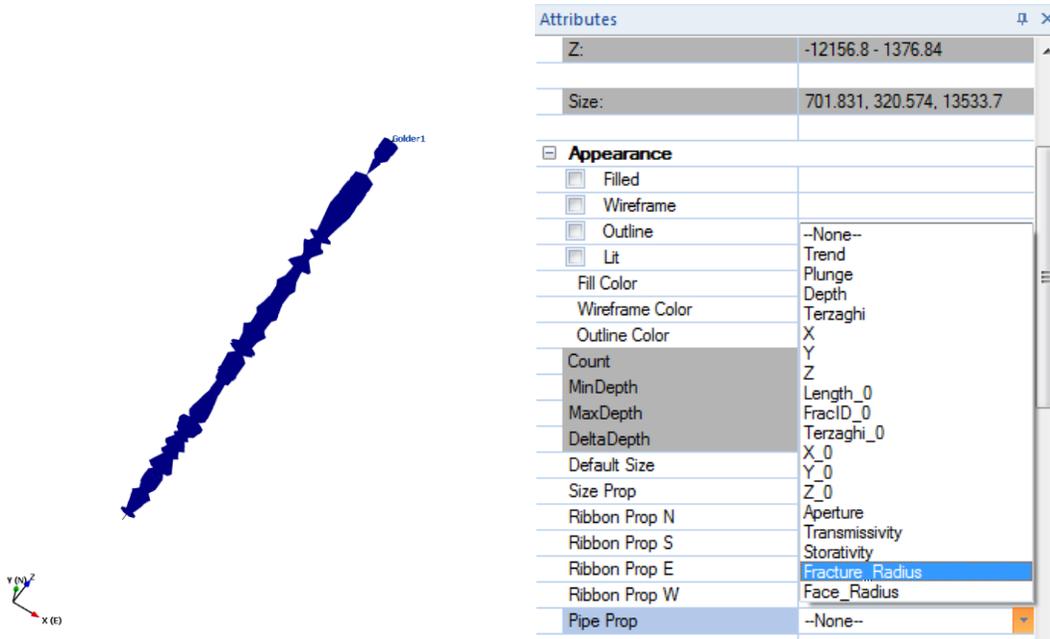
Visualizing a Well Log

By default a well log appears as a series of discs along the well trajectory. Since a well log uses Measured Depth for its location, it must be a child object of a well object. To move a log to a well (if it doesn't have one) simply drag the object in the Objects Bar and drop it on the well object, or right click on the log object and select “Connect to Well”.

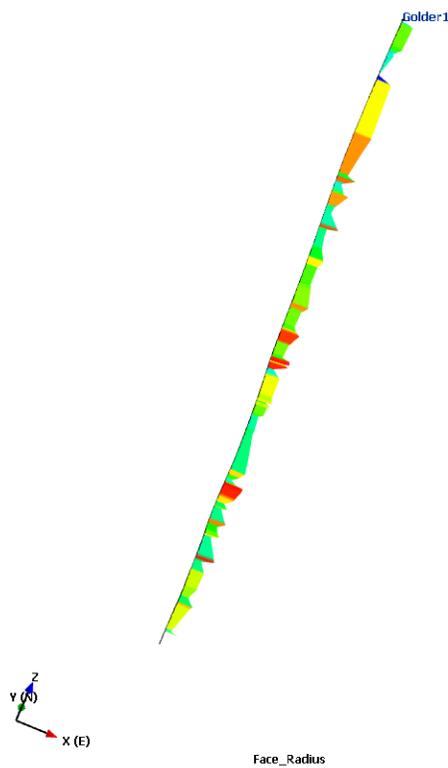


You can change the size of these discs by selecting the log object and setting the “Default Size” value in the Attributes Bar. You can also set the size to correlate to a property value by selecting the property in the “Size Prop” line in the Attributes Bar.

You can visualize the log as a lathe / 3D pipe by selecting a log property on the attributes window from the “Pipe Prop” box. You should also deselect the Filled / Wireframe / Outline options when using the lathe display. The max radius of the pipe is controlled by the “Default Size” line in the attributes window. This mode will also use the color specified by the Legend Bar.

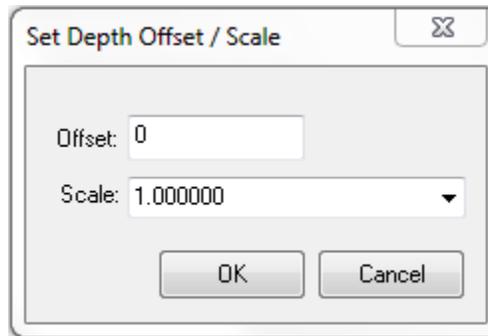


You can visualize up to four separate properties as 2 dimensional “ribbons” running along the well (1 for each compass direction). Select the well log object and select the property in the Ribbon Prop N, S, E or W. This mode will also use the color specified by the Legend Bar.



Converting Well Log Units

Often well logs are stored with measure depth stored in feet, to convert to meters, right click on the log object(s) in the Objects Bar and select “Set Depth Offset / Scale”.

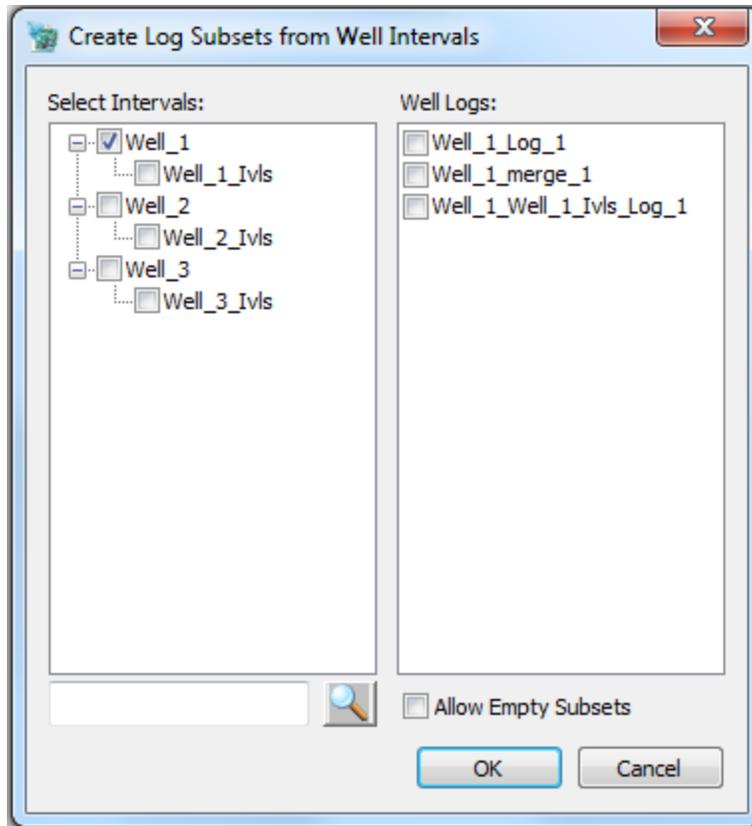


Extracting Subsets

In many cases there will be a number of different types of data mixed together in a single well log. You can create a new well log that contains a subset of the data by selecting “Create Data Subset” and specifying the filter criteria of the data you want extracted.

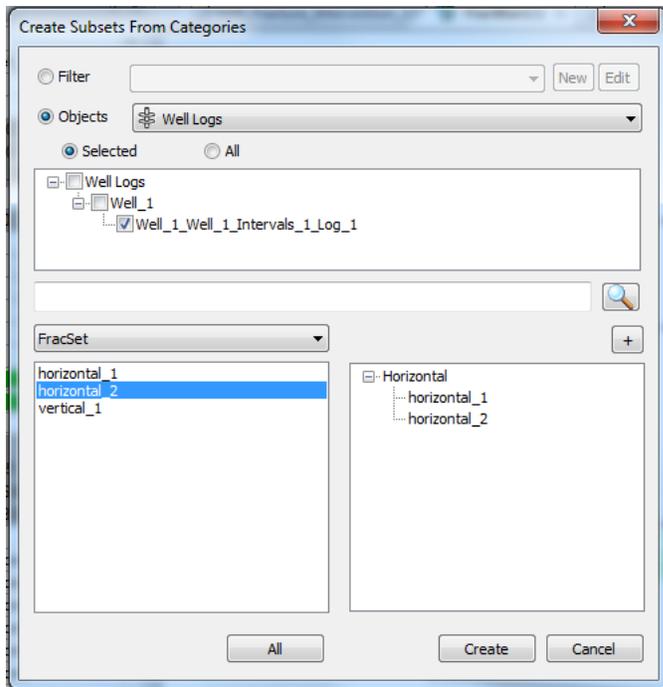
Create Well Log Subsets from Well Intervals

It is now simple to separate well logs into subsets by depth range. If you set up all your intervals of interest as interval sets you can create well logs of those intervals over your entire model in one operation. Select **Create Log Subsets From Well Intervals** from the **Edit | Well** menu. Select the intervals you are interested in on the right and the corresponding well logs on the left. You can select multiple well intervals by entering a portion of the well name along with one or more wildcard characters and pressing the search button. Checking the “Allow Empty Subsets” box will create a subset object even if there are no values in that interval.



Create Subsets from Categories

Well logs occasionally have string properties with values like “Open” and “Closed” (often these are termed “Categories”). A common operation is to split this well log into multiple subsets according to the values of the string property. This can be done via filters, but you have to do them one value at a time – dividing the well log into multiple subsets, one for each string value. Using the function “Create Subsets from Categories” allows the user to divide a well log into subsets using multiple category values. The left pane of the dialog window shows all the values (categories) from the selected property, drag them over to the right pane to define the subsets you want. For example, you could select values of “horizontal_1” and “horizontal_2” to be combined into one new “Horizontal” subset. To do this, first click on the “+” symbol to create a new subset, give it a name and then drag & drop the two values “horizontal_1” and “horizontal_2” to the new subset. Click Create to make the new subset.



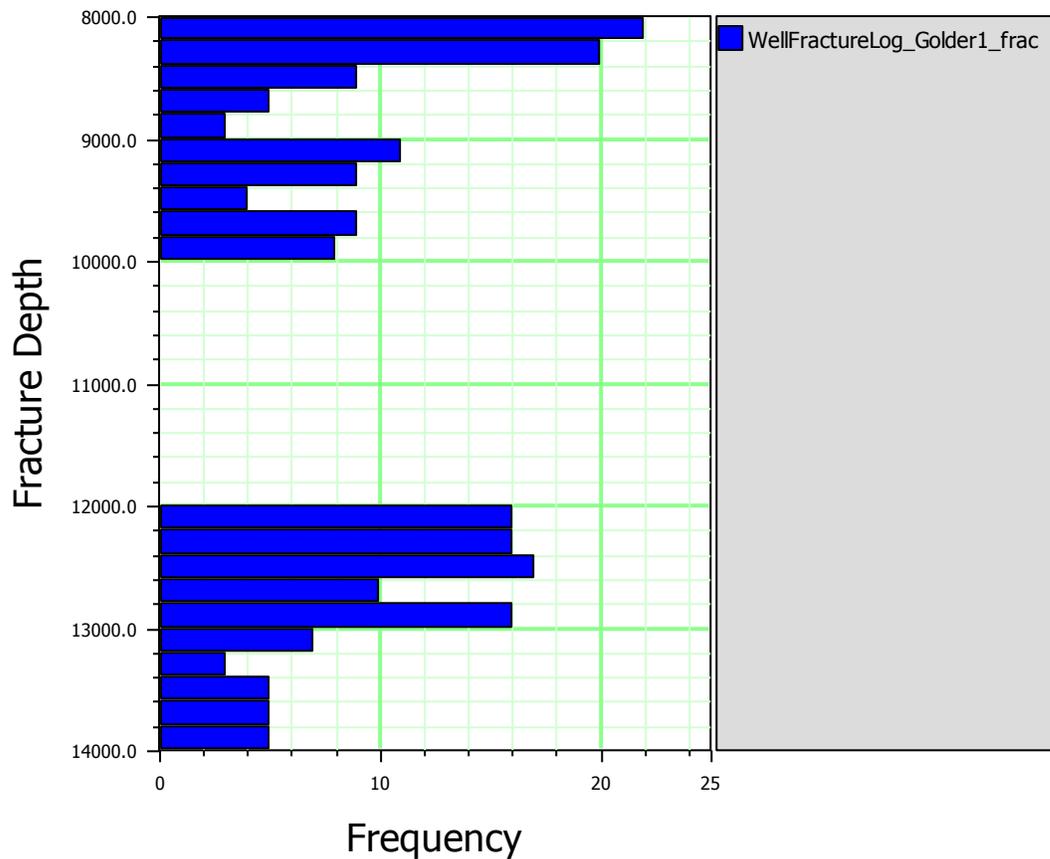
Merge Well Logs

This is the inverse of “Create Data Subset”; when this operation is performed, the selected well logs will be merged together into one. First select two or more well logs in the Objects window; then select *Edit > Well > Merge Well Logs* from the menu bar. The resulting single, comprehensive well log will appear in the objects window with the appropriate well name and the suffix “_merge”. The original logs will not be deleted by the function.

Depth Histogram

Right click on one or more well logs and select **Histogram | Depth Histogram** you will get a vertically oriented histogram showing the fracture counts along the well depth.

Fracture Depth



Cumulative Fracture Intensity Plot

See the topic [Cumulative Fracture Intensity Plot](#)

Well Log Statistics

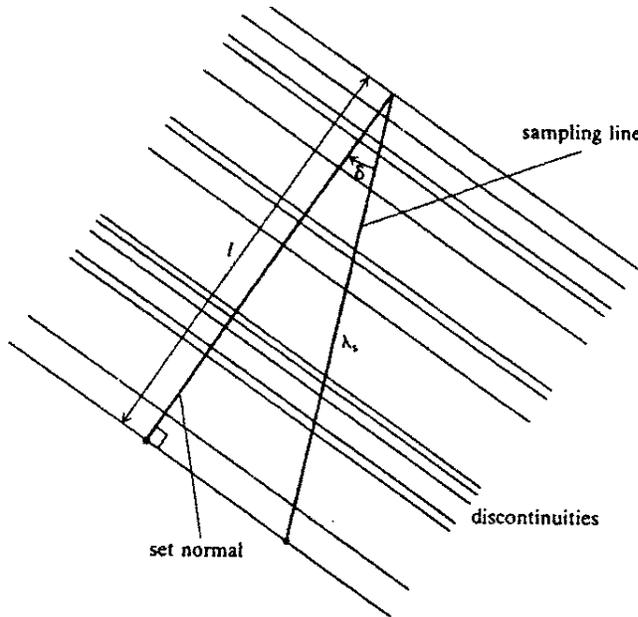
You can compute statistics of one or more selected well logs by right clicking the well log and selecting "Statistics" from the "Analysis" submenu, or by selecting Static Analysis->Statistics.

The analysis results include basic fractures statistics as well as:

- RQD: The Rock Quality Designation (RQD) for the simulated core. RQD is defined as the total length of core pieces of length over 10 cm divided by the borehole length.
- RQDQ: The RQD, adjusted for borehole diameter D using the method of Heuze (1971). The corrected RQD is equal to the total length of core pieces greater than Q meters, divided by the borehole length. Q is calculated from the borehole diameter D in meters as:
$$Q = 0.00625348 \text{ m} + 0.021835 \text{ D.}$$
- RQD13: This is defined as the total length of core pieces of length over 13 cm divided by the borehole length.

Terzaghi / Mauldon Correction

When three-dimensional features such as fractures are sampled along a one-dimensional line (such as a borehole centerline or a scanline along a tunnel wall), a systematic sampling bias occurs. This bias arises from the fact that the probability of intersecting a fracture of a given orientation is a function of the solid angle between the sampling line and the fracture normal. The smaller the solid angle is, the lower the probability of intersection. This bias can lead to the under-estimation of fracture sets that are oriented roughly sub-parallel to the sampling line. Orientation bias is covered in great detail by Terzaghi (1965) and Priest (1993).



To compensate for the orientation bias, Terzaghi (1965) proposed a geometrical correction factor based on the observed solid angle between the sampling line and the normal to a particular fracture. In FracMan, such a correction is called the “Terzaghi correction”. The acute angle between a normal to the fracture plane and the borehole is termed δ . The highest probability for intersection occurs when $\delta = 0^\circ$. The lowest probability of intersection is zero; this occurs when $\delta = 90^\circ$. Any direction of sampling line will therefore produce a sample that is biased to contain

a lower amount of fractures than the actual amount. The reduced sample size at the higher values of δ can be compensated for by assigning a higher weighting to those fractures that are sampled. A fracture sampled by a sampling line is assigned a weighting W given by:

$$W = \frac{1}{\cos \delta} \quad \delta < 90^\circ$$

This value W is assigned as a property “Terzaghi” for each fracture, and is used in [Bootstrap Monte Carlo Simulation](#), and in [Set Identificatoin Stereonet](#) plotting and orientation distribution fitting.

Mauldon and Mauldon (1997) provide a significant improvement on the Terzaghi (1965) method. Terzaghi (1965) is based on the assumption of a zero-thickness sampling line intersecting an infinite fracture. This leads directly to the $1/\cos \delta$ term, which goes to infinity for $\delta = 90$, and to unreasonably large values for δ near 90. Mauldon and Mauldon (1997) calculate the probability of intersection between a fracture of known radius R and cylindrical sampling object of radius r . The resulting equation is used to provide a better estimate for the “Terzaghi Factor” W .

Correction Input Values

The Terzaghi correction is used in FracMan as follows. The user selects one or more fracture logs from the object menu. From the right click menu, the user then chooses **Analysis/Terzaghi-Mauldon** to calculate the Terzaghi correction which is saved as a fracture property. The user must specify the following inputs:

- Select either **Terzaghi** or **Mauldon** for the correction type.

It is recommended to select **Mauldon** wherever possible, as this produces a significantly more accurate result for most cases. The **Terzaghi** correction is over 50 years old, and well established as an industry standard. Therefore, the Terzaghi correction should be used where it is important to provide comparisons to results from other groups, or where it is distracting to explain the advantages of the Mauldon correction.

- Specify the **Maximum Correction**.

This is necessary as the term $1/(\cos \delta)$ becomes unreasonably large for fractures sub-parallel to the sampling surface. A value between 5 and 10 is generally used in the industry (Yow, 1987; Priest, 1993). The recommended value for maximum correction of 7 corresponds to a minimum bias angle of 8.2°.

- Specify the mean **Fracture Radius** if using the Mauldon correction.

The Mauldon correction utilizes the ratio (r/R), where r is the wellbore (or sampling line) radius, and R is the fracture radius to adjust the factor W to more accurately adjust for the linear sampling bias. The user should estimate this mean fracture radius from available data, where possible, or based on geological understanding if no data is available. Generally, the appropriate value is on the order of 5 to 50 meters. Note that for the Mauldon correction, FracMan assumes an Exponential Distribution, such that the mean and standard deviation are equal. The exponential distribution is used as an approximation for convenience – the Mauldon correction is not made significantly more accurate if the true radius distribution is used.

Note that both Terzaghi and Mauldon can only be used for fracture orientation data attached to a well or traceplane. Mauldon can only be used for orientation data attached to a well interval with assigned radius r .

Monte Carlo Simulation (Bootstrap)

[Using a "Bootstrap" orientation during fracture set creation](#) generates Monte Carlo values of fracture orientation by resampling from selected orientation data. If the data selected have not

been Terzaghi/Mauldon corrected, then the generated fractures have the same bias as the selected data – which can introduce significant error to the analysis. In contrast, when the data used for the Bootstrap distribution have been “Terzaghi” corrected, then the probability of using each measured value is adjusted in proportion to the assigned value of W. While this does not completely correct for the sampling bias, it can provide a significantly more accurate population for Bootstrap sampling.

The advantage of Bootstrap sampling from Terzaghi/Mauldon corrected data can be verified by sampling the Bootstrap-generated fractures with a borehole, and comparing this simulated sample with the selected data used for the Bootstrap.

Stereonets and Orientation Distribution Fitting

FracMan can use assigned values of the “Terzaghi” property in preparing contoured Schmidt and Wolf stereonets, and in fitting orientation distributions. This is recommended where linear sampling bias is present in the fracture data selected for plotting or analysis. To include the Terzaghi/Mauldon correction during Set Identification analysis, first calculate the correction using the **Analysis/Terzaghi-Mauldon** function from the right click menu of the selected well log. Then assign the Terzaghi property to the Weight on the Set Identification Definition dialog.

Limitations of Terzaghi Correction

As mentioned above, the Terzaghi correction, and even the Mauldon correction has significant limitations. The value of the Terzaghi weight is asymptotic as α (the angle between the borehole axis and the plane of the fracture) approaches zero (Yow 1987). This has the potential to produce an unacceptably large Terzaghi weight for fractures with orientations very close to that of the borehole axis. It is generally accepted (Yow, 1987; Mauldon and Mauldon, 1997) that a ‘blind zone’ exists between $\delta = 0^\circ$ and $\delta = 20^\circ$. Within this blind zone, the number of fractures sampled is frequently not large enough to apply either a Terzaghi or a Mauldon correction accurately – remember that these corrections only increase the frequency of sampling from fractures which have been sampled... if no fractures have been sampled, no value of the Terzaghi parameter “W” is calculated.

Davy et al. (2006) showed that, for fracture networks where fracture size follows a Pareto (power-law) length distribution, the Terzaghi correction will over-estimate the frequency of fractures sub parallel to the borehole.

Compute P10

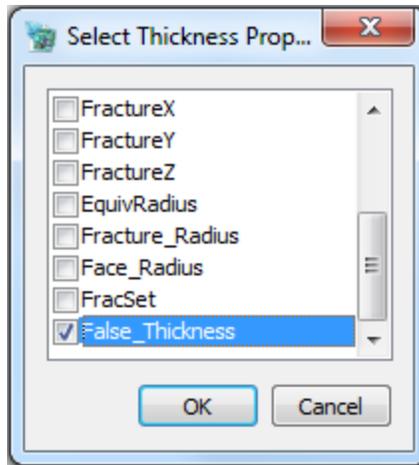
See Manual section [Computing Linear Intensity \(P10\)](#).

Compute Sonic To Elastic Materials Properties

See Manual section [Compute Material Properties from Sonic Log Data](#) for details.

Compute True Thickness

If you have a well log that shows the layering in a system, you can compute the actual thickness of the layers, taking into account the orientation of the layers, the trajectory of the well log and a thickness value. The computed thicknesses will be stored as a new “TrueThickness” property. This function can also be used to correct the apparent apertures of fractures.

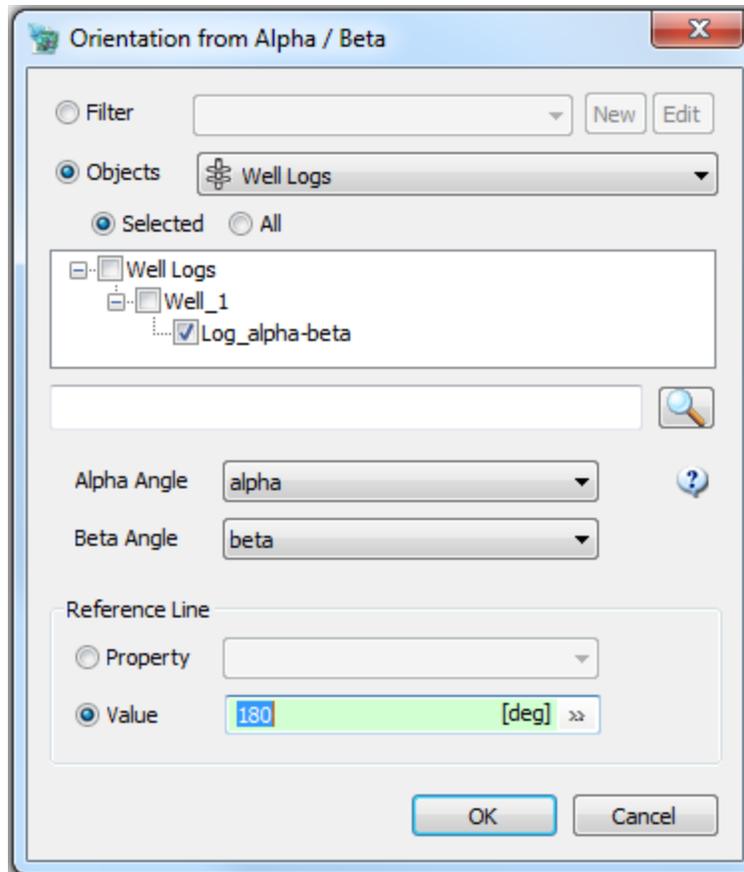


Compute Alpha Angle

Creates a new well log property, “AlphaAngleDeg”, containing the acute angle formed between the well trajectory and a well log planar feature in degrees. This will be equal to $90^\circ - \text{DipAngle}$ for vertical wells.

Compute Orientation From Alpha/Beta

Used to convert between alpha and beta angles to pole trend/pole plunge and dip angle/dip direction in inclined or deviated boreholes. Alpha and beta values can be entered as “ASCII Well Fracture Files” for a “Well Log” using the [File Format Assistant](#). You do not need to have measured depth associated with the alpha and beta measurements as long as the borehole is of constant inclination. The borehole must be loaded into FracMan either from trajectory data or by inserting a new well. The only required data for the borehole are the inclination/deviation trend and plunge (and possibly the measured depth or segment length if the borehole is deviated and alpha and beta values have measured depth values). Assign the well log to the well/borehole. Select the well log, right-click and select **Compute** -> **Compute Orientation From Alpha/Beta**. The **Orientation from Alpha / Beta** dialog will open.



Select the well log properties containing the alpha / beta values. Enter the reference angle and click **OK**. If the data has a varying reference angle, this needs to be imported along with the alpha and beta values and then select it as a **Property** in the **Reference Line** section instead of entering a constant **Value**. View the results by right-clicking on the well log object and selecting **View Data**. The newly calculated properties can be exported to a file using the top menu **File -> Export -> Object Properties** or simply by copy and pasting the values by right-clicking on the View Data results screen.

Tracemaps

Tracemap objects are a set of line segments in 3 dimensional space. They are generally used to represent the intersection between a fracture and a face of a surface, traceplane or tunnel object. Tracemap objects have vertex, trace and segment properties. You can generate fractures from a tracemap object using a [Tracemap fracture set definition](#).

Loading a Tracemap

Select Import | Tracemap from the File menu. Select the file and type that corresponds with that file in the file dialog that appears. See [Appendix A FracMan Supported File Formats](#) for a list of supported file types.

Select one or more files and select “Open.”

Creating a Tracemap

A Tracemap can be created from the intersection between a fracture set and a surface or region object.

In the Objects Window, right click on a surface or region object and select "Compute Fracture Traces". All fractures in the model will be used to compute fracture traces in the object.

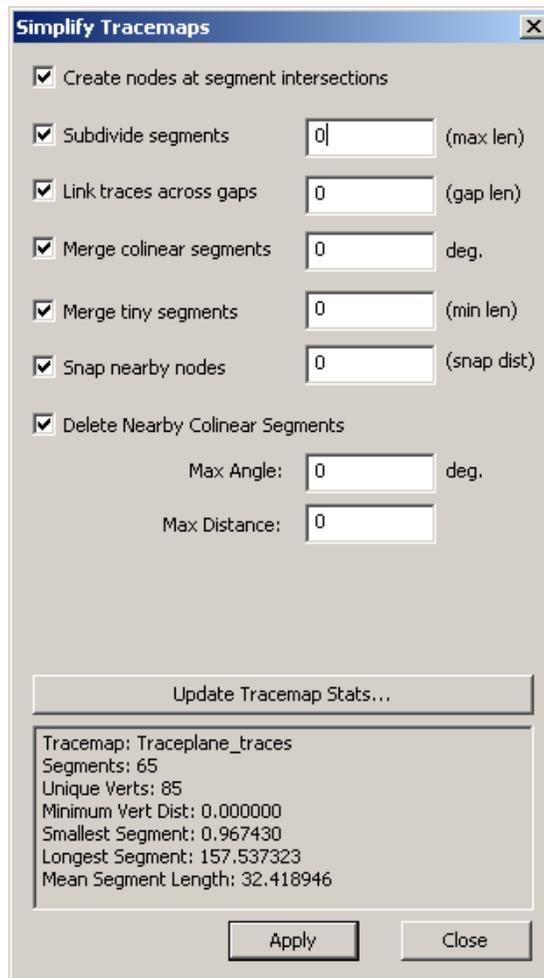
The tracemap object will appear in the objects window as a child object of the sample object.

Creating Tracemaps from Region or Tunnel Objects

You can also create a tracemap by selecting "Intersecting Fractures" from the "Surface" submenu under "Static Analysis" see [Surface/Region Intersecting Fractures](#). This will allow you to specify which fractures to use and what surfaces (or regions) to intersect with.

Simplify Tracemap

In order to use a tracemap within the ELFEN software it usually needs to be simplified and cleaned up. To do this, select one or more Tracemap objects and select "Edit -> Tracemap -> Simplify Tracemap"



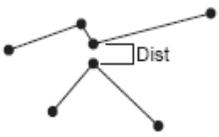
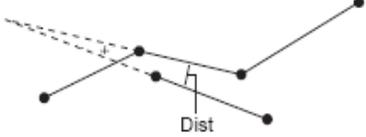
There are seven possible "cleanup" operations (see figure below):

- Create nodes at segment intersections
- Subdivide segments (specify the maximum length you wish the resulting smaller segments to be; all segment subdivisions will be this size or smaller)
- Link traces across gaps (specify the largest gap size you wish to link traces across; all gaps that size and smaller will be eliminated)
- Merge colinear segments (specify the maximum number of degrees by which the segments' orientations may differ and still be considered "collinear" and therefore mergeable)
- Merge tiny segments (specify the maximum segment length that can be considered "tiny"; all segments smaller than this will be eliminated)
- Snap nearby nodes (specify the maximum distance that nodes can be from each other and still be considered "nearby" and therefore snap-worthy)
- Delete nearby colinear segments (specify a maximum angle defining colinearity and a maximum distance defining segment "nearness"; for two segments this far apart and closer, whose orientations differ by the given angle or less, the smaller segment will be deleted)

To select which operation(s) to use, simply check the appropriate box. The Update Tracemap Stats button will display a list of statistics about your tracemap, including the number of traces and the average segment length.

When you are ready to simplify your tracemap, click Apply. This process cannot be reversed.

Tracemap Simplification

Create Nodes at Intersections		
Subdivide Segments (max length)		
Link Traces (max gap)		
Merge Colinear Segments (degrees)		
Merge Tiny Segments (max length)		
Snap Nearby Nodes (snap dist)		
Delete Nearby		

The different tracemap-simplification operations are shown above.

Merge Tracemaps

FracMan will also allow you to merge multiple tracemaps together into a single object. In order to do this, just select *Edit > Tracemap > Merge Tracemaps*.



Select the tracemaps you wish to combine from the list. If “Retain Original Objects” is checked, the merged tracemaps will not be deleted when the single combined tracemap is created. Otherwise, only the product of the combination will be retained. When you are done, hit **OK**. The merged tracemap will appear in the Objects window under the name you specified.

Create Data Subset

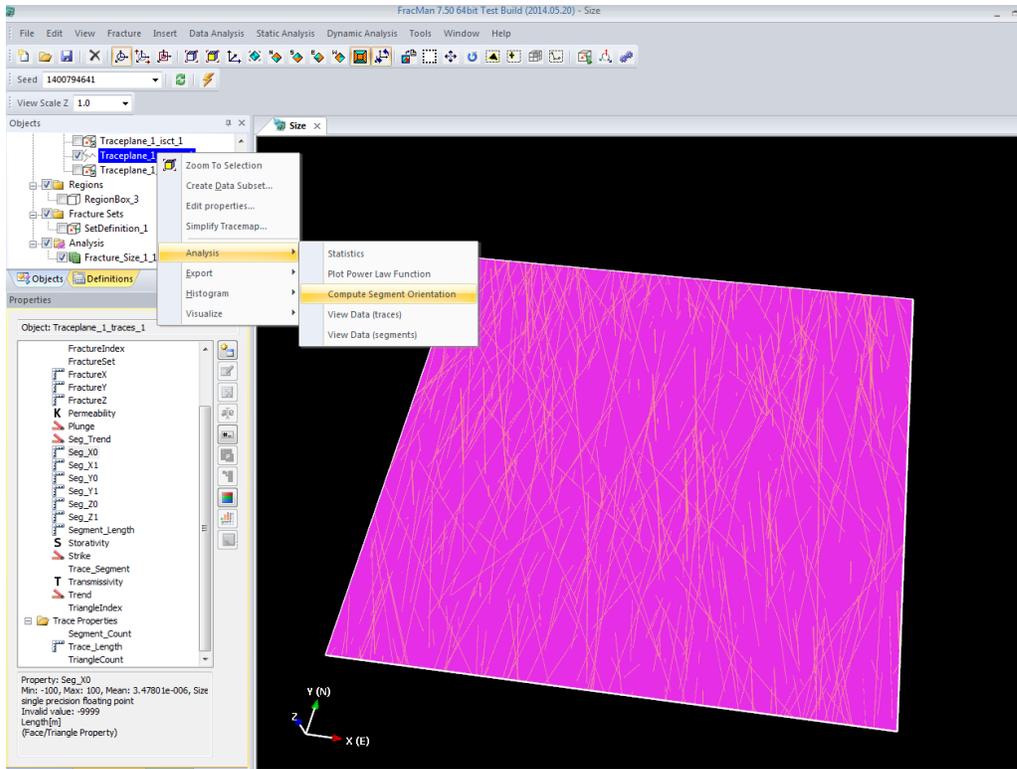
You can extract portions of a tracemap by selecting "Create Data Subset". This will allow you to remove short tracemaps or to select tracemaps defined by a specific property value or type. You can extract a new tracemap containing a subset of an existing tracemap by selecting “Create Data Subset” from the Edit -> Tracemap menu.

Create Subset From Selection

FracMan allows you to create a tracemap subset that contains only traces that pass a specified filter criteria. Simply right-click on the tracemap you wish to subdivide and hit “Create Data Subset”. You can then define a trace filter in the same way you define a fracture filter. You can also create a tracemap filter by clicking *Edit > Tracemap > Create Data Subset*. When you click **OK**, the tracemap subset will appear in the objects window as a child of the tracemap from which it was derived.

Compute Segment Orientation

Compute the XY projected trend of tracemap segments.



Disconnect Traces ELFEN Interface

In the field on rock mechanics and engineering, conventional numerical techniques are either a continuum or a discontinuum. For the continuum approach, the use of the finite element, finite difference, or boundary element method is based on the assumption that the rock mass behaves as a continuum medium. One of the fundamental limitations of the continuum approach is the inability to simulate the kinematic aspects of rock mass failure and fabric-joint interaction. To accurately simulate this behaviour by numerical modelling, the current combined FracMan – ELFEN permits a more realistic representation of the mechanical behaviour of discrete fracture systems. This involves realistic simulation and representation of fracture networks, either as individual entities or as a collective system of fracture sets, or a combination of both. By using an integrated finite element/discrete element–discrete fracture network approach it is possible to study the failure of rock masses in tension and compression, along both existing pre-existing fractures and through intact rock bridges, and incorporating complex kinematic mechanisms. The proposed modelling approach fully captures the anisotropic and inhomogeneous effects of natural jointing and is considered to be more realistic than methods relying solely on continuum or discontinuum representation.

We use the DFN capabilities of FracMan to provide an initial description of the naturally fractured rock mass, where the fractures generated within the DFN are key descriptors and are generated from available 1D/2D/3D geological, geophysical and geomechanical attributes. We are able to take tracemaps of the natural fracturing present and perform rudimentary

simplification to alleviate numerical issues associated with very close sub parallel fractures and extremely small fractures. These tracemaps are then directly imported into the ELFEN using the native GEM file format of ELFEN. The import of the fracture tracemap into ELFEN permits the manipulation and assignment of geometry, loadings, constraints, material properties, mesh assignments and solution data in the normal way.

Some of the key applications on which the combined FracMan – ELFEN simulation approach has been used include:

- Caveability assessments of deep underground block cave mines;
- Assessment of rock fragment stability around underground excavations;
- Slope stability of open pit excavations;
- Hydraulic fracture assessments for low permeability gas formations; and
- Prediction of surface observed subsidence from underground mineral extraction.

GEM Export

Tracemaps can now be exported to a .GEM file suitable for use by the ELFEN software. Simply select “GEM Files” in the “files of type” box when exporting tracemaps.

Point Data Object

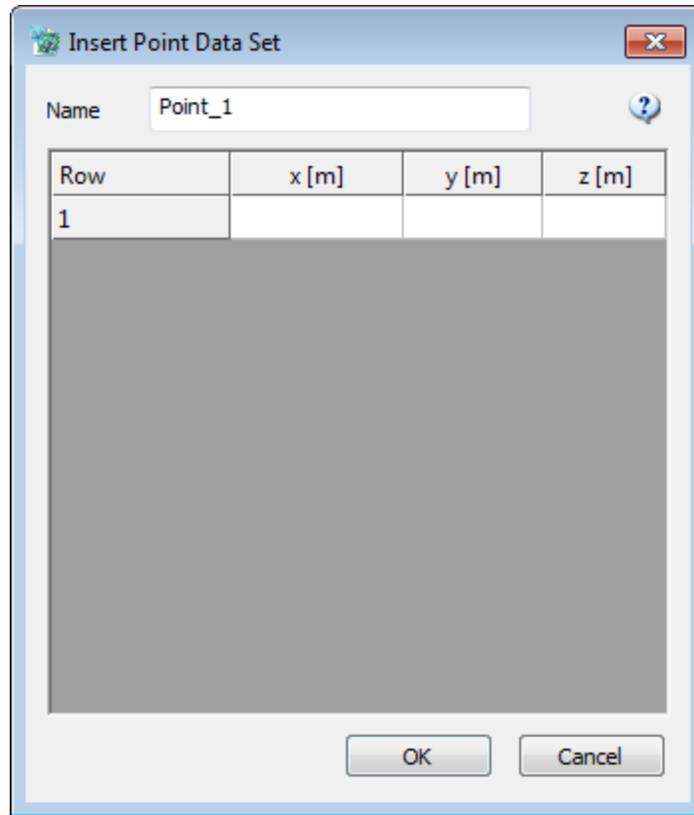
Point data object represent any location / value type data. Object locations, microseismic values – (real or simulated) and many other types can be stored in this type. Point data sets are draw as points – either as world unit sized spheres or pixel sized dots.

Importing / Exporting Point Data

FracMan supports a number of different file formats for importing and exporting point data. Import a point data set by selecting “Point Data” from the File -> Import menu. Select the input file type and press “Open”. Point data set objects support multiple properties per point and the “Space Delimited Column Data” import type allows you to specify the property name of each column. See [Appendix A FracMan Supported File Formats](#) for details on the file types supported for reading and writing point data objects.

Creating Point Data

By Selecting “Point Dataset” from the “Insert” menu you can create a new empty point data object. You can then manually enter points or paste from your favorite spreadsheet.

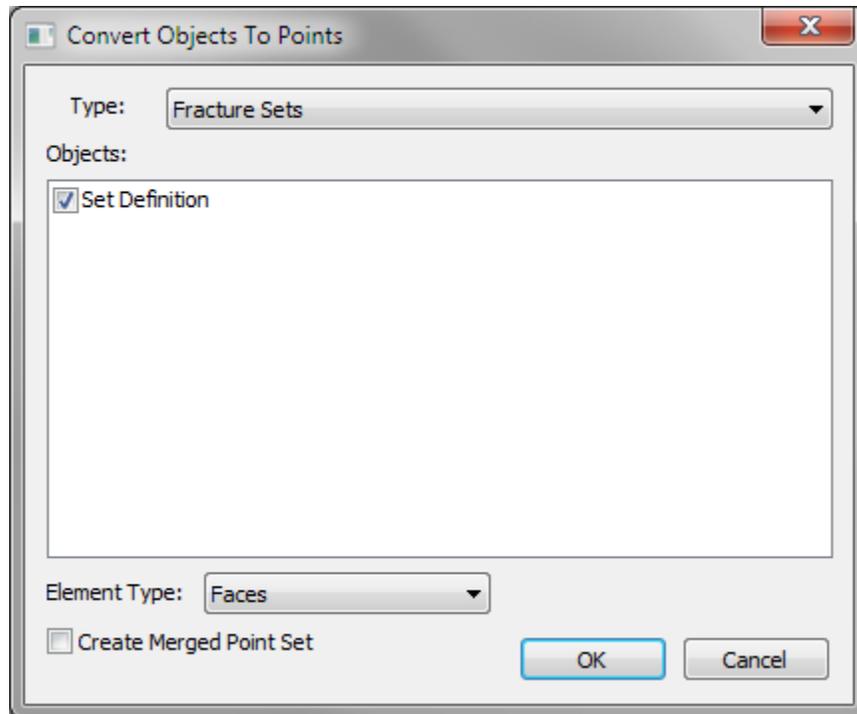


Visualizing Point Data

By default point data set objects are drawn as a set of pixel sized points – the value in the “Size” box of the attributes window is used as the number of pixels to draw each point. Although this is a very efficient you may also want to visualize the points in world units – check the “Draw Face Spheres” box in the attributes window and each point will be drawn as a sphere instead of a point. You can color the points by a property value by right clicking on the point data object and selecting the property from the visualize menu.

Converting an Object to Point Data

You can extract point data from other objects by selecting “Convert Objects to Points” from the Edit -> Point Data menu. Select the object type from the type pull down and then select one or more objects from the Objects list. Select the element type (Faces / Cells / Vertices etc) from the Element Type menu. If “Create Merged Point Set” is selected you will get a single point data object containing the points of all selected objects, otherwise you will get one point data set for each selected object.



Convert Selection To Points

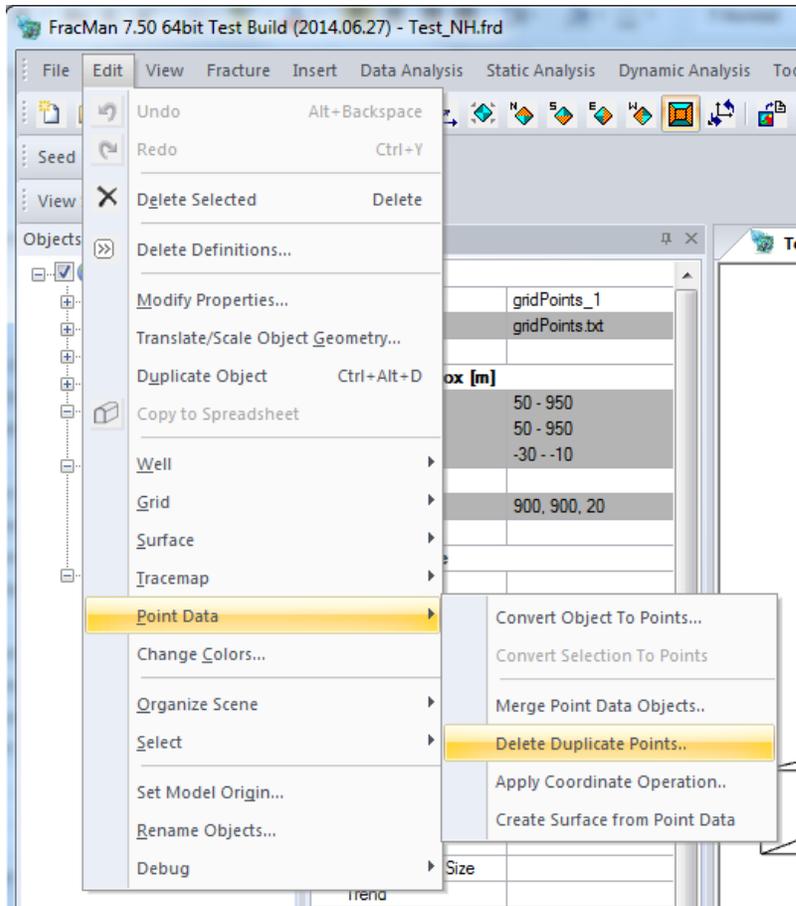
This function allows you to create a point data set from a set of selected elements (grid cells/surface triangles/fractures etc).

Merge Point Data Objects

Select multiple point data objects and merge them into a single object – the merged object will have the intersection of the properties of all the source objects.

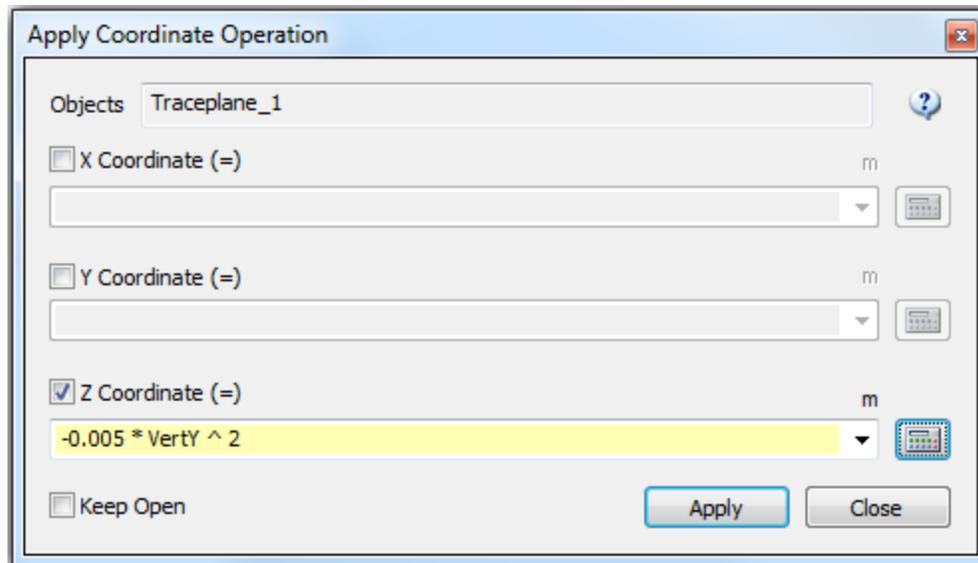
Delete Duplicate Points

Deletes points that occur at the exact same location. This is often useful for large point data sets. Select the Point Data object that you want to modify in the Objects window and then select Point Data | Delete Duplicate Points from the Edit menu.

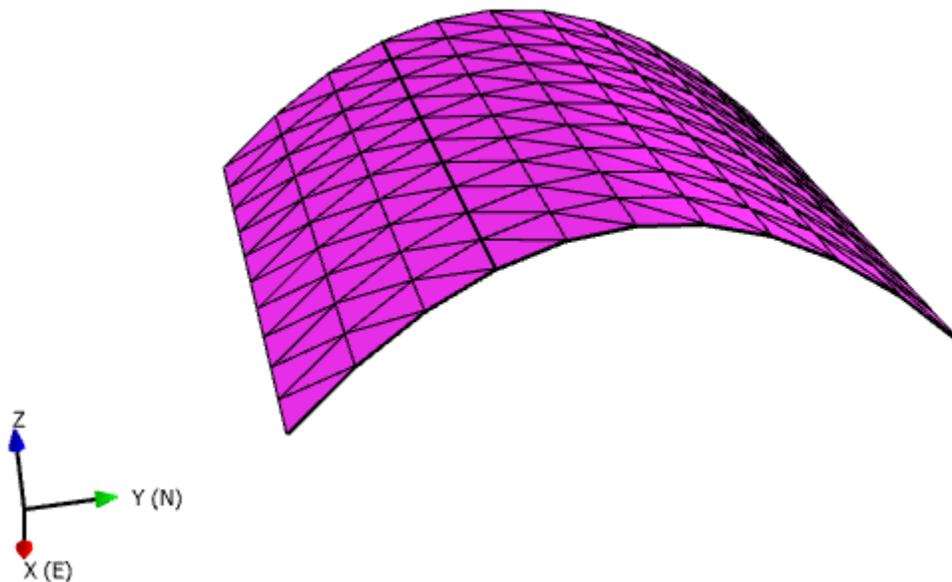


Apply Coordinate Operation

Use this function to manipulate the XYZ coordinates of one or more point data sets. This includes modifying the vertices of objects such as surfaces or traceplanes. To use, select the Point Data object(s) and then select Edit->Point Data->Apply Coordinate Operation. Use the function buttons to create the transformation desired. Use the Keep Open checkbox to keep the dialog active after clicking Apply so you can run more operations. This function provides more freedom to define your own transformations beyond what is provided in the Edit->[Translate/Scale Object Geometry](#) menu option.



Starting with a traceplane in the xy plane, centered at 0,0,0 that has been decimated to 10x10, apply the operation “-0.005 * VertY^2” in the Z Coordinate box to make the flat plane into something more interesting.



Create a Surface from Point Data

You can create a triangulated surface out of point data (should be semi-planar) by selecting one or more point data objects and selecting Edit->Point Data->Create Surface from Point Data. A new surface object will be created under the original point data object in the Objects window.

Creating a Subset of a Point Data Object

You can filter a set of point data by property value by selecting “Create Data Subset” and specifying filter criteria.

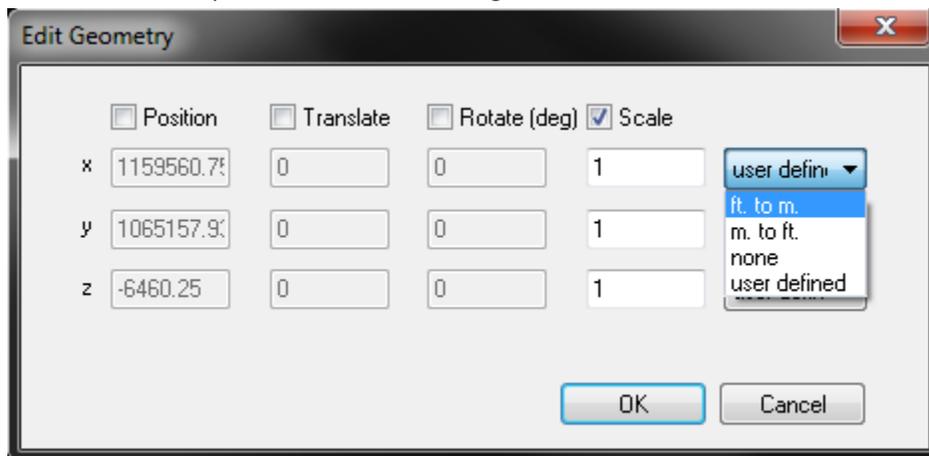
Visualizing a Random Subset

Large point data sets can often be difficult to see and can overwhelm many graphics cards. In addition to being able to toggle an entire set on or off, you can also set the object to only display a fraction of its points. Select the point data object in the objects bar. In the attributes window, enter a value 0-100 in the “Display Percent” box.

Converting Object Units

When objects are imported from external applications they may be in many different unit systems. The way to change an object’s units is to select the object and choose “Translate / Scale Object Geometry” from the “Edit” menu.

Check “Scale” and then enter the units conversion in the x, y, and z boxes, or select a standard units conversion from the pull down lists on the right.



Well logs and Well intervals are generally specified in measured depth, so conversions are not automatically applied when the parent well is converted. To these object types, select the objects and select “Offset Scale Intervals” or “Set Depth Scale / Offset” to apply a scale factor to the measured depth values.

Handling UTM Coordinates

In the past, high UTM coordinates caused problems with analyses that require high degrees of precision. In order to solve this and maintain accuracy in analyses at high UTM coordinates, FracMan has an internal “model origin”, allowing it to treat the objects in the model as if they were centered around (0,0,0). This is only an internal function and will not affect displayed or exported model coordinates.

If you wish to change the model origin, you may do so by selecting *Edit > Set Model Origin* on the menu bar. The current model origin’s UTM coordinates will be displayed in the grey boxes at the top of the dialog box. You may change the origin to:

- A custom point, whose UTM coordinates you must specify in the X, Y, and Z boxes on the right side of the dialog box,
- The point with UTM coordinates (0,0,0), which is FracMan’s default origin,

- The center point of all the objects currently contained in your model, which FracMan will automatically calculate and input for you,
- The point in your model whose UTM coordinates are smallest, or
- The UTM coordinates of a given point marker object or well.

You can change from UTM Coordinates to the local coordinates in the Model Scale box by checking the **XY Scale Local** option in the Attributes panel of the Scene object.

Generating Fractures

FracMan's most fundamental function is fracture generation.

There are six basic fracture generation algorithms:

- [Geometric](#)
- [Geocellular](#)
- [Geologic](#)
- [Tracemap](#)
- [Stratigraphic](#)
- [Stress](#)

Each of these uses a different basis for its stochastic processes, yielding a distinct type of fracture set.

The Theory Behind Fracture Generation

Fracture generation depends on defining basic features of the fractures:

- [Location](#)
- [Orientation](#)
- [Intensity](#)
- [Size](#)
- [Elongation](#)
- [Termination](#)
- [Fracture Properties](#)

Based on your definition, FracMan allows you to define exactly how your fractures are constructed.

Fracture Location

FracMan has three separate fracture location algorithms:

- [Enhanced Baecher](#)
- [Nearest Neighbor](#)
- [Levy Lee](#)

The Enhanced Baecher Model

The Baecher model (Baecher, Lanney and Einstein, 1978) was one of the first well-characterized discrete fracture models. In this model, the fracture centers are located uniformly in space, and, using a Poisson process, the fractures are generated as disks with a given radius and orientation. The Enhanced Baecher model extends the Baecher model by providing a provision for fracture

terminations and more general fracture shapes. The Enhanced Baecher model utilizes fracture shapes initially generated as polygons with three to sixteen sides. These polygons can be equilateral (aspect ratio of one) or elongate, with the aspect ratio (major to minor axis size and orientation) defined by the user.

In the Enhanced Baecher Model, termination is specified as the termination probability $P[T/I]$ (called *Termination Percentage in the generation wizard*), which is the probability that a fracture will terminate at a pre-existing fracture, given that it intersects a fracture. In the Enhanced Baecher Model, all fractures are generated from center locations distributed uniformly in space. For every fracture, FracMan checks whether the fracture intersects a pre-existing fracture. If it does, FracMan utilizes the termination probability to determine whether to truncate the fracture such that it terminates at the intersection. If the fracture is terminated at that intersection, the portion of the fracture beyond the intersection is discarded. This can result in some distortion in the size distribution for that fracture set. Note that in this model, fracture termination at intersections only occurs where fracture intersection occurs. If no fracture intersections occur for the fractures generated from uniformly located centers, no fracture terminations at intersections will be defined, even if $P[T/I]$ is set at 100%.

The Nearest Neighbor Model

The nearest neighbor model is a simple, non-stationary model in which fracture intensity P_{32} decreases exponentially with distance from "major features" identified by the user. The intensity P_{32} at any point in space is defined by:

$$P_{32}(x) = Ce^{-bd_{x,f}}$$

where C and b are constants and $d_{x,f}$ is the distance between location x and the nearest major feature. The user inputs the mean distance, which (effectively) is the mean value of an exponential distribution. The mean value is used to calculate the rate parameter (i.e. the decay rate) b as:

$$\text{Mean} = b^{-1}$$

Note: For fracture set generation using the Nearest Neighbor model to occur, one or more fracture set objects must be checked "Visible" in the Objects Pane. This is so that, for multiple fracture sets in a FracMan model, nearest neighbor generation can be limited to surrounding certain target fracture sets.

The Levy-Lee Fractal Model

Fractal patterns may be either "self-similar" (topologically identical at different scales) or "self-affine" (topologically similar but anisotropically distorted at different scale). Fractal fracture patterns can be generated in three ways:

1. Utilize a recursive generation scheme which produces fracture patterns at one scale, then superposes them at different scales (with self-affine distortion if appropriate) to directly produce a fractal pattern;
2. Generate fractures according to a process such as "Levy-Flight" which has been shown mathematically to produce fractal patterns; and
3. Generate fractures using non-fractal processes, and then test the resulting patterns to determine whether the resulting pattern is fractal.

The Levy-Lee fractal fracture model utilizes the second of the above approaches and is based upon "Levy Flight" (Mandelbrot, 1985). The Levy Flight process is a type of random walk, for which the length L of each step is given by the probability and cumulative distribution functions:

$$\text{pdf}(L) = 0 \text{ when } L \leq L_s$$

$$\text{pdf}(L) = \frac{D}{L} \left(\frac{L_s}{L}\right)^D \text{ when } L > L_s$$

$$\text{cdf}(L) = 0 \text{ when } L \leq L_s$$

$$\text{cdf}(L) = 1 - \left(\frac{L_s}{L}\right)^D \text{ when } L > L_s$$

where D is the fractal mass dimension of the point field of fracture centers, L is the distance from one fracture center to the next generated fracture center, and L_s is the minimal hop distance in the generation sequence. For $D=0$, the distribution of the step length is uniform, such that there is no clustering or heterogeneity. For large D , there is a very low probability of large steps and therefore fractures are formed close to each other in concentrated clusters. Levy Flight is a one-dimensional process for points on a line. In FracMan, this is extended to two dimensions (fracture traces on a trace plane) and three dimensions (fractures in space).

The Levy-Lee fractal model requires derivation of a dimension D for the three-dimensional process of fracture centers from the two-dimensional pattern of fracture traces. La Pointe (2002, "Derivation of parent fracture population statistics from trace length measurements of fractal fracture populations", Int. J. Rock Mech. Min. Sci., Vol. 39, pp. 381-388) presents a derivation to convert the fractal mass dimension of points on a plane to the dimension of points in three-dimensional space required for fracture generation. This derivation indicates that the three-dimensional fractal process should be generated using a fractal dimension one greater than the dimension calculated for the two-dimensional trace plane.

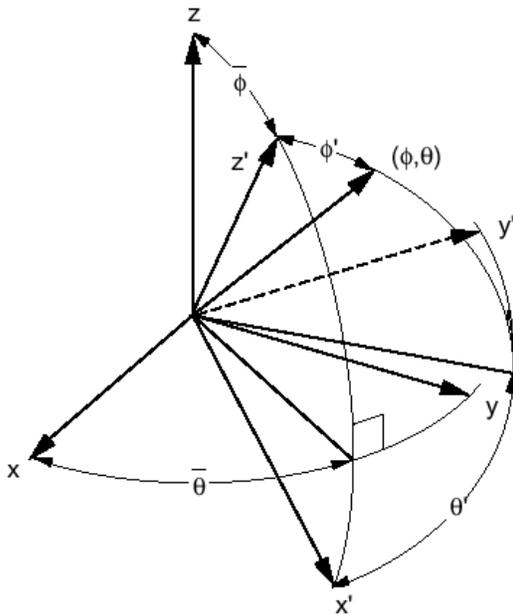
Orientation

FracMan supports a number of different orientation distributions. The current version of FracMan provides five types of probability distributions for directional variables:

- [Constant](#)
- [Fisher](#)

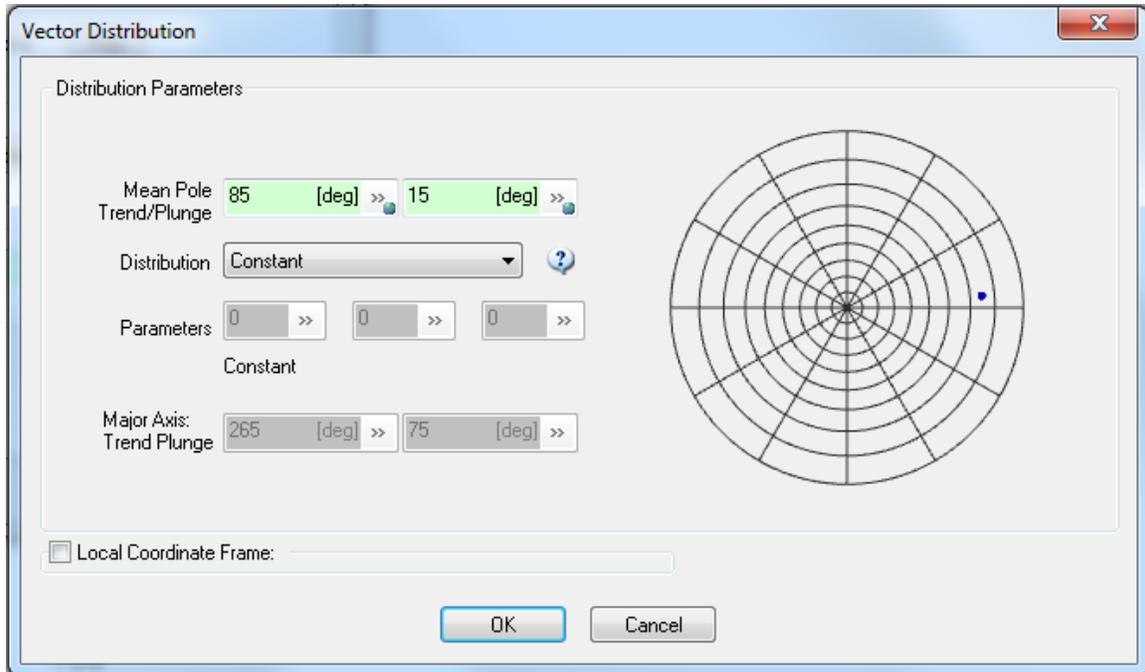
- [Bivariate Fisher](#)
- [Elliptical Fisher](#)
- [Bivariate Normal](#)
- [Bivariate Bingham](#)
- [Bootstrap](#)

These distributions are stated in terms of their probability density functions, $f(\varphi',\theta')$, for a variation (φ',θ') about the mean direction (φ,θ) . The actual direction (φ,θ) chosen from a directional distribution is the composite of (φ',θ') and (φ,θ) .



Constant Distribution

Select a Constant Distribution when all the fractures in the set should have the same trend and plunge.



Fisher Distribution

The Univariate Fisher Distribution is the analog of the Normal (Gaussian) distribution on a sphere, and is theoretically justified when a sum of multiple, uncorrelated variations in rock properties and stress conditions cause the variation in fracture orientation. It is relatively easy to fit and easy to explain. However, its major drawback is that it requires a symmetrical variation of orientation around the mean pole. The variability of strike is forced to be equal to the variability of dip, such that the variability of orientation plots as a circular cluster on an equal area stereonet.

The Fisher Distribution is defined by the probability density function (Mardia, 1972):

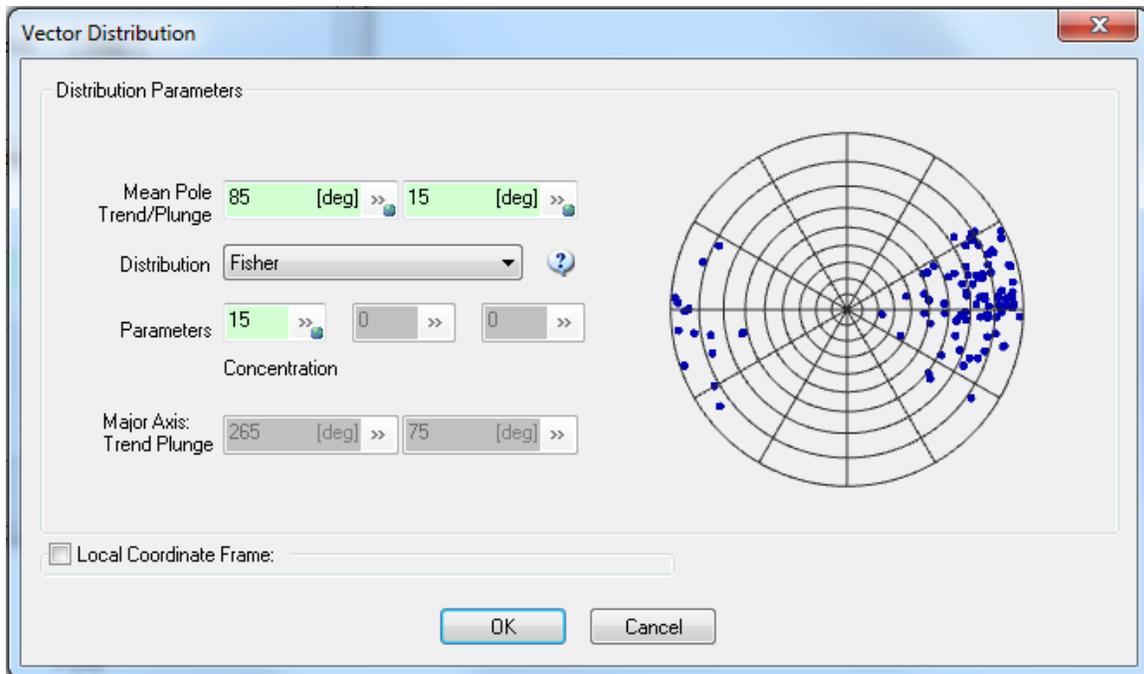
$$f(\phi', \phi') = \frac{\kappa \sin \phi' e^{\kappa \cos \phi'}}{2\pi(e^\kappa - 1)} \quad 0 \leq \phi' \leq \pi, 0 \leq \phi' \leq 2\pi \quad (1)$$

where κ is the distribution parameter, concentration (specified by the user) and ϕ' is the deviation angle from the mean pole. This distribution is unimodal and symmetric about the ϕ' axis. Increasing κ produces a distribution more concentrated around the ϕ' axis.

The Fisher concentration parameter κ is generally found from the approximate equation

$$\kappa \approx \frac{N_F}{N_F - |R|} \quad (2)$$

where $|R|$ is the magnitude of the vector sum of the unit vectors for orientation, and N_F is the number of fractures. This approximation is only valid for κ greater than 5. Uniform dispersion corresponds to $\kappa = 0$.



Bivariate Fisher Distribution

The Bivariate Fisher Distribution is defined by the probability density function (Dershowitz, 1979):

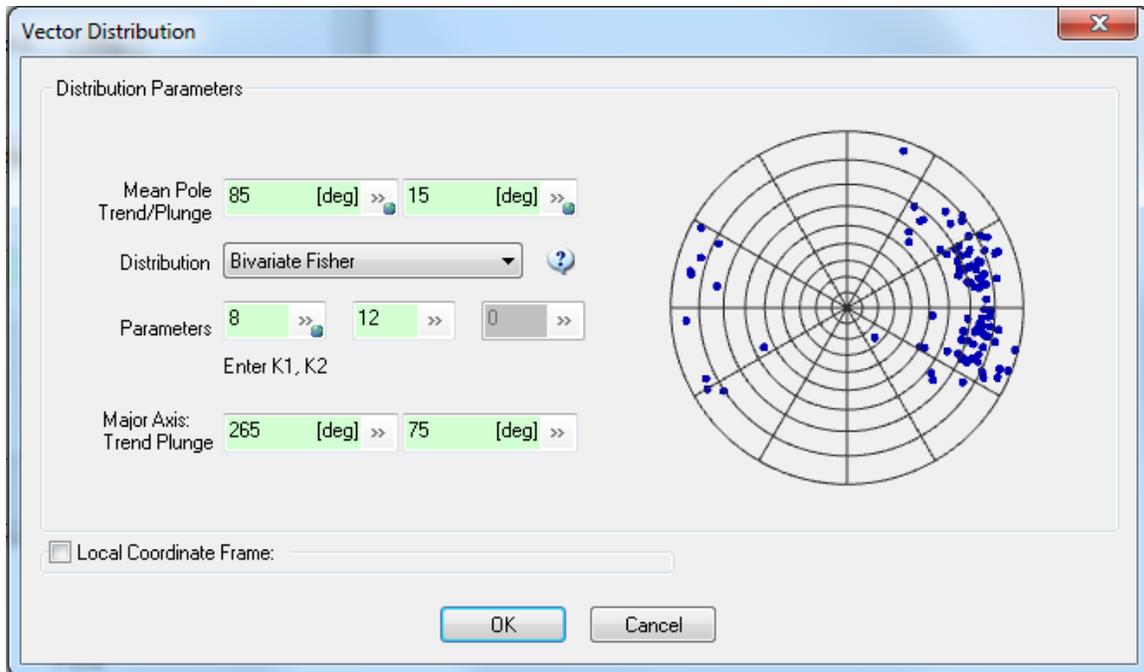
$$f(\phi, \theta) = C^{-1} \sin \phi' \exp[(\kappa_1 \sin^2 \theta' + \kappa_2 \cos^2 \theta') \cos \phi'], \quad (3)$$

$$0 \leq \phi' \leq \pi/2, 0 \leq \theta' \leq 2\pi,$$

where C is the normalizing constant:

$$C = \int_0^{2\pi} \int_0^{\pi/2} \sin \phi' \exp[(\kappa_1 \sin^2 \theta' + \kappa_2 \cos^2 \theta') \cos \phi'] d\phi' d\theta',$$

and where κ_1 and κ_2 are the distribution parameters (specified by the user). When $\kappa_1 = \kappa_2$, this distribution reduces to the Univariate Fisher distribution. When $\kappa_1 > \kappa_2$, the distribution is more concentrated in the $\theta = 0^\circ$ and $\theta = 180^\circ$ directions, and more scattered in the $\theta = 90^\circ$ and $\theta = 270^\circ$ directions. When $\kappa_1 < \kappa_2$, the opposite is true.



Elliptical Fisher Distribution

The Elliptical Fisher (EF) Distribution provides a powerful, flexible way to analyze and model fracture sets having different amounts of variability in fracture strike and dip that plot as an ellipse on an equal area stereonet. The EF distribution has four parameters fit automatically by FracMan’s orientation module, *Set Identification*. The first two parameters are the same as for a Fisher distribution: the set mean pole (trend and plunge (Θ, Φ)), and Fisher Concentration¹ parameter (κ_1). The direction in which orientation dispersion is greatest (trend and plunge $[\Theta, \Phi]$), and the ratio (κ_2) of concentration in the directions of greatest-to-least orientation dispersion are two additional parameters used by EF Distribution.

This distribution works by transforming orientations—using the Eigenvalues and Eigenvectors of the orientation data—so they are symmetrical about the mean pole, with equal variation in all directions. When the orientations are transformed to be symmetrical about the mean pole, the Fisher distribution can be fit to obtain a value of the Fisher concentration parameter κ in the transformed space, and then fitting a Fisher distribution to the transformed data.

The probability density function of the Elliptical Fisher distribution is:

¹ In FracMan 7.5, the Fisher “Dispersion” parameter κ , is renamed as the Fisher “Concentration” parameter – since the higher the value of κ , the greater the concentration is.

$$f_{EF}(\theta, \varphi) = f_{\theta} f_{\varphi} = \frac{R\kappa \sin \varphi e^{\kappa \cos \varphi}}{2\pi(R^2 \cos^2 \theta + \sin^2 \theta)(e^{\kappa} - 1)}, 0 \leq \theta \leq 2\pi, 0 \leq \varphi \leq \frac{\pi}{2}$$

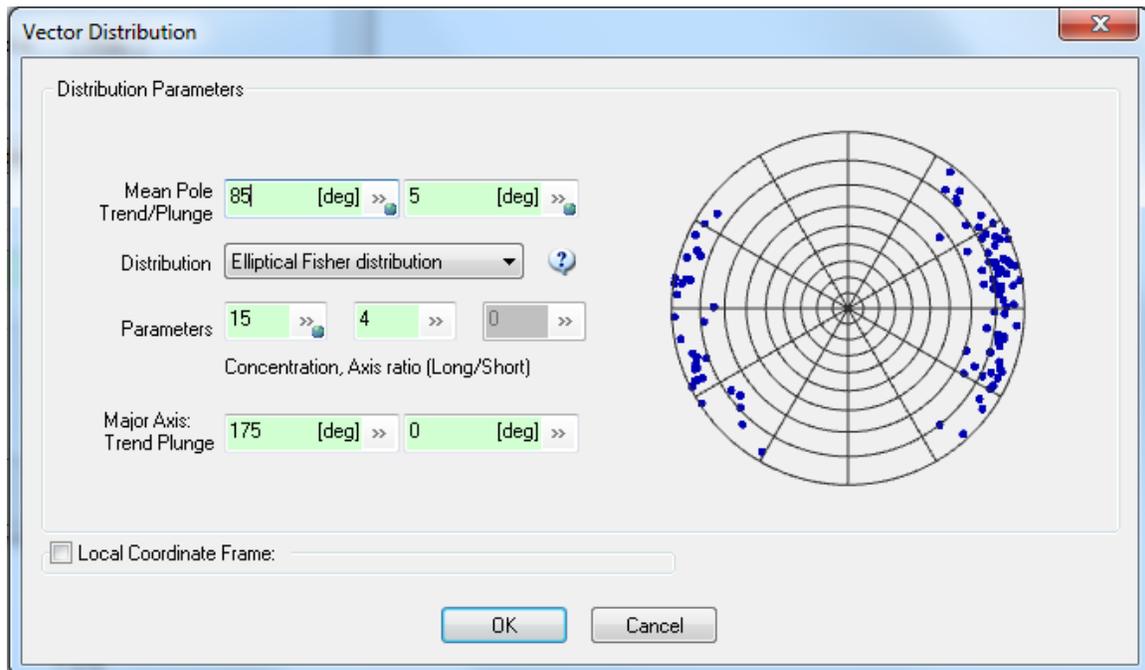
Where κ is the Fisher concentration parameter for the transformed data, θ is the mean pole trend (clockwise from north), φ is the mean pole plunge (down from horizontal), and R is the ellipse axial ratio.

When to use Elliptical Fisher distribution in FracMan

In general, use EF for fracture orientation data which plots as an ellipse on an equal area stereonet. For such data, also think about the geologic controls on the fracture system. A few important things to consider:

- What mechanisms formed the fractures and why would the data plot as an ellipse?
- Are the formations in question folded or faulted?
- Are there strong lithological or mechanical heterogeneities that need to be accounted for?
- Might fracture orientations be better modeled relative to a local coordinate system rather than using FracMan's global coordinate system?

If you contour the data on a stereonet and observe a much larger variation in strike than dip—particularly when plotted in a local coordinate system like in the figure below—you will want to use the EF parameters from a Set Identification analysis to characterize the set.



Bivariate Normal Distribution

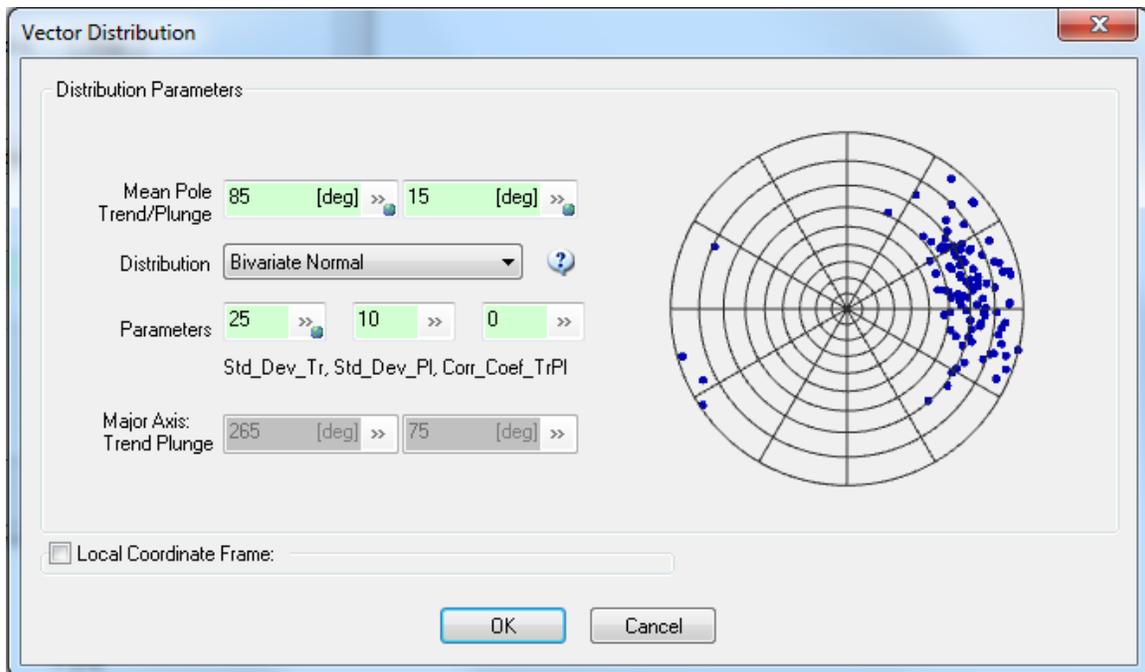
To account for a distribution for fracture orientations where the variation of strike is greater than the variability of dip, FracMan includes several possibilities. Bivariate Normal Distribution can be used for very small (e.g., $<\pm 4^\circ$) variations in orientation, such that the Euclidean space Bivariate Normal Distribution provides a tangent approximation.

The Bivariate Normal Distribution for orientation is defined by the probability density function:

$$f(\phi, \theta) = \frac{1}{2\pi \phi_\sigma \theta_\sigma \sqrt{1 - \rho^2}} \exp \left\{ -\frac{1}{2(1 - \rho^2)} \left[\left(\frac{\phi - \bar{\phi}}{\phi_\sigma} \right)^2 - 2\rho \frac{(\phi - \bar{\phi})(\theta - \bar{\theta})}{\theta_\sigma \phi_\sigma} + \left(\frac{\theta - \bar{\theta}}{\theta_\sigma} \right)^2 \right] \right\}$$

(4)

where $(\phi_\sigma, \theta_\sigma)$ are the standard deviations of dip and dip direction, and r is the correlation coefficient. Decreasing ϕ_σ and θ_σ produces a distribution more concentrated around the mean orientation. In the current implementation of this distribution, the correlation coefficient ρ can only have the values 0 or 1. Intermediate values require the solution of an eigenvalue problem and are too computationally intensive to be useful.



Bivariate Bingham Distribution

The Bivariate Bingham distribution is diagnostic of girdle distributions, such as those found by plotting data from multiple points in a fold structure. When data is found to fit a Bingham Distribution, users should consider using a local coordinate system to account for the variation across the fold.

The Bivariate Bingham distribution is defined by the probability density function (Dershowitz, 1979):

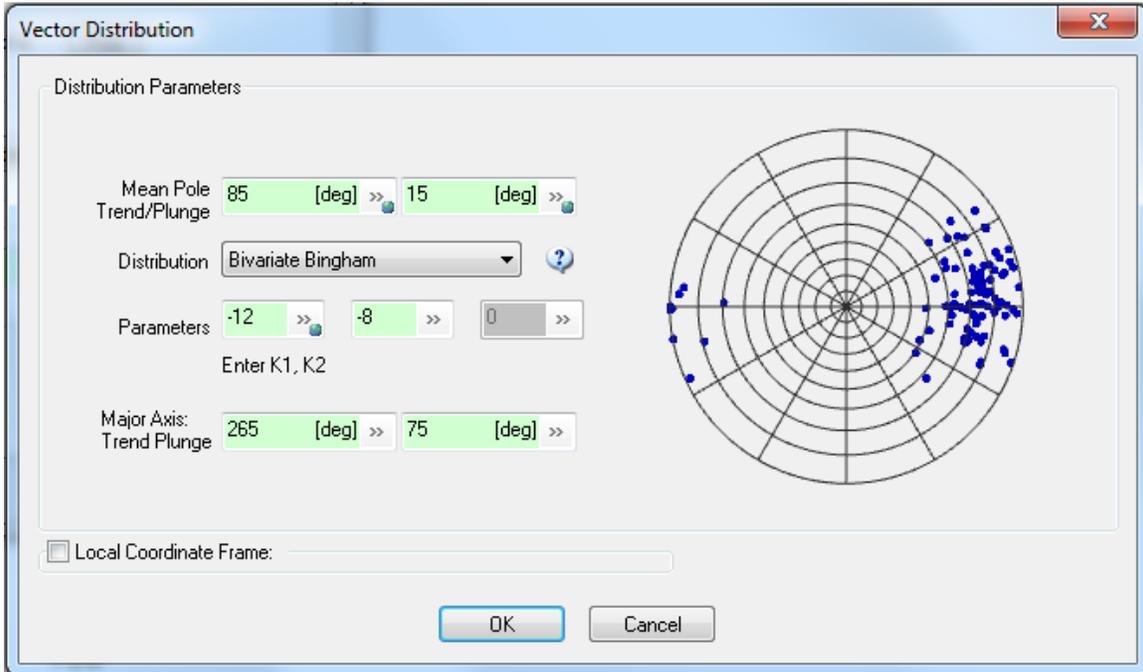
$$f(\phi, \theta) = C^{-1} \sin \phi \exp[\kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta] \sin^2 \phi, \quad (5)$$

$$0 \leq \phi \leq \pi, \quad 0 \leq \theta \leq 2\pi,$$

where C is the normalizing constant:

$$C = \int_0^{2\pi} \int_0^{\pi} \sin \phi \exp[(\kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta) \sin^2 \phi] d\phi d\theta,$$

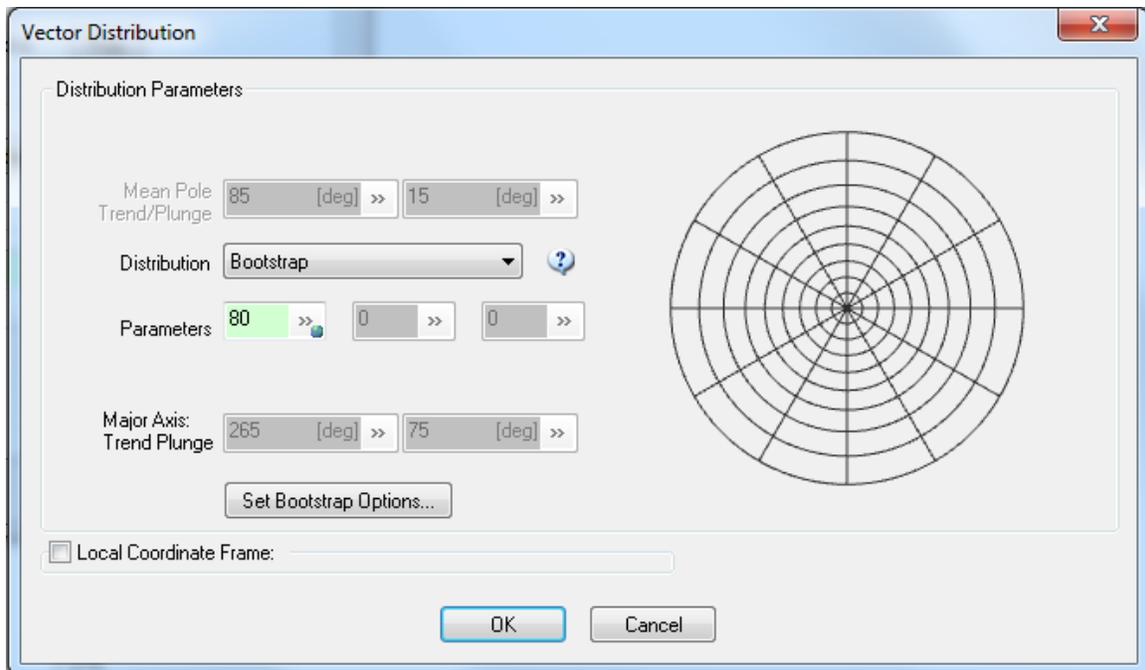
and where κ_1 and κ_2 are the distribution parameters (specified by the user). When $\kappa_1 = \kappa_2$, this is a girdle distribution, with the region of highest concentration lying around the equator $\phi = 90^\circ$. When $\kappa_1 > \kappa_2$, this is a bimodal distribution with the regions of highest concentration lying at $(\phi = 90^\circ, \theta = 0^\circ)$ and $(\phi = 90^\circ, \theta = 180^\circ)$; when $\kappa_1 < \kappa_2$, this is a bimodal distribution with the regions of highest concentration lying at $(\phi = 90^\circ, \theta = 90^\circ)$ and $(\phi = 90^\circ, \theta = 270^\circ)$. Thus this distribution will produce a population of directions predominantly normal to the specified mean direction (ϕ, θ) .



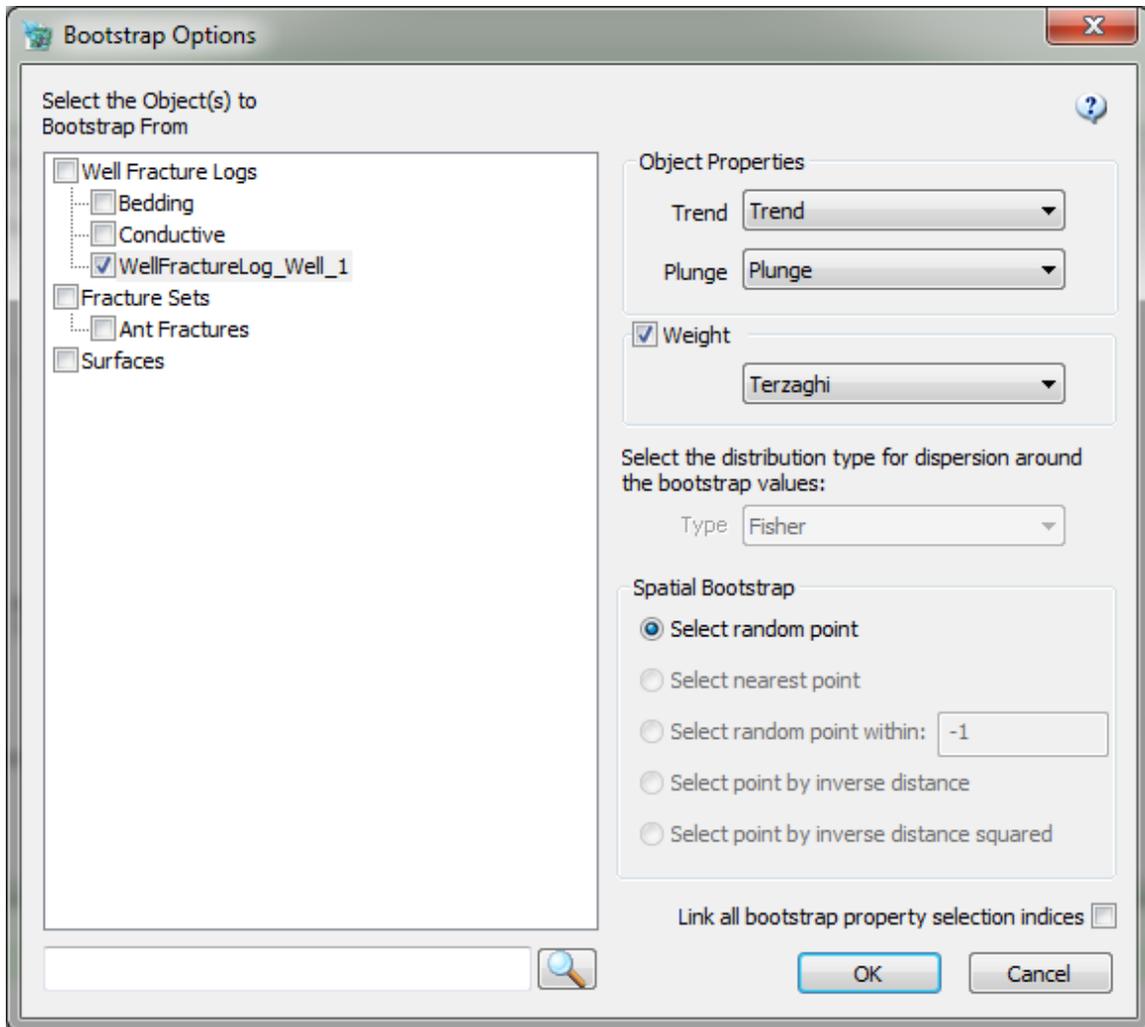
Orientation Bootstrap

Bootstrapping allows you to use orientation data in the model for your fracture generation.

If you select “Bootstrap” distribution in the [Vector Distribution dialog](#), you must click on the **Set Bootstrap Options** button that appears at the bottom of the dialog box. This will open another box asking you to specify which datasets you wish to bootstrap from.



Note: bootstrapping for a vector distribution does not use the specific values from which you bootstrap, but rather uses the existing data to determine parameters for a Fisher distribution – you can match these values closely by entering a large concentration value in the Vector Distribution dialog **Parameters** box.



First check all the objects from which you wish to bootstrap.

- **Object Properties:** Select the specific properties containing your trend and plunge data values.
- **Weight:** Optionally set a weight on the bootstrap selection (often the Terzaghi/Mauldon correction factor).
- **Spatial Bootstrap:** This allows you to specify (for objects that support it) whether to pick a point at random, to select a point at random within a radius, to select the nearest point, to select a random point weighted by distance, or to select a random point weighted by distance squared.
- **Link all bootstrap property selection indices:** When this is checked, FracMan links all bootstrapped properties for the same fracture, such that each individual fracture takes all of its bootstrapped properties from the same source (i.e., another individual fracture). This is critical in the GeoCellular model where the trend/plunge parameters are defined individually.

Click the **OK** button to return to the Vector Distribution window.

Probability Distributions for Scalar Values

FracMan supports a number of different probability distributions for defining scalar values.

These distributions can be used to specify size, aspect ratio as well as fracture properties such as Aperture.

Uniform Distribution

The uniform distribution is specified by the probability density function:

$$f(x) = 1/2a, \quad \bar{x} - a \leq x \leq \bar{x} + a, \quad \text{otherwise}$$
$$f(x) = 0$$

where a is the maximum deviation (specified by the user) and \bar{x} is the mean value (specified by the user). This distribution is also known as the "rectangular" distribution.

Exponential Distribution

The exponential distribution is specified by the probability density function:

$$f(x) = \lambda e^{-\lambda x}$$

and has a mean value $\bar{x} = 1/\lambda$. Thus the exponential distribution is completely specified by its mean value \bar{x} .

The truncated exponential distribution has two additional parameters (x_-, x_+), the minimum and maximum values. Although FracMan uses the mean and standard deviation values above (i.e., for the un-truncated distribution), the actual mean \bar{x}' and standard deviation x'_σ of truncated exponential distributions are:

$$\bar{x}' = \frac{x_+ e^{-x_+/\bar{x}} - x_- e^{-x_-/\bar{x}}}{e^{-x_+/\bar{x}} - e^{-x_-/\bar{x}}} + \bar{x}$$
$$x'_\sigma = \left\{ \frac{(x_+^2 + 2\bar{x}x_+)e^{-x_+/\bar{x}} - (x_-^2 + 2\bar{x}x_-)e^{-x_-/\bar{x}}}{e^{-x_+/\bar{x}} - e^{-x_-/\bar{x}}} + 2\bar{x}^2 - (\bar{x}')^2 \right\}^{1/2}$$

Normal Distribution

The normal distribution is specified by the probability density function:

$$f(x) = \exp\{- (x - \bar{x})^2 / 2x_\sigma^2\} / \sqrt{(2\pi x_\sigma^2)}$$

where \bar{x} is the mean value of x , and x_σ is the standard deviation of x .

The truncated normal distribution has two additional parameters (x_-, x_+), the minimum and maximum values. Although FracMan uses the mean and standard deviation values above (i.e., for the un-truncated distribution), the actual mean \bar{x}' of the truncated normal distribution is:

$$\bar{x}' = \bar{x} + \sqrt{\frac{2}{\pi}} x_{\sigma} \frac{e^{-y_-^2} - e^{-y_+^2}}{\text{erf}(y_+) - \text{erf}(y_-)}$$

where $y_- = \frac{x_- - \bar{x}}{\sqrt{2} x_{\sigma}}$, $y_+ = \frac{x_+ - \bar{x}}{\sqrt{2} x_{\sigma}}$, and erf() is the error function.

Lognormal Distribution

The lognormal distribution is specified by the probability density function:

$$f(x) = \frac{1}{x \ln(10) y_{\sigma} \sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{\log x - \bar{y}}{y_{\sigma}} \right)^2 \right\}$$

where \bar{y} and y_{σ} are the mean and standard deviation in log10 space.

The truncated lognormal distribution has two additional parameters (x_-, x_+), the minimum and maximum values. Although FracMan uses the mean and standard deviation values above (i.e., for the un-truncated distribution), the actual mean \bar{x}' of the truncated lognormal distribution is:

$$\bar{x}' = \log \bar{x} + \sqrt{\frac{2}{\pi}} \log x_{\sigma} \frac{e^{-y_-^2} - e^{-y_+^2}}{\text{erf}(y_+) - \text{erf}(y_-)}$$

where $y_- = \frac{\log x_- - \log \bar{x}}{\sqrt{2} \log x_{\sigma}}$, $y_+ = \frac{\log x_+ - \log \bar{x}}{\sqrt{2} \log x_{\sigma}}$

Normal of Log Distribution

It is often desirable to specify the mean and standard deviation of a lognormal distribution in log10 space. This can be done using the “Normal of Log” distribution in FracMan.

Given the mean \bar{y} and standard deviation y_{σ} in log10 space, the mean \bar{x} and standard deviation x_{σ} in arithmetic space are:

$$\bar{x} = e^{(c\bar{y} + 1/2(cy_{\sigma})^2)}$$

$$x_{\sigma} = \bar{x} (e^{(cy_{\sigma})^2} - 1)^{1/2}$$

where c is the natural log of 10 (2.303).

Given the mean \bar{x} and standard deviation x_{σ} in arithmetic space, the mean \bar{y} and standard deviation y_{σ} in log10 space are:

$$\bar{y} = \frac{1}{c} (\ln(\bar{x}) - 1/2 (cy_\sigma)^2)$$

$$y_\sigma = \frac{1}{c} \left(\ln \left(1 + \left(\frac{x_\sigma}{\bar{x}} \right)^2 \right) \right)^{1/2}$$

Power Law Distribution

The version of the power law distribution known as the Pareto distribution can be useful for describing the distribution of fracture size, which can exhibit scale independent ("fractal") behavior. The power law distribution is described by two parameters, the minimum value x_{min} , and the exponent D :

$$\text{Probability Distribution Function: } f_x(x) = \frac{D(x_{min})^D}{x^{D+1}} \text{ when } x \geq x_{min}, D > 0 \quad (1)$$

$$\text{Cumulative Distribution Function: } F_x(x) = 1 - \left(\frac{x_{min}}{x} \right)^D \text{ when } x \geq x_{min}, D > 0 \quad (2)$$

The mean is only defined if the exponent D exceeds 1 and power law distributions have a finite variance only when the exponent D exceeds 2. The mean \bar{x} and standard deviation x_σ are related to x_{min} and D by:

$$\bar{x} = \left(\frac{D}{D-1} \right) x_{min}, \text{ when } D > 1, \text{ otherwise} \quad (3)$$

$$\bar{x} = \infty, \text{ when } 0 < D \leq 1$$

$$x_\sigma = \left(\frac{D}{D-2} \right)^{1/2} \left(\frac{x_{min}}{D-1} \right), \text{ when } D > 2 \quad (4)$$

If \bar{x} and x_σ are known, Equations 3 and 4 can be used to calculate x_{min} and D :

$$x_{min} = \left(\frac{D-1}{D} \right) \bar{x}, \text{ when } D > 1 \quad (5)$$

$$D = 1 + \left(1 + \left(\frac{\bar{x}}{x_\sigma} \right)^2 \right)^{1/2}, \text{ when } D > 2 \quad (6)$$

The truncated power law distribution has two additional parameters (x_-, x_+), the minimum and maximum values. The truncated minimum value, x_- , should not be smaller than the distribution

minimum value, x_{min} . The truncated mean \bar{x}' and standard deviation x'_{σ} are related to x_+ , x_- , x_{min} and D by:

$$\bar{x}' = \left(\frac{D}{D-1} \right) x_{min} \left\{ \left(\frac{x_{min}}{x_-} \right)^{D-1} - \left(\frac{x_{min}}{x_+} \right)^{D-1} \right\}, \text{ when } D > 1, x_- \geq x_{min} \quad (7)$$

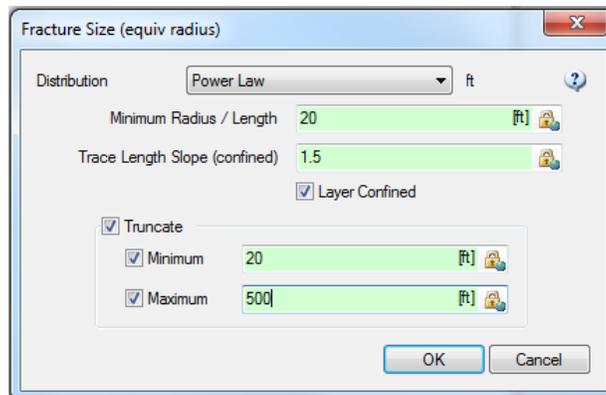
$$\bar{x}' = \infty, \text{ when } 0 < D \leq 1$$

$$x'_{\sigma} = \left\{ \left(\frac{D}{D-2} \right) x_{min}^2 \left[\left(\frac{x_{min}}{x_-} \right)^{D-2} - \left(\frac{x_{min}}{x_+} \right)^{D-2} \right] - (\bar{x}')^2 \right\}^{\frac{1}{2}}, \text{ when } D > 2, x_- \geq x_{min} \quad (8)$$

$$x'_{\sigma} = \infty, \text{ when } 0 < D \leq 2$$

Note that in FracMan version 7.4 and earlier, the exponent b was used for the power law distribution instead of the exponent D. The relationship between the two is that $D = b - 1$.

In the FracMan 7.5 interface, the exponent for the power law used to define fracture size is given by entering a Trace Length Slope and checking whether or not it is layer confined or not. This is provided as a convenience as many users estimate fracture size populations based on power law fits of fracture trace length data.



The relationship between the Trace Length Slope in unconfined layers, TLS, and D is:

$$D = TLS + 1$$

When the Trace Length Slope is calculated from a confined layer, then D is equal to the TLS. See the Manual section on [Working with Charts -> Power Law Plot](#) for details.

Weibull Distribution

The Weibull distribution is frequently used in the geosciences to describe the distribution of particle sizes based on what is called the 'weakest link' model. This implies that the strength of a material (such as rock) with flaws in it decreases with increasing sample size, as the probability of encountering a weak flaw increases with increasing volume. It is useful because, by changing the shape parameter, you can effectively model other distributions (i.e. power-law / exponential

/ lognormal). With respect to fractures, it has been suggested that fracture spacing can be best modeled as a Weibull distribution *if* the original flaws in the host rock (the fracture initiation points) are random (Poisson point process) in space. In FracMan, the Weibull is likely to be useful in the spatial model, looking at the spacing of fracture intersections along a borehole, and as a potential model for the distribution of P_{32} to be tested.

The probability density function of the Weibull distribution is:

$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0, \end{cases}$$

Where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution.

Gamma Distribution

The gamma distribution is appropriate for describing the uncertainty in the Poisson process intensity parameter N . It is therefore used for uncertainty in fracture intensity measures P_{31} (n/volume), P_{32} (area / volume), and P_{33} (volume / volume).

The probability density function for the gamma distribution is:

$$F_x(x) = \frac{e^{-ax} (ax)^{r-1}}{(r-1)!} a$$

where a and r are related to the mean and standard deviation by $x=r/a$.

Poisson Distribution

Poisson distribution (Benjamin and Cornell, 1970) is:

$$f_n(n) = \frac{\bar{n}^n e^{-\bar{n}}}{n!} \quad (3-15)$$

where \bar{n} is the Poisson process rate, which is equal to the expected value of n . The conductive fracture frequency is given by $f_c = n/L_i$, where L_i = the length of the test zone.

Intensity Theory

Fracture intensity can mean different things to different people. It can be intensity, density, frequency or more. A wide variety of measures are available for quantifying the intensity of fractures. FracMan allows specification of intensity in terms of:

- [Areal intensity \(\$P_{32}\$ \)](#)
- [Volume intensity \(\$P_{33}\$ \)](#)
- [Number of fractures \(NF\)](#)
- [Lineal Intensity \(\$P_{10}\$ \)](#)

A note about intensity “P_{xy}” notation: The two subscripts refer to the dimension of the sampling space and dimension of sample measure (respectively), so that ‘3’ is volume, ‘2’ is area, ‘1’ is length, and ‘0’ is count.

		Dimension of Measurement				
		0	1	2	3	
Dimension of Sample	1	P10 <i>No of fractures per unit length of borehole</i>	P11 <i>Length of fractures per unit length</i>			Linear Measures
	2	P20 <i>No of fractures per unit area</i>	P21 <i>Length of fractures per unit area</i>	P22 <i>Area of fractures per area</i>		Areal Measures
	3	P30 <i>No of fractures per unit volume</i>		P32 <i>Area of fractures per unit volume</i>	P33 <i>Volume of fractures per unit volume</i>	Volumetric Measures
		Density		Intensity	Porosity	

All of these measures depend on the distribution of fracture size. The area is also related to the distribution of spacing between fractures encountered in a well segment, which is a more common measure of fracture density.

Areal Intensity (P₃₂)

The preferred intensity measure for fracture intensity is P₃₂, since it is invariant with respect to the distribution of fracture size.

P₃₂ is defined as the total area of fractures per unit volume (Dershowitz, 1984):

$$P_{32} = A_f / V_t$$

where:

A_f = Total area of fractures and

V_t = Total Volume.

P_{32} is dimensional (in units of 1/m), but is scale independent (i.e., it does not depend upon the volume studied or the orientation of the measurement). Alternative measures, such as the number of fractures per unit area, are scale dependent.

The intensity of faults and fractures are frequently represented in mean spacing, $E[S_f]$, between fractures or faults within a set. The relationship between $E[S_f]$ and P_{32} depends upon the distribution of the orientation of fractures relative to the line along which the spacing S_f was measured:

$$P_{32} = C_p / E[S_f]$$

where:

C_p = constant dependent upon the distribution of the orientation of fractures relative to the line along which the spacing S_f was measured.

For a uniform distribution of fracture organization, Dershowitz (1984) found a value of C_p of 2.0. For most fracture geometries, C_p will vary between 1.0 and 3.0. The exact value of C_p for a particular fracture and borehole geometry can be found by simulated well sampling using FracMan.

Fracture intensity on trace planes can be quantified in terms of the mean fracture spacing, the number of fractures per unit area, or by the measure P_{21} defined by Dershowitz (1984) as:

$$P_{21} = L_f / A_t$$

where:

L_f = Total length of fracture traces on the surface, and

A_t = Total area of the surface.

This measure is used, for example, when fracture intensity must be derived from lineament maps.

The relationship between P_{21} and P_{32} depends upon the relationship between the fracture orientation distribution and the orientation of the surface on which P_{21} was calculated:

$$P_{32} = C_{32} P_{21}$$

where C_{32} is a proportionality constant related to the orientation distribution of the fractures.

For most fracture geometric patterns, values for C_{32} vary between 1 and 3 (Dershowitz, 1984). For a uniform distribution of orientations, a value of 1.3 ($4/\pi$) can be used. The value for C_{32} may be obtained by FracMan simulated trace plane sampling with appropriate size and orientation distributions.

Volume Intensity (P_{33})

Volume intensity, P_{33} is the proportion of the volume of fractures over a particular reservoir volume:

$$P_{33} = V_f / V_t$$

where:

V_f = Total volume of fractures, and

V_t = Total volume.

Since the relationship between fracture volume V_f and fracture area A_f depends upon the aperture of the fractures, the conversion between P_{33} and P_{32} depends upon the distribution of apertures. It is also dependent upon any correlation between fracture aperture and surface area:

$$P_{32} = P_{33} E[A_f] / E[V_f]$$

As a first-order approximation, $E[A_f]/E[V_f]$ can be approximated by the reciprocal of the expected value of fracture aperture $E[e]$, such that

$$P_{32} = P_{33} / E[e]$$

Number of Fractures (N_f)

The use of number of fractures N_f (also referred to as P_{30}) as an intensity measure is strongly discouraged, due to the strong dependence on the volume of the generation region and the size distribution. For a given borehole orientation and fracture orientation distribution, the constant C_p which relates spacing to P_{32} , will not change. However, if the mean fracture size is doubled, the mean fracture spacing will decrease by 50% for the same value of N_f . Similarly, where the fracture size is large relative to the size of the generation region, the number of fractures will not change in proportion to volume changes, because fractions of fractures are counted toward N_f .

Lineal Intensity (P_{10})

Lineal fracture intensity is expressed as the number of fractures per unit length. P_{10} is a one-dimensional measurement, typically recorded as the position of fracture intersections along a

line in space, such as the core axis for boreholes, the well centerline for image log data, or a scanline on surface outcrops or tunnel walls.

Adjusting P_{32} with Truncated Distributions

Fracture Size

Fracture size is specified in terms of the equivalent radius of the fractures before termination. The equivalent radius, r_e , is defined by the radius of the circle with an area equal to the area of the polygon, A_f , which represents the fracture:

$$A_f = \pi r_e^2$$

Elongation

The direction and degree of elongation of a fracture is a function of its dip vector and a random elongation vector that can vary both in magnitude and direction. This functional relationship allows simulation of fractures that tend to be elongated more or less in one direction on a regional scale, but whose direction and degree of elongation is variable. Fractures that lie parallel to the local field of elongation will be the most elongated, while fractures that are normal to the local field of elongation will not be elongated. The degree of elongation is specified in terms of the aspect ratio, which can either be set constant or specified as a uniform distribution. The aspect ratio is related to the ellipticity e by:

$$a = (1 + e)/(1 - e)$$

The direction of elongation of a particular fracture will be E_f , where:

$$E_f = (n \times E) \times n$$

and n is the normal vector to the fracture, calculated from the dip vector, d , by:

$$n_f = d_f + \pi/2$$

where (n_r, n_f, n_q) are the components of n and (d_r, d_f, d_q) are the components of d in spherical polar coordinates, with f being the colatitudinal angle (from vertical) and q being the longitudinal angle from south. The aspect ratio of the fracture will be:

$$a_f = (1 + e_f)/(1 - e_f)$$

where e_f is the ellipticity of the fracture calculated from:

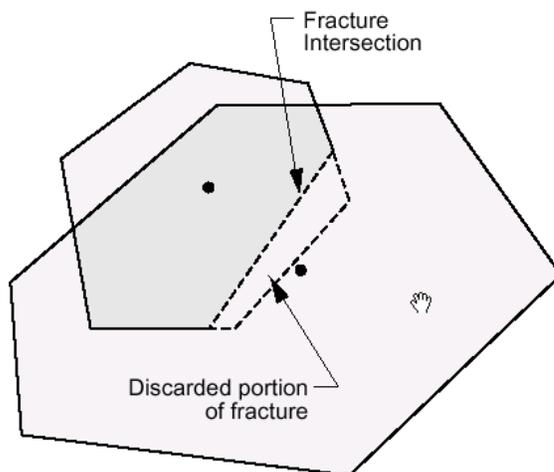
$$e_f = \frac{1}{2} E_f \times E \frac{1}{2} (a-1)/(a+1)$$

Termination

For all models, fracture termination is specified by the termination probability, P_T . P_T is the probability that a fracture will terminate given an intersection with another fracture:

$$P_T \cong \frac{N_T}{N_{Fi}}$$

where N_T is the number of T intersections for fractures in this set with fractures from previously generated sets, and N_{Fi} is the number of fractures in the set having intersections with fractures from previously generated sets. If N_{Fi} is zero, then P_T is zero. The relationship is not exact, because when FracMan terminates fractures at intersections, any intersections with fractures on the portion of the fracture discarded cannot be counted toward N_{Fi} .



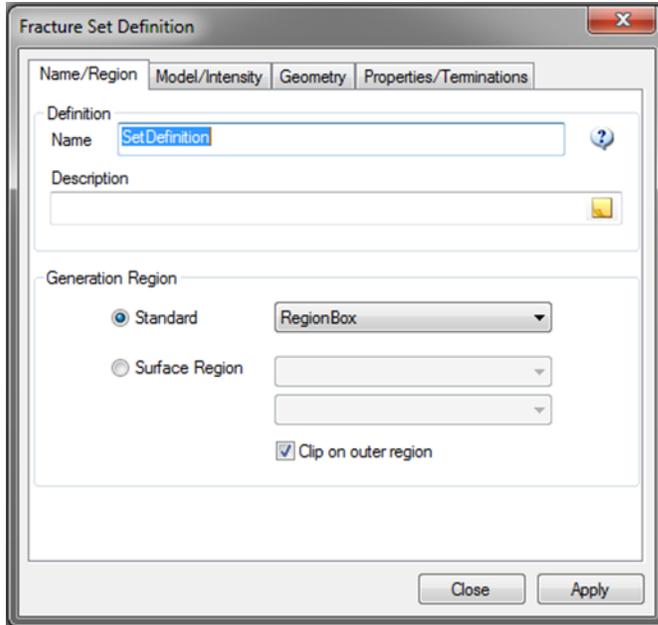
This form of termination disturbs the distribution of fracture size, since portions of fractures are "discarded" if they extend beyond the intersection where termination occurs. This is only a problem for high termination percentages.

Fracture Properties

All Fractures in FracMan have 3 reserved properties that are always present. These properties are controlled by the user option (see [Options](#)) and are either Permeability, Compressibility, Aperture (PCA) or Transmissivity, Storativity, Aperture (TSA). These properties are generally defined in the fracture definition, but may also be interpolated from an external data source or specified manually (see [Modify Properties](#)).

Geometric

The most basic of the fracture sets, this uses a bounding region for the fracture generation area. From the **Fracture** tab on the file menu select **Define New Set > Geometric**. This will open a wizard that walks you through the set definition process.



Specify your fracture set's name in the box next to "Name", and optionally provide a description of your fracture set and attach a note in the box under "Description". You may define the fracture generation by either selecting a bounding region from the drop-down menu next to "Standard" or by selecting the "Surface Region" option and two surfaces that will define the generation region.

Checking "Clip on outer region" will ensure that fractures do not extend out of the selected region by trimming off any portions of the fractures that go beyond the region's confines.

Generation Regions

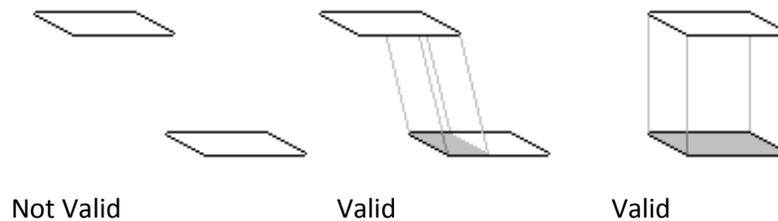
In FracMan, fractures are generated within a generation region. A generation region can be a geometric body like a box or a cylinder, it can be the volume between two surfaces, it can be a grid, or it can be a geologic structure.

Generating fractures inside a box or prism region

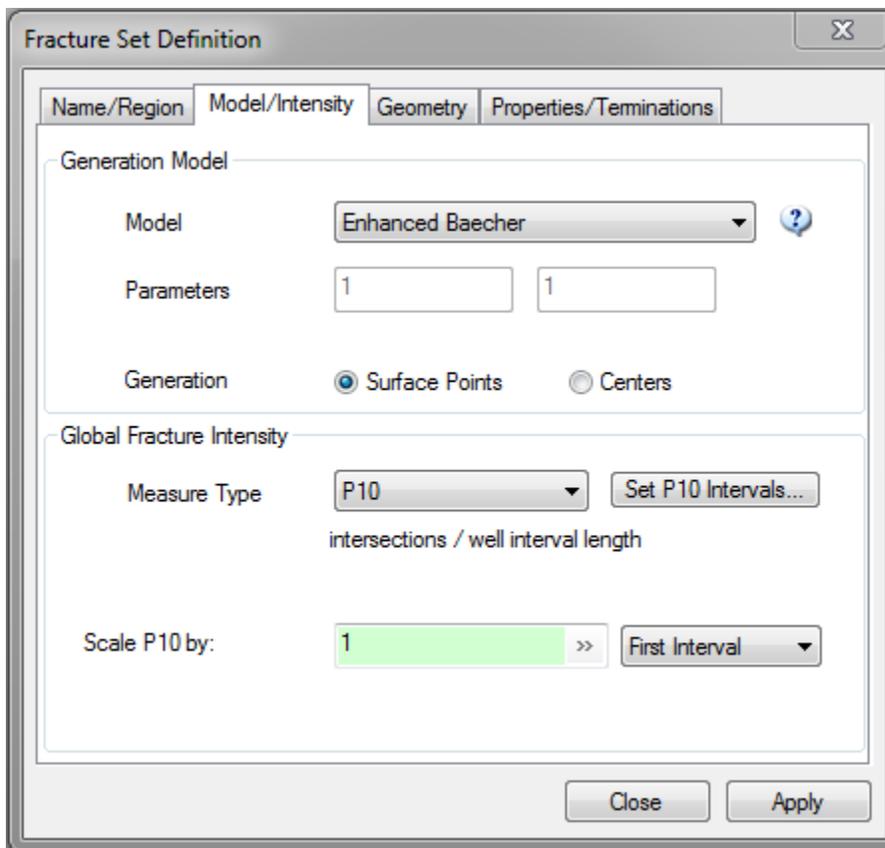
Using the box or prism region can be useful when modeling only a subset of the reservoir, for example for a well model. This option generates fractures more quickly than the alternative option of generation of fractures between surfaces.

Generating Fractures between Surfaces

This option allows for a much more realistic modeling of fractures in a reservoir layer. A layer region is defined as the volume that exists between two overlapping surfaces. There are restrictions regarding how the two layers overlap. As shown below, the surfaces must shadow each other. That is to say, a portion of one surface must lie directly below a portion of the other surface and vice versa to define a valid layer model region.



When you have finished defining the name and region, click on the **Model/Intensity** tab to continue.



Generation Model

There are three different fracture generation algorithms on which you may base your model: [Enhanced Baecher](#), [Nearest Neighbor](#), and [Levy Lee](#). Which you choose depends on what sort of behavior you wish your fractures to exhibit. The generation methods are described in the The Theory Behind Fracture Generation section of this chapter of the manual.

In addition to the generation model itself you can select an option which describes how the location model is used:

- **Surface Points** – The generation point is considered to be a random point within the fracture. Selecting this option can reduce edge effect when generating to P10, P32 values.
- **Centers** – The generation point is the center of the fracture. This is faster than Surface Points

Intensity

Fracture intensity is, simply speaking, the "amount" of fractures present in a given rock mass. Traditional measures of fracture intensity have included terms such as 'fracture spacing' or 'rock quality', which largely ignored the thorny issues of orientation bias, spatial arrangement, and discontinuity size. With the advent of three-dimensional discrete fracture network modeling, more universal definitions of fracture intensity were required. In particular, a method of characterizing fracture intensity that encompasses both the dimensionality of the intensity measurement and the dimensionality of the sampling structure or measurement region is necessary. Modern fracture modelers generally characterize fracture intensity in one or more of the following terms (after Dershowitz, 1984):

- P_{10} : Lineal fracture intensity, expressed as the number of fractures per unit length. P_{10} is a one-dimensional measurement, typically recorded as the position of fracture intersections along a line in space such as the core axis for drillholes, the well centerline for image log data, or a scanline on surface outcrops or tunnel walls.
- P_{32} : Volumetric fracture density, expressed as the sum of the one-sided surface areas of the number of n-sided polygon per unit area, in units of L^2/L^3 . As most DFN modeling techniques decompose fractures to simple convex polygons in space, P_{32} is by far the most useful intensity metric for geometric descriptions. It is, however, extremely difficult to actually measure P_{32} , so much so that it is generally estimated analytically or calculated through stochastic simulation based on other metrics (such as P_{21} or P_{10}).
- P_{33} : Fracture porosity, expressed as the sum of the one-sided surface area for a given fracture, multiplied by its geological aperture e , per unit volume, in units of L^3/L^3 (dimensionless). P_{33} is extremely difficult to actually measure, as it requires not only knowledge of the size and intensity distributions of fracturing, but also additional information about the distribution of fracture aperture. It is generally a target output of fracture modeling, especially when upscaling to full-field simulators such as ECLIPSE.
- **Count**: The total number of fractures generated

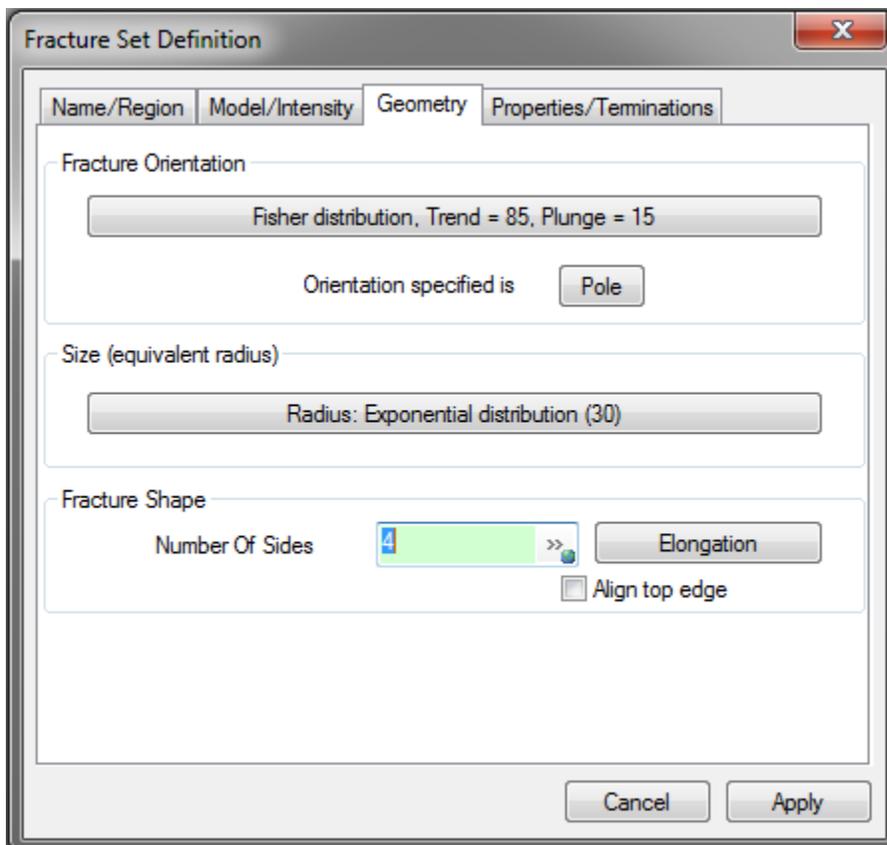
If the measure type chosen is "P10", you must specify which P10 intervals from which wells the fractures are generated to. You must also specify the "Termination Criteria" you wish for your generated fractures.

- **First Interval**: fracture generation will stop once the P10 of just one of your selected intervals has been met, no matter how close the other P10s are. This option tends to produce the least amount of fractures.

- **Average:** fracture generation will cease when the P10 of the generation region equals the average P10 of all the selected intervals. This option is commonly used when several distinct P10 intervals are being matched.
- **All Intervals:** fracture generation will be stopped only when the P10s of all selected intervals have been met or exceeded. This option produces the most fractures of the three options.
- **Scale P10 by:** This provides a mechanism to easily scale up or down the number of fractures generated. All the P10 values will be multiplied by this number for gauging the completion of generation.

If the measure type chosen is “P32” you will also have the option of selecting “Adjust Intensity for Truncation”. When generating fractures to a P32, you may want to reduce the total number of fractures generated by leaving out small fractures or low transmissivity fractures by supplying a truncation in the generation parameters. Since your input P32 represents 100% of the PDF for size (transmissivity / aperture, etc.), we then want to reduce the P32 that we’re generating to in order to account for these “left out” fractures.

When you have finished defining fracture model and intensity, click on the Geometry tab to continue.

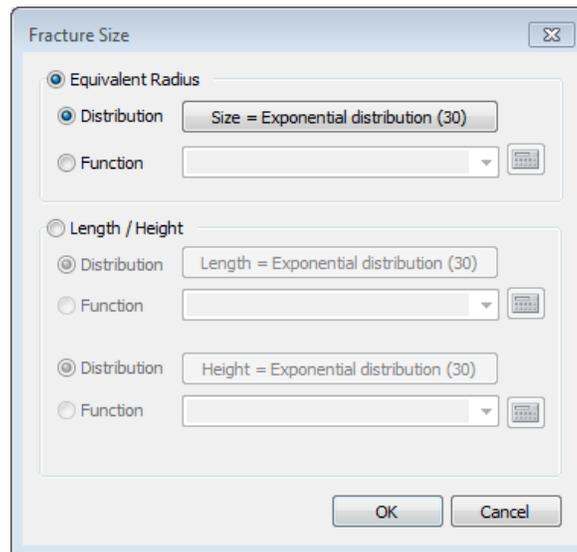


Fracture Orientation

The Orientation specified is a vector distribution of the pole or dip of the fractures. See the [Orientation](#) section for complete details on defining the orientation of fractures.

Size

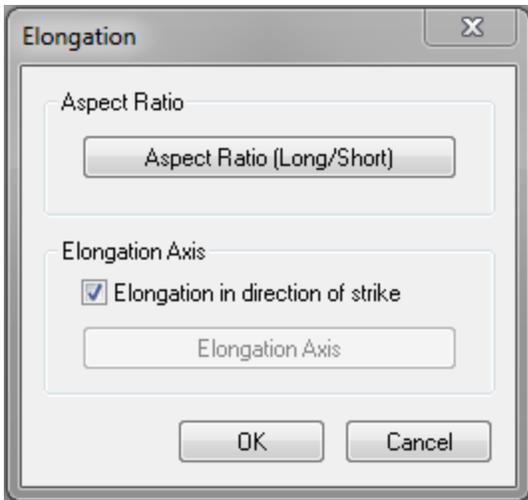
Fracture Size can be defined as either Equivalent Radius (square root of area / PI) or as a Length / Height (4 sided fractures only). Radius / Length / Height can all be defined as scalar distributions or as functions of other geometric properties. You can correlate Length to Height or vice versa. Scalar distributions are described in the “[Distributions](#)” section of this chapter of the manual.



Fracture Shape

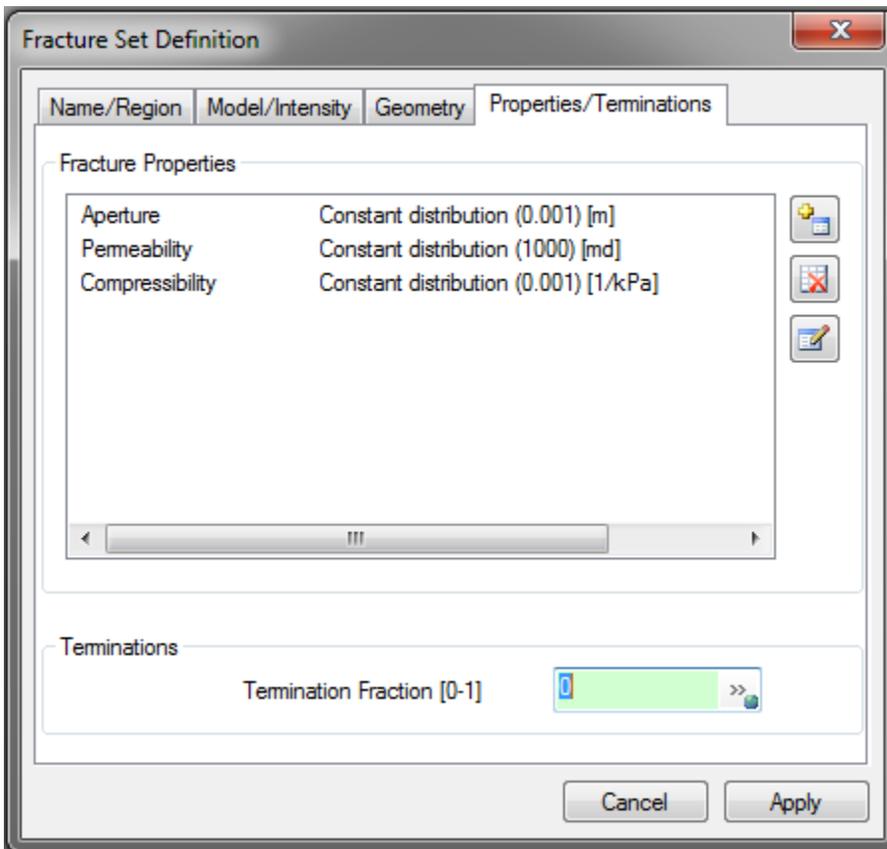
Fractures are generated as polygons with the given number of sides. By default, generated fractures will be equilateral.

Elongation allows you to specify an aspect ratio and a direction in which the fractures are to be stretched (elongation axis). These are defined by a scalar and vector distribution, respectively, which the user may specify by clicking the appropriate button. By default, the elongation axis is aligned in the direction of strike.



Align top edge: This option is only available for 4 sided fractures. It will align one side of the fracture with the x-y plane.

OK will take you back to the fracture generation wizard. When you have finished defining your fracture geometry, click on the **Properties/Terminations** tab to continue.



Fracture Properties

This is where you specify your fracture properties.

To change the definition of these properties, select the desired property and click the **Edit**

Property button . A new window showing the current property parameters will pop up.

Click the **Add Property** button  to add an additional property to the fracture set definition. Again, a new window will pop up, which allows you to specify the property distribution.

You can delete an existing property by selecting the property and clicking the **Delete Property** button . The three default properties cannot be deleted.

Termination Fraction

The Termination Fraction defines the probability that a generated fracture will terminate against an existing fracture. Note that new fractures will terminate on any preexisting fractures in the model.

Warning: for cases where you specify a high termination percentage and there are few preexisting fractures, your fracture potential created by either the Generation Model (in Geometric sets) or the Relative Fracture Intensity (Geocellular sets) can be overridden. See the theory section on [Termination](#) for more detail on how fracture size can be affected as well. For more control over which fracture sets should terminate on other fracture sets, use the [Termination Matrix](#) functionality available from the Fracture -> Termination Matrix menu option after the fracture set generation. The Termination Fraction functionality should only be used for well-connected sets, meaning that most of the new fractures intersect an existing fracture in the model. Also, keep the fraction small to minimize the effect on the size distribution.

Editing/Creating a New Property

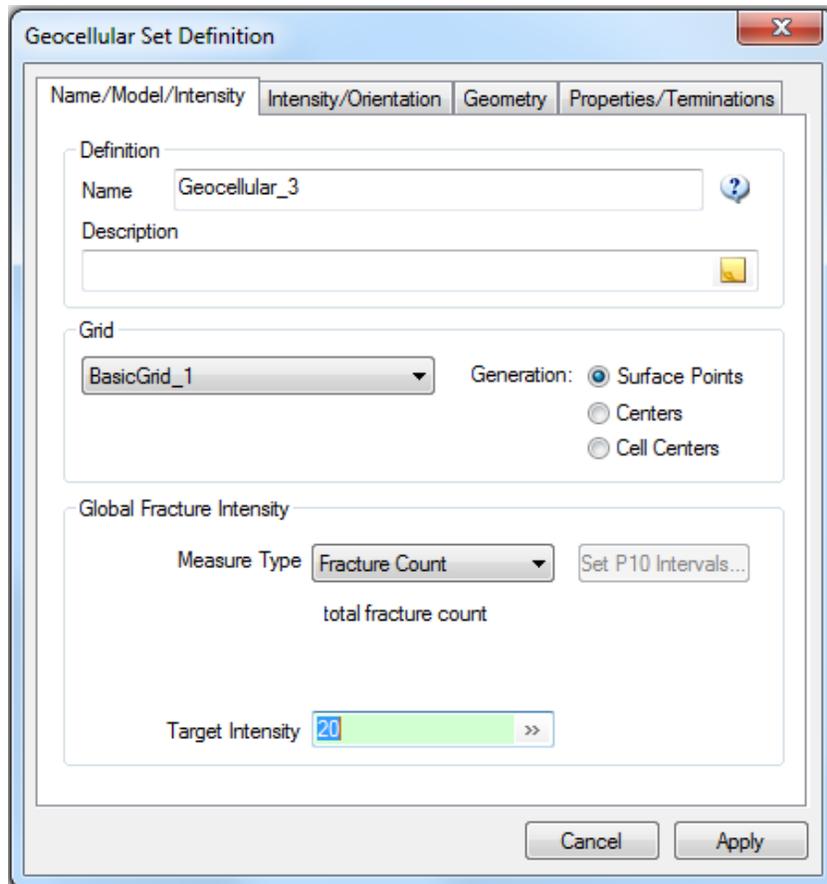
The screenshot shows the 'Define Property' dialog box. The 'Name' field is set to 'Aperture'. The 'Type' is 'Aperture' and 'Units' is 'meters'. Under 'Distribution', the 'Distribution' radio button is selected, and the dropdown is set to 'Constant'. The 'Parameters' section has two input fields: the first contains '0.001 [m]' and the second contains '0'. There is a 'Truncate...' button below the parameters. Under 'Correlation', the 'Correlated to' radio button is selected, and the dropdown is set to 'Property EquivRadius'. Below this is a section for 'Parameters for A + B*x^C + D*e^(E*x)' with five input fields for A, B, C, D, and E, each containing a value and a '>>' button. At the bottom are 'OK' and 'Cancel' buttons.

- *Name*: the name of the property you are editing/creating.
- *Type*: the type of property you are editing/creating.
- *Units*: this allows you to define the unit type of your property
- *Distribution*: this allows you to define your property as a scalar distribution, complete with the option to truncate it to some numerical bounds.
- *Correlation*: this allows you to correlate the property to some other property using a function you define here.

When you have finished defining the property, hit **OK** to return to the fracture definition wizard. When you have completed your fracture set definition, hit **Apply** and **Close**. A new fracture set definition will appear in the Definitions window. To generate the set, navigate to the **Definitions** window, click on the fracture set definition name, and select **Generate**. The newly-generated set will appear in the **Objects** window under “Fracture Sets”. You may reopen the definition of the fracture set for modification by clicking on the fracture set definition name in the **Definitions** window and selecting **Edit Definition**.

Geocellular

The geocellular fracture generation model creates fractures in a volumetric grid based upon properties of the grid.



On the first tab of the Geocellular Set Definition dialog, **Name/Model/Intensity**, specify a definition name, base grid, and target global intensity for your fracture set. Generation mode types “Surface Points” and “Centers” as well as Global Fracture Intensity parameters are explained in the Geometric section above. The Generation mode “Cell Centers” forces the generation point of each fracture to be the center of a grid cell.

The **Intensity/Orientation** tab allows you to specify a distribution for the local relative fracture intensity and another distribution for the fracture orientation. Click on the distribution button under each heading to change these. See the Manual section [Intensity](#) for Geometric fracture sets for a description of these options.

Global Fracture Intensity

Measure Type P32 Set P10 Intervals...

total fracture area / region volume

Adjust Intensity for Truncation ?

Target Intensity 0.0005 [1/m] ↔

Scale Factor 1.17648811361694 Compute P32 Scale Factor

The **Compute P32 Scale Factor** button can estimate a P32 scale factor necessary to match a given P32 allowing for fractures that extend beyond the edge of the grid object. Once the apply button has been pressed, the Compute P32 Scale Factor button will automatically generate some fracture, upscale to the grid, compute the resulting P32 and then fill in the necessary scale factor to match the entered target P32. See [Relative Fracture Intensity](#) for a detailed discussion of Generation P32 vs Actual P32.

Geocellular Set Definition

Name/Model/Intensity Intensity/Orientation Geometry Properties/Terminations

Relative Fracture Intensity

Constant distribution (1) ?

Orientation

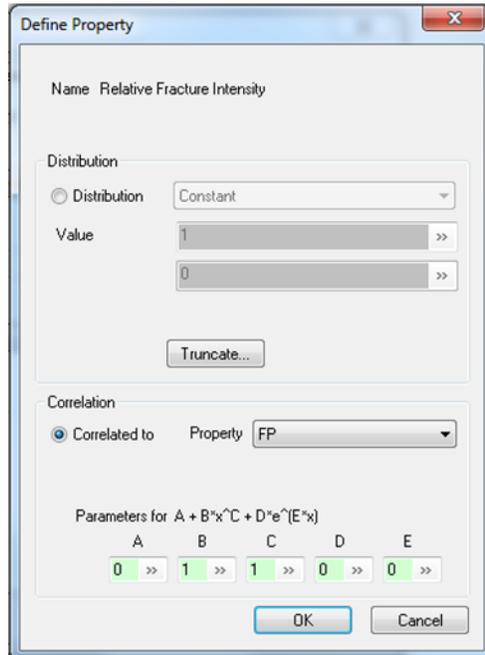
Fisher distribution

Close Generate

Relative Fracture Intensity

The Relative Fracture Intensity is defined as a property distribution so that it can be correlated to match a grid property, such as grid cell P32 or some user-defined property. It can be thought of as an intensity probability. The assigned scalar distribution or function correlated to a grid

property is normalized over the grid so that the sum of the local fracture intensities will equal the global fracture intensity assigned on the Name/Model/Intensity tab. This is useful for defining how the fracture intensity should vary spatially in the model. Leaving it as the default Constant Distribution with Value equal to 1 will result in an equal fracture intensity across the grid. When correlating to a property, the property value is used as the “x” in the correlation equation.



Warning: The relative fracture intensity is only used to locate the generation points of each fracture, the size of the fracture is not considered. Therefore, if the local grid cell P32 is used for the relative fracture intensity and the fracture size is similar to or greater than the grid cell size, the local P32 will not be conserved.

Difference between Global and Relative Fracture Intensity

FracMan’s Geocellular fracture set generation option requires the specification of two parameters used to describe fracture intensity.

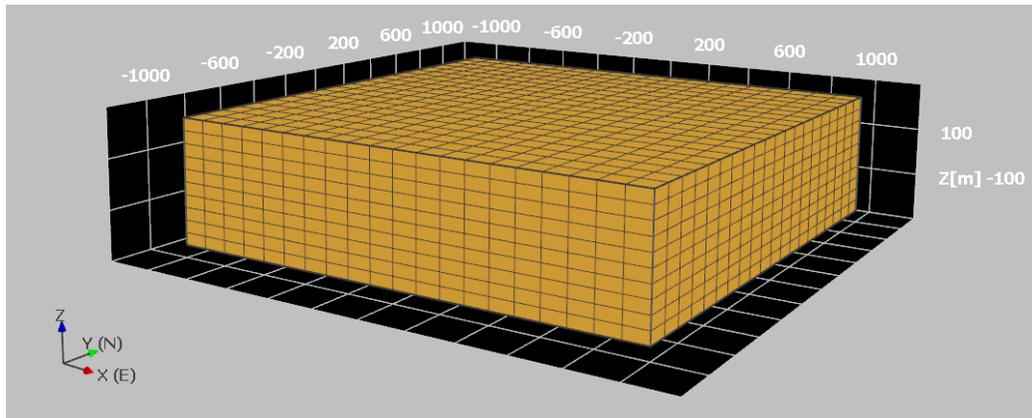
- 1) Global Fracture Intensity, specified as:
 - Fracture count, P_{32} , P_{33} , or P_{10}
 - This represents the total fracture intensity over the entire simulation volume
 - Put another way, for a 100m x 100m x 100m grid, with grid cells 25m x 25m x 25m, this value is the P_{32} over the entire grid (within the 1,000,000 m³ region the grid encloses), i.e. the arithmetic mean (average) P_{32} of all grid cells combined.
- 2) Relative Fracture Intensity, specified as:

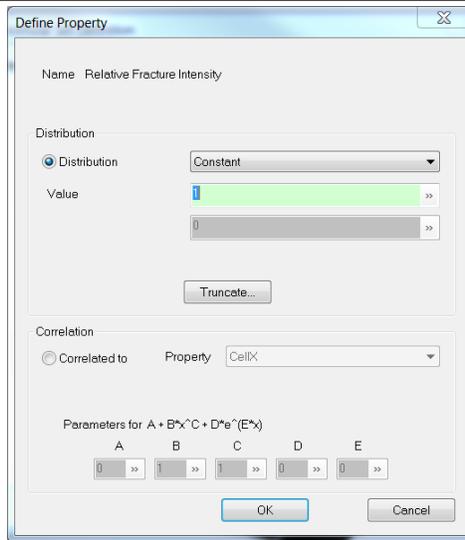
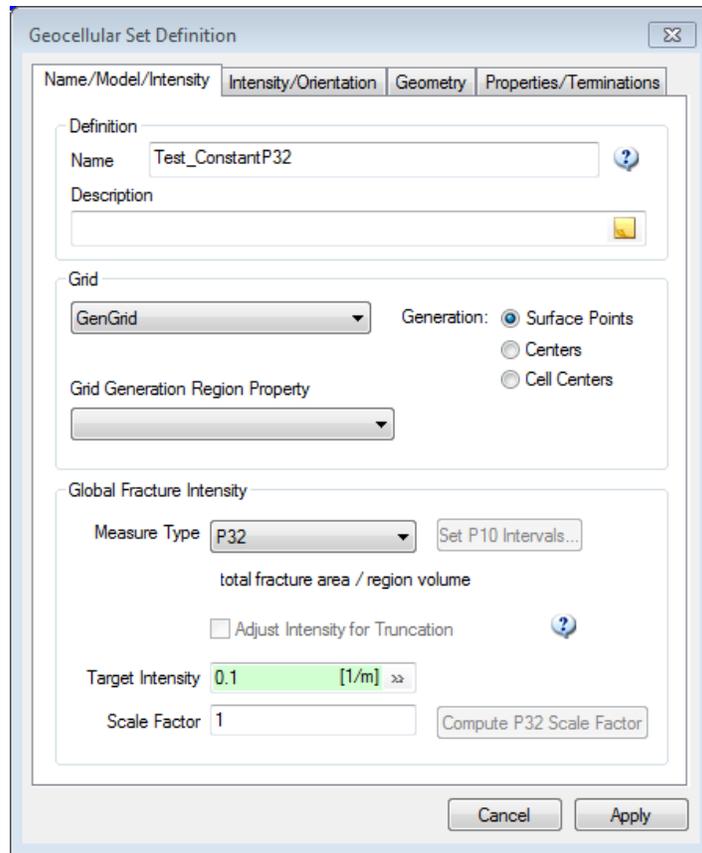
- A constant value, such that the generation of a fracture in ANY grid cell is equiprobable to the generation in any other grid cell
- A non-spatially correlated stochastic distribution defined in FracMan from supported probability distributions
- A value correlated to an existing property in the grid. This can be used to couple intensity to variables such as rock stiffness, seismic facies, or lithology. It can also be used to include an external intensity function, such as the results of a geostatistical simulation

FUNDAMENTAL ASSUMPTION FOR GEOCELLULAR MODELS: The fractures generated are small relative to the dimensions of the grid cell. This is *essential* to getting out what you expect.

Global Fracture Intensity Only Example

In this case, we specify a global P_{32} of $0.1 \text{ m}^2/\text{m}^3$. Our fracture size is a constant 10m radius, which is small relative to our grid cell size. Our Relative Fracture Intensity = 1 everywhere (no variance).

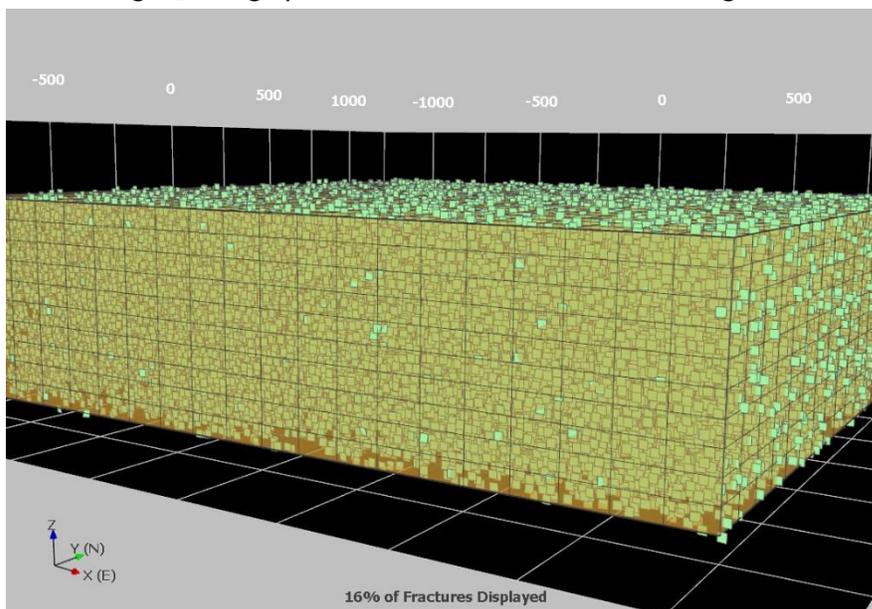




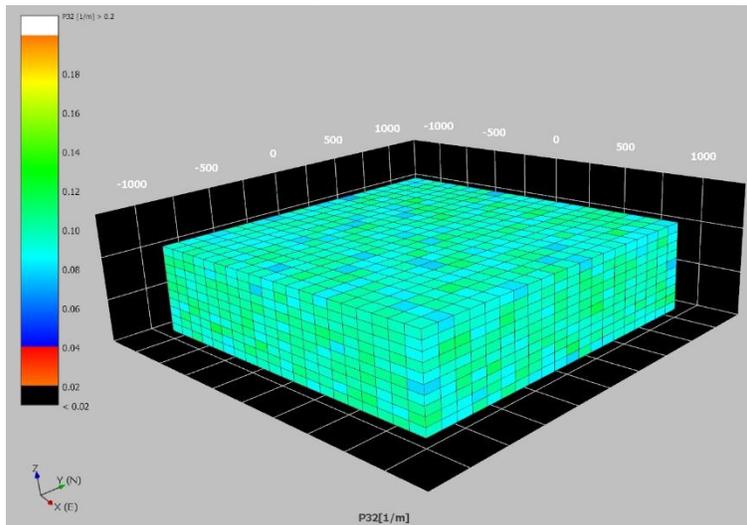
Results show our single simulation mean P_{32} (0.098 1/m) generated is very close to what we specified (0.10 1/m).

Intensity.frd:1 Object Statistic	
Name	GenGrid
Cell Count	4000
Vertex Count	4851
IJK Size	20, 20, 10
Cell Properties	
CellX	meters
CellX min	-1000
CellX max	1000
CellX mean	0
CellX dev	0
CellY	meters
CellY min	-1000
CellY max	1000
CellY mean	0
CellY dev	0
CellZ	meters
CellZ min	-250
CellZ max	250
CellZ mean	0
CellZ dev	0
Active	
Active min	1
Active max	1
Active mean	1
Active dev	0
Cell_Volume	cubic meters
Cell_Volume min	500000
Cell_Volume max	500000
Cell_Volume mean	500000
Cell_Volume dev	0
P32	1/m
P32 min	0.072858684
P32 max	0.1244669
P32 mean	0.098302968
P32 dev	0.0077961232

The missing P₃₂ is largely due to the bits that stick out of the grid.



Upscaled P_{32} (calculate the area of all fractures within each cell, then divide by cell volume) shows the P_{32} is effectively constant across the entire cellular grid.



Definitions for Relative Fracture Intensity

- Relative Fracture Intensity (Rf), is defined as Fracture Potential (Fp) times the Cell Volume:

$$Rf = Fp * Cell_Volume$$

- Rf is a 'hidden intrinsic property' of the grid; the user does not generally directly access it. Instead, changing the distribution of the number or intensity of fractures between different cells (i.e. the relative fracture intensity) in the same grid is controlled by adjusting either the Cell Volume or the Fracture Potential. Note: the non-cumulative Rf may be viewed directly by changing the **StoreGeocellPotential** to "1" in the Macro (see later section).
- Fracture Potential can be a constant value, a function of multiple other grid properties, a stochastic random distribution drawn from FracMan's built-in probability distributions, the result of a regression equation against multiple property values, or the results of an external interpolation process
- Relative Fracture Intensity is generated as a running cumulative sum, normalized to the total Relative Fracture Intensity for the entire grid. The calculation is explained on the next section

Relative Fracture Intensity, Worked Example

Geocellular grid within a 100m x 100m x 100m volume

Each grid cell is 25m x 25m x 25m on a side, such that there are 64 total grid cells

Each grid cell has a volume of $(25m)^3$ or $15625 m^3$

Fracture Potential is 1 in every cell

For Cell #1: Relative Fracture Intensity = Fracture Potential * Cell Volume:

$$Rf_{Cell1} = 1 * 15625 m^3 = 15625 m^3$$

For Cell #2: $R_{fCell2} = F_{pCell2} * Cell_Volume_{Cell2} + R_{fCell1}$

$$R_{fCell2} = [1 * 15625 \text{ m}^3] + 15625 \text{ m}^3 = 31250 \text{ m}^3$$

For Cell #3: $R_{fCell3} = F_{pCell3} * Cell_Volume_{Cell2} + R_{fCell2}$

$$R_{fCell3} = [1 * 15625 \text{ m}^3] + 31250 \text{ m}^3 = 46875 \text{ m}^3$$

Next, normalize each cell by the total cumulative relative fracture intensity, which should be the value of R_f for the last cell calculated. In our example, the cumulative relative fracture intensity would be:

$15625 \text{ m}^3 * 64 \text{ (cells)} = 1,000,000 \text{ m}^3$, which, conveniently, is the volume of 100m x 100m x 100m cube!

The normalized value of R_f (R_f') for the first cell will be:

$$15625 \text{ m}^3 / 1,000,000 \text{ m}^3 = 0.015625 \text{ or } 1/64^{\text{th}}, \text{ since our fracture potential is constant.}$$

How Fractures Are Generated using Relative Fracture Intensity

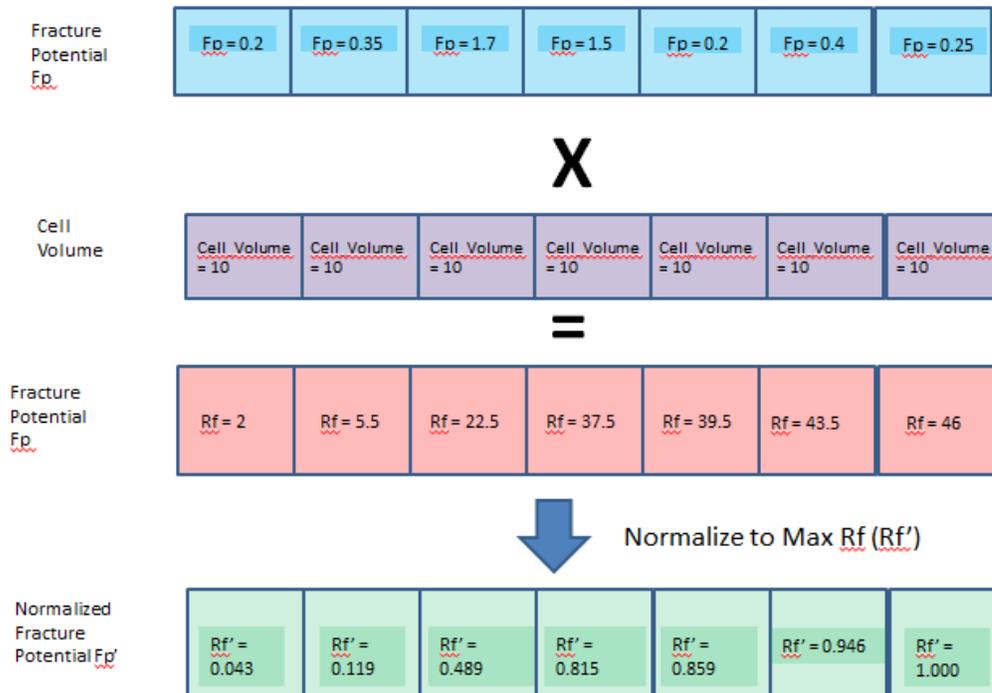
1. Relative Fracture Intensity is calculated for each cell in the grid as Fracture Potential * Cell Volume. This is a cumulative value, such that for each successive cell, R_f is equal to the sum of all previous cells, plus the value of the current cell.
2. Relative Fracture Intensity is then normalized (R_f') for each cell to the total cumulative sum*, such that the distribution of R_f' is now effectively a CDF between 0 and 1. Each grid cell corresponds to a range of numbers on the CDF. Larger volumes OR higher fracture potential means a given cell occupies m
3. A random number between 0 – 1 is now chosen.
4. The random number is used to select a position on the CDF. This corresponds to a specific grid cell. In point of fact, it represents a range of values on the CDF which that cell 'occupies'
5. A fracture location (center or surface point) is chosen at random within the target grid cell. The orientation, size, shape, and properties are generated according to the fractures set definition.
6. Depending on the Global Fracture Intensity metric chosen (P_{32} , P_{33} , Count), the area, volume, or the number of the fracture is added to a cumulative count
7. Steps 1 – 6 are repeated until the Global Fracture Intensity criteria is met

* By inspection, one can see that:

- If R_f is constant and Cell_Volume is variable, larger cells will have more fractures
- If R_f is variable and Cell_Volume is constant, cells with large values of R_f will have more fractures than cells with small values of R_f

Relative Fracture Intensity, Illustrated

Case 1: Constant Cell Size (volume), Variable Relative Fracture Intensity. Fracture Potential (F_p) in blue, Relative Fracture Intensity (R_f) in Red. Cell Volume = 10 units



So, how does this work WRT fracture generation?

<u>Cell1</u>	<u>Cell2</u>	<u>Cell3</u>	<u>Cell4</u>	<u>Cell5</u>	<u>Cell6</u>	<u>Cell7</u>
$R_f' = 0.043$	$R_f' = 0.119$	$R_f' = 0.489$	$R_f' = 0.815$	$R_f' = 0.859$	$R_f' = 0.946$	$R_f' = 1.000$

- If PNRG returns a number between 0 and 0.043, a fracture is generated in Cell 1
- If PNRG returns a number between 0.489 and 0.815, a fracture is generated in Cell 4
- If PNRG returns a number between 0.946 and 1.000, a fracture is generated in Cell 7

Generation continues until the Global Fracture Intensity metric (P_{32} , P_{33} , Fracture Count, or target P_{10} values in wells) is met.

Using a Macro to Create Relative Fracture Potential as a Grid Property

There is a feature that allows you to save the relative fracture potential into a grid property that you can visualize. It's the non-cumulative version, which is just the raw potential * volume for each cell. You can see in the macro emitted when you define a geocellular definition:

```
#Flag to save resulting geocellular potential as a grid property for diagnostic purposes
StoreGeocellPotential = 0
```

Just change the 0 to a 1 (you can do this in the macro bar and just re-execute that definition)

and after you generate you'll get a property called "DefinitionName_pot". This can be useful to figure out why fractures aren't generating where you expect them to.

Some Caveats with Relative Fracture Intensity

If your fracture size is at the scale of the grid cell or larger, generated fractures will cross into adjacent grid cells (leak). The algorithm tracks cumulative total P_{32} (or P_{33} , or fracture count) not the generated P_{32} in each cell. The result will likely be a P_{32} pattern in the grid that DOES NOT match your input distribution in terms of relative fracture intensity.

This problem is exacerbated when the change in the property used as a proxy for relative fracture intensity is very large between cells (i.e. not changing smoothly with distance).

It is important to remember that the relative fracture intensity is NOT the P_{32} value for each cell. Even if you create a property in the grid called P_{32} , during geocellular generation, that value is used as a proxy to calculate the relative fracture intensity, relative to other cells in the grid. It is NOT a local P_{32} .

Calculating a Sub-region P_{32} from a Regional P_{32} using a Local Fracture Potential in a Grid

There are times when you have a local fracture potential defined on a grid and you want to generate geocellular fracture sets on portions of this grid, perhaps in only one formation at a time. For this, you generally need the P_{32} of the sub-region that you are defining your fracture set, which may be different than the P_{32} of the entire region. This may be calculated using the following formula:

$$P_{32_{sub}} = (RFI_{sub} * A_{total}) / (RFI_{total} * V_{sub})$$

Where RFI is the Relative Fracture Intensity = Cell_Fracture_Potential * Cell_Volume, A is the fracture area and V is the volume. The subscripts "sub" and "total" refer to the sub-grid and the full-grid respectively.

To get the RFI_{sub} and the RFI_{total} , you need to first create a grid cell property for the Relative Fracture Intensity (FP * Cell_Volume) and then sum this cell property over the region of interest (either the sub-grid that you will be using to generate the geocellular fracture set or the entire grid area where the P_{32} is known). The fracture area A_{total} can be calculated from the known P_{32} of the entire region and the total volume of the entire region ($A_{total} = P_{32_{total}} * V_{total}$). The volumes of the total region and the sub-region of interest need to be summed from the individual cell volume values.

The key assumption made when deriving the formula for $P_{32_{sub}}$ shown above is that the ratio of the Relative Fracture Intensity summed over the sub-region to the Relative Fracture Intensity summed over the total region times the fracture area summed over the total region equals the desired fracture area summed over the sub-region:

$$(RFI_{sub} / RFI_{total}) * A_{total} = A_{sub}$$

This assumption is quite good for DFNs where the fracture sizes are small compared to the grid cell sizes and there are a large number of fractures but is only a rough approximation as fracture sizes get large compared to the size of the grid cells. Below is an example of how this

assumption works using a total grid consisting of five cells, two of which belong to Formation A and three which belong to Formation B:

Formation A		Formation B		
$RFI_1 = FP_1 * CV_1$	$RFI_2 = FP_2 * CV_2$	$RFI_3 = FP_3 * CV_3$	$RFI_4 = FP_4 * CV_4$	$RFI_5 = FP_5 * CV_5$

$$RFI_A = RFI_1 + RFI_2$$

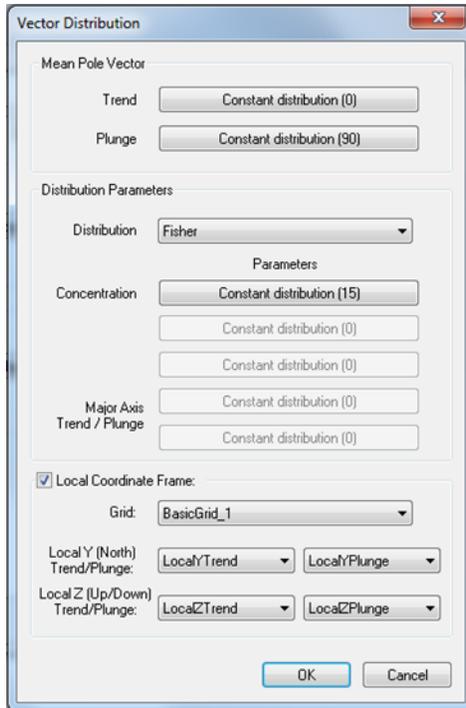
$$RFI_B = RFI_3 + RFI_4 + RFI_5$$

$$RFI_{total} = RFI_A + RFI_B$$

During geocellular fracture set generation, FracMan normalizes the Relative Fracture Intensity over the entire generation region such that the distribution of RFI is effectively a cumulative distribution function (CDF) between 0 and 1. Each grid cell corresponds to a range of numbers on this CDF. For each fracture created, a random number between 0 and 1 is selected and compared with the RFI CDF. The location of the fracture is chosen to be within the grid cell matching the location where that random number lies on the CDF. The location of the fracture within the grid cell is then randomly selected through another process. If the Global Fracture Intensity metric used for the geocellular fracture set definition is P_{32} , then the area of the newly created fracture is added to a cumulative count and fracture generation continues until the target fracture area is reached.

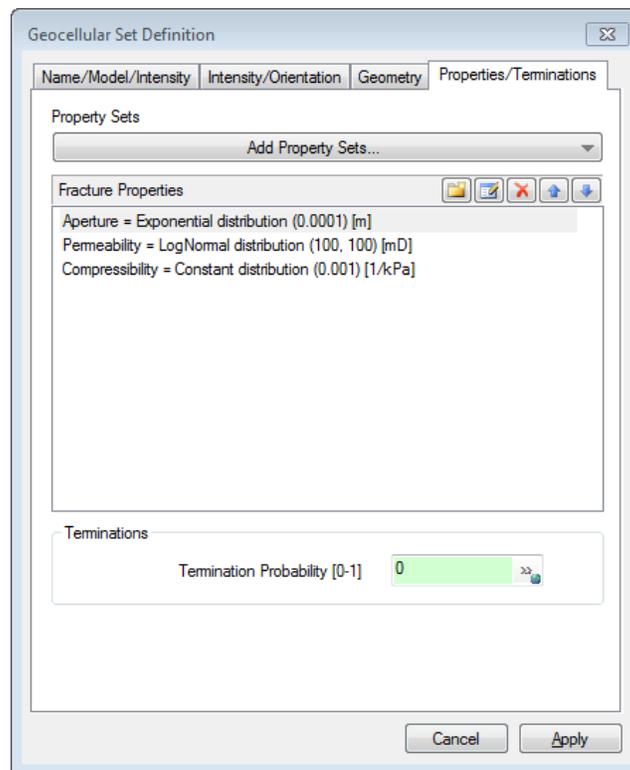
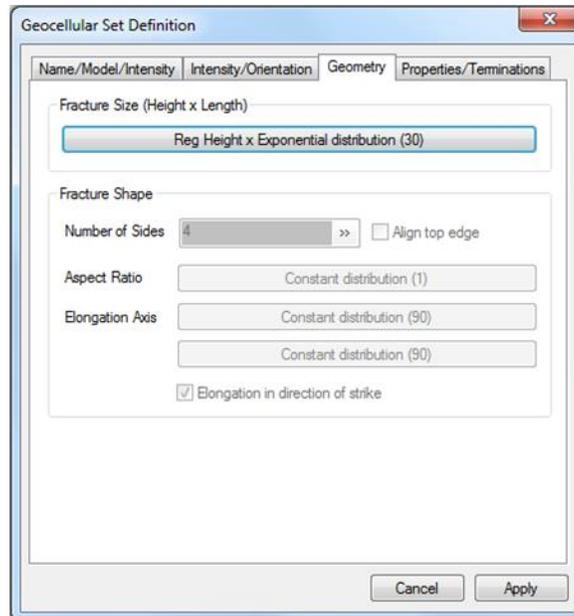
Fracture Orientation

Fracture orientation is a vector distribution, but its definition window is more complex than the regular orientation distribution window. This is so that you can either specify distributions or correlate distribution parameters to grid properties using the Local Coordinate Frame parameters **Local Y (North) Trend/Plunge** and **Local Z (Up/Down) Trend/Plunge**.



All parameters here are defined as property distributions, so that you may define, for example, trend by a function linking it to grid curvature. If you wish, you may also use a local coordinate frame. Just check the box next to **Local Coordinate Frame** and specify the grid by whose properties you would like to transform your fracture orientations. Then select grid properties for each of the four coordinate frame parameters. If you want your local coordinate frame to be based on the grid cell orientation, use the Compute->[Compute Local Coordinate Frame](#) function (right-click on the grid) to generate the required properties before completing this tab.

The **Geometry** tab is used to specify size and shape. Fracture size, like everything else in the Geocellular model, is defined as a scalar property distribution which can be correlated by some function to a grid property. Cell height can be used as one of the input parameters for determining the fracture geometry. This feature is useful when the grid represents stratigraphy and the fractures are assumed to be layer bound. Likewise, the elongation parameters for your fractures can be made a function of a grid property as well. Specify the number of sides you wish your fractures to have; if they are square, you may align them to the x-y plane by checking “Align top edge”.



Property Sets

You can add property definitions to the Geocellular set definition manually or you can add predefined sets of properties as well as property correlations.

Fracture Properties

Each property has a corresponding entry, double click to edit. You can change the order of the fracture properties using the up and down arrows. You can only correlate to a property that is higher in the list. If you want to correlate to a grid property or other geometric property you can add those via the property sets button.

Terminations

See the Warning for using the Termination Fraction safely in the [Geometric Fracture Sets](#) section. The Termination Fraction functionality should only be used for well-connected sets, meaning that most of the new fractures intersect an existing fracture in the model. Also, keep the fraction small to minimize the effect on the size distribution.

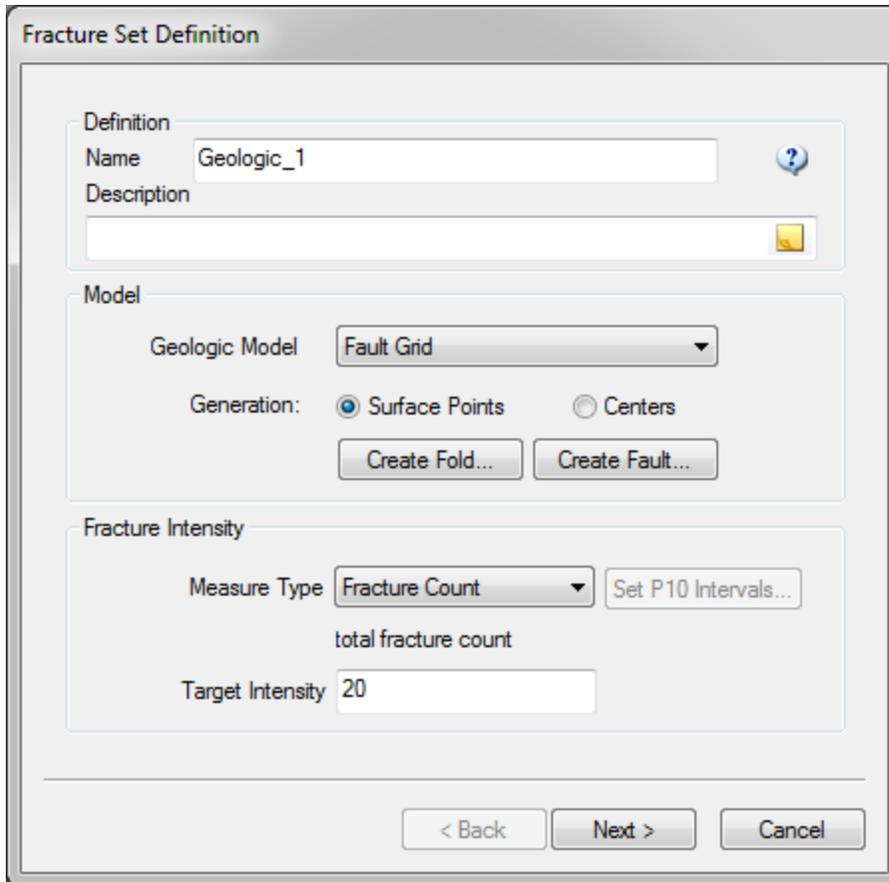
Generation Area

The only difference in this **Properties/Terminations** tab with the one for the Geometric Fracture sets is the option to use only certain grid cells in your fracture generation via a grid generation region property. This is a user-defined, Boolean grid property (either 0 or 1). Grid cells having a value of 1 (or at least not 0) are used during the geocellular fracture generation.

When you have completed all the tabs, click **Apply** and then either **Close** or **Generate**. The fracture set definition will appear in the Definitions window; you can generate the set later by right-clicking on the definition name and selecting **Generate Set**.

Geologic

This generation method is very similar to the Geocellular fracture set, but it requires a Fold, Fault, or Stress Model Grid to work.



The image shows a software dialog box titled "Fracture Set Definition". It is divided into three main sections: "Definition", "Model", and "Fracture Intensity".

- Definition:** Contains a "Name" field with the text "Geologic_1" and a help icon (?). Below it is a "Description" field with a yellow square icon.
- Model:** Contains a "Geologic Model" dropdown menu set to "Fault Grid". Below this are two radio buttons for "Generation": "Surface Points" (which is selected) and "Centers". At the bottom of this section are two buttons: "Create Fold..." and "Create Fault...".
- Fracture Intensity:** Contains a "Measure Type" dropdown menu set to "Fracture Count" and a "Set P10 Intervals..." button. Below this is the text "total fracture count". At the bottom of this section is a "Target Intensity" field with the value "20".

At the bottom of the dialog box are three buttons: "< Back", "Next >", and "Cancel".

Specify a unique name for the fracture set. Then select an existing fault or fold model grid to which you would like to generate your fractures.

Choose an intensity type to which you wish to match your fractures. Fracture set target intensity works the same way as with the Geometric model and is explained above.

When you have completed the screen, hit **Next**.

Geologic Properties

Fracture Subset

Fracture Subset

Conjugate Angle Alpha Beta

Intensity Parameters

$A + Bx + C + Ex$

	A	B	C	D	E
Slip Magnitude	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="1"/>
Tipline Distance	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="1"/>

Stress Function

$X = A + B \text{ sig1} + C \text{ sig2} + D \text{ sig3}$

A B C D

Fold Model

The Fold Model allows the generation of fracture networks associated with folded rock layers. Within a multi-layer fold geometry specified by the user, fracture networks can be generated based on accepted theoretical models for fracturing in folds.

Each fold is comprised of one or more layers. Each layer, in turn, consists of one or more elements of specified thickness. The intensity of fracturing in each element is controlled by a quantity termed the *fracture potential*, which is a measure of relative probability for fracture occurrence. In the fold model, fracture potential is a function of bed thickness and distance from the layer bottom.

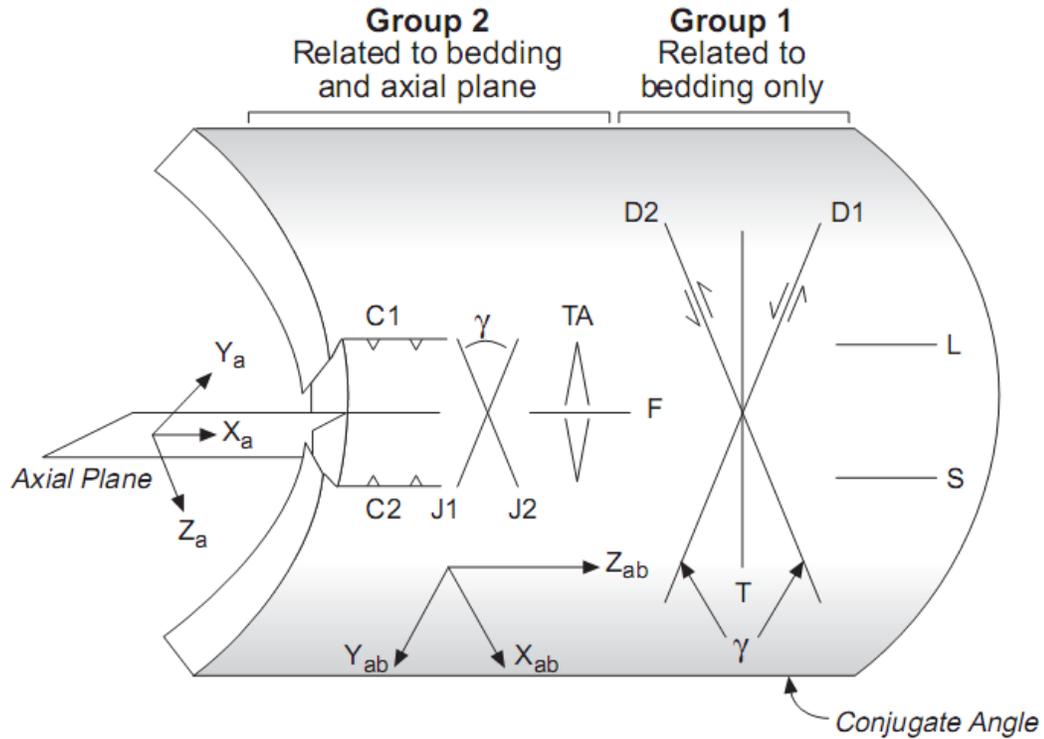
Fracture orientation is controlled by the local orientation of bedding and by the orientation of the axial surface of the fold. The theory underlying these conditioning relations is described in greater detail below.

Fold Fracture Orientation

Fractures generated in the fold model may be separated into three groups. The orientations of Group 1 fractures are constrained only by the geometry of the local bedding. The orientations of Group 2 fractures, on the other hand are constrained both by the geometry of the local bedding and by the geometry of the nearest axial surface element. Group 3 fractures lie along

the top or bottom surfaces of fold layers. Their geometry is completely prescribed by and identical to the top or bottom surface of a fold element.

The orientation of this subset will be used as the mean pole vector for the orientation distribution specified in step 6.



The orientations of Group 1 and Group 2 fractures are calculated as a function of the three angles α , β and the conjugate angle γ . For Group 1 fractures these are computed in the following way:

From each fold element definition, a fold element (bedding) coordinate system $\{X_b, Y_b, Z_b\}$ is created such that:

- X_b = strike vector of fold element
- Y_b = dip vector of fold element
- Z_b = fold element normal vector

The orientation of each axis is computed as follows:

- Z_b = normal vector computed by applying the right-hand corkscrew rule to the element nodes.
- The top and bottom of the fold element are defined by the $+Z_b$ and $-Z_b$ faces, respectively.
- $X_b = Z_b \otimes Z_g$ (Special Case: If $Z_b = \pm Z_g$, then $X_b = \pm X_g$)

$$Y_b = Z_b \otimes X_b$$

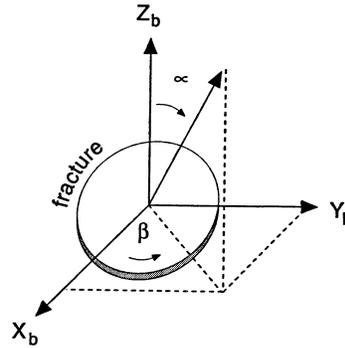
The fracture orientation is then calculated in this hybrid coordinate system using two orientation angles α and β . These orientation angles constrain the fracture pole unit vector, N_f , as follows:

α = angle between N_f and Z_b
 β = angle between X_b and the projection of N_f onto the X_b - Y_b plane

$$N_f(X_b) = \sin\alpha \cos\beta$$

$$N_f(Y_b) = \sin\alpha \sin\beta$$

$$N_f(Z_b) = \cos\alpha$$



Note that for $\alpha = 0$, the X_b and Y_b components of N_f are zero and β is of no significance.

In FracMan, a number of subsets have been predefined to simplify user input. Each subset is distinguished by unique orientation angles α and β . Some subset definitions make use of the user-specified conjugate fault angle, γ .

Subset	α	β
T	0°	0°
S	90°	90°
L	90°	90°
D1	$\gamma/2$	90°
D2	$\gamma/2$	-90°

For Group 2 fractures the orientations α , β and the conjugate angle γ are calculated as follows:

From the axial plane element definition, an axial plane element coordinate system $\{X_a, Y_a, Z_a\}$ is created such that:

X_a = axial plane hinge vector

$$Y_a = Z_a \otimes X_a$$

Z_a = axial plane element normal vector

where:

Z_a = normal vector computed by applying the right-hand corkscrew rule to nodes a_1 through a_4 .

The fold element coordinate system described above is then combined with the axial plane coordinate system to create a combined axial plane/fold element coordinate system $\{X_{ab}, Y_{ab}, Z_{ab}\}$ in which:

$$Z_{ab} = X_a$$

$$Y_{ab} = Z_{ab} \otimes Z_b \text{ (Special Case: If } Z_{ab} = \pm Z_b, \text{ then } Y_{ab} = \pm Y_b)$$

$$X_{ab} = Y_{ab} \otimes Z_{ab}$$

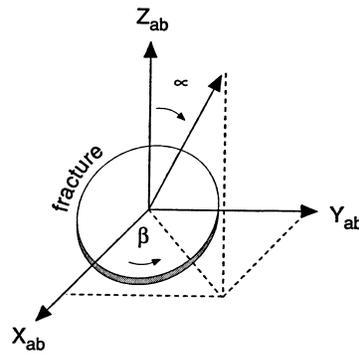
The fracture orientation is then calculated in this hybrid coordinate system using two orientation angles α and β . These orientation angles constrain the fracture pole unit vector, N_f , as follows:

- α = angle between N_f and Z_{ab}
- β = angle between X_{ab} and the projection of N_f onto the X_{ab} - Y_{ab} plane

$$N_f(X_{ab}) = \sin\alpha \cos\beta$$

$$N_f(Y_{ab}) = \sin\alpha \sin\beta$$

$$N_f(Z_{ab}) = \cos\alpha$$



Note that for $\alpha = 0$, the X_{ab} and Y_{ab} components of N_f are zero and β is of no significance.

Again, a number of subsets have been predefined to simplify user input.

Subset	α	β
TA	90°	0°
F	0°	0°
J1	$90^\circ - \gamma/2$	0°
J2	$90^\circ - \gamma/2$	180°
C1	$90^\circ - \gamma/2$	90°
C2	$90^\circ - \gamma/2$	-90°

Fold Fracture Intensity

Fracture intensity can vary throughout a fold. The user specifies the *average* fracture intensity over the entire fold in terms of the fracture porosity, P_{33} , fracture intensity, P_{32} , or the total number of fractures.

Each layer of the fold is described by elements, which are prismatic cells with a triangular or quadrilateral cross-section and a specified thickness. Associated with each fold cell is a fracture potential which determines the relative likelihood that it will receive a newly-generated fracture.

Elements with higher fracture potentials will tend to have more fractures generated in them. The fracture potential of a cell is specified by the user as a function of layer thickness. In addition, the distribution of fractures within a particular element can vary as a function of distance from the layer (element) bottom.

Note that fracture potential determines the relative number of fracture *generation points* received by an element and not the number of fractures or amount of fracture surface area. If fractures are of significant size relative to element dimensions, they may span more than one element, possibly smoothing other measures of fracture intensity in the process.

Specify the intensity parameters. The fracture potential of each cell of the geologic region is computed as the sum of up to two individual fracture potentials. These fracture potentials are calculated from scalars specified for each grid cell using a polynomial/exponential function of the form:

$$f(x) = A + Bx^C + De^{Ex}.$$

When you have completed the screen, hit **Next**.

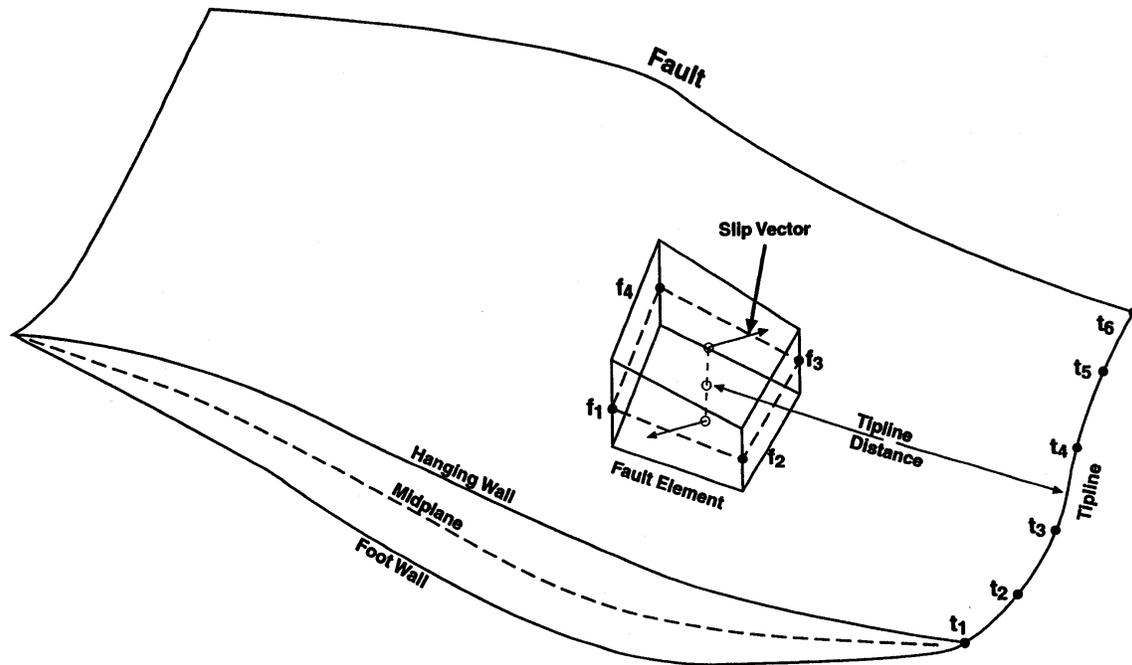
Geometry and Properties/Termination

The final two screens of the wizard are identical to the final two screens of the “Geometric” fracture wizard described above.

Fault Model

The fault model allows the generation and sampling of fracture networks associated with geologic faults. Within a fault geometry specified by the user, fracture networks can be generated to match intensities, orientations or other properties constrained by data or predicted by theory.

In FracMan, a fault is a quasi-planar feature comprised of one or more triangular or quadrilateral elements of specified thickness. The intensity of fracturing in each element is controlled by a quantity termed the *fracture potential*, which is a measure of relative probability for fracture occurrence. In the fault model, fracture potential is a function of slip magnitude, distance from the fault tip, distance from the fault midplane, and up to two user-specified scalar intensity parameters. Fracture orientation is controlled by the local fault plane orientation, the slip direction, and the inclination of principal stresses to the slip plane. The theory underlying these relations is described in greater detail below.

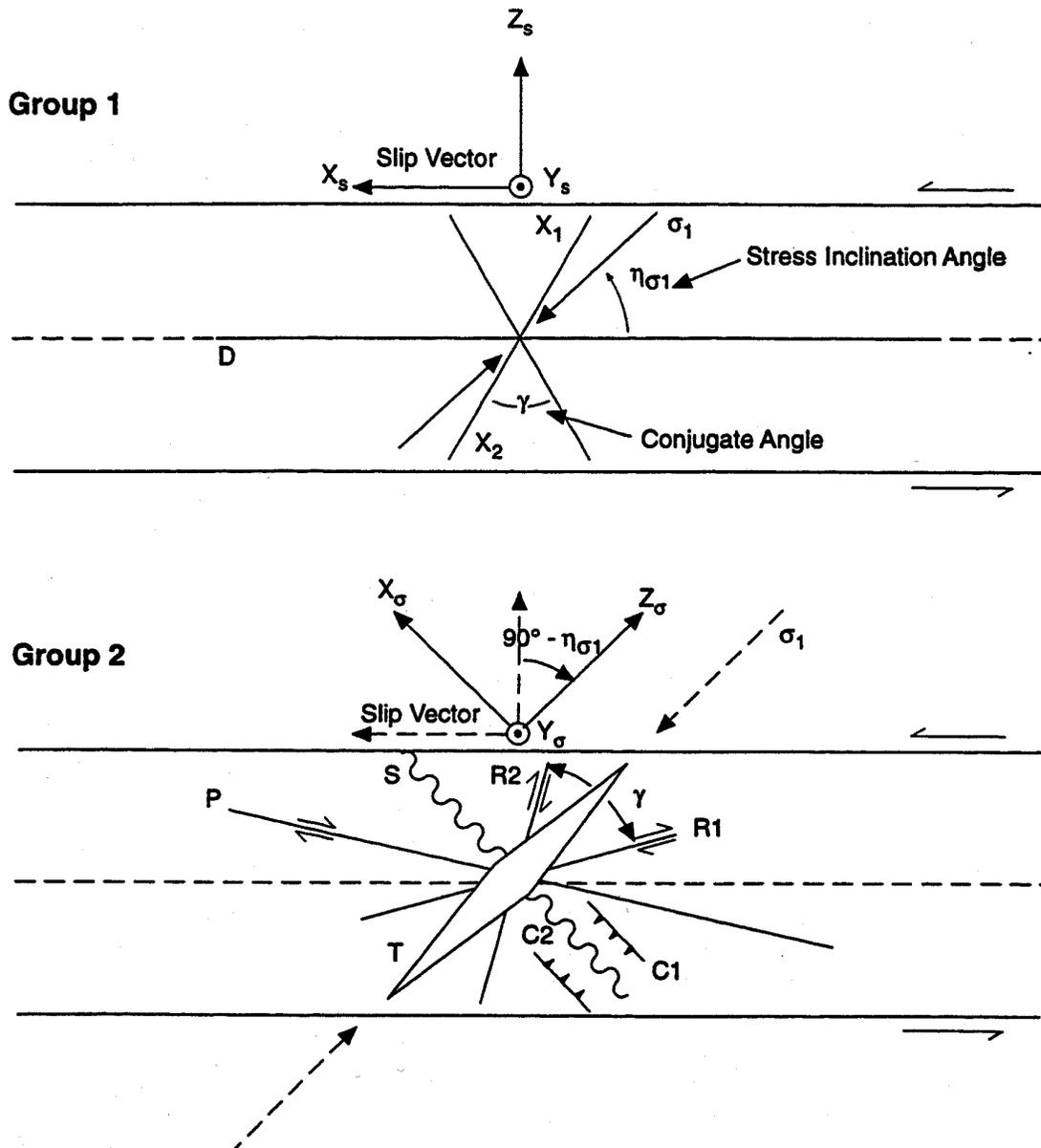


Fault Fracture Intensity

Fracture intensity can vary throughout the user-defined fault. The user specifies the average fracture intensity over the entire fault in terms of the fracture porosity, P_{33} (fracture volume/unit volume), fracture intensity, P_{32} (fracture area/unit volume), or the total number of fractures. The fault itself is described by a grid of prismatic cells with a triangular or quadrilateral cross-section and a specified thickness. Associated with each fault grid element is a fracture potential, which determines the relative likelihood that it will receive a newly-generated fracture. Elements with higher fracture potentials will tend to have more fractures generated in them. The total fracture potential of a cell is calculated as the product of individual fracture potentials arising from scalar parameters of slip magnitude and thickness plus up to two user-defined intensity parameters. The distribution of fractures within a particular element can vary as a function of distance from the element mid-plane.

Fault Fracture Orientation

Fractures generated in the fault model may be separated into three groups. The orientations of Group 1 fractures are constrained by the geometry of the fault element in which they are generated and by the local slip vector. The orientations of Group 2 fractures depend on one additional parameter: the inclination of principal stresses to the current plane of slip. Groups 3 fractures lie along the hanging wall, footwall, or midplane of the fault. Their geometry is completely prescribed by and identical to the top, bottom, or midplane surface of a fold element.



The orientations of Group 1 and Group 2 fractures are calculated as a function of the three angles α , β and the conjugate angle γ . For Group 1 fractures these are computed in the following way:

From the fault element definition, a fault element coordinate system $\{X_f, Y_f, Z_f\}$ is created such that

X_f = strike vector of fault element

Y_f = dip vector of fault element

Z_f = fault element normal vector

The orientation of each axis is computed as follows:

Z_f = normal vector computed by applying the right-hand corkscrew rule to the element nodes.
 The hanging wall and foot wall of the fault element are defined by the $+Z_f$ and $-Z_f$ faces, respectively.

$$X_f = Z_f \otimes Z_g \quad (\text{Special Case: If } Z_f = \pm Z_g, \text{ then } X_f = \pm X_g)$$

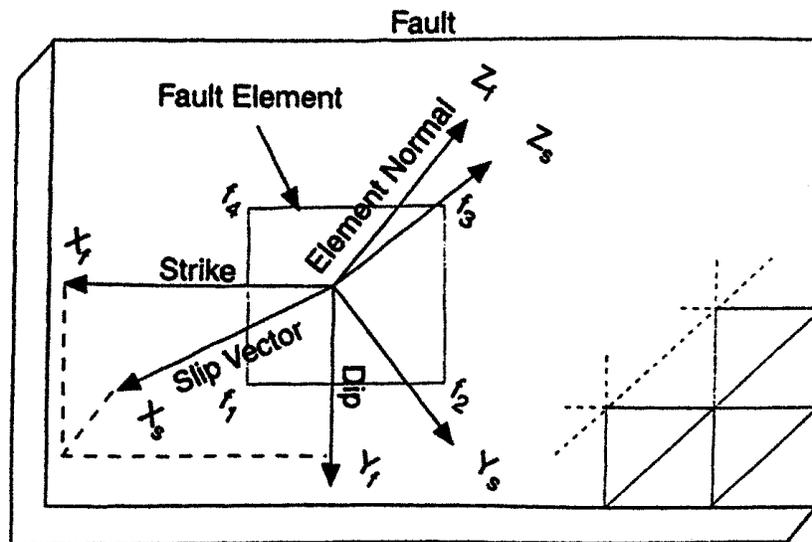
$$Y_f = Z_f \otimes X_f$$

For each fault element, a slip vector is also specified. The slip vector need not lie in the plane of the fault element (i.e. the X_f - Y_f plane). Using the fault element coordinate system and this slip vector, a *fault-slip coordinate system* $\{X_s, Y_s, Z_s\}$ can be calculated as follows:

$$X_s = \text{slip vector}$$

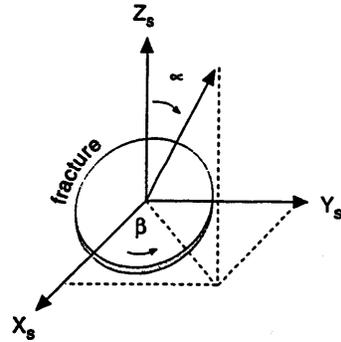
$$Y_s = Z_f \otimes X_s \quad (\text{Special Case: If } X_s = \pm Z_f, \text{ then } Y_s = \pm Y_f)$$

$$Z_s = X_s \otimes Y_s$$



Fracture orientation is then calculated in the fault slip coordinate system using two orientation angles α and β . These orientation angles constrain the fracture pole unit vector, N_f , as follows:

α = angle between N_f and Z_s
 β = angle between X_s and the
 projection of N_f onto the
 X_s - Y_s plane



$$N_f(X_s) = \sin\alpha \cos\beta$$

$$N_f(Y_s) = \sin\alpha \sin\beta$$

$$N_f(Z_s) = \cos\alpha$$

Note that for $\alpha = 0$, the X_s and Y_s components of N_f are zero and β is of no significance.

In FracMan, a number of Group 1 subsets have been predefined to simplify user input. Each subset is distinguished by unique orientation angles α and β . Some subset definitions make use of the user-specified conjugate fault angle, γ .

Subset	α	β
D	0°	0°
X1	$90^\circ - \gamma/2$	0°
X2	$90^\circ - \gamma/2$	180°

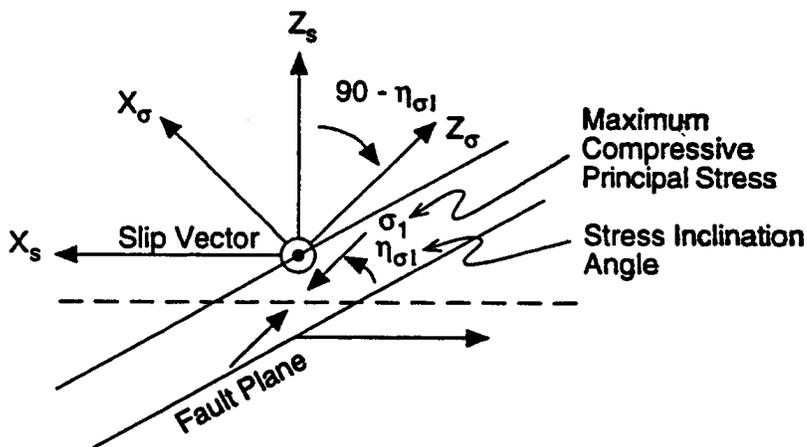
For Group 2 fractures the orientations α , β and the conjugate angle γ are calculated as follows:

The inclination of the largest compressive principal stress relative to the X_s - Y_s plane is given for each fault element by the angle η_σ . A fault-stress coordinate system $\{X_\sigma, Y_\sigma, Z_\sigma\}$ is defined by a counter-clockwise rotation about Y_s of $90^\circ - \eta_\sigma$ such that:

$$X_\sigma = X_s \cos(90^\circ - \eta_\sigma) - Z_s \sin(90^\circ - \eta_\sigma)$$

$$Y_\sigma = Y_s$$

$$Z_\sigma = X_s \sin(90^\circ - \eta_\sigma) + Z_s \cos(90^\circ - \eta_\sigma)$$



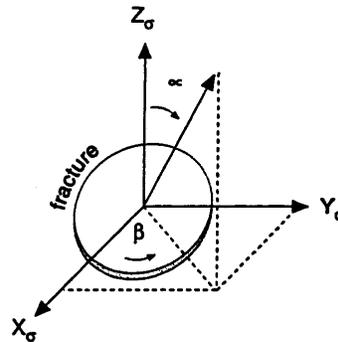
Fracture orientation is then calculated in the fault-stress coordinate system using two orientation angles α and β . These orientation angles constrain the fracture pole unit vector, N_f , as follows:

α = angle between N_f and Z_σ
 β = angle between X_σ and the projection of N_f onto the X_σ - Y_σ plane

$$N_f(X_\sigma) = \sin\alpha \cos\beta$$

$$N_f(Y_\sigma) = \sin\alpha \sin\beta$$

$$N_f(Z_\sigma) = \cos\alpha$$



Note that for $\alpha = 0$, the X_σ and Y_σ components of N_f are zero and β is of no significance.

In FracMan, a number of Group 2 subsets have been predefined to simplify user input. Each subset is distinguished by unique orientation angles α and β . Some subset definitions make use of the user-specified conjugate fault angle, γ .

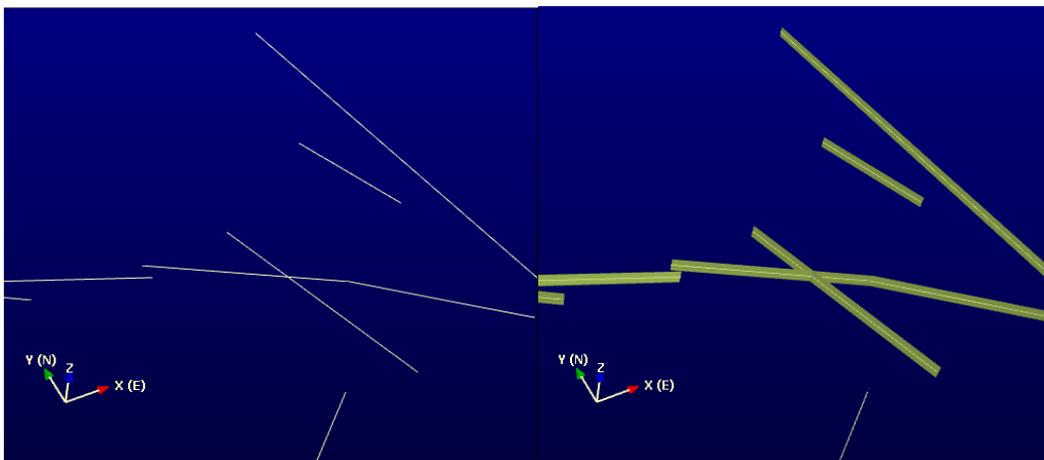
Subset	α	β
T	90°	90°
S	0°	0°
P	$\gamma/2$	0°
R1	$90^\circ - \gamma/2$	0°
R2	$90^\circ - \gamma/2$	180°
C1	$90^\circ - \gamma/2$	90°

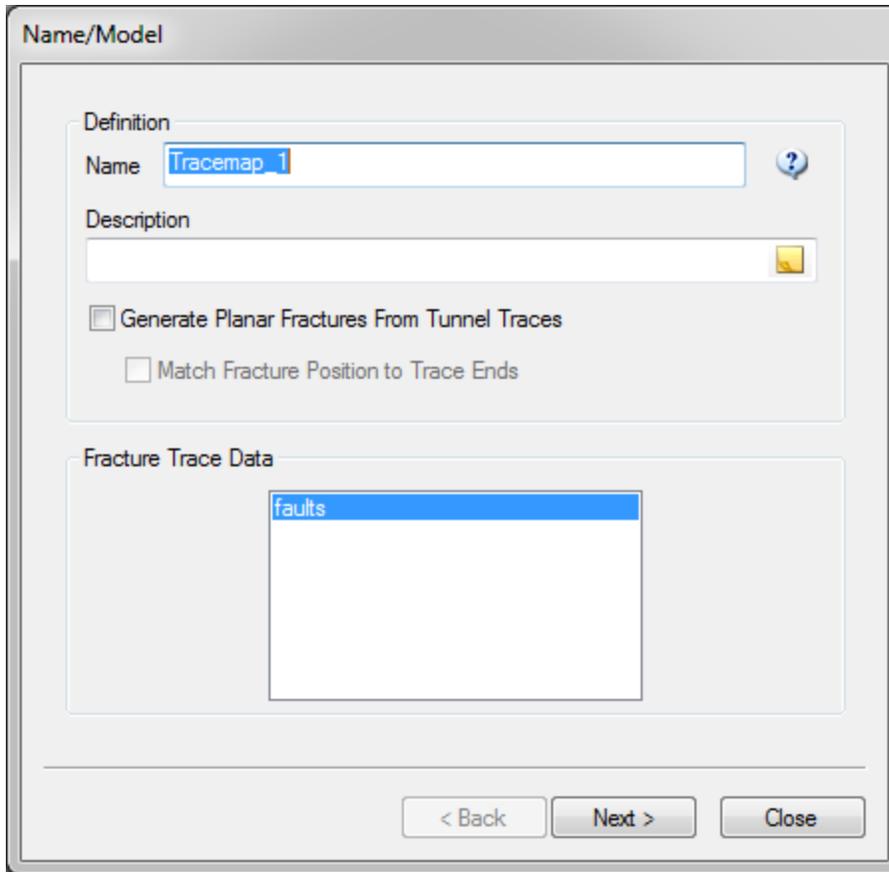
C2	$90^\circ - \gamma/2$	-90°
----	-----------------------	-------------

Group 3 fractures represent the hanging wall, the footwall, or the midplane surface of a fault element. The orientation and geometry of a Group 3 fracture is therefore completely constrained by the element in which it is generated. For triangular fault elements, a single fracture is generated with vertices identical to the element's top or bottom triangular surface. For quadrilateral fault elements, two fractures are generated, the first identical to the triangle formed by nodes f_1 , f_2 and f_3 and the second identical to the triangle formed by nodes f_1 , f_3 and f_4 . This triangularization is necessary because the surfaces of a fault element are required only to be approximately planar, while FracMan requires all fractures to be strictly planar.

Tracemap Fracture Set

This method takes a tracemap and builds a basic fracture set based on the traces, such that the intersection between the fracture set and the surface containing all the fractures would be identical to the traces themselves.





Specify a name for your fracture set definition in the “Name” box.

Fractures may be generated from **Tunnel Traces**, which allows fast generation of planar fractures which closely approximate those observed in the tunnel itself. In this mode, a single planar polygonal fracture is generated from each tunnel trace that is fit to approximate the orientation, size and location of the tunnel trace. These parameters are determined in the following manner:

Orientation: If the tunnel trace is mostly collinear, the orientation of the fracture is determined from the trace dip distribution specified by the user. If the trace is not collinear, the orientation of the fracture is determined by the trace geometry.

Size: The size of the fracture is determined by a distribution specified by the user or the minimum size needed to enclose all the trace segments whichever is greater.

Location: The polygon is then positioned such that trace segments are entirely enclosed by it and the endpoints of the trace lie on the edge of the fracture polygon.

Choose a tracemap from the list under “Fracture Trace Data” that you wish to generate a fracture set to match, and click **Next** to proceed to define the fracture orientation and size (for non-tunnel traces).

Tracemap Orientation

Because the trends of the fractures are determined by the fracture traces themselves, the scalar property defined here represents the dip direction and dip angle of the fracture relative to the plane containing the fracture traces. By default, the fracture dip direction is defined by the trace strike azimuth, but can also be defined by one of the tracemap properties. Fracture dip angle can either be defined as a scalar distribution or correlated by a function to one of the tracemap's properties.

Trace segments have uniform orientation: ensures that all traces are similar along their entirety with respect to tracemap plane. If not checked, generated fractures may change shape.

The screenshot shows a 'Geometry' dialog box with the following settings:

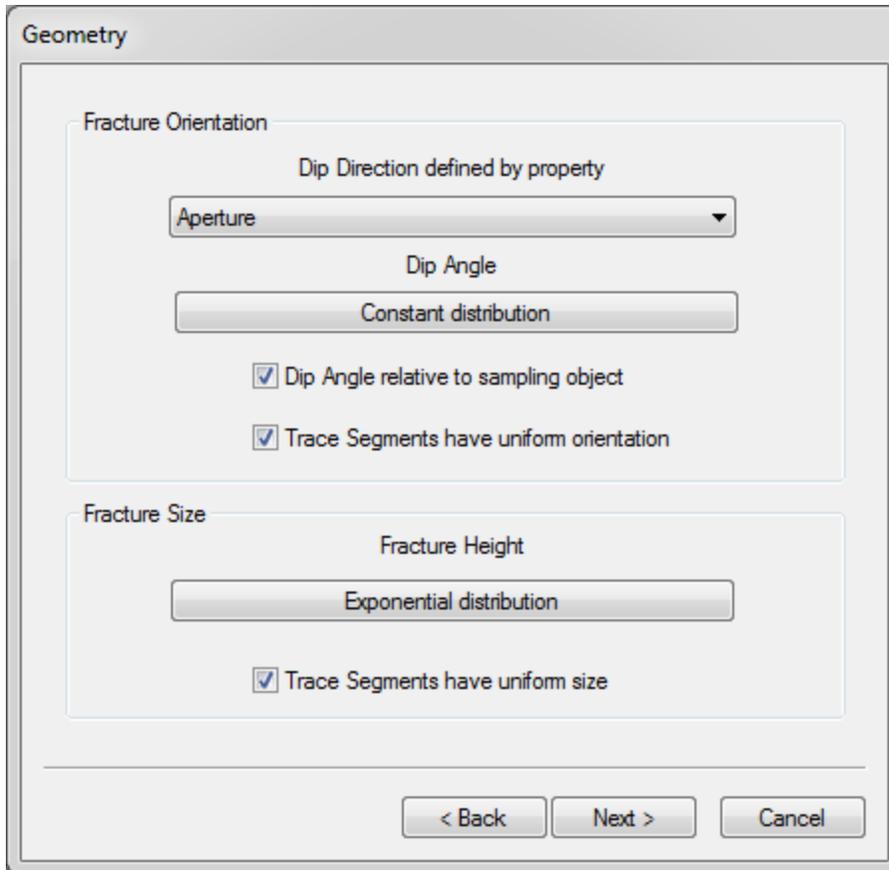
- Fracture Orientation:**
 - Dip Direction defined by property: Aperture
 - Dip Angle: Constant distribution
 - Dip Angle relative to sampling object
 - Trace Segments have uniform orientation
- Fracture Size:**
 - Fracture Height: Exponential distribution
 - Trace Segments have uniform size

Navigation buttons at the bottom: < Back, Next >, Cancel.

Tracemap Fracture Size

This is a scalar property distribution that can be correlated to a tracemap property. As the length of the fractures has already been determined by the traces, this property tells FracMan how tall to make the fractures. The traceplane will be centered along the fracture height axis, with the fracture extending away from the traceplane an equal distance on both sides.

Trace segments have uniform size: ensures that all traces are similar along their entirety with respect to tracemap plane. If not checked, generated fractures may twist.



Click **Next** to proceed to define the fracture properties.

Tracemap Fracture Properties

Finally, define your fracture properties. Fracture properties can be correlated to any properties on the corresponding trace or defined by distribution. Properties are not automatically inherited from the originating tracemap traces. If the same properties are needed in the fractures, the user must first create new properties in the tracemap fracture set definition (or add them later) and correlate these to the original properties in the tracemap. The property name used in the new fracture set must be different from that used in the original tracemap object.

When you have completed the definition wizard, hit **Finish**. The fracture set definition will appear in the Definitions window; you can generate the set by right-clicking on the definition name and selecting **Generate Set**.

Stratigraphic

Fractures generated by the stratigraphic model are created in layers. Each layer is defined by a pair of grid layers. Users can create a primary set or one primary set plus some secondary sets of fractures. The primary set is the fractures generated within grid layers. The secondary sets are fractures confined by the primary set. In general, the principal directions of primary and

secondary sets are orthogonal to each other. The principal directions cannot be parallel to each other, and the secondary set cannot be truncated by the primary set.

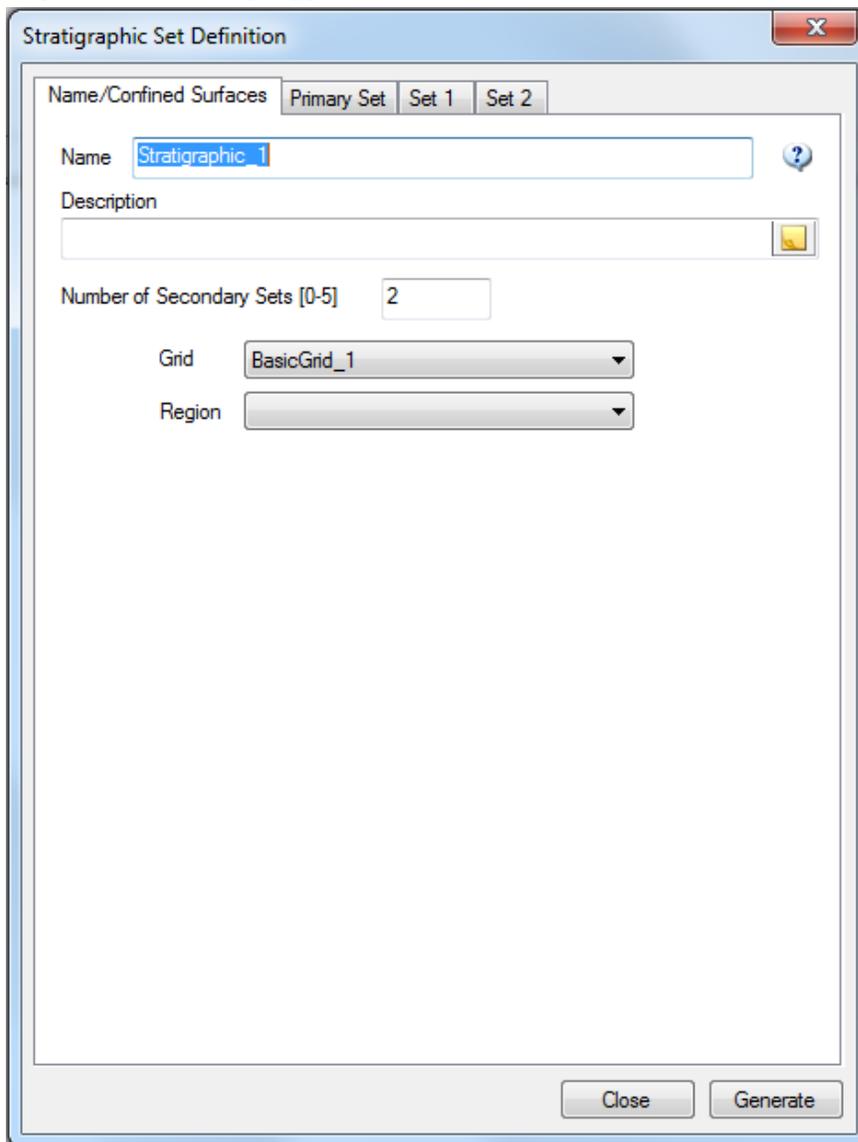
Name/Confined Surfaces

Specify a name for your fracture set definition in the “Name” box and optionally a Description. The key input parameters for the stratigraphic model are grid layers, fracture spacing and fracture termination.

Number of Secondary Sets: Enter the number of secondary sets desired, between 0 and 5. Click “Apply” to show tabs for any secondary sets required.

Grid: Select the Grid to use. The K-layers in the grid will be used as the stratigraphic horizons. If you want to use surfaces to define these horizons, create an appropriate grid using the “Grid from Surfaces” functionality.

Region: not currently supported.



The image shows a software dialog box titled "Stratigraphic Set Definition". At the top, there are tabs for "Name/Confined Surfaces", "Primary Set", "Set 1", and "Set 2". The "Name/Confined Surfaces" tab is active. Inside the dialog, there is a "Name" text box containing "Stratigraphic_1" and a "Description" text box which is empty. Below these is a "Number of Secondary Sets [0-5]" field with the value "2". There are two dropdown menus: "Grid" set to "BasicGrid_1" and "Region" which is currently empty. At the bottom of the dialog are "Close" and "Generate" buttons.

Primary and Secondary Sets

Stratigraphic Set Definition

Name/Confined Surfaces Primary Set Set 1 Set 2

Fracture Intensity

Spacing

A= Constant distribution (0) B= 1

Fracture Orientation / Property

Fisher distribution (85, 5) Orientation is Pole

Assign Fracture Properties..

Fracture Length and Height

Length = M + N * Thickness M Length Distribution.. N 1

Fracture height in bed thickness Min 90 % Max 110 %

Fracture Termination

A: Stop Growth at Both Beds		100 %
B: Stop Growth at One Bed		0 %
C: Grow Within Two Beds		0 %
D: Grow Beyond Both Beds		0 %

Normalize Height Types A-D

Close Generate

Fracture Intensity

Fracture intensity can be defined by Spacing and Fracture Count (total number of fractures generated).

Spacing: Fracture spacing is a function of bed thickness, which is normally inversely proportional to fracture intensity. The following equation is used to correlate fracture spacing to thickness:

Spacing = $A + B * \text{Thickness}$, $A \geq 0$, $B \geq 0$, $A+B > 0$ (both cannot be all 0's)

Fracture Count: Specify the total number of fractures to include in the set. The Intensity field is not currently supported.

Fracture Orientation/Property

The Orientation specified is a vector distribution of the pole or dip of the fractures. See the [Orientation](#) section for complete details on defining the orientation of fractures.

You can specify your fracture properties by clicking on the “Assign Fracture Properties” button. To change the definition of these properties, select the desired property and click the **Edit**

Property button . A new window showing the current property parameters will pop up.

Click the **Add Property** button  to add an additional property to the fracture set definition. Again, a new window will pop up, which allows you to specify the property distribution.

You can delete an existing property by selecting the property and clicking the **Delete Property** button . The three default properties cannot be deleted.

Fracture Length and Height

Fractures are generated as quadrilaterals before they may be truncated to other shapes by bedding surfaces or other fractures. Fracture size is defined by its length and height. Fracture length is determined by the following equation:

$$\text{Length} = M + N * \text{Thickness}$$

where M is the fracture length distribution. The value for N is entered by the user and controls the proportionality of fracture length to bed thickness (Thickness). The fracture length is defined by a stochastic distribution, which is selected by clicking on the “Length Distribution” button. The length in the secondary fracture sets is often affected by fracture termination.

Fracture height is measured along the direction of bedding thickness, and is defined by the minimum and maximum percentages of bed thickness (BT). For example, for bed thickness = 100m, min = 75% and max=150% then the minimum fracture height (MinHt) is 75m, and the maximum height (MaxHt) is 150 m.

The height is related to fracture growth. If it is Type A: Stop growth at both beds, fracture height always equal to bed thickness. For Type B: Stop growth at one bed, fracture height is ranged from MinBT less than BT. It is in [75 m, 100 m] in this case. For Type D: Beyond both beds, fracture height is beyond BT up to MaxHt, i.e. [100m, 150m].

The distribution of how many fractures in Type A-D is controlled by user's input parameters. Users enter the distribution percents that do not necessarily sum to 100%. The program will

rescale these to 100% during computation or users can hit the button “Normalize Height Types A-D” to view. For example, if the input data are 1, 2, 2, and 5. The percents will be adjusted to 10%, 20%, 20% and 50% respectively.

During fracture generation if the set is newly defined both primary and secondary sets would be generated together. While in generation from the selected set, or from the definition panel the set can be generated separately. Although a secondary set is generated alone it is still constrained by its primary set.

Fracture Termination

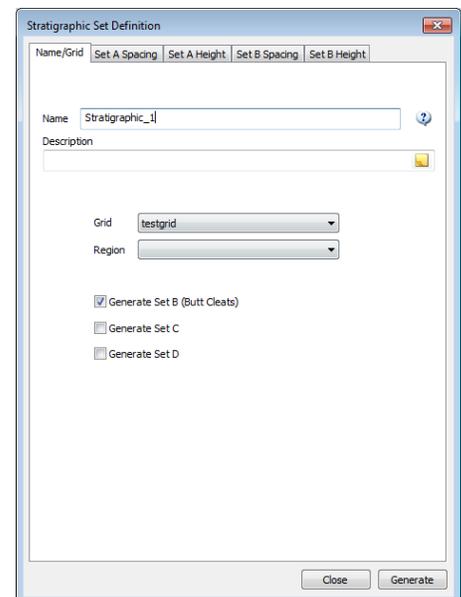
Fracture termination is only applied to the secondary fracture sets. This parameter controls how much a secondary fracture can grow beyond a primary fracture set.

Layered Fracture Spacing Model

This model has replaced the older stratigraphic model for most purposes as it provides more features and higher performance while generating more realistic fractures. Some of the features of this model

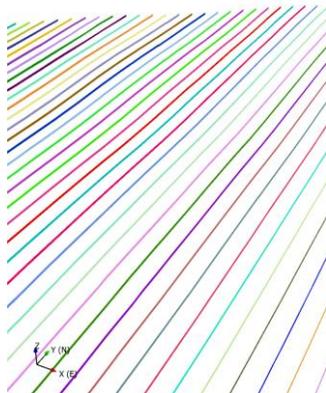
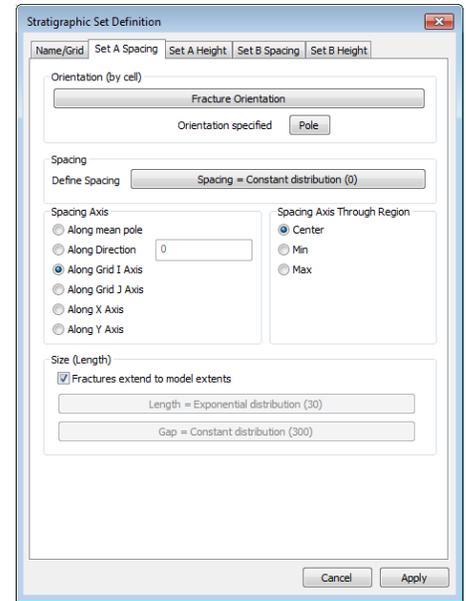
- Fractures are generated in layers as defined by a grid
- Intensity is determined by spacing which can be correlated to a grid property
- Fractures from a given set do not cross
- Orientation is grid cell based and can vary spatially, allowing fractures to follow topology, stress or geologic formations

Definition name, description etc are specified on the first page of the dialog. Then select a grid for the generation region. The “Region” box allows selection of a grid property that defines which cells are used for fracture generation. Grid layers with no non-zero values are completely ignored, cells set to zero in a layer that has non-zero values will have the portions of fractures in that cell marked as non-persistent which excludes them from other analyses. Finally, select up to three secondary sets to be generated – these sets have additional options for terminating against previous sets.

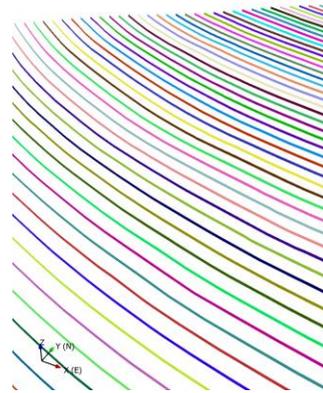


For each set there will be two pages, one for the orientation and spacing and length and one for the height, vertical position and properties.

Orientation can be defined by either the Pole or Dip vectors and trend/plunge can be defined separately either by random variable or by a correlation to a grid property. The orientation is calculated for each cell in the grid so fractures within the same cell will have the same orientation – this prevents fractures from a single set from crossing even though their orientation may change through the model.



Constant Orientation

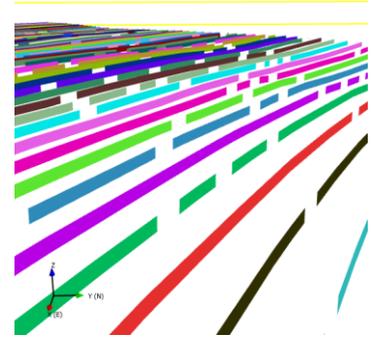


Variable Orientation

Spacing can also be defined randomly or correlated to a grid property (often grid cell height). When you are working with fractures of varying orientation, the question of where is the spacing specified is an important one. FracMan allows selection of the spacing axis to be in the direction of the set's mean pole vector, along the x, or y axis directions, along the grid's I or J axis directions or along any azimuth you specify. Additionally the spacing axis can run through the center of the model or through the minimum or maximum position of the spacing direction. Selecting the mean pole and the center are usually the safest options but there are situations for instance if spacing data comes from an oriented traceplane where you will want to use a spacing axis that is as close as possible to the original data source.

Fractures generated by the Layered Spacing Model are inherently infinite in length. They penetrate all the way through the generation region. FracMan then uses surface persistence (a surface property that works like “Active” in a grid) to split each surface into fracture sections and non-fracture “gap” sections. The length of each of these can be controlled using the length and gap buttons.

Generated fractures have various properties that allow you to access individual virtual fractures as well as virtual fracture length properties.

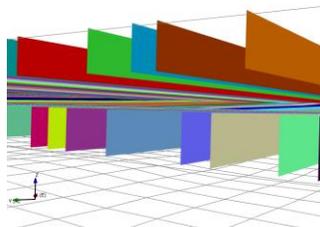
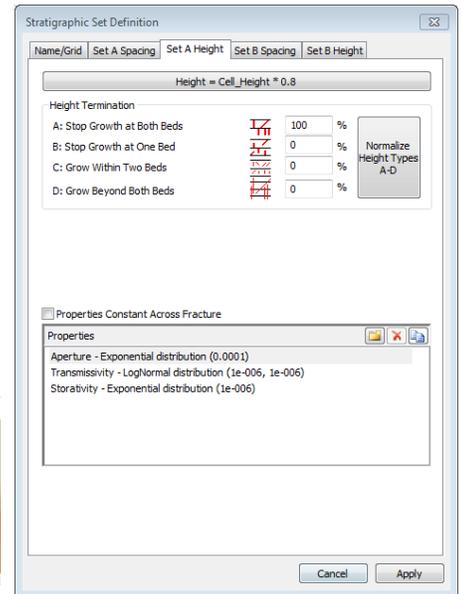
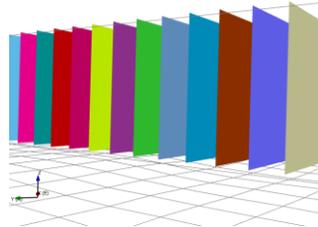


The next page controls the height of the fractures. For layer bound fractures where all fractures extend from the top to the bottom of the layer enter 100% in box A and then specify a constant height which will essentially be ignored. More often, fracture height is correlated to cell height along with specification of how many of the fractures terminate at both beds, at one bed, non-terminated and beyond both beds.

Height Termination:

Case A Stop Growth at Both Beds

Fracture height is ignored and all fractures extend between the bounding layer boundaries

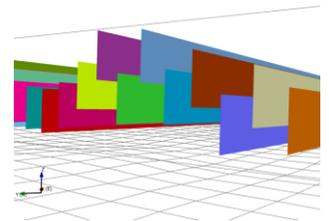


Case B Stop Growth at One Bed

Fractures are generating using the height distribution and are locked to either the top or bottom layer boundary (regardless of size)

Case C Grow Within Two Beds

Fractures are limited to 90% of the cell height and are positioned anywhere within the layer boundaries





Case D Grow Beyond Both Beds Fractures are allowed to extend up to beyond the fracture boundaries (fracture center is constrained to be within the layer)

Like the other values, fracture properties can be defined either as a stochastic value, correlated to one or more grid properties, or as a combination of both. Fracture properties can be constant across a given virtual fracture or they can be computed for each fracture triangle.

Stress

This generates fractures based on stress field properties as defined by a grid. Geomechanical codes (e.g. Elfen, PFC3D) generally require considerable time to run fragmentation simulations on discrete models with fracturing. Understanding how the fragmentation process evolves from these codes would allow for the implementation of the development of stress-induced fractures within an initial DFN model stochastically. Therefore we need to capture some basic rules of fragmentation that will allow the stochastic simulation of fracturing during cave development and the tracking of the evolution of fragmentation.

The Stress fracture generation algorithm aims to capture the first order effects of fragmentation by simulating fracture growth during a progressive series of time steps for which the broad stress history is known.

The stress field (principal stresses) is specified in the grid properties by three directional magnitudes and major and minor directions in (trend, plunge). The maximum stress direction (the resultant of the principal vectors) is the main direction along which wing cracks and new tensile cracks will be generated.

Fracture Set Parameters 1

Fracture Set Definition
Name Crack Set

Global Fracture Intensity
Measure Type Fracture Count

Generation Region
 Region
 Grid hyStress
 Clip on outer region

Fracture Generation Order
Shear Extension Cracks First
Tensile Wing Cracks Not used
New Tensile Cracks Not used

< Back Next > Cancel

Fracture Set Definition Parameters 2

Fractures:

All Fractures

Use Filter:

Principal Stresses

	Magnitude	Trend	Plunge
Major (Sigma 1)	<input type="text" value="Sigma1"/>	<input type="text" value="Sigma1_Trend"/>	<input type="text" value="Sigma1_Plunge"/>
Minor (Sigma 3)	<input type="text" value="Sigma3"/>	<input type="text" value="Sigma3_Trend"/>	<input type="text" value="Sigma3_Plunge"/>
Intermediate (Sigma 2)	<input type="text" value="Sigma2"/>		

Shear Extension Cracks ✕

Selected by Shear force ▼ Fracture Count 20

Termination % (stop on intersection?) 0

Fracture Size

Extend fractures by 1.25*Area 

Fracture Shape

Aspect Ratio Constant distribution

Elongation Axis Constant distribution

< Back Next > Cancel

Tensile Wing Cracks

Selected by **Compressive force** Fracture Count 20

Termination % (stop on intersection?) 0

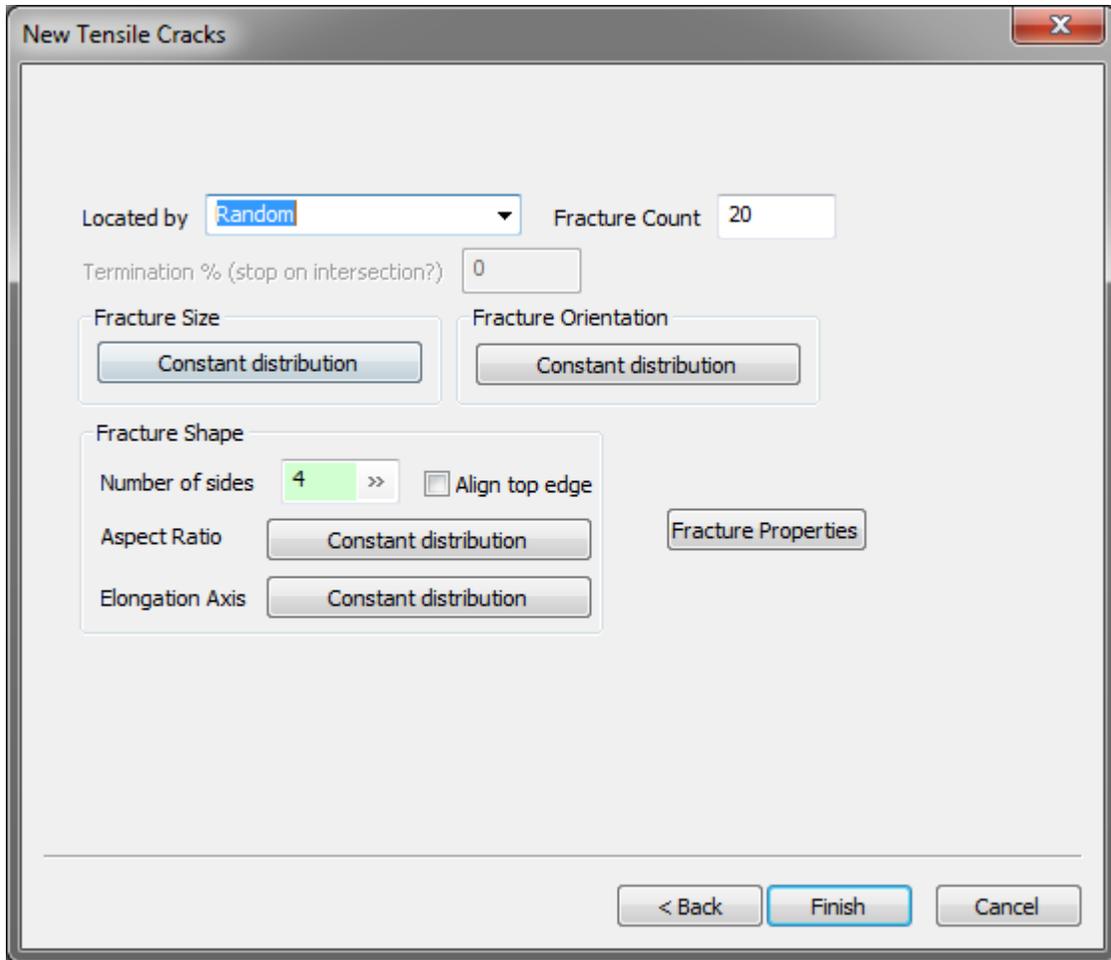
Fracture Size
Extend fractures by 0.25*Area

Fracture Shape
Aspect Ratio **Constant distribution**
Elongation Axis **Constant distribution**

Fracture Orientation
Constant distribution

Fracture Properties

< Back Next > Cancel

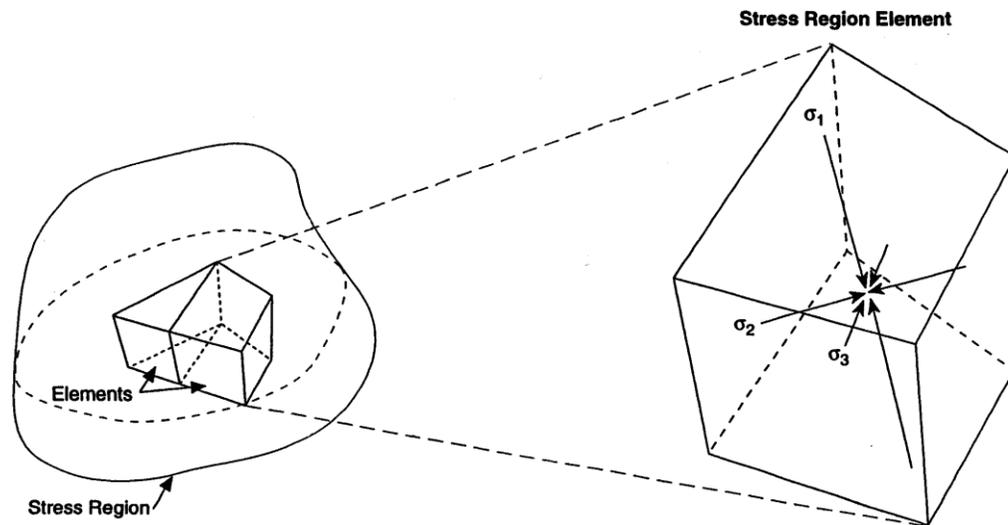


Stress Model

The stress field model allows the generation and sampling of fractures conditioned to the magnitude and orientation of *in-situ* rock stresses. The user defines a stress field over a generation region. Within this region, fracture networks can be generated with intensities and orientations controlled by the stress field.

In FracMan, a stress field is comprised of one or more *stress regions*. A stress region is a finite volume of arbitrary shape and size, which has been discretized into one or more prismatic elements with triangular or quadrilateral cross sections. The intensity of fracturing in each stress cell is controlled by a quantity termed the *fracture potential*, which is a measure of relative probability for fracture occurrence. In the stress model, fracture potential is a function of linear combination of principal stress magnitudes (e.g., mean normal stress, maximum differential stress, greatest compressive principal stress, etc.) plus up to two user-specified scalar intensity parameters. Fracture orientation is controlled by the local orientations of the

principal stress vectors. The theory underlying these relations is described in greater detail below.



Note that fracture potential determines the relative number of fracture *generation points* received by an element and not the number of fractures or amount of fracture surface area. If fractures are of significant size relative to element dimensions, they may span more than one element, possibly smoothing other measures of fracture intensity in the process.

Stress Field Fracture Intensity

Fracture intensity can vary throughout a user-defined stress field. The user specifies the *average* fracture intensity over the entire stress field in terms of the fracture porosity, P_{33} (fracture volume/unit volume), fracture intensity, P_{32} (fracture area/unit volume), or the total number of fractures. Associated with each stress cell is a fracture potential which determines the relative likelihood that it will receive a newly-generated fracture. Elements with higher fracture potentials will tend to have more fractures generated in them. The total fracture potential of a cell is calculated as the product of individual fracture potentials arising from some linear combination of principal stress magnitudes plus an arbitrary user-specified scalar intensity parameter. Because the stress field across each grid cell is assumed uniform, the distribution of fractures within a particular cell is uniform.

Stress Field Fracture Orientation

Fracture orientation is controlled by the local orientations of principal stress vectors. If the stress field is represented by a stress tensor, this needs to be transformed into principal stress components using the Edit->Grid->Compute->[Convert Tensor to Principal Vector](#) function before generating fractures with this method (note that compressive stress is positive while tensile stress is negative).

For isotropic stress fields, the principal stress orientations, and hence, the fracture orientations are random. For non-isotropic stress fields, fracture orientation is calculated as follows. The

stress tensor at the fracture generation location is given by the magnitudes and directions of the three principal stresses ($\sigma_1, \sigma_2, \sigma_3$). From this stress tensor, a *local stress coordinate system* $\{X_\sigma, Y_\sigma, Z_\sigma\}$ is created such that:

$X_\sigma = \sigma_3$ direction

$Y_\sigma = \sigma_2$ direction

$Z_\sigma = X_\sigma \otimes Y_\sigma = \sigma_1$ direction

Fracture orientation is calculated in the local stress coordinate system using two orientation angles α and β . These orientation angles constrain the fracture pole, N_f , as follows:

α = angle between N_f and Z_σ

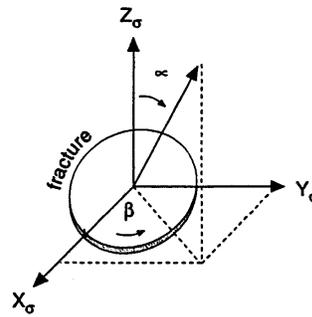
β = angle between X_σ and the projection of N_f onto the X_σ - Y_σ plane

X_σ - Y_σ plane

$$N_f(X_\sigma) = \sin\alpha \cos\beta$$

$$N_f(Y_\sigma) = \sin\alpha \sin\beta$$

$$N_f(Z_\sigma) = \cos\alpha$$



Note that for $\alpha = 0$, the X_σ and Y_σ components of N_f are zero and β is of no significance.

In FracMan, a number of subsets have been predefined to simplify user input. Each subset is distinguished by unique orientation angles α and β . Some subset definitions make use of the user-specified conjugate fault angle, γ .

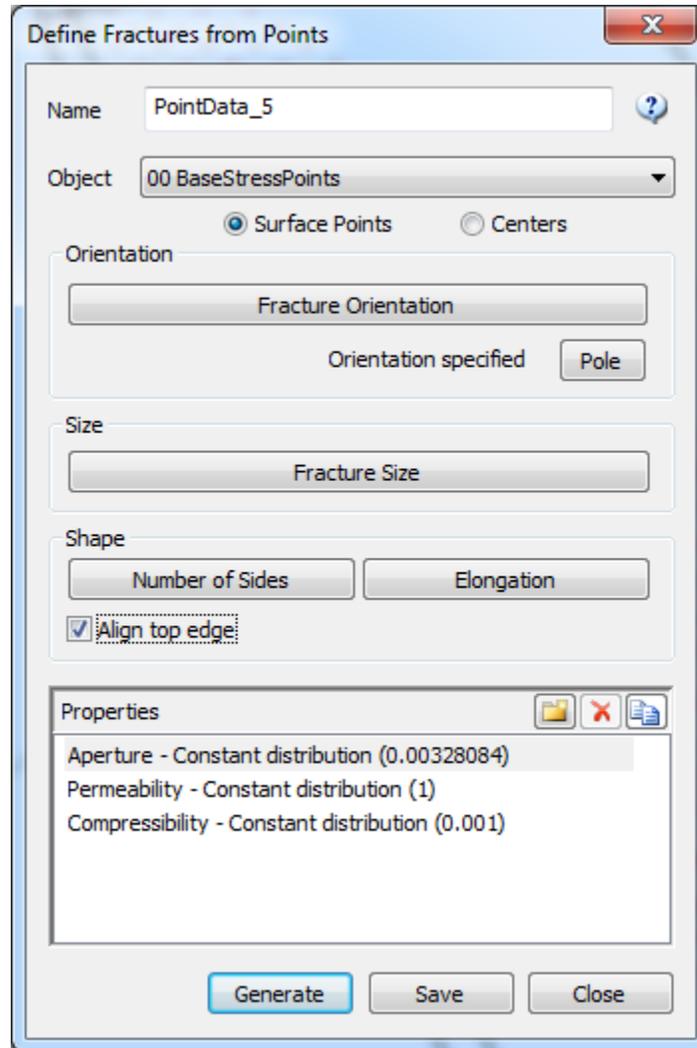
Subset	α	β
T	90°	0°
S	0°	0°
D1	$90^\circ - \gamma/2$	90°
D2	$90^\circ - \gamma/2$	-90°
C1	$90^\circ - \gamma/2$	0°
C2	$90^\circ - \gamma/2$	180°

Create from Point Data

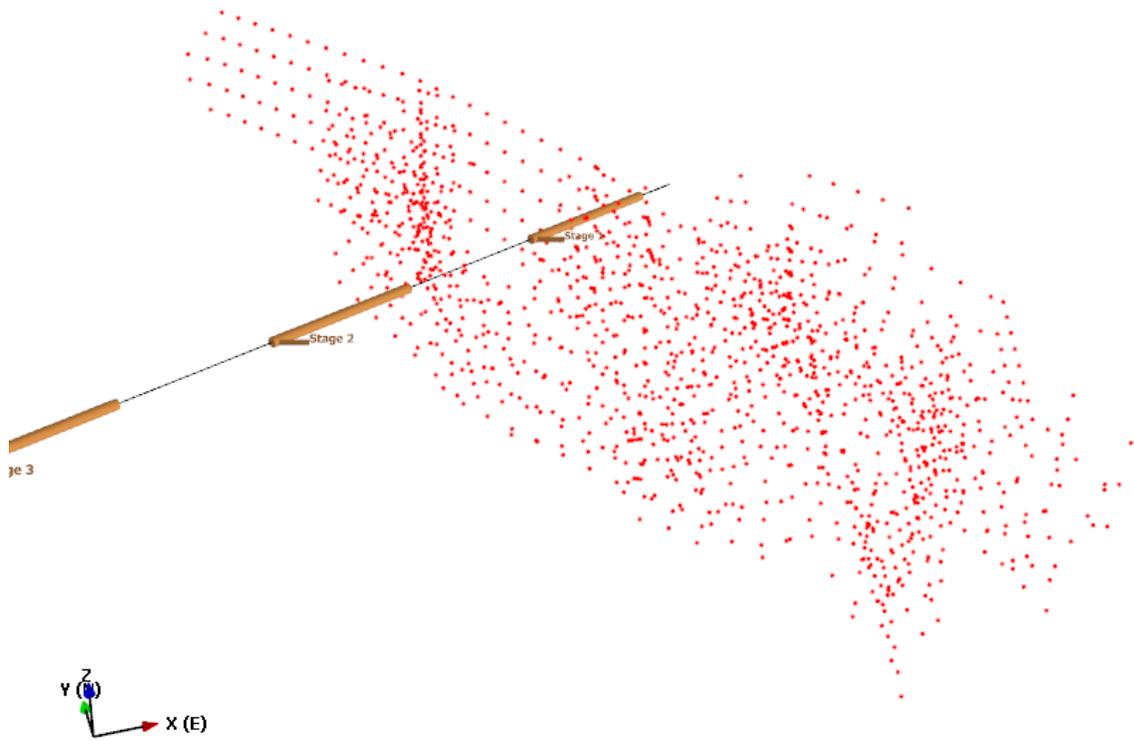
When location is known but other parameters for creation of a DFN are still needed, setting up a DFN set definition based on point data can now be used in FracMan. In the case of

microseismic, the data can show the location of potential fractures. The point data serve as a generation point for the fracture where the Set Definition to create a DFN from point data can help fill in the missing pieces of information.

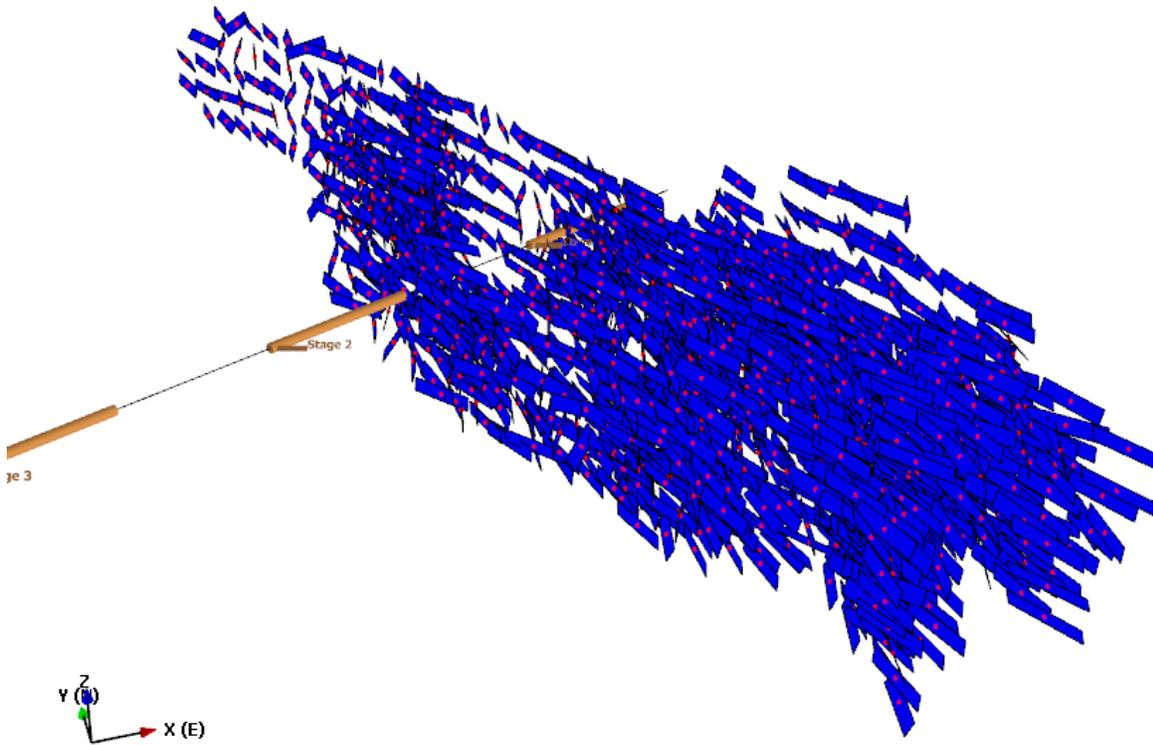
The user specifies the point data set, and other parameters used in other set definitions (ie. Size and orientation). Once the parameterization is set and the DFN generated from the point data set definition, the fractures will be generated in the point data locations. It should be noted that if Surface Points are selected, the only a part of the fractures will not be centered on the point data, where if center is selected, then the fracture centroid will be centered on the point data.



Point data before fracture generation



Point data and fracutres with surface points selected



Condition To Traces

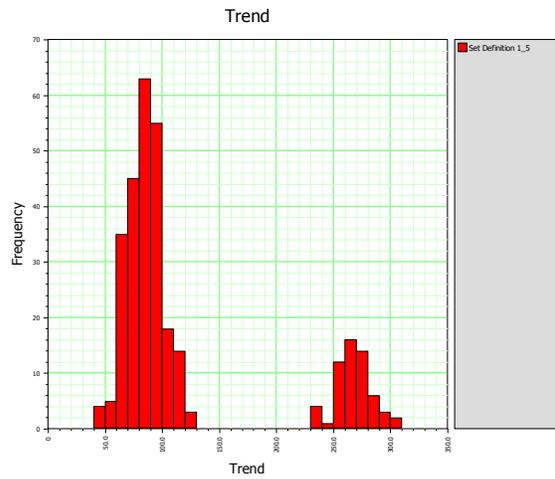
Create Pseudo Matrix Fracture Working with Charts

The most basic analysis available to the FracMan user is the graph. There are a number of different graphs available in FracMan: stereonet, CFI plots, CDFs, rose diagrams, histograms, etc. To make a chart, import a set of data and tell the program what kind of chart you want and FracMan will do the rest.

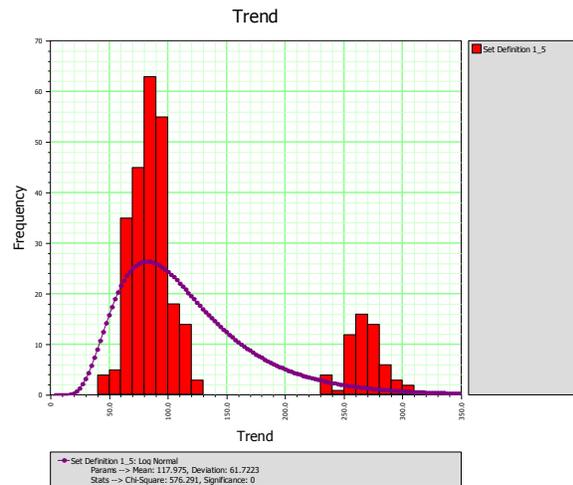
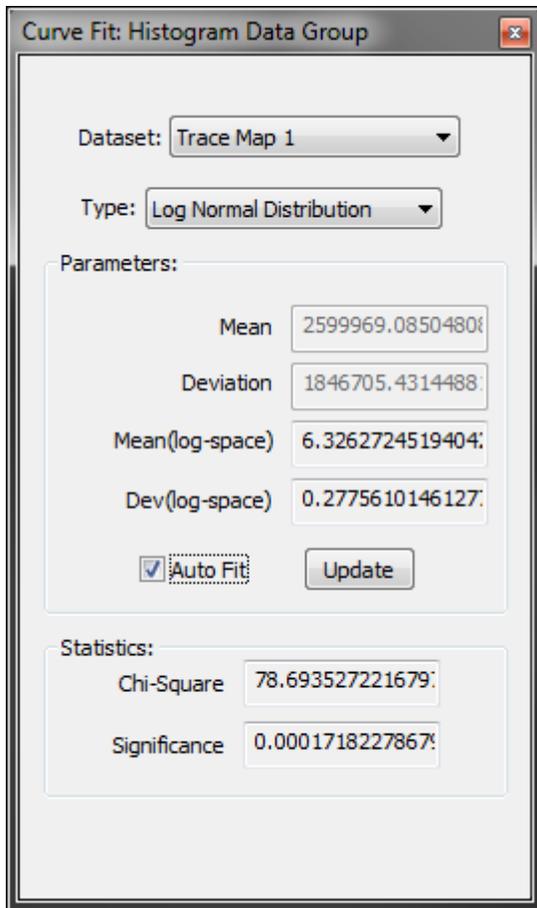
Histogram

A histogram sorts data by the frequency of each value's occurrence. FracMan can generate histograms of properties of objects with multiple datapoints, such as grids, tracemaps, and fracture sets. Right-click on the object name in the Object window and select **Histogram** and the name of the property you wish to plot. A new window will open with the desired histogram.

Note: If the object you select does not have any properties, the Histogram option will not appear in the right-click menu. As new properties are added to the object, the option to graph each property will appear under Histogram.



Curve Fitting. You can fit a curve to the data in order to determine what distribution best matches it. Right click on the graph and select **Curve Fit**.



Above: Histogram with log normal curve fit.

Left: Curve fit dialog box containing properties for above curve.

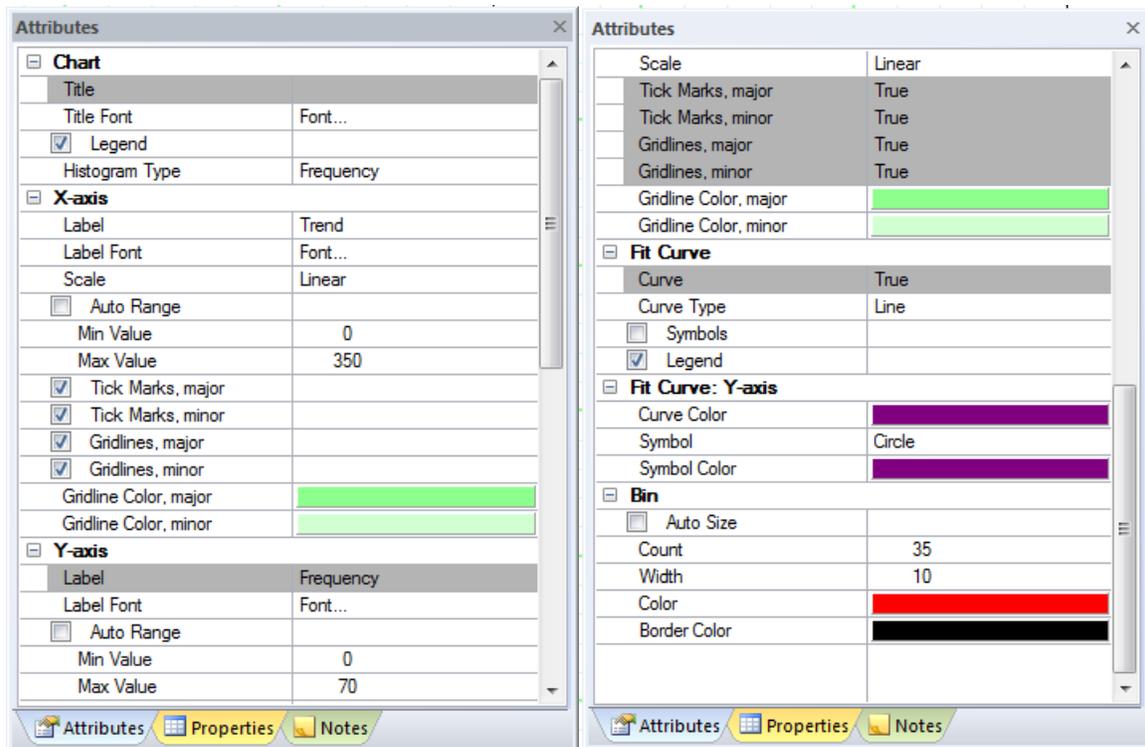
FracMan will automatically fit a curve to your data so long as the box next to “Auto Fit” is checked. If you wish to enter parameters for the curve yourself, uncheck the box and enter your own specifications in the boxes provided. The parameters you must specify change depending on the distribution you are modeling. Once you change the parameters, you must hit the **Update** button for the changes to take effect.

There are eight different distributions which the program can attempt to fit your data to. These are: Power Law, Normal, Log Normal, Normal of Log, Exponential, Poisson, Gamma, and Weibull distributions. To change the fit curve, select the desired distribution from the drop down menu next to “Type”.

At the bottom of the dialog box is a section called “Statistics”. This contains, for distributions, a Chi-square value and a Significance value. The higher the significance and the lower the Chi-square value, the better your curve fits the data. For the Power Law curve fit, the statistics are an SSQ, or the sum of the squares of the data relative to the mean and standard deviation of the data, and a standard deviation error value. The smaller these are, the better the fit.

The drop down menu next to “Dataset” allows you to select which dataset on which to base your curve fit if your histogram contains more than one dataset. You may have a different fit curve for each distinct dataset, but FracMan does not allow you to fit one curve to multiple datasets. You can make a histogram containing multiple datasets by selecting two or more objects in the Object window while holding the Ctrl key. Then right click on one of the sets while all are selected and choose your histogram property. *Note:* This does *not* combine the data into one set. The datasets will appear side-by-side so that they can be readily compared. Each dataset will be in a different color; the key on the right side of the graph tells you which color corresponds to which set. In order to combine the datasets into a single histogram, you must select the “Merge Data Sets” option under the **Chart** tab of the Attributes window. This will combine the data value frequencies into a single set for the purposes of the graph, but not in the Objects window.

Changing the Histogram’s Attributes. In the Attributes window there will be a list of boxes: Chart, X-axis, Y-axis, Fit Curve, Fit curve: Y-axis, and Bin. Click on the title of the box to expand it.



Most of the controls are self-explanatory, such as “Title”, whose field allows you to change the graph’s title. In order to modify a field, click inside the white box and type the value you want. Fields in grey boxes cannot be modified until the “Auto” option above them has been unchecked. For instance, once the “Auto Range” box has been deselected, the grey “Min Value”

and “Max Value” boxes below it will become white, indicating that FracMan will now allow you to enter values of your own choosing into the fields.

In order to change the color of some part of the graph, such as the fit curve, you must select the colored box next to “Color” or the name of the object whose color you wish to change. This will bring up a dialog box in which you may either choose your preferred color from a selection or define your own color. The “Color” field under **Bin** controls the fill color of the histogram’s bars; the “Border Color” field controls their outlines.

Some of the fields allow the user to pick one of several options.

- “Histogram Type” toggles between Frequency – the traditional plot of value versus number of occurrences of that value – and Relative Frequency, which adjusts for the size of the dataset.
- “Scale” allows you to toggle between Linear and Logarithmic scale.
- “Curve Type” toggles between Line and Scatter; in the latter only the symbols are visible along the fit curve and not the line itself.
- “Symbol” toggles what shape is used for the points along the fit curve.
- Other fields contain a box which the user checks to turn a feature on. For instance, if the “Legend” box is deselected, the sidebar of the graph - which gives statistics for a fit curve, or a key telling which color represents which dataset, depending on which legend is specified - will disappear. Similarly, the user can hide the data points along a fit curve by deselecting the “Symbols” box, or hide the curve itself by deselecting the box next to “Curve”.

Right-clicking on the histogram and selecting **View**, you are presented with a list of options: Histogram, CDF, CCDF, and Source Data. Clicking **Source Data** will open a new window containing a list of the values that were used to generate the histogram. By right-clicking on this screen, you can either save the data as its own separate file (**Save As**) or copy it to the clipboard for use in another program (**Copy**). If you select **View > CDF** or **CCDF**, FracMan will change the way your data is displayed without opening a new window.

Cumulative and Complementary Distribution Function Plot

The cumulative distribution function (CDF) and complementary cumulative distribution function (CCDF) are used to delineate mechanical layering and its geological controls. Any property of an object can be graphed in this way. First, make a histogram, then right click anywhere in the histogram window and select **View > CDF** or **CCDF**, whichever you prefer.

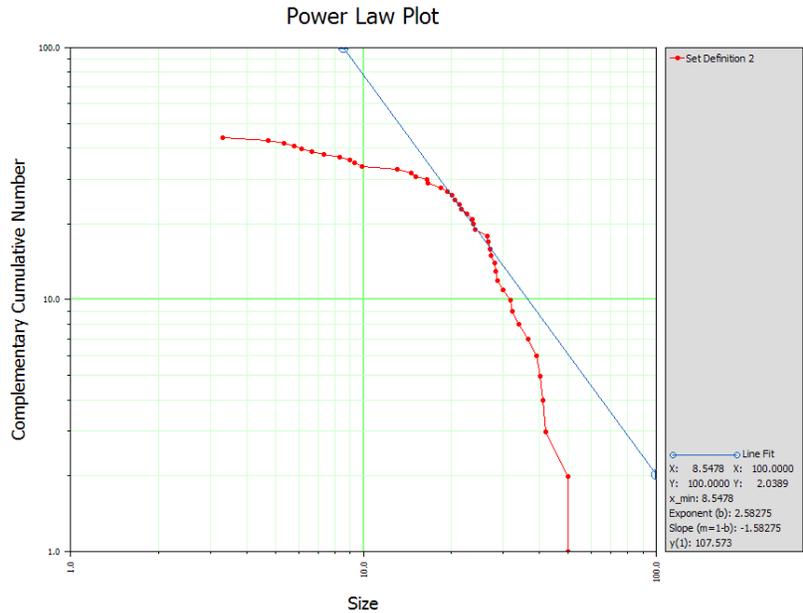
Attributes	
Chart	<<
Title	Trend - CDF
Title Font	Font...
Merge Data Sets	<input type="checkbox"/>
Legend	<input checked="" type="checkbox"/>
X-axis	>>...
Y-axis	>>...
XY Curve	<<
Curve Type	Line
Symbols	<input checked="" type="checkbox"/>
XY Curve: Y-axis	<<
Curve Color	
Symbol	Circle
Symbol Color	

The attributes of a CDF or CCDF differ from those of a histogram only insofar as there are fewer of them. The properties under the **XY Curve** and **XY Curve: Y-axis** tabs are found under **Fit Curve** and **Fit Curve: Y-axis** in the histogram Attributes window.

Power Law Plot

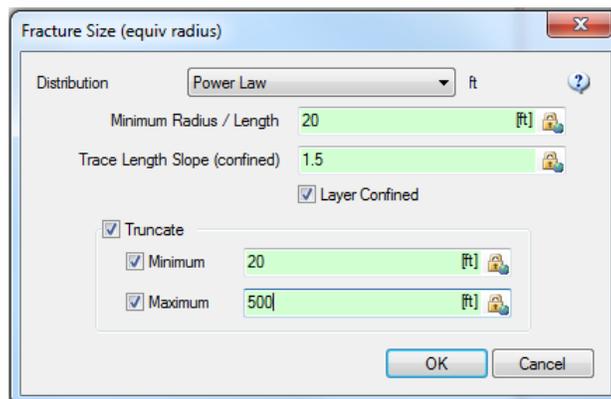
This shows whether the selected data follows a fractal size-intensity scaling law. Power law fits may be done on tracemaps and fracture sets. In order to create a power law plot of a fracture set property, right click on the name of the desired set in the Objects window and select **Analysis > Fit Power Law**. Alternatively, you may highlight the set in the Objects window and select **Data Analysis > Spatial Patterns > Power Law Plot** from the menu bar. You will be given a list of the set's properties. Choose which one you wish to graph and hit **OK**. In order to create a power law plot of trace length for a tracemap, right click on the tracemap name in the Objects window and select **Plot Power Law Function**. A new window will open containing your power law plot.

There will be a black line in the plot area with a circle at each end. This is the Fit Line, and it allows you to quickly determine the slope and other parameters of portions of your curve. You can move the fit line by dragging and dropping the endpoints. As you change the Fit Line the parameters in the lower right corner of the graph window will update to match the new line.



The only power law plot attribute not common to the histogram is the Y-axis “Scale” field, which makes it possible to toggle between Log and Linear scale on the Y-axis as well as on the X-axis. This allows you to make Log-Log plots of your data as well as Semi-log and Linear plots.

NOTE: The exponent for radius distributions (e.g. layer confined) should be one greater than the exponent for trace length distribution (La Pointe, 2002, “Derivation of parent fracture population statistics from trace length measurements of fractal fracture populations”, Int. J. Rock Mech. Min. Sci., Vol. 39, pp. 381-388). For example, if the slope of the trace length power law distribution is -1.5, the exponent for the trace length distribution (Dt) is 1.5, and the exponent for the radius distribution (Dr) is 2.5. This is automatically calculated in the dialog for entering a Power Law fracture size distribution when defining fracture sets by checking the option “Layer Confined”:



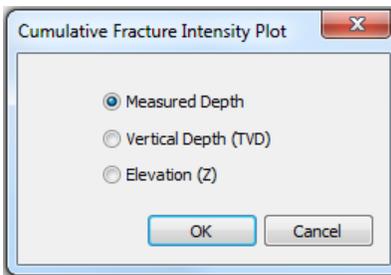
Cumulative Fracture Intensity Plot

A cumulative fracture intensity plot, or CFI, shows the percentage of total fractures in the well fracture log versus the depth of the fractures. This allows you to see if there are any depth

intervals – for instance, a particular stratigraphic unit – in which the fracture intensity is significantly higher or lower than elsewhere, or to make clear any breaks in fracture intensity.

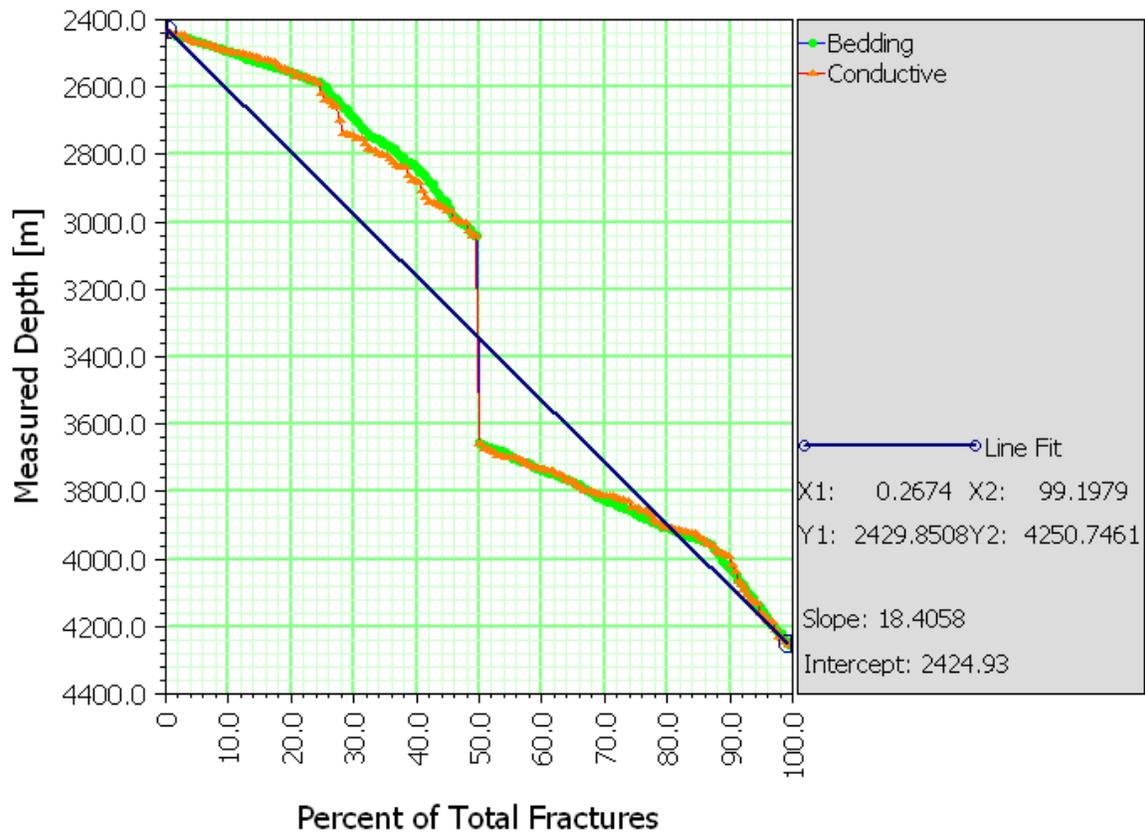
FracMan only supports CFIs for well fracture logs. Right click on the name of the log in the Objects window and select **Analysis > Cumulative Fracture Intensity Plot**. Alternatively, you may highlight the log in the Objects window and select **Data Analysis > Intensity > Cumulative Fracture Intensity Plot** from the menu bar. This will open a dialog box asking you to specify the units of your Y-axis.

- **Measured Depth (MD):** Depth along the well trace.
- **Total Vertical Depth (TVD):** Depth from the surface.
- **Elevation (Z):** Distance below sea level.



Select the proper depth units; then hit **OK**. A new window containing your chart will appear.

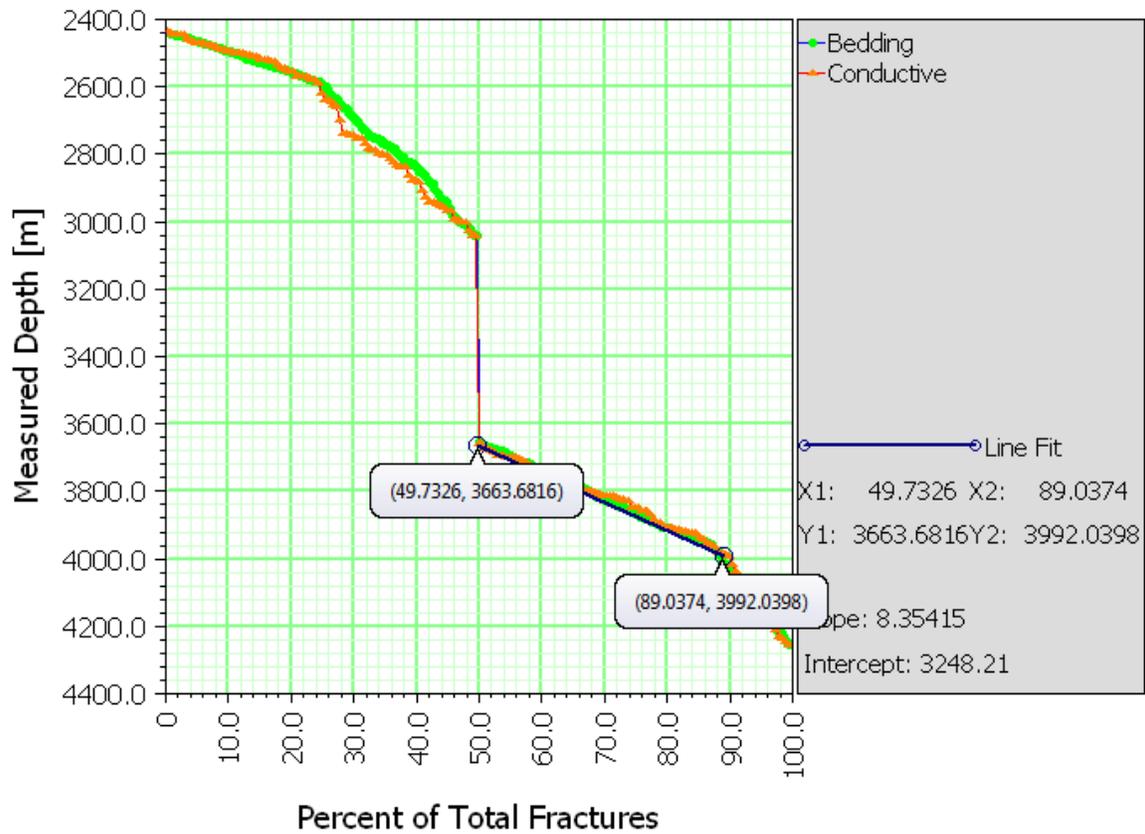
Cumulative Fracture Intensity Plot



Like the power law plot, the CFI has a fit line that can be moved around the window to obtain the slope and intercept of different portions of the graph. This fit line also has a feature not found in the power law: it can be used to calculate fracture intensities (P_{10}) over a particular interval. Drag the endpoints of the fit line to the boundaries of the area over which you wish to derive a P_{10} , and then click near the line at the same Y-axis level as the endpoints. A bubble containing a set of coordinates will appear corresponding to each point. If a bubble is too far from its intending counterpart, you can click inside the bubble and the point will be reset.

Once you have two satisfactory points selected, right click anywhere inside the graph window and select **Transfer Coordinates**. A new window will open containing depths of your endpoints, the calculated P_{10} of the log segment between them, and the name of the data series from which the P_{10} was obtained ("Segment ID"). You can save this information as its own file or copy it to the clipboard by selecting any line of the data and right clicking. If you wish to add more P_{10} intervals to the list, return to the graph window *without* closing the "Saved Points" window, and repeat the process as many times as you desire. This will create a statistics view that contains the end points from the line which can be saved. *Note:* Once you close out the Saved Points window, the list will be lost unless you copy or save the points, even if the CFI window is still open.

Cumulative Fracture Intensity Plot



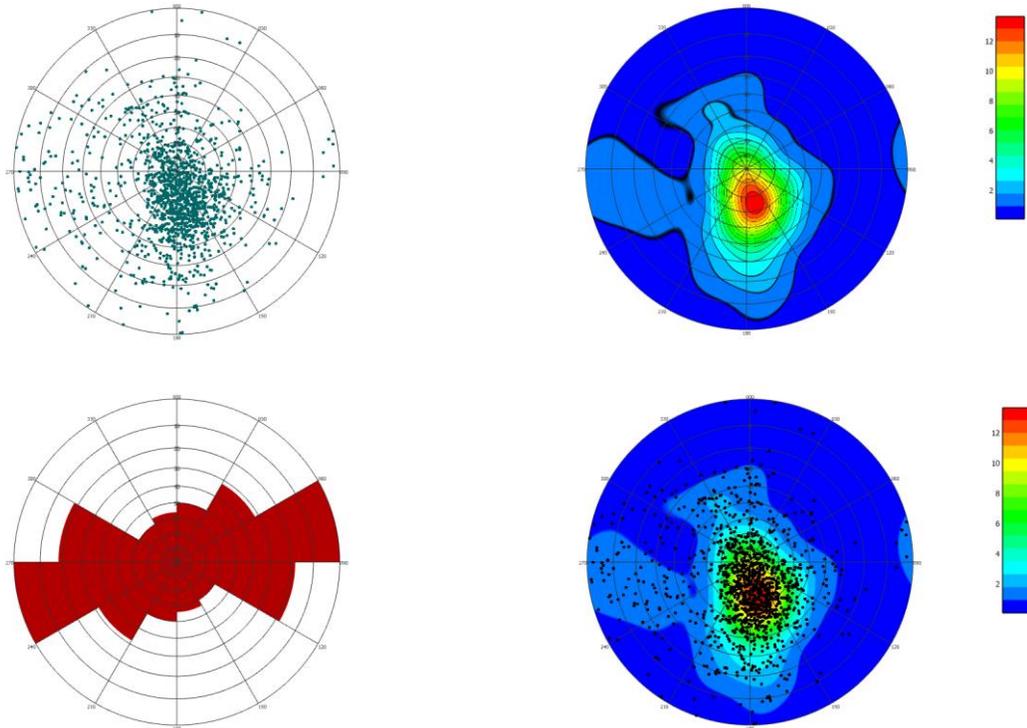
Segment ID	Starting Depth	Ending Depth	P10
Bedding_1	2444.776	2589.055	1.576
Bedding_2	2618.906	3016.916	0.574
Bedding_3	3673.632	3972.139	1.215
Bedding_4	3992.040	4245.771	0.375

You may make a CFI plot with more than one set of data by selecting multiple well logs in the object window. Data will appear on the same plot as separate lines. There are no attributes unique to the CFI plot.

Stereonet

A stereonet visualizes the orientations of a set of planar features – be they fracture poles or bedding planes – as if they were projected onto a sphere viewed from above. The stereonet's radii show the trend of the feature's pole, while the concentric circles indicate the plunge.

To make a stereonet, right click on a fracture set or well fracture log and select **Analysis > Stereonet**. This will open a new window containing your stereonet.

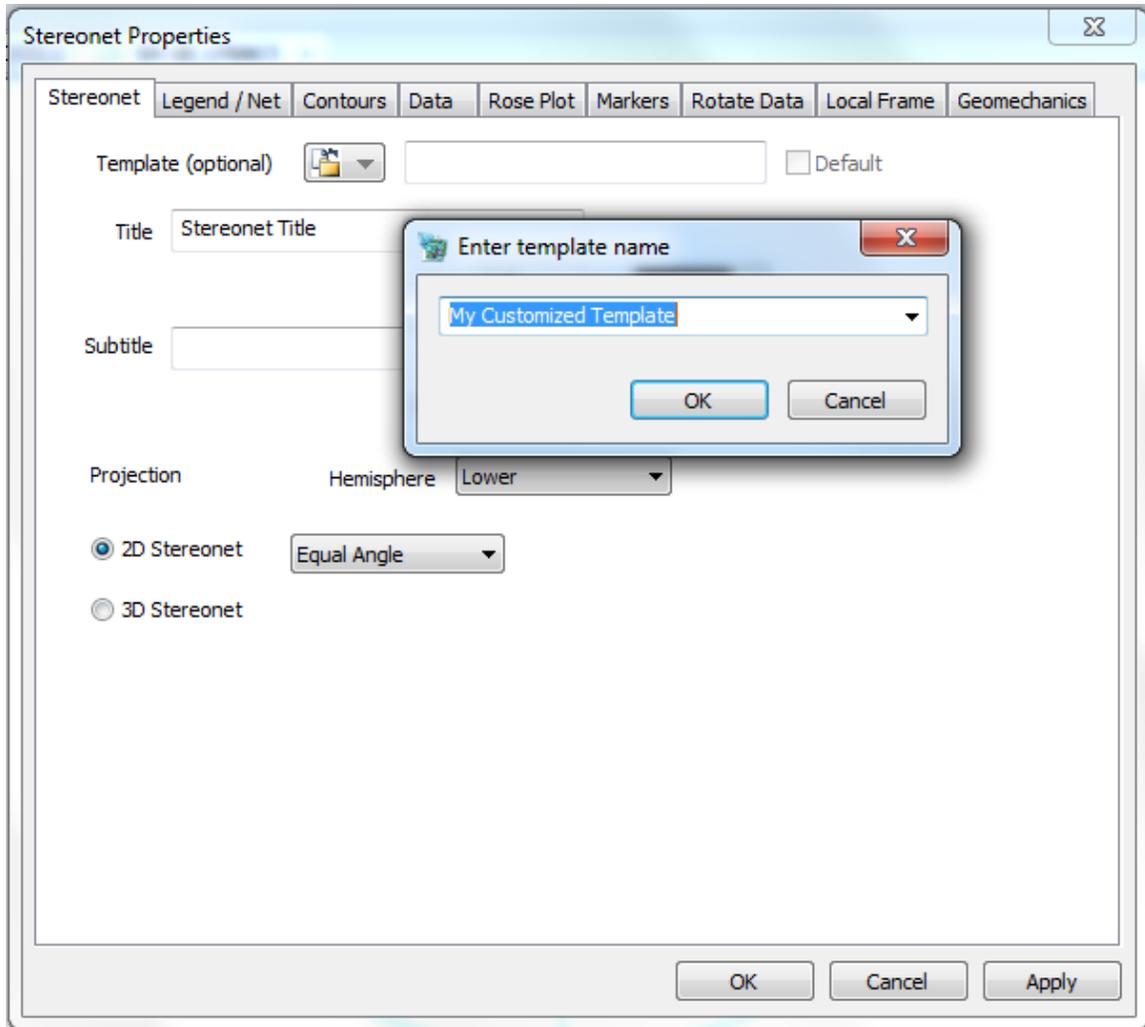


Clockwise from top left: Scatter plot, Contour plot with contour lines overlaid, Contour plot with points and contour lines, Rose diagram oriented by strike.

Stereonet Templates

Stereonets may be customized using all the options available in the **Stereonet Properties** dialog. It is often useful, however, to use templates to store common options to save time and so that produced stereonets have a common look. Two templates are provided by default, “Contour Plot” and “Rose Plot”. To load a stereonet template, click on the **Template (optional)** button and select the template by name. It will appear in the text box next to the selection box.

New templates customized by the user can also be saved. Create a stereonet having all the desired properties and on the **Stereonet** tab, click on the **Template (optional)** button and select **Save Settings**. The small dialog Enter template name will open. Enter a name for the new template and click **OK**.



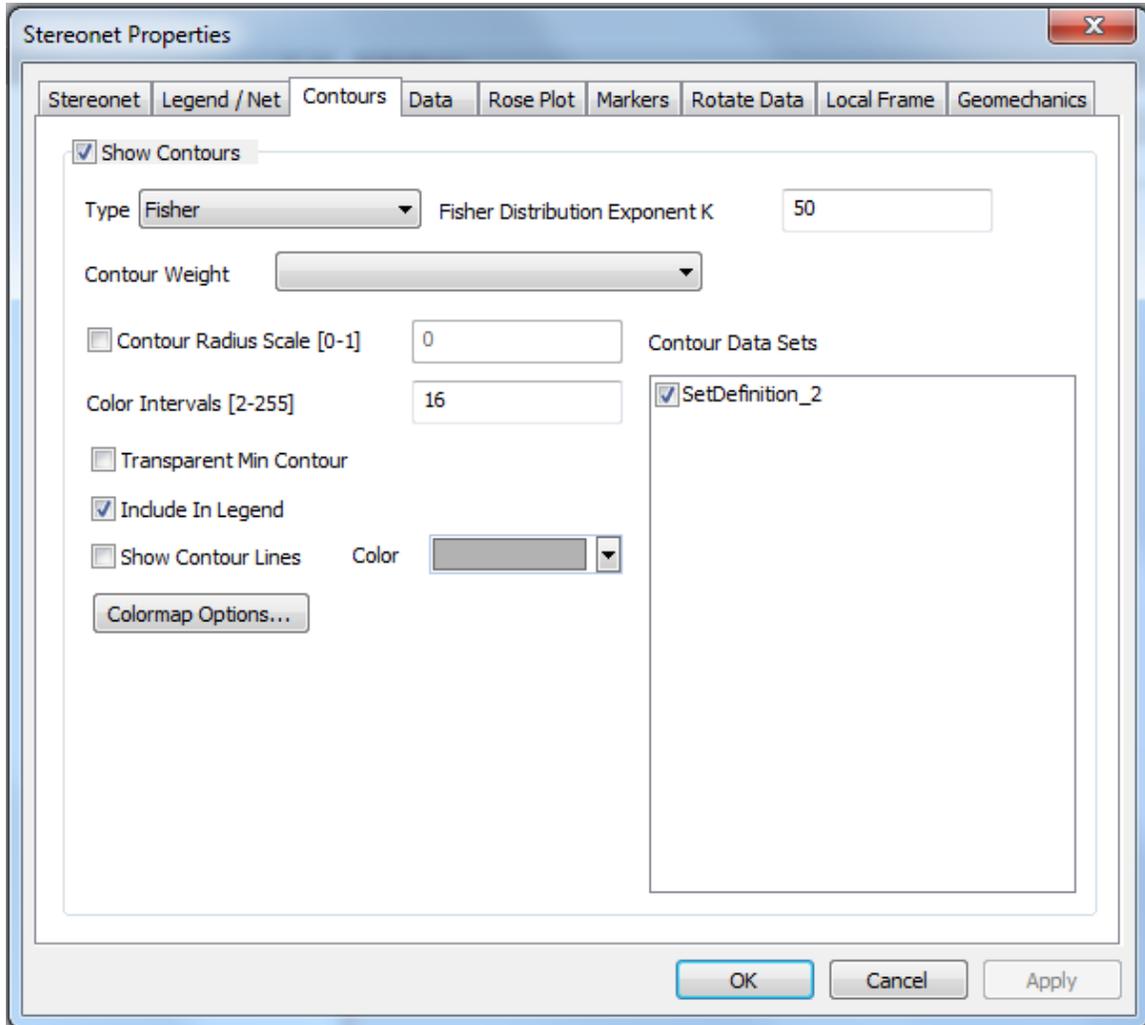
If you want to always start with a particular stereonet template loaded when opening the **Stereonet Properties** dialog, select the template and then check the **Default** checkbox. Click on **Apply** to save this as the default. The next time you open the **Stereonet Properties** dialog, the default template will be pre-loaded. To remove having a default template, uncheck the **Default** option and again click on **Apply**. The next time that the **Stereonet Properties** dialog is opened, the standard default settings will be in effect.

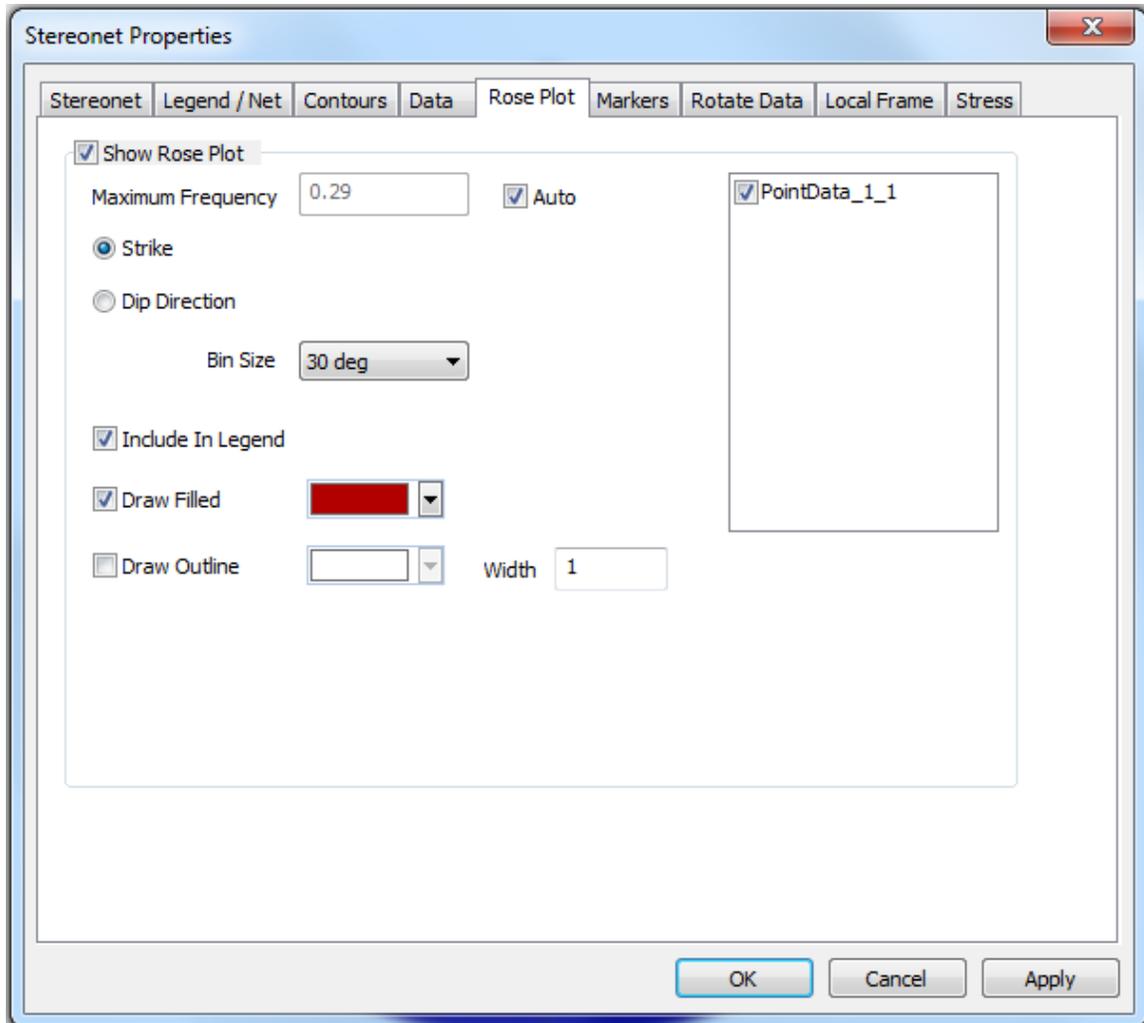
All stereonet templates are stored as files in the default directory "C:\Users\Public\Documents\FracMan7\ChartTemplates". If you want to delete one of your saved templates, simply delete the file in this location.

Changing the Type of Stereoplot

There are three types of stereoplot: Scatter, Contour, and Rose. Contours can be added to your stereonet by right-clicking on the stereonet window and clicking **Show > Show Contours**. Contours and rose plots can also be added to the stereonet display in the Stereonet Properties window. Open the Stereonet Properties window by right-clicking anywhere in the stereonet

window and selecting “Properties...” Depending on the type of stereoplot you wish to display, check the box next to “Show Contours” and/or “Show Rose Plot” under the **Contours** and/or **Rose Plot** tabs (respectively). Click **Apply** to visualize the changes.





Choosing Sets by Orientation

There are two different approaches to selecting subsets based on orientation data in FracMan: an algorithmic approach implemented using the built-in functionality called Set Identification, and a manual approach based on user visual identification. The method using Set Identification is described in detail in the Manual section [Set Identification](#). Here we describe the manual method of set identification. Subsets may be created for well logs, fracture data and point data.

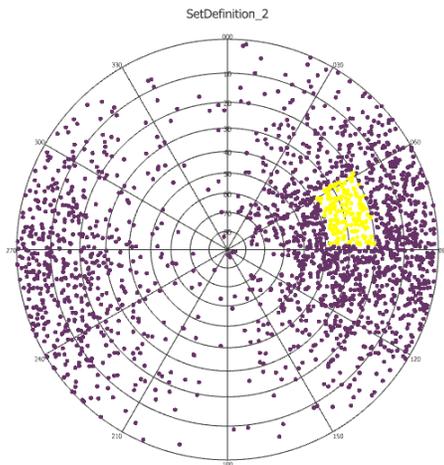
The first step in creating subsets by orientation is to select plotted data points on the stereonet that belong to a subset. This is done using the Select functionality available by right-clicking on the stereonet. The available methods include:

None: Used to de-select all currently selected stereonet points. This can also be done by clicking on the stereonet outside of the plotted range. It also stops the cursor from selecting data points on the stereonet.

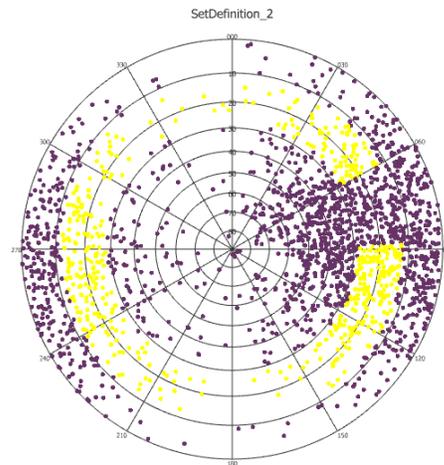
Single: Selects the nearest point to the mouse click.

Rectangular: Selects all points within a rectangular area delineated by dragging the cursor from one corner to another along a diagonal of the rectangle.

Polar: Selects all points within a region defined by a range of plunge and trend as defined by dragging the mouse along a diagonal. Behavior is different if the cursor is dragged in a clockwise vs counter-clockwise direction (selected points are shown in yellow):



Clockwise



Counter-clockwise

The second step in creating the subsets is to assign the selected stereonet points to a group. This can be done by right-clicking on the stereonet, ensuring that the desired points for inclusion are currently selected, and choosing one of several options in the Groups category:

Create Group: Add selected points to a new group with the user supplying a name.

Assign Selection To: Add selected points to an existing group, if any, and if none are already defined, to a new group with a default name. Once points have been assigned to a group, their assignment can be seen as a column value in the results displayed by right-clicking on the stereonet and selecting Analysis | View Data.

Once a group is defined, two other grouping functions are enabled:

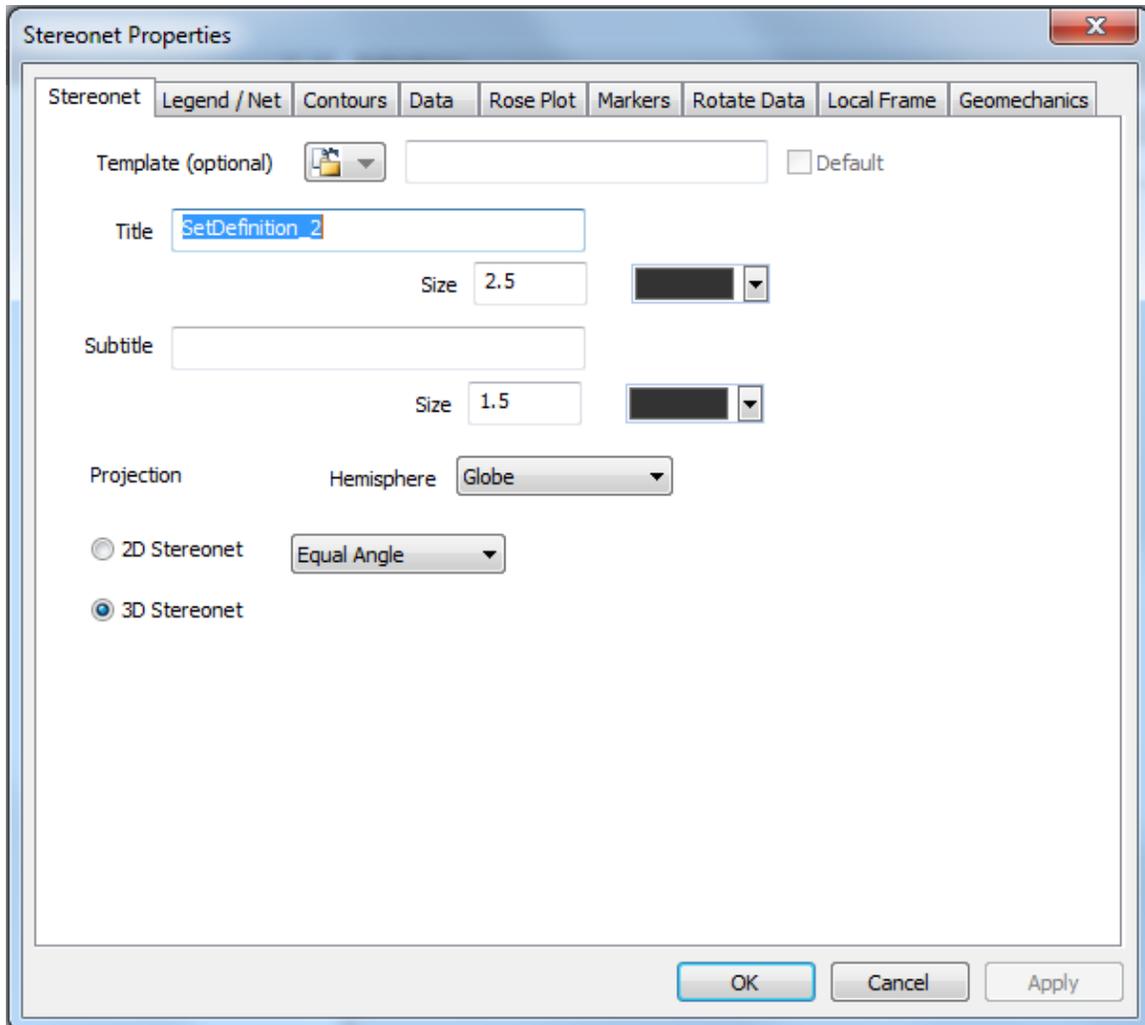
Group Unassigned Points: Add all remaining points that are not included into a group into the nearest group.

Clear Groups: Delete any existing groups.

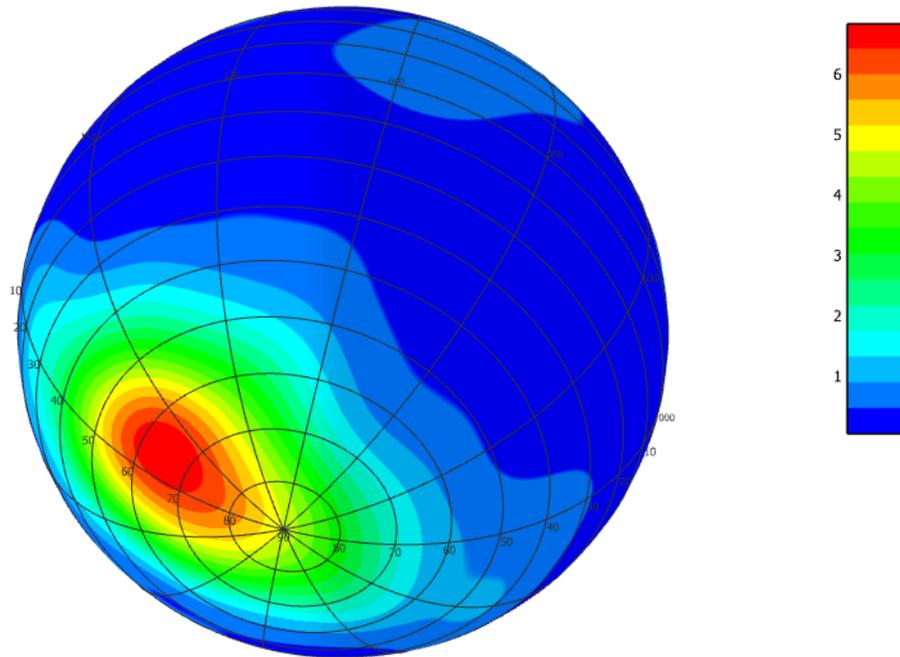
Once the group(s) are defined, the final step is to create the subset(s) consisting of the newly identified sets. This is accomplished by right-clicking on the stereonet and selecting Groups | Create Subsets. The new subset(s) will appear as child objects in the Object window.

3D Stereonet

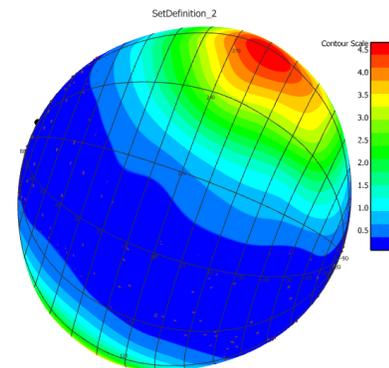
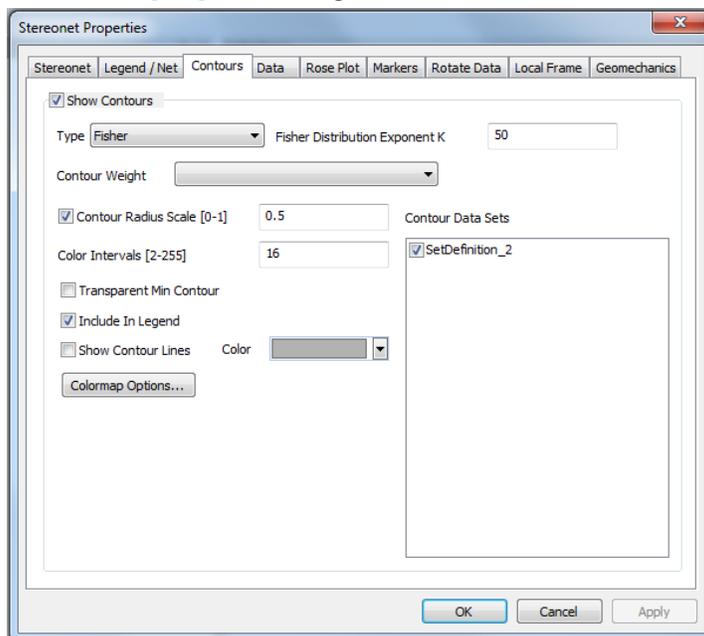
The user has the option to view a 3D version of the stereonet. Once you have made a stereonet of your data, right-click anywhere in the stereonet window, select “Properties...” and under the **Stereonet** tab click the radio button next to “3D Stereonet”. When the 3D stereonet is enabled, the user will be able to view the 3D globe as Upper and Lower Hemispheres, as well as the entire 3D globe (selected from the drop-down menu next to “Hemisphere” in the Stereonet Properties window).



The image in the display window can then be rotated by holding the left-click mouse button and dragging.



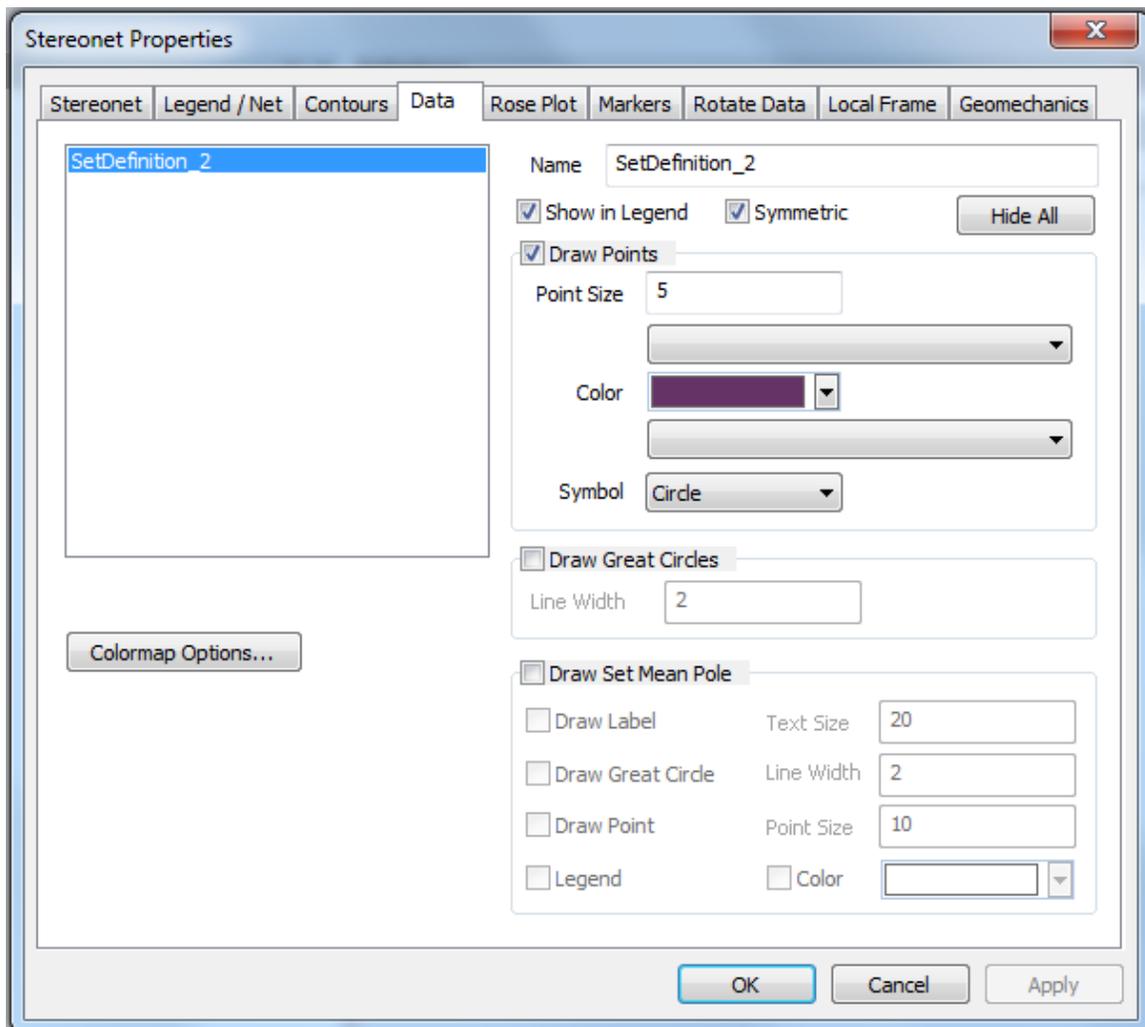
The contours in the 3D image can be enhanced topographically by checking the box **Contour Radius Scale [0-1]** and setting the radius to a value between 0 and 1.



Data Display Options

Under the **Data** tab in the Stereonet Properties window, the user can specify the symbols, colors, and sizes of the data displayed on the stereonet. The point data can be removed to show

only the Contour or Rose stereoplots by unchecking the box next to “Draw Points”. The user can also display the great circles for the points, as well as the mean pole with labels.



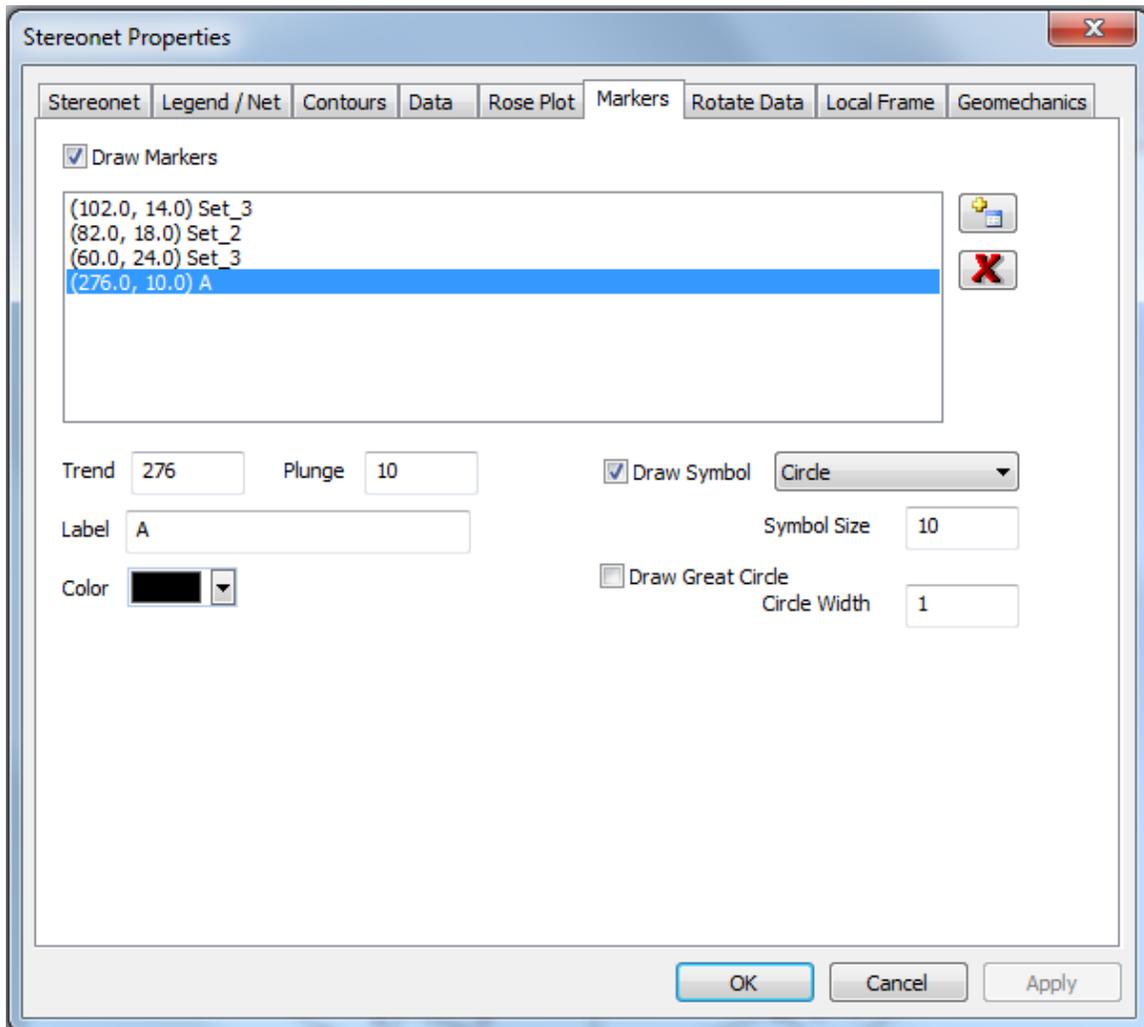
Markers

Markers can be added either by clicking on their location on the stereonet itself, or by entering them in the Properties window.

From the stereonet, right-click on the location and select **Create Marker**. If you want to add or remove several markers, you can also right-click on the stereonet and choose **Select | Marker Toggle** which activates the cursor to create or remove markers when left-clicking. When finished adding or deleting markers, go back to the **Select | Marker Toggle** option and de-select it.

From the Stereonet Properties window, click on the box next to “Draw Markers” under the **Markers** tab and click the “Add Data” button  and then edit the trend, plunge, and label as

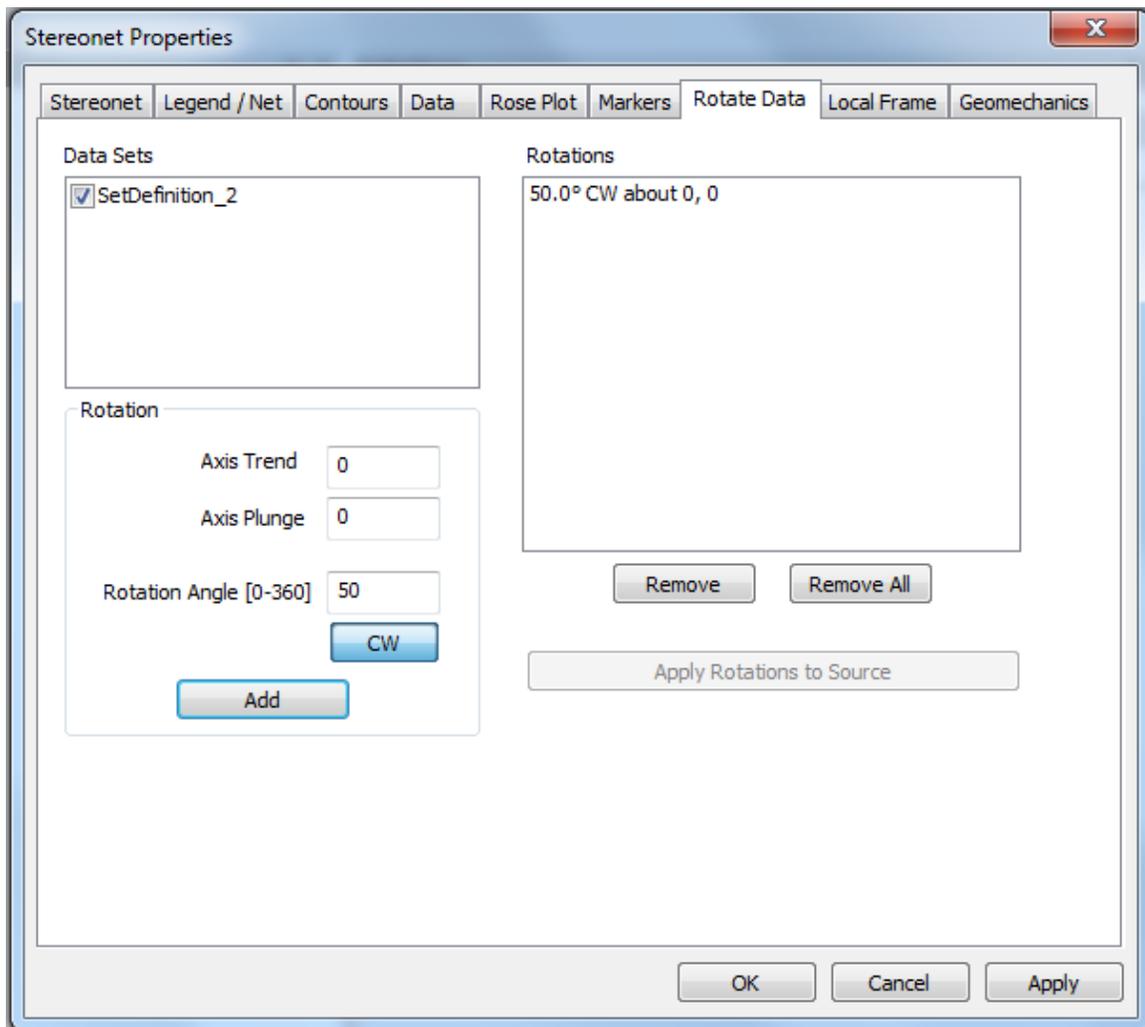
desired. Markers created directly from the stereonet can be edited in the Properties window, Markers tab as well.



Rotate Data

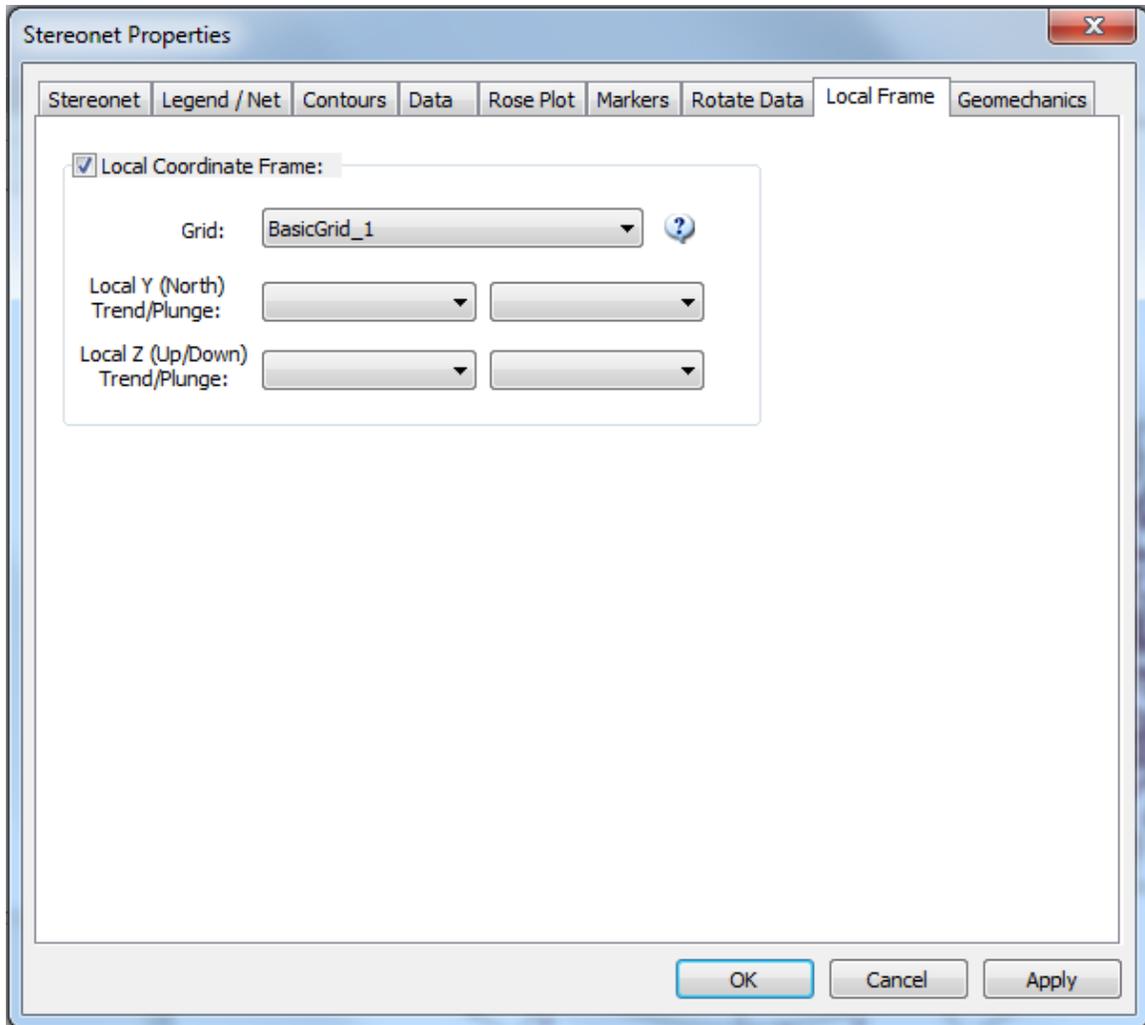
Data can be rotated by a specified axis under the **Rotate Data** tab in the Stereonet Properties window. Under the "Rotation" box, specify the axis trend, plunge, and rotation angle you wish to rotate the data by, and click **Add**. Your rotation will appear in the Rotations area. Click **Apply** to update the stereonet visualization. You can toggle the button "CW"/"CCW" to indicate whether movement should be clockwise or counter-clockwise. Rotations can be removed by selecting the rotation you wish to remove, and clicking the **Remove** button. These changes only affect the visualization shown in the stereonet, not the underlying data. If your data set is a well log or point data, the button "Apply Rotations to Source" is active and, if clicked, will create a copy of the object with the new orientations. This option is not available for other data sets such as fractures. If you want to save the rotated orientations of objects other than well logs or point data, you can make the rotations for the stereonet and then on the stereonet visualization,

right-click and select Analysis->View Data. The data can then be saved to file or copied into a spreadsheet (right-click on data).



Local Coordinate Frame

To make a stereonet based on a local coordinate frame, you must first have a grid in your file that has properties such as the local coordinate frame vectors computed from grid cell orientation (see [Compute Coordinate Frame](#)), local curvature or stress values that you can use as your new reference frame. Once you have made a stereonet of the data you wish to rotate, right-click anywhere in the stereonet window and open the “Properties...” window. Click the box next to “Local Coordinate Frame” under the **Local Frame** tab in the Stereonet Properties window.



First, select the grid that contains the desired properties from the drop-down menu next to “Grid:” which will list all available grids in your file. Then select which grid properties you wish to use as your new coordinate frame from the drop-down menus for trend and plunge (respectively) next to the local axes “Local Y (North)” and/or “Local Z (Up/Down)”. The drop-down menu will list all available grid properties; you may choose the same property more than once. When you have finished, hit **Apply**. This will rotate the stereonet’s frame of reference for each point based on the selected property values for the grid cell containing that point.

As long as the stereonet window is open, the local coordinate frame you have selected will remain; once you close the window, however, the changes will be lost. Note that the coordinate frame will not affect the actual object from which the stereonet was generated. If you want to save the rotated orientations, on the stereonet visualization right-click and select Analysis->View Data. The data can then be saved to file or copied into a spreadsheet (right-click on data).

Geomechanics Stereonet

The Geomechanics tab in the Stereonet Properties window can be used to show how stable orientations are with respect to a constant stress field. Check the **Show Stress** checkbox and choose what to display using the **Contour Value** options (these are documented in the Manual section on [Critical Stress](#)):

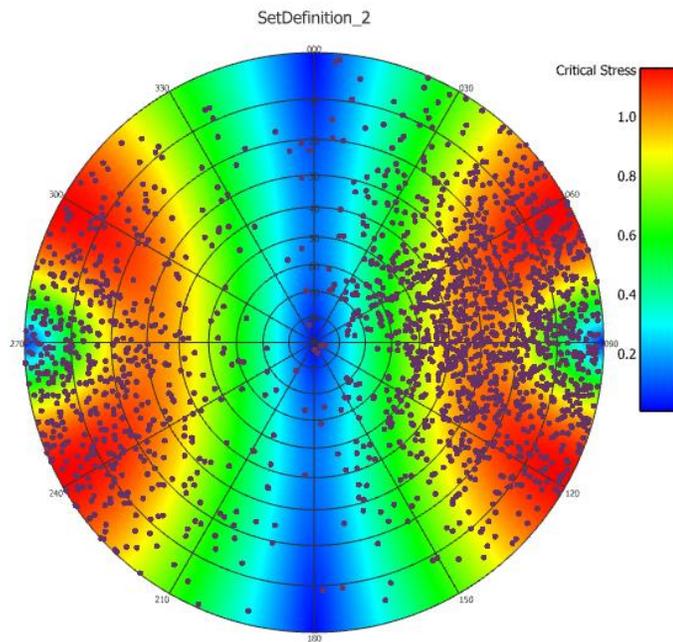
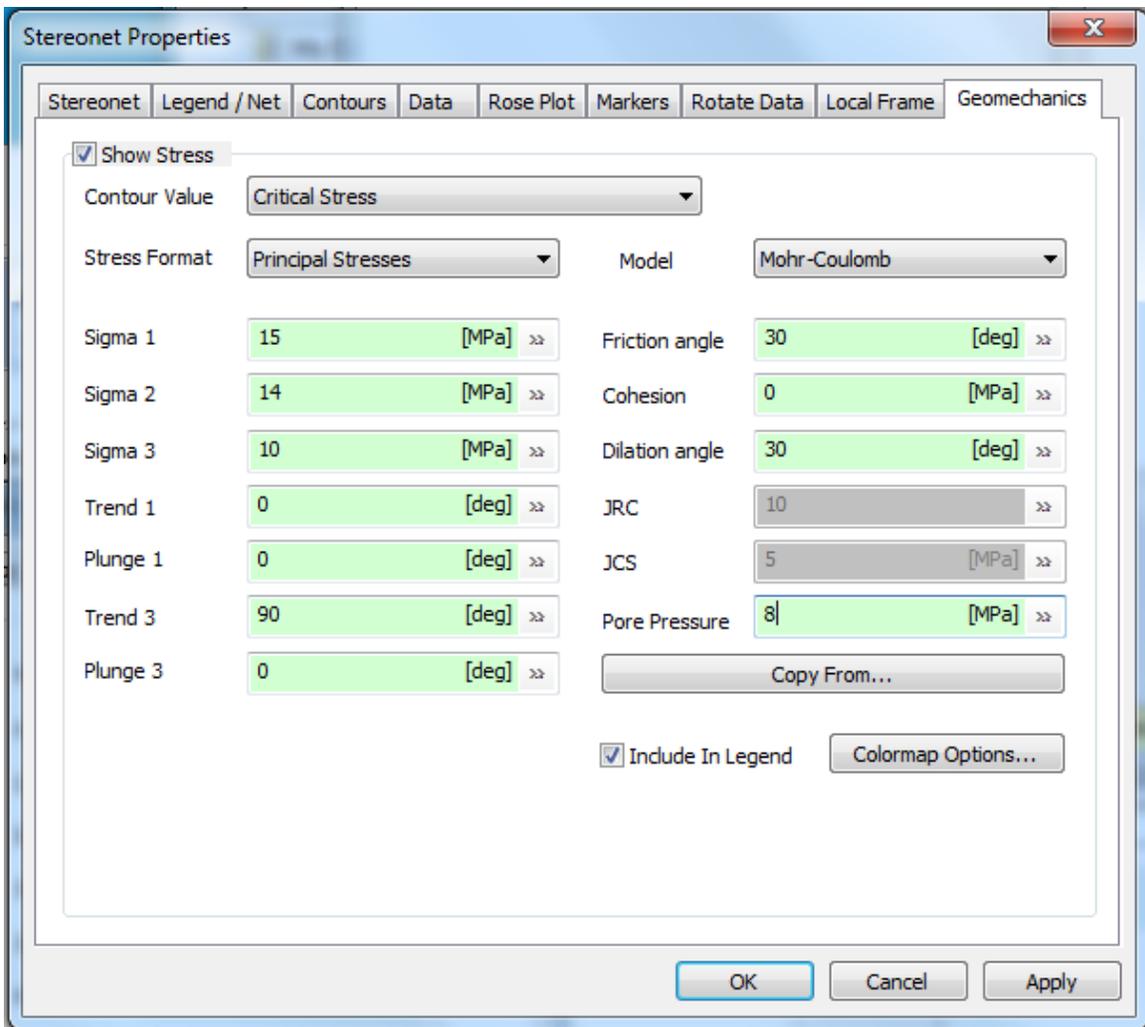
- Critical Stress
- Critical Stress Factor
- Effective Normal Stress
- Reactivation Pressure
- Reactivation Pressure Increment
- Tendency – Slip
- Tendency – Dilation
- Tendency – Leakage

The stress field can either be loaded from an existing critical stress definition using the **Copy From** button or entered manually. If entered manually, select the **Stress Format** (choices are Principal Stresses or Cartesian Stresses) and the failure **Model** (choices are Mohr-Coulomb, Barton-Bandis and High Stress Barton Bandis) and enter the appropriate parameter values.

Include in Legend: Check if you want the stress contour option included in the stereonet legend.

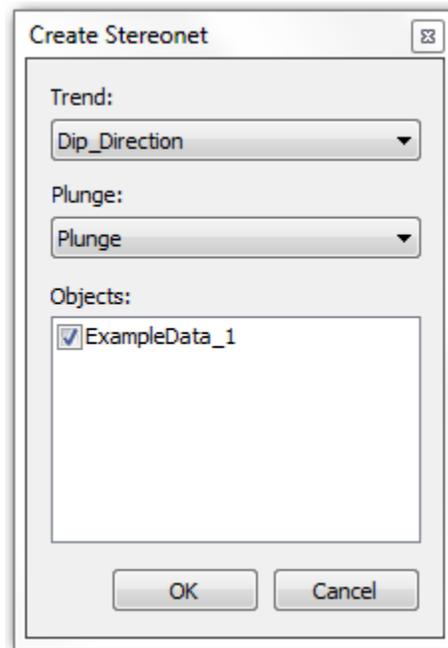
Note: the **Show Legend** option must be checked on the **Legend / Net** tab for this to be visible.

ColorMap Options: Provides options controlling the colors used in the contouring as well as the appearance and position of the contour legend.



Property Stereonet

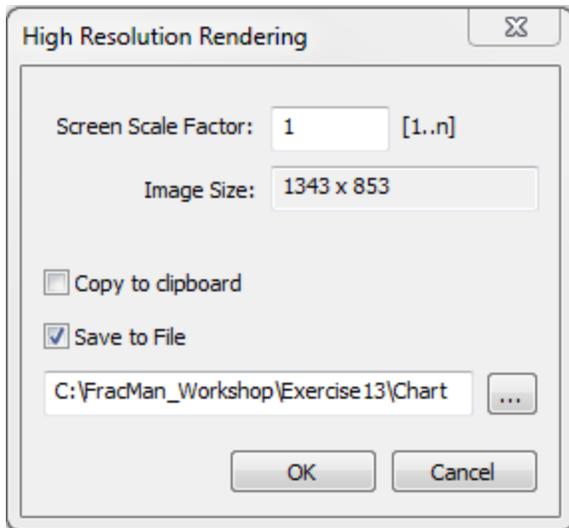
FracMan gives you the option of making a stereonet based not on the traditional trend and plunge, but on any two numerical properties your data set possesses. In order to make such a graph, select the object in the Objects window and then choose **Static Analysis > Property Stereonet** from the menu bar. This will open a dialog box asking you to specify which properties to use.



Exporting Your Chart

There are a number of options available to export your chart. No matter what type of chart you select, you can copy the open graph to the clipboard by selecting the “Copy to Clipboard”

toolbar button  so that you can paste it into another application as an image. Alternatively, you may choose save the graph as its own file by selecting **View > High Resolution Screen Capture** in the file toolbar. This function allows you to copy the image to the clipboard, or save the image in one of five common image formats (BMP, JPEG, GIF, TIFF, and PNG).



Data Analysis

Data analysis is the “how” of the fracture modeling process. Data analysis includes tools for analyzing your data and building and optimizing your DFN. There are seven types of data analysis:

[Flow Capacity](#)

[Spatial](#)

[Orientation](#)

[Intensity](#)

[Size](#)

[Spatial Patterns](#)

[Well Log Report](#)

[Object Statistics](#)

Set Identification

FracMan uses the Set Identification to identify fracture sets in fracture data. Set Identification defines fracture sets from field data using an adaptive probabilistic pattern recognition algorithm. It calculates the distribution of orientations for the fractures assigned to each set, and then reassigns fractures to sets according to probabilistic weights proportional to their similarity to other fractures in the set. The orientations of the sets are then recalculated and the process is repeated until the set assignment is optimized.

The Theory Behind Set Identification

Set Identification defines fracture sets from field data using an adaptive, probabilistic pattern recognition algorithm. It allows the user to select the fracture properties of concern and define their relative significance for the analysis. Unlike conventional approaches to fracture set definition, which define sets by contouring orientations on stereoplots, Set Identification is as powerful for overlapping fracture sets as for clearly defined sets. The premise of the Set Identification approach for definition of fracture sets is that the sets should be groups of

fractures with similar properties. Orientation does not need to be the only property defining sets; size, termination, fillings and other properties can identify sets as well.

The goal of the Set Identification approach is to assign groups of fractures which are similar geologically or hydrologically to sets which are statistically homogeneous. This is achieved by deriving the statistical properties of each set from the statistical properties of the fractures assigned to the set and then removing and reassigning fractures which have a low probability of being part of that set. The algorithm begins by requiring the user to select and assign weighting factors W_j to each of the n_j fracture characteristics of concern and to provide initial guesses for the properties of the each set k . In addition, the user must specify the convergence method, either hard sector, soft sector, sector preconditioning or none (see below). For each fracture i , it then calculates a probability that the fracture belongs to that set. During the first iteration, this probability is calculated using the value of the probability density function for that property and set, taken at the value of that fracture.

$$P[i \in k] = C p_{w,k} \quad (3-1)$$

$$p_{w,k} = \sum_{j=1}^{n_j} W_j f_j(k_i) P_{si} + f_{si} \quad (3-2)$$

$$C = 1 / \sum_{k=1}^{n_k} p_w \quad (3-3)$$

where $P[i \in k]$ is the probability that fracture i is an element of set k , $p_{w,k}$ is the weighted probability density function of set k for property j , $f_k(k_i)$ is the probability density function of set k for property j evaluated for fracture i , W_j is the user-assigned weighting for property i , P_{si} and f_{si} are sector factors, (see below) and C is a proportionality constant defined to ensure that the sum of probabilities for all sets equals 1. Set Identification does not require complete information on every fracture property j for all fractures i . When a piece of information about fracture i is missing, $f_k(k_i)$ is set at 0 for that fracture over all sets k , such that only those properties for which values are specified are used in the assigning that fracture to a set.

The probability density functions $f_k(k_i)$ for each property of each set are provided by the user for the first iteration. For subsequent iterations, they are calculated using the distributional forms specified by the user, with moments calculated using the fractures assigned to sets during the previous iteration. The form of the probability density function $f_k(k_i)$ is also determined by the user and may be Fisher, Bivariate Normal, Bivariate Bingham, and Elliptical Fisher (for orientation), normal, lognormal, exponential or a histogram and is dependent on the property. See the sections [Orientation](#) and [Probability Distributions for Scalar Values](#) for the probability density function equations.

Once the probability $P[i \in k]$ has been calculated for each fracture and set, the fractures are reassigned to sets by Monte Carlo simulation, such that each fracture is assigned to set k with probability $P[i \in k]$.

When all fractures have been reassigned to sets, the set distribution parameters are recalculated for each property j using the fractures assigned to that set. For orientation statistics, the mean pole vector or mean dip vector and the Fisher concentration parameter k (Fisher et al., 1987) are calculated. For continuous distributions such as lognormal and normal, the mean and standard deviation are recalculated, and for histogram distributions the percentages for each class are calculated. For example, for rock type histogram distributions, the property statistics for rock-type for set k are set to the percentage of each rock type among the fractures currently assigned to set k .

When the distributions of fracture properties for each set have been recalculated, it repeats the process, reassigning sets to fractures. In order to speed convergence, a relaxation factor γ is used in calculating the probability $P[i \in k]$. This relaxation factor controls the rate of change in $P[i \in k]$ using a weighted average of probability density functions for the fractures' current and previous set assignments:

$$P[i \in k] = Cp_{w,k} \quad (3-4)$$

$$p_{w,k} = \sum_{j=1}^{n_j} (1 - \gamma_j) W_j f_j(k_i) + \sum_{j=1}^{n_j} \lambda_j W_j f_j^{(old)}(k_i) \quad (3-5)$$

$$C = \frac{1}{\sum_{k=1}^{n_k} p_w} \quad (3-6)$$

where $f_j^{(old)}$ is the probability density function from the previous iteration. For each property j , a relaxation factor of 0 calculates the probability $P[i \in k]$ based solely upon the current assignment of fractures to sets, and a relaxation factor of 1 calculates the probability $P[i \in k]$ without any change as fractures are reassigned and as set-properties change. A value of approximately 0.5 is appropriate for most properties.

Set Identification also uses damping factors in the calculation of updated mean directions and concentration parameters for each set using the equation

$$\kappa^{(updated)} = (1 - d_{\kappa}) \kappa^{(old)} + d_{\kappa} \kappa^{(new)} \quad (3-7)$$

$$v^{(updated)}(\bar{\varphi}, \bar{\theta}) = \frac{(1 - d_u) v^{(old)}(\bar{\varphi}, \bar{\theta}) + d_u v^{(new)}(\bar{\varphi}, \bar{\theta})}{|(1 - d_u) v^{(old)}(\bar{\varphi}, \bar{\theta}) + d_u v^{(new)}(\bar{\varphi}, \bar{\theta})|} \quad (3-8)$$

where:

d_{κ} T damping parameter for Fisher concentration

d_u The damping parameter for mean orientation

$\kappa^{(old)}$ The Fisher concentration parameter from the previous iteration

$\kappa^{(new)}$ The Fisher concentration parameter based on the fractures currently assigned set

$\kappa^{(updated)}$ The Fisher concentration parameter to be used by the set in the next iteration.

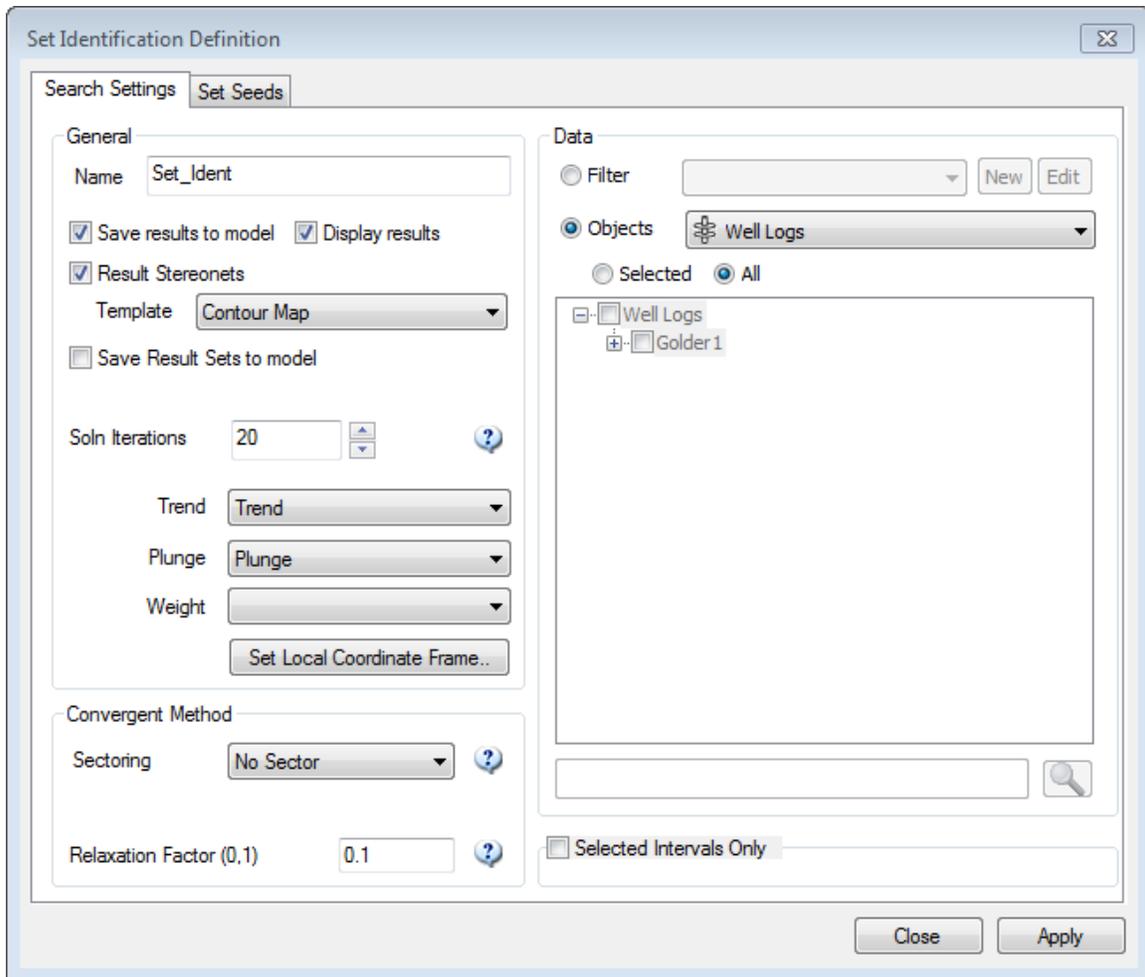
v The unit vector for mean orientation (Φ, θ) .

This process is repeated for a number of iterations specified by the user. When the analysis is complete, it calculates the Kolmogorov-Smirnov goodness-of-fit statistics for each of the properties and distributions used in the analysis, based upon the fractures assigned to the sets at the end of the analysis. This process of regrouping the fractures is repeated until the distribution parameters of the sets stabilize.

Using Set Identification

You can run Set Identification by right-clicking anywhere in a stereonet window and selecting **Analysis > Define Set Identification**, or by clicking the **Data Analysis** tab on the menu bar, then **Orientation > Set Identification**. This will bring up a dialog box setting the definition. Before opening up this dialog, you may want to select starting set positions by right-clicking at the approximate location of the set pole on the stereonet and selecting "Create Marker". These markers will be used to populate the starting seeds for the analysis (they can also be manually entered in the **Set Seeds** tab of the **Set Identification Definition** dialog itself).

Search Settings



General Section

- **Name:** You may specify a name for the definition in the box next to “Name” at the top of the “General” section.
- **Save Results to Model:** Save analysis in the project.
- **Display Results:** Show analysis results after completion.
- **Result Stereonets:** If checked, will show resulting sets plotted on stereonet. Choose a template (optional) from the drop-down list to indicate the desired formatting if different from the default scatter plot.
- **Save Result Sets to Model:** If checked, the resulting sets identified will be saved as objects in the project.
- **Soln Iterations:** Specify how many run-throughs the analysis undergoes before presenting your results. Sets that are already spatially well-defined on a stereonet may only need one iteration while intermingled sets are best analyzed using 20 to 200 iterations.
- **Trend/Plunge/Weight:** Select the appropriate fracture properties to use for trend, plunge and weight (optional) from the respective drop-down menus. Be sure to select

the Terzaghi property for the Weight if you have previously applied a Terzaghi or Mauldon correction to the fractures. This is important for fracture sets imported from well logs. For instance, a vertical well is far more likely to intersect a horizontal fracture than a vertical fracture; thus, a higher frequency of sub-horizontal than sub-vertical fractures in a vertical well log does not necessarily mean that the natural fracture regime has more sub-horizontal fractures than sub-vertical (see section on [Terzaghi/Mauldon Correction](#) for details). Other fracture properties can be used as weighting factors as well. If multiple properties need to be combined for the weighting function, create a new fracture property before defining the analysis.

- **Set Local Coordinate Frame:** If you rotated your data to a local coordinate frame prior to starting, you can return it to its original orientation now by clicking “Set Local Coordinate Frame”.

Convergent Method Section

The “Convergent Method” section allows you to enter two parameters, one for Sectoring: either No Sector, Preconditioning Sector (Pre-Sector), Soft Sector, or Hard Sector, and one for the Relaxation Factor: a floating number between 0 and 1. Both are used to control the convergent speed in set assignment.

The sectoring parameter controls the stability of set assignment to fracture. Sectors are defined by the closest mean pole (or dip vector) as defined by the fracture set mean pole. For the No Sector option, the assignment is a probabilistic random process. The set number with a higher probability value gets a higher chance to be assigned to the fracture. For the Hard Sector option, the assignment is not a random process and the fracture is always assigned to the set with the highest probability score. Refer to the section [The Theory Behind Set Identification](#) for details and equations. For Pre- and Soft Sectors, one “Adds-in” probability value, f_{si} , mentioned in Equation (3-2), is added to the most favorite set before following a probabilistic random process to assign set. The sectoring parameters are used in Equation (3-2) for the terms P_{si} and f_{si} .

Convergence Factors and Sectoring in Set Identification

	Fracture i in same sector as set k mean pole (or dip)	Fracture i in different sector from set k mean pole (or dip)
Hard Sector	$P_{si} = 1$ $f_{si} = 0$	$P_{si} = 0$ $f_{si} = 0$
Soft Sector	$P_{si} = 1$ $f_{si} = 0.25$	$P_{si} = 1$ $f_{si} = 0$
No Sector	$P_{si} = 1$ $f_{si} = 0$	$P_{si} = 1$ $f_{si} = 0$
Pre- Sector	Same as Hard Sector first iteration Same as Soft Sector afterwards	

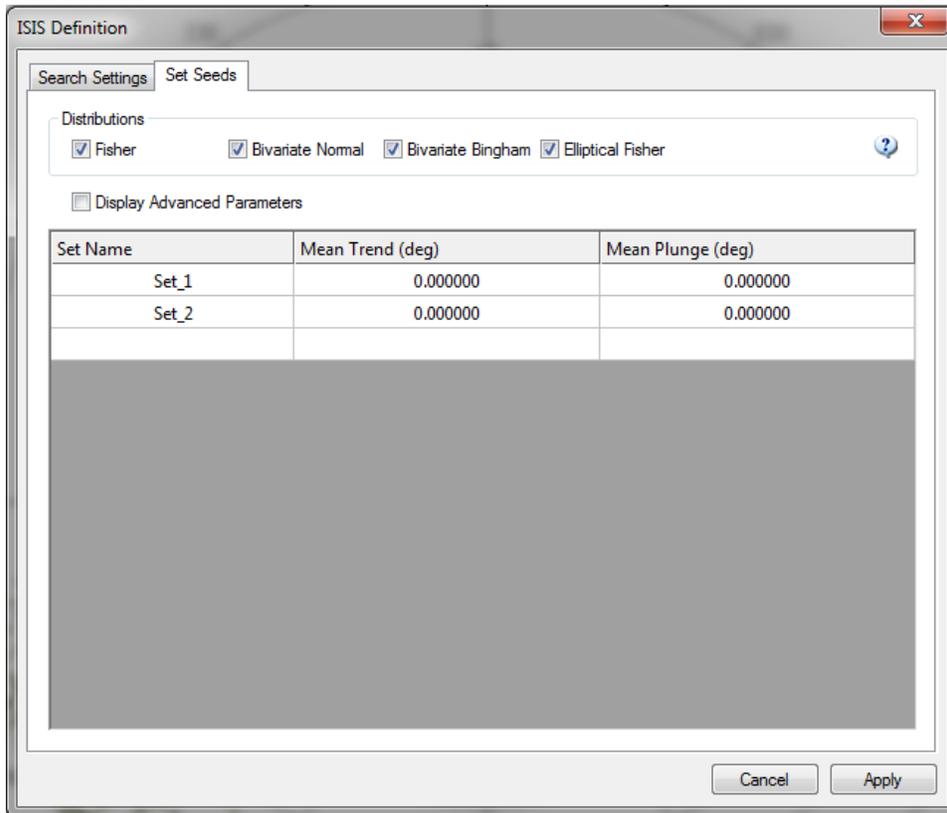
The relaxation factor (f) is used in Equations (3-7) and (3-8) for the calculation of updated parameters. It controls the rate of change in parameter update from the previous to the current iteration. A single input value of f is applied to both mean orientation, d_{κ} , used in (3-7) and the scalar parameter, d_u , used in (3-8). A value of approximately 0.5 is appropriate for most properties.

Data Section

The “Data” section on the right half of the box is where you specify exactly what data you wish to analyze. By default, this will contain the data in the stereonet from which you initiated Set Identification, but if you began the analysis from the file menu or if you wish to change the selected set, you may enter it here. Select the object from the drop-down menu next to “Objects” from which you would like to select data. Click the “Selected” radio button to choose a specific dataset, or click the “All” radio button to analyze all the data under that object. If you checked the “Selected” radio button, check the fracture sets and logs on which you wish to use. If you want to analyze only fractures below a certain depth, for instance, or based on some other property, you may either select an existing filter from the dropdown menu beneath “Filter:” or define a new one by clicking the “Create New” button. You may also edit an existing filter here.

If you are operating on a well log whose well has perforation or intensity intervals defined, you may check the “Selected Intervals Only” box and select one or more P_{10} or perf intervals to confine the analysis to.

Set Seeds



- **Distributions:** Check which probability distributions you want to explore. See the sections [Orientation](#) and [Probability Distributions for Scalar Values](#) for the probability density function equations.
- **Set Seeds:** In order to run Set Identification, you must define a seed pole for each set you wish to isolate. You may do this under the **Set Seeds** tab in the Set Identification Definition dialog box by manually entering the poles, or by setting a marker on the stereonet itself at the general location of each pole (right-click at approximate location of pole and select “Create Marker”). The trend and plunge of each selected point will automatically be logged under the Set Seeds tab. You may also modify seeds under this tab after they have been entered. Check the **Display Advanced Parameters** checkbox if you want to set other initial probability distribution parameters.

When you have finished defining your analysis, click **Apply, Run**, and then **Close** the Definition dialog box.

Once Set Identification is complete, a new stereonet for each checked Distribution choice (from the Set Seeds tab) and a table of data will appear (if you had checked the boxes for Display Results and Result Stereonets). The table contains set definitions fitting the orientation distributions against which the data is tested (Fisher, Bivariate Normal, Bivariate Bingham, and Elliptical Fisher) and statistical measures of how well the data fit them. Each stereonet

represents a guess of what the sets might be, based on the given distribution. Each set, as projected by the analysis, is shown in a different color. By comparison, you can figure out which of the distributions best fits your data, or if you should run more tests using different weighting functions to determine if something besides orientation is differentiating your sets.

Flow Capacity (OxFILET) Analysis

The OxFILET ("Osnes Extraction from Fixed-Interval-Length Effective Transmissivities") module is used to determine the transmissivity distribution and frequency of conductive fractures from packer test data by using an approach adapted from Osnes et al. (1988).

To run OxFilet, select **Data Analysis > Intensity > OxFilet Flow Capacity Analysis** from the file menu.

Select OxFilet Input Data

You may specify a name for the definition in the box next to "Name" and a description in the box next to "Description". The only input required by OxFilet is a Well Interval Set containing Transmissivity / Flow Capacity data ($k \cdot h$). Select one or more well interval sets and then select the property containing the Transmissivity / Flow Capacity data.

The screenshot shows the "OxFilet Flow Capacity Definition" dialog box. It has three tabs: "Select Oxfilet Input Data" (selected), "OxFilet Parameters", and "Auto Search Options".

- Name:** A text box containing "OxFilet Definition".
- Description:** An empty text box.
- Select the well interval sets containing the Flow Capacity / Transmissivity data:**
 - Filter:** A dropdown menu.
 - Objects:** A dropdown menu showing "Wells / Intervals".
 - Selected / All:** Radio buttons, with "All" selected.
 - Tree View:** A tree structure showing "Wells" expanded to show "Golder 1" and "Golder 2".
- Flow Capacity Property:** A dropdown menu.

At the bottom are "Close" and "Apply" buttons.

OxFilet Parameters

OxFilet Flow Capacity Definition

Select Oxfilet Input Data | OxFilet Parameters | Auto Search Options

Analysis Parameters

Minimum Flow Capacity: 1e-010 [m²/s] >> ?

Network Depth: 1 >>

Flowing Fracture Intensity (frac/m): 2.5 >>

Flow Capacity Distribution

Transmissivity: LogNormal distribution

Cancel Apply

Minimum Flow Capacity

Specify the minimum value of Transmissivity / Flow Capacity which can be measured by the packer test equipment used to produce the values in the data being analyzed. Any interval measurements less than this value, either in the input data, or in the simulation results, will be considered as "No Flow" intervals, with transmissivity set to this value.

Network Depth

Choose the interpretation to be used to relate the cross-fracture transmissivity value to the transmissivity seen in the packer test. If Network Depth is 1 (Single Fracture), the mean, standard deviation, and distribution of fracture transmissivity specified above are used directly to calculate packer interval transmissivity according to equation (3-14). If Network Depth is greater than 1 (Fracture Network), the values of "cross-fracture" transmissivity are converted to

"fracture network" transmissivities according the equation (3-17) before being used to calculate packer interval transmissivities.

Flowing Fracture Intensity

Specify the assumed mean number of conductive fractures per meter. The mean value of the Poisson distribution equals the packer interval length multiplied by the number of fractures per unit length; simulated packer intervals will have a varying numbers of fractures, with this value as the mean.

Flow Capacity Distribution

Select the distribution and parameters for Transmissivity (or Permeability / Aperture) simulation.

Auto Search Options

The screenshot shows the 'OxFilet Flow Capacity Definition' dialog box with the 'Auto Search Options' tab selected. The dialog has three tabs: 'Select Oxfilet Input Data', 'OxFilet Parameters', and 'Auto Search Options'. The 'Auto Search Options' tab contains the following settings:

- Choose Search Type:** Radio buttons for 'No Auto Search' (unselected) and 'Simulated Annealing Search' (selected). A help icon (?) is visible to the right.
- Search Parameters:**
 - Radio buttons for 'Based on K-S statistics' (selected) and 'Based on Chi-Squared statistics' (unselected).
 - # of Iterations: 1000
 - Quenching factor [0.5,1]: 0.9
 - Parameter Temperature: 0.5
 - Mean Permability (Min, Max): 80000 (Min), 120000 (Max)
 - StDev Permability (Min, Max): 80 (Min), 120 (Max)
 - # Fractures/Meter (Min, Max): 2 (Min), 3 (Max)
- Results:**
 - Display Results
 - Save Results Object
 - Export Results (with an empty text field and a browse button '...')

Buttons for 'Cancel' and 'Apply' are located at the bottom right of the dialog.

Search Type

If more than one iteration is desired, Select “Simulated Annealing Search” to provide the parameters used in the specified optimized search for distribution of fracture radius providing the best fit between simulated and measured trace length distributions.

Search Criterion

Select the radio button next to either Kolmogorov-Smirnov (K-S) or Chi-Squared (χ^2) statistics tests to compare simulated and measured packer interval transmissivity distributions. The K-S test is based upon the single worst match in the distribution, and therefore tends to produce matches which balance differences between different parts of the cumulative density functions (CDFs). The χ^2 test is based upon the weighted average difference between the probability density functions (PDFs), and therefore trades off worse fits in some parts of the distribution for better fits in other parts.

Iterations

Enter the number of iterations to be performed.

Quenching Factor

A higher number makes the quench speed slower, the search range reduction slower. Recommended value is 0.9.

Parameter Temperature

The parameter temperature is a parameter used to control the convergence of the simulated annealing search for the best fit. The value of parameter temperature scale should be equal to approximately 10 times the change in K-S or χ^2 statistic (see equation (3-13)).

Mean Permeability (Minimum, Maximum)

Enter the range of values of mean "cross-fracture" transmissivity to be considered in searching for the best-fit distribution of "cross-fracture" transmissivity.

Standard Deviation Permeability (Minimum, Maximum)

Enter the range of values of standard deviation for "cross-fracture" transmissivity to be considered in searching for the best-fit distribution of "cross-fracture" transmissivity.

Fractures/Meter (Intensity) (Minimum, Maximum)

Enter the range of values of the mean number of conductive fractures per unit length of packer interval to be considered in searching for the best-fit distribution of conductive fracture frequency.

When you have finished defining your analysis, click **Apply**, **Generate**, then **Close** the OxFilet Flow Capacity Definition dialog box.

OxFilet Theory

This method assumes that the net transmissivity of a test zone is equal to the sum of the transmissivities of the conductive fractures that intersect that test zone (Figure 3-8):

$$T_1 = \sum_{j=1}^{n_i} T_{ij} \quad (3-14)$$

where T_{ij} is the apparent transmissivity of the i th packer interval, n_i is the number of conductive fractures in the i th interval, and T_{ij} is the transmissivity of the j th conductive fracture within the i th interval. Within any given interval, the number of conductive fractures, n_i , is assumed to be a random number defined by a Poisson distribution (Benjamin and Cornell, 1970):

$$f_n(n) = \frac{\bar{n}^n e^{-\bar{n}}}{n!} \quad (3-15)$$

where \bar{n} is the Poisson process rate, which is equal to the expected value of n . The conductive fracture frequency is given by $f_c = n/L_i$, where L_i = the length of the test zone.

The distribution of fracture transmissivities is assumed to be independent within each packer interval, with a given distributional form. The distribution of T_i is the sum of a random number of random events, and is therefore a compound Poisson process (Feller, 1971). In this approach, the mean number of fractures in a given interval is defined by the Poisson distribution rate parameter, n , and the distribution of fracture transmissivities T_{ij} is described by a lognormal distribution with a mean and standard deviation, $m_{\log T}$ and $s_{\log T}$.

For any given set of parameters describing the distribution of fracture transmissivity $f(T_{ij})$ and conductive fracture frequency f_c , the distribution of packer interval transmissivities $f(T_i)$ are found by Monte Carlo simulation, with the best fit value found by a simulated annealing search routine. Simulated intervals that contain no conductive fractures, or that have values of T_i less than $T_{threshold}$, the lowest threshold transmissivity that could be reliably measured in the field, are assigned a transmissivity equal to $T_{threshold}$.

The intensity and transmissivity distributions for the conductive fractures are then estimated by finding the best match between the observed distribution of packer interval transmissivities $f(T_i)$ and the distribution of test zone transmissivities found by simulation for given fracture frequency and single-fracture transmissivity distributions. This match is found both visually and by comparison of Kolmogorov-Smirnov (K-S) or Chi-Squared statistics (χ^2).

The values of n , $\mu_{\log T}$ and $\sigma_{\log T}$ that provided the best K-S or χ^2 minimize D are taken to be the best estimates of those parameters.

MAFIC flow and transport simulations require that the fracture transmissivity used be the effective transmissivity through a fracture between the fractures intersecting the fracture ("cross-fracture transmissivity", T_{fi}). OxFilet provides three alternative interpretations for the

relationship between T_{fi} and T_{ij} , the transmissivity seen in the fixed interval packer test (Figure 3-9)

The packer test influences one fracture at a time, such that the transmissivity seen for each fracture T_{ij} is equal to the cross fracture transmissivity, T_{fi} .

The packer test is strongly influenced by the local fracture aperture near the borehole. In this case, the transmissivity seen by the packer test is a small-scale ("at-borehole") transmissivity, and the cross-fracture transmissivity T_{ij} must be found using a correlation of the form:

$$T_{ij} = B_f T_{fi} \quad (3-16)$$

In OxFilet, the proportionality constant B_f is described as a normally distributed random variable with mean and standard deviation provided by the user. An example of the relationship between T_{fi} and T_{ij} from numerical simulation is shown in Figure 3-10 (Kenrick et al, 1991). The packer test influences a network with a number of interconnected fractures. In this case, the transmissivity seen by the packer test, T_{ij} , is a network transmissivity related to the cross-fracture transmissivity of a number of fractures. Assuming series flow through the n fractures influenced by the packer test, the relationship between T_{ij} and T_{fi} could be given approximately by:

$$T_{ij} = \frac{\bar{m}}{\sum_{i=1}^m \frac{1}{T_{fi}}} \quad (3-17)$$

For this option, an additional parameter \bar{m} , the mean number of fractures per network seen by packer tests must be specified. The distribution of \bar{m} may be constant or Poisson.

OxFilet Output

Skewness

The arithmetic skewness (third moment) of the packer interval transmissivity distribution. Skewness of zero indicates a symmetrical distribution. Increasingly negative or positive kurtosis values indicate distributions with peaks further to the left or right of the mean, respectively.

Kurtosis

The arithmetic kurtosis (fourth moment) of the packer interval transmissivity distribution. Kurtosis is always positive. Increasing values of kurtosis indicate flatter distributions for the same standard deviation.

(Smirnov, % Signif)

The Kolmogorov-Smirnov (K-S) statistic obtained by comparison of simulated and measured packer interval transmissivity distributions, and the corresponding percent significance. The K-S statistic is the largest absolute difference in CDF values between simulated and measured CDFs, with larger values indicating worse fits. The percent significance is the probability that the observed mean and standard deviation could be seen, given that the actual mean and standard deviation were the values used to produce the simulated packer interval transmissivity distribution.

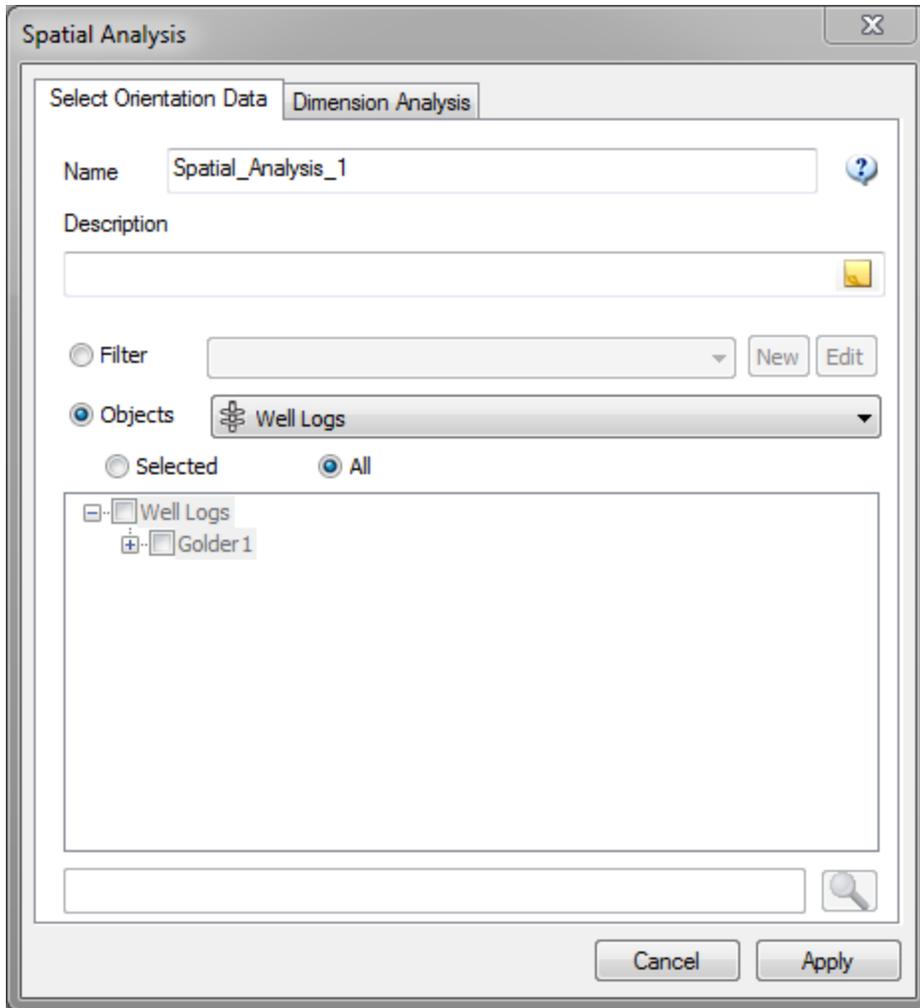
(Chi-Sqr, % Signif)

The Chi-Squared (χ^2) statistic obtained by comparison of simulated and measured packer interval transmissivity distribution, and the corresponding percent significance. The χ^2 statistic is the weighted average difference between counts of observations in histogram probability density functions (PDFs) for simulated and measured PDFs, with larger values indicating worse fits. The percent significance is the probability that the observed mean and standard deviation could be seen, given that the actual mean and standard deviation were the values used to produce the simulated packer interval transmissivity distribution.

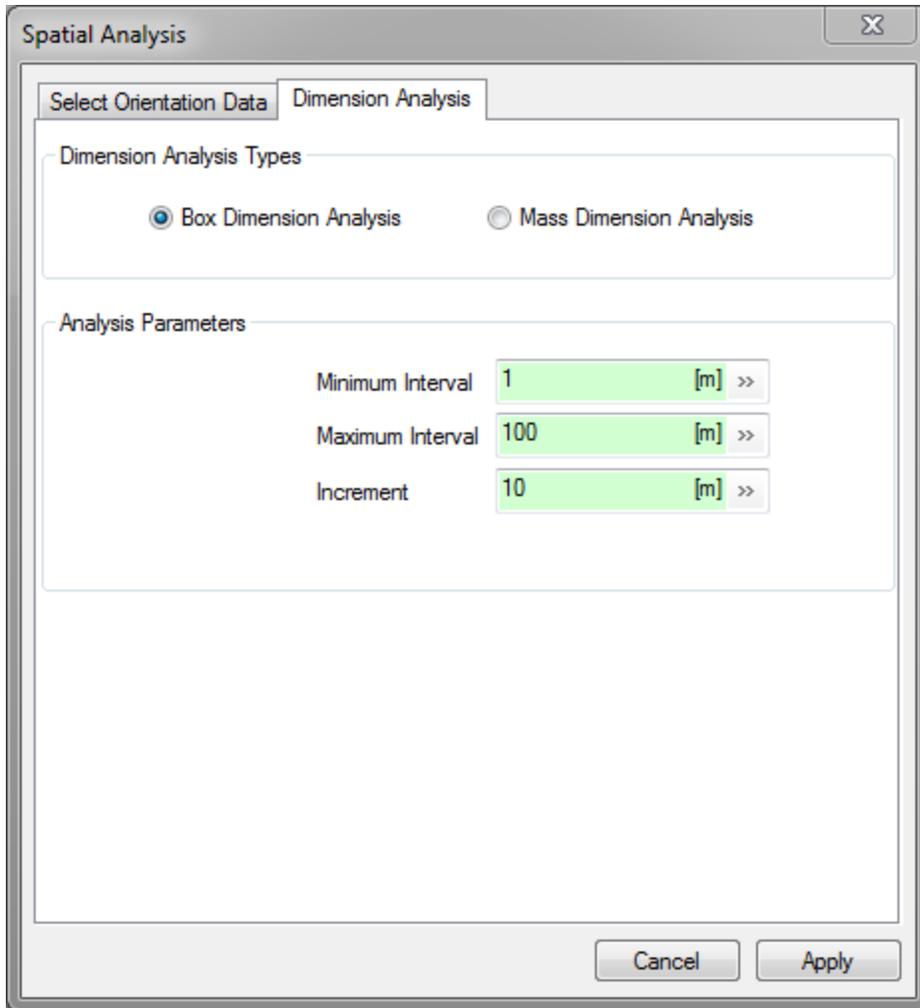
Spatial Analysis

The spatial analysis module allows you evaluate whether either of two common fractal modules – the box dimension and the mass dimension – describes the fracture intensity in a well. Both methods subdivide the well into segments of equal lengths. The box dimension analysis then determines the number of intervals that have fractures in them. The mass dimension analysis determines how many fractures are in a segment of a particular length.

To create a spatial analysis, select **Data Analysis > Spatial > Set Spatial Parameters** from the file menu. A new window will appear that lets you select the fracture data to be included in the analysis.

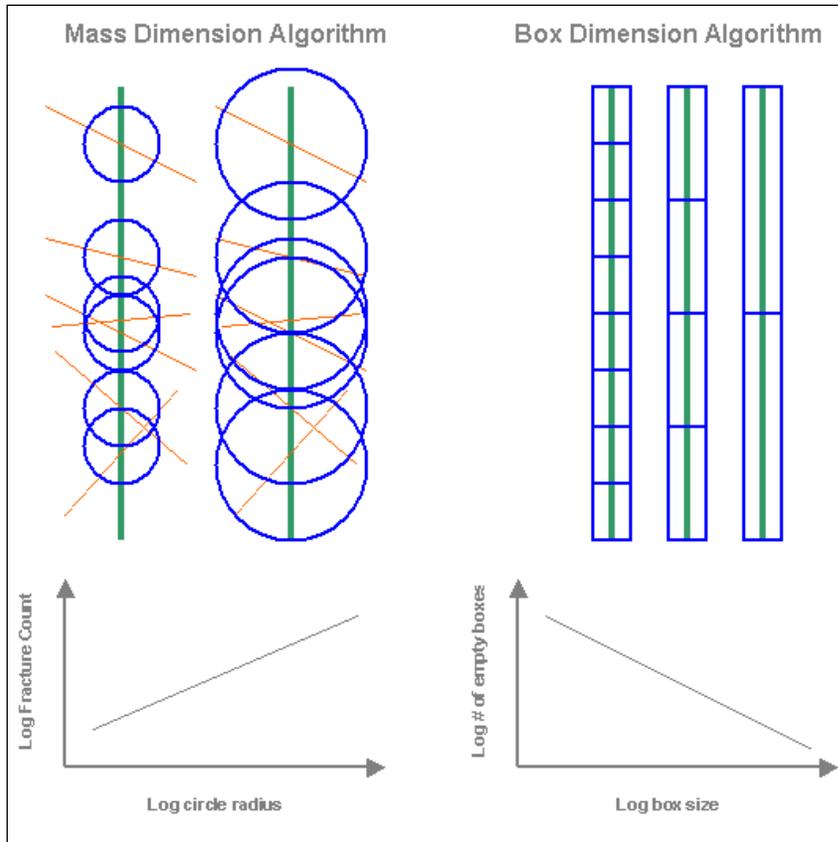


The analysis definition will be saved as the name specified in the “Name” box, and you can provide a description of the analysis in the box under “Description”. Select one or more fracture logs to be considered in the analysis. Click on the **Dimension Analysis** tab to continue.



There are two available analysis types: **Box Dimension Analysis** and **Mass Dimension Analysis**. These are explained in greater detail in the following section. Select the type you wish to conduct, and then specify the parameters for the analysis. The *Minimum* and *Maximum Intervals* are the smallest and largest scales within which the fracture intensities are measured, respectively. The *Increment* is the number of scaling steps between the *Maximum* and *Minimum Interval*. When you have completed your definition, click **Apply**. To run the analysis, click **Compute** in the Spatial Analysis dialog box, or you may select **Data Analysis > Spatial > Run Spatial Analysis**. When the analysis has finished, two windows will appear showing the analysis results in textual and graphical format. As with all such windows, you may save or copy the results by right-clicking on them and selecting **Save to File** or **Copy**.

The Theory Behind Spatial Analysis. The spatial analysis determines the fractal dimension for the spacing of fractures in a well. Two different fractal dimensions can be computed by FracMan: Mass dimension and Box dimension. The following graphic illustrates the two different methods:



The mass dimension algorithm places a circle at every fracture location and counts the number of fractures within the circle. This exercise is repeated for different size circles. If the average number of fractures in a circle is plotted against the circle radius on a log-log scale it can be approximated by a straight line that can be expressed using the following equation:

$$N(L) = \rho L^{D_M}$$

where

- $N(L)$ = average number of fractures,
- L = interval length
- D_M = mass dimension
- ρ = constant.

The box dimension is calculated by placing a series of same-sized boxes over the well. Next the number of boxes that contain at least one fracture is counted. If this number is plotted against the box size on a log-log scale it can be approximated by a straight line using the following equation.

$$N(L) = L^{-D_B}$$

where

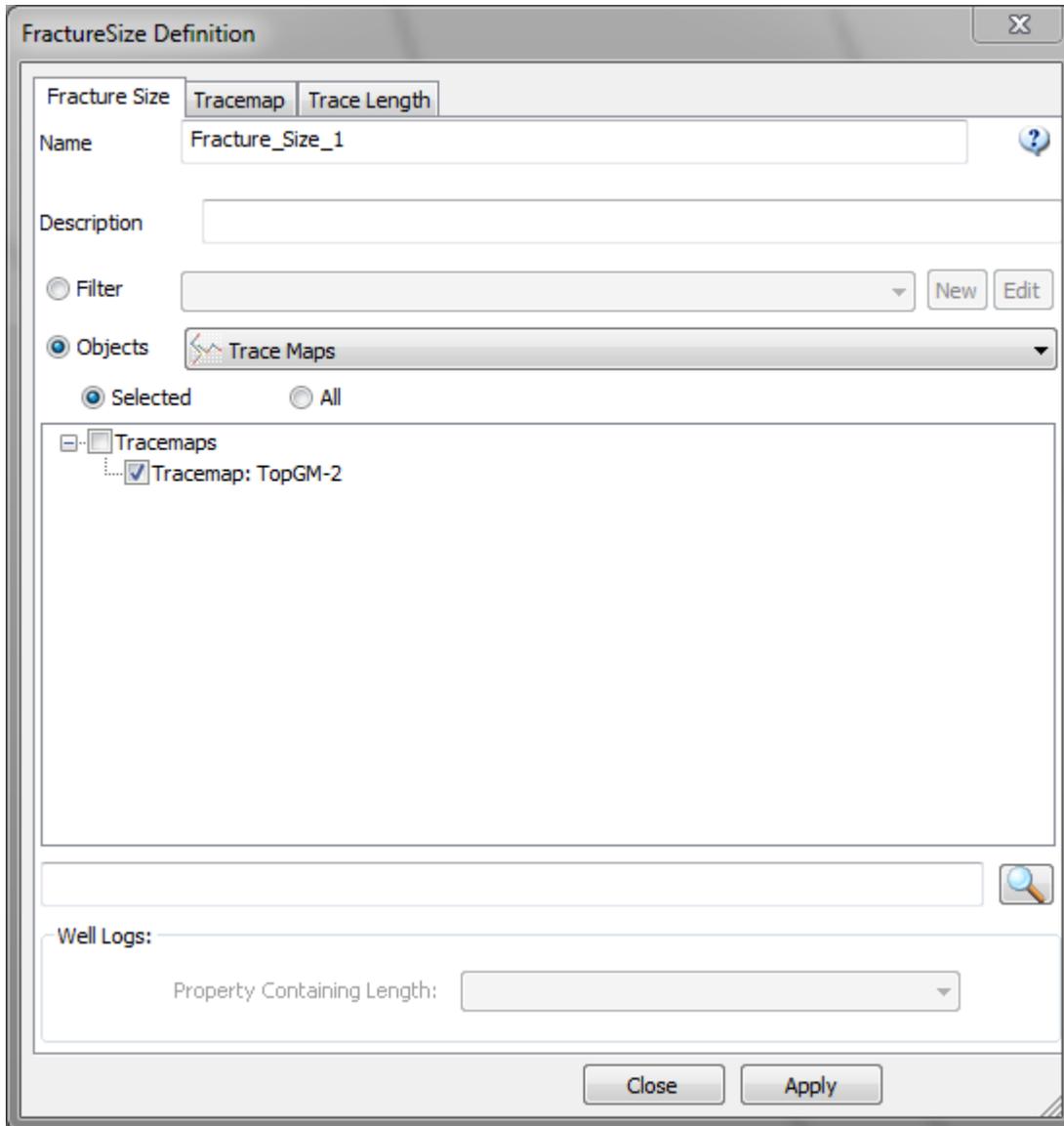
$N(L)$ = number of filled boxes,

L = box edge length

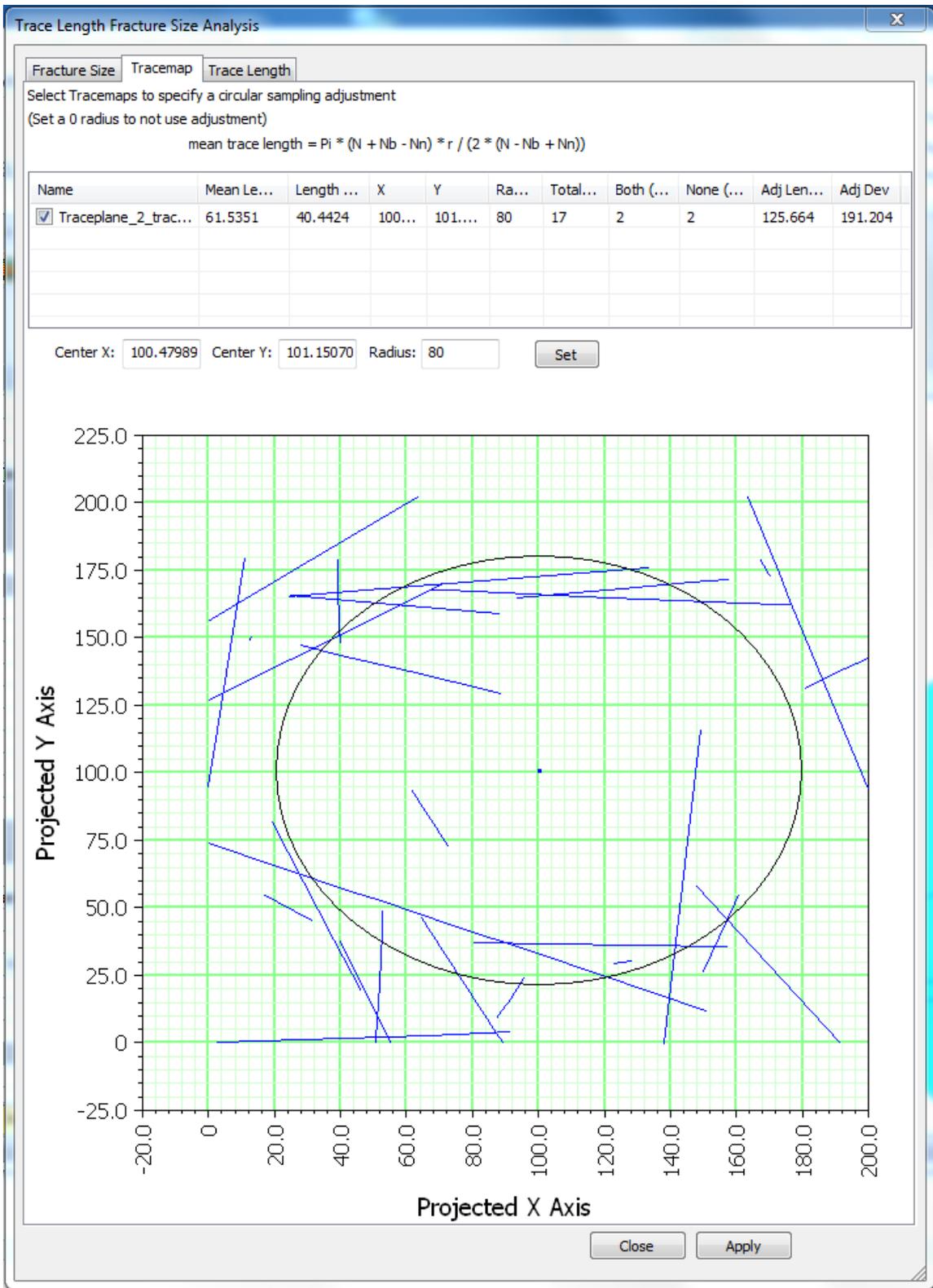
D_B = box dimension.

Trace Length Fracture Size

This allows the user to convert trace length to fracture radius. The workflow is based on the paper “Stereological relationship between trace length and size distribution of elliptical discontinuities” by Zhang, L. Einstein, H. H. and Dershowitz, W. S. (2002) *Géotechnique* **52**, No. 6, 419-433. In order to effect the conversion, you may either feed FracMan raw trace length data or upload a tracemap. The procedure is similar no matter which way you enact it. Once your trace data is loaded into the program, select **Data Analysis > Size > Trace Length Fracture Size** from the file menu.

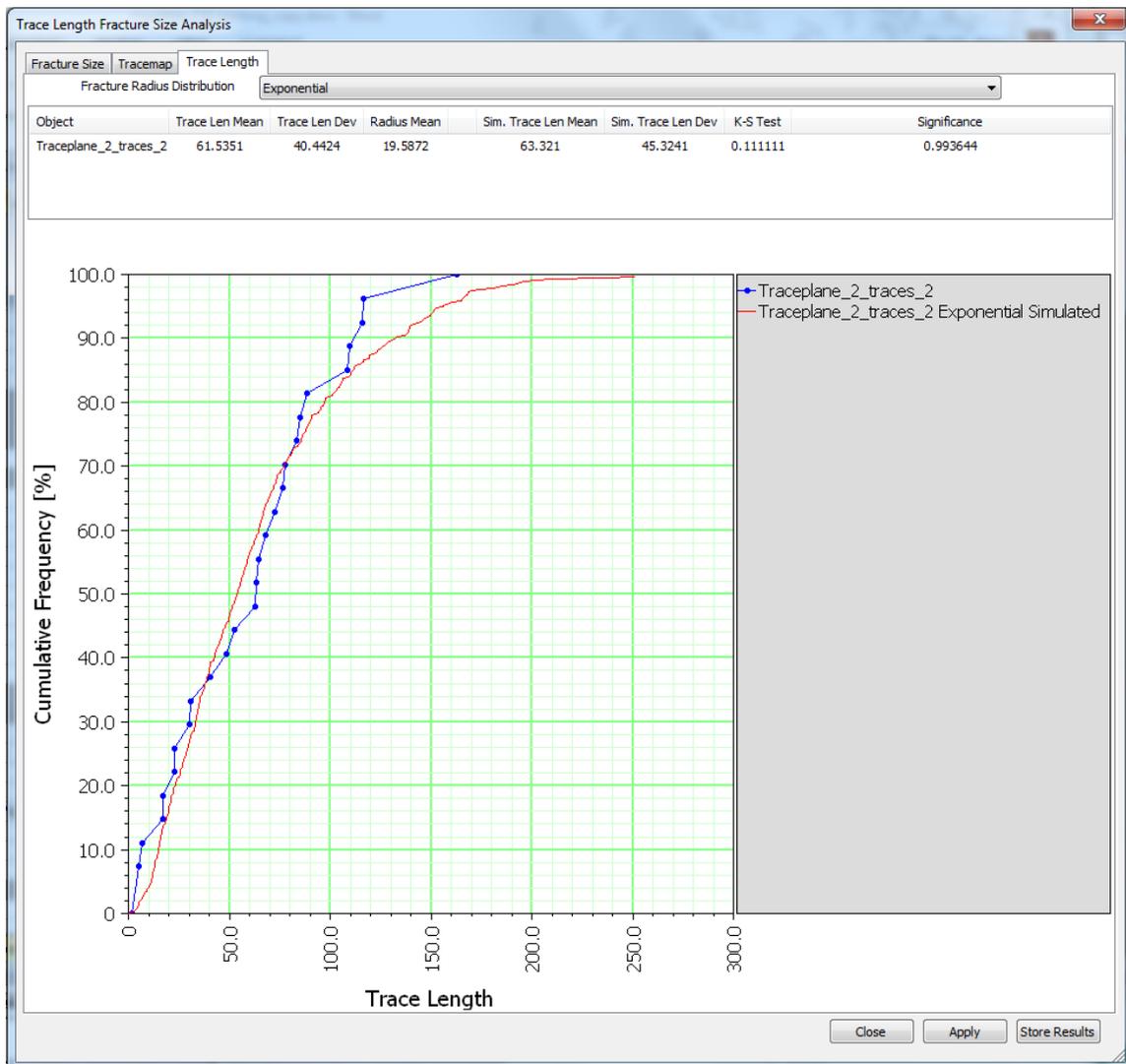


On the **Fracture Size** tab, select the tracemap or object containing trace length data (usually a well log) that you wish to analyze. You may select more than one object; datasets will be superimposed rather than combined. You may also apply a fracture filter to the selected trace data so that only certain traces are considered. If you select a well log – that is, an object that is not a tracemap – you must choose which property of the object lists the trace length by selecting the appropriate property from the dropdown menu at the bottom of the dialog box under “Well Logs:”. The name you specify in the “Name” box will be saved under the Definitions window.



The **Tracemap** tab only applies when you have chosen to analyze a tracemap. The tab allows you to apply a trace length value adjustment to account for sampling bias. Check the box next

to the tracemap(s) to have them plotted. The graph at the bottom of the tab shows the tracemap with a circular region superimposed about the tracemap’s center (looks elliptical due to the window scaling). This circle controls what portion of the traces are used for the tracelength distribution. The center and radius of the circle are controlled by the “Center X”, “Center Y”, and “Radius” boxes in the middle of the dialog box. You can change the center’s coordinates and the circle’s radius here; then click **Set** to record your changes. The adjusted length distribution will appear on the spreadsheet at the top of the dialog box in the row for the appropriate tracemap. You can control which tracemaps you wish to be considered by checking or unchecking the boxes next to their names in this spreadsheet. If you do not want to use an adjusted trace length, set the ellipse radius equal to zero.



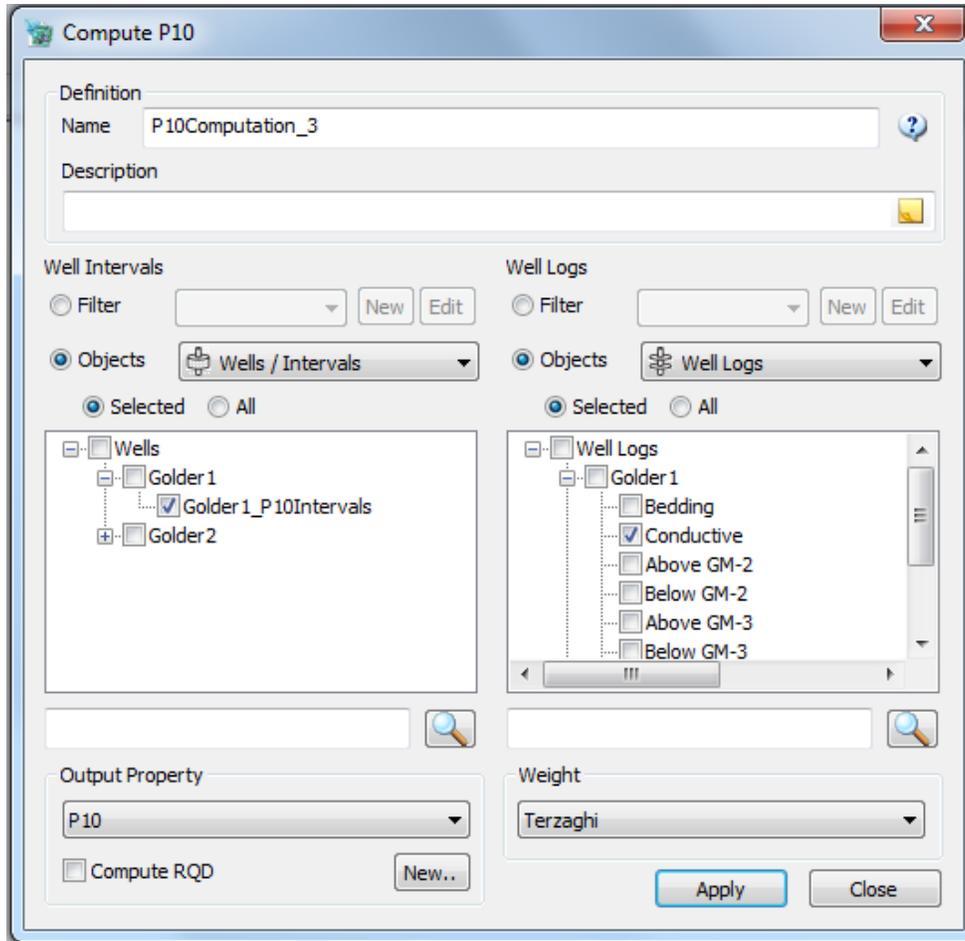
The **Trace Length** tab is where you fit curves to your trace length data in order to find which distribution most closely matches the trace data. Select which distribution you wish to compare from the dropdown menu next to “Distribution”. A red fit curve matching the selected

distribution will appear on the graph in the lower half of the window; the parameters of the curve will appear in the list of trace data at the top of the dialog box. Equations describing the calculation of the Log-Normal, (negative) Exponential, and Gamma fit curves are available in “Stereological relationship between trace length and size distribution of elliptical discontinuities” by Zhang, L. Einstein, H. H. and Dershowitz, W. S. (2002) *Géotechnique* **52**, No. 6, 419-433. The equation describing the calculation of the Power Law curve fit is available in “Derivation of parent fracture population statistics from trace length measurements of fractal fracture populations” by P. R. La Pointe (2002) *International Journal of Rock Mechanics & Mining Sciences*, **39**, 381-388.

When you have chosen the best fit for your data, click **Apply**, then **Store Results** to save the definition. This will open up two new windows with your analysis parameters and the CDF graph. The analysis will be saved under the Objects window under “Analysis”, and under the Definitions window under “Fracture Size”. If you wish to see the graph or data again, you can right-click on the definition name in the Objects window and select **View Data**, or right-click on the definition name in the Definitions window and select **Edit Definition**, which will reopen the wizard where you defined your analysis.

Computing Linear Intensity (P10)

FracMan can compute the Linear Intensity (or P10) from existing well log data. Once your well log data is loaded into the program, select **Data Analysis > Intensity > Compute P10** from the file menu.



In the **Well Intervals** box, check all the wells and/or well intervals along which you wish P10 to be calculated. In the Well Logs box, check the well logs you wish to use to calculate P10.

You can apply a filter to well log entries to ignore well log entries outside of a range of property values.

You can quickly find wells / well intervals by entering a matching string (wildcards *? included) in the search box and pressing the search button .

Select the property to store computed P10 values in or press **New..** to create a new property. These values can be used later in fracture generation or correlations.

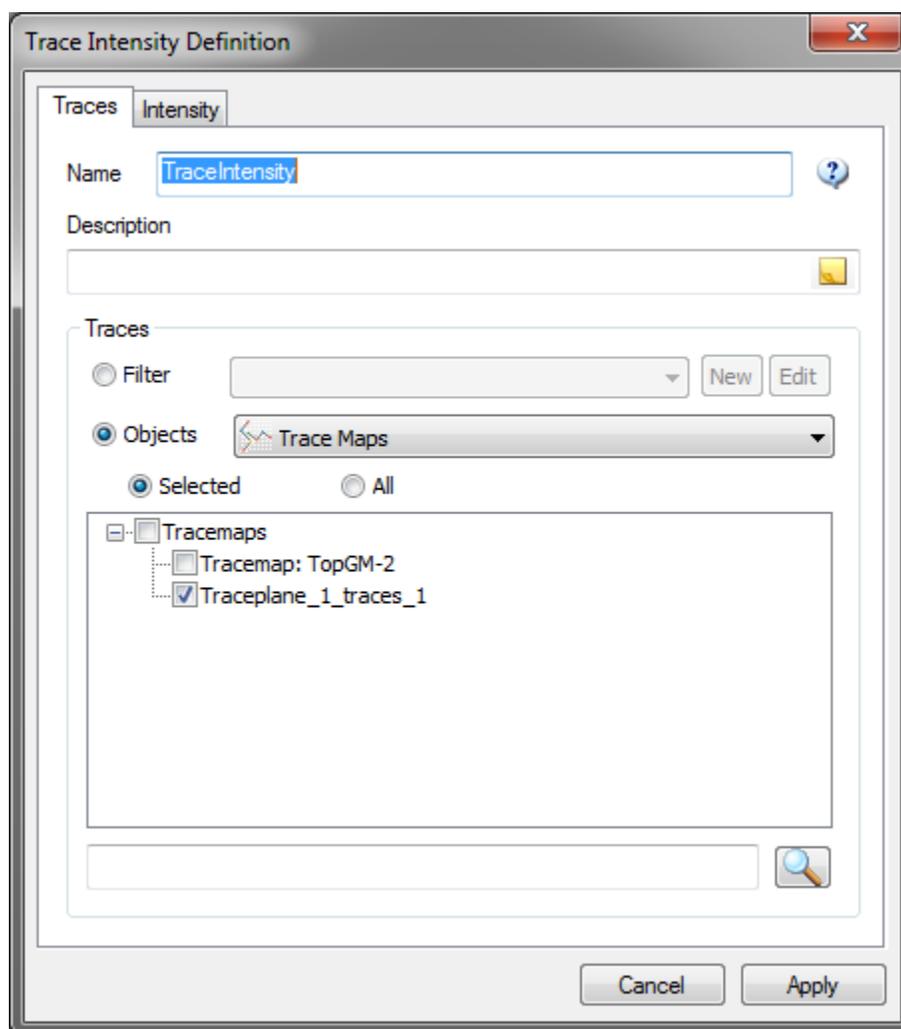
Check **Compute RQD** if you also wish to calculate this for well intervals – see Manual section [Well Log Statistics](#) for a description.

Optionally select a **Weight** parameter such as Terzaghi/Mauldon to adjust the results (see Manual section [Terzaghi / Mauldon Correction](#)).

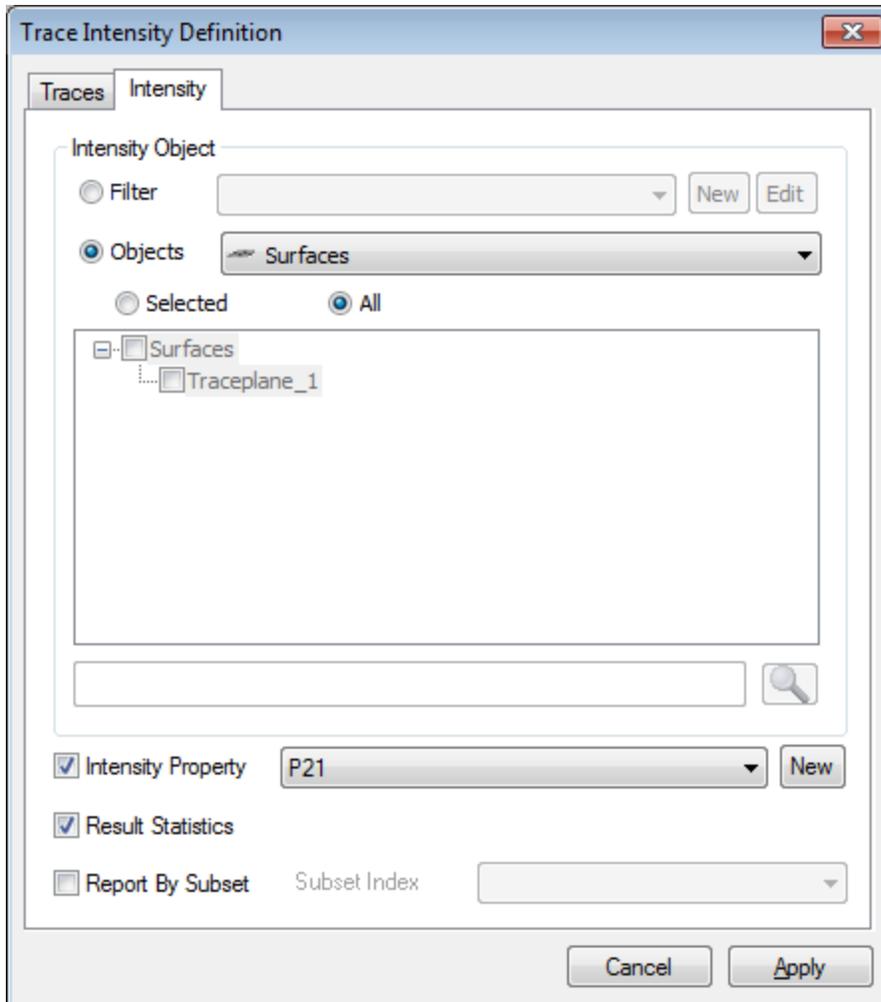
Click **Apply** then **Run** to compute P10 for the selected wells and/or intervals. The computed P₁₀ values can be viewed by right-clicking on the P10 Interval object just created in the Objects window, and selecting **View Data**, or by right-clicking on the well name in the Objects window and selecting **Edit Well**. Click on the P10 intervals tab, then the intervals will be show with the P₁₀ Value in the right-hand column, under “P10”.

Computing Trace Intensity (P21/P31)

FracMan can compute the length of fractures per unit area (P21/P31) from existing surfaces, tracemaps, and grids. Once your data is loaded into the program, select **Data Analysis > Intensity > Compute P21/31** from the file menu.



On the **Traces** tab, select the tracemap you would like to use to calculate P21/P31. You can apply a filter to tracemap entries to ignore tracemap entries outside of a range of property values.



On the **Intensity** tab, select the surface or grid that you would like to use to calculate P21/P31. Select the appropriate intensity property in the box next to “Intensity Property”. If the property doesn’t exist, press “New...” to create a new property to store the value. You can also output a statistical display of the P21/P31 values.

Click **Apply**, and then **Run** to calculate the trace intensity values. A new window will open with the resultant values.

Intensity Conversion (C31) Computation

Once you have computed P_{10} for a set of wells / intervals, you can compute a conversion factor (C31) that can be used to convert linear intensity (P_{10}) to volumetric intensity (P_{32}).

This numerical approximation is based on the doctoral research by /Wang 2005/ on stereological relationships between fracture orientation and fracture intensity. Wang’s estimator for C_{13} is based on several key assumptions:

- The method assumes one-dimensional (line) sampling; i.e. a zero-radius borehole. This implies that every fracture recorded for a given cored borehole crosses the entire

diameter of the core. In situations where the fracture radius is of a comparable size to the borehole radius, this can result in a systematic under-prediction of C_{13} .

- Wang's C_{13} assumes independence of fracture size, shape, and orientation.
- Wang's C_{13} assumes that the fracture population in a single set follows a Univariate Fisher hemispherical probability distribution. The method is not suitable for use with other types of probability distributions; it would be inappropriate to use Wang to estimate C_{13} for fractures that appear to follow a girdle distribution (Bingham/Bivariate Normal).

Given a relationship where $C_{13} \times P_{10} = P_{32}$, the conversion factor C_{13} is defined by /Wang 2005/ as:

$$C_{13} = \left[\int_0^\pi |\cos \alpha| f_A(\alpha) d\alpha \right]^{-1} \quad \text{Equation 1-8}$$

where α is the solid angle between the sampling line (borehole centerline) and the fracture normal. Assuming line sampling (P_{10}) of fractures distributed according to a Univariate Fisher hemispherical probability distribution, the theoretical probability distribution function given by /Wang 2005/ for α is given by:

$$f_A(\alpha) = \frac{1}{\pi} \int_{R_\delta} \frac{\sin \alpha}{\sqrt{\sin^2 \delta \sin^2 \rho - (\cos \alpha - \cos \delta \cos \rho)^2}} \frac{\kappa e^{\kappa \cos \delta} \sin \delta}{e^\kappa - e^{-\kappa}} d\delta \quad \text{Equation 1-9}$$

where:

κ represents the Fisher concentration parameter

α is the solid angle between the sampling line (well) and the fracture normal,

δ is the solid angle between the fracture normal and the Fisher distribution mean pole vector; and

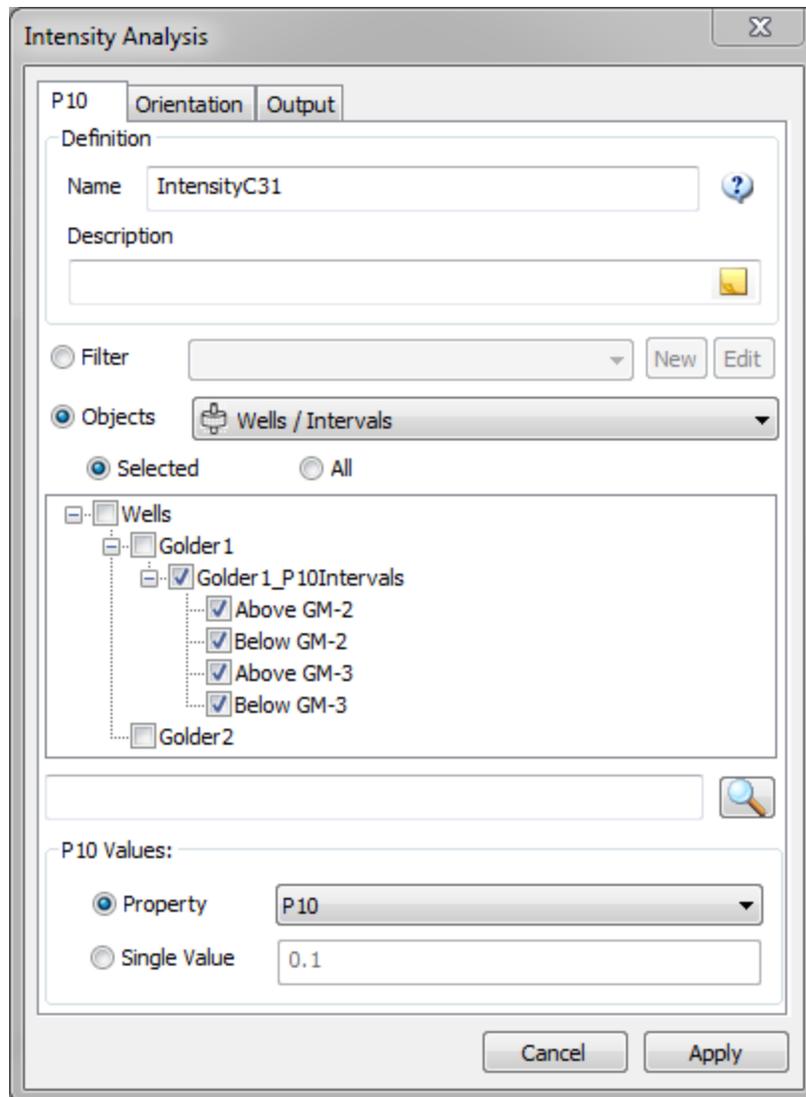
ρ is the solid angle between the Fisher distribution mean pole vector and the sampling line (well).

For α in the range $|\delta - \rho| < \alpha < \delta + \rho$, where the range of integration of R_δ is given by:

$$R_\delta = [\rho - \alpha, \rho + \alpha] \text{ if } \alpha < \rho, \text{ or}$$

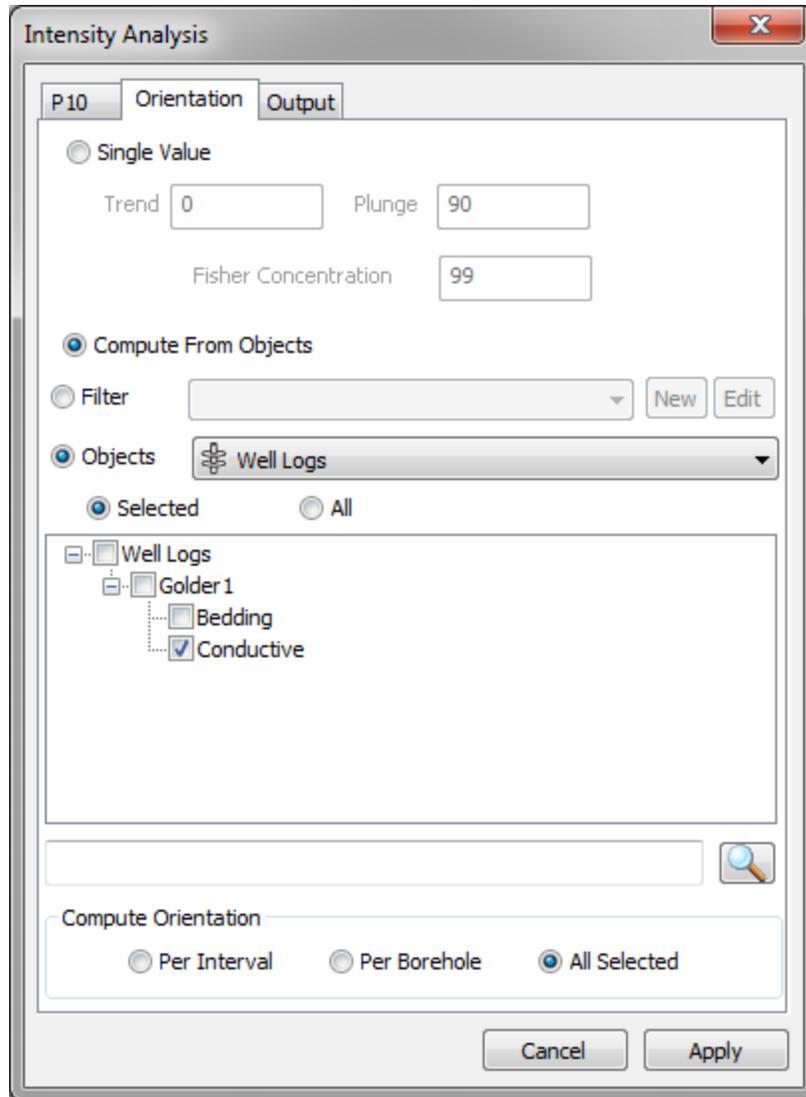
$$R_\delta = [0, \alpha + \rho] \text{ if } \alpha > \rho.$$

No closed-form solution for Wang's correction exists; rather, the correction factor is either taken from a lookup table as a function of the solid angle between the mean pole and the well centerline and the Fisher concentration parameter. Alternatively, computer codes can be used to compute the full integrations for a given set of fractures. In situations where fracture orientations can truly be modeled using a Univariate Fisher distribution, the Wang correction tends to produce the best results.



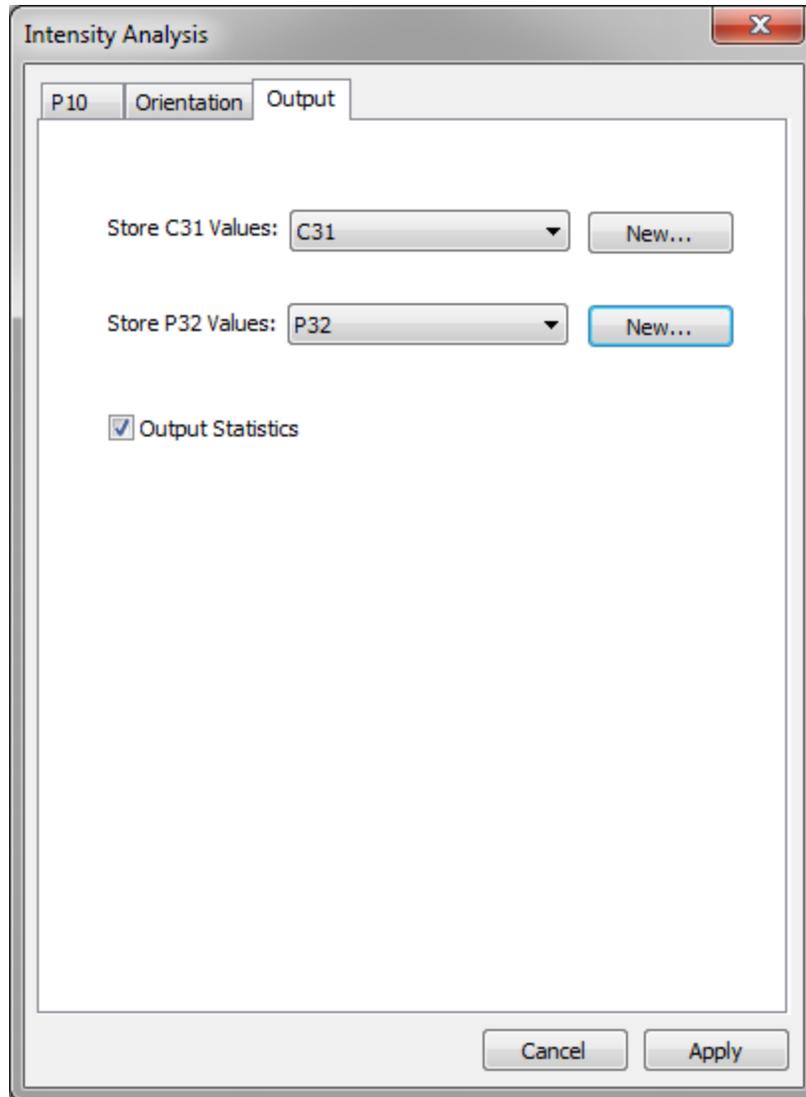
To run the conversion, select **Data Analysis > Intensity > Intensity Conversion (C31) Calculation** from the file menu. On the **P10** tab, select the well intervals you want to compute over. See [Defining Well Intervals](#) for more information about creating and editing Well Intervals.

Next you need is to specify P10 values – this can either be a property from an early p10 calculation (see [Computing Linear Intensity \(P10\)](#)) or you can specify a single value that will be applied to all selected intervals.



This **Orientation** tab is where you specify the fracture orientation population. Orientation can be specified either using one or more well log objects or by specifying a single orientation as well as a Fisher concentration value.

You can specify whether FracMan treats the orientation data by interval, by borehole, or altogether.



On the **Output** tab, you can identify where to store the resultant C31 values and optionally Computed P32 values in properties of the selected interval sets. If these properties don't exist, press "New..." to create a new property to store these values. You can also output a statistical display of the C31 values.

Click **Apply**, **Generate**, and then **Close** the dialog box. The results are written into the specified C31 and P32 on the interval set objects (to view data, right-click on the well/interval, then select **Edit Well/Intervals**):

Interval Set Definition - Golder1_P10Intervals

Edit Intervals Properties

Insert Full Length Add Property...

Golder1_P10Intervals

	Name	Depth Start[m]	Depth End[m]	P10[1/m]	RQD	RQDQ	RQD13	C31	P32[m ² /m ³]
1	Above GM-2	2440.000	2590.000	0.32	17.2656	17.2665	17.2648	1.969281	0.78757119
2	Below GM-2	2590.000	3045.000	0.107692	6.68709	6.68731	6.68709	1.637856	0.82476513
3	Above GM-3	3660.000	3965.000	0.24918	12.9971	12.9971	12.9971	1.856349	0.87359614
4	Below GM-3	3965.000	4260.000	0.0813559	14.4328	14.4328	14.4328	1.756893	0.76248536
5									

Cancel Apply

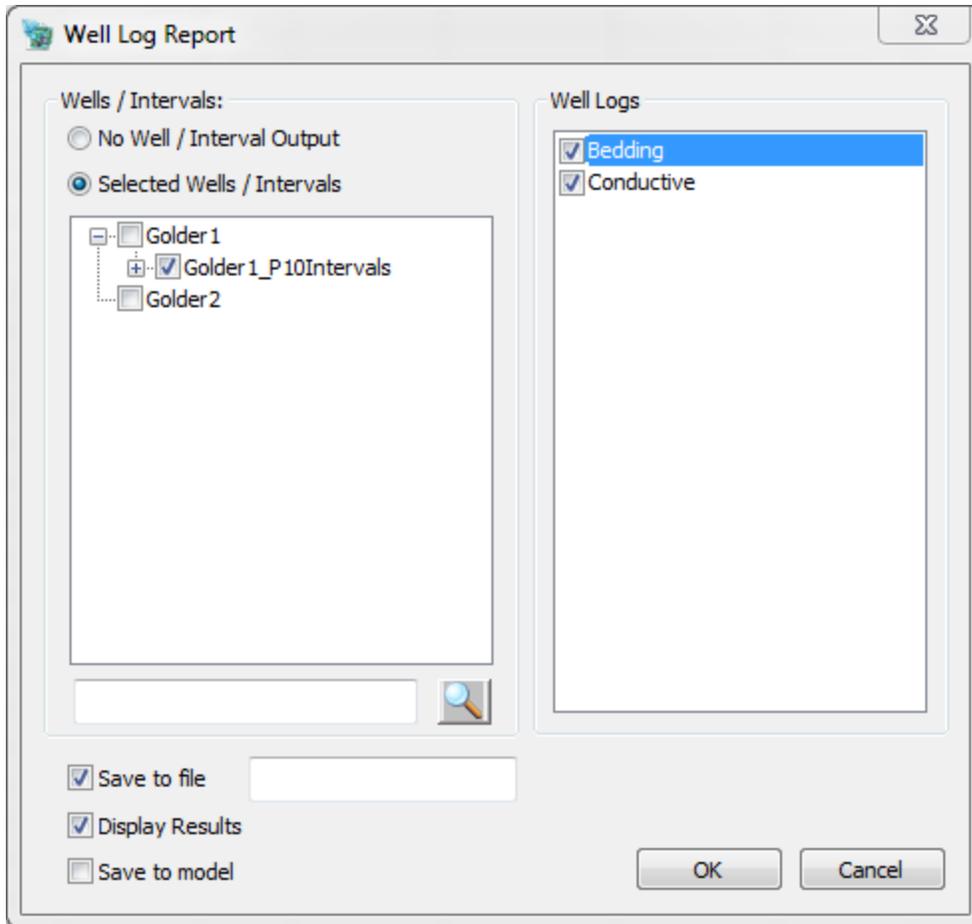
They are also output in a statistics display

Intensity Analysis IntensityDef_2

Well	320B
Interval Set	320B_Intervals
Interval	320B_IntervalsFull
MD Top	0
MD Bottom	10.6
Orientation	303.453, 23.247
Fisher Kappa	3.8249326
Solid Angle	53.936063
C31	1.968928
P10	0.1
P32	0.1968928

Well Log Report

This function works similar to, but is more comprehensive than the **Statistics** feature, discussed below. Select one or more well logs in the Objects window. Then click **Data Analysis > Well Log Report...** on the file menu.



Select the desired wells/intervals you would like to create a report for and click **OK**. This will open a window of the minimum, maximum, count, average, and standard deviation for each of the selected log's fracture properties. You may save the report directly to file in the dialog box, or save/copy the output table by right-clicking on the data and selecting either **Copy** or **Save to File**.

Statistics

This is the most straightforward of static analyses. Unlike the **Well Log Report** feature, it can be applied to fracture sets as well as well logs. Select one or more fracture sets or logs in the Objects window and hit **Static Analysis > Statistics** on the file menu. A new window will open giving the maximum, minimum, mean, and standard deviation for each of the object's properties. If more than one item is selected, columns will appear for each set individually and for all of the sets combined. Besides the property averages, other statistics given include the number of fractures in the set, the total fracture area and volume, the fracture intensity, and the Fisher mean pole of the set. Intersection statistics are also given for well logs.

If you wish to save the statistics for your sets, right click on the data in the chart window and select **Copy**, which will copy the data to the clipboard as a table, or **Save to File**, which will save

the data as an ASCII text file in whatever directory you specify. You may also graph a histogram of any property in the table by right-clicking on that property and selecting **Histogram**. A new window will open containing a histogram chart of the chosen value for every set in the table. This is not a histogram of the property values themselves, but of whatever statistical value you selected – for instance, minimum plunge. The histogram will display only that value. If you have statistics for only one set, only one value will appear on the chart.

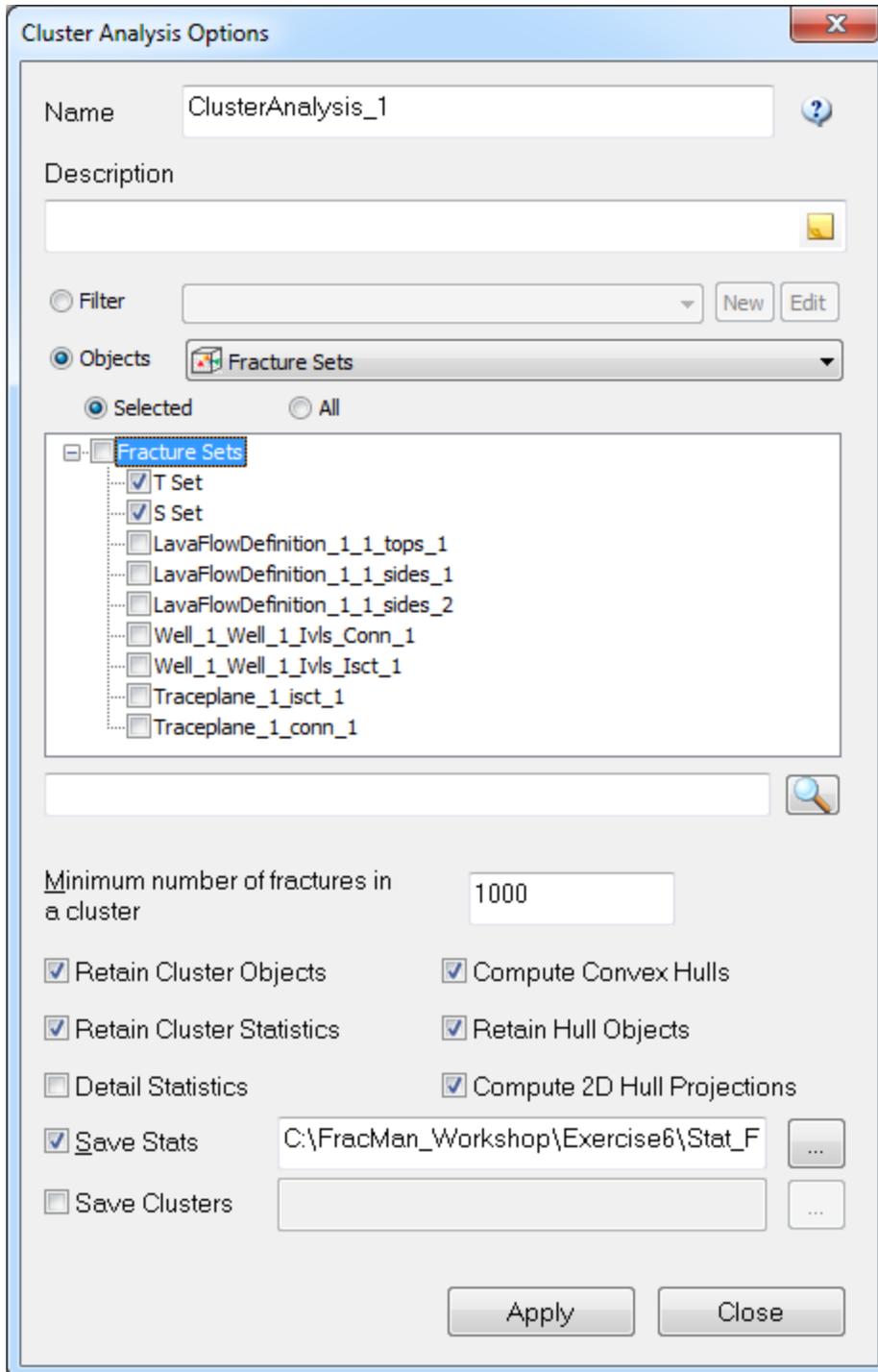
Static Analysis

Fracture-Based Analyses

Cluster

When Cluster Analysis is run, FracMan goes through the selected fracture sets and picks out isolated groups of self-connected fractures, thus showing how the set is compartmentalized. Compartmentalization in reservoirs where fractures dominate permeability often leads to wells that produce at different rates and volumes than expected. Within compartments, pressure communication can be nearly instantaneous, while nearby wells in different compartments may have very little communication. Estimating the degree of compartmentalization is therefore important at all stages of field development in order to properly-engineer the field and to provide realistic recovery estimates and rates for financial decisions.

To conduct a cluster analysis, you may either select Static Analysis > Fracture > Cluster from the file menu, or you may right-click on any fracture set name in the Objects window and select Analysis > Cluster.



Enter the minimum number of connected fractures that should be considered a cluster; any groups with less fractures than this minimum will be ignored. You may also select a Fracture Filter to define which fractures should be considered as part of the cluster analysis. If you do not use a fracture filter to narrow the data, the analysis will consider all fracture sets in your file, not just the one you happened to click on. You must also specify whether you wish to save the

analysis definition and under what name. If you check “Save Results”, FracMan will create a text file containing volume, area, and size information for every cluster found by the analysis. Hit the “...” button to specify where this list will be saved and what it is to be called.

- Retain Cluster Objects – Check this to retain a fracture reference set object for each found cluster.
- Retain Cluster Statistics – Check this to retain the cluster statistics output as an object in the model
- Compute Convex Hulls – Check this to compute a convex hull around each cluster object.
- Retain Hull Objects – Check this to retain the convex hull region objects for visualization.
- Compute 2D Hull Projections – Check this to compute the projection of the hull objects onto the xy plane.
- Detail Statistics – Check this to generate a detailed report of each cluster’s statistics.

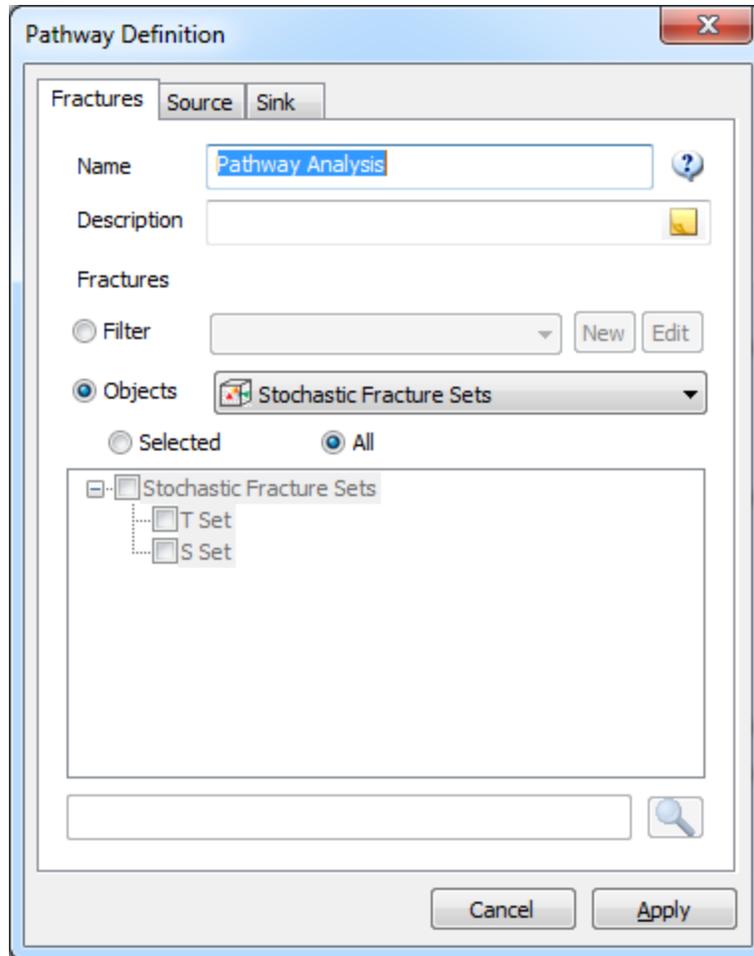
When you have finished defining your analysis, hit OK.

When the analysis is complete, a data window will appear containing the aforementioned cluster data. You may save or copy this by right-clicking. A new object will also have appeared in the Objects window under “Analysis” with the name you specified in your analysis definition. All clusters chosen by the analysis will be listed as children of it, numbered by size, with the largest cluster being “Cluster1”, the second-largest “Cluster2”, etc. Unless otherwise specified, each cluster will be shown in a different color in the Visualization window.

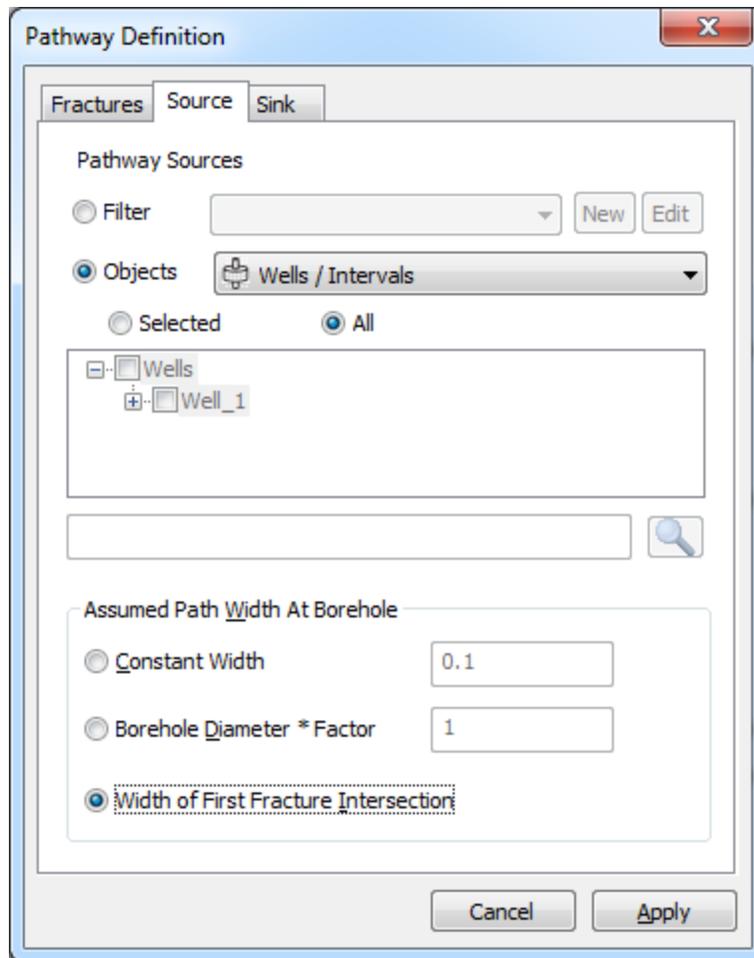
Pathway

This calculates the geometric connections, or pathways, between two objects (either wells or surfaces) via a fracture network. FracMan computes pathways by searching through the fracture system between user-defined start and end points. It then calculates pathway properties, such as pathway length and conductance, and reports either the shortest or the most transmissive path, depending on which the user requests.

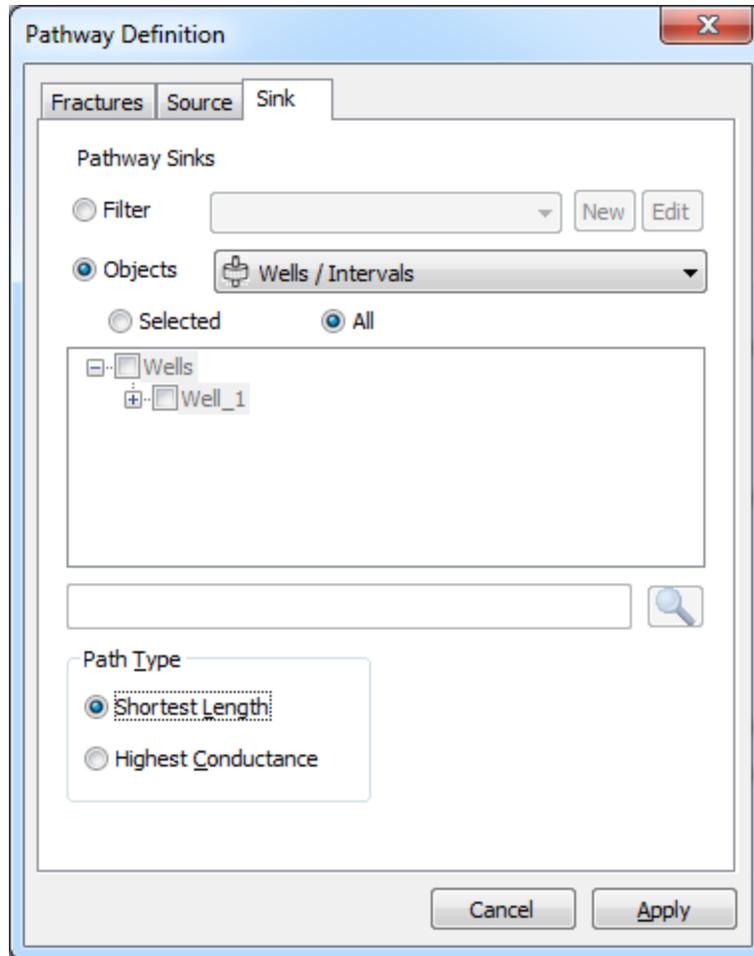
To conduct a pathway analysis, you may right click on a fracture set object in the Objects bar and select Analysis > Pathway, or you may select Static Analysis > Fracture > Pathway from the file menu.



On the **Fractures** tab, first specify a **Name** and optionally a **Description** for the analysis. A default name is provided. Select all the fracture sets that you want included in the analysis. You can select the general category 'Fracture Sets' to display all of the available sets, choose a specific category of fracture sets such as 'Stochastic Fracture Sets', or use a Filter to select the sets. The search functionality provided at the bottom of the dialog can help you find the sets of interest when there are a large number displayed.



The **Source** tab allows you to choose one end-point for the path. When you are using a well as either the source or sink object (or both), you should define how you want the path width at the borehole to be determined. You can use a constant value, a multiple of the borehole diameter or the width of the first fracture intersection.



The **Sink** tab allows you to choose other end-point for the path. You must also tell FracMan which pathway you want it to calculate: “Shortest Length”, which is determined by total connection distance, that is, length of fractures linking the source and sink; or “Highest Conductance”, which will give the pathway with the highest total conductivity of all fractures combined adjusted for length. The conductance of a path is calculated as:

$$C_P = \frac{\sum L_n}{\sum \frac{L_n}{W_n \times T_n}}$$

Where C is the path conductance, L_n is the distance between the midpoints of two adjacent intersections, W_n is the average of two adjacent intersection widths and T_n is the transmissivity of the fracture.

When the analysis is complete, FracMan will show a report of the pathway’s statistics – that is, the path’s source and sink, conductance and length, and the number, area, and volume of its component fractures. These statistics can be saved or copied from their right-click menu. A new object will also appear under the Analysis object folder in the Objects window, with the

report object, the pathway, and the connected fractures along the pathway as its children objects. In the Visualization window, the pathway is depicted as a series of triangles built from connecting the fractures intersections along the path.

Critical Stress

Critical Stress is defined as the magnitude of shear stress divided by shear strength. Given a stress field and one or more fracture sets, FracMan can compute which fractures are critically stressed with respect to a user-specified strength model. In FracMan, a stress field can be represented either by using constant values or by a grid whose cell properties correspond to the particular stress tensor format.

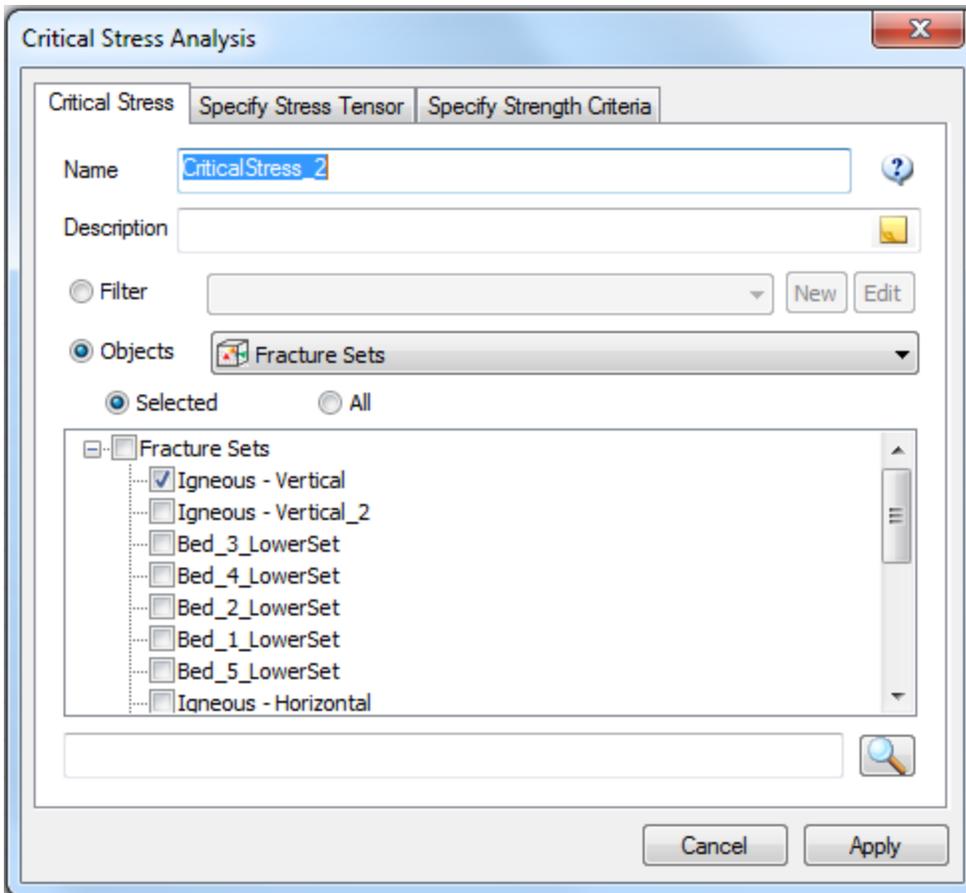
The Mohr model assumes that the shear strength of a fracture is given by a function, $f(\sigma)$, and shear failure occurs if

$$|\tau| \geq f(\sigma)$$

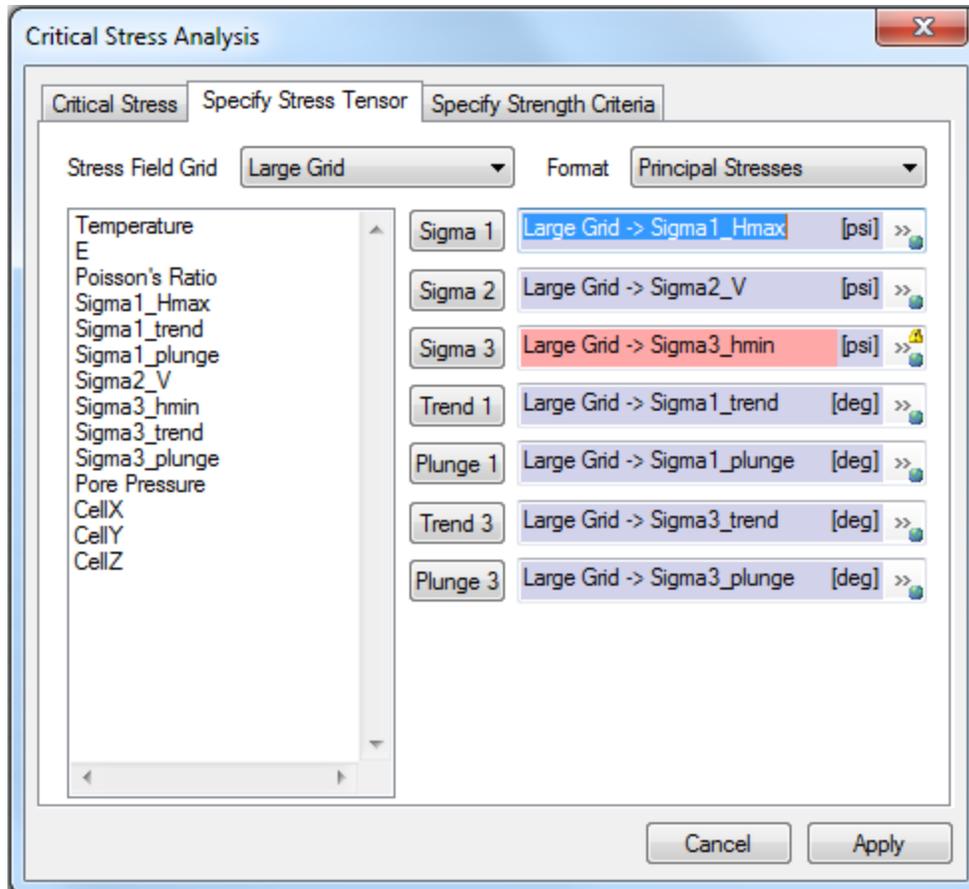
In other words, if the shear stress across the fracture exceeds its shear strength, the two opposing faces slip against one another. In this case, the fracture is said to be *critically stressed*, and its permeability may be enhanced.

To Perform a Critical Stress Analysis:

On Objects Window, select the fracture sets to analyze by ctrl-clicking on their labels. Right click and select **Critical Stress...** from the **Analysis** submenu. Alternatively, select **Static Analysis | Fracture | Critical Stress** from the top menu.



Specify Stress Tensor



Stress Field Grid: (Optional, but will show any grids in the model if any are present) Select the grid whose cell properties correspond to the desired stress tensor in each cell. When a particular grid is selected, its cell properties are listed in the Cell Properties list box. If no grid is selected, constant values may be entered using the buttons on the right side of each text box to adjust units if necessary.

Format: Specify the format of the stress tensor. Two stress tensor formats are supported: the six independent values of a 3x3 symmetric matrix (“Symmetric Matrix”) or the three principal stress values, σ_1 , σ_2 , and σ_3 , along with the trend and plunge (in degrees) of the *first* and *third* principal vectors (“Principal Stresses”).

Any stress tensor can be represented by a 3x3 symmetric matrix \mathbf{M} , so that the total stress \mathbf{p} across a fracture with unit normal \mathbf{n} is given by

$$\mathbf{p} = \mathbf{M} \cdot \mathbf{n}$$

The signed *normal* stress across this fracture is the scalar projection of \mathbf{p} onto \mathbf{n} :

$$\sigma = \mathbf{p} \cdot \mathbf{n}$$

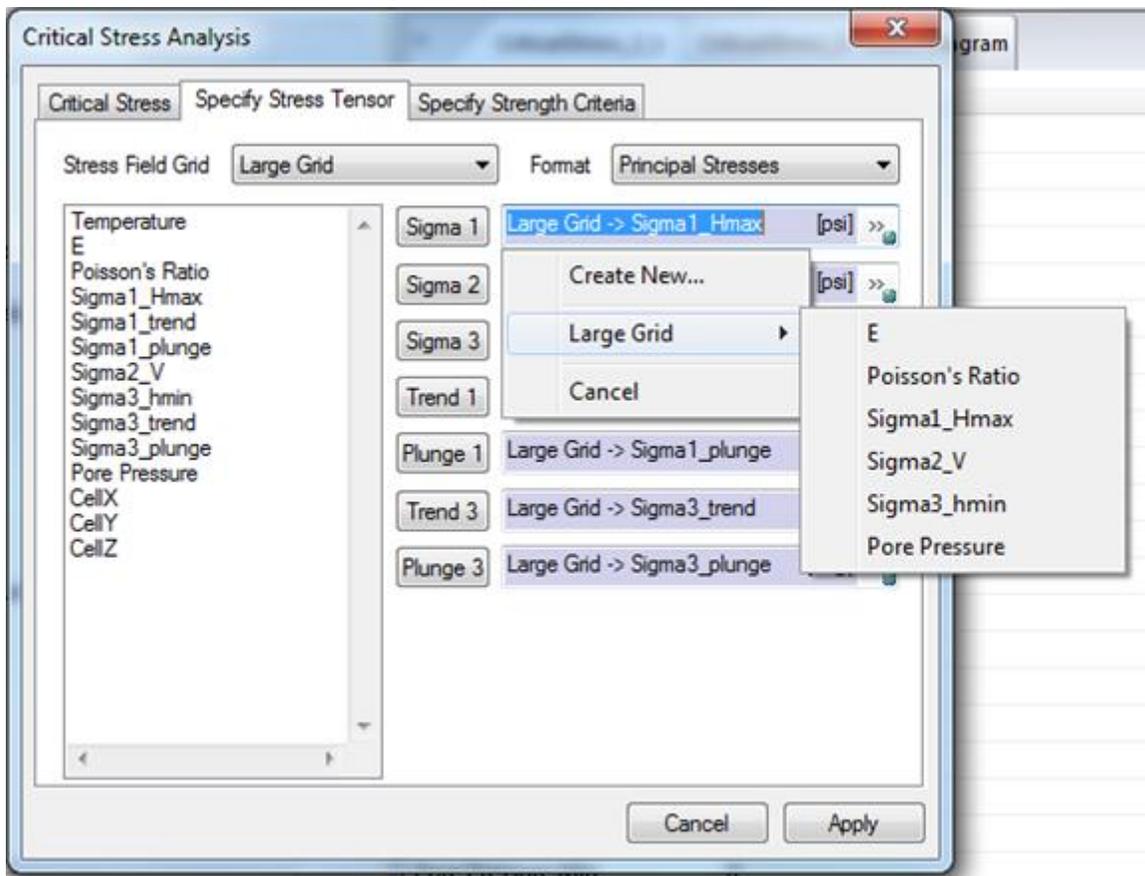
FracMan uses the sign convention that *positive* normal stresses ($\sigma > 0$) are *compressive*, and negative normal stresses are *tensile*. The normal stress vector given by

$$\sigma = \sigma \mathbf{n}$$

therefore represents the force per unit area that the *negative* side of the fracture exerts on the *positive* side (the side of the fracture pointed to by the normal). The *shear* stress τ exerted by the negative side on the positive side is therefore

$$\tau = \mathbf{p} - \sigma$$

If using a grid to provide spatially varying stress components, specify the stress tensor values in each parameter text box by selecting the grid property name in the left list box and then clicking the button of the appropriate field. The name of the property will appear in the text boxes to the right of this button. Alternatively, since spatial parameters are supported for each of the stress tensor values, click on the symbol located on the far right side of each parameter and select the desired parameter or grid property value.



Specify Strength Criteria

Frictional Model: FracMan supports three strength models:

Mohr-Coulomb: In this case, fracture shear strength is given by

$$f(\sigma) = S_0 + \sigma \tan \phi_f$$

where S_0 is the *cohesion* of the fracture, σ is the fracture normal stress, and ϕ_f is the *friction angle* in degrees.

Barton-Bandis: In this case, fracture shear strength is given by

$$f(\sigma) = \sigma \tan\left(JRC \log 10\left(\frac{JCS}{\sigma}\right) + \phi_f + \phi_d\right)$$

where σ is the fracture normal stress, JRC is the *joint roughness coefficient*, JCS is the *joint wall compression strength*, ϕ_f is the *friction angle* in degrees and ϕ_d is the *dilation angle* in degrees. **Warning** – this frictional model should **only** be used when the normal stress is less than JCS .

High Stress Barton-Bandis: A form of Barton-Bandis strength model appropriate when the normal stress is greater than the joint wall compression strength.

$$f(\sigma) = \sigma \tan\left(JRC \log 10\left(\frac{S1-S3}{\sigma}\right) + \phi_f\right)$$

where σ is the fracture normal stress, JRC is the *joint roughness coefficient*, $S1$ and $S3$ are the largest and smallest principal stresses respectively, and ϕ_f is the *friction angle* in degrees.

Friction angle: This parameter is related to the coefficient of static friction by

$$\mu_s = \tan \phi_f$$

A friction angle of 0 implies no friction, and an angle of 90 implies infinite friction.

Cohesion: (Mohr-Coulomb) The constant force per unit area that resists shearing, independent of the normal stress.

Dilation angle: (Barton-Bandis) The angle that the jagged features of the fracture make with its average plane. Like the friction angle, an angle of 0 implies a perfectly smooth surface and an angle close to 90 implies a very rough surface.

Joint Roughness Coefficient (JRC): (Barton-Bandis) An empirical coefficient which describes the “roughness” of the fracture surface, with 0 for perfectly smooth surfaces and 20 for very rough surfaces.

Joint Wall Compression Strength (JCS): (Barton-Bandis)

Pore Pressure: Can be specified as a constant value, a fracture property or a spatially varying grid property.

Increase Permeability %: The percentage by which to increase the existing permeability value of critically stressed fractures. A negative value will decrease the permeability. A value of 0 leaves permeability unchanged.

Store stress normalized by Sigma 1: Normalizes the stress with respect to the largest principal stress. This option will also affect the displayed Mohr diagram axes.

Tendency Parameters (Mohr Coulomb Only): Causes extra outputs to be generated and saved as fracture properties (see following section on [Fault Reactivation Potential](#) for details).

Generate Mohr Diagram: Will display a Mohr Circle with fractures plotted using their shear stress and normal stress or effective normal stress (depending upon choice of **Store stress normalized by Sigma 1**).

Report Fractures: Report one result per fracture. If tessellated fractures are present, the resulting properties will be averages over all the fracture faces. The Mohr diagram will only show one point for each fracture.

Report Fracture Faces: Report one result for each fracture face. If the fractures are not tessellated, the results will be the same as for checking **Report Fracture**. If tessellated fractures are present, there will be one result for each fracture face and each point on the Mohr diagram will reflect one fracture face.

Click **Apply** to save the definition and **Compute** to run it. The new critical stress definition will appear in the Definitions window under “Critical Stress”. To run the analysis later, click on the definition name and select **Run Critical Stress**

Fault Reactivation Potential

A knowledge of the reactivation potential of faults is a critical issue in the development of hydrocarbon reservoirs, where hydraulic fracture stimulation treatments are routinely applied to

enhance reservoir permeability; the associated reservoir pore pressure increases have also been shown to sometimes induce undesirable seismicity.

The effects of stress field changes on fault kinematic behaviour need to be understood, and fault reactivation potential should be estimated before stimulation treatment. In this section, we have described a slip tendency analysis based on frictional constraints to assess the likelihood of fault reactivation from potential hydraulic fracture stimulation. The principal aim of this technique is to test the likelihood of induced seismicity along fault structures close to possible well injection or stimulations locations with certain orientations from the perspective of fault reactivation related to stress field perturbation.

Fault Slip and Reactivation Potential

Fault slip tendency analyses the potential for slip along any fault orientation with respect to the ambient stress field is investigated and from this it is possible to assess the fault reactivation potential.

To do this we calculate the fault local shear and normal stresses along mapped and suspected faults of the reservoir, and evaluate the slip potential, and where possible compare the results with recorded and analysed fault behaviour.

Slip tendency is the ratio of resolved shear stress to resolved normal stress on a surface. It is based on Amonton's law that governs fault reactivation:

$$\tau = \mu_s \cdot \sigma_{neff}$$

where τ is the shear stress, σ_{neff} the effective normal stress ($= \sigma_n - P_f$), and μ_s the sliding friction coefficient (Byerlee, James D., July 1978, "Friction of Rocks". *Pure and Applied Geophysics*. **116** (4-5): 615–626). According to this law, stability or failure is determined by the ratio of shear stress to normal stress acting on the plane of weakness and is defined as the slip tendency T_s . Slip is likely to occur on a surface if resolved shear stress (the component of shear stress that is resolved in the direction of slip), τ , equals or exceeds the frictional sliding resistance. Hence the slip tendency is given by:

$$T_s = \tau / \sigma_{neff} \geq \mu_s$$

The shear and effective normal stress acting on a given plane depend on the orientation of the planes within the stress field that is defined by principal effective stresses:

$$\sigma_{1eff} = (\sigma_1 - P_f) > \sigma_{2eff} = (\sigma_2 - P_f) > \sigma_{3eff} = (\sigma_3 - P_f); \quad (\text{Jaeger et al., 2007}):$$

$$\sigma_{neff} = \sigma_{1eff}^2 \cdot l^2 + \sigma_{2eff}^2 \cdot m^2 + \sigma_{3eff}^2 \cdot n^2; \text{ and}$$

$$\tau = [(\sigma_1 - \sigma_2)^2 \cdot l^2 \cdot m^2 + (\sigma_2 - \sigma_3)^2 \cdot m^2 \cdot n^2 + (\sigma_3 - \sigma_1)^2 \cdot l^2 \cdot n^2]^{1/2}.$$

where l, m and n are the direction cosines of the fracture (or fault) plane's normal with respect to the principal total stress values, σ_1 , σ_2 and σ_3 , respectively.

The dilation of faults and fractures is largely controlled by the resolved normal stress which is a function of the lithostatic and tectonic stresses, and fluid pressure. Based on the above equations, the magnitude of normal stress can be computed for surfaces of all orientation within a known or estimated stress field.

The normal stress component can be normalized by comparison with the differential stress to give the dilation tendency, T_d , for a fault surface segment defined by:

$$T_d = (\sigma_1 - \sigma_n) / (\sigma_1 - \sigma_3)$$

The fault leakage factor T_L is defined as the pore fluid pressure normalised by the difference between the normal stress and shear stress components acting on the fault surface,

$$T_L = P_f / (\sigma_n - \tau)$$

This slip and dilation tendency analysis is a technique that permits rapid and easy visual assessment of stress states and related potential for fault activity and reactivation.

Slip, dilation and leakage tendency stereoplots can be obtained by solving the above equation for all surface facets orientations constituting the fault structure in 3D space (see Manual section [Geomechanics Stereonet](#)).

Critical Stress Outputs

Once the critical stress analysis has completed, the selected fracture sets will contain new properties:

Critical_Stress is the magnitude of shear stress divided by shear strength.

Critical_StressFactor equals the shear strength minus the shear stress magnitude.

Critically_Stressed is just a Boolean value (True if **Critical_Stress** > 1 for any part of the fracture).

This property value will be 1.0 for fractures that are critically stressed, and 0.0 otherwise.

EffectiveNormalStressMagnitude is the normal stress magnitude minus the pore pressure.

ShearStrength can be either Mohr-Coulomb, Barton-Bandis or High Stress Barton-Bandis.

ShearStressMagnitude is the magnitude of the shear stress.

Additional properties if **Tendency Parameters (Mohr Coulomb Only)** checked:

Reactivation_Pressure is the pressure required to reactivate the fault.

Reactivation_PressureIncrement is the pressure change required to reactivate the fault.

Tendency_Dilation is the normal stress component normalized by comparison with the differential stress.

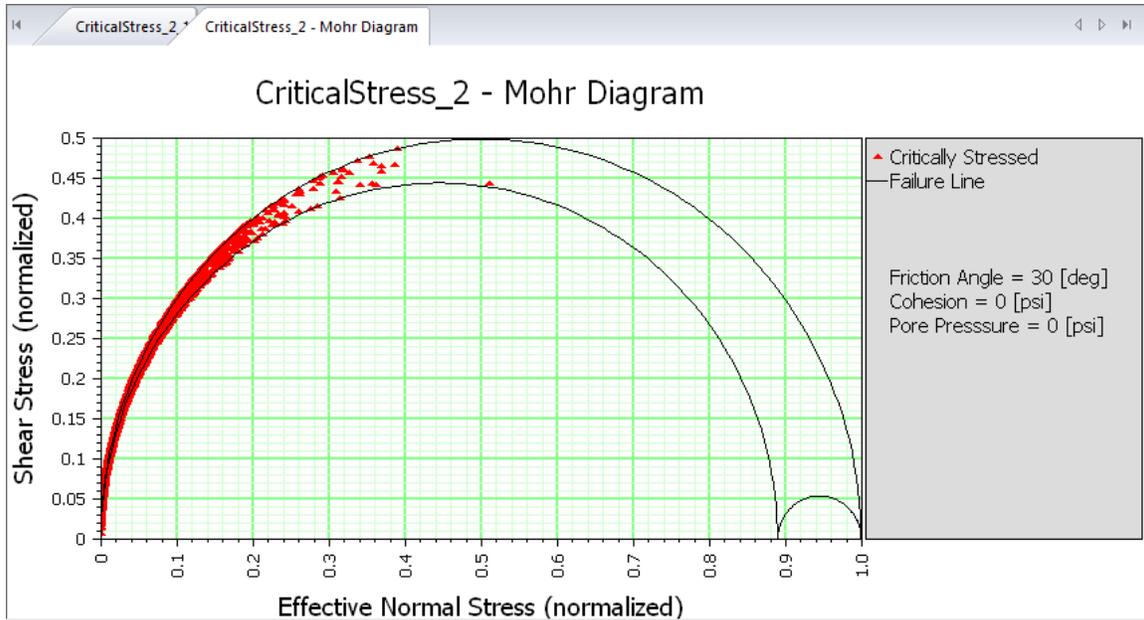
Tendency_Leakage is the pore fluid pressure normalised by the difference between the normal stress and shear stress components acting on the fault surface.

Tendency_Slip is the ratio of shear stress to normal stress acting on the plane of weakness.

The Visualization window will display the statistics of the analysis for the fractures.

CriticalStress_2_1		CriticalStress_2 - Mohr Diagram
CriticalStress_2		
Object	Igneous - Vertical	
Total Count	1371	
Number Critical	1371	
Critical Stress Min	1.5040258	
Critical Stress Max	92.264107	
Critical Stress Mean	9.7134404	
Critical Stress Dev	9.8116608	
Stress Factor Min	-0.067435719	
Stress Factor Max	0.20682387	
Stress Factor Mean	0.15100542	
Stress Factor Dev	0.045042338	
Shear Stress Min	0.0079309177	
Shear Stress Max	0.48700643	
Shear Stress Mean	0.22610838	
Shear Stress Dev	0.099453531	
Effective Normal Stress Min	0.00014888511	
Effective Normal Stress Max	0.51223373	
Effective Normal Stress Mean	0.075102951	
Effective Normal Stress Dev	0.067691828	
Pore Pressure Min	0	
Pore Pressure Max	0	
Pore Pressure Mean	0	
Pore Pressure Dev	0	

If you selected **Generate Mohr Diagram** on the **Specify Strength Criteria** dialog, then you will also see the fractures plotted in the Visualization Window. Each point represents one fracture or fracture face (depending upon the choice of the **Report Fractures** or **Report Fracture Faces** option).



Fracture/Fracture Traces Compute Height/Length

Grid-Based Analyses

There are a number of related analyses that populate grid property data based on fractures within the grid region.

[Fracture Count, P32, P33](#)

[Sigma Factor](#)

[Export Cell Info](#)

[Fracture Upscaling](#)

[Grid Layer Property Report](#)

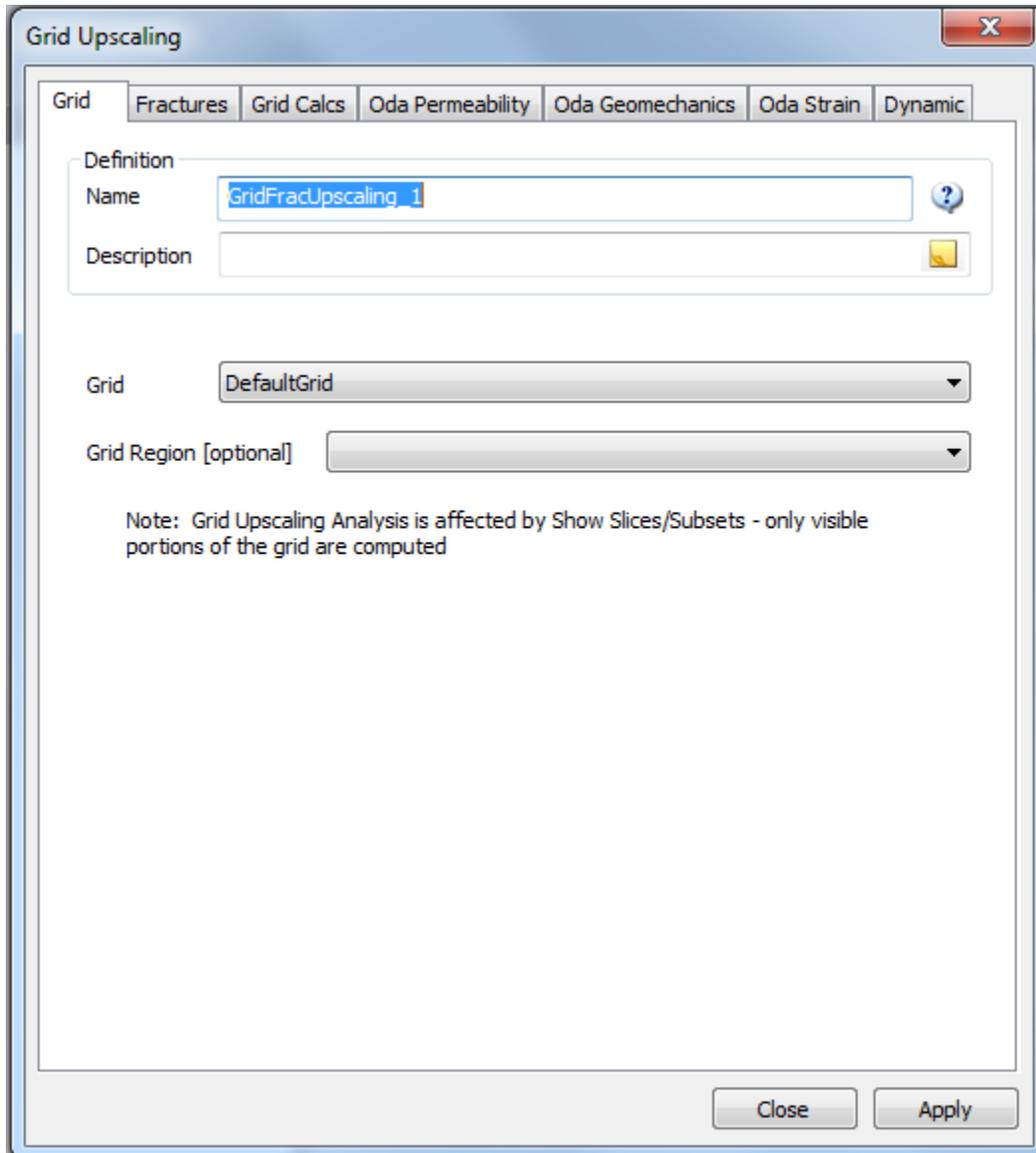
Export Cell Info

This function, which can also be reached by right-clicking on the grid in the Objects window and selecting **Analysis > Export Cell Info**, outputs an ASCII text file - containing the axis trend, plunge, and length, the location in the grid, and the coordinates of the center of each grid cell - to the specified directory.

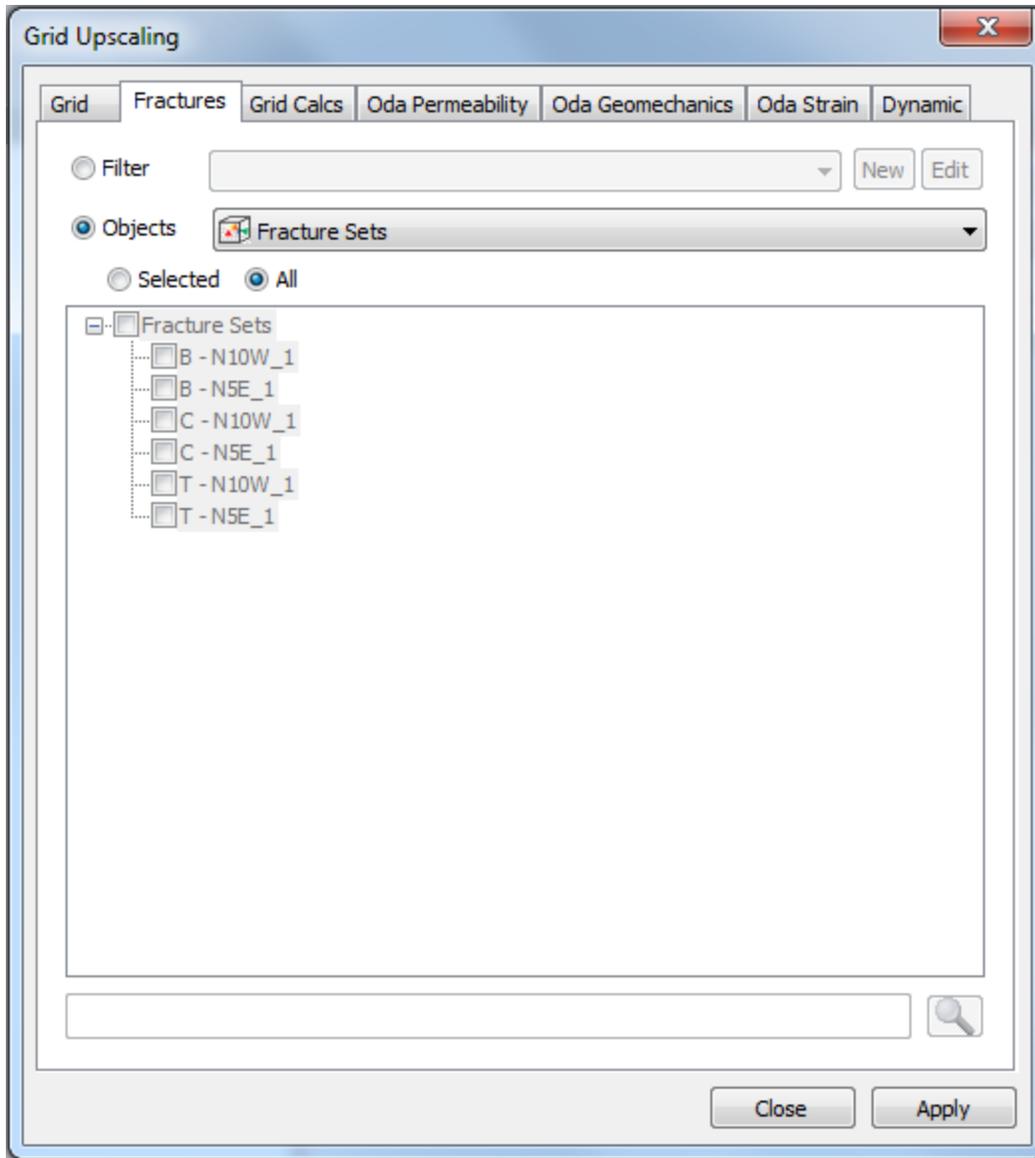
Fracture Upscaling

You can perform a number of analyses on grids to upscale fracture network properties to grid properties suitable for export to flow simulators. These upscaling analyses include the two Fracture K analyses, **Block K** and **Oda**; **Sigma Factor** analysis; and the **Fracture Count, P32, P33** analysis. (You can read more about what these analyses do under their respective sections in this manual.) Fracture Upscaling groups them all under one heading, allowing multiple analyses to be run at once. You can run Fracture Upscaling by selecting **Dynamic Analysis > Fracture Upscaling** from the menu bar or by right clicking on any grid in the Objects window and selecting **Analysis > Fracture Upscaling**.

On the **Grid tab**, select the grid you wish to upscale from the drop-down list next to **Grid**, which will include all grids in your file. The drop-down list next to **Grid Region** includes all the properties of the Grid. If one of these is defined so as to select a portion of the Grid, you may select it to restrict the analysis to this sub-region. Be careful, you can pick any property even if is nonsensical for defining a region. All non-zero values for a grid cell will mark it to be included in the analysis. The analysis definition will be saved in the Definitions window as the name you specify in the **Name** box.



On the **Fractures tab**, select which fracture sets will be included in the upscaling analysis. A filter may be used to select fractures as well.



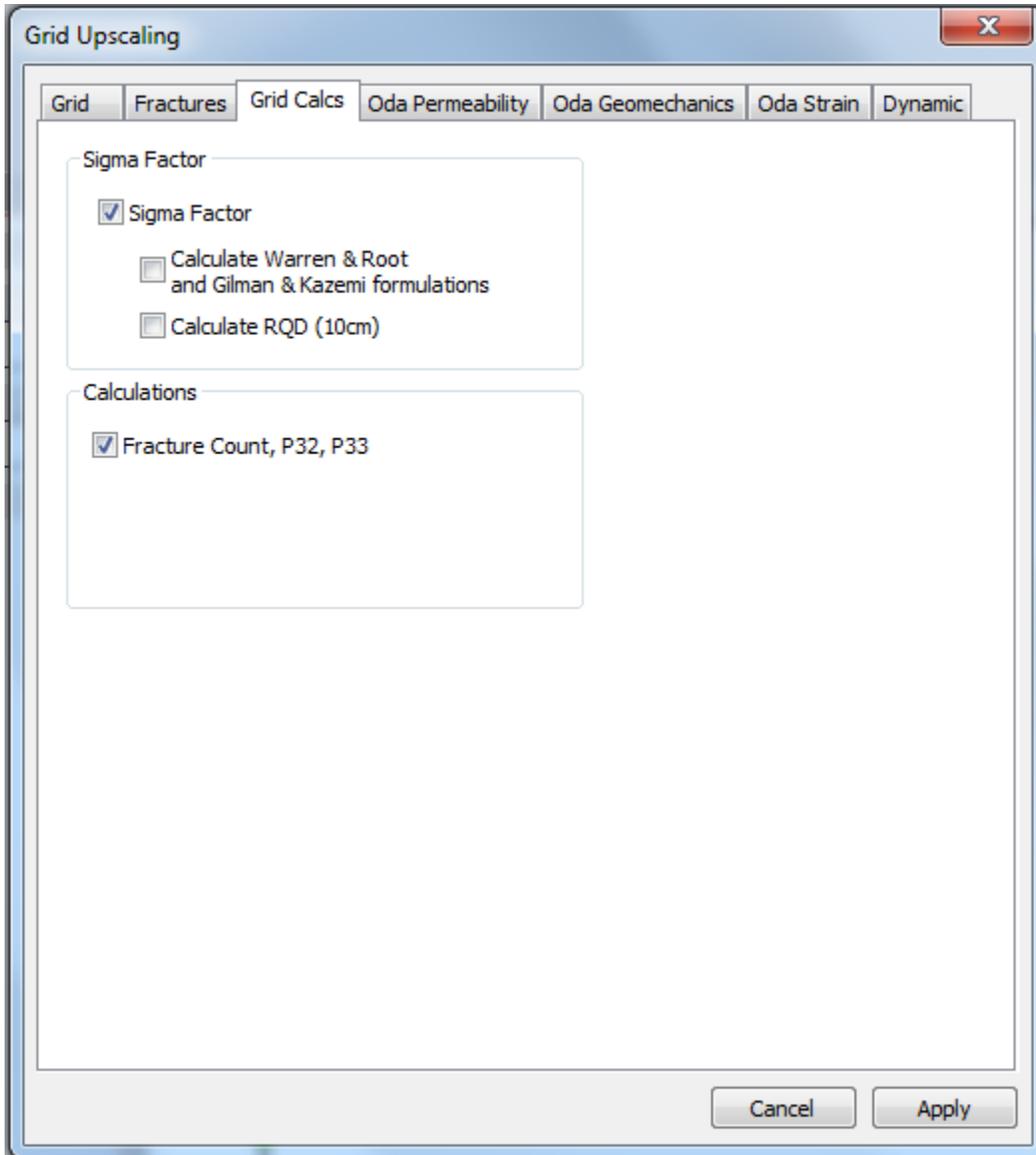
The remaining four tabs are used to select which analyses to perform on the selected grid and fractures. Select one or more analyses and provide any additional parameters as required. Click on **Apply** to save the definition and then either **Compute** to run the calculations now or **Close**. A new definition will be added under “Grid Fractures Analysis” in the Definitions window. To run the calculations later, right-click on the definition and select “Compute Grid Properties”.

Grid Calcs

Sigma Factor

This is essentially a measure of fracture/matrix flow dimension. The analysis calculates the fracture spacing along an imaginary borehole in the x-, y-, and z-directions, adds the reciprocals of the squares, and multiplies this by four. The result is the sigma factor. If the fracture spacing in one direction is very large relative to the other two, the sigma factor will be large; this is known as “matchstick” flow. If the fracture spacing is very small in one direction relative to the

other two, the sigma factor will be of middling size; this is “slab” flow. If fracture spacing is similar in all three directions, the sigma factor will be small; this is “block” flow. Sigma factors are used in Eclipse as a measure of fracture regime as well.



To run sigma factor, select **Static Analysis > Fracture Upscaling**, or right-click on the desired grid and select **Analysis > Fracture Upscaling**. In either case, select the Grid and Fractures on their respective tabs, and on the Grid Calcs tab check the box next to Sigma Factor.

When calculating the sigma factor you have the option of calculating additional shape factors “Warren and Root and Gilman and Kazemi” or “RQD”. Check the boxes next to these options to generate these values as well. Their definitions are provided in the discussion below.

As a general rule, fluid flow between small porous regions is proportional to a transmissibility that consists of a sum of terms that are each a surface area multiplied by a permeability and divided by a regional dimension perpendicular to the area. By multiplying the numerator and denominator of such terms by the regional dimension, the terms become volumes multiplied by permeabilities, which are divided by the squares of regional dimensions.

If the regions are now taken to be disconnected portions of the matrix porosity in a grid block, separated by thin fracture planes, the regional dimensions become the fracture spacings. Assuming that there are many similar such matrix regions connected to fractures that pervade the grid block, a total transmissibility for matrix to fracture flow within the block can be obtained by scaling up the disconnected matrix portion volumes to the total matrix porosity volume. The shape factor to be discussed here concerns what then happens to the other terms making up the total transmissibility.

$$\sigma = 4 * (1/L_i^{**2} + 1/L_j^{**2} + 1/L_k^{**2})$$

The **Warren and Root** formulation has matrix to fracture flow within a grid block proportional to:

$$W \ \& \ R \ \text{Transmissibility} = 20/3 * ((1/L_x + 1/L_y + 1/L_z)^{**2}) * \text{Cell_Volume}$$

the **Gilman and Kazemi** formulation uses:

$$G \ \& \ K \ \text{Transmissibility} = 4 * (1/(L_x^{**2}) + 1/(L_y^{**2}) + 1/(L_z^{**2})) * \text{Cell_Volume}$$

The above assumes that fracture planes exist in all three directions with fracture spacings L_x , L_y , L_z .

RQD (10 cm): The Rock Quality Designation (RQD) for the simulated core. RQD is defined as the total length of fracture spacings of length over 10 cm divided by the imaginary borehole length.

When the analysis is complete, the grid will have four new properties besides the sigma factor itself: the sigma block volume, or the average volume of a matrix block within the grid cell, and sigma lengths i, j, and k. The sigma lengths are the dimensions of an average matrix block in the given coordinate reference directions. The Sigma Block Volume is just the product of the three fracture spacings $L_x * L_y * L_z$.

Fracture Count and Volumetric Intensity

This calculates and creates new properties for the selected grid based off of the fractures contained therein. When run, four new properties will appear: `FRAC_COUNT`, `FRAC_PORO`, `P32`, and `R_MTX`.

- Fracture count is simply the number of fractures that are contained, partially or entirely, in each cell.
- Fracture porosity, or P33, is the total fracture volume (Area * Aperture) divided by the cell volume.

- P32, also known as volumetric intensity, is the number of fractures per unit volume (fracture area divided by cell volume) – a sort of normalized fracture count for each cell.
- R_MTX, or the equivalent matrix block radius of the cell, reflects what portion of the cell block is drained by each fracture. The volume of the cell is divided equally among the fractures in the cell.

Oda Permeability

Oda analysis sets out to calculate permeability and/or hydraulic conductivity tensors in three dimensions and all directions for each cell of a given grid. The Oda tensor represents a simplification of Darcy's Law for laminar flow through an isotropic porous medium. It works by projecting the fracture isotropic permeability (k) onto the plane of the fracture, then scaling it by the ratio between the fracture volume (from Porosity analysis) and the volume of the simulation grid cell. The result is a 3x3 matrix describing the directional permeability in each fracture; these matrices can be summed to give a net permeability tensor (k_{net}) for the grid cell. The principal permeabilities and flow directions (K_i , K_j , and K_k) can be found by eigenvector analysis of the net permeability tensor.

Oda's (1984) method begins by considering the orientation of fractures in a grid cell, expressed as a unit normal vector n . Integrating the fractures over all of the unit normals N , Oda obtained the mass moment of inertia of fracture normals distributed over a unit sphere:

$$N = \int_{\Omega/2} n_i n_j E(n) d\Omega$$

where:

N = number of fractures in Ω

n_i, n_j = the components of a unit normal to the fracture n

$E(n)$ = probability density function that describes the number of fractures whose unit vectors n are oriented within a small solid angle $d\Omega$.

Ω = entire solid angle corresponding to the surface of a unit sphere

For a specific grid cell with known fracture areas, A_k , and transmissivities, T_k , obtained from the DFN model, an empirical fracture tensor can be calculated by adding the individual fractures weighted by their area and transmissivity:

$$F_{ij} = \frac{1}{V} \sum_{k=1}^N A_k T_k n_{ik} n_{jk}$$

where:

F_{ij} = fracture tensor

V = grid cell volume

N = total number of fractures in grid cell

A_k = area of fracture k

T_k = transmissivity of fracture k

n_{ik}, n_{jk} = the components of a unit normal to the fracture k

Oda's permeability tensor is derived from F_{ij} by assuming that F_{ij} expresses fracture flow as a vector along the fracture's unit normal. Assuming that fractures are impermeable in a direction parallel to their unit normal, F_{ij} must be rotated into the planes of permeability

$$k_{ij} = \frac{1}{12} (F_{kk} \delta_{ij} - F_{ij})$$

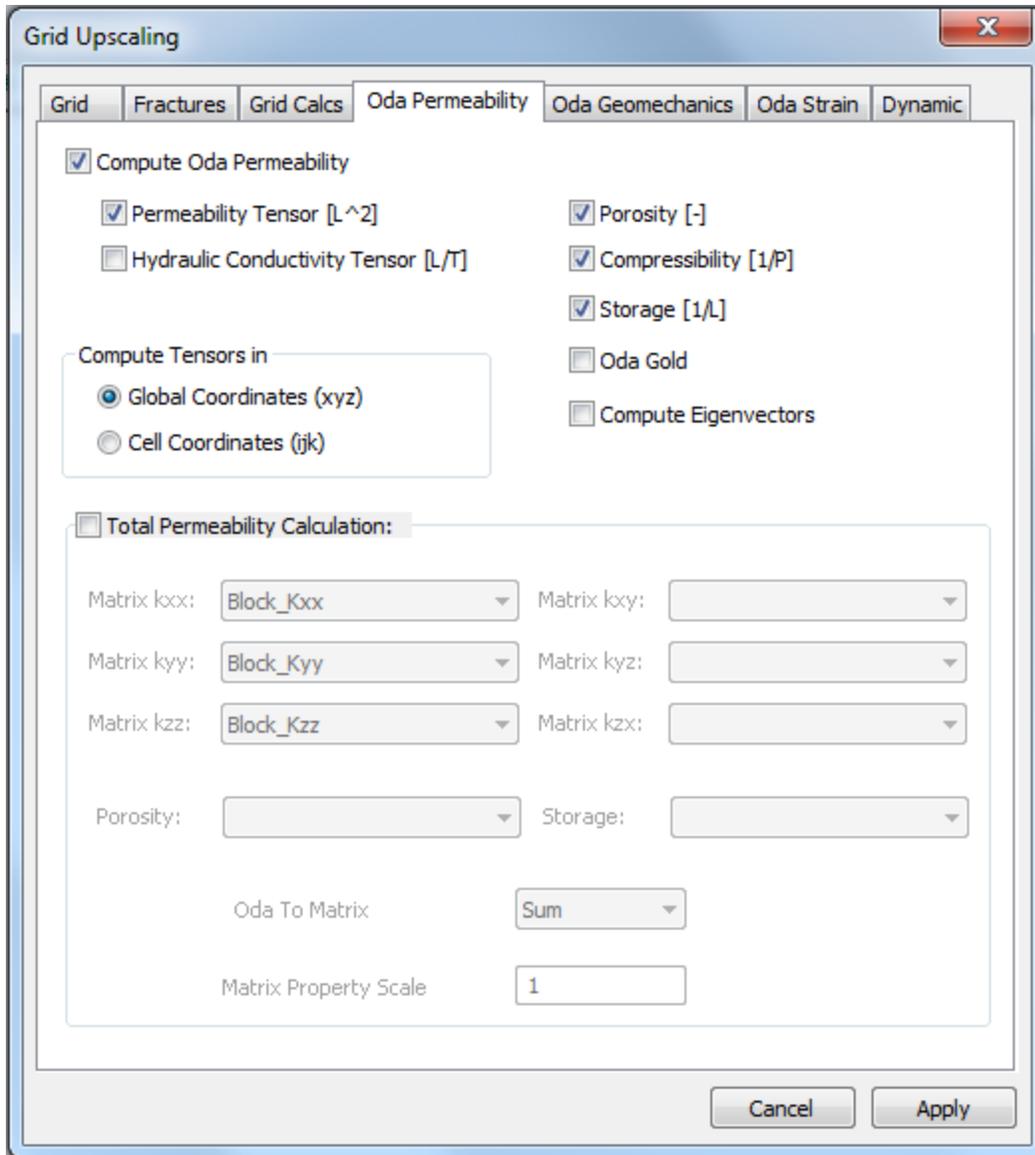
where:

k_{ij} = permeability tensor

F_{ij} = fracture tensor

δ_{ij} = Kroenecker's delta

Oda's solution has the advantage that it can be calculated without requiring flow simulations. However, it does not take fracture size and connectivity into account, and is therefore limited to well-connected fracture networks, and tends to over-predict block permeability compared to dynamic approaches.



Output...

- **Permeability Tensor [L²]** – Output will include grid properties for the permeability tensor components (Oda_Kxx, Oda_Kxy, Oda_Kyy, Oda_Kyz, Oda_Kzx, Oda_Kzz) in units of inverse length squared which is appropriate for PCA models. If the Oda output is in permeability and the model is using transmissivity for the fractures, then the Oda results will be sensitive to the fluid viscosity.
- **Hydraulic Conductivity Tensor [L/T]** – Output will include grid properties for the hydraulic conductivity tensor components (OdaHC_Kxx, OdaHC_Kxy, OdaHC_Kyy, OdaHC_Kyz, OdaHC_Kzx, OdaHC_Kzz) in units of length over time which is appropriate for TSA models.
- **Porosity** – Output includes the calculated grid cell porosity in the parameter named “P33”.

- **Compressibility [1/P]** – Output includes a grid cell based compressibility (“Compressibility”) having units of inverse pressure calculated by summing the compressibility and volume of fractures and portions of fractures in the cell, i.e. (fracture compressibility * area * aperture) / cell volume.
- **Storage [1/L]** – Output includes total storage of each grid cell (“Storage”) calculated by the sum of (fracture storativity * fracture area) / cell volume.

Compute Permeability Tensors in...

- **Global Coordinates (xyz)** – Tensors will be calculated in the x, y and z directions without regard for cell orientation
- **Cell Coordinates (ijk)** – Tensors will be calculated in the cell’s IJK directions (regular grids only)

ODA Gold - ODA Gold is an improvement to the existing Oda Upscaling calculation. This analysis performs a connectivity check on each grid cell and excludes fractures that are not part of a connected path. If you wish to use the analysis, just check the “Oda Gold” box in the Fracture Upscaling definition dialog.

Compute Eigenvectors This will calculate the eigenvectors of the Oda permeability tensor.

Total Permeability Calculation - Merge the Fracture permeability with imported matrix permeability. Select the matrix properties you wish to merge with along with optional **Porosity** and **Storage** components.

Oda to Matrix Interpolation - Select an interpolation method for combining the two permeabilities, SUM will add the two values together.

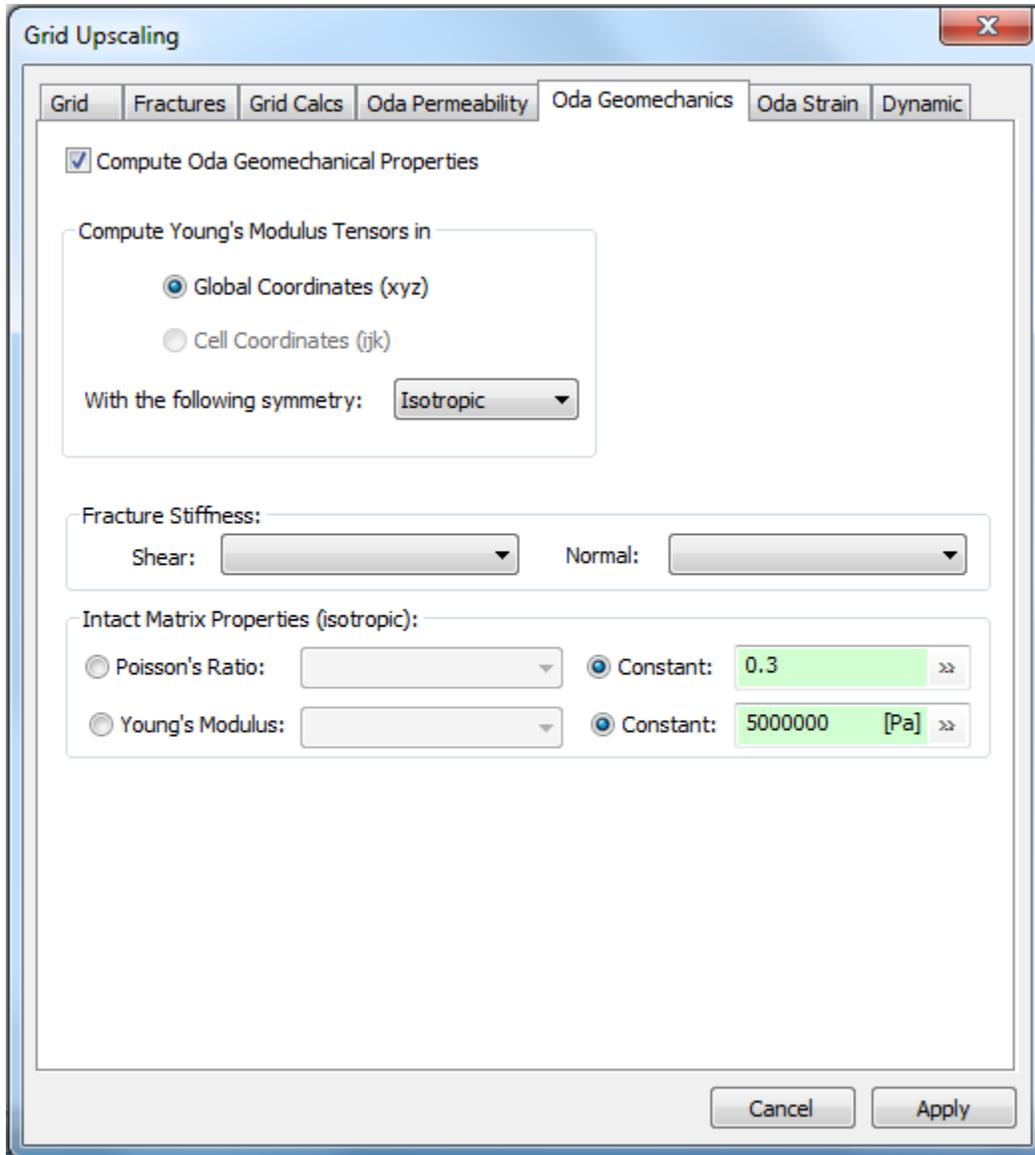
Matrix Property Scale Factor - this is a scale factor that is applied to the matrix property values. The ODA results are divided by this value before being combined with the matrix values.

Once you have chosen all your parameters, click **Apply** and **Calculate**. When the analysis is complete your grid will have six new properties representing the Oda permeability or hydraulic conductivity tensors in six directions (xx, xy, yy, yz, zx, and zz), as well as the other properties selected (Compressibility, P33, Storage). If you selected “Total Permeability Calculation”, there will also be new properties representing the total permeability in the same six directions, as well as a total storage property containing the storage of the entire grid.

Oda Geomechanics

The **Oda Geomechanics** tab sets up Oda calculations for the selected fracture sets. First the compliance tensor for the fractures is calculated and combined with the compliance tensor for the rock matrix. The resulting tensor is inverted to give the corresponding stiffness tensor for the combination of rock matrix and fracture sets.

The user can select the type of symmetry they wish to force on the resulting stiffness tensor, and this will determine the properties that are calculated from the stiffness tensor.



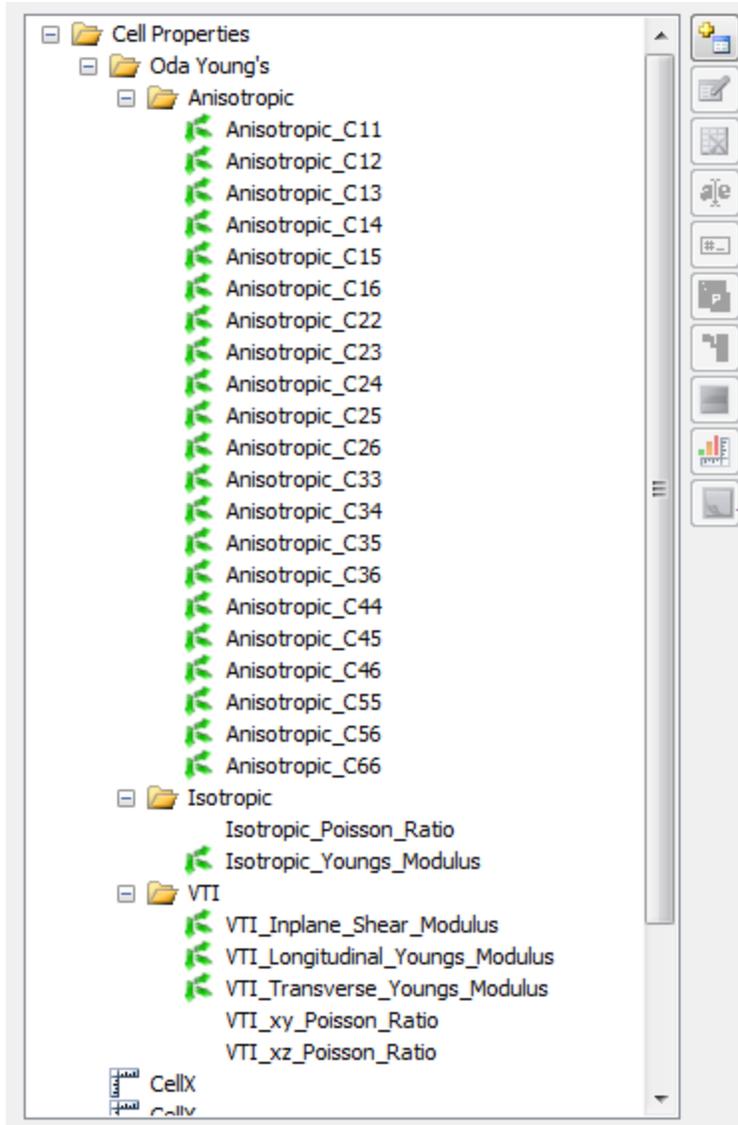
The following table shows the properties that are calculated for each symmetry:

Symmetry	Grid Properties Created
Isotropic	Isotropic_Poisson_Ratio Isotropic_Youngs_Modulus
VTI	VTI_Inplane_Shear_Modulus VTI_Longitudinal_Youngs_Modulus VTI_Transverse_Youngs_Modulus VTI_xy_Poisson_Ratio VTI_xz_Poisson_Ratio
Anisotropic	21 components of the Stiffness tensor: Anisotropic_C11, Anisotropic_C12 ... Anisotropic_C66

If the user selects 'Anisotropic' then the full 21 components of the stiffness tensor are output. For other symmetries, the stiffness tensor is projected onto the subspace defined by the symmetry. This gives the best approximation to the original stiffness tensor within the subspace. The higher the symmetry the fewer properties are required to uniquely define the approximation. Thus for the Isotropic case the output is just two properties: Poisson ratio and Young's modulus, whilst the VTI option creates five properties (based on equations A8 to A13 in 'Decomposition of the elastic tensor and geophysical applications' by Jules Thomas Browaeys and Sebastien Chevrot). In either case the full stiffness tensor for the given symmetry can be calculated from these properties. Note that the order of the stress and stiffness components follows the convention xx, yy, zz, xy, yz, zx which may differ from other systems. Thus for the projection of the Anisotropic components onto the VTI matrix, C55 should equal C66 (not C44).

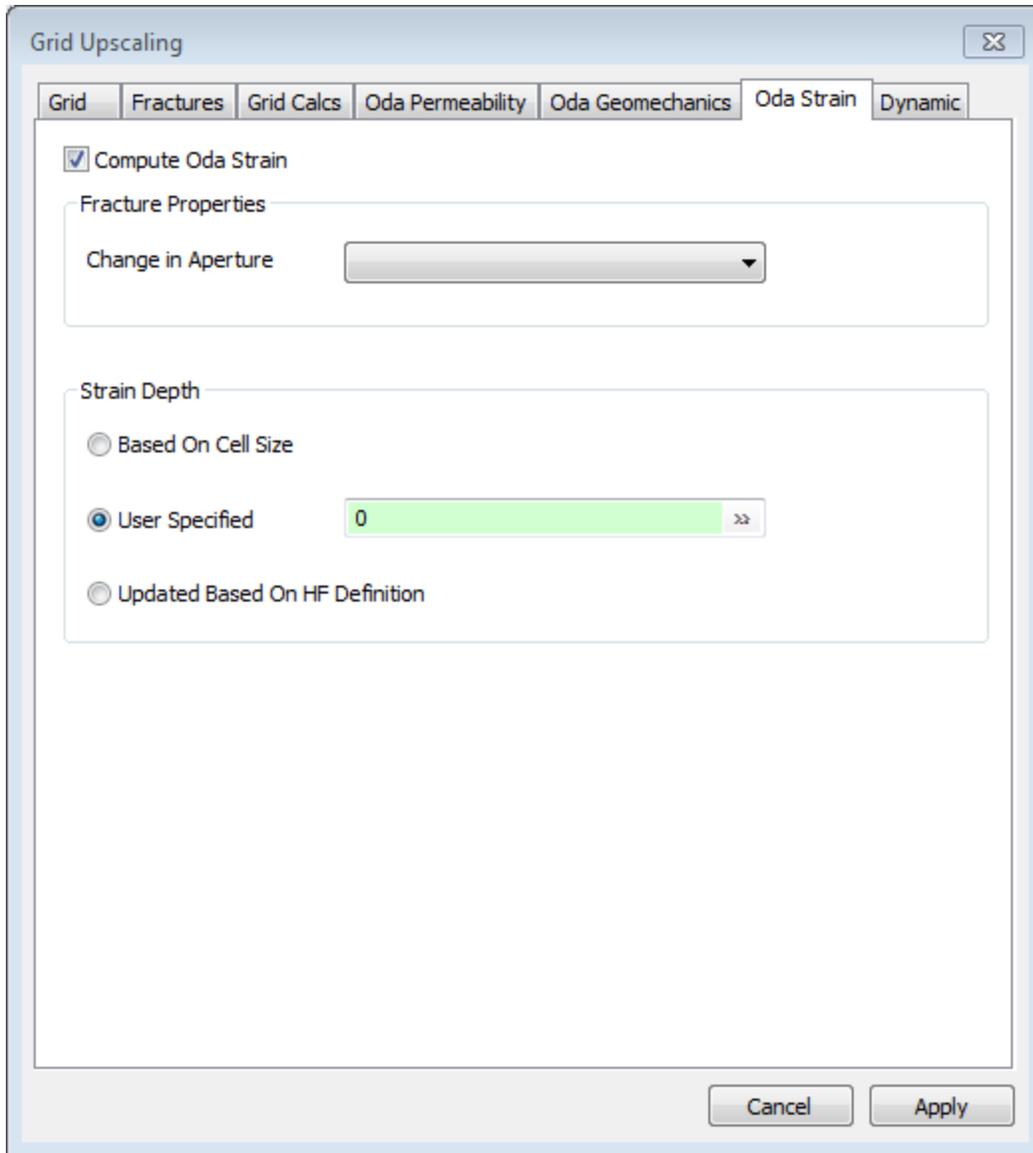
The user also specifies the shear and normal **Fracture Stiffness**, which are used to calculate the fracture contribution to the compliance tensor; and the matrix properties, **Poisson Ratio** and **Young's Modulus** which are used to calculate the matrix contribution to the compliance tensor. Note that the rock matrix itself is assumed to be isotropic.

All the properties are grouped in a folder with the name of the symmetry, and the property names also have the symmetry as a prefix. Thus it is possible to generate properties for more than one choice of symmetry if desired.



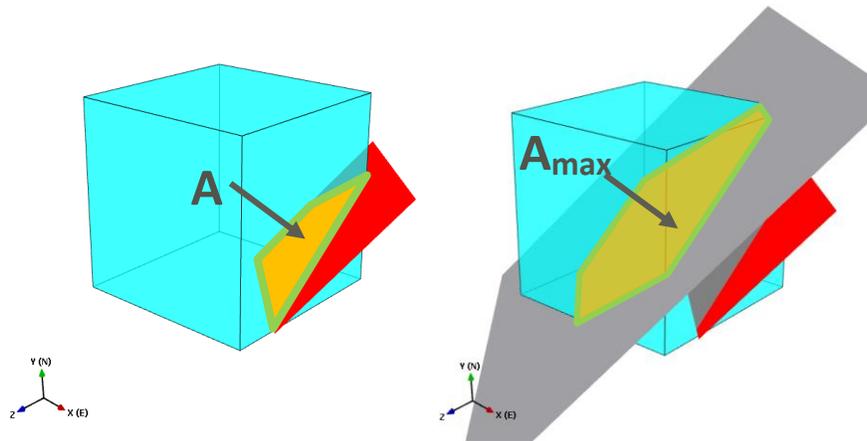
Oda Strain

This functionality is used for calculating rock strain induced by a change of fracture aperture. It is useful in conjunction with [FracStress](#) and [Hydraulic Fracturing Simulation](#). This feature computes the strain values due to change in fracture aperture during hydraulic fracture simulation. The **Change in Aperture** drop down menu reads the delta aperture (change in fracture volume) created after hydraulic fracture simulation. Then the strain values are calculated (normally based on the Cell Size unless user would like to specify another value depending on the situation).



Based on elastic 1D and Secor-Pollard theory

$$\epsilon_{z,loc} = \frac{dAperture}{L} \cdot \frac{A_f}{A_{max}}$$

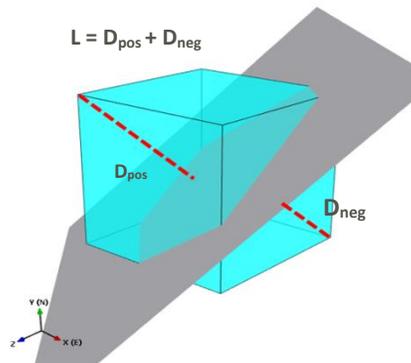


A_f represents area of fracture inside cell volume

A_{max} is calculated as area of a plane parallel to the intersecting fracture and placed at cell center

L represents a theoretical distance over which a strain is calculated. FracMan offers 3 different options of how to calculate the distance.

- A. Based on Cell Size: L is computed as a maximum distance between furthest cell vertex and a plane parallel to the intersecting fracture. L is defined as the sum of the maximal distance taken from the plane's positive and negative side.



- B. User specified: The user can specify a function for the strain depth, L .
- C. Updated based on HF definition: this is based on maximal change in fracture aperture, d_{Max} . This method is derived from combining 1D elastic theory and Secor-Pollard equation. The strain depth, L , is equal to $4 \cdot (1 - \nu^2) \cdot d_{Max}$.

■ 1D solution:

$$\begin{aligned}\sigma &= \varepsilon E \\ \sigma &= P_p - \sigma_n \\ \sigma_n &= \sigma_{n,init} + \varepsilon E \\ &\Downarrow \\ P_p - (\sigma_{n,init} + \varepsilon E) &= \varepsilon E \\ \varepsilon &= \frac{P_p - \sigma_{n,init}}{2E} \\ e = \varepsilon L &= \frac{P_p - \sigma_{n,init}}{2E} \cdot L\end{aligned}$$

■ Secor-Pollard equation:

$$e = \frac{1 - \nu^2}{E} (P_p - \sigma_{n,init}) 2 d_{max} \sqrt{1 - 4 \left(\frac{d}{d_{max}} \right)^2}$$

$$\frac{P_p - \sigma_{n,init}}{2E} \cdot L = \frac{1 - \nu^2}{E} (P_p - \sigma_{n,init}) 2 d_{max} \sqrt{1 - 4 \left(\frac{d}{d_{max}} \right)^2}$$

$$L = (1 - \nu^2) 4 d_{max} \sqrt{1 - 4 \left(\frac{d}{d_{max}} \right)^2} \quad d = 0$$

$$L = (1 - \nu^2) 4 d_{max}$$

Algorithm workflow -for each grid cell:

1. Identify intersecting fractures
2. Calculate area of the intersecting fracture inside the grid cell (A_f)
3. Calculate grid center plane area (A_{max})
4. Calculate strain distance (L)
5. Compute local strain and apply area weight ($\varepsilon_{z,loc}$)
6. Rotate strain tensor to global coordinate system (ε tensor)

Dynamic Upscaling

Rather than directly compute flow through a fracture network using finite element simulation, equivalent grid cell permeability tensors may be computed as parameters to a dual-continuum simulator by performing a “Block-K” calculation. In Block-K simulation, users select a range of cells from I, J, K cell indices, flow directions either All or just One direction, and the boundary conditions. For each grid cell FracMan assigns two kinds of boundary conditions onto the six faces of grid cell. One for two cell faces perpendicular to the flow direction, I, J, or K. The other is for the other four faces parallel to the flow direction.

The boundary condition perpendicular to flow direction is fixed as constant head conditions. The head value on one cell face is assigned as 0. The other end face is assigned to the length of flow path that is the cell dimension in the flow direction. This way a unit hydraulic gradient (dH/dx) is

created and maintained along the flow direction. The boundary condition for the four surrounding faces has three options. Users can select it from FracMan menu: Dynamic Analysis/Dynamic Upscaling/Dynamic/Boundary.

- **No-Flow** boundary: No-flow condition has a zero boundary flow rate and its boundary type is -1 ($Q=0$, Type = -1). This condition is assigned to every node on the block faces parallel to the applied hydraulic gradient direction.
- **Linear Pressure** boundary: This condition is a constant head condition (Type = 1). The head values vary between 0 and L, where L is the cell dimension in the flow direction. Depending on the boundary node location the head value is assigned to the flow path distance . i.e. Head = x, where x is the distance from the zero-head cell end.
- **Periodic Flow** boundary: This is a group no-flow boundary condition ($Q=0$, Type = -3). The four surrounding faces are equally divided into five sub-divisions along flow path. Each sub-division includes two sets of opposite faces. Each pair of opposite faces is assigned to one boundary group. The zero group flux rate is to simulate fluid entering from one face and exiting to the opposite face. Its net flow rate is null. In total there are ten group boundaries: Q_{11} , Q_{12} , ... Q_{15} and Q_{21} , Q_{22} , ... Q_{25} . When fewer fractures intersect at the boundary faces the sub-division may or may not contain any periodic boundary nodes. The flow condition approaches to no-flow boundary.

After simulation ten periodic boundary groups are mapped back to four cell faces. Flow rate across each cell dimension, I, J, or K, is then recalculated and reported from Block-K.

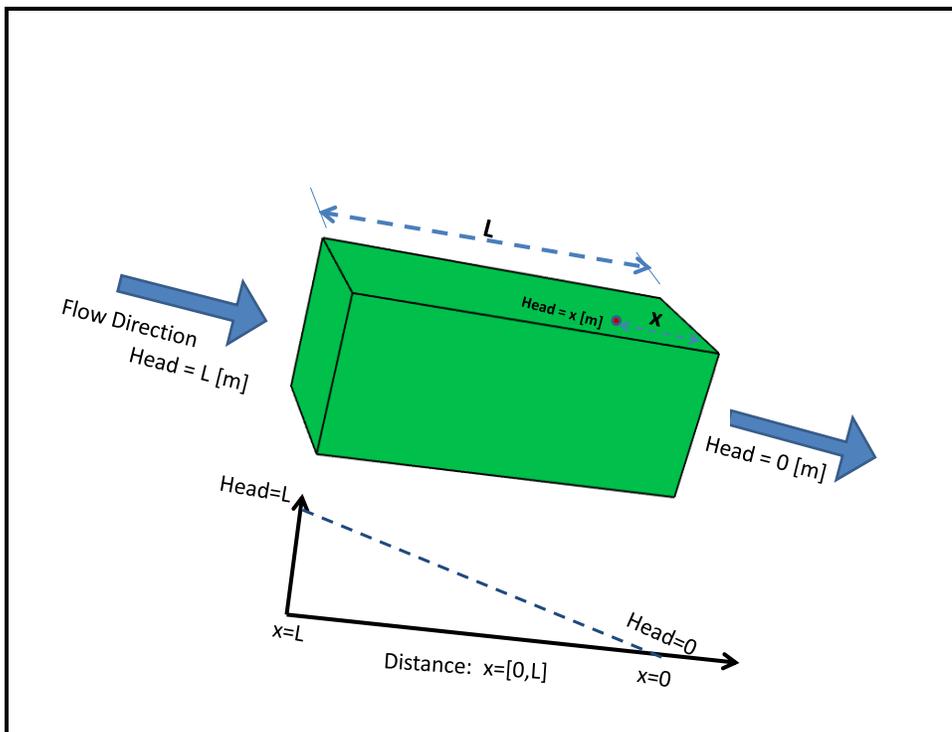


Figure 1: Linear Pressure Boundary

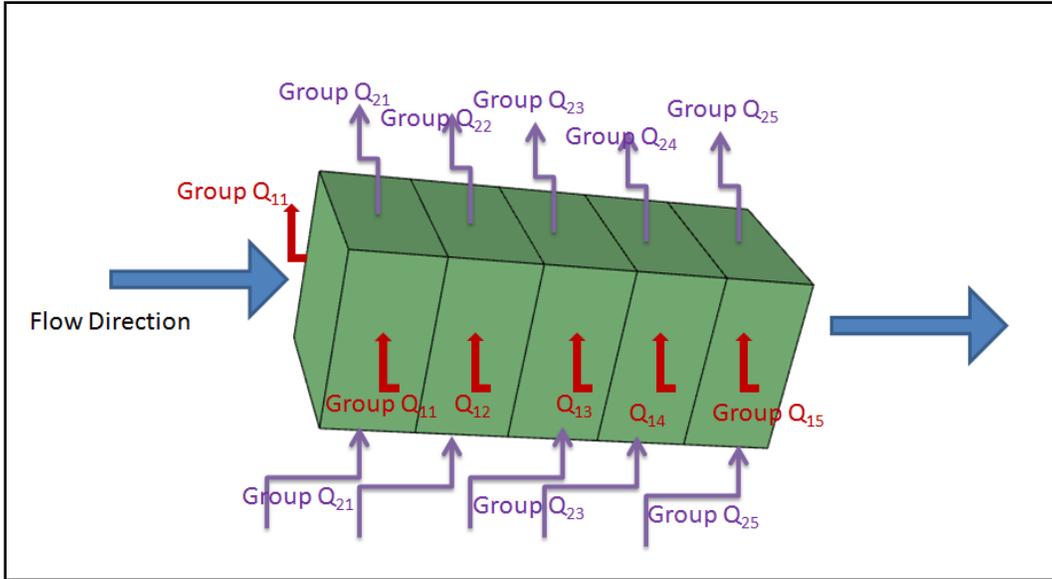
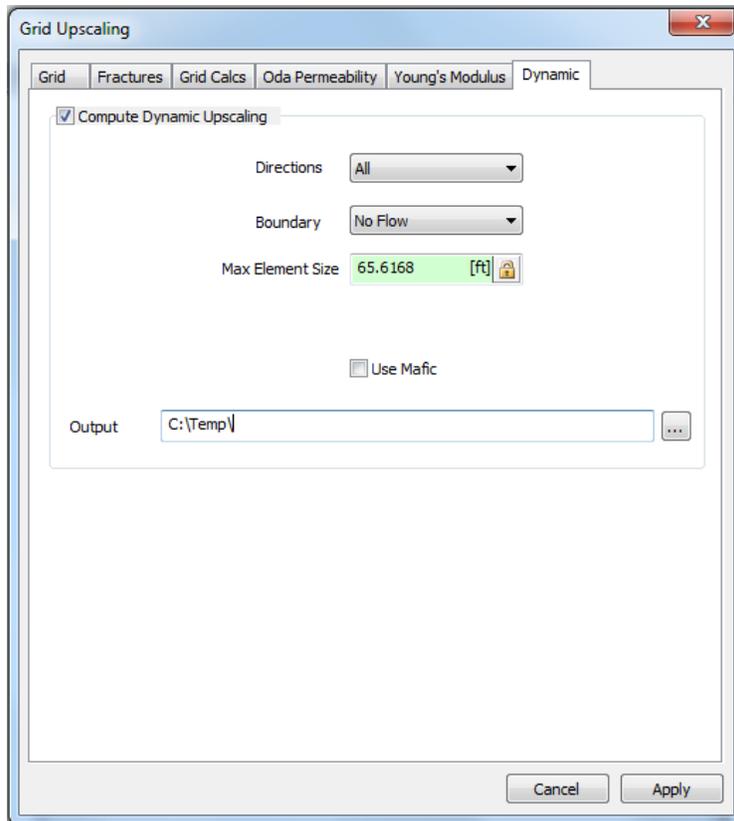


Figure 2: Periodic Flow Boundary Condition

The resulting cell fluxes are inverted to solve for the block hydraulic conductivities K_{xx} , K_{yy} , and K_{zz} . In order to compute block K values, a model must contain both a discrete fracture network and an axis-aligned grid which overlaps this fracture network.



Directions: the directions along which equivalent permeabilities are calculated in the local coordinate system of the cell.

Boundary: boundary condition applied to the sides of each grid cell.

- **No-Flow** - fluid is not permitted to exit the cell through the sides.
- **Linear Pressure** - pressure along the sides will vary linearly from the higher pressure at one end of the cell to the lower pressure at the other end of the cell (Figure 1). The “Linear Pressure” boundary condition will permit fluid to exit through the sides of the cell.
- **Periodic Flow** – a group no-flow boundary condition (Figure 2).

Max Element Size: maximum length in meters that an edge of any finite element in the mesh is allowed to have. *This size must be smaller than the size of the smallest grid cell; otherwise, elements will be too large and no “free” nodes will be produced by the meshing algorithm. This will stop the calculation because all finite elements directly connect to boundary faces.*

Output: the directory into which temporary and output files are written.

Once these calculations finish, the block K values will automatically be loaded into the property list of the selected grid.

The nature of the block K approach is that the results are very dependent on the assumed boundary conditions in the directions parallel to flow, and by the grid cell discretization. *Warning:* In cases of pinch out grids, one of the side areas is zero, and since area is part of the denominator of block K calculation, it fails in such cases. One walk-around approach to avoid this issue is to set all pinch out grids as inactive grids, which will not be calculated during the block K calculation process.

Note that block K results will show a correlation to cell height. This is determined by the formula of block K calculation, for example, if there is a constant pressure drop along the flow direction and no flow in any other directions, the k for the flow direction will be $k = q\mu L / (A\Delta p)$. Where A is the area perpendicular to the flow. Say this flow is in x direction, then k is k_x , and A is $dy * \text{CellHeight}$.

Computation can be very slow if there are many grid cells. Each block calculation has three parts: assign boundary conditions, generate meshes and compute flow rates. For each direction, the boundary assignment and flow calculation have to repeat so if you have 100x100x100 grid cells, it takes 10^6 MeshMaster runs and 3×10^6 MaficOil runs.

If there are many fractures but zero flow rate, then it is possible that the fracture network does not connect to both sides of the boundary faces.

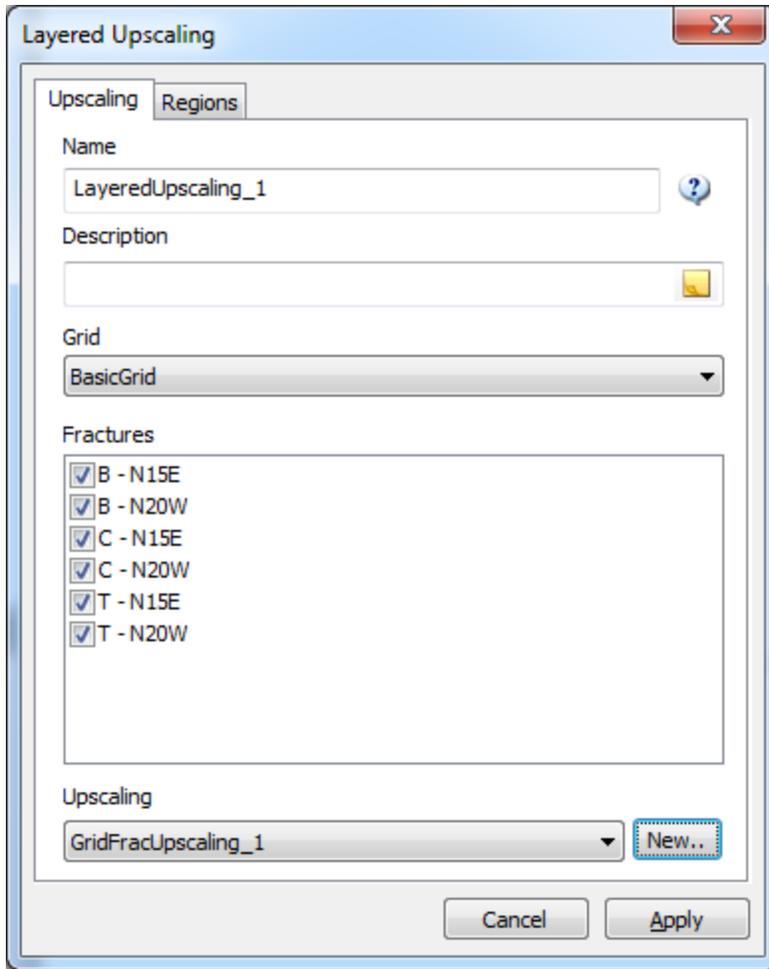
Like any other grid property, you can visualize the tensors in the Visualization window. Results from any block K calculation can be imported into FracMan for visualization and analysis, provided the grid geometry matches that for which the block K values were originally computed.

To import equivalent permeabilities directly, right-click on the grid name in the Objects window and select **Import Permeability**. Select a file of type `“*.block_k”` to import. The block K values will be placed into the property list of the grid object.

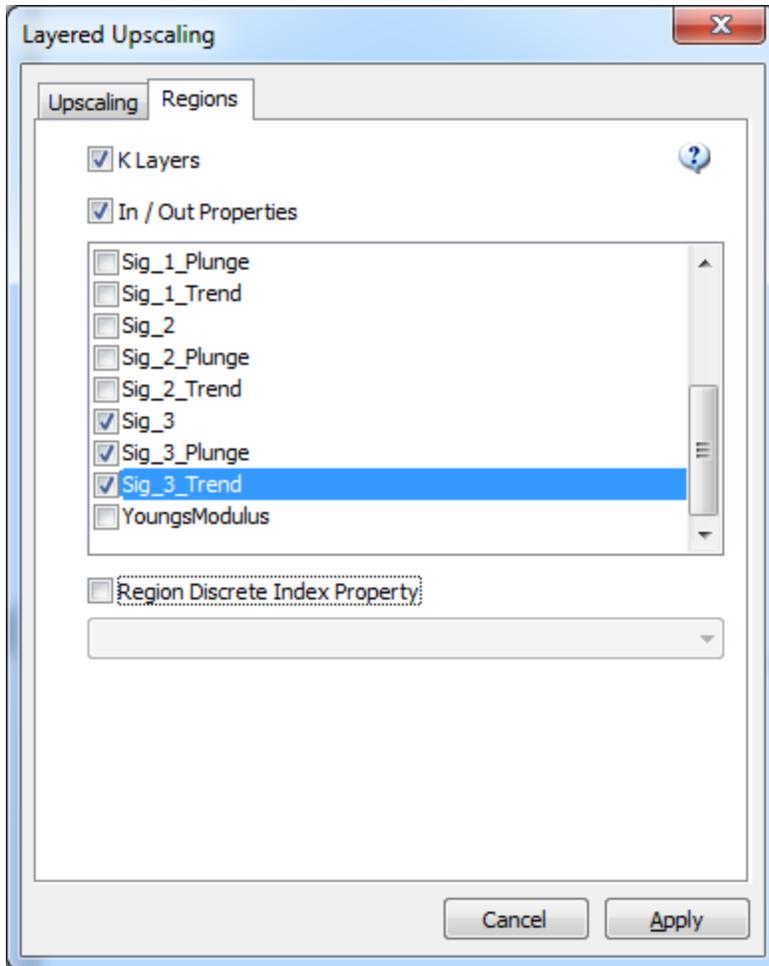
Layered Upscaling

FracMan allows the user the ability to generate and upscale a DFN layer by layer. The purpose of this feature is to create a layered system that is not well connected and represented in a standard finite difference grid (i.e. Eclipse). For example, for each fracture set definition, FracMan generates to 1 single grid layer, and then upscales that fracture set, and then moves to the next layer until all layers have a fracture set that has been generated and upscaled. The result is grid layers with upscaled fracture properties corresponding to the fracture set that were generated in that layer. Where, if the standard upscaling approach is used all fractures inside the grid will be upscaled at the same time, which will result in a grid representing a very connected layered fracture network.

First, the user specifies which Fracture sets and grid, along with a upscaling definition.



On the Regions tab, K layers, In/Out properties can also be included in the layer by layer upscaling process.



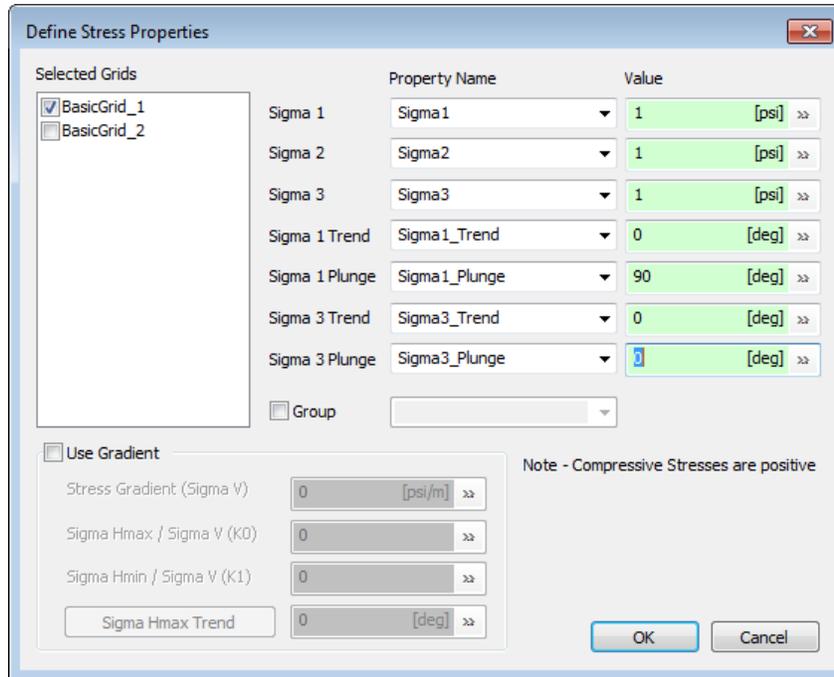
Intersecting Grid Layers

Grid Layer Property Report

This will report the mean/min/max/count of all properties of a grid, layer by layer. Select a grid object in the objects bar and select "Grid Layer Property Report" from the Static Analysis -> Grid menu. The report will exclude inactive and invisible cells. Right-click on the report to save to a file or copy to the clipboard for easy import into a spreadsheet.

Create Stress Properties

This simplifies the process of defining an initial stress field for use in Critical Stress Analysis or Hydraulic Fracture Simulation. You can enter values for Sigma1-3 and the directions of Sigma 1-3 directly or you can define these properties based on a stress gradient (SigmaV) and min/max horizontal stress components.



Compute Grid Properties

The following functions are described in the [Compute](#) section for Grids in Building Your Model:

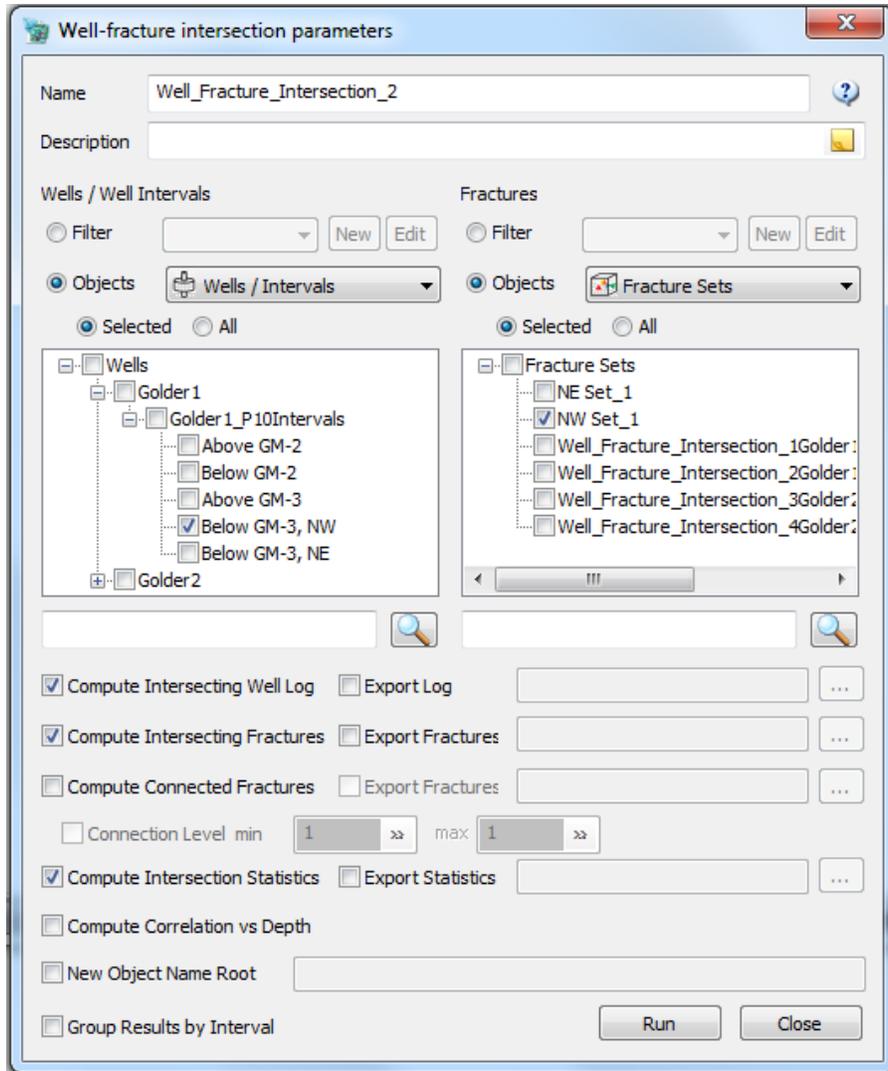
- Compute Grid Cell Volume
- Compute Grid Cell Height
- Compute Grid Cell IJ Dimensions
- Compute Grid Cell XY Area

Well-Based Analyses

These all require a well to proceed.

Intersecting/Connected Fractures

There are two possible ways to conduct the analysis. You may select from the file menu **Static Analysis > Well > Intersecting Fractures**, or you may right click on the name of a well in the Objects window and select **Analysis > Intersecting Fractures**. A dialog box will appear.



Name: Specify a customized name for the definition or accept the default. The new definition will be saved in the “Intersecting Fractures” folder in the Definitions window.

Definition: Optionally provide a description for the analysis.

Wells / Well Intervals: You may select one or more wells, or some specific interval(s) along the well(s). A search text box is provided below the well selection area in case you have large numbers of wells/well intervals in your project.

Fractures: If you wish only to use a certain subset of the fractures in your file to compute intersections, you may choose particular sets or define a fracture filter to narrow the selection. You may select from various categories of fracture sets and their associated reference sets as well as surfaces defined in your model. A search text box is provided below the fracture set selection area in case you have large numbers of sets/surfaces to choose from.

Select the results you are interested in using by selecting one or more of the following five options. The first four allow the specification of file names and directories for automatically saving the results external to the project file.

Compute Intersecting Well Log: The resulting calculated log will be saved as a new well log object having the well or well interval as a parent object and having “_Log” as part of the object name. If **Export Log** is checked, the well log can be saved to an .ors file once you specify the filename and directory.

Compute Intersecting Fractures: The resulting calculated fracture reference set will be saved as an object having the well or well interval as a parent object and having “_Isct” as part of the object name. If **Export Fractures** is checked, the intersecting fractures can be saved to a .fab file once you specify the filename and directory.

Compute Connected Fractures: When computing connected fractures you can optionally specify a connection range using the **Connection Level, min** and **max** parameters. Only fractures with a connection level in the specified range will be included in the result set. (Fracture intersecting the well have a connection level of 1). The resulting calculated fracture reference set will be saved as an object having the well or well interval as a parent object and having “_Conn” as part of the object name. If the **Export Fractures** option is checked next to this output selection, the new connected fracture set can be saved to a .fab file once you specify the filename and directory.

Compute Intersection Statistics: Statistics for the selected result objects including the intersecting well log, the intersecting fractures and the connected fractures will appear in a tab in the Visualization window once the analysis is run as well as being saved as a results object under Analysis in the Object window. The well log statistics appear below the fracture set statistics if both types of results have been checked. These results can be automatically saved to a .sts file during runtime if the **Export Statistics** option is checked and a file name and directory provided in the definition. They can also be saved manually from the tab results after the analysis has run by right-clicking in the report and selecting either **Save to File** or **Copy**.

Compute Correlation vs Depth: A plot showing correlation vs depth will appear in a tab in the Visualization window once the analysis is run as well as being saved as a results object under Analysis in the Object window.

New Object Name Root: The default result object name roots can be quite long as they include the definition name and the well or well interval name. The user can use this option to specify their own result object root name (the resulting objects will still have the identifying suffixes such as “_Conn” and “_Isct” plus a numeric extension to ensure uniqueness.)

Group Results by Interval: If checked, this option will make FracMan output a separate set of results for each selected interval; otherwise all intervals will be output together.

When you have completed your analysis definition, click **Apply** to save the definition and **Run** if you wish to compute the results right away. The analysis can be run at a later time by right-clicking on the definition name and selecting **Compute Intersections**.

When the analysis is complete, the new object(s) representing the new well log, intersecting fractures and/or connected fractures found by the analysis will appear in the Objects window as child objects of the well or well interval. Note: the intersecting fracture set and the connected fracture set are not independent objects. If any of the fractures from which they were derived are deleted, the reference fractures will also be deleted. You may change this by converting the reference set to a regular fracture set (see [Convert to Fracture Set](#)). The analysis statistics and the correlation vs depth plot (if checked in the definition) will appear in the Visualization window as result tabs.

Conditioning Statistics

After a discrete fracture network has been generated, FracMan can produce statistics on how closely the well fracture intersections match the P10 values specified in the well **Definitions Dialog** (accessible via the **Set P10** button in the **Attributes** window). The global statistics include the following information:

- Number of intervals selected, N
- Total sum length of all intervals, L , in meters
- Total RMS error, $E = \sqrt{\sum_{i=1}^N (c_i - t_i)^2}$, where c_i is the number of intersections over the i^{th} interval with the current fracture network, and t_i is the number of fracture intersections over the i^{th} interval computed from the target P₁₀ value specified in the Attributes window.
- Total bias, B , which is the total number of fracture intersections with the current fracture network minus the total number of fracture intersections as computed from the target P₁₀ values over all selected intervals. If this number is *positive*, then the target number of intersections over all intervals has been *exceeded*. If this number is *negative*, then the target number of intersections has not been met.
- Relative RMS error, E/L
- Relative Bias, B/L
- Average Bias, B/N

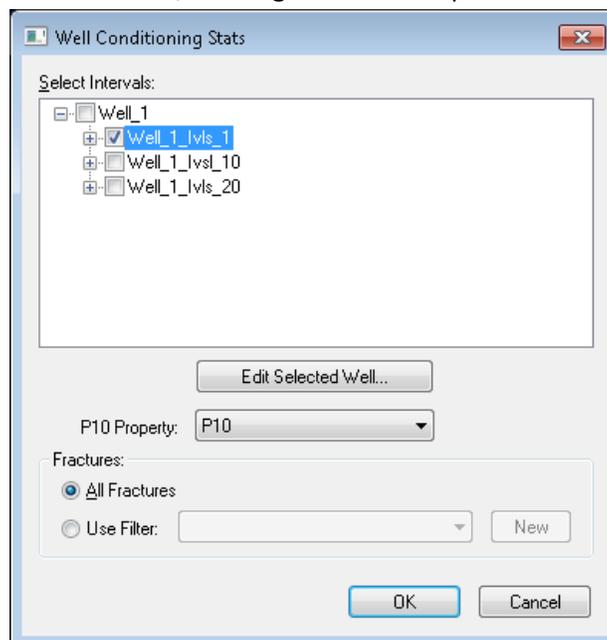
The statistics computed for each interval are:

- Interval length
- Target P₁₀
- Measured P₁₀

- Bias = $c_i - t_i$
- Relative Bias = Bias / L = $P_{10,measured} - P_{10,target}$
- Relative error squared, $e_i^2 = \frac{(c_i - t_i)^2}{L^2}$

To get well conditioning statistics, select **Static Analysis > Well > Conditioning Statistics**. In the dialog, select the P10 intervals over which to compute conditioning statistics, then click **OK**. The results will be displayed in a text window.

The conditioning statistics function has been improved to allow more control over what fractures are included and what P10 / well log data it is compared to.

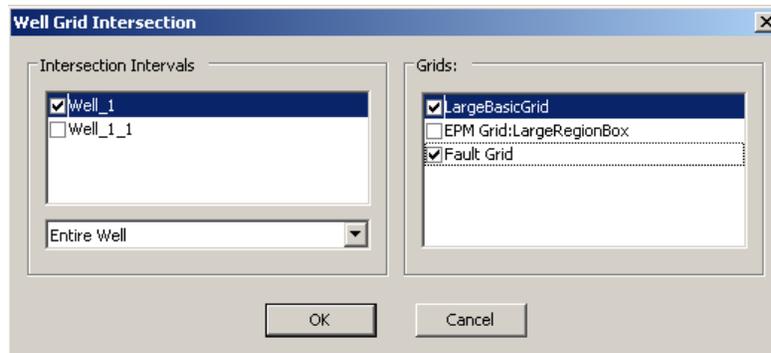


Intersecting Grid Cells

This computes what grid cells a well touches and gives a list of the grid property values in those cells. You may also initiate analysis by right-clicking on the name of any well in the object window and selecting **Analysis > Intersecting Grid Cells**. A dialog box will appear asking you to specify which wells and which grids you wish to compute intersections for. Check the boxes next to the ones you wish to analyse. You may also compute intersections of a grid with only selected perforation or P10 intervals rather than the entire well by selecting the desired interval from the drop-down menu in the lower left corner of the dialog. Then hit **OK**. FracMan will create a table listing the property values of the grid at each intersection point, as well as the grid name, well name, and location of the intersection. You may copy or save the table by right-clicking and selecting **Copy** or **Save to File**.

Correlate P10 to Grid

This allows the user to quickly discover any controls on fracture intensity, such as seismic or spatial properties, by comparing grid property values against a well's P10 intervals. You may access it from the file menu or by right-clicking on a well and selecting **Analysis > Correlate P10 to Grid**.

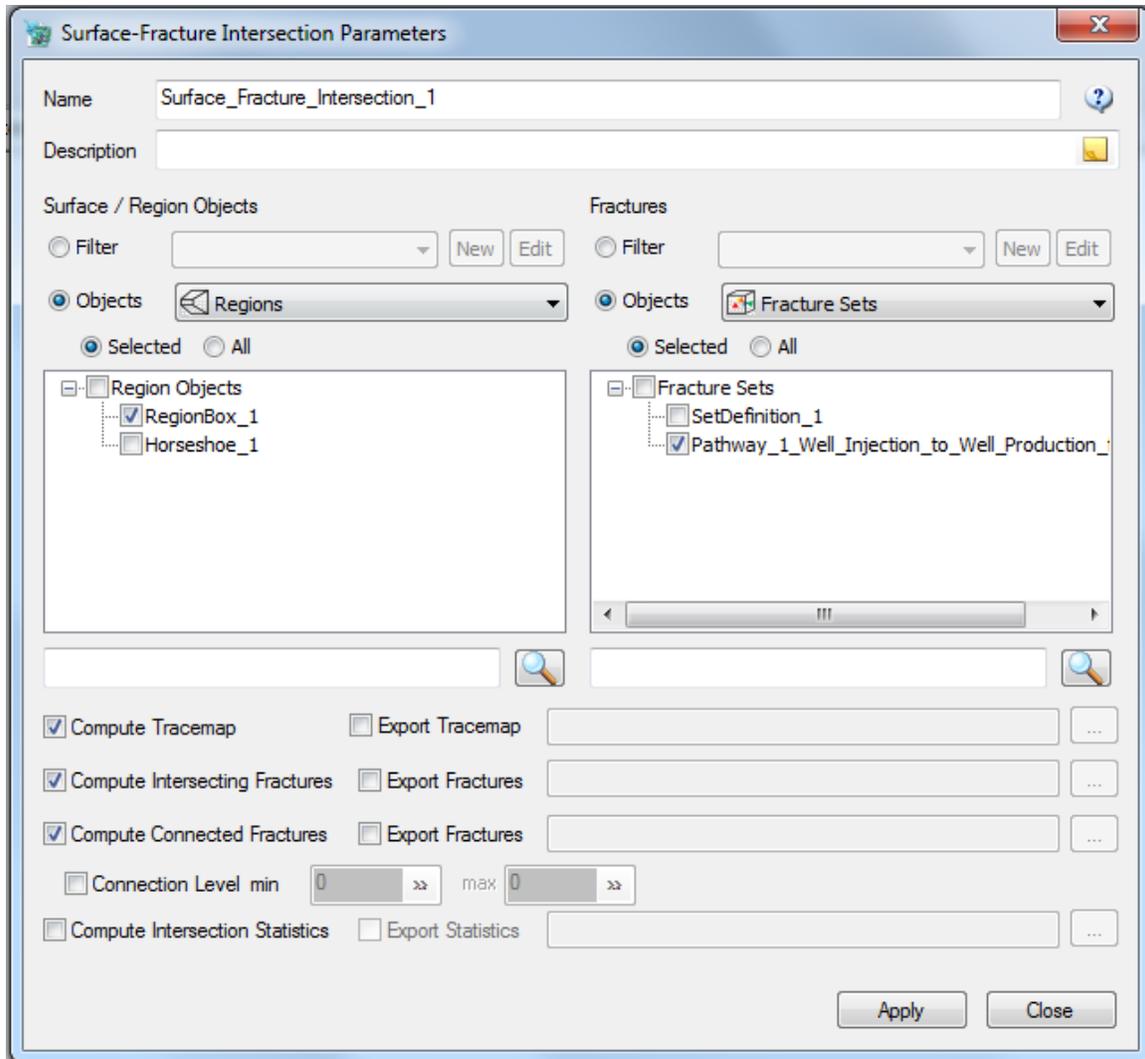


At the dialog box, select the wells or well intervals whose P10 values you wish to compare. Then select the grid or grids against which you intend to correlate your P10s. Hit **OK**. Another dialog box will appear, listing all available grid properties. Select the one you wish to measure against P10. Then click **OK**. If you have chosen multiple grids, be sure that the property you select is common to all of them or the analysis will not run properly. When the analysis is complete FracMan will produce a cross-plot showing P10 against the chosen property. You may copy and paste or save the graph by right-clicking on it and selecting the appropriate option; the graph also contains a fit line so that you may easily determine if and how the property and the P10 are related.

Surface-Based Analyses

Surface/Region Intersecting Fractures

There are two possible ways to conduct the analysis. You may select from the file menu **Static Analysis > Surface > Intersecting Fractures**. You may also right click on the name of a surface or a region in the Objects window and select **Intersecting Fractures**.



A default definition **Name** is provided and may be changed. Choose the region(s) or surface(s) for which you wish to compute intersections. Specify the fractures to be included in the analysis explicitly using **Objects** or by using a **Filter**.

You can also specify the outputs you want: Tracemap, Intersecting Fractures, Connected Fractures or Intersection Statistics. These may also be exported as files.

Click **Apply** when you are done. When the analysis is complete, the specified regions/surfaces will each have child objects corresponding to the selections you made.

Note: the Intersecting / Connected Fractures sets are not independent objects; if any of the fracture sets from which it was derived are deleted, the “reference sets” will be deleted as well. If you wish to make the set independent of the original fracture sets, you must convert it to a fracture set (see Manual section [Convert to Fracture Set](#)).

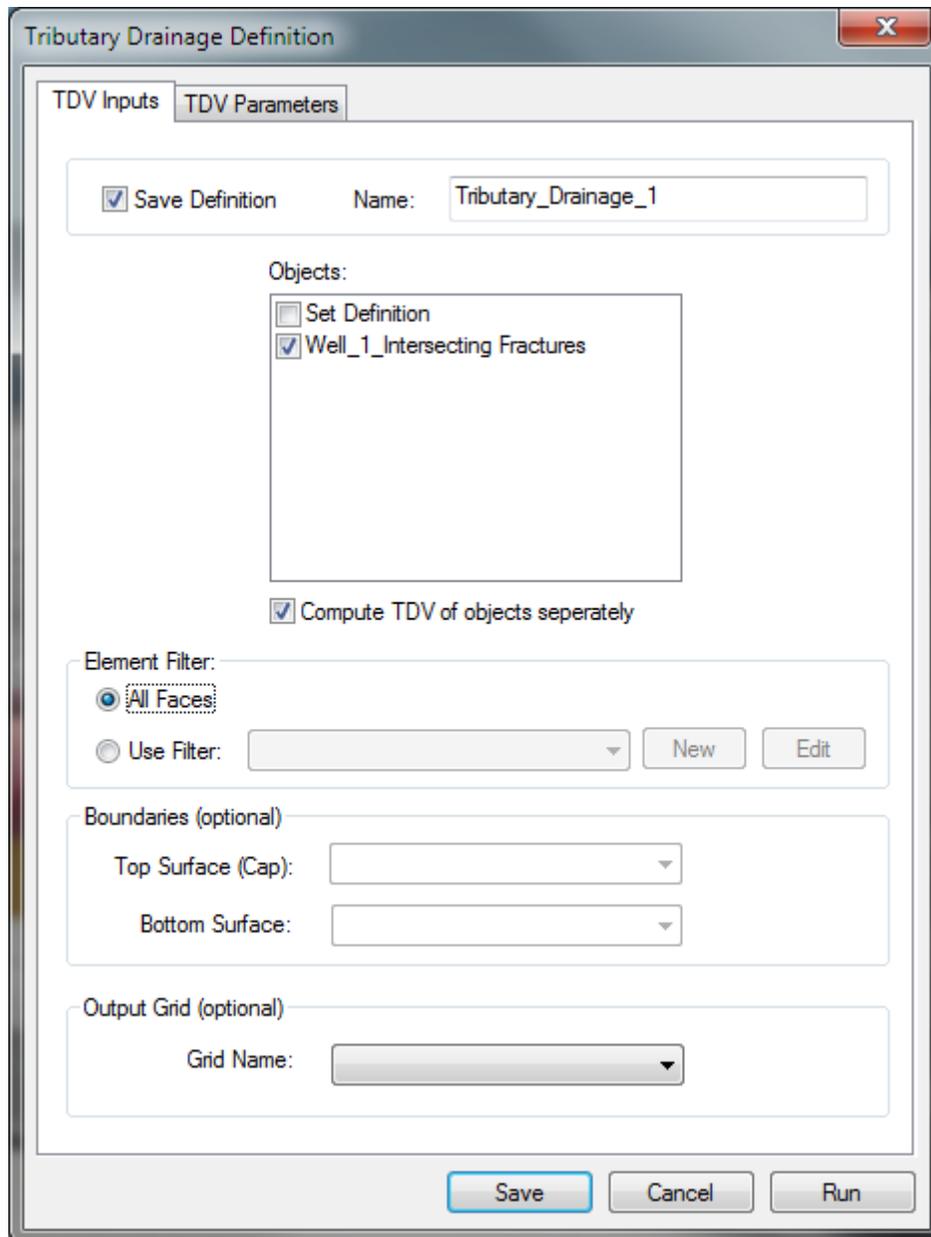
Depending on the geometry of your intersection object it is possible to have a larger number of traces than intersecting fractures. This is caused when a fracture intersects two (or more) unconnected faces (such as two sides of a tunnel) – each section will be a separate trace.

Volumetric-Based Analyses

Tributary Drainage Volume

In a fractured reservoir, a well produces from the fracture network intersecting the well and from the matrix that feeds into that portion of the fracture system. As a result, production from a well is limited to the portion of the fracture network to which it is connected. This portion is called a tributary drainage volume and FracMan can be used to calculate it for a well.

To compute the tributary drainage volume for a well, you must first have run the Connected Fractures analysis on the well you wish to test and have an existing Connected Fractures object for that well. Tributary drainage can also be computed on connected fractures, clusters or any set of fractures created from a fracture filter. Then under the file menu select **Static Analysis > Volumetric > Tributary Drainage Volume**. The following dialog box will appear:

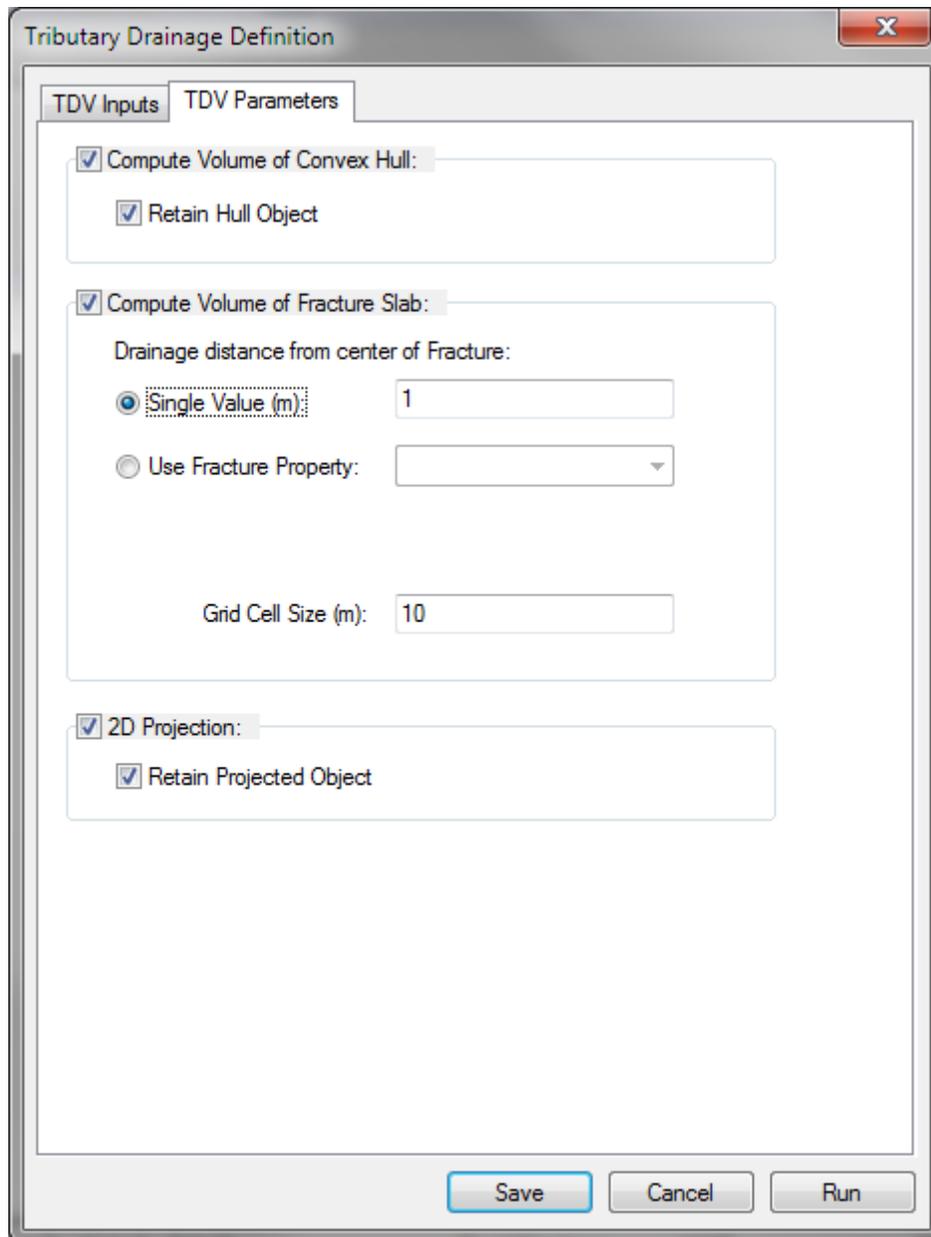


Select the sets of intersecting fractures, connected fractures, or clusters you wish to analyze.

Element Filter: You can optionally apply a filter to exclude fracture faces based on property values.

Boundaries: You can specify top and bottom surfaces to represent oil/water barriers – the fractures will be clipped to these surfaces before being considered for the volume computation.

Output Grid: You can specify a grid which will be updated with the TDV of fractures in each cell.

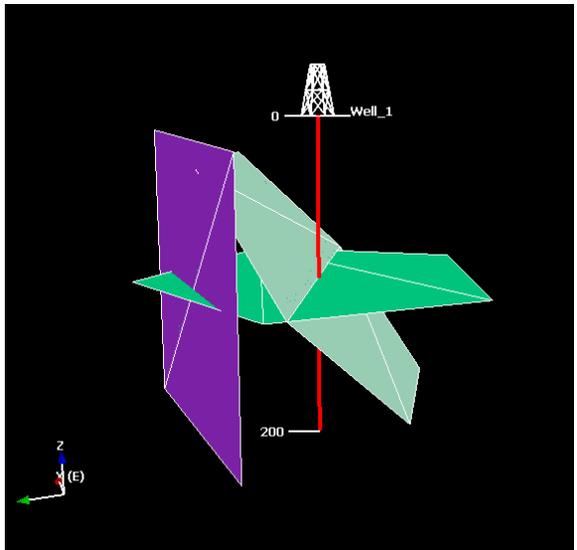


On the next tab there are two different ways of calculating the tributary drainage volume. The first “Compute Convex Hull” creates a region around the user specified fracture set. The output of this operation is Fracture Count, Fracture Area, Convex Hull Area, Convex Hull Volume, and Convex Hull P32.

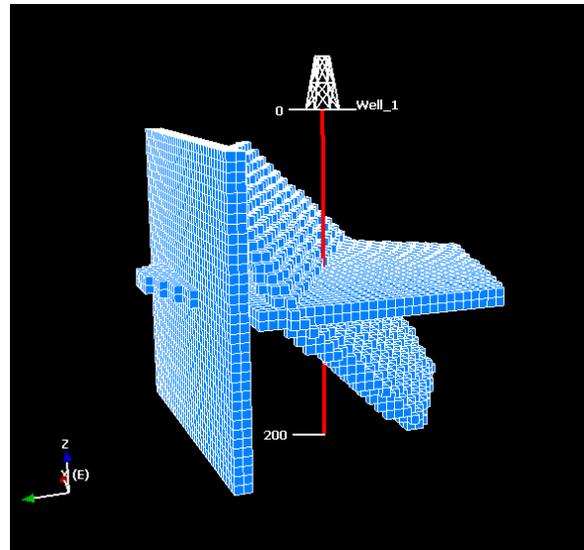
The second option is the “Use Fracture Thickness” function. This will create a grid of the fractures. If the center of a grid cell is inside the fracture it will become active. For this you can define a constant thickness for the entire fracture set or choose from among the pre-defined fracture properties. It is important to set the grid cell size equal to or less than the assigned fracture thickness. Both functions can be run simultaneously. *Note:* the “Compute Convex Hull”

function assumes a greater matrix contribution than the “Use Fracture Thickness” function. If your model is reasonably small, a new grid object built out of filled cells will be created. Since this grid quickly gets too large to store, larger models will have these values accumulated in the statistics and stored in the corresponding cells of the “output” grid (if provided).

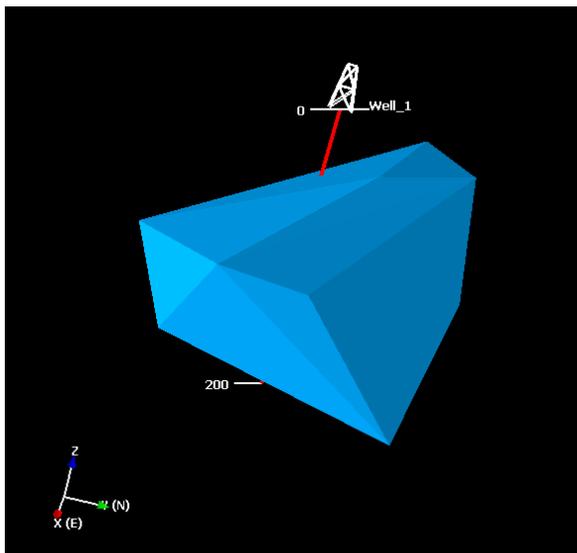
Below are visualizations of a connected fracture set and the various tributary drainage volume calculations of the set.



Initial Fracture Set



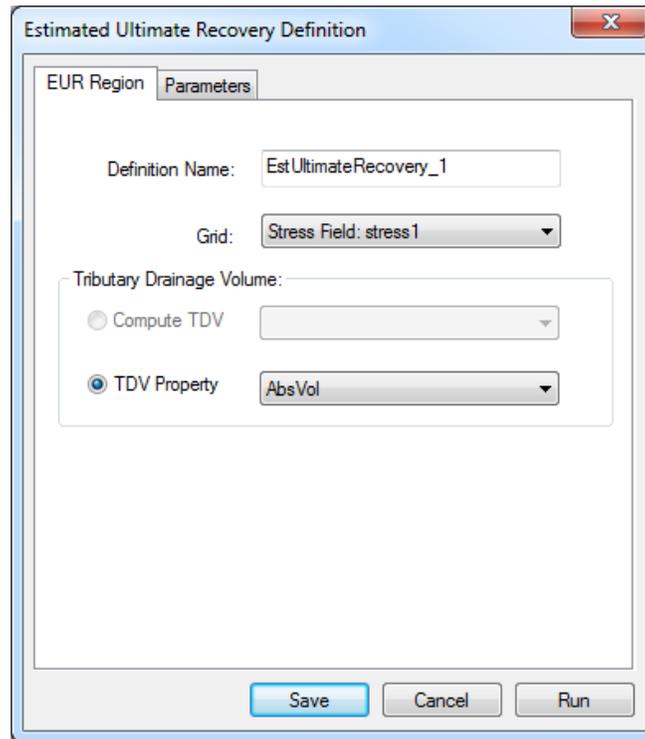
“Use Fracture Thickness” function



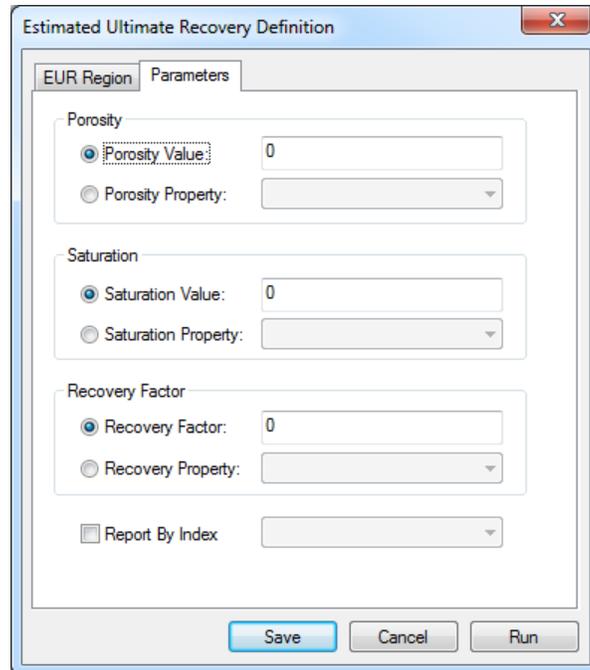
“Compute Convex Hull” function

Estimated Ultimate Recovery

Estimated Ultimate Recovery, or EUR, is the amount of recoverable oil one predicts to be contained in a given volume of fractures. Before you can run an EUR analysis, you must first either run a TDV analysis or upload a TDV property into the grid region for which you wish to calculate the EUR. To run the analysis, click *Static Analysis > Volumetric > Estimate Ultimate Recovery...* from the menu bar.



Here you must name your EUR Analysis definition, specifying what the definition will be saved as in the Definitions window. Then specify the grid region whose EUR you wish to calculate. If you have a saved TDV Analysis definition in your FracMan file, you may run the TDV analysis now by selecting “Compute TDV” and choosing the desired definition from the drop-down menu; if you have already have a TDV property in your grid – whether imported or from an earlier analysis – you can select “TDV Property” and choose the appropriate property from the drop-down menu. The program will then use this property for the analysis rather than re-running TDV calculation.



The second tab of the definition dialog allows you to specify which values or properties FracMan should use for fracture porosity, oil saturation value, and recovery factor. You may use a grid property for these, or you may specify a single constant value. When you have defined your analysis, click **Save** to save the definition or **Run** to save and run the analysis.

$$\text{EUR} = \text{TDV} * \text{Porosity} * \text{Saturation} * \text{Recovery Factor}$$

When the analysis is complete, it will give a short printout of results, telling the name of the definition, the name of the grid used, and the mean value of your EUR property. Your grid will also have a new property called EUR.

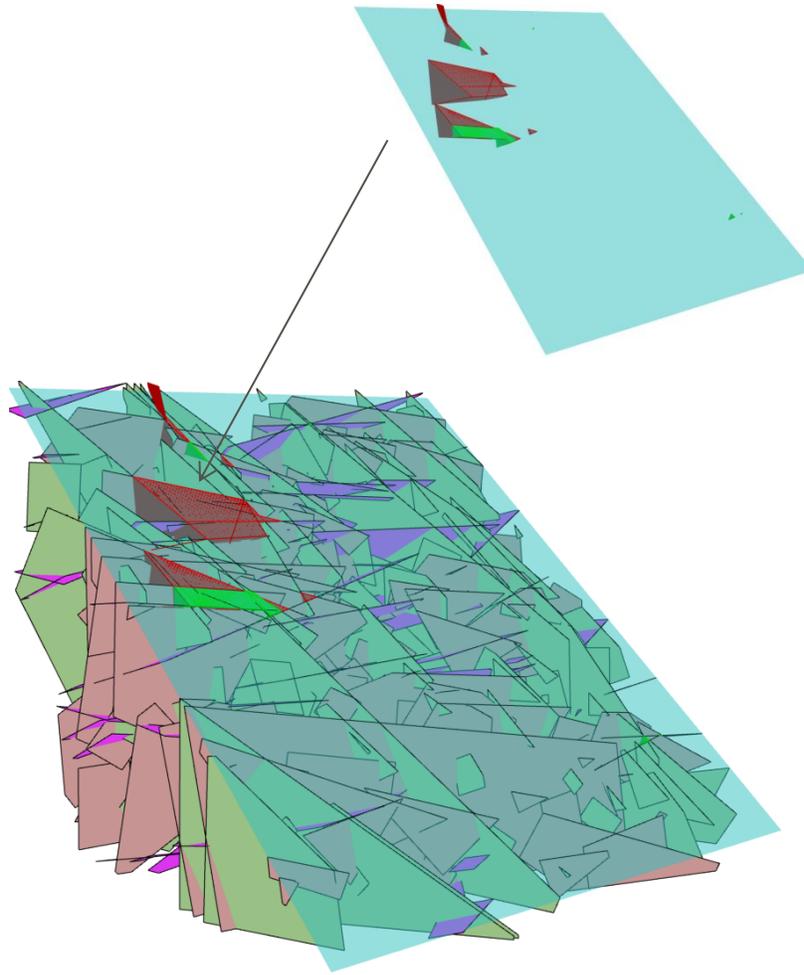
Geomechanics

Rock Wedge

The Rock Wedge analysis in FracMan simulates block stability on slopes or mine and roads tunnel, allowing a truly probabilistic assessment of the kinematic stability in surface and underground rock excavations. This analysis is carried out upon realistic rock blocks defined by the user's DFN model and therefore not subjected to any of the conventional limitations of block stability being defined by a prescribed number of fracture sets. Therefore, this approach considers the probability of adverse wedge formation rather than just the possibility.

To conduct such an analysis, your file must first contain either a surface (the simpler the surface the better and **ideally defined to quadrilateral rather than triangular elements**), representing a slope, or underground excavation.

If the tunnel/surface does not intersect any modeled fractures, the analysis will run but the results will be meaningless.

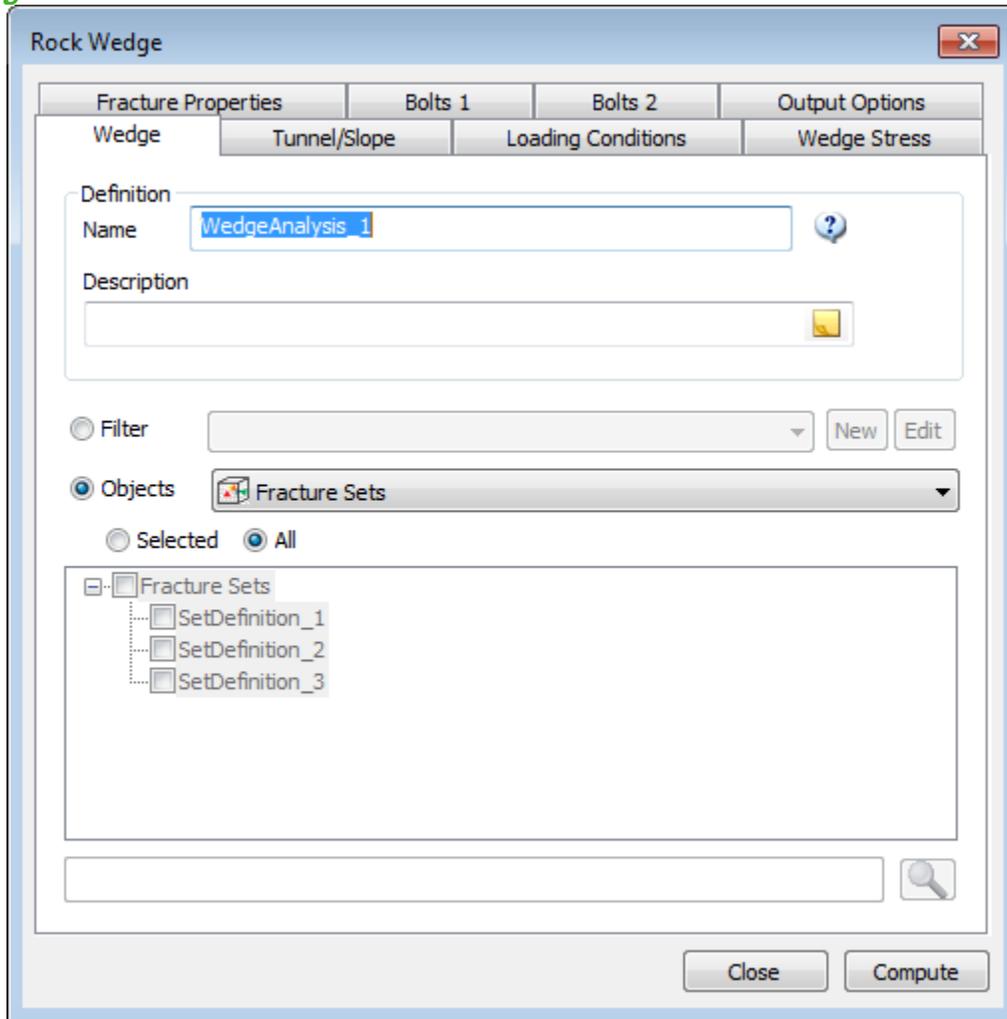


It is recommended that the complexity of the DFN model is minimized as a highly detailed model will take a considerable time to run owing to the very high number of potential search pathways within the fracture network. Consideration should therefore be given to reducing the number of connections to the modeled slope/tunnel by either using a smaller model or through removing the smaller fractures from the DFN.

Tessellated fractures and T-truncated fractures also do not work with the wedge analysis. If you have tessellated fractures, run **Fracture | Edit | Tessellate/Untessellate Fractures** to convert to polygon fractures before running the wedge analysis. If you have fractures terminating on surfaces or on other fractures, it is recommended that you increase their size by a small percentage to ensure that they penetrate those surfaces (**Fracture | Edit Scale Fractures**).

Once your DFN model is set up, you can access the analysis by selecting Static Analysis > Geomechanics > Rock Wedge Analysis... from the top menu. The user interface to the analysis definition dialog is described below. Refer to the [Rock Wedge Theory](#) section at the end for details regarding the computations.

Wedge

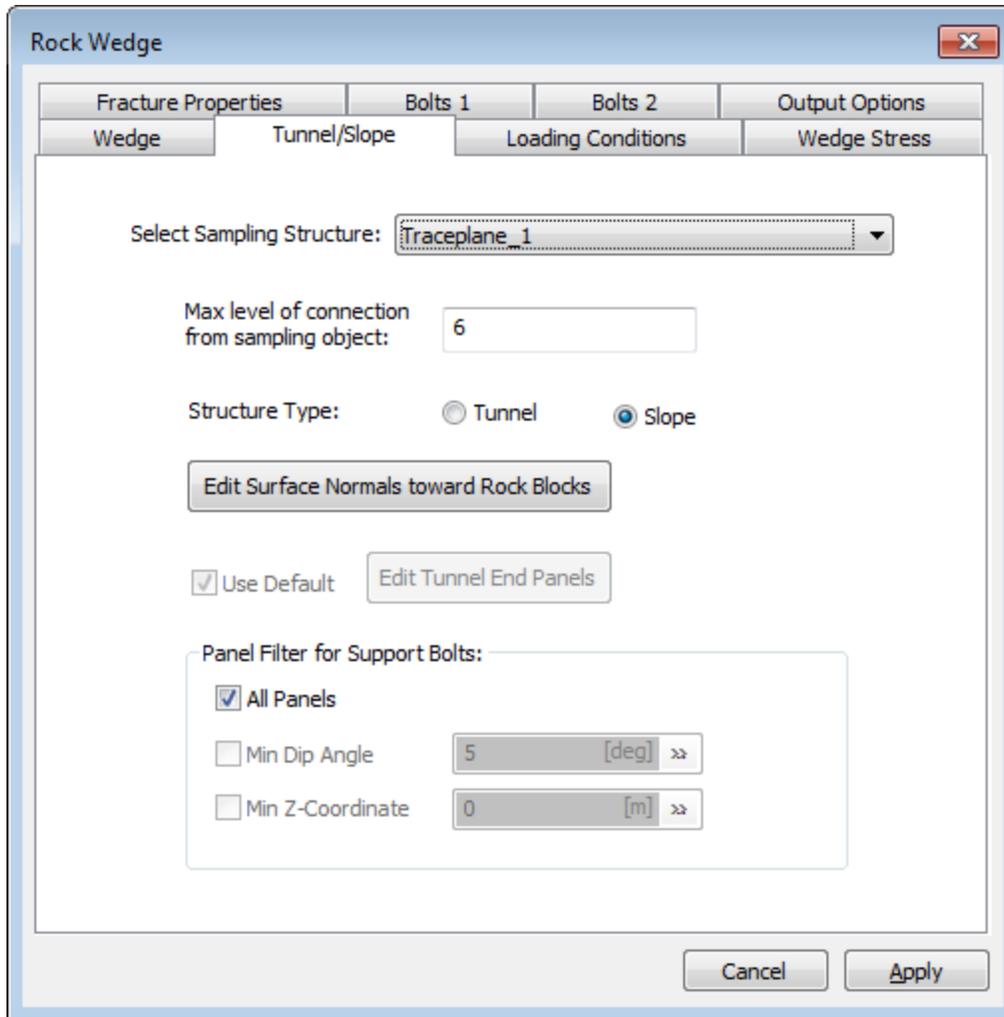


The “Wedge” tab allows you to specify the fracture sets used in the analysis.

- **Name:** the name you wish your analysis to have in the Definitions window.
- **Filter/Objects:** select the fracture sets that you want to include in the analysis.

Tessellated fractures do not work well, so it is recommended that tessellated fractures are “glued” back to polygons using the Fracture > Edit > Tessellate / Untessellate Fractures... function before running the analysis. Fractures that have been clipped on a surface or designed to have “T” intersections may not extend far enough to show fracture/surface intersections so it can also be helpful to slightly increase the size of the fractures using the Fracture > Edit > Scale Fractures... function to ensure that fractures show appropriate intersections.

Tunnel/Slope



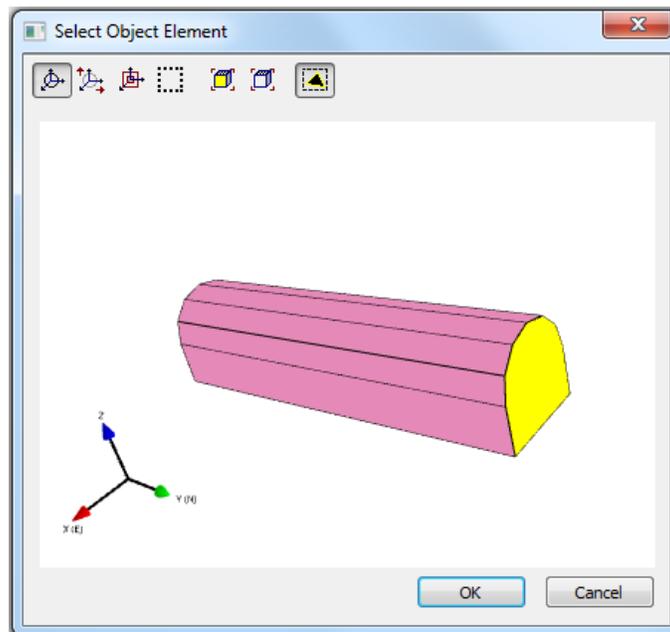
The “Tunnel/Slope” tab allows you to specify the tunnel or slope surface used in the analysis.

- **Select Sampling Structure:** choose the surface or region you wish to use as your slope/tunnel for the model from the dropdown menu.
- **Max level of connection from sampling object:** This represents the number of fracture connections away from the surface that FracMan will search in order to find an enclosed block. This prevents FracMan searching indefinitely for pathways that may not be present.
- **Structure Type:** This identifies which side of the surface to treat as the free side. When slope is selected the upper side is void and with a tunnel, it is the lower side.
- **Edit Surface Normals toward Rock Blocks:** opens the Edit Surface dialog as described in the Manual section [Edit Surface Normals](#).
- **Edit Tunnel End Panels:** if “Tunnel” is selected, you must specify the entrance and exit directions either by accepting the default or by unchecking “Default” and clicking on this button which opens the Select Object Element dialog.

- **Use Default:** FracMan automatically selects the end panels based on finding panels having more edges.
- **Panel Filter for Support Bolts:** If bolts are used for the system, the surfaces that include bolting can be restricted by either dip angle or by vertical coordinates.

Tunnel ends

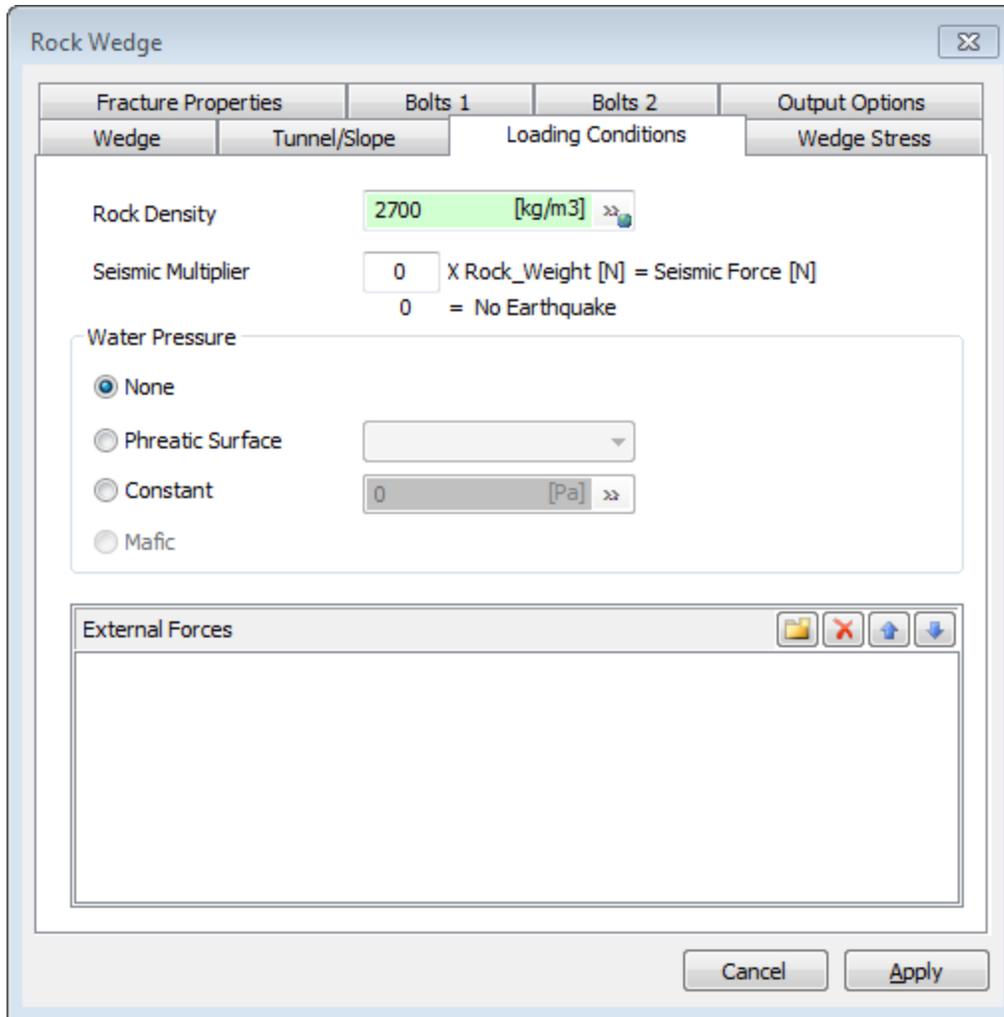
Tunnel ends are treated as empty space and are not taken into account as potential rock-block sources when computing the analysis. To specify the ends of the tunnel, click **Edit Tunnel End Panels**. (Again, “Use Default” must be unselected to enable this.) This will bring up **Select Object Element** dialog box, and you may select one or more faces of your region to be treated as empty space.



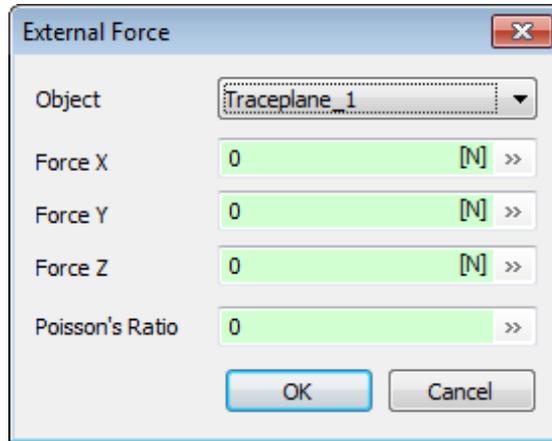
You can move the image here just as you can manipulate the visualization window containing your entire model – that is, you can rotate, translate, zoom, and center your view just the same, using the same controls. In order to select which surface to use as the floor of the tunnel, simply click that face of the region in the image. The face will appear highlighted in the image to show that it is selected. To unselect, simply click the face again. The highlighting will vanish, showing that the face has been unselected. You may select more than one face by using Ctrl-Click. When you have chosen your tunnel ends satisfactorily, click on **OK**.

Loading Conditions

The “Loading Conditions” tab allows you to specify some of the parameters used to calculate the forces acting on a block including rock density (used calculate the weight of the rock block), seismic (earthquake) acceleration, water pressure and custom external forces for the area of the analysis. Other forces which may be acting on the block include bolt forces which can be defined on other tabs.

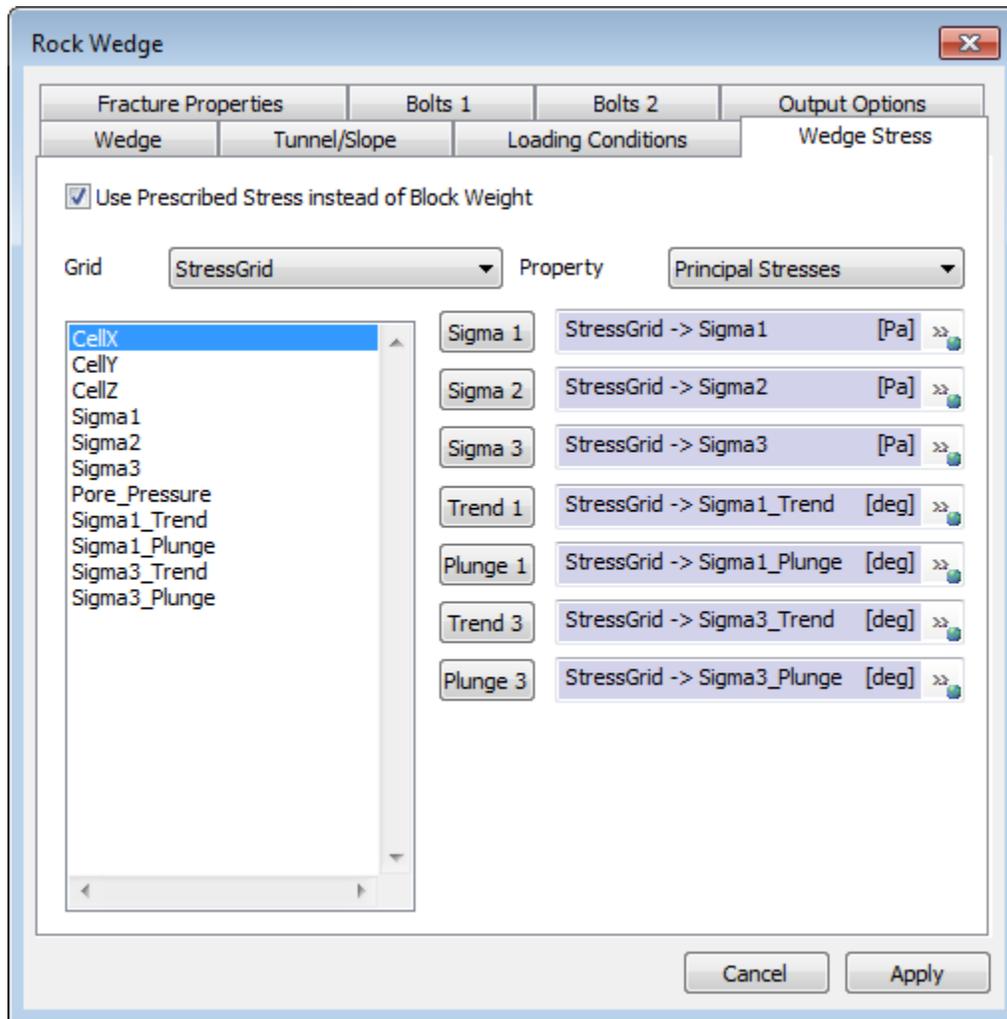


- *Rock Density:* density of the rock through which the fractures and slope/tunnel pass.
- *Seismic Acceleration:* this allows the addition of a seismic loading force as a value of gravitational acceleration.
- *Water Pressure:* “None” implies a completely dry system. “Phreatic Surface” allows you to choose an existing surface or single fracture in your model to define the water table in your model. “Constant” allows you to specify a constant water pressure for the model. In the future a “Mafic” flow grid will be able to be loaded to allow a variable pressure field derived by flow simulation to be used.
- *External Forces:* one or more external forces can be added by clicking on the New(Insert) icon on the External Forces panel. See the section on [External Loads](#) in the Rock Wedge Theory section for details.



Wedge Stress

Normally the weight of a rock block is included in the factor of safety calculation for the block. This option allows you to specify a stress field that includes the effects of the block's weight, and use this instead of the block's weight in this calculation



For example you might use FracStress to generate a stress field and use the output from this as the inputs for this option in the Rock Wedge analysis. The FracStress definition would typically include the 'Use Gravity' option to ensure that the output stress field included the effects from the weight of the rock. Note that for this to give reasonable results the grid on which the stress field is defined needs to be fine compared with the rock block sizes.

There are also associated new output options that allows you to output the calculated x, y and z components of the force on a block due to the specified stress field; and some critical stress statistics (minimum, maximum and area weighted mean critical stress for the block faces).

Fracture Properties

The "Fracture Properties" tab allows the user to specify the fracture properties and failure model that are used to determine the stability of potential sliding blocks.

The screenshot shows the 'Rock Wedge' software window with the 'Fracture Properties' tab selected. The window has a title bar with a close button. Below the title bar are four tabs: 'Wedge', 'Tunnel/Slope', 'Loading Conditions', and 'Wedge Stress'. The 'Fracture Properties' tab is active, and it contains a sub-tab 'Bolts 1' and 'Bolts 2'. The 'Fracture Properties' section includes a 'Model' dropdown menu set to 'Mohr-Coulomb'. Below this are several input fields for fracture properties, each with a 'Value' column and a 'Fracture' column. The 'Value' column contains input boxes with numerical values and units, and the 'Fracture' column contains dropdown menus. The properties listed are: Pore Press. (0 [Pa]), Cohesion (0 [Pa]), Friction Angle (26.5 [deg]), Dilatation Angle (0 [deg]), JRC (7), and JCS. At the bottom of the 'Fracture Properties' section is a checked checkbox labeled 'Convert Tessellated Fractures To Planar Polygons'. At the bottom of the window are 'Cancel' and 'Apply' buttons.

Property	Value	Fracture
Model	Mohr-Coulomb	
Pore Press.	0 [Pa]	
Cohesion	0 [Pa]	
Friction Angle	26.5 [deg]	
Dilatation Angle	0 [deg]	
JRC	7	
JCS		

- *Model:* Mohr/Coulomb or Barton/Bandis failure criteria. If the Mohr Coulomb model is selected, the user needs to enter the rock mass friction angle and cohesion. If the

Barton-Bandis model is chosen, the user needs to enter the friction angle, dilation angle, joint roughness coefficient (JRC) and joint wall compression strength (JCS).

- *Properties*: FracMan allows the user to either specify constant global properties to the model or alternatively map the properties from each fracture element or grid cell if fully spatially variable properties have been defined as part of the DFN model.
- *Convert Tessellated Fractures to Planar Polygons*: the rock wedge analysis performs better when non-tessellated fractures are used so it is best to leave this option checked. This option will convert any tessellated fractures into a single planar polygon having the same outline and an orientation equal to the mean pole of the tessellated fracture.

Bolts 1 and 2

In addition to rock wedge identification and stability assessment, FracMan also allows two levels of bolting support to be added to stabilize the rock wedges. The “Bolt 1” tab is designed to add a basic pattern of bolting to the structure, with “Bolt 2” aimed at adding an additional level of support such as longer cable bolts. The support patterns are activated by checking “Add Support Bolts” for each set of rock bolts you wish to define.

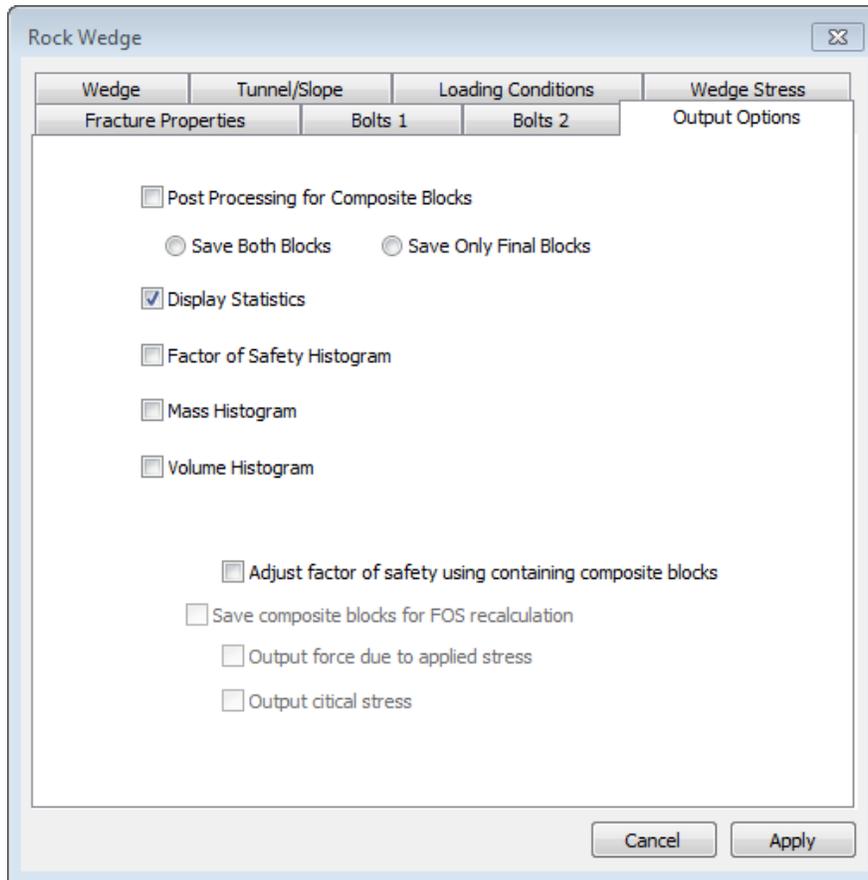
Rock Wedge

Wedge	Tunnel/Slope	Loading Conditions		Wedge Stress
Fracture Properties		Bolts 1	Bolts 2	Output Options
<input checked="" type="checkbox"/> Add Support Bolts				
Spacing				
<input checked="" type="radio"/> Regular Spacing		<input type="radio"/> From Object		
Starting Position		<input checked="" type="radio"/> Fixed Distance		
Along Perimeter		0.5 [m] >>		
Axial Trend		10 [m] >>		
Axial Trend		90 [deg] >>		
Down axis		10 [m] >>		
Plunge		0 [deg] >>		
Angle From Daylight Face Normal				
Trend		0 [deg] >>		
Plunge		0 [deg] >>		
Bolt length		1.8 [m] >>		
Tensile Capacity		18000 [kg] >>		
Ratio of Bonded length (0-1)		1 >>		
Plate Capacity		18000 [kg] >>		
Bolting starts above floor		2 [m] >>		
Bond Strength		60000 [kg/m] >>		
<input type="button" value="Cancel"/>		<input type="button" value="Apply"/>		

- *Spacing*: spacing of bolts with respect to each other. The spacing “Along Perimeter” is the horizontal bolt spacing on a slope or the spacing perpendicular to a tunnel’s axial plane, while the spacing “Down Axis” is the vertical spacing on a slope or the spacing parallel to the direction of a tunnel’s trend.
- *Axial Trend and Plunge*: The default is that bolts are normal to the surface they penetrate but this can be altered to allow bolts that are inclined to the surface.
- *Angle From Daylight Face Normal*: angle of the bolts from upwards-facing side of slope.
- *Bolt Length*: the length of the rock bolts.
- *Ratio of Bonded Length*: fraction of the rock bolt’s length that is glued into the rock.
- *Bolting starts above floor*: distance above bottom edge.
- *Tensile Capacity*: bolt tensile capacity.
- *Plate Capacity*: plate nut-threaded capacity.
- *Bond Strength*: tensile strength of the rock bolt glue.

Output Options

The “Output Options” tab is where you tell FracMan what kind of results you want from the analysis. Just check the box next to the outputs that you desire.



- *Post Processing for Composite Blocks*: If “Save Both Blocks” is selected, the composite blocks will be saved as their component pieces. If “Save Only Final Blocks” is selected, conversely, only the composite block itself will be saved, with the components merged into it.
- *Display Statistics*: If checked, will cause a printout of average and maximum block weight, volume, height, and surface area, as well as the number of blocks, stable blocks, freefall blocks, blocks with a high or low safety factor, factor of safety min, max, average, number of sliding faces and total sliding area.
- *Factor of Safety Histogram*: a histogram of the factor of safety for each block.
- *Weight Histogram*: creates a histogram of the weight of each block.
- *Volume Histogram*: creates a histogram of block volume.
- *Adjust factor of safety using containing composite blocks*

Click Apply to save the definition. The defined rock wedge analysis will now appear in the Definitions window under “Geomechanics”. To run the analysis, you can either right-click on the definition name and select Generate Rock Blocks, or you can open the definition and click

Compute. Depending on the output options you selected, once the analysis is complete, your statistics and histograms will appear. In the Objects window, under the parent “Rock Blocks”, any blocks/wedges that FracMan picked out will be shown, grouped by whether or not they are composite depending on your chosen output options.

Rock Wedge Theory

Identify Rock Blocks

Rock blocks are identified corresponding to the entire interior of the tunnel with the tunnel/slope tracemap. Blocks are constructed by taking fractures which form closed two-dimensional blocks in the tunnel/slope tracemap, forming tracemaps on those fractures and repeating the process until all fractures participating in tracemaps have tracemaps of their own. At this stage, we have a collection of faces and connection information. The collection of faces is then processed using an “unfolding” algorithm to form the minimum volume polyhedron which connects to a tunnel/slope face. Checks are made to ensure that the polyhedron generated is closed. Blocks are constructed until all tunnel/slope faces have been tried.

Rock Block Volume and Mass

The rock block volume is computed by a process of three-dimensional tessellation. The mass is then determined using this computed volume and the rock-density specified in the “Loading Conditions” tab of the Rock Wedge Analysis definition.

External Loads

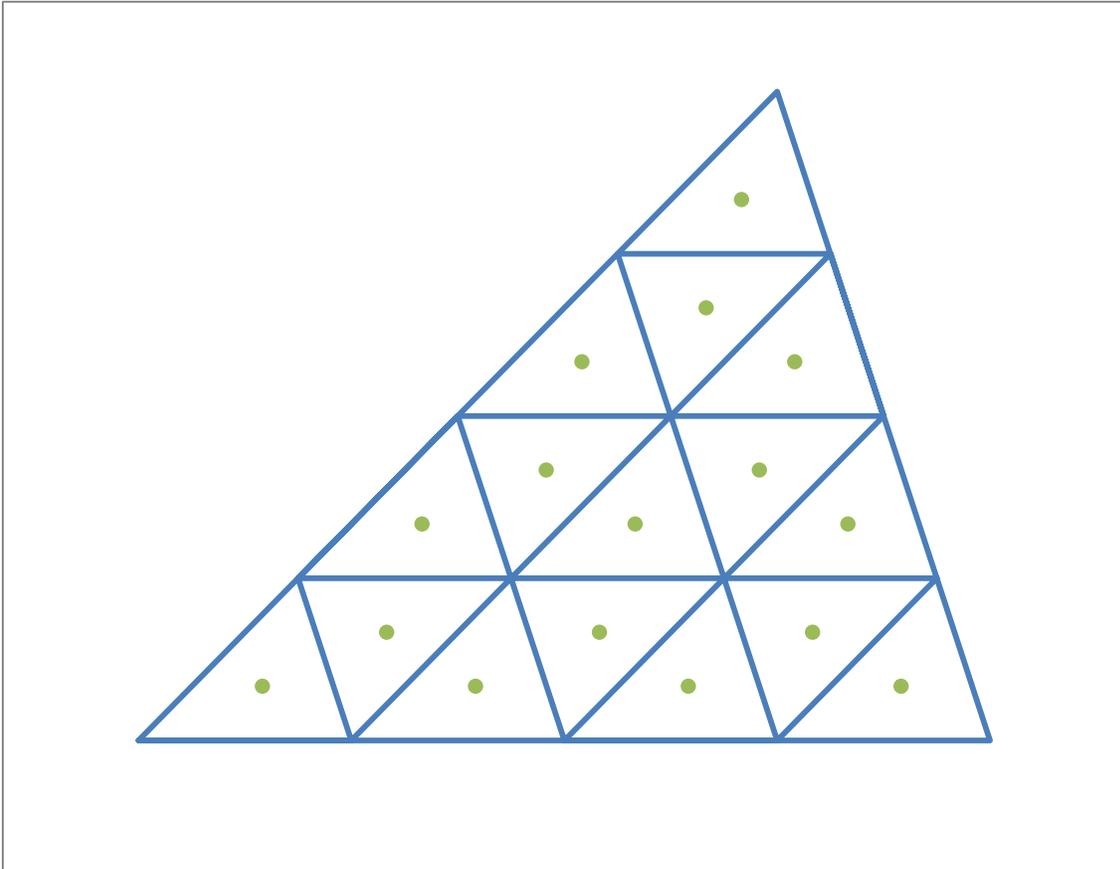
An external load applied to the rock can have a resultant force on a rock block. Fracman allows the user to specify multiple external loads. Each one is applied to a surface and has x, y and z components of force. The Poisson ratio of the rock must also be specified.

Each load is modeled as a constant pressure applied to a rectangular area on an elastic half space with the rectangle on the surface of the half space. An analytical solution for the stress at any point within the half space is used. This is calculated at sample points within each triangle of each block face and used to calculate the force exerted on the block over each of these triangular areas, and hence the resultant force on the block as a whole.

First the load rectangle is determined for the surface. Ideally the surface would be rectangular, but if not, a rectangular approximation is obtained having the following characteristics:

- It’s center is the center of gravity of the surface
- It’s area is the total area of the surface
- It is in the plane defined by the average normal vector to the surface
- Its orientation and aspect ratio are determined by the minimum (non-aligned) bounding rectangle for the projection of the surface vertices onto this plane.

Next points are chosen on the block faces and the stress due to the external load is calculated at each of these points. These points are chosen by subdividing each triangle within the face into a number of congruent triangles, and taking the center of each of these as the sample point.



For each triangle we start with 16 sample points as shown above, and then quadruple the number of sample points until this makes less than 1% difference to any component of the calculated force. (However the number of sample points is limited to 65536 per triangle.)

We could calculate the force exerted on each of the sub-triangles using the stress tensor calculated at its center, and then sum these individual forces. However as each triangle has the same area this is equivalent to using the average of the stress tensors to calculate the force on the whole triangle.

To calculate the stress tensor at a sample point, we first transform the coordinates so that the center of the load rectangle becomes the origin; the downward normal becomes the z axis; and the x and y axes are parallel to the sides of the load rectangle.

We transform the load pressure (i.e. load force divided by the area of the load rectangle) to this coordinate system, and calculate the stress from the three components of this separately (one normal load and two tangential loads). These three stress tensors are then summed, and transformed back into the original coordinate system. The transformation of the stress tensor is equivalent to multiplication on the right by the rotation which converts coordinates from global

to local coordinates, and multiplication on the left by the rotation which converts coordinates from local to global coordinates. For the implementation it is best to view this as a double sum, because the stress tensor is symmetric and you therefore only need to calculate and store six components rather than nine. This corresponds to the tensor form of the transformation:

$$\sigma_{ij} = a_{mi}a_{nj}\sigma_{mn}$$

The calculation of the stress from the tangential and normal loads are based on equations (10) to (15) and (19) to (24) in the paper “Elasticity Solutions for Constant and Linearly Varying Loads Applied to a Rectangular Surface Patch on the Elastic Half-Space” by James R. Dydo and Henry R. Busby. These equations use the pressure components and the Poisson ratio of the rock, but also require values for the partial derivatives of the three harmonic functions ψ , χ and Ω . There are 22 of these partial derivatives in total, which have been calculated for use in FracMan as follows:

$$\begin{aligned} \frac{\partial \psi}{\partial x} &= [[-\log(\bar{y} + \rho)]_{-a}^a]_{-b}^b \\ \frac{\partial \psi}{\partial y} &= [[-\log(\bar{x} + \rho)]_{-a}^a]_{-b}^b \\ \frac{\partial \psi}{\partial z} &= \left[\left[-\tan^{-1} \left(\frac{\bar{x}\bar{y}}{z\rho} \right) \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^2 \psi}{\partial x^2} &= \left[\left[\frac{\bar{x}}{\rho(\bar{y} + \rho)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^2 \psi}{\partial x \partial y} &= \left[\left[\frac{1}{\rho} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^2 \psi}{\partial y^2} &= \left[\left[\frac{\bar{y}}{\rho(\bar{x} + \rho)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^2 \psi}{\partial x \partial z} &= \left[\left[\frac{z\bar{y}}{\rho(z^2 + \bar{x}^2)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^2 \psi}{\partial y \partial z} &= \left[\left[\frac{z\bar{x}}{\rho(z^2 + \bar{y}^2)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^2 \psi}{\partial z^2} &= \left[\left[\frac{\bar{x}\bar{y}}{\rho} \left(\frac{1}{\bar{y}^2 + z^2} + \frac{1}{\bar{x}^2 + z^2} \right) \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial \chi}{\partial x} &= \frac{\partial \psi}{\partial x} z + [[-\bar{x}T_2 - \bar{y} \log(z + \rho)]_{-a}^a]_{-b}^b \\ \frac{\partial \chi}{\partial y} &= \frac{\partial \psi}{\partial y} z + [[-\bar{y}T_1 - \bar{x} \log(z + \rho)]_{-a}^a]_{-b}^b \\ \frac{\partial^2 \chi}{\partial x^2} &= [[T_2]_{-a}^a]_{-b}^b \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 \chi}{\partial x \partial y} &= [[\log(z + \rho)]_{-a}^a]_{-b}^b \\ \frac{\partial^2 \chi}{\partial y^2} &= [[T_1]_{-a}^a]_{-b}^b \\ \frac{\partial^3 \chi}{\partial x^3} &= \left[\left[\frac{\bar{y}}{\rho(\rho + z)} - \frac{\bar{y}z}{\rho(\bar{x}^2 + z^2)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^3 \chi}{\partial y^3} &= \left[\left[\frac{\bar{x}}{\rho(\rho + z)} - \frac{\bar{x}z}{\rho(\bar{y}^2 + z^2)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^3 \chi}{\partial x^2 \partial y} &= \left[\left[-\frac{\bar{x}}{\rho(z + \rho)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^3 \chi}{\partial x \partial y^2} &= \left[\left[-\frac{\bar{y}}{\rho(z + \rho)} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^3 \Omega}{\partial x^3} &= \left[\left[-\frac{\bar{y}}{\rho + z} + \log(\bar{y} + \rho) \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^3 \Omega}{\partial y^3} &= \left[\left[-\frac{\bar{x}}{\rho + z} + \log(\bar{x} + \rho) \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^3 \Omega}{\partial x^2 \partial y} &= \left[\left[\frac{\bar{x}}{z + \rho} \right]_{-a}^a \right]_{-b}^b \\ \frac{\partial^3 \Omega}{\partial x \partial y^2} &= \left[\left[\frac{\bar{y}}{z + \rho} \right]_{-a}^a \right]_{-b}^b \end{aligned}$$

Here the limits from the definite integrals shown as $\pm a$ and $\pm b$ correspond to $\bar{x} = a - x$, $\bar{x} = -a - x$, $\bar{y} = b - y$ and $\bar{y} = -b - y$ where (x, y, z) is the point at which we are evaluating the stress. So if we denote the expression inside the square brackets as $f(\bar{x}, \bar{y})$ then we would calculate:

$$f(a - x, b - y) - f(-a - x, b - y) - f(a - x, -b - y) + f(-a - x, -b - y)$$

However it is best to order the calculations so that all the expressions are evaluated at one of the four points at a time. This allows common sub-expressions to be re-used, and in particular allows logs and arctangents to be calculated a single time for each argument.

Note that $\rho = \sqrt{\bar{x}^2 + \bar{y}^2 + z^2}$ which differs from equation (1) in the paper in that it appears to be missing the square root.

T_1 and T_2 are given by:

$$T_1 = \tan^{-1}\left(\frac{\bar{x}}{\bar{y}}\right) - \tan^{-1}\left(\frac{\bar{x}z}{\bar{y}\rho}\right)$$

$$T_2 = \tan^{-1}\left(\frac{\bar{y}}{\bar{x}}\right) - \tan^{-1}\left(\frac{\bar{y}z}{\bar{x}\rho}\right)$$

Once the values of the partial derivatives have been evaluated, the stress tensors resulting from tangential and normal components of the load are calculated using equations (10) to (15) and (19) to (24) in the paper (see below). These are then summed and transformed back to the global coordinates as described earlier.

For the tangential load in the x direction the stress tensor is given by the following six values:

$$\sigma_{11} = \sigma_{xx} = \frac{P_x}{2\pi} \left\{ 2(1 + \nu) \frac{\partial\psi}{\partial x} + 2\nu \frac{\partial^3\Omega}{\partial x^3} - z \frac{\partial^3\chi}{\partial x^3} \right\}$$

$$\sigma_{22} = \sigma_{yy} = \frac{P_x}{2\pi} \left\{ 2\nu \frac{\partial\psi}{\partial x} + 2\nu \frac{\partial^3\Omega}{\partial x\partial y^2} - z \frac{\partial^3\chi}{\partial x\partial y^2} \right\}$$

$$\sigma_{33} = \sigma_{zz} = \frac{P_x}{2\pi} \left\{ -z \frac{\partial^2\psi}{\partial x\partial z} \right\}$$

$$\sigma_{12} = \tau_{xy} = \frac{P_x}{2\pi} \left\{ \frac{\partial\psi}{\partial y} + 2\nu \frac{\partial^3\Omega}{\partial x^2\partial y} - z \frac{\partial^3\chi}{\partial x^2\partial y} \right\}$$

$$\sigma_{13} = \tau_{xz} = \frac{P_x}{2\pi} \left\{ \frac{\partial\psi}{\partial z} - z \frac{\partial^2\psi}{\partial x^2} \right\}$$

$$\sigma_{23} = \tau_{yz} = \frac{P_x}{2\pi} \left\{ -z \frac{\partial^2\psi}{\partial x\partial y} \right\}$$

where P_x is the x component of the pressure applied to the load rectangle.

The equations for the tangential load in the y direction are the same, but with x and y interchanged:

$$\sigma_{11} = \sigma_{xx} = \frac{P_y}{2\pi} \left\{ 2\nu \frac{\partial\psi}{\partial y} + 2\nu \frac{\partial^3\Omega}{\partial x^2\partial y} - z \frac{\partial^3\chi}{\partial x^2\partial y} \right\}$$

$$\sigma_{22} = \sigma_{yy} = \frac{P_y}{2\pi} \left\{ 2(1 + \nu) \frac{\partial\psi}{\partial y} + 2\nu \frac{\partial^3\Omega}{\partial y^3} - z \frac{\partial^3\chi}{\partial y^3} \right\}$$

$$\sigma_{33} = \sigma_{zz} = \frac{P_y}{2\pi} \left\{ -z \frac{\partial^2\psi}{\partial y\partial z} \right\}$$

$$\sigma_{12} = \tau_{xy} = \frac{P_y}{2\pi} \left\{ \frac{\partial\psi}{\partial x} + 2\nu \frac{\partial^3\Omega}{\partial x\partial y^2} - z \frac{\partial^3\chi}{\partial x\partial y^2} \right\}$$

$$\sigma_{13} = \tau_{xz} = \frac{P_y}{2\pi} \left\{ -z \frac{\partial^2\psi}{\partial x\partial y} \right\}$$

$$\sigma_{23} = \tau_{yz} = \frac{P_y}{2\pi} \left\{ \frac{\partial\psi}{\partial z} - z \frac{\partial^2\psi}{\partial y^2} \right\}$$

The equations for the normal load (z direction) are as follows:

$$\begin{aligned}\sigma_{11} = \sigma_{xx} &= \frac{P_z}{2\pi} \left\{ 2\nu \frac{\partial\psi}{\partial z} - z \frac{\partial^2\psi}{\partial x^2} - (1 - 2\nu) \frac{\partial^2\chi}{\partial x^2} \right\} \\ \sigma_{22} = \sigma_{yy} &= \frac{P_z}{2\pi} \left\{ 2\nu \frac{\partial\psi}{\partial z} - z \frac{\partial^2\psi}{\partial y^2} - (1 - 2\nu) \frac{\partial^2\chi}{\partial y^2} \right\} \\ \sigma_{33} = \sigma_{zz} &= \frac{P_z}{2\pi} \left\{ \frac{\partial\psi}{\partial z} - z \frac{\partial^2\psi}{\partial z^2} \right\} \\ \sigma_{12} = \tau_{xy} &= \frac{P_z}{2\pi} \left\{ -(1 - 2\nu) \frac{\partial^2\chi}{\partial x\partial y} - z \frac{\partial^2\psi}{\partial x\partial y} \right\} \\ \sigma_{13} = \tau_{xz} &= \frac{P_z}{2\pi} \left\{ -z \frac{\partial^2\psi}{\partial x\partial z} \right\} \\ \sigma_{23} = \tau_{yz} &= \frac{P_z}{2\pi} \left\{ -z \frac{\partial^2\psi}{\partial y\partial z} \right\}\end{aligned}$$

The stress tensors for each load rectangle are then summed and the average of these across the sample points used to calculate the force on the triangle. Once the convergence of this is determined, as described earlier, the force on the block face is calculated and these are summed to give the resultant force on the block as a whole. And this is then combined with other types of forces acting on the block, such as its weight.

Stability Analysis

The original theory was developed for four-sided polygons:

I. Check for unconditional stability:

- If block is in contact with 3 non co-planar faces, then it is unconditionally stable (locked).
- Loop on all faces 3 by 3 (excluding the daylighting face(s), i.e., the faces created by the tunnel)
- Assume that the block is in contact with all 3 faces.
- Solve for the normal force on each face:

$$\begin{bmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \begin{Bmatrix} N_1 \\ N_2 \\ N_3 \end{Bmatrix} = \begin{Bmatrix} -\sum_1^{nf} U_i \cdot \ell_i \\ -\sum_1^{nf} U_i \cdot m_i \\ -\sum_1^{nf} U_i \cdot n_i + W \end{Bmatrix}$$

where

ℓ_1, m_1, n_1 – are the direction cosines of the outward normal to face 1

ℓ_2, m_2, n_2 – are the direction cosines of the outward normal to face 2

ℓ_3, m_3, n_3 – are the direction cosines of the outward normal to face 3

N_1, N_2, N_3 – are the normal forces to faces 1, 2 and 3, respectively

nf – is the number of faces on the block

nb – is the number of bolts crossing the block

U_i – is the water force on face i

ℓ_i, m_i, n_i – are the direction cosines of the outward normal to face i

W – is the weight of the block

If all 3 normal forces are positive, the assumption is correct and the block is unconditionally stable (locked).

II. Check for sliding on two planes:

- If block is not unconditionally stable, check for sliding along the edge between the two planes. (edge connected to the daylight face)
- Loop on all faces 2 by 2 (excluding the daylighting face(s), i.e., the faces created by the tunnel)
- Assume that the block is in contact with the 2 faces
- Solve for the normal force on each face and the shear force along the edge created by the 2 faces:

$$\begin{bmatrix} \ell_1 & \ell_2 & \ell_{12} \\ m_1 & m_2 & m_{12} \\ n_1 & n_2 & n_{12} \end{bmatrix} \begin{Bmatrix} N_1 \\ N_2 \\ S_{12} \end{Bmatrix} = \begin{Bmatrix} -\sum_1^{nf} U_i \cdot \ell_i \\ -\sum_1^{nf} U_i \cdot m_i \\ -\sum_1^{nf} U_i \cdot n_i + W \end{Bmatrix}$$

where

ℓ_1, m_1, n_1 – are the direction cosines of the outward normal to face 1

ℓ_2, m_2, n_2 – are the direction cosines of the outward normal to face 2

$\ell_{12}, m_{12}, n_{12}$ – are the direction cosines of the edge created by faces 1 and 2

N_1, N_2 – are the normal forces to faces 1 and 2, respectively

S_{12} – is the shear force along the edge created by faces 1 and 2

nf – is the number of faces on the block

nb – is the number of bolts crossing the block

U_i – is the water force on face i

ℓ_i, m_i, n_i – are the direction cosines of the outward normal to face i

W – is the weight of the block

If the normal forces to the 2 faces are positive, the assumption is correct and the block slides on those 2 planes.

III. Check for sliding on a single plane

- If block is not unconditionally stable and it does not slide on 2 planes, check for sliding on a single plane.
- Loop on all faces 1 by 1 (excluding the daylighting face(s), i.e., the faces created by the tunnel)
- Assume that the block is in contact with the face.
- Solve for the normal force on the face and 2 shear forces, one is down dip, the other is horizontal:

$$\begin{bmatrix} \ell_1 & \ell_{1d} & \ell_{1h} \\ m_1 & m_{1d} & m_{1h} \\ n_1 & n_{1d} & n_{1h} \end{bmatrix} \begin{Bmatrix} N_1 \\ S_{1d} \\ S_{1h} \end{Bmatrix} = \begin{Bmatrix} -\sum_1^{nf} U_i \cdot \ell_i \\ -\sum_1^{nf} U_i \cdot m_i \\ -\sum_1^{nf} U_i \cdot n_i + W \end{Bmatrix}$$

where

ℓ_1, m_1, n_1 – are the direction cosines of the outward normal to face 1

$\ell_{1d}, m_{1d}, n_{1d}$ – are the direction cosines of the shear direction down dip

$\ell_{1h}, m_{1h}, n_{1h}$ – are the direction cosines of the horizontal shear direction

N_1 – is the normal force to face 1

S_{1d} – is the shear force down dip

S_{1h} – is the shear force in the horizontal direction

nf – is the number of faces on the block

nb – is the number of bolts crossing the block

U_i – is the water force on face i

ℓ_i, m_i, n_i – are the direction cosines of the outward normal to face i

W – is the weight of the block

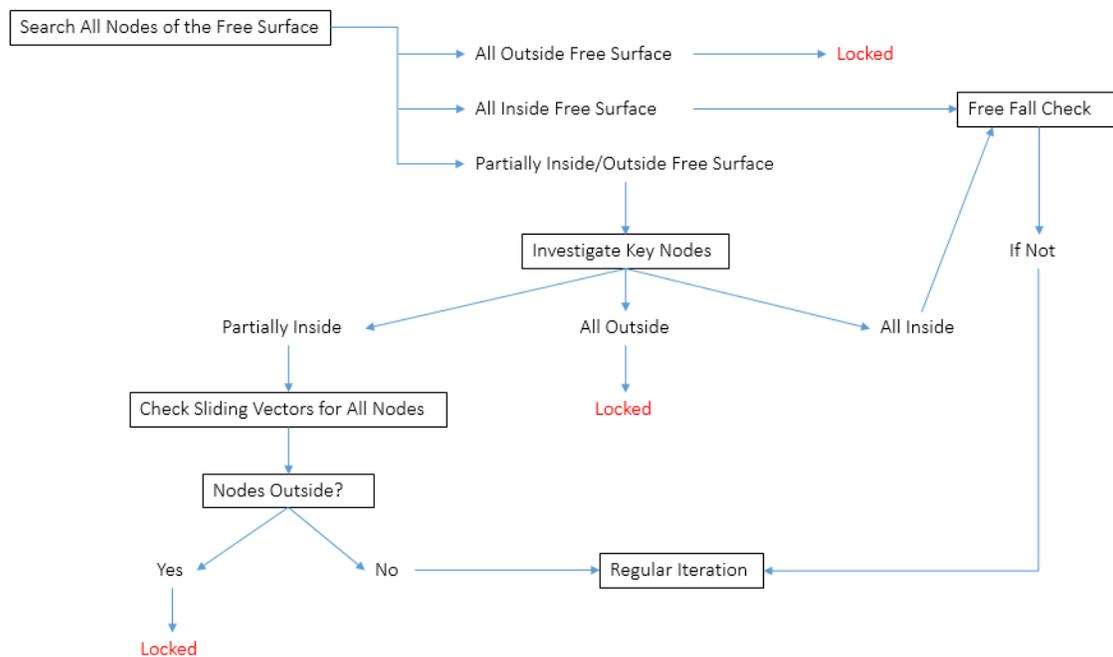
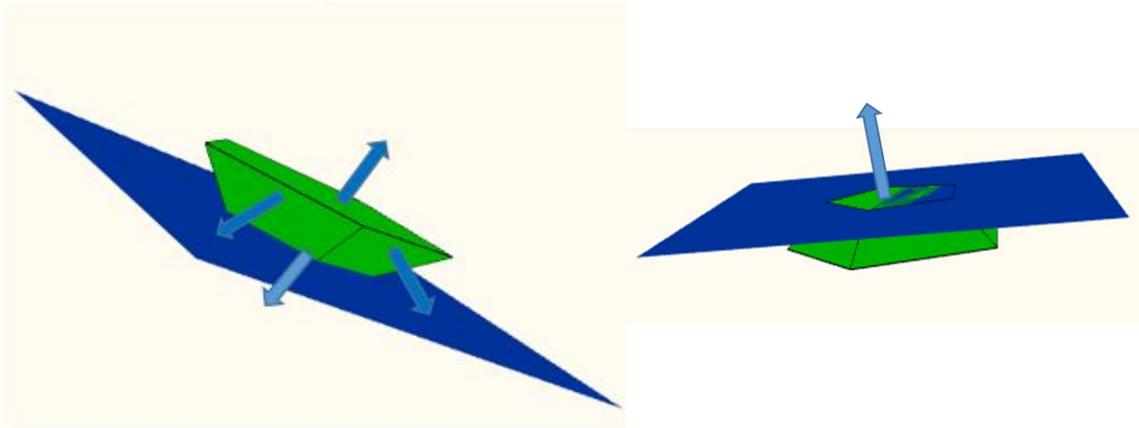
If the normal force is positive, the block can slide along the plane.

IV. Check for free fall

- If block is not unconditionally stable (locked), it does not slide on 2 planes and it does not slide on a single plane, it free falls.

This theory was extended to handle blocks having more than four sides:

- I. Does block have free faces at the bottom (any outward normal vectors to the free faces have a negative z-component)?
- II. Yes -> Get block sliding vector \mathbf{s} .
- III. Remove free faces that are parallel to \mathbf{s} .
- IV. Get internal nodes on the block, V_i .
- V. For each V_i , move it along the \mathbf{s} direction and check if it can slide through any block face.
- VI. If any V_i fails, this block is locked.



Factor of safety calculation

The factor of safety (FS) of a block is calculated in different ways depending on the failure mode of a given block. Stable blocks have an effectively infinite factor of safety and are assigned an FS of 100. Free falling blocks have a factor of safety of zero. Between these two extremes lie the cases of one-plane sliding, two-plane sliding or three-plus-plane sliding which is not well defined. The factor of safety for these one- and two-plane sliding failure modes can be calculated using either a Mohr-Coulomb or Barton-Bandis model as shown below.

For single plane sliding

Using the Mohr-Coulomb model:

$$FS = \frac{A \cdot c + |N'| \cdot \tan(\phi)}{S}$$

where

A – is the area of the face

c – is the cohesion parameter

N' – is the normal force to face

ϕ – is the dilation angle

S – is the magnitude of the shear force

Using the Barton-Bandis model:

$$FS = \frac{|N'| \cdot \tan(\phi)}{S}$$

For two plane sliding

Using the Mohr-Coulomb model:

$$FS = \frac{A_1 \cdot c_1 + |N'_1| \cdot \tan(\phi_1) + A_2 \cdot c_2 + |N'_2| \cdot \tan(\phi_2)}{S_{12}}$$

where

N'_1, N'_2 – are the normal forces to faces 1 and 2, respectively

A_1, A_2 – are the areas of faces 1 and 2, respectively

S_{12} – is the shear force along the edge created by faces 1 and 2

c_i – is the cohesion parameter of face i

ϕ_i – are the dilation angle of face i .

Using the Barton-Bandis model:

$$FS = \frac{|N'_1| \cdot \tan(\phi_1) + |N'_2| \cdot \tan(\phi_2)}{S_{12}}$$

Adjust Factor of Safety using containing composite blocks

Traditionally FracMan created simple blocks on the tunnel or slope, and calculated the stability and factor of safety for these. These 'simple' blocks could be described as atomic because they cannot be split into smaller blocks using the fractures. There are a couple of limitations with this that composite blocks are designed to address: firstly you can have two or more simple blocks that effectively lock each other in, so that no single one of them is unstable, but when taken together they would form an unstable composite block. Secondly as these simple blocks are only the ones on the slope or tunnel face (ones that aren't couldn't possibly be unstable on their own so there would be no point testing them), there is no chance of detecting instability further back into the rock.

To address these issues FracMan has a new option called “Adjust factor of safety using containing composite blocks”. When this option is checked FracMan will:

- Generate all possible composite blocks on the slope or tunnel face using at most 5 fractures per block.
- Generate simple blocks that aren’t on the tunnel or slope as well as those that are, provided they are contained in at least one composite block.
- Do the stability and factor of safety calculation on the composite blocks as well as the simple blocks on the tunnel or slope.
- Update the factor of safety of all simple blocks based on all of the composite blocks that contain them.

The composite blocks themselves are not intended to be viewed by the user, and by default they are not saved in the model. However there is an option that allows them to be retained called “Save composite blocks for FOS recalculation”. As the name suggests this is so that it is possible to redo the stability and factor of safety calculations, perhaps with different parameters, but without having to repeat the block generation. One downside to this is that the model file may become very large due to the number of composite blocks in the model.

Without any constraints, the number of possible composite blocks for even a modest number of fractures would be enormous. Fortunately not all of these would be suitable for running the stability analysis. This calculation is only reliable when the block is formed using a relatively small number of fractures. FracMan therefore restricts the number of fracture faces allowed in a composite block (currently this limit is 5). Even so the number of generated composite blocks in some models can be very large – far larger than the number of simple blocks even though we are now including simple blocks not on the tunnel or slope.

Because we are generating composite blocks in order to do the stability analysis on them, there would be no point in generating a composite block that was not on the tunnel or slope. Although conceptually composite blocks consist of multiple simple blocks, they are not generated by gluing simple blocks together. The generation process is actually more similar to the way simple blocks are generated, as it uses the same set of ‘simple’ faces to form a block. Simple block generation essentially works like this: you start from a simple face on the tunnel or slope and continue adding new faces at each existing edge using the viable face with the minimum angle to the adjacent face until a closed polyhedron is formed (or you can’t add any more faces in which case a block can’t be formed). Composite block generation works in a similar way starting from a simple face on the tunnel or slope, but it considers all possible new faces to add at existing edges (subject to the fracture number limit), and unlike the simple block generation it is implemented using a recursive function.

Before the stability analysis is run on a composite block, its geometry is simplified by gluing together coplanar simple faces that have come from the same fracture or tunnel/slope face. Actually this is the same as for simple blocks, because a simple block can still have fractures that cut into it, but not right across it, and these may split the block’s faces into the smaller simple faces that the block was initially formed from. In terms of the simple block generation this also

explains why the candidate new face with the smallest angle to the adjacent face is not always viable, in which case it has to consider other alternatives.

Once the stability analysis has been run on a complex block, FracMan looks for all simple blocks that are contained within it and updates the stability type and factor of safety values for the simple blocks. For example if the composite block was found to be unstable, then all contained simple blocks are considered to be unstable. The factor of safety that a simple block ends up with is the minimum of the factors of safety of itself and all composite blocks that contain it. Simple blocks not on the tunnel or slope are generated in layers around the ones that are. This works using any starting face that is on the outside of the previous layer. It keeps generating layers until no more blocks can be formed that lie within at least one composite block. Doing it this way ensures that we don't include simple blocks that don't connect to the tunnel or slope at all, and also means that we can re-use the existing simple block generation code without many modifications (we are just using a different starting face). There would be no point in creating a simple block not on the tunnel or slope if it were not contained in at least one composite block, and this helps to restrict the expansion of the simple blocks in models where the fracture network fills a large volume.

At the end of the stability analysis, any stable simple blocks not on the tunnel or slope face are hidden. This gives the user a better view of the blocks that are of more interest – otherwise stable blocks tend to cover everything up so that you can't see the unstable ones.

When it comes to generating statistics about composite blocks, we don't actually use these composite blocks! One reason for that is that the original composite blocks may no longer exist. But perhaps a better reason is that we don't really want to be lumbered with the constraint on the number of fractures. Instead we analyze the simple blocks based on some property such as stability type. We could for example look for unstable simple blocks that have a common face, and keep joining them up until we find a 'composite' unstable block with say the largest mass. This is likely to be larger than if we had used the original composite blocks that had a limit on the number of Fractures involved. One complication though is that we need to base the joining up of the simple blocks on the original simple faces rather than the ones that we end up with after simplifying the geometry. This shows another respect in which the original composite blocks with the fracture limit should only be regarded as temporary objects – not only are there large numbers of them, but they don't include all composite blocks that we might be interested in for statistical purposes.

You might ask why we don't form the complex blocks used for stability analysis the same way, by joining up adjacent simple blocks using common faces. The reason is that when adding a new simple block the number of fractures used in the resulting composite block can increase, but it can also decrease. When recursing through all possibilities that makes it difficult to eliminate branches of the recursion tree, whereas if we are doing it by adding new faces we know that

once we have used up that limit, we can't use faces that are on a new fracture, and that means that the number of possibilities we need to consider is reduced: we can eliminate any such branches on the recursion tree.

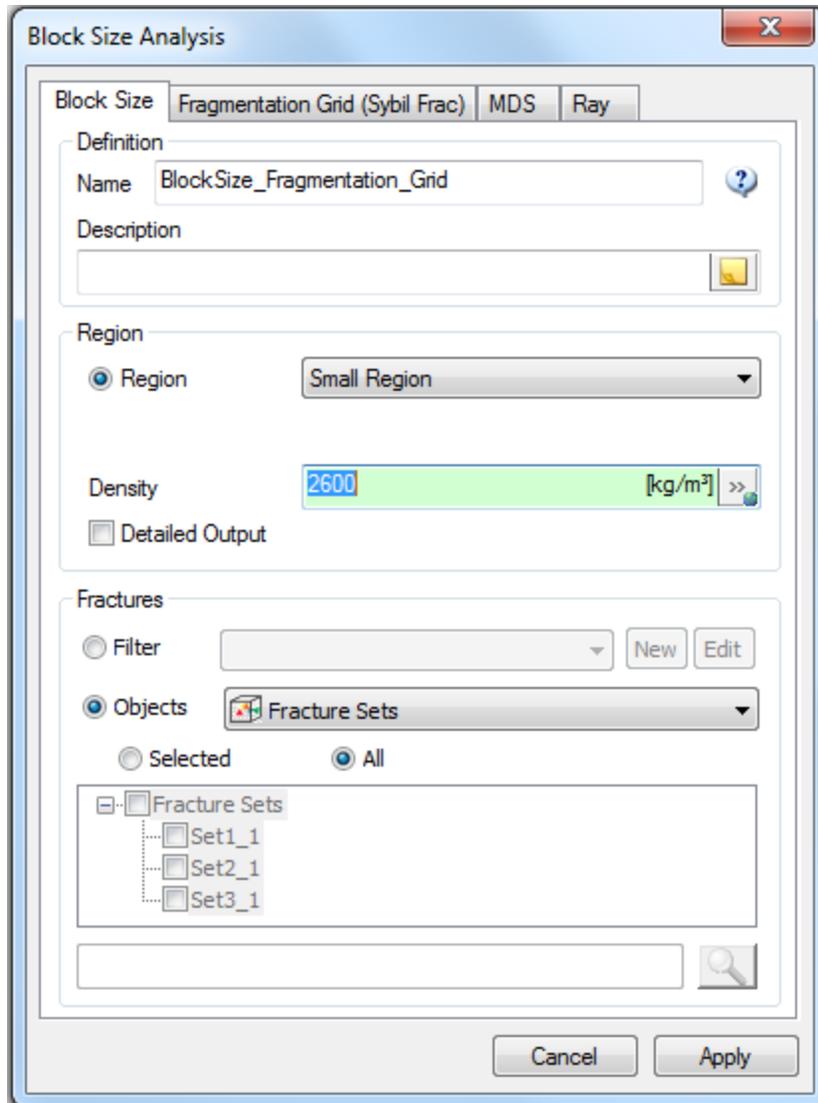
Block Size

Dual porosity behavior is characteristic of fractured reservoirs, and is one of the few aspects of fractured reservoirs which is well addressed by current reservoir assessment technologies. In dual porosity behavior, the different behavior of fluid storage in the fracture system and the matrix is seen in graphs of depletion pressure versus time.

Dual porosity reservoir simulation requires calculation of several parameters having to do with rock matrix block shape and size distributions. The fracture surface area of matrix blocks within a simulation grid cell influences the rate and quantity of fluids that can move between the matrix and the fracture system. The vertical or Z-dimension of matrix blocks influences gravity drainage mechanisms. The shape of the matrix blocks influences the choice of sugar cube, matchstick or slab idealizations in dual-porosity simulators/

The matrix block analysis is designed to provide a quantitative description of block size and shape distributions using parameters which can be directly incorporated into existing dual porosity simulators. These algorithms analyze blocks formed by geologically realistic three-dimensional fracture systems, and provide output in the forms required by conventional dual-porosity simulators.

The purpose of the Block Size analysis is to more accurately map the distribution of rock block size and shape formed from a DFN model. It is more accurate than the Sigma factor approach but needs to be applied to smaller model volumes due to the greater calculation requirements.



On the first tab of the **Block Size Analysis dialog**, you can enter a unique **Name** and optionally a **Description** of this definition. When saved, the definition will appear using this name in the Geomechanics folder in the Definitions window. Select a predefined **Region** to define the volume of the analysis. Enter the rock matrix **Density** (if constant) or click on the symbol on the far right of the field to enter a grid property for a varying density. Check the **Detailed Output** if you want to see a table listing properties for each individual rock block in the report output.

Select which Fracture sets should be included in the analysis in the Fractures section. The search field at the bottom can help find particular sets (matching full or partial names) if there are a large number in the project.

There are three different algorithms for computing the block size, one or more of which can be selected on the remaining tabs.

Fragmentation Grid

The Fragmentation Grid method for determining the block size distribution works by overlaying a fine grid of cells, mapping fractures onto those cells and then assembling rock blocks by joining up the connected grid cells. It is best to determine the approximate average fracture spacing between block faces and ensure that the grid block size is approximately 25% of that value. Therefore the intensity of the model defines the practical volume that can be searched. The maximum number of cells in the fine grid is suggested to be five million such as 150 X 150 X 150. It is best to run the analysis first with fewer cells to explore what minimum cell size is required.

The default view contains fractures (filled), outer box region (frame), grid map (wireframe), and blocks (filled). The statistics reported contain two important values: Shape Factor alpha and Shape Factor beta.

Shape Factor – dimensionless value relating surface area and volume:

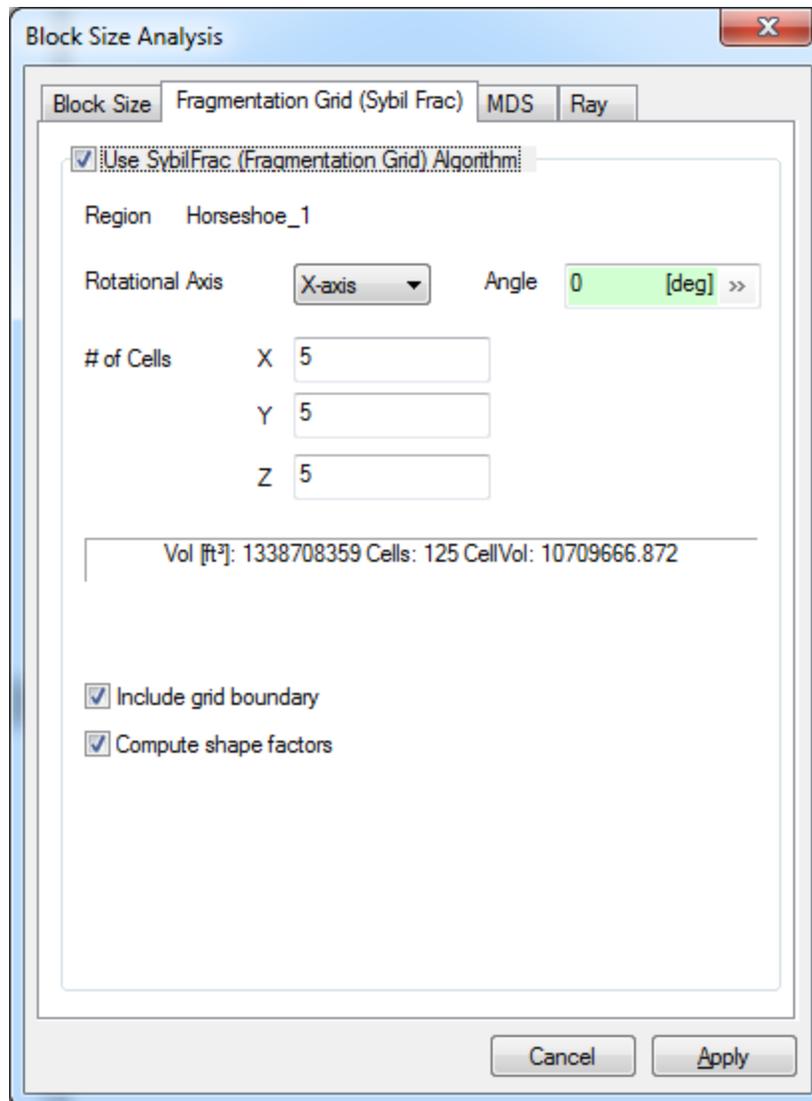
$$\text{Shape Factor} = [(\text{surface area}) * (\text{average chord length})] / [7.7 * (\text{block volume})].$$

A numerical factor of 7.7 is used to normalize the Shape Factor to a value of 1 for a cube. This parameter reflects the flatness of an object. A perfect cube has a shape Factor = 1, and as the aspect ratio of an object increases, the value can approach infinity. The Shape Factor for equidimensional polyhedra with more than 6 sides falls between 0.52 and 1. A perfect sphere yields a shape Factor of 0.52. (“Characterizing block geometry in jointed rockmasses”, Katherine S. Kalenchuk, Mark S. Diederichs, Steve McKinnon, International J. of Rock Mechanics & Mining Sci. 43 (2006) 1212-1225).

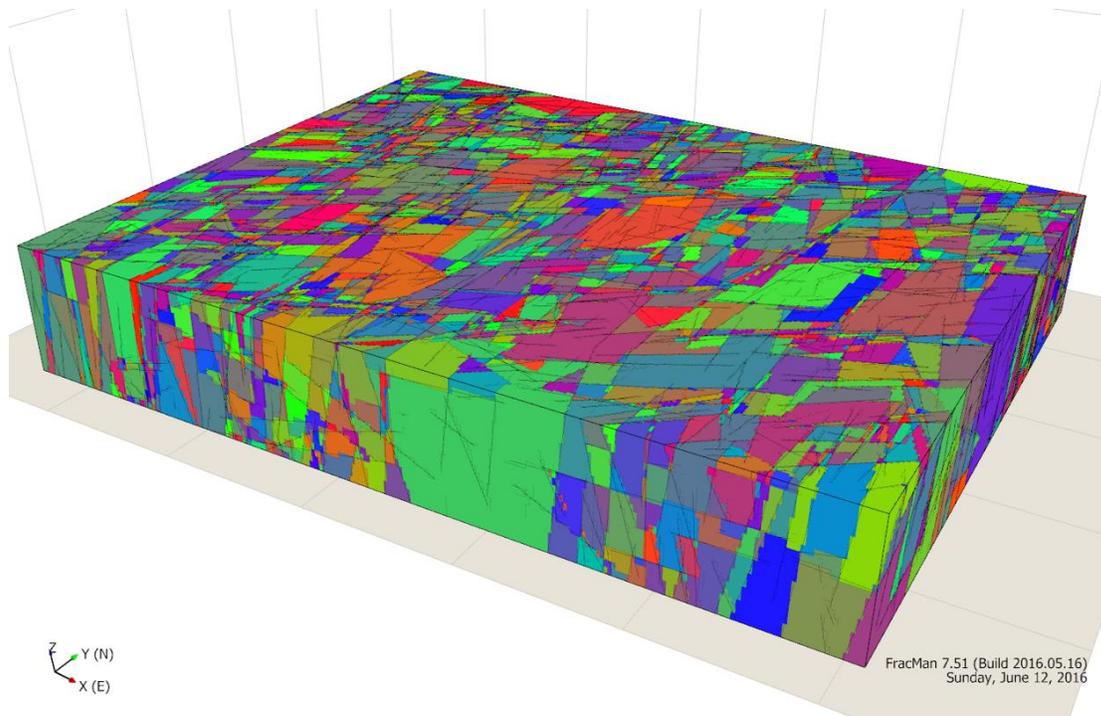
Collinear Factor – Relating to inter-vertex co-linearity in the block. For more elongated shapes, the collinear factor is larger.

$$\beta = 10 \left[\frac{\sum (a \cdot b)^2}{\sum \|a\|^2 \|b\|^2} \right]^2$$

Where a and b are long inter-vertex vectors in the block. The vectors involved in the computation are long chords, no shorter than the median of all inter-vertex chords.

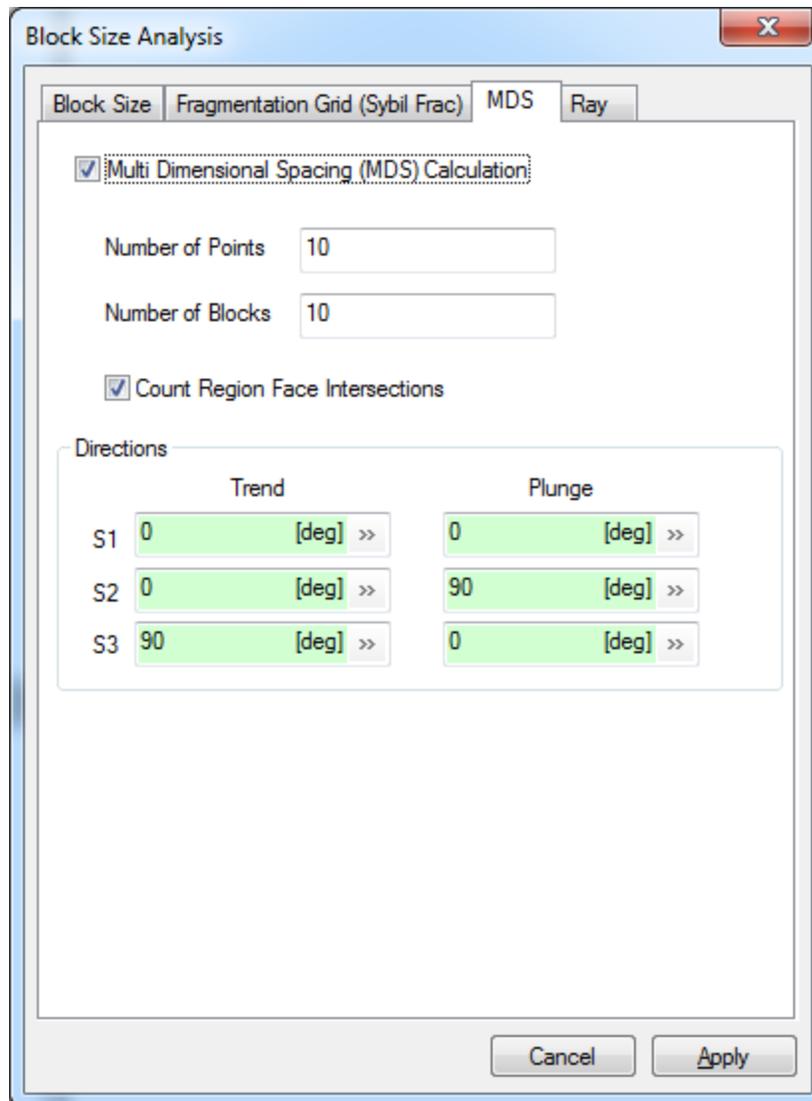


When the analysis is complete, FracMan will open a statistics **Summary** report including a tab for statistics (cell counts and average and standard deviations of block volume, surface area, shape factor, collinearity, and inter-vertex length) and one tab for a chart of the % passing by block volume. The report can be copied or saved by right-clicking on the data and selecting the appropriate option. Also, under the Objects window a new parent named after the Block Size Definition will appear with the report and generated fragmentation grid.

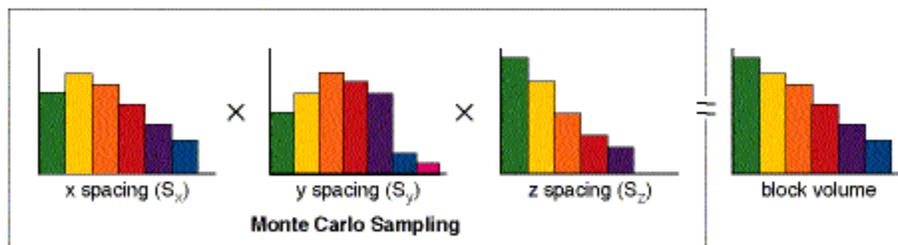
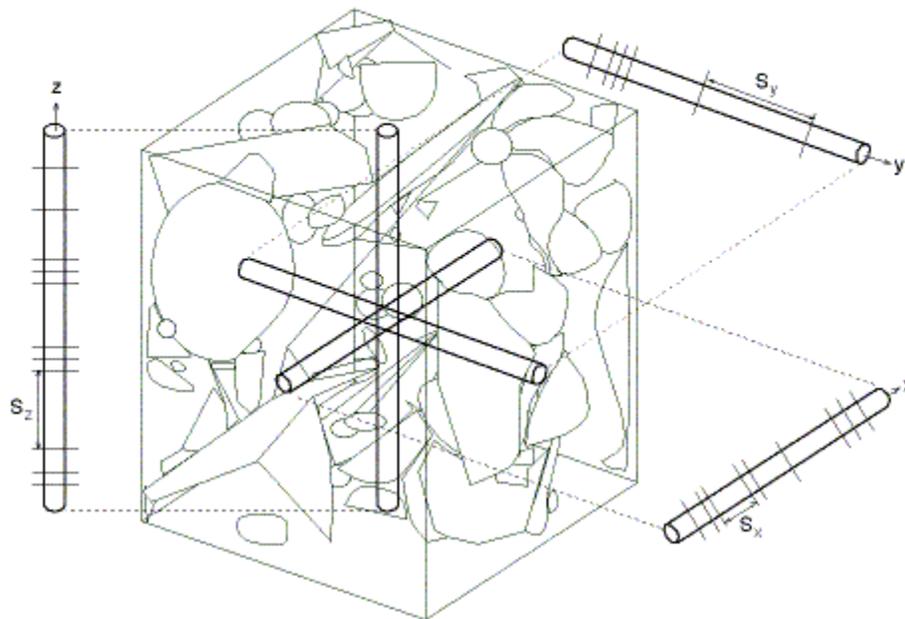


Multi-Dimensional Spacing (MDS)

The second algorithm for computing Block Size is the “Multi-Dimensional Spacing (MDS)” algorithm.



- Number of Points: The number of random points in the region to test
- Number of Blocks: The number of blocks to generate from the spacings at each point.
- Directions: The directions to sample from each random point



For each randomly selected point in the discrete fracture model, a series of randomly-located lines in user-specified directions are generated. The location of fractures intersected by each line is recorded. This leads to a spacing frequency distribution in several directions. Typically, the directions include the vertical direction, in order to calculate the vertical dimension of blocks for gravity drainage considerations, and in two or three orthogonal directions that relate to simulator grid layering geometry and the fracture system.

The spacing probability distributions are multiplied together using Monte Carlo sampling techniques to produce a frequency distribution of block volumes and surface areas. This is carried out by selecting X, Y and Z spacing values at random with selection probability proportional to their frequency, and multiplying them together to create a prismatic block.

The volume of the block for the i-th Monte Carlo computation is the product of the spacing values:

$$Volume_i = X_i * Y_i * Z_i$$

The surface area of the i-th block is given by:

$$SA_i = 2 * (X_i * Y_i + Z_i * Y_i + X_i * Z_i)$$

The block shape is given by the ratios of the mean X, Y and Z spacings:

$$\bar{X} : \bar{Y} : \bar{Z}$$

The vertical dimension of the blocks is taken as the mean value of the Z spacing distribution:

$$Vertical\ Dimension = \bar{Z}$$

The sigma factor of the i-th block for dual porosity analysis from Kazemi, 1976:

$$\sigma_i = 4 * \left(\frac{1}{X_i^2} + \frac{1}{Y_i^2} + \frac{1}{Z_i^2} \right)$$

The sigma factor is used as a shape factor in dual-porosity reservoir simulators.

Ray Cast Volume

Dershowitz (1987) presents a simple algorithm for calculating rock block sizes. In this algorithm, random points are located within the fracture network, and rays are sent out in random or user specified directions until they intersect fractures. The Ray Cast Volume algorithm is a good estimate of block volume when the block is spherical. If the block has a slab shape, then the input ray directions should be adjusted to align with the principle directions of the slab.

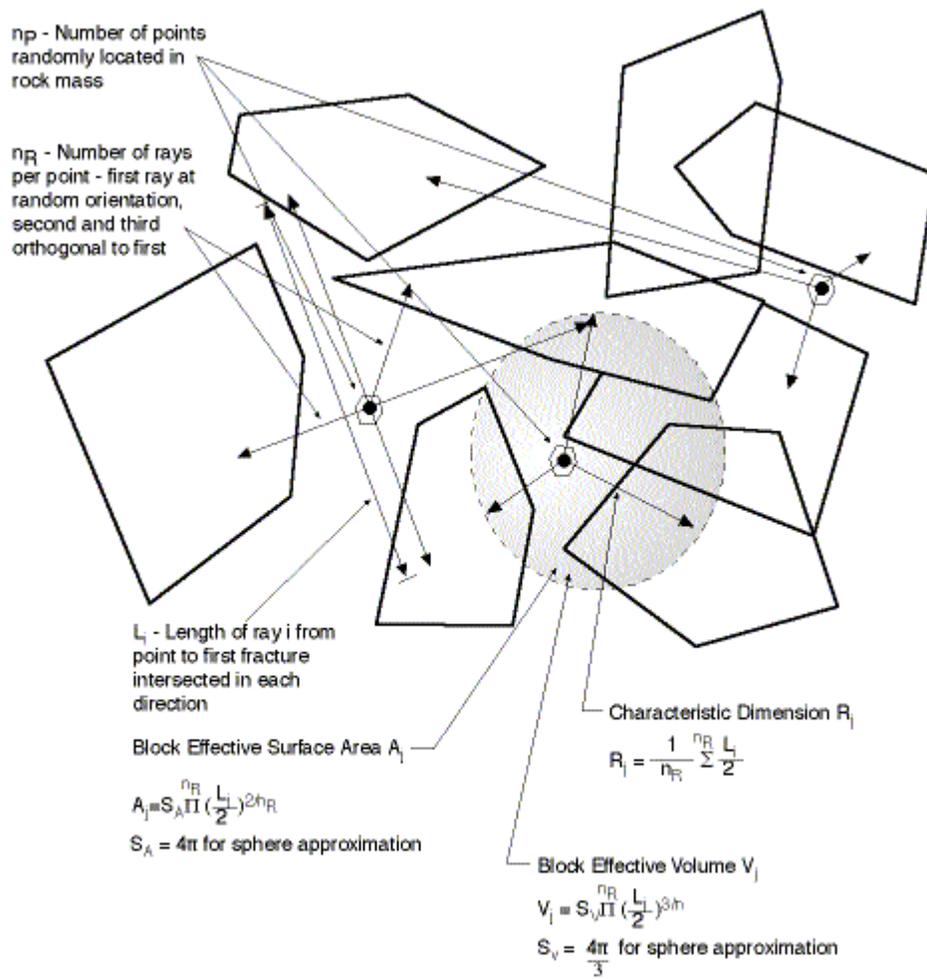
The characteristic dimension R, surface area A, and volume V for each rock block can be calculated from these ray lengths as:

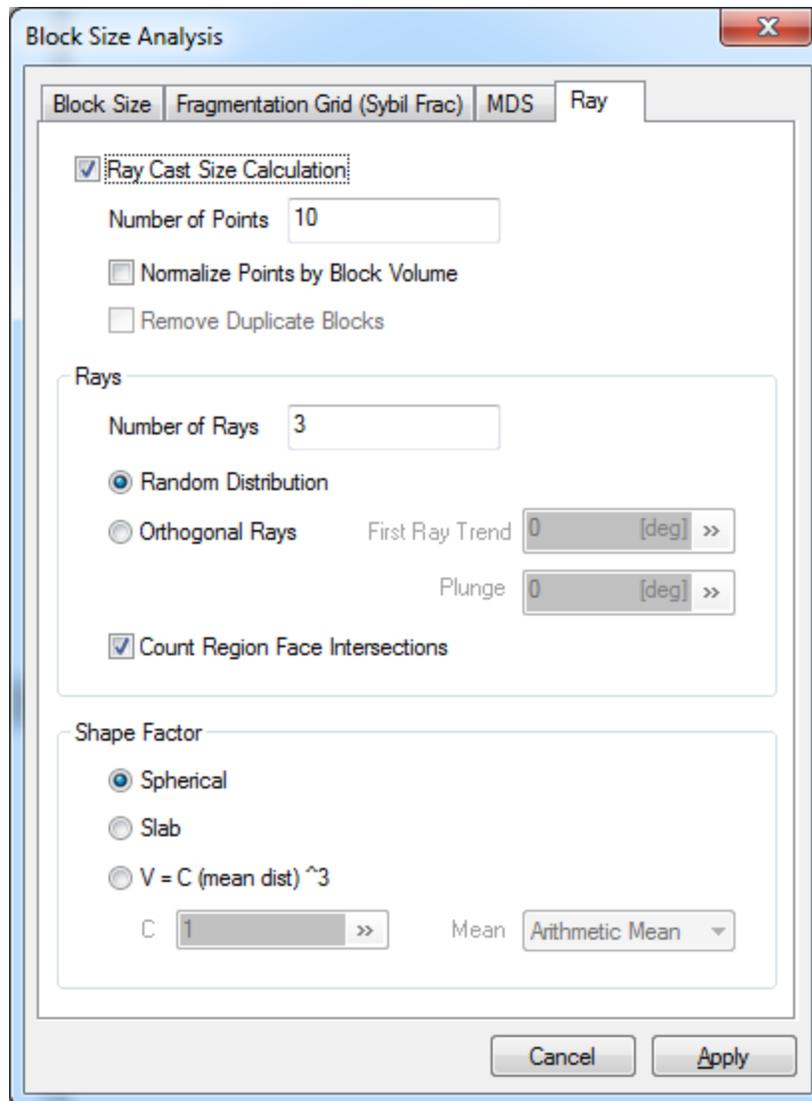
$$R = \frac{1}{n_R} \sum_{i=1}^{n_R} \left(\frac{L_i}{2} \right)$$

$$A = S_A \prod_{i=1}^{n_R} \left(\frac{L_i}{2} \right)^{2/n}$$

$$V = S_V \prod_{i=1}^{n_R} \left(\frac{L_i}{2} \right)^{3/n}$$

where S_A and S_V are area and volume shape factors dependent on the block shape, n_R is the number of rays for each point, and L_i is the length measured along a line through the point defining the block to fractures on either side of the point. As initially presented, the problems with this algorithm include (a) a bias toward sampling of larger blocks, since random points are more likely to occur in larger blocks, and (b) the sensitivity to shape factors S_A and S_V . These limitations can be overcome to some extent by normalizing the blocks by volume and having the user specify the most appropriate shape factor.





- **Number of Points:** The number of random points to sample
- **Normalize Points by Block Volume:** Helps correct the bias toward sampling of larger blocks
- **Remove Duplicate Blocks:**
- **Number of Rays:** the number of rays to sample if a **Random Distribution** is selected
- **Random Distribution** or **Orthogonal Rays:** For each point, you can specify the rays to be random or orthogonal which will attempt to evenly spaced around the sphere (beginning at the direction specified)
- **Count Region Face Intersections:** This will count the intersection with the outer region as a fracture intersection.
- **Shape Factor:**
 - Spherical** – equal to 4π
 - Slab** –

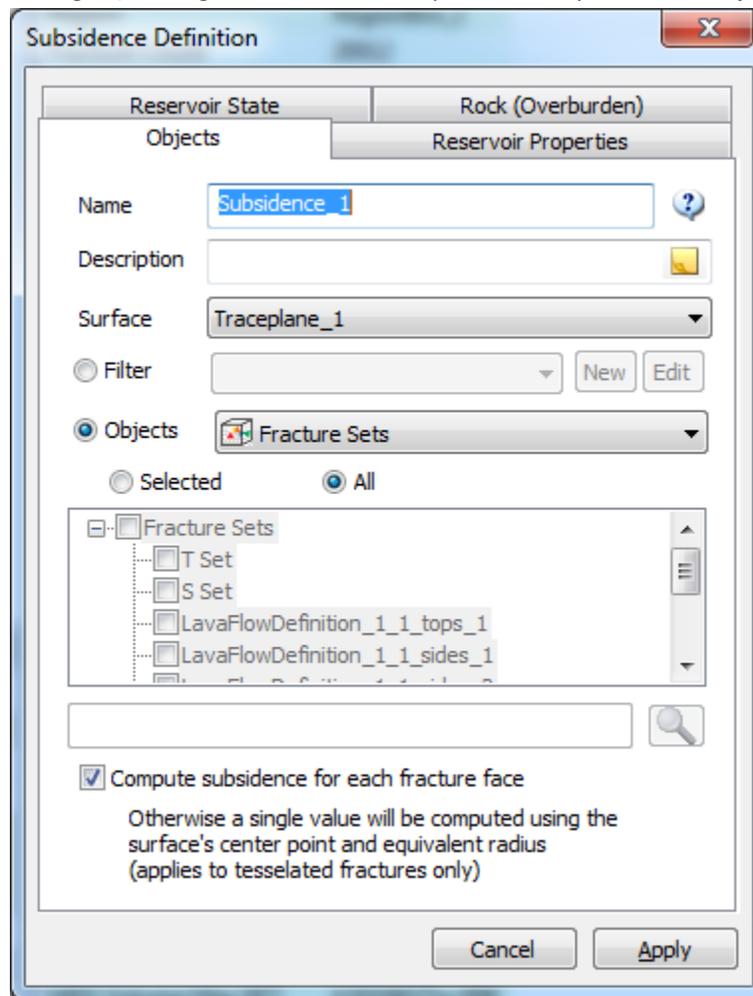
$V=C(\text{mean dist})^3$ – cubic where the user specifies the scaling factor, **C**, and the method of calculating the **Mean** of the Characteristic dimension

Subsidence

This feature calculates ground surface subsidence due to compaction of a reservoir using the Geertsma approximation (1973). Compaction of the reservoir is caused by a change in pressure or temperature or both. The analysis creates or updates a vertex property on the surface with the same name as the subsidence definition.

Subsidence Objects

Fracture objects are used to represent the reservoirs. The fracture objects should be tessellated (discretized into triangles). The ground surface is represented by a surface object.



Reservoir Properties

The image shows a software dialog box titled "Subsidence Definition". It has two tabs: "Reservoir State" and "Rock (Overburden)". The "Rock (Overburden)" tab is active, and within it, the "Reservoir Properties" sub-tab is selected. The dialog contains three property groups, each with a radio button for "Constant" and a dropdown menu for "Fracture Property".

Property	Value	Units
Young's Modulus (E)	0.000406106	[psi]
Poisson's Ratio (nu)	0.25	
Coefficient of linear thermal expansion	0	

The rock mass between (and including) the reservoir and the ground surface is assumed to be homogeneous and isotropic. The bulk modulus, Poisson's ratio, and the coefficient of linear expansion are used to determine both the compaction of the reservoir and the resulting surface subsidence.

Reservoir State

The image shows a software dialog box titled "Subsidence Definition". It has two tabs: "Objects" and "Reservoir Properties". The "Reservoir Properties" tab is active, and within it, the "Rock (Overburden)" sub-tab is selected. The dialog is divided into three sections, each with a "Constant" radio button selected and a "Fracture Property" radio button. The "Reservoir Thickness (h)" section has a value of 20 [m]. The "Reservoir Pressure Drop" section has a value of 0.08 [psi]. The "Average Change in Temperature" section has a value of 0 [°C]. At the bottom of the dialog are "Cancel" and "Apply" buttons.

The thickness, pressure change, and temperature change of multiple reservoirs can be taken as constants or represented as fracture properties. Note that a positive pressure drop or a positive temperature change results in reservoir compaction.

Rock (Overburden)

The image shows a software dialog box titled "Subsidence Definition". It has two main tabs: "Objects" and "Reservoir Properties". Under "Reservoir Properties", there is a sub-tab for "Rock (Overburden)".

Under "Rock (Overburden)", there are two sections:

- Young's Modulus (E):** The "Constant" radio button is selected. The value is 0.000406106 [psi]. There is a "Fracture Property" dropdown menu which is currently empty.
- Poisson's Ratio (nu):** The "Constant" radio button is selected. The value is 0.25. There is a "Fracture Property" dropdown menu which is currently empty.

At the bottom of the dialog, there are "Cancel" and "Apply" buttons.

Hydraulic Fracturing Simulation

Hydraulic fracturing simulation in FracMan is based on the theory of Critical Stress Analysis and involves solving a constitutive relation conserving the material mass on the domain of a stochastic fracture network. An induced tensile fracture develops from the well perpendicular to the minimum stress direction. Intersecting natural fractures are checked for a dilation criterion and slurry is pumped into the dilatable ones. A dilatable fracture is the one with a normal stress less than the fracture pore pressure. FracMan simulates the hydraulic fracturing by maintaining a volume balance between the pumped slurry and the expanded volume of the natural fractures and the developed hydro-fracture. The dilatable volume of fractures is related to the elastic properties of the rock, stress regime, and the internal fracture pore pressure through a constitutive equation which is solved in time steps for the discretized fractures.

Constitutive Equations and Fracture Geometry

The total volume of pumped fluid is divided between the hydraulic fracture and the connected network of natural fractures prone to dilation. At each time step a user specified fraction of the

total volume ($\text{Pumping_rate} \times \text{Duration} / \text{Number of Time_Steps}$) \times HF% is pumped to the hydraulic fracture developing from the wellbore. Subsequently, the hydraulic fracture expands according to the Secor and Pollard (1975)² solution. The intersected natural fractures (first level connections) are checked for their injection possibility, i.e., if the fracture pore pressure is more than the normal stress ($P_{frac} > \sigma_N$). In the case where normal stress is higher than fracture pore pressure, the fracture is not dilatant. The same search is performed for the second level connections and so on. The aperture of the dilatant natural fractures also expands according to Secor and Pollard model. If the connected network of dilatant natural fractures cannot accommodate the total volume of pumped material ($\text{Pumping_Rate} \times \text{Duration} / \text{Number_of_Time_Steps}$) \times (1-HF%), the remaining volume of slurry is pumped into the hydraulic fracture to develop it further. Otherwise, the slurry flows into the network based on a priority function discussed in the manual section [Injection Priority](#).

Fracture aperture is calculated from the elastic equation similar to a width-pressure relation equation in elliptical elastic fracture:

$$e = \left(\frac{2(1-\nu)}{\mu} \right) (P_{frac} - \sigma) d_{max} \sqrt{1 - \left(\frac{d}{d_{max}} \right)^2}$$

Where, μ is the shear modulus. The aperture is a function of the elastic properties, pressure, and distance. The relationship between the Young's modulus E and shear modulus μ is:

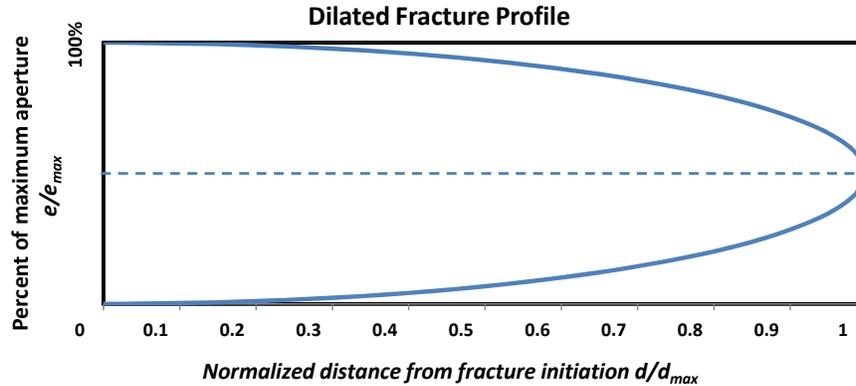
$$\mu = \frac{E}{2(1+\nu)}$$

Making this substitution, the equation for aperture becomes the following:

$$e = \frac{4(1-\nu^2)}{E} (P_{frac} - \sigma_N) d_{max} \sqrt{1 - \left(\frac{d}{d_{max}} \right)^2}$$

The initial apertures of the natural fractures do not affect the final values as each fracture opening is only a function of the fracture pore pressure, normal stress difference, fracture length and elastic properties of the rock mass. A high fracture pore pressure will result in a wide aperture and consequently short hydraulic fracture length. Similarly, the elastic modulus is inversely proportionate with the fracture aperture and directly proportionate with the hydraulic fracture length. The hydraulic fracture network in a stiff rock mass tends to be long while a soft rock mass will result in wide fractures with short hydraulic fracturing extent. The plot below illustrates how the aperture size decreases as the flow distance extends.

² Secor, Donald T., and David D. Pollard. "On the stability of open hydraulic fractures in the Earth's crust." *Geophysical Research Letters* 2.11 (1975): 510-513.



The elastic modulus and Poisson’s ratio of the matrix can be spatially-variable grid properties as is the case for the stress components. In other words, different elastic properties can be assigned to different stratigraphic layers each corresponding to a portion of the grid of elastic properties.

Based on the number of user defined time steps, the total volume of pumped slurry is divided and each fraction is pumped into the hydraulic fracture network in multiple computational steps—sequences. In each sequence pumping pressures and apertures of all of the elements of a developing fracture are updated.

At the beginning of a sequence, newly detected fracture intersections are checked for dilatibility: the pore pressure of the intersecting elements are calculated based on the equation specified by the user on the [Fracture Pore Pressure](#) tab of the Hydraulic Fracture definition, with distance, d , measured from the well-bore along the flow path in fractures and it is compared with the normal stress, σ_N , on the fracture with the normal vectors (n_x, n_y, n_z) .

$$\sigma_N = \sigma_x n_x^2 + \sigma_y n_y^2 + \sigma_z n_z^2 + 2\tau_{xy} n_x n_y + 2\tau_{xz} n_x n_z + 2\tau_{yz} n_y n_z$$

A fracture element is marked as dilatible if the fracture pore pressure is more than the normal stress on it. The dilatible fracture elements intersecting the inflated element from the previous sequence are added to a pool of available elements to receive the slurry material in the next sequence. The priority of the elements available in the pool to inflate is determined through the user defined priority function. The next element to receive fluid is the one with the largest priority value (see Manual entry for [Injection Priority](#)).

The volumes of the injected element and the other previously inflated elements of the fracture are calculated and / or updated. If the total volume of pumped material in the time step is not consumed by the fracture network, the next sequence is performed. The algorithm detects new fracture intersections, calculates their priority indices and updates the injection order. In summary, the dilatible elements are inflated one at a time until the time step volume is

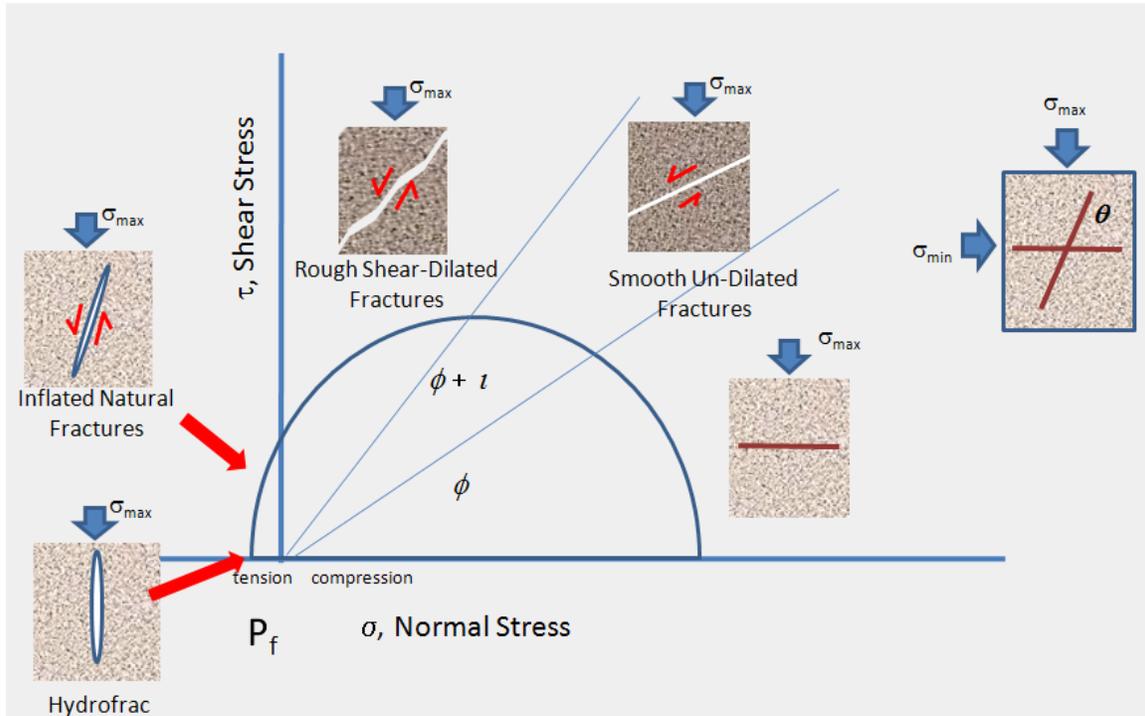
accommodated in the fracture network. The choice of next element to be inflated is based on the injection priority values in the search pool. The search pool is updated after every injected element.

The maximum volume of slurry that can be pumped into the natural fracture network depends on the stress regime, fracture pressure, elastic properties of the medium, and the connectivity, size, and intensity of the natural fractures. If this volume is less than the slurry volume assigned to the natural fracture network, $(\text{Pumping_Rate} \times \text{Duration} / \text{number Time_Steps}) \times (1 - \text{HF}\%)$, the remaining material is pumped into the hydraulic fracture and develops it further and wider. The priority function will be irrelevant in this case since all of the dilatable fractures irrespective of their priority will inflate. The dilatable natural fractures intersecting the developed hydraulic fracture are considered in the next pumping time step.

When all time steps of the hydraulic fracturing simulation are performed, the updated fracture pore pressures are used to conduct a critical stress analysis to delineate the stable and critically stressed fractures. The stress tensor used for the critical stress analysis is built from the grid stress values and the strength parameters (Mohr-Coulomb or Barton-Bandis) of the rock mass are user inputs. A microseismic event is generated as a point in 3D space for each fracture that is critically stressed. The critical stress property of the points is the ratio of shear stress on the fracture to shear strength. Therefore, if the value of the critical stress property is greater than one, then the point is critically stressed. The critically stressed property is set to one if the point is critically stressed, otherwise it is set to zero. All of the dilated fractures will be critically stressed since they have failed in tension (*refer to [Critical Stress Analysis](#) Section to read more about the theory*).

The stress condition of the other fractures will depend on the extent of diffusion of the fracture fluid from the dilated fractures into them and their initial pore pressure and stress condition. Hydro shear effects along the natural fractures can be estimated using the Hydro Shear option. After hydraulic fracture simulation is complete, the hydro shear portion of the calculation is run. The pressure at the tips of the inflated fractures is used to calculate the pressure dissipation along other connected fractures which did not inflate, but may experience enough shear stress to slip. For more information on Hydro shear, refer to the [Hydro Shear](#) Section.

Depending on the values of the fracture pore pressure (P_{frac}), maximum and minimum horizontal stresses (σ_{max} and σ_{min}), the fracture orientation and the strength parameters (ϕ , c , dilation angle l , JRC, JCS), the fracture pressure can be stable, smooth un-dilated, rough shear-dilated, or inflated.



Probable fracture conditions after a critical stress analysis. Depending on the values of the fracture pore pressure (P_{frac}), maximum and minimum horizontal stresses (σ_{max} and σ_{min}), the fracture orientation (θ) and the strength parameters (ϕ , c , dilation angle ι , JRC , JCS), the fracture pressure can be stable, smooth un-dilated, rough shear-dilated, or inflated. The dilated (and not inflated) fractures can be conduits for fast pore pressure diffusion.

Hydraulic Fracture Meshing

The hydrofracture and natural fractures/faults are meshed based on a user specified element size. The element size should be defined considering the size and complexity of the model, the thickness of structural layerings and the computational constraints. The simulation time linearly increases with the number of fracture elements in the model. The meshing algorithm explained below tends to generate square-shaped elements of user specified average size.

To mesh a planar fracture represented by a convex polygon, a 2D bounding rectangle around the fracture is assumed and meshed into smaller, equal-sized grids (also rectangles). The number of grids is determined from the user's specification of element size as following:

$$\text{numGridX} = \text{INT}(\text{Bounding_Length} / \text{element_size} + 0.5)$$

$$\text{numGridY} = \text{INT}(\text{Bounding_Width} / \text{element_size} + 0.5)$$

where numGridX is the number of grids along the bounding rectangle length, and numGridY is the number along the width. The grid dimension, gridLength and gridWidth, thereby slightly differs from the original fracture size and the specified element size.

$gridLength = Bounding_Length / numGridX$
 $gridWidth = Bounding_Width / numGridY$

Further steps are also performed to correct the edges of the fractures with triangular or quadrilateral elements and to ensure that the area of the fracture after meshing is preserved. For small fractures that have an area smaller than 150% of the specified element area ($element_size \times element_size$) only one element is created. The element is set as the bounding rectangle of the fracture or the fracture original geometry if the fracture is quadrilateral.

For tessellated fractures the meshing process depends on whether the fracture is planar or curved. For the planar case, the outline of the tessellated fracture is first constructed and simplified into one polygon, then the above steps are followed to form meshes on the fracture. For curved fractures, the meshing process follows the “surface decimation” method in FracMan. The user’s input parameter of element size is the key parameter in surface decimation where equal-sized triangles are formed on the fracture surface. For hydrofracture simulation, the majority of triangular elements from “surface decimation” are combined into quadrilateral elements pair by pair. Odd numbers of triangles on the edge of surface may be left as triangular elements.

Hydraulic Fracture Definition

The screenshot shows the 'Hydraulic Fracture Definition' dialog box. It has a tabbed interface with 'Fracture Properties' selected. The 'Name' field contains 'HydroFrac_1'. There are checkboxes for 'Report stage details' and 'Full report for every injected element'. Under 'Hydraulic Fracture', 'Hydro-fracture Growth' is set to 'From Intersecting Fractures' and 'Random'. 'Element Size' is set to '100 [m]'. There are also checkboxes for 'Connected Fracture Network', 'Exclusion Zone' (with 'Induced Hydrofrac Only' selected), and 'Update Stress Between Well Stages'. The 'Apply' button is highlighted.

Name – Provide a unique name for the hydraulic fracture definition. A default is provided.

Description – Optional field to provide any additional commentary.

Report stage details – Create an Analysis object containing a report on each hydrofrac stage.

Full report for every injected element – Include element details in the report for stage details.

Hydro-fracture Growth Options – During the simulation, if pumping pressure exceeds the minimum stress, an induced hydraulic fracture will be formed for each stage defined. There are a number of options for hydraulic fracture growth which control the initiation seed of the hydraulic fracture from the well interval:

New Tensile: Hydraulic fracturing will initiate from a random point along the well interval and develop in the direction perpendicular to the minimum principal stress.

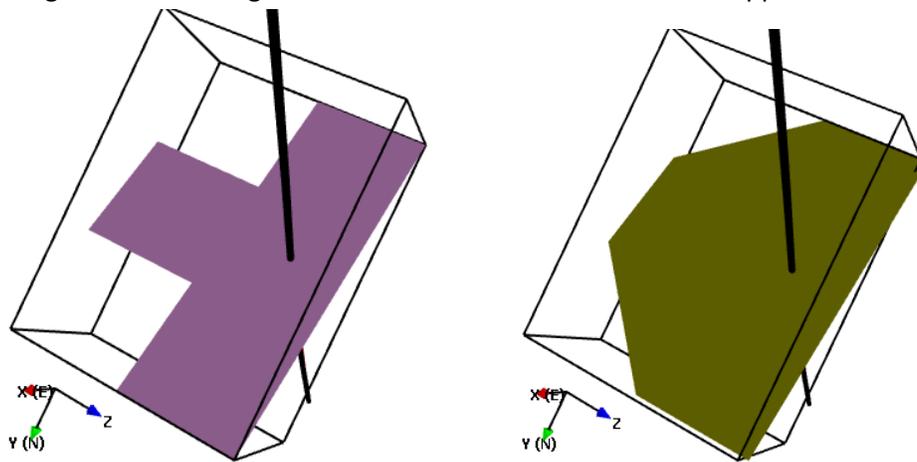
From Intersecting Fractures: The initiation point will be determined based on the existing natural fractures intersecting the well interval. If the choice is made to initiate growth of a hydro-fracture from a pre-existing fracture intersecting the well, then you also need to select how to prioritize which fracture intersection to choose, assuming there are more than one fracture intersections available. The second drop-down box lists the choices:

- Largest Shear stress – Choose the intersecting fracture having the highest shear stress
- Largest Fracture size – Choose the intersecting fracture having the largest size
- Closest to Sigma 2 Orientation – Choose the intersecting fracture oriented closest to the Sigma 2 stress direction
- Random – Choose one of the intersecting fractures at random

Bounding Region – Select a Region object to provide the limits on the simulation volume.

Element Size – Choose an element size large enough to capture the detail required but small enough to run efficiently. See the discussion above about [Meshing](#).

Smooth Outline of Generated Fracture – By default the hydraulic fracture algorithm generates fractures that have a stair-step edge appearance. This option will fill in the stair-steps with added triangles to make the generated surface have a more rounded appearance.

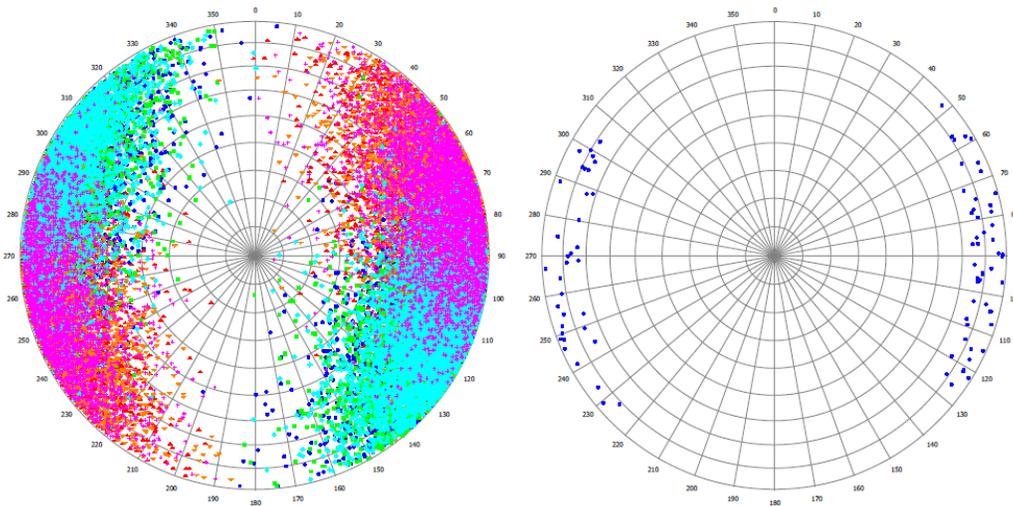


Smooth Off (left); Smooth On (right)

Connected Fracture Network – Select the filter object which contains all of the pre-existing fractures in your simulation that you want to include. You may select one that has already been defined, create a new Filter Definition, or select fractures from the model directly by selecting “Selected Fractures” and pressing the “Select..” button.

The discrete fracture network must be created prior to hydraulic fracturing analysis. For the DFN generation guidelines please refer to corresponding sections of the User’s Manual. It should be noted that the vertical connectivity of the fractures in different geological layers and the fracture orientations and intensities are critical factors in hydraulic fracturing analysis as the sub-vertical fractures sub-parallel to the maximum horizontal stress direction can be major conduits for the progress of the hydraulic fracture.

To improve the solution time, the user can also filter only for the sub-vertical fractures in case of a stress condition where the vertical stress is considerably higher than the horizontal ones. If maximum horizontal stress (σ_{Hmax}) is also considerably higher than minimum horizontal stress (σ_{Hmin}), the trend of the fractures used in the hydraulic fracturing analysis can also be filtered.



Stereonets of the natural fracture network (left) and simulated natural fractures (right). Due to the high vertical stresses only the sub-vertical fractures can be considered in the search for dilatable fractures.

Exclusion Zone – Check here if you are running multiple hydraulic fracturing stages within one definition and you want to mimic stress shadow effects where fractures stimulated during one stage are not re-stimulated from a neighboring stage of stimulation.

- **Induced Hydrofrac Only** – Apply the exclusion zone to only the induced hydro-fracture.
- **Hydrofrac and Inflated Natural Fractures** – Apply the exclusion zone to both the induced hydro-fractures and any inflated natural fractures.
- **Minimum Exclusion Dist** – Enter a value to use for the exclusion zone distance from the stimulated fracture area.

Update Stress Between Well Stages – Identify FracStress and Upscaling definitions to update the stress grid during hydraulic fracturing. In order to use this feature, you need to first set up FracStress and Upscaling definitions. See Manual section on [Dynamic Fracture Analysis -> FracStress](#) for details.

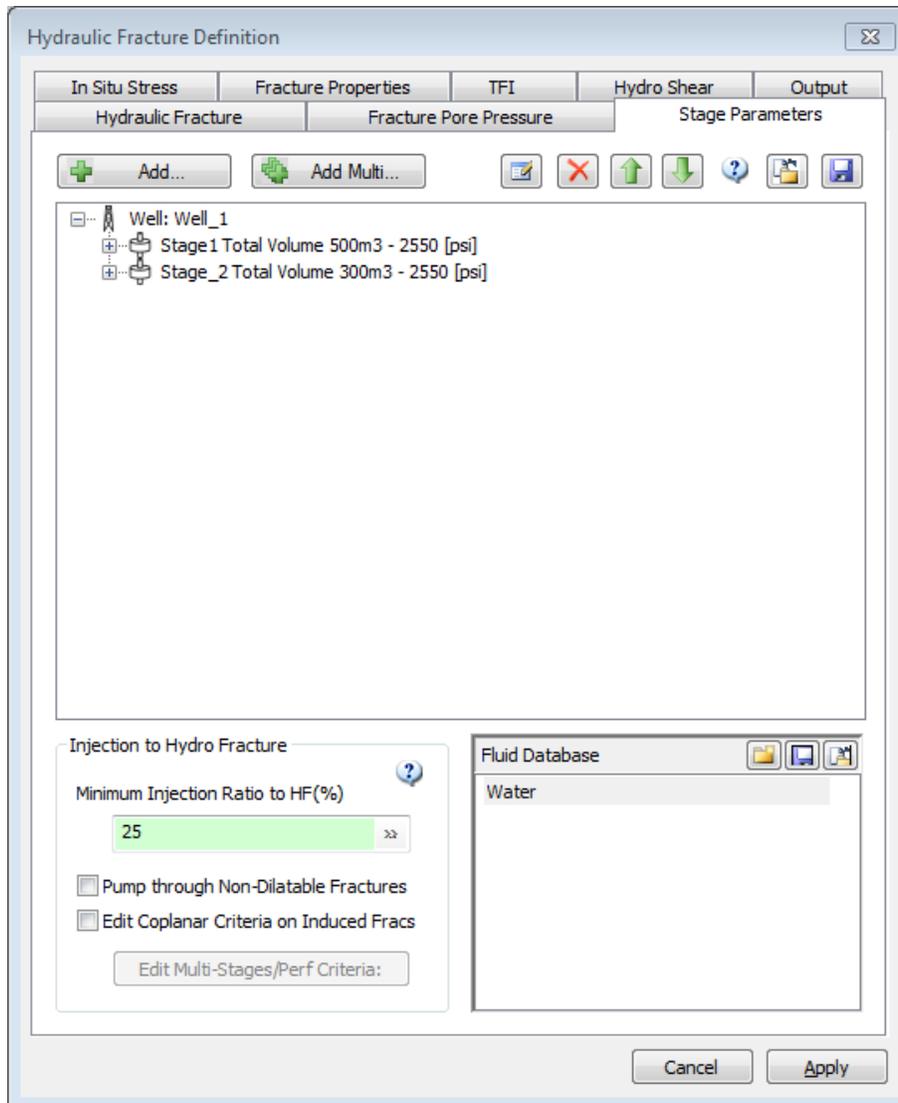
Please note: if you are running *FracStress* in conjunction with your hydraulic fracture simulation (see the image below where the option is selected), the stress grid will get updated during the hydraulic fracture simulation. Therefore, you may not need to select the option Exclusion Zone anymore as it is calculating the stress-strain effects during the simulation and therefore automatically includes stress shadow effects.



Stage Parameters

A hydraulic fracture definition needs to include at least one stimulation stage. These are defined in the top half of this tab. Stage information can be added one at a time using the **Add Stage**

button or loaded from a previously saved definition using the  button. If you want to copy an existing stage to use as a starting point for a new stage, select it and click on the **Duplicate Stage** button.



Opens menu providing some shortcuts to working with a large number of stages:

- Duplicate selected
- Expand All
- Collapse All
- Delete Selected
- Delete From Selected
- Delete All
- Delete All From Selected Well



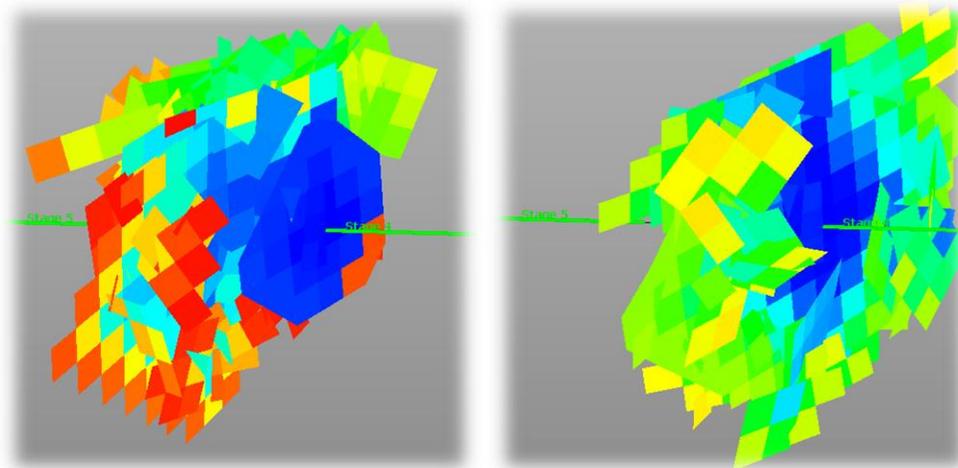
Delete selected.



Move selected Stage or Well up in the list.

-  Move selected Stage or Well down in the list.
-  Open online help to relevant page.
-  Load Pumping Stages from a saved text file.
-  Save Pumping Stages to a text file.

Minimum Injection Ratio: The user specifies the minimum percentage of the total material injected into the hydro-fracture ($HF\% > 0$). Physically, the pressurized fluid preferably migrates into the existing dilatable natural fractures. Thus it is advised to minimize the minimum percentage of the fluid injected to the hydro-fracture. At each time step at least HF% of the total volume is going to be pumped into the hydro-fracture developing it further. So specifying a large HF% results in development of a large induced tensile fracture oriented in the maximum horizontal stress direction and fewer stimulated natural fractures.

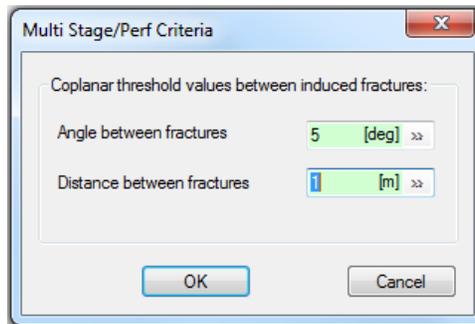


Fracture injection sequence with HF=10% (left) and HF=50% (right). The induced hydro-fracture size in the left figure (dark blue) is considerably smaller than the right figure (dark blue-green). In the former case, 486 natural fracture elements are dilated in the former, compared to 383 elements in the latter.

Pump through Non-Dilatable Fractures: If selected, the pumped fluid will be able to penetrate into all of the fractures but the aperture of them will not be changed from their predefined value. *Warning:* This option is generally *ONLY* useful to visualize the connected network of natural fractures around the hydro-fractured well interval. It does *NOT* show just the stimulated fractures.

Edit Coplanar Criteria on Induced Fracs: In most cases, the user will not need to use this feature. However, in some particular instances, for example when a vertical well has closely spaced perforation intervals, multiple induced fractures may be created simultaneously from different perforations and grow with slightly different angle or small distance from one another.

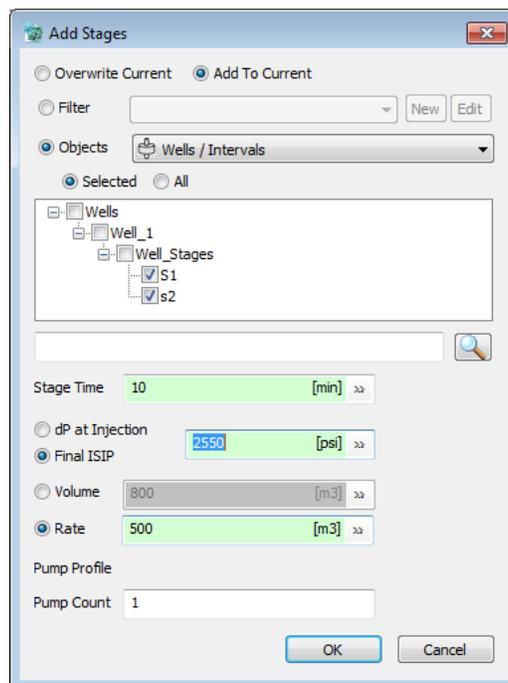
Use the **Edit Multi-Stages/Perf Criteria** button to adjust the acceptable thresholds to let induced fractures of the same stage within the specified distance/angle range get connected without overlapping each other.



The **Fluid Database** is also displayed on the bottom right corner of this tab. Fluids are the combinations of liquids and proppants that are pumped into the well, also called slurries. New fluids can be defined by clicking on the New (Insert)  icon or by loading a previously saved fluid database clicking on the Import Fluid Database  icon. Use the Export fluid Database  icon to save your fluid definitions to a text file.

Add Multi...

You can add multiple stages at once by pressing this button and selecting all the well interval(s) that define the stages, enter pressure and volume or rate information as well as the number of pump stages.



Add/Edit Stage

Pumping parameters include the duration, volumes and type of the hydrofrac fluid. If no proppants or other additives are used, the **Clean** volume will be the same as the **Slurry** volume. Units for the Duration, Clean and Slurry volumes can be changed by right-clicking or double-clicking in the header fields.

The pressure is considered the pressure of the fluid at the point of contact with the rock, i.e. just outside of the well casing. In the Define Fracture Stage dialog the user can specify either **Final ISIP** or **dP at Injection** (pressure difference between pore pressure and normal pressure on fractures) and **Flowback %** along with entering specific pumping information for that stage.

Name	Duration[min]	Clean[gpm]	Slurry[gpm]	Fluid System
1	30.000	3745.000	3745.000	Fluid1
2	30.000	3745.000	3745.000	Fluid2
3	30.000	3745.000	3745.000	Fluid3
4	30.000	3745.000	3745.000	Fluid4
5	30.000	3745.000	3745.000	Fluid5

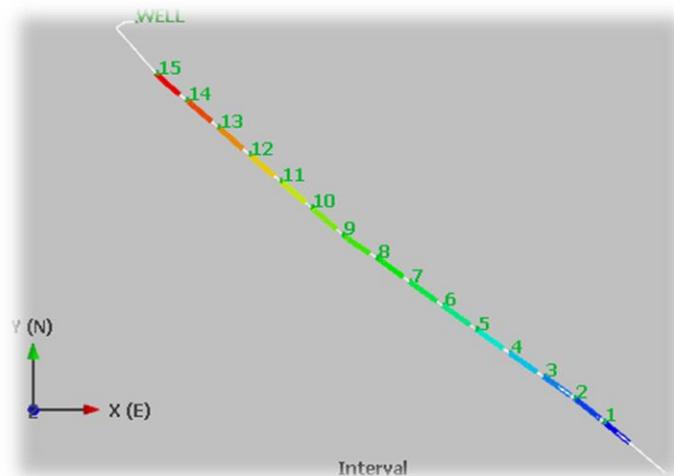
The “pumping pressure” is the slurry pressure outside of the injection casing which can be estimated by the shut-in pressure (ISIP). In other words, pumping pressure is slightly larger than the minimum stress magnitude. An excessive evaluation of the pumping pressure results in a short and wide hydrofracture which can dilate natural fractures with a large range of orientations while a low estimate of the pumping pressure results in dilation of a small number of natural fractures which may not conform to the microseismic or other evidence. It is recommended that the user choose “dP at Injection” which is the pressure difference between pore pressure (pumping pressure) and normal pressure on the fracture at the injection point. This provides the user with the option to let FracMan calculate the appropriate pumping pressure required to initiate fracture at the injection point for each stage independently:

Calculated Pump Pressure for each stage by FracMan

$$\begin{aligned} &= \text{Normal Stress on a fracture plane at the injection point} \\ &+ dp \text{ (User defined delta pressure at the injection point)} \end{aligned}$$

Drop in fracture pore pressure along the fracture away from the injection point can be captured or controlled using the “pressure drop slope coefficient”, s , which is explained in the [Fracture Pore Pressure](#) section of this manual. If there are a large number of stages, checking “dP Apply to All Stages” would let the user make changes to “dP at Injection” and apply the new value of dP to all other stages by one click.

Location: Click on the button to select one well interval that will undergo stimulation during this stage. The well completion and stimulation information including the measured depth of the packers, perforations, and ball seats are used to define well intervals. A hydraulic fracturing stage will correspond to each well interval. After importing/defining the well data, the well intervals are defined based on the measured depth (MD) of the installed packers and perforations. It should be noted that the MD of the interval start is less than the MD of its end. Also the interval corresponding to the first hydraulic fracturing stage (at the toe of a horizontal well) will have the deepest MD (compared to the hydro-fracture stages in the heel).

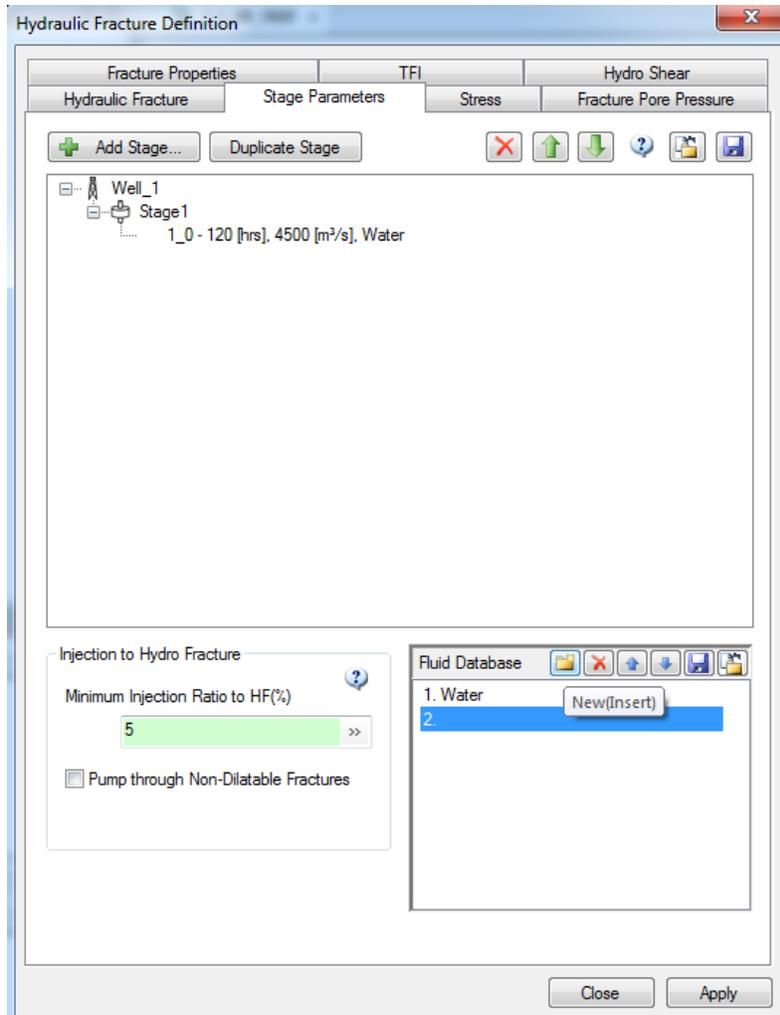


Plan view of a horizontal well with 15 well intervals/hydrofracture stage. The counters are the Measured Depths (MD).

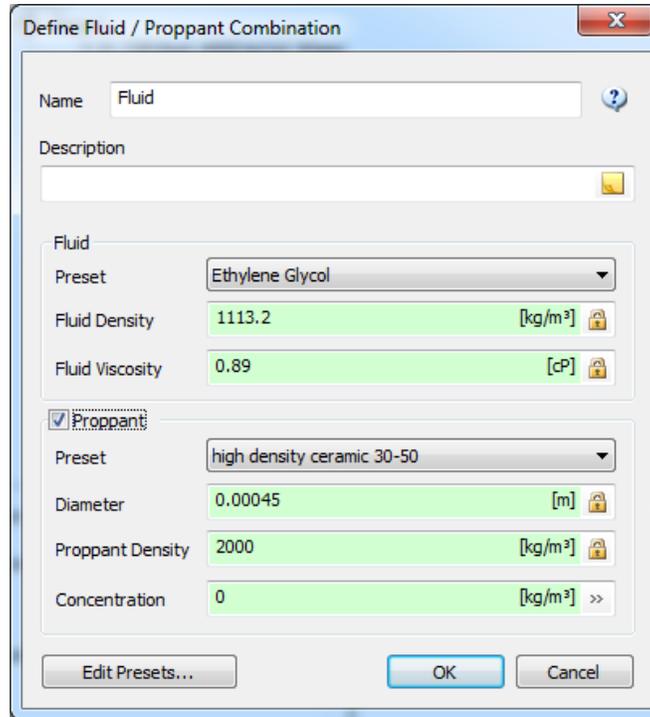
Customize Timesteps: Generally it is recommended that the user not adjust the timesteps used during the simulation. If this is checked, a new column appears for the pumping stages and the number of timesteps used for each pumping stage can be set manually.

Define Fluid: A new fluid/ proppant combination can be defined by clicking on the “New (Insert)” button in the Fluid Database in the Stage Parameters tab.

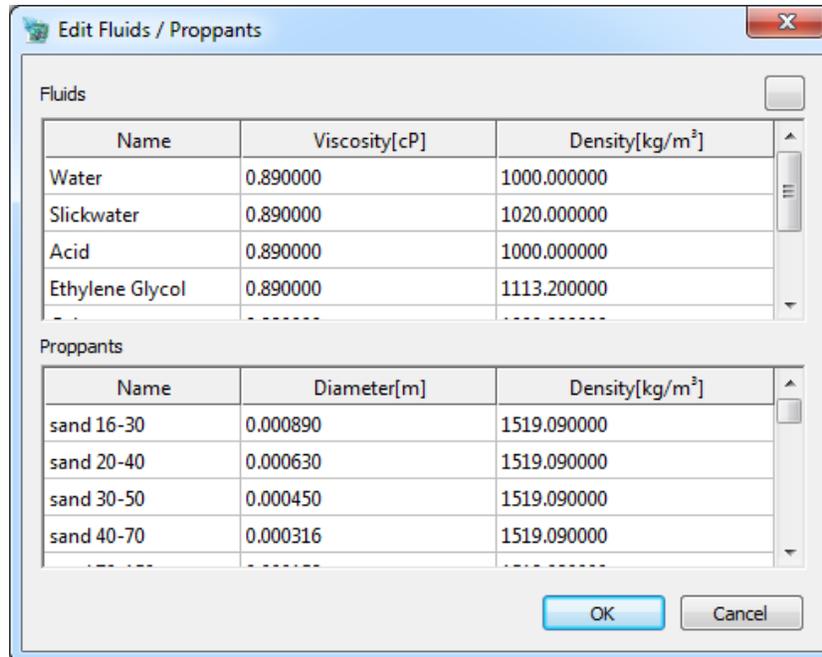
[Define Fluid](#)



The fluid can be given a name and a description and the fluid/ proppant combination can be set using a selection of presets from the dropdown menus. To get the list of proppants the user must first select the Proppant toggle button next to the word “Proppant”.



The presets can also be edited using the “Edit Presets...” button. Once everything for the new fluid/ proppant has been specified, Click Ok to add the new fluid/proppant combination to the fluid data base.



Stress/Elastic Properties Grid

The state of stress including the magnitudes and directions of the principal stresses and the magnitude of the pore pressure can either be saved as grid properties or can be entered as

constants in the **Stress** tab. The stress condition in a grid is defined as principle stresses or as a symmetric stress matrix. If using the symmetric stress matrix, note that compressive stress is positive while tensile stress is negative.

Fracture Pore Pressure

To model the leak-off to matrix or non-dilatable fractures and also the pressure drop due to resistance of fracture surface to flow, the fracture pore pressure can decrease linearly from

well-bore pressure (P_{pump}) by specifying a pressure drop slope s . The user may also choose to include hydrostatic weight of slurry in the fracture pore pressure calculation.

Hydraulic Fracture Definition

In Situ Stress | Fracture Properties | TFI | Hydro Shear | Output

Hydraulic Fracture | Fracture Pore Pressure | Stage Parameters

Fracture Pressure Mode: Secor Pollard

Fracture Pore Pressure (Ppore)

Slope, s : 0

Include static pressure, the vertical elevation, effect

Set max flow distance through fracture network: 1000 [m]

$$P_{\text{pore}} = (P_{\text{pump}} - \sigma_N) \left(1 - \frac{s \cdot d}{d_{\text{max}}} \right) + \sigma_N$$

Graph showing $\frac{(P_{\text{pore}} - \sigma_N)}{(P_{\text{pump}} - \sigma_N)}$ vs d/d_{max} for slopes $s = 0, 0.25, 0.5, 0.75, 1$.

Flow to Fracture Order

$$Prob = \frac{Transmissivity^M \times Orientation^N}{Connection_Level^L}$$

L: 0 | M: 0 | N: 0

Cancel | Apply

The maximum flow distance can be a predefined parameter or estimated iteratively from simulation. It is used in calculating the fracture pore pressure and storage aperture after a fracture that has received fluid. Pore pressure slope describes the variation of pressure drop between a point at the well and a point at the maximum flow distance from the well. Fracture pore pressure is calculated from the empirical equation below:

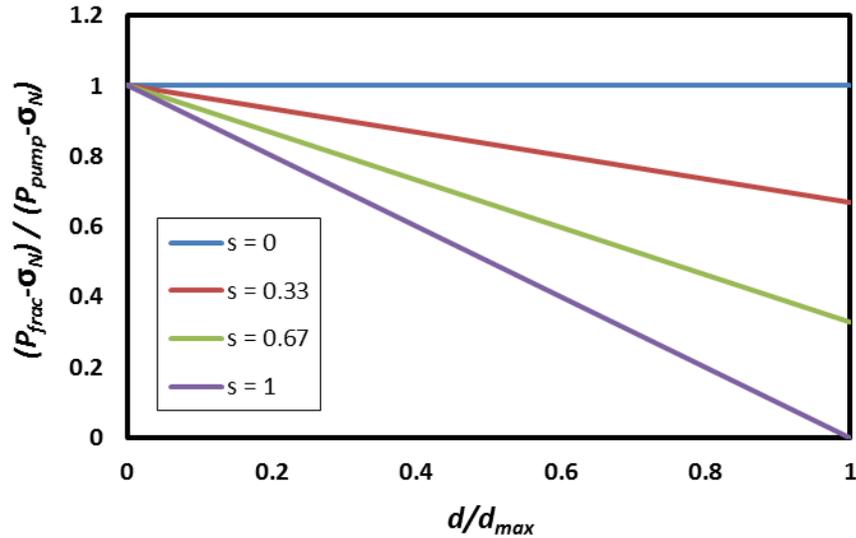
$$P_{frac} = (P_{pump} - \sigma_N) \left(1 - s \frac{d}{d_{max}}\right) + \sigma_N + \rho g(\Delta h)$$

Where:

- P_{pump} = pumping pressure;
- σ_n = normal stress on the fracture end element;
- d = flow distance from the well intersection point to the fracture element;
- d_{max} = maximum flow distance which gets automatically adjusted based on hydraulic fracture length at each time step (can also be user-specified);
- ρ = slurry density;
- g = gravitational acceleration; and
- Δh = vertical elevation difference from the well-intersection point to the fracture element.

The last term, $\rho g \Delta h$, is added to P_{frac} only when the user selects to include static pressure, a vertical elevation effect. If the option to **Include Static Pressure** is selected, the weight of slurry column from the wellbore to the fracture element is considered in the fracture pore pressure calculation. For example, a proppant-laden fluid stage with an 8-ppg concentration has a slurry density of about 1440 kg/m³. Using this slurry, there will be a 0.7 MPa (100 psi) difference between the pressure at the top and bottom of a hydraulic fracture 100 m thick. Consideration of slurry density can be important when the in situ stresses in the model are not considered constant and they have a vertical gradient. In this case, if the slurry's weight is not considered, there will be a propensity for the pumped material to migrate to shallower layers.

The plot below shows how pore pressure drops along the flow distance as a function of slope parameter when static pressure is not included. At the well injection point pressure is set to the pumping pressure. Pore pressure dissipates along the flow path that may be caused by leakage to surrounding rock matrix or resistance from the fracture surface. This factor is taken into account by the parameter, s . Specifying s as zero results in no pressure drop and considering only the hydraulic component of the pressure. This is the recommended approach as the pressure-drop along the length the fractures considering the non-linear, non-Newtonian behavior of the slurry material is not well understood. On the other hand, specifying $s = 1$ results in a fracture pore pressure profile where the pressure is equal to the normal stress on the fracture tip.



Fracture pore pressure drop function for different values of s .

Pressure Mode

Secor Pollard

KGD

PKN

Injection Priority

The pumped slurry preferably flows into the existing network of natural fractures which are favorably oriented with respect to the direction of the maximum horizontal stress. In most cases the volume of the material received by natural fractures is limited due to limited connectivity of dilatable fractures. In these cases, the total volume received by the hydraulic fracture is more than the HF% specified by the user. This can be checked in the analysis report (Fluid_Efficiency). In cases where the natural fractures can accommodate more fluid than the pumped volume or a large fraction of the total slurry volume is designated to the hydraulic fracture development (large HF%), a priority function determines the propensity of different fractures to receive the slurry. The injection priority of different natural fractures in each time step is a function of the fracture's orientation with respect to the intermediate principal stress direction, fracture transmissivity, and its connection level. The relative importance of these factors is complicated but it may be modeled with the following function.

$$Priority = \frac{Transmissivity^M \times Orientation^N}{Connection_level^L}$$

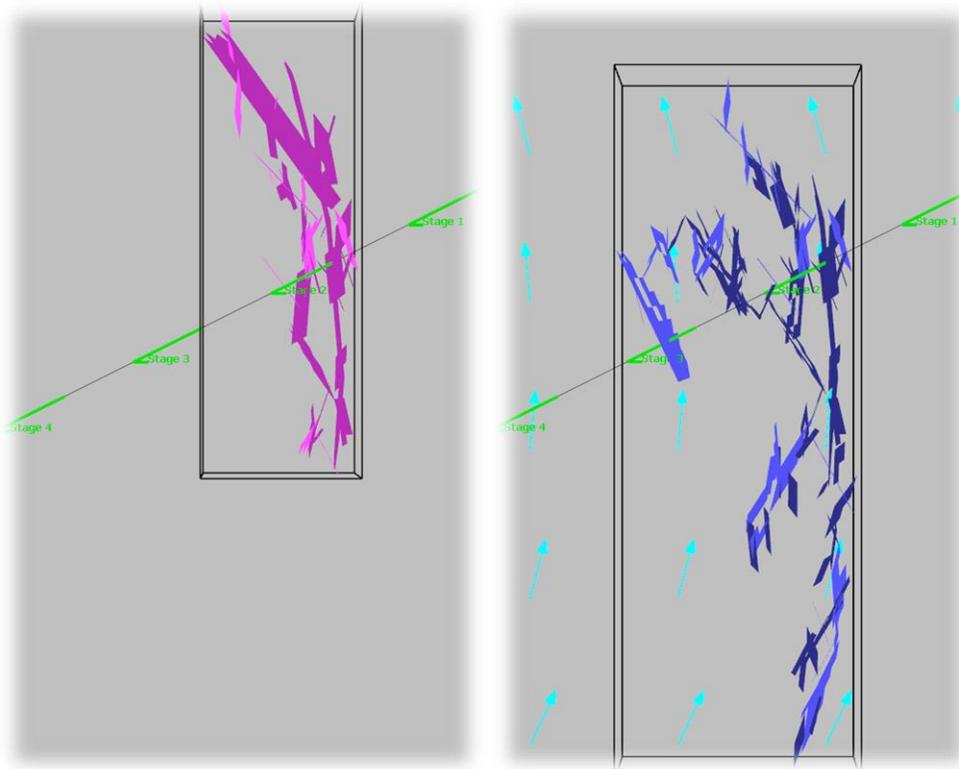
Parameter	Influencing Factor	Expected Scenario
L	Connection Level	The fractures closer to the perforation will have a higher priority
M	Transmissivity	The flow will be towards more permeable fractures

N

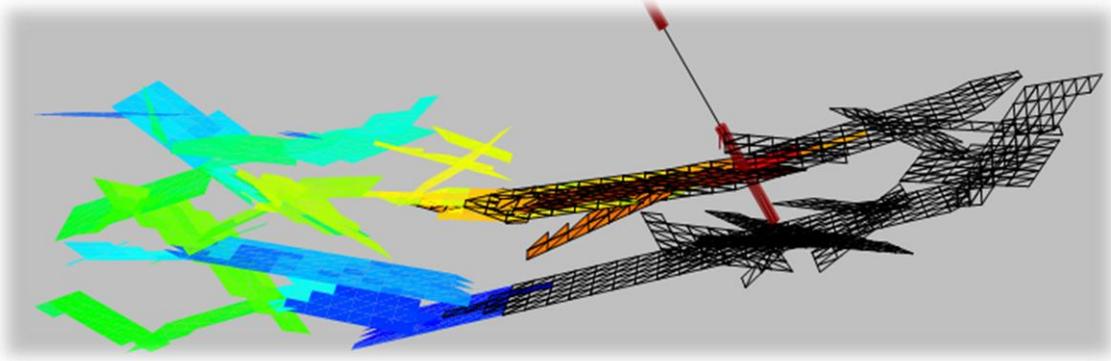
Orientation

The fractures sub-parallel to the maximum horizontal stress will have a higher priority

The flow to fracture order parameters L, M, and N provide the simulator a basis for determining which natural fracture among several connected fractures, will receive the next quantity of fluid. The natural fracture with the highest priority is injected first based on the assigned exponents. L=1 and M=N=0 results in a preference of fractures closer to the hydraulic fracture. M=1 and L=N=0 results in preference of fractures with greater transmissivity. N=1 and L=M=0 results in preference of fractures whose normals are more parallel to the minimum stress direction. It is recommended that the user change one of the parameters, M, N or L, at a time in order to evaluate the effect and sensitivity of the results to the priority function parameters.



Shown above is a hydrofracture network with a connection level priority (L=1, M=N=0, left) and with an orientation priority (N=1, L=M=0, right). In this example, a high connection level priority results in 345 injected elements in the natural fractures while a high orientation priority results in 636 elements and a smaller hydrofracture accordingly. The latter case is more probable due to requiring less energy to develop the hydrofracture.



Hydraulic fracture network with a with orientation priority (N=1, L=M=0, sequence plot) compared to the base model with connection level priority (L=1, M=N=0, wireframe). Prioritization of orientation against connection level results in increase of maximum connection level to 65 (compared to 35) and number of injectable fracture elements to 1690 (compared to 978).

The transmissivities, orientations, and the connection levels of all the natural fractures are normalized as follows between 0 and 1 before being using in the above relationship:

$$Transmissivity = \frac{\log(T_{element})}{\log(T_{max}) - \log(T_{min})}$$

$$Orientation = \cos(\theta)$$

$$Connection_level = d / d_{max}$$

where T is the transmissivity of the fractures and θ is the angle of fracture shear stress direction with respect to the intermediate principal stress direction.

Fracture Properties

Fracture properties will be changed after receiving fluid. Fracture aperture, e , for example, is enlarged when takes in hot fluid. Fracture transmissivity, T , is also changed due to the change in aperture. Not all fractures can take in fluid along the injection path, only when the fracture's pore pressure exceeds the surrounding normal stress, σ . Fracture inflation degree depends on flow distance and the fracture elasticity property that is characterized by the Poisson ratio, ν , and Young's modulus, E . See the discussion in section [Constitutive Equation and Fracture Geometry](#) for more detail. Both elasticity parameters can be prescribed as constant or spatial variables defined as grid properties. Elastic properties of the rock are used in the constitutive equation to relate the volume of dilating fractures to fracture pore pressure.

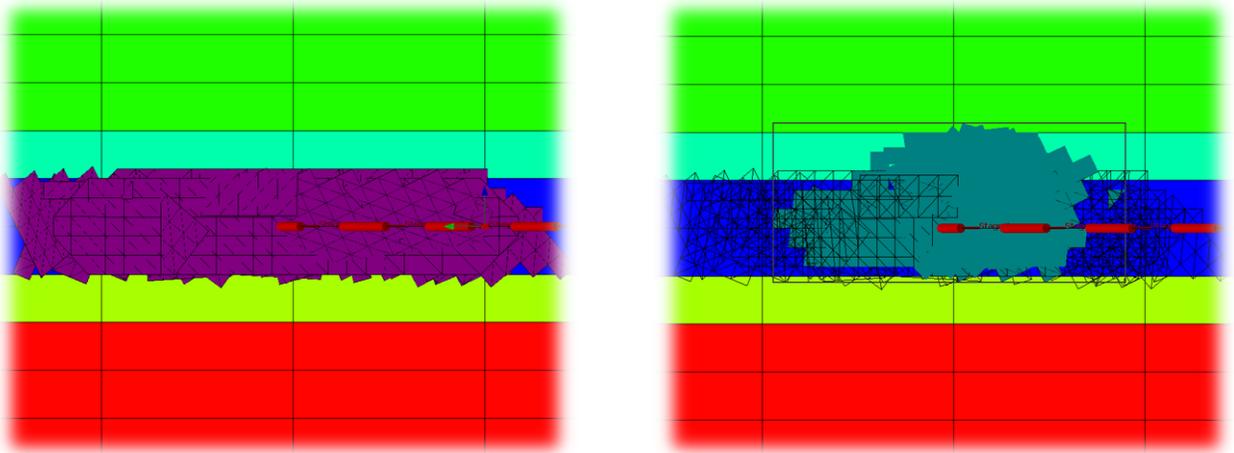
Hydraulic Fracture Definition

Hydraulic Fracture		Fracture Pore Pressure		Stage Parameters	
In Situ Stress	Fracture Properties	TFI	Hydro Shear	Output	
<p>Stress Configuration ?</p> <p><input checked="" type="radio"/> Isotropic <input type="radio"/> Orthotropic</p>					
<p>Young's Modulus E=\sim[1000-50000 MPa]</p> <p>20000000 [Pa] >></p>			<p>Poisson's Ratio ν=[0.0-0.5]</p> <p>0.25 >></p>		
<p><input type="checkbox"/> Inclusion of Quench Grid (Optional)</p> <p>0 >></p>					
<p>Aperture Limit</p> <p>Maximum Aperture for Inflated Fractures</p> <p>10 [m] >></p>					
<p>Control Frac Growth</p> <p><input type="checkbox"/> Stop Inflation of Natural Fractures</p> <p>Ratio To Stop Growth 5 >></p>					
<p><input type="checkbox"/> Update Permeability (P) from Post-Fracture Aperture:</p> <p>$P = A + B * \text{Aperture}^C$ A: 0 [m] >> B: 0 >> C: 0 >> <input type="checkbox"/></p>					
				<p>Cancel Apply</p>	

Max. Aperture: This option allows the user to define an upper limit for the aperture size of hydraulic fractures. It prevents having too large an aperture size due to sudden change in stress values or an unrealistic stress grid model.

The bounds of the hydraulic fractures are based on field-dependent structural constraints such as faults or lithostratigraphy. Stress contrast between layers could potentially constrain fractures in lower stress zone as the hydraulic fracture network develops only where the slurry

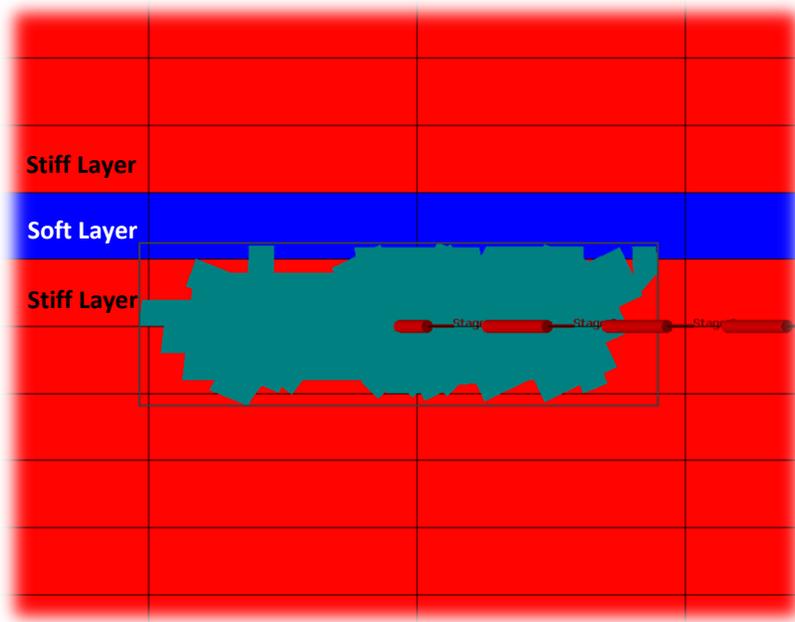
(fluid) pressure exceeds the normal stress on a fracture. In the figures below, the hydraulically stimulated fracture network is bounded by grid zones with lower stress levels.



Hydraulically-stimulated fractures are bounded to low stress zones. Increasing the pumping pressure from 4200 psi (base model, left image) to 4800 psi (right image) extends the network of hydraulic fractures into the highly-stressed layers.

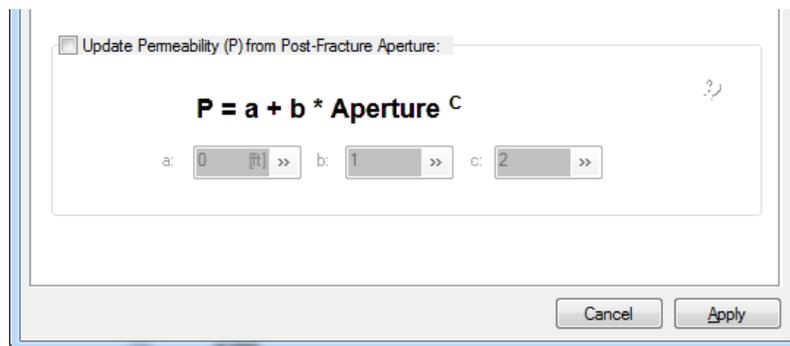
Ratio to Stop Growth : Propagation of a fracture in a geological layer can be restrained if the layer is adjacent to a much softer one (Teufel and Clark 1981³). FracMan can confine the hydraulic fractures to the stiff zones through specifying the **Ratio to Stop Growth** when the elastic modulus is defined as a grid property. The specified **Ratio to Stop Growth** (>1) is the ratio of high to low Elasticity so it is always greater than 1. To confine the stimulated natural fractures to the stiff layer, the **Stop Growth on Natural Fractures** option must be selected. The hydro-fracture grows in the higher Young's Modulus grid cells and stops growing at the cell boundary where E has a sudden drop.

³ Teufel, L. W., and J. A. Clark. "Hydraulic fracture propagation in layered rock: experimental studies of fracture containment, SPE/DOE 9878, Procs." *Low Perm. Symp., Denver, CO*. 1981.



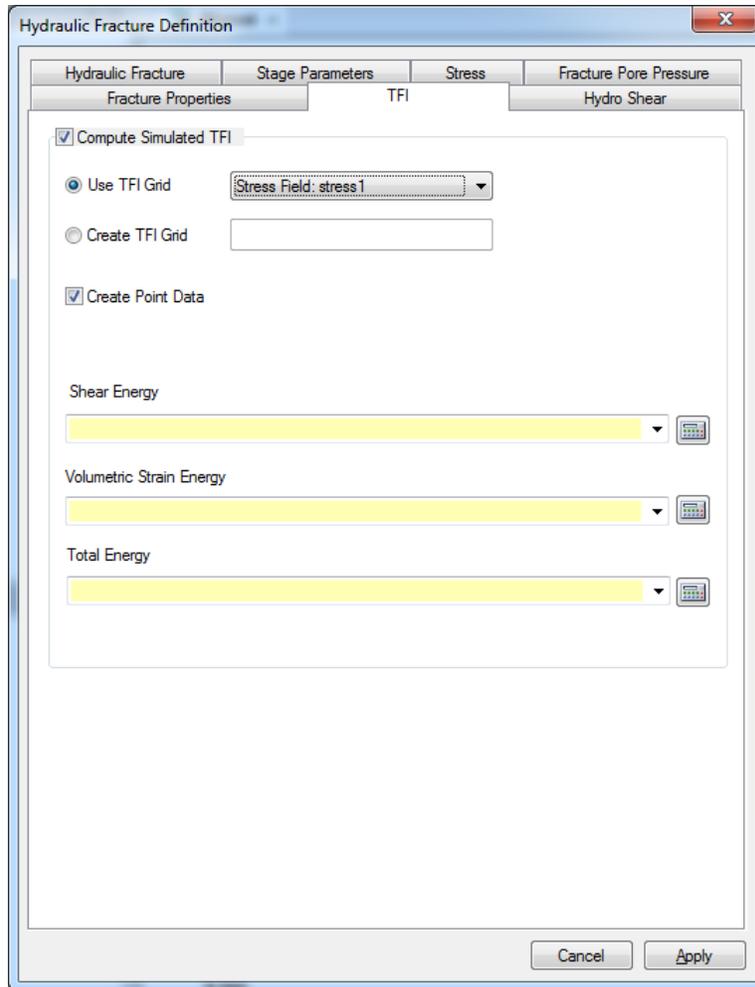
An example of constraining the vertical growth of the hydraulic fracture network by interposing a stiffness contrast to the reservoir rock. The blue grid cells contain softer rock.

The changed fracture properties are saved in the new fracture sets containing a new hydrofracture and all existing fractures in the flow path. If the “**Update Transmissivity (T) from Post-Fracture Aperture**” box is checked (image below), then the fracture transmissivity or permeability will be updated to the original fractures based on the user-input parameters **a**, **b** and **c**.



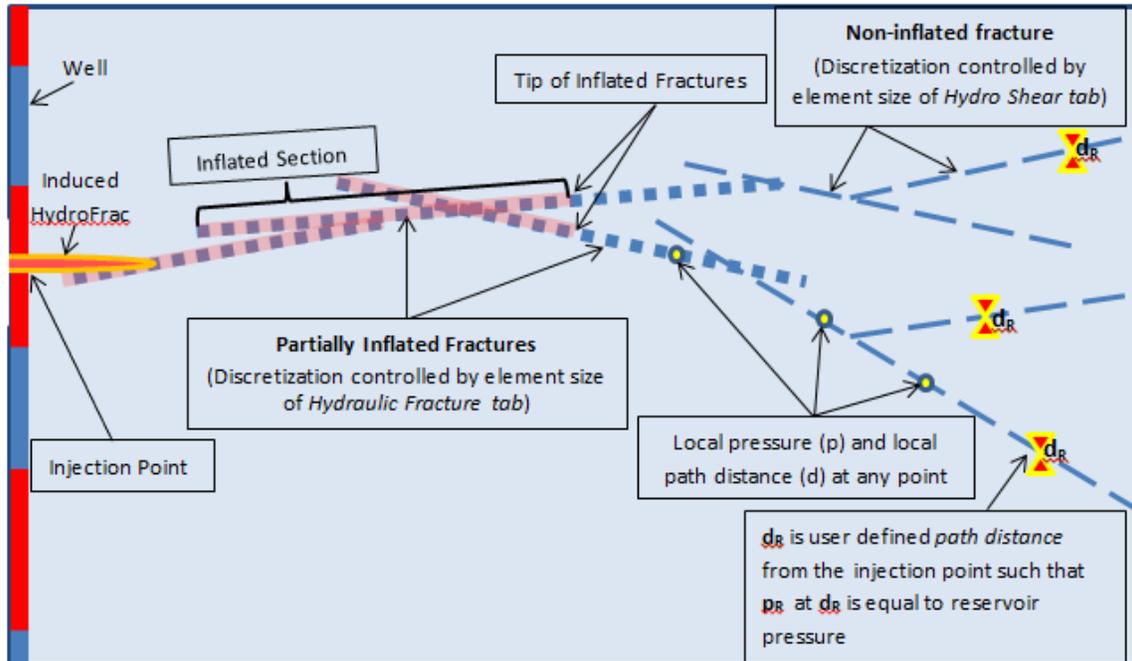
TFI

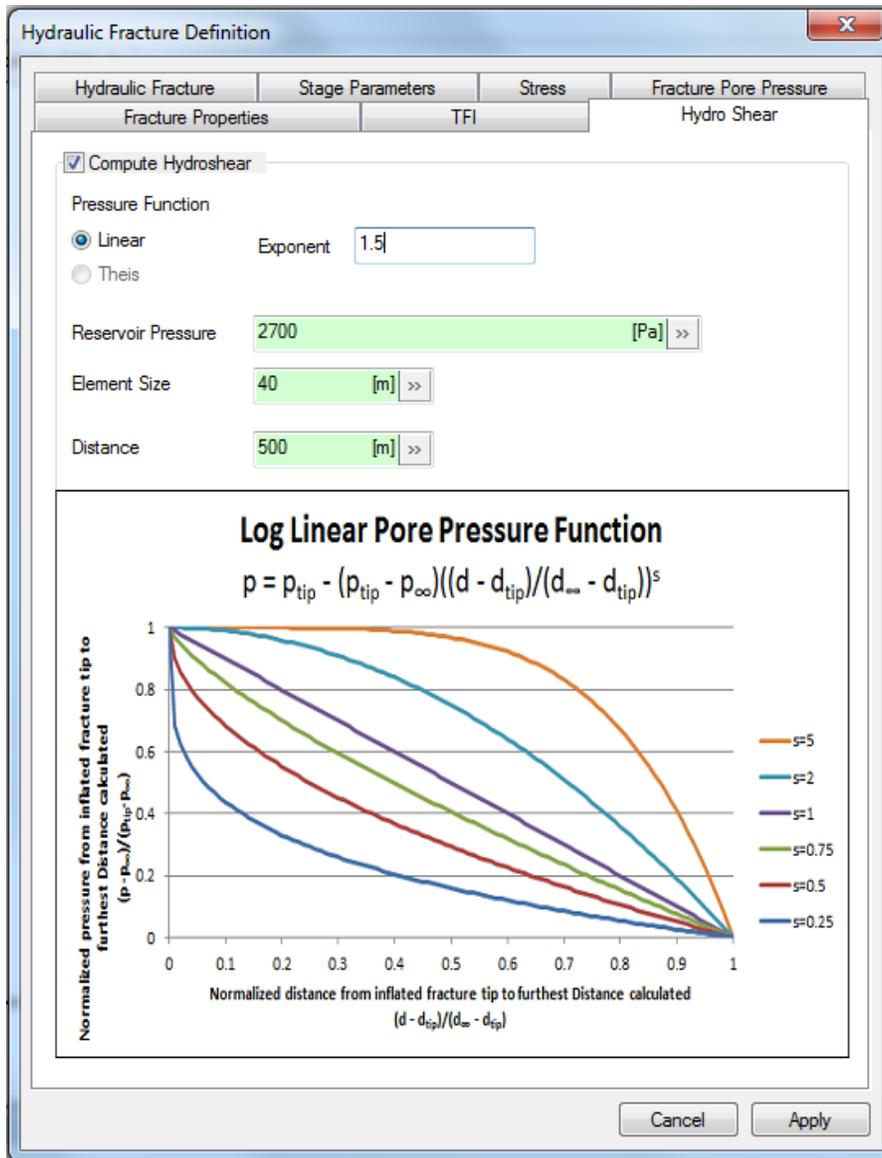
The Tomographic Fracture Imaging (TFI) tab allows the user to specify custom formulas for generating simulated TFI results. This feature is under active development and if the user wants to utilize this feature, they are encouraged to contact FracMan technical support for assistance.



Hydro Shear

After hydraulic fracture simulation is complete, the hydro shear portion of the calculation is run. The pressure at the tips of the inflated fractures is used to calculate the pressure dissipation along other connected fractures which did not inflate, but may experience enough shear stress to slip.





Compute Hydroshear: Check this box if you want to include the hydro-shearing calculations following the hydraulic fracture simulation.

Pressure Function: The user must specify how quickly the pore pressure at the tip of the inflated fracture dissipates back down to the unperturbed reservoir pressure. This is done by choosing the exponent, s , of the decay function:

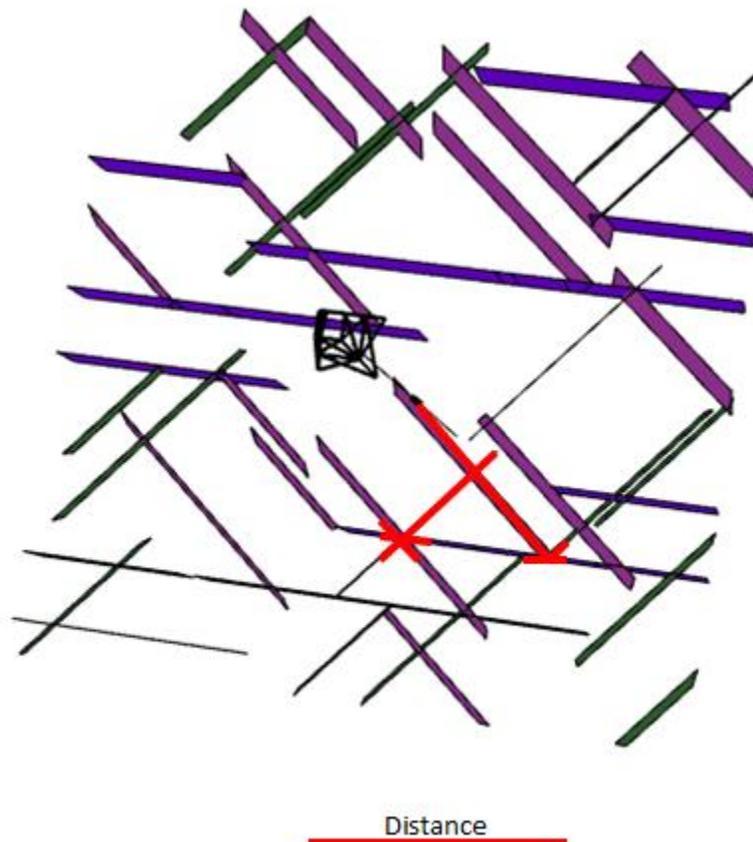
$$\frac{p_{tip} - p}{p_{tip} - p_R} = \left(\frac{d - d_{tip}}{d_R - d_{tip}} \right)^s$$

Where p_{tip} is the pore pressure taken from the tip of the inflated fracture, d_{tip} is the path distance from the injection point to the tip of the inflated fracture, p is the local pressure at any point along the fracture path, d is the path distance from injection point to the local point

(beyond the tip of inflated fracture), p_R is the Reservoir Pressure at d_R and d_R is the path distance from the injection point supplied by the user where the pore pressure is unperturbed by the hydro-fracturing and is equal to reservoir pressure.

Element Size: For inflated and partially-inflated fractures that have already been discretized for the hydraulic fracturing phase, the element size from the Hydraulic Fracture tab has already been applied. If the distance specified by the user on hydro shear tab exceeds the path length of these stimulated fractures, further discretization occurs using Element Size from Hydro Shear tab. It is generally recommended that it be equal to or greater than the element size used for the Hydraulic Fracture phase.

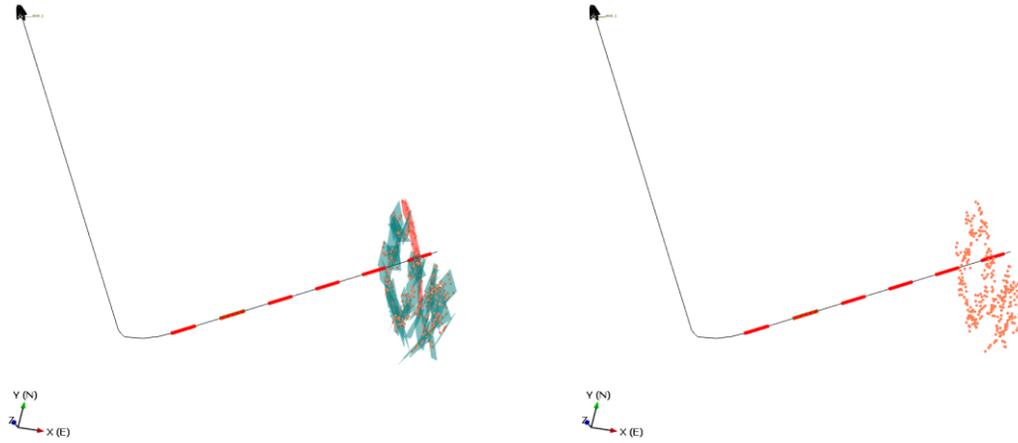
Distance: The Distance is the *path distance* along the fractures where the pressure perturbation dissipates to zero. It needs to be short enough to fit in the model but long enough to account for micro seismicity recorded at distance.



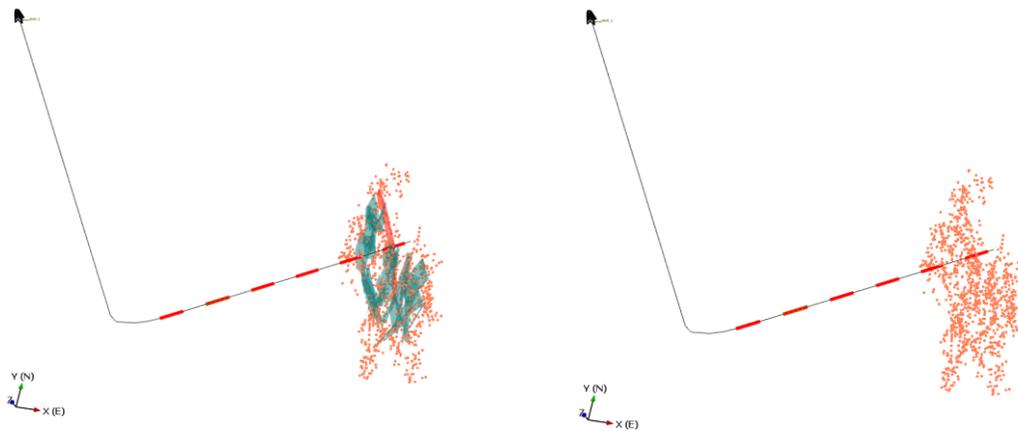
The Distance field controls how far along the flow path the pressure is perturbed and how many fractures will be included in the Hydro Shearing analysis. Only fractures touching the red line will be included with the Distance field set to the value shown above.

Hydro Shear Calculation Output – points along the fractures located within the path Distance that experience a $\text{ShearStressMagnitude} > \text{ShearStrength}$ are Critically_Stressed and will be

saved as points in the Microseismic object. Fractures containing points that are critically stressed that have not already been included in an Induced or Inflated reference set are added to a hydrosheared reference set saved with the Analysis objects. Note that fractures in this set include the entire fracture, not just the portion that may be hydrosheared.



Simulated hydraulic fracturing only along with associated microseismic events.



Simulated hydraulic fracturing along with hydroshear effects and associated microseismic events. Microseismic cloud in this case is larger due to capture of hydroshearing.

Output Options

Hydraulic Fracture Simulation includes a number of optional output options

Hydraulic Fracture Definition

Hydraulic Fracture	Fracture Pore Pressure	Stage Parameters
In Situ Stress	Fracture Properties	TFI
		Hydro Shear
		Output

Perforation Friction

Perforation Diameter: 0.05 [m] >> Interval Property: []

Perf Angular Spacing: 360 [deg] >> []

Perf Discharge Coef: 0.9 >>

Tubing Friction

Friction Coefficient: 4524.12 [Pa/m] >>

Tributary Drainage volume

Compute Convex Hull

Fracture Slab

Drainage Distance: 1 [m] >>

Grid Cell Size: 10 [m] >>

Individual Stages

All Stages Together

Cancel Apply

Perforation Friction

These are optional values used in the display of the resulting time vs pressure display.

Tubing Friction

These are optional values used in the display of the resulting time vs pressure display.

Tributary Drainage Volume

FracMan can automatically compute tributary drainage from both individual stage as well as entire well induced and activated fractures.

Hydraulic Fracture Analysis Output

Output includes a detailed report, hydrofracture and natural fractures that received fluid, microseismic points, hydro-sheared fractures (if computed) and non-stimulated fractures. Review of the analysis report is a necessary step for evaluation, modification and confirmation of the hydraulic fracturing analysis. The user can review and export the details of input and output of the analysis.

The volume balance of the injected slurry and the expanded natural fractures and hydraulic fracture indicates sufficient number of time steps. An in-balance of 1-2% can be considered acceptable.

In the absence of dilatant natural fractures, the relative volume of induced hydraulic fracture will be large compared to that of the connected inflated fractures.

The report includes the values of the important input parameters, meshing information, total number of fractures, number of meshed fractures (# of built-up fracture connection lists), number of elements, and number of injectable fracture elements.

Flow distances of hydrofracture and natural fractures indicate the extent of stimulated volume.

A summary of the results of each time steps including the pumped volume, and fracture network volume is reported.

In the summary of connection levels, the fractures activated in each connection level are reported in the two categories of hydraulic fracture and natural fractures.

A detailed report of the stimulated elements can be included if the user selects the option for the “Full report for every injected elements”. In this report includes the sequence number, element ID, connection level, pressure in the fracture, normal stress on the fracture, resulting apertures, element area, volume and location.

For each of the dilated fractures, the normal stress, pore pressure, sequence number, and storage aperture are calculated and set as new fracture properties which can be visualized or statistically analyzed. Also the “critical stress” and “critically stressed” properties are defined after the critical stress analysis. The “critical stress” property of the points is the ratio of shear stress on the fracture to its shear strength. Therefore, if the value of the critical stress property is greater than one, then the point is critically stressed. The “critically stressed” property is set to one if the point is critically stressed, otherwise it is set to zero.

Why Volumes Received can be Greater Than Volume Pumped: It is often the case that the total volume received by the induced and inflated fractures exceeds the volume pumped. This can be a function of the discretization size of the fractures – either an element inflates or does not (no partial inflation). In this case, decreasing the element size in the hydraulic fracture definition can make the volumes closer in size. Another reason is due to the gradual expansion of an elliptic shaped network as each element gets injected. Each previously expanded element gets enlarged as the network increases, so injection to the last element causes all the previous elements to expand a lot and the mass balance can be off as a result.

HydroFrac_1_Stage_and_HS_Stats_1	Analysis Parameters	Treatment Schedule	HydroFrac_1_Stage_and_HS_Stats_1		
Hydraulic Fracture	HydroFrac_1_Stage_and_HS				
Stage_1 Induced	Value / Mean	Min	Max	Dev	Count
Storage Aperture[m]	0.011720	0.011720	0.013875	0.001530	2245.000000
Total Frac Area [m ²]	897999.987640				
Total Frac Volume [m ³]	8266.234421				
Frac Height [m]	0.000000				
Frac Length [m]	480.832623				
Well Distance [m]	240.416312	240.416312	240.416327	0.000022	2.000000
Frac Aspect Ratio [Height / Length]	1.#INF00				
Box Volume [m ³]	0.000000				
Pore Pressure[MPa]	78.250000	78.250000	78.250000	0.000000	2245.000000
Normal Stress [MPa]	78.050000	78.050000	78.050000	0.000000	2245.000000
Shear Stress [MPa]	0.000001	0.000001	0.000001	0.000000	2245.000000
Frac Direction (tr,pl)	225.000, 0.000				
Stage_1 Inflated	Value / Mean	Min	Max	Dev	Count
Total Frac Area [m ²]	0.000000				
Total Frac Volume [m ³]	0.000000				
Frac Height [m]	0.000000				
Frac Length [m]	0.000000				
Well Distance [m]	240.416312	240.416312	240.416327	0.000022	2.000000
Frac Aspect Ratio [Height / Length]	-1.#IND00				
Box Volume [m ³]	0.000000				
Stage_1 HydroSheared	Value / Mean	Min	Max	Dev	Count
Total Frac Area [m ²]	1403036787.924492				
Frac Height [m]	999.994250				
Frac Length [m]	999.992362				
Well Distance [m]	499.996181	499.996181	500.002848	0.009429	2.000000
Frac Aspect Ratio [Height / Length]	0.999998				
Box Volume [m ³]	499965691.577437				
Pore Pressure[MPa]	0.000072	0.000072	0.000078	0.000003	2778731.000000
Normal Stress [MPa]	0.000077	0.000077	0.000100	0.000007	2778731.000000
Shear Stress [MPa]	0.000007	0.000007	0.000014	0.000003	2778731.000000
Frac Direction (tr,pl)	180.002, -0.001				

Output report

FracStress Geomechanics

FracStress calculates stress in a grid based on various inputs such as boundary conditions, material properties, and a strain field output from dynamic upscaling ([Oda Strain](#)). The FracStress definition's property pages allow the user to specify the various input and output options as well as some options controlling how FracStress performs the calculations. There are seven property pages in all: Solution, Material Properties, Fixities, Far-Field Stress, Boundary Conditions, DFN-FEM Coupling, and Elastic Properties Update. Details of each of these can be found below.

FracStress uses a finite element model (FEM) with nodes corresponding to the vertices of the grid cells, and tetrahedral elements (several per grid cell). Each grid cell has material properties such as Young's modulus as specified by the user (for example using a spatial property on the grid), and these material properties are shared by all the tetrahedral elements that make up the cell.

Forces are applied to the nodes. For example the user can specify boundary conditions to apply stress to the outer surface of the grid, and FracStress converts these into forces on the corresponding outer nodes. It then runs the finite element model which calculates the displacements of the nodes for which the system is in equilibrium. FracStress then calculates the strain and stress in each grid cell from these

displacements. The stress is output as cell properties for the grid, and there are also options for outputting strain and displacement values in a similar fashion.

FracStress

Boundary Conditions DFN-FEM Coupling Elastic Properties Update

Solution Material Properties Fixities Far-Field Stress

Definition

Name: FracStress_1

Description:

Solution Grid

BasicGrid

Stress Averaging Method: Minimum vertex force difference

Solution Options

- Distribute resultant force on external vertices
- Output Principal Stresses
- Output Cartesian Direct Stresses
- Output Stress Invariants
- Output Boundary Forces
- Output Displacements
- Output Strain

Use Boundary Conditions instead of Far-Field Stress

Boundary Assignments Available

Regions

- RegionBox - C
- RegionBox - T
- RegionBox - B

Calculation

Convergence criterion (epsilon): 1e-006

Cancel Apply

The Solution property page allows the user to specify a name and description for the FracStress definition, and the all-important grid which is used for both input and output values, and also for creating the elements and nodes of the model. A grid with a large number of small cells allows for more detailed input and output, but will also increase the time taken to carry out the calculations.

The Stress Averaging Method allows the user to control the method used at the end of the calculation, to combine the stresses within the tetrahedral elements that make up a cell. The first three options 'Mean', 'Volume weighted mean' and 'Minimum vertex force difference' work on each grid cell in isolation. But the fourth option 'Smoothing' calculates the average stress on each node based on all the elements that include that node; and then calculates an average for each cell based on the vertices of that cell. So in that case the stress in a cell depends on the stress within a wider range of elements (not just the ones within the cell).

The 'Smoothing' option is intended for use when there is a strain input specified on the DFN-FEM Coupling property page: in other circumstances it is best avoided because it reduces the detail available in the results and may lead to slight anomalies in the strain outputs (if these are selected).

'Mean' just calculates the sum of the stresses in the tetrahedral elements (in symmetric matrix format) and divides by the number of tetrahedral elements in the cell.

'Volume Weighted Mean' is similar, but weights the stress in each tetrahedral element by the volume of the tetrahedron.

'Minimum Vertex Force Difference' calculates the sum of the forces on the vertices of the cell due to the stress in the cell's tetrahedral elements, and then calculates the stress for the whole cell that would most closely replicate those forces.

If 'Distribute resultant force on external vertices' is checked FracStress makes the resultant force applied to all nodes zero by adjusting the forces on outer nodes in an even fashion. This may be useful in circumstances where the model fails to converge, though it is not normally necessary to use this option.

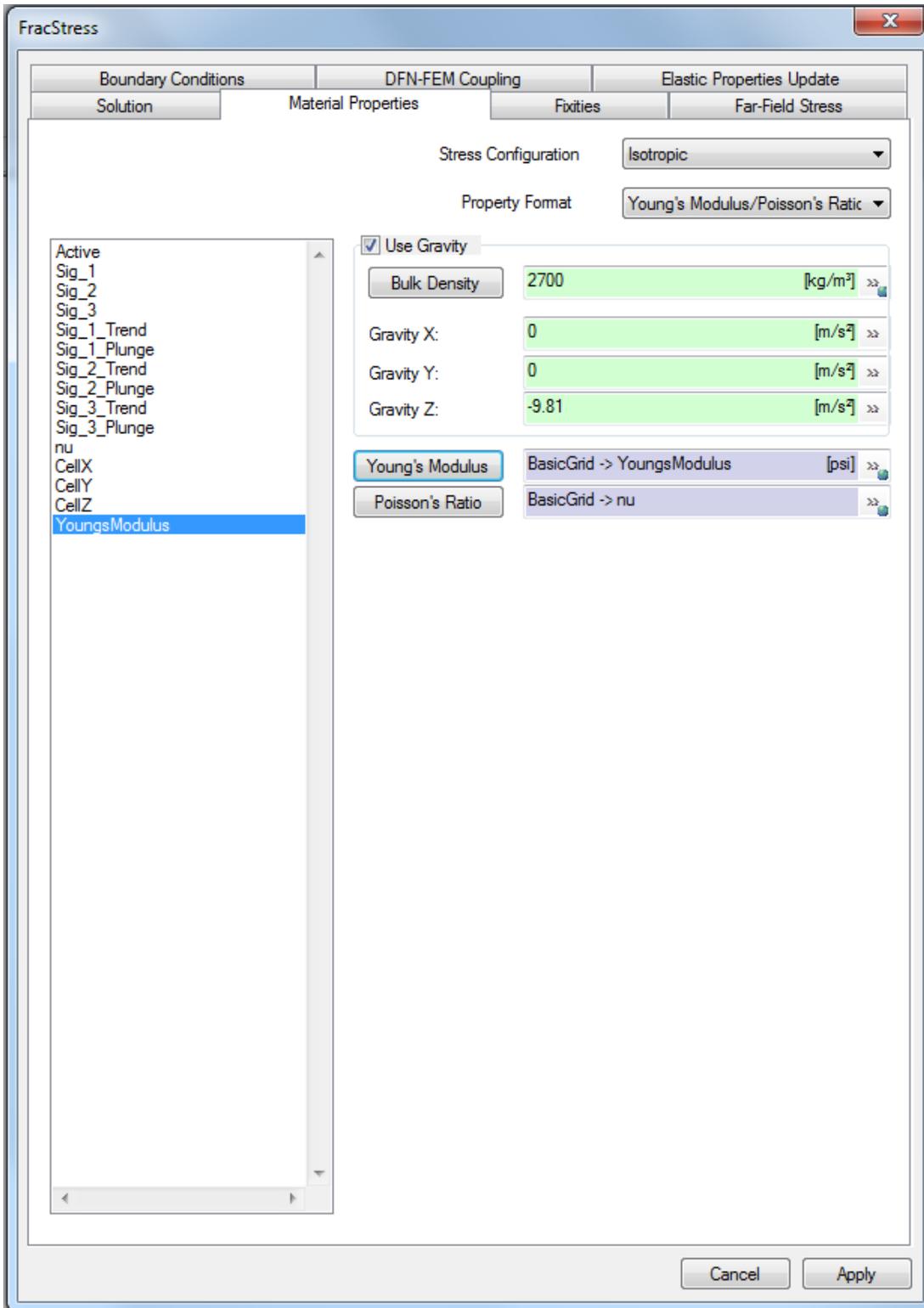
The next three check boxes control the format of the output stress. You can select any number of the three formats available. These are to output the stress tensors as principal values and directions, as a symmetric matrix, or as the following stress invariants:

- Von Mises stress as $\sqrt{\frac{1}{2}[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]}$
- Tresca stress as $\frac{1}{2} \max(|\sigma_1 - \sigma_2|, |\sigma_2 - \sigma_3|, |\sigma_3 - \sigma_1|)$
- Peak stress difference as $\sigma_1 - \sigma_3$
- Pressure as $\frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3)$

There are three check boxes for selecting which of the optional outputs are required: 'Output Boundary Forces' allows the user to see the forces that FracStress has calculated on the nodes as inputs to the FEM (these are vertex properties for the grid). 'Output Displacements' and 'Output Strain' create cell properties. Note that the strain tensor will be output in principal and/or symmetric matrix format depending on which of these formats are selected for stress output.

Next the user can select whether to use boundary conditions instead of Far-Field stress to specify vertex forces. This determines which of the Far-Field and Boundary Conditions property pages are available for further input. If using boundary conditions the user selects the surfaces and region boxes that will be used to apply boundary conditions. It is important not to use overlapping surfaces with different boundary conditions as this makes the boundary condition to apply ambiguous.

The calculation options currently just has a single setting which controls when the solution is considered to have converged. The default setting of 1e-6 gives about as much accuracy as you can expect to get from the solution. Normally this is fine, but if your model doesn't converge with this setting you can use a larger value. While the FEM is running, the progress message shows you the minimum value for the convergence parameter that would have given convergence by the current iteration. So this can help you to choose a larger value for the convergence parameter in cases where the solution fails to converge within the maximum number of iterations available. Values in the range 1e-6 to 0.01 should be OK, but the smaller the value the more accurate the results will be.



The Material Properties page allows you to specify the bulk density and elastic properties of the rock. If you wish to use spatial properties these must be cell properties of the grid which you have selected on

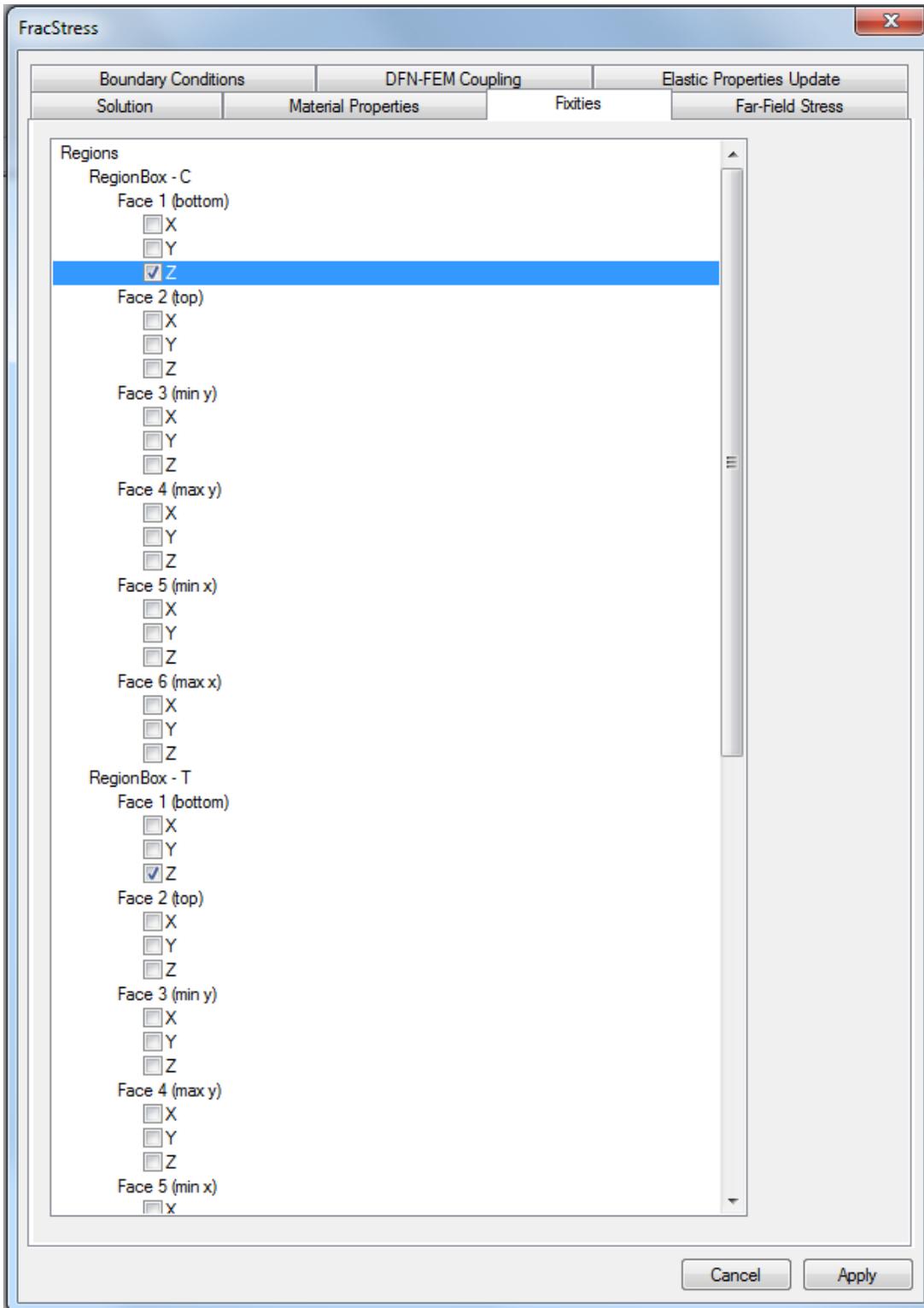
the Solutions property page. For convenience this grid's cell properties are listed on the left of the page, and you can select one of these and click one of the buttons to assign the selected property to one of the available fields.

Bulk density is only currently used to apply forces representing the weight of the rock, and this use is optional: if you want to apply gravity check the 'Use Gravity' check box, and specify the bulk density and acceleration due to gravity. The latter can have x, y and z components. The Gravity Z figure is applied in the positive z direction – in other words upwards. So you should provide a negative value such as -9.81 m/s^2 . The use of gravity is typically combined with fixities in order to ensure that the solution converges.

For the elastic properties there is a choice of Isotropic, Orthotropic or Anisotropic symmetry each of which currently gives a single option for the format in which the elastic properties are specified (further formats may be added in future). For example if you select 'Isotropic' which is the simplest option, then you will need to specify a single Young's modulus value and a single Poisson ratio value. The buttons and fields visible beneath change according to your selection.

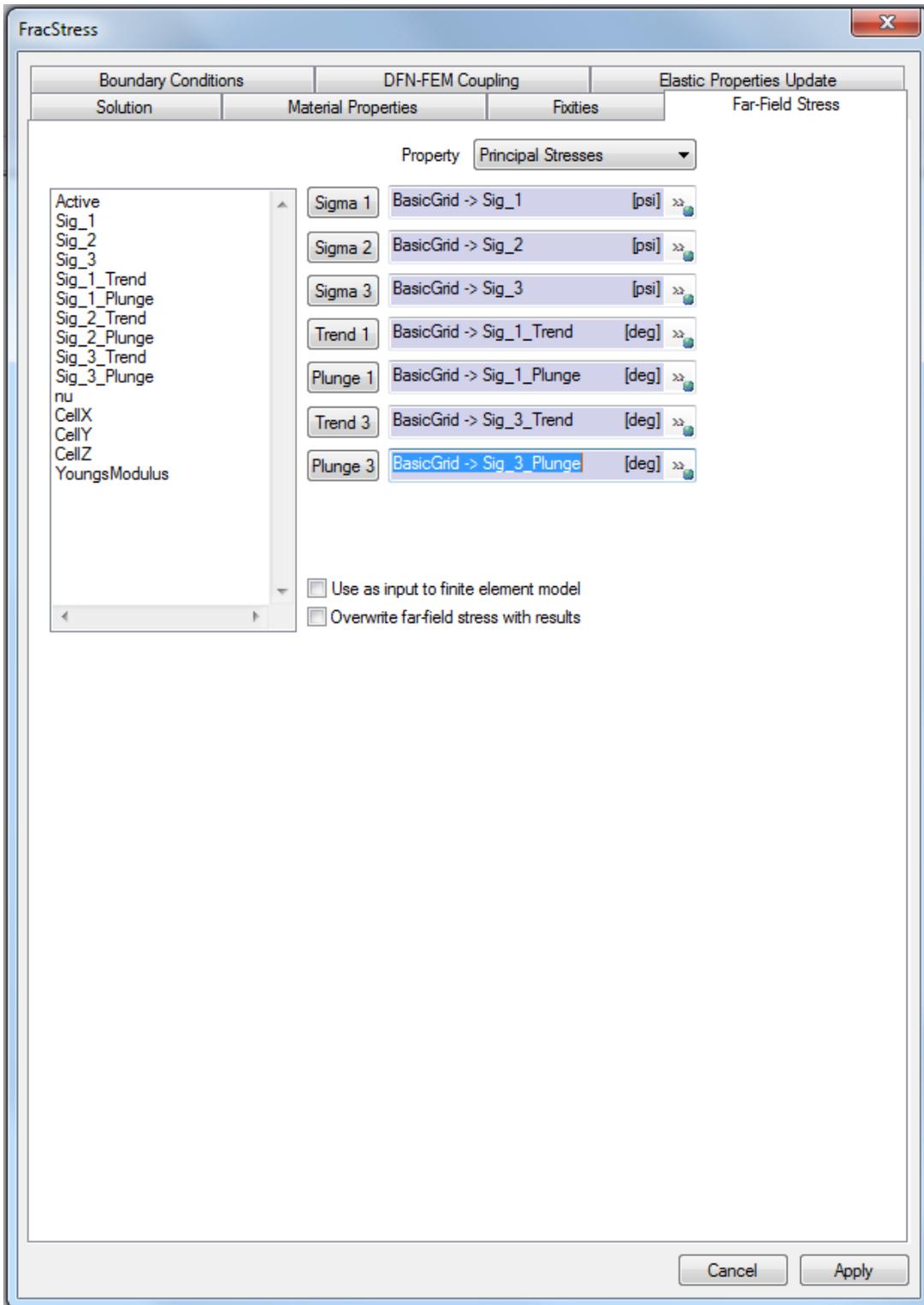
For orthotropic materials you need to specify three values for Young's modulus, three values for Poisson ratio and three values for shear modulus, together with a pair of trend and plunge values that specify the axes of symmetry.

For anisotropic materials you need to specify all 21 components of the stiffness matrix. This is mainly of use if your material properties have come from another calculation such as Dynamic Upscaling/Oda Geomechanics. By using the full stiffness matrix you avoid the necessity of projecting this onto a lower symmetry and the associated loss of accuracy.



The 'Fixities' page allows you to fix any combination of the X, Y and Z coordinates of vertices of the grid. To do this you create suitable surfaces that cover the vertices you want to fix, and then you can just click

on X, Y or Z under the surface name to fix the corresponding coordinates. By using multiple surfaces you can apply different restrictions to different vertices. As well as surfaces you can also use region boxes, in which case you apply the fixed coordinates to one or more of the faces of the region box. This provides the easiest way of applying the same restrictions to all the vertices on an outer face of the grid. In the example above the z coordinates of all the vertices on the bottom face of the region boxes are being fixed. This might be for example be to ensure that the solution converges when using gravity.

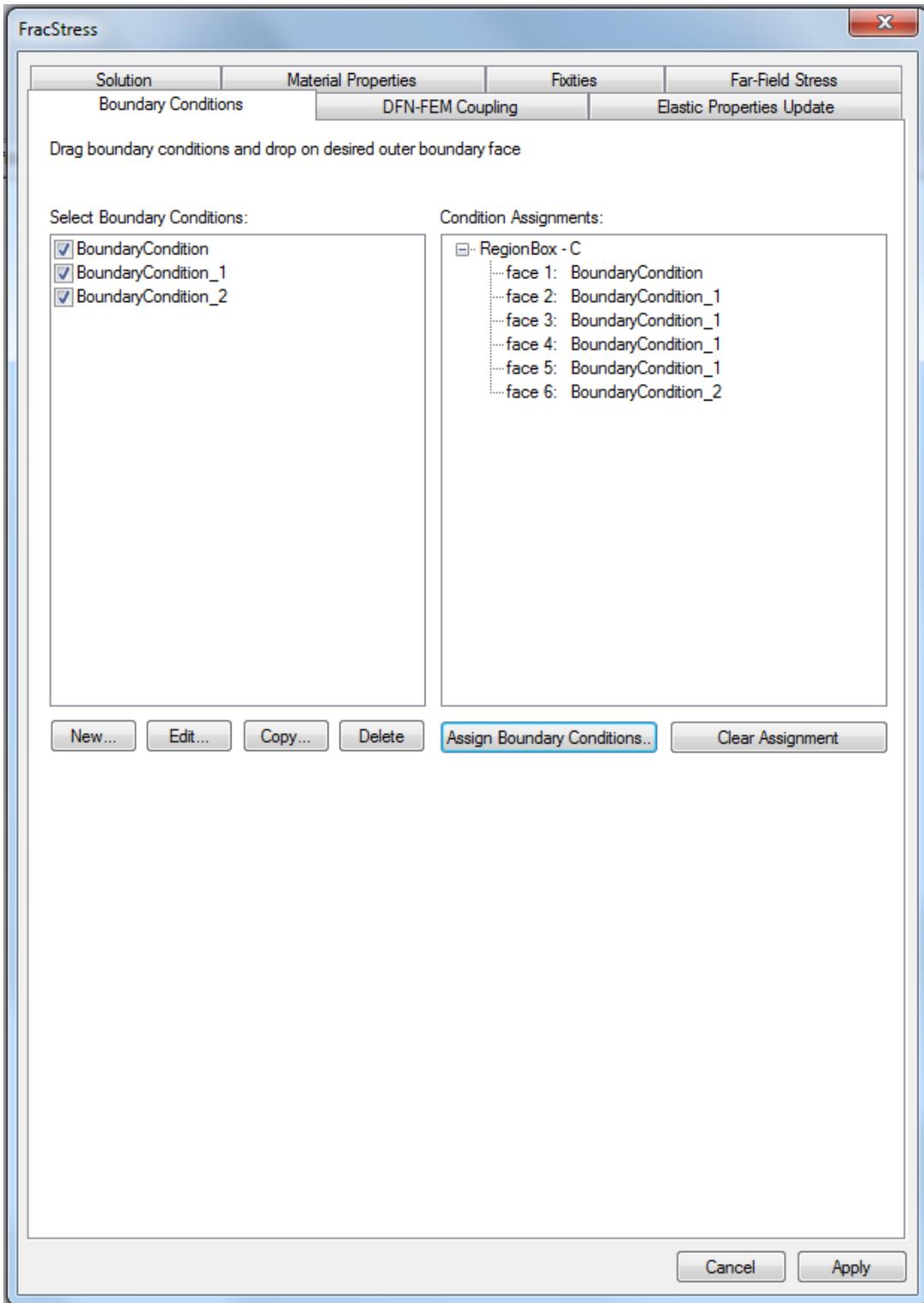


The Far-Field Stress page allows you to specify the initial stress within the grid. The controls on this page will be disabled if you have elected to use Boundary Conditions instead. Far-Field stress can be used in

two alternative ways: if 'Use as input to the finite element model' is checked then it is used to calculate the force to apply to nodes in the finite element model; otherwise it is merely a stress field to which the calculated stress is added before output. If you are not using a strain input, then it only really makes sense to use the Far-Field stress as an input – otherwise you will end up not applying any forces to the nodes, so the calculated stress will be zero and the output stress will be identical to the Far-Field stress.

You can specify Far-Field stress either as principal stresses and directions; as a symmetric matrix (Cartesian Stresses); or as a pressure gradient. If using spatial properties, these must be cell properties of the grid selected on the Solution page. A list of this grid's cell properties appears on the left of the page, and buttons are provided for assigning the selected property to one of the required fields. If using a pressure gradient, you can select a surface object in your model for the 'Ground Surface'.

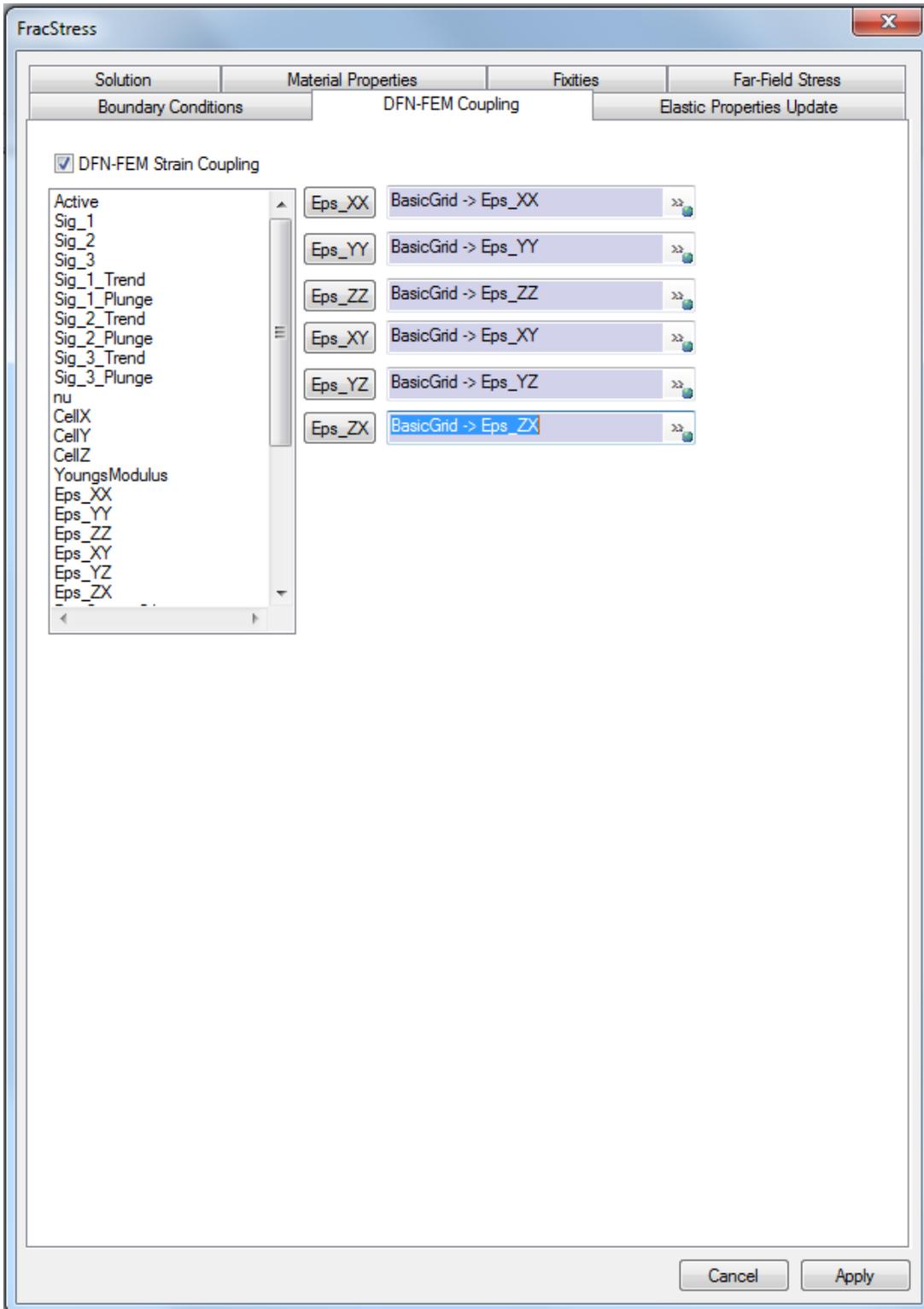
You can also elect to overwrite the Far-Field stress properties with the stress results using the 'Overwrite far-Field stress with results'. For this to work, Far-Field stress must not be a pressure gradient, and all the stress fields must be spatial. In that case this option will override the stress output format selected on the Solution page.



The Boundary Conditions property page is used to specify stresses to apply to the boundary surfaces and regions selected on the Solutions page. It is only available if you have elected to use boundary conditions

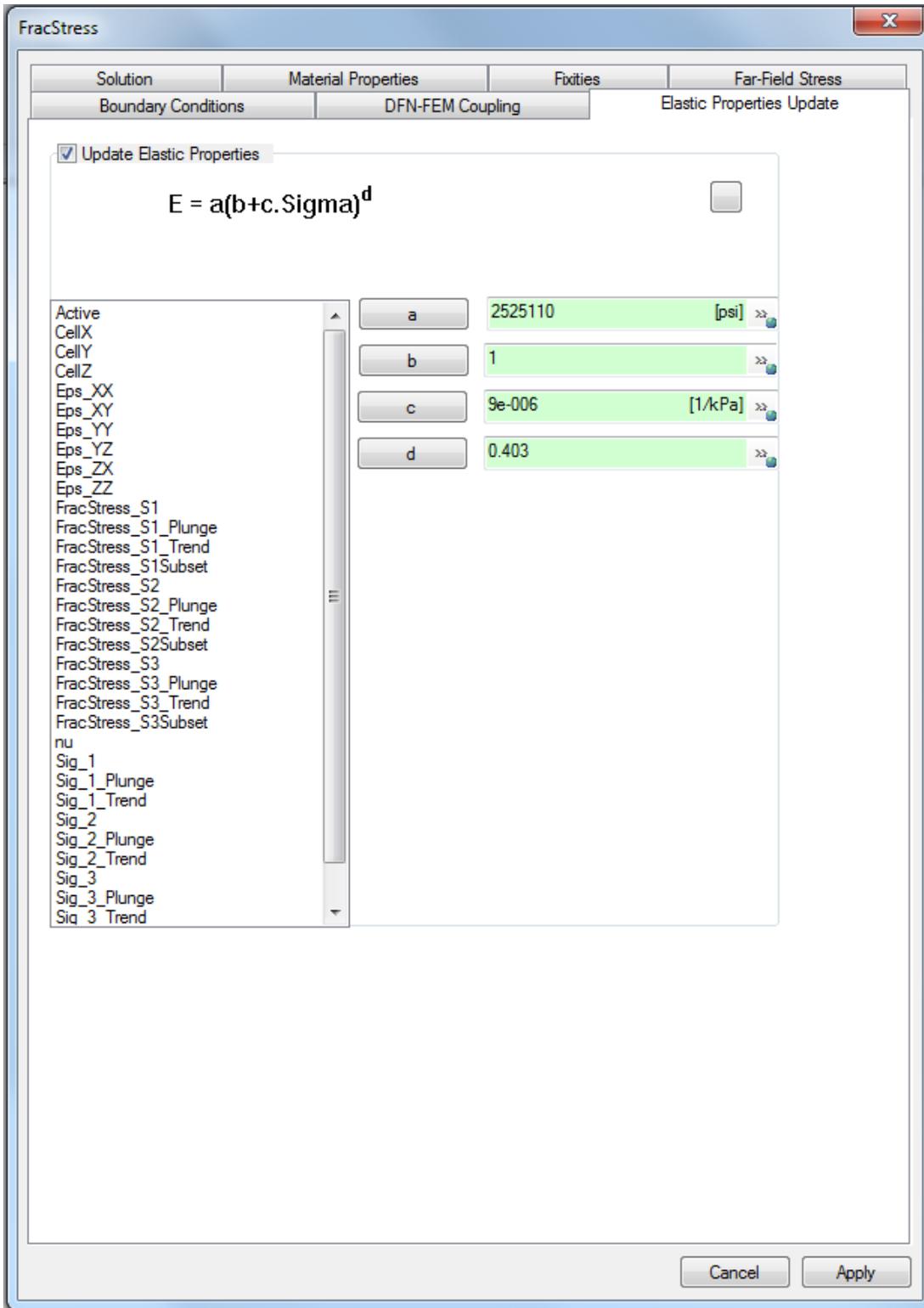
instead of Far-Field stress on the solutions page. The boundary conditions are essentially just constant stress tensors – you can specify them in principal or symmetric matrix formats, pressure gradients and constant pressure. You cannot use spatial properties. Having created one or more boundary conditions using the ‘New’ button, you can assign boundary conditions to surfaces (and region faces) by dragging them across the page.

The ‘Assign Boundary Conditions’ button takes you to another similar screen with the added advantage of an image showing you which surface or region face you have selected.



The DFN-FEM coupling page allows you optionally to specify a strain input. This is used to calculate forces on the nodes in addition to forces resulting from any Far-Field stress or boundary conditions specified.

Normally this strain input comes from the output of Dynamic Upscaling/Oda Strain which calculates a strain field corresponding to changes in aperture of fractures intersecting a grid cell. The strain field must be in symmetric matrix format, and use face properties of the grid selected on the Solutions page. There is a list of this grid's properties on the left of the page, and buttons are provided to assign the selected property to one of the required fields.



The Elastic Properties Update page optionally allows you to calculate and output grid properties indicating the change in material properties due to the change in stress. The format used for the output is the same

as the format selected for the input material properties on the material properties page. However at present the anisotropic version isn't supported (it outputs a zero stiffness matrix) and the orthotropic version is experimental.

For the Isotropic case the formula $E = a(b+c.\text{Sigma})^d$ is used to calculate a Young's modulus corresponding to the output and Far-Field stress, and the difference between these is output as a change in Young's modulus `YoungsModulus_inc`. The value used for Sigma is the average of the three principal stresses. Note that it doesn't currently handle the case where boundary conditions are used instead of Far-Field stress. Also note that the output `Poissons_inc` is actually just a copy of the input Poisson ratio, and not an increment as the name suggests.

The four parameters a, b, c and d can be specified on this screen either as constant values or as spatial properties of the grid specified on the solution page. In order to make the formula consistent with regards to units, b and d are unitless, a has a value type of pressure and c has a value type of compressibility.

Grouting

Grouting is the placement of a sealing material such as neat cement or bentonite into the annular space between a well casing and the borehole created during well construction, into fractures surround tunnels and shafts or into the foundation of a dam or other large structure.

Performing a grouting simulation in FracMan requires that a model already have defined a well, one or more sets of fractures, a grid containing the initial pressure field and stress, a stress grid for definition of fracture critical stress analysis after grout injection and optionally a fracture filter for selection of fractures. To conduct the analysis, select from the file menu **Static Analysis > Geomechanics > Grouting** and enter the required parameters in the three Grouting Definition tabs.

Grouting Algorithm

Enter input parameters: fracture network, injection well, rheological properties (yield strength, dynamic viscosity), grout grain size (critical aperture, critical transmissivity), in situ fluid pressure, pump pressure and injection volume.

Select injection method: by constant pump pressure or injection rate; select injection volume: by injection total volume or total pumping time.

Prepare injection computation: remove fractures that critical properties are smaller than grout particle size.

Calculate intersecting fractures at injection well.

Refine connecting fractures into fine elements.

Calculate in situ fluid pressure at each fracture element. If it is smaller than pump pressure, mark it.

Compute injection sequence among all direct connected elements.

Inject grout one by one element.

Repeat previous steps (5) to (7) until the injection volume or time is reached or the penetration distance is reached.

Bingham Fluid:

$$\tau_{yx} = c' + \mu(\dot{\gamma}_{yx}) \quad \text{for } |\tau_{yx}| > |c'| \quad (1)$$

$$\tau_{yx} = 0 \quad \text{for } |\tau_{yx}| < |c'| \quad (2)$$

Where c' = yield strength and μ = dynamic viscosity

Flow Rate (q) and Pressure (P) Equations:

flow of a Bingham fluid through a smooth parallel sided aperture of unit width, given in Chhabra and Richardson (1999):

$$q = \frac{b^2}{12\mu} \left(-\frac{dP}{dx} b \right) \left(1 - \frac{3}{2}\Phi + \frac{1}{2}\Phi^3 \right) \quad (3)$$

where

$$\Phi = -\frac{2c'}{\frac{dP}{dx}b} \Leftrightarrow \frac{dP}{dx} = -\frac{2c'}{\Phi b} \quad (4)$$

b is the fracture aperture and x is the grout penetration distance.

Φ in Eq(4) is obtained from α and θ below:

$$\alpha = \left(-3 - \frac{12\mu q}{b^2 c'} \right) \quad (7)$$

Then the first root of Eq. 5, which is needed to calculate the pressure gradient, is given by:

$$\Phi = 2 \left(\frac{-\alpha}{3} \right)^{0.5} \cos \left(\frac{1}{3}(\theta + \pi) \right) \quad (8)$$

And in which

$$\theta = \cos^{-1} \left(\sqrt{\frac{27}{-\alpha^3}} \right) \quad (9)$$

Computation Procedures:

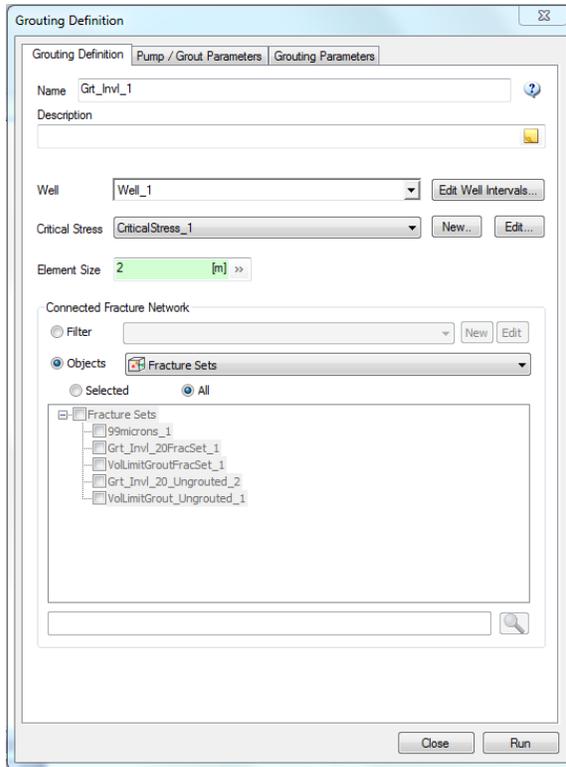
For constant rate injection:

- 1) With q , compute α in Eq. (7).
- 2) With α , compute θ in Eq. (9).
- 3) With θ , compute Φ in Eq. (8).
- 4) After Φ is obtained, using Eq.(4) the pressure drop, dP/dX , can be calculated.
- 5) The pressure drop at each injected element is calculated from steps 1). To 4).
- 6) To keep constant injection rate, q , during the grouting, the pump pressure at the inlet increases as the penetration distance goes further and further. Simulation ends at the injection volume or time is reached or until all the fractures get injected.
- 7) When multiple fractures connect to the grout front, the amount of grout fluid taken by the fracture, i , is controlled by its transmissivity, T_i value:

$$q_i = T_i / \text{sum of } \{T's\}$$

For the constant pressure pumping method the computation procedures are similar, one element at a time and calculate its pressure drop. The grouting continues until the pressure driving force is down to zero relative to the reservoir pressure when the max penetration distance is reached or until the required pumping time or volume is met.

Grouting Definition



Name: grouting definition name

Description: optional comment field

Well: select the well and well intervals

Critical Stress: optional field to indicate a critical stress definition. This is used to perform a critical stress analysis after grouting is performed. An additional output object “Stress Points” gets created under the grout result container.

Element Size: used to discretize fracture into small elements. A smaller size helps mass balance between the desired injection volume and the received grout volume but increases computational time. An element cannot take a partial amount of grout, it is either fully injected or empty.

Connected Fracture Network: select the fractures that will be included

Pump/Grout Parameters

Grouting Definition

Grouting Definition Pump / Grout Parameters Grouting Parameters

Grain Size Parameters

Critical Aperture 0.1 [m] >>

Critical Transmissivity 1e-012 [m²/s] >>

Rheologic Properties (Bingham Flow)

Yield Stress 0.00145038 [psi] >> Dyn. Viscosity 10 [cP] >>

Pump Method

Constant Pressure 0.0145038 [psi] >> Incremental dL 0.1 [m] >>

Constant Rate 8640000 [m³/d] >>

Injection Controlled by

Injection Time 0.0166667 [hrs] >>

Injected Volume 20000 [m³] >>

Initial Pressure 1.45038e-006 [psi] >>

Close Run

Grout Grain Size Parameters

- **Critical Aperture:** grout grain size defines the minimum aperture considered as open
- **Critical Transmissivity:** minimum transmissivity of a fracture considered available to grout

Grout Rheological Properties (see equations governing a Bingham fluid in the [Grouting Algorithm](#) section)

- Yield Stress: yield stress used in Equations 1 and 2
- Dyn. Viscosity: dynamic viscosity used in Equation 1

Pump Method

- **Constant Pressure:** the injection amount is controlled by pumping time or volume, whichever is satisfied first
- **Constant Rate:** either injection time or total volume is required to control grout quantity
- **Injection Time:** maximum duration of pumping
- **Injected Volume:** maximum volume of grout

Initial Pressure: constant value for initial reservoir pore pressure

Grouting Parameters

Grouting Definition

Grouting Definition Pump / Grout Parameters Grouting Parameters

Branch Selection Order:

$$Probability = Pressure^L \times Transmissivity^M \times Aperture^N$$

$$= \frac{\Delta P^L}{\Delta P_{max}^L} \times \frac{\log T - \log T_{min}^M}{\log T_{max} - \log T_{min}^M} \times \frac{\epsilon - \epsilon_{min}^N}{\epsilon_{max} - \epsilon_{min}^N}$$

L: 0 M: 0 N: 1

Fracture Aperture (e), Transmissivity (T or Kh):

Save to Original Fracture Set

Use Aperture (e) from Elastic Eq.

Poisson's Ratio, ν [0-1] >>

Young's Modulus, E [3e4-6e7 KPa] [psi] >>

Max Flow Distance (Dmax) [m] >>

$T = a + b * Aperture^c$ a: b: c:

Display full report on screen

Keep ungrouted portion of fractures in output data

Close Run

Branch Selection Order: When multiple fractures meet at the grout front, this probability rule is used to decide which fracture element (branch) gets injected first. The denominator terms are used to normalize each group into a value [0,1] so that each group has similar weight in the probability calculation. Final results are usually not too sensitive to the **L**, **M**, **N** parameters chosen.

Fracture Aperture (e), Transmissivity (T or Kh): this section allows users to modify fracture properties for fractures that receive grout fluid. The revised properties are used in the critical stress analysis that may be performed after the grouting simulation.

Save to Original Fracture Set: update original fracture set with new values for aperture and transmissivity

Use Aperture (e) from Elastic Eq.:

- **Poisson's Ratio**
- **Young's Modulus**
- **Max Flow Distance**

a,b,c: parameters for updating fracture transmissivity using equation shown on left

Display full report on screen: show report upon simulation completion

Keep ungrouted portion of fractures in output data: include a fracture set containing the fractures that did not receive grout in the output

Grouting Results

After simulation, a result container is created under Analysis. It contains objects including the fracture set that receives grout, stress points (if a critical stress analysis was performed), a report and, optionally, a fracture set containing fractures that did not receive grout.

Seven new properties are added to fracture set that contains grout injected fractures :

"Sequence" – a sequential integer number for the order that fracture element receives grout.

"Volume (m³)" – element volume, equal to fracture element area times storage aperture.

"PenetrationLength (m)" – total penetration length at the exit of this element.

"Time (s)" – the travel time at the exit of element.

"VolRate (m³/s)" – volumetric flow rate, flow_width [m] X storage_aperture [m] X grout velocity [m/s].

"Pressure (Pa)" – The reported pressure here is negative and only meaningful when the pump method is constant rate injection. This property magnitude represents the min pump pressure to push grout all the way through this element. The earlier the elements receiving grout are, the smaller magnitudes of P's. If we plot P's in negative values along the penetration depth or travel time, the curve becomes decreasing. For injection method of constant pump pressure it is set as zero for all elements for now.

"StorageAperture (m)" – Storage aperture is carried from input fracture aperture in the beta version. In the future version we may change storage aperture due to grout injection.

The grouting report includes the following sections:

- Input and output grout volume
- Echo input parameters.
- Summary of DFN in simulation
- DFN connection level and volume
- Detail injection history to each fracture element.

Statistics

Stereonet

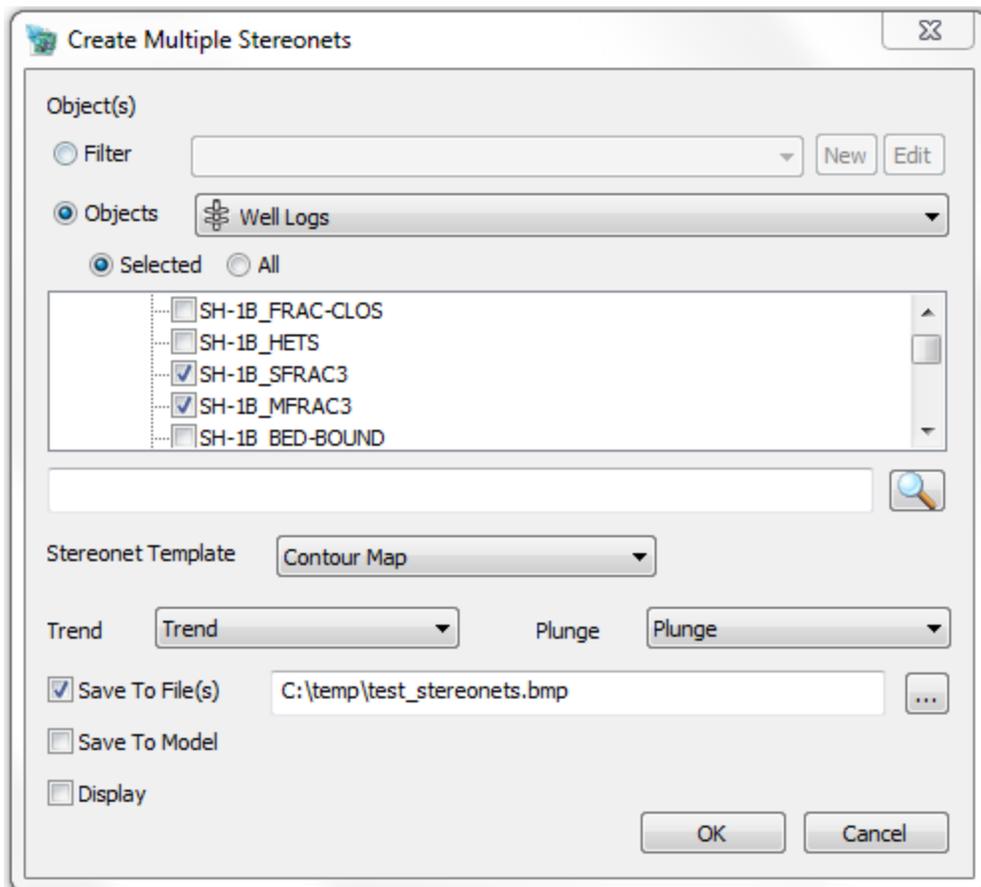
See the section on [Stereonet](#) in the Working with Charts section of the Manual.

Property Stereonet

See the section on [Property Stereonet](#) in the Working with Charts section of the Manual.

Create Multiple Stereonets

Often you want to create several stereonet sharing the same template. Perhaps these are well log data subdivided by depth or fracture type, or just many different fracture sets. This can be accomplished using the **Create Multiple Stereonets** functionality. With this functionality, each stereonet can display data for single objects or for data selected using a filter. If you select objects, be sure to combine or subdivide objects as necessary before trying to create the stereonets. Also, you need to have a stereonet template created if you want anything other than the default templates provided. This can be created by customizing one stereonet as desired and then saving the template for later use (see Manual section for [Stereonet Templates](#)). Once these tasks are done, open the **Create Multiple Stereonets** dialog by selecting from the top menu **Static Analysis -> Create Multiple Stereonets**.

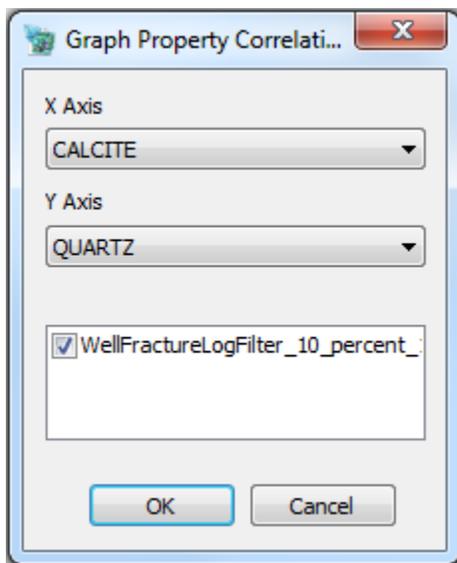


Select either a **Filter** or a set of **Objects** to create stereonets for. Select your desired template. By default, the stereonet uses Trend and Plunge properties if they are available, but any two object properties may be selected (e.g. Dip_Direction and Dip). Check one or more choices to save the stereonets. If you select **Save To File(s)**, enter the full path to a file, including the extension. The created files will use the given filename as a prefix and add in the object name for each stereonet created. In the dialog shown above, the two files created would be named “test_stereonetsSH-1B_SFRAC3.bmp” and “test_stereonetsSH-1B_MFRAC3.bmp”. If you select **Save To Model**, the stereonet will be saved as an **Analysis** object and you can display it by right-clicking and selecting **View Data**. Click **OK** to run.

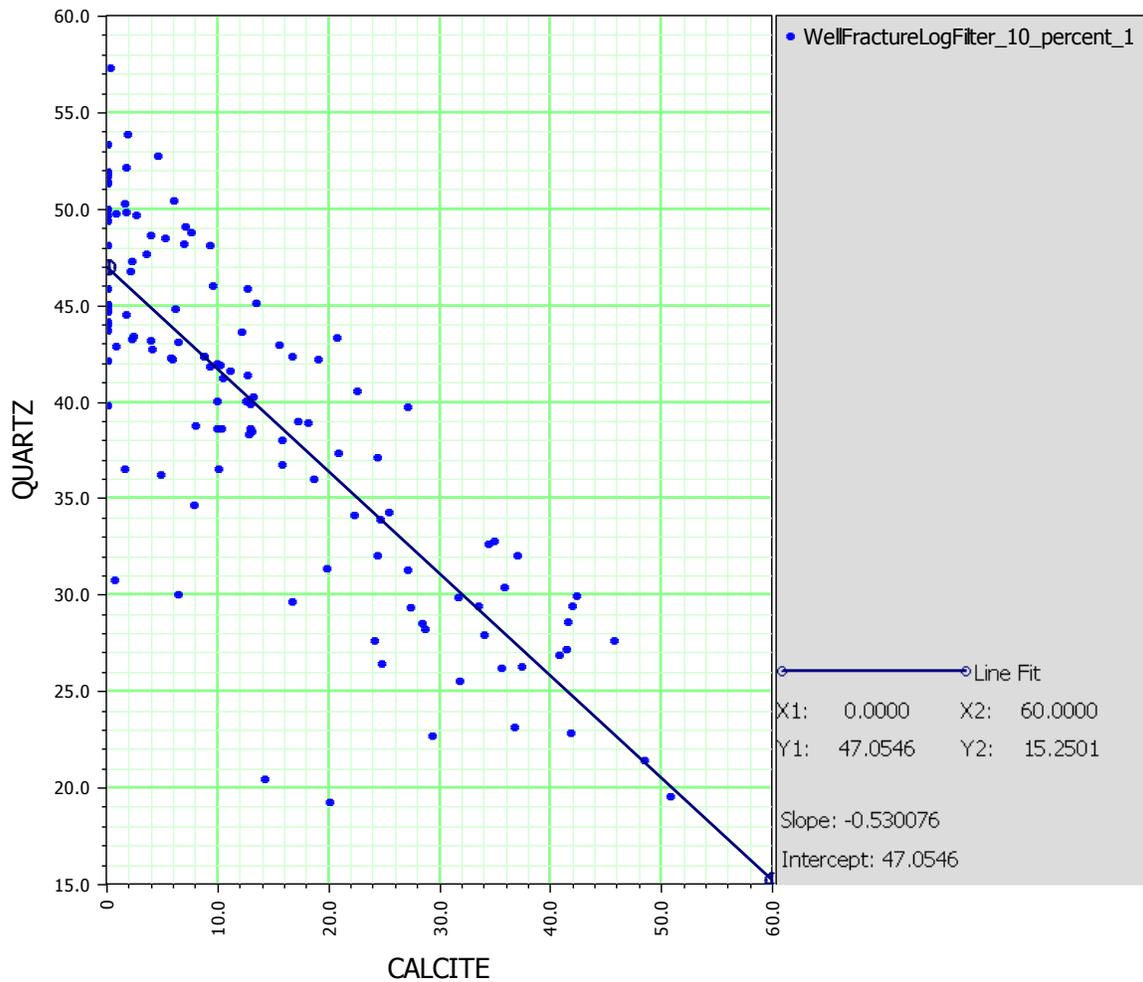
Property Correlation Graph

The Property Correlation Graph allows you to plot one property as a function of another (Y vs X). This can be especially useful for grid or well log properties. The number of property values in the two lists need to be the same which is usually the case for choosing two properties from the same object, but may not be case when properties from two different objects are chosen.

To create this graph, select the object(s) that have the properties that you want to compare in the Object window. Select Static Analysis -> Property Correlation Graph from the top menu. The Graph Property Correlation dialog will open with the objects selected in the bottom pane and the list of available properties listed in the X Axis and Y Axis drop-down select boxes. Note that if two different objects were chosen, the available properties in both lists are those having names common to both objects.



Select the properties and click on **OK**. The graph will appear as a new tab in the **Visualization window**. The graph may be modified as usual using the **Attributes window** (see Manual section on [Working with Charts](#) for more detail). Right-clicking on the graph allows you to save the chart to the clipboard or file, view the data, zoom in or out or change the type of curve fitting applied.

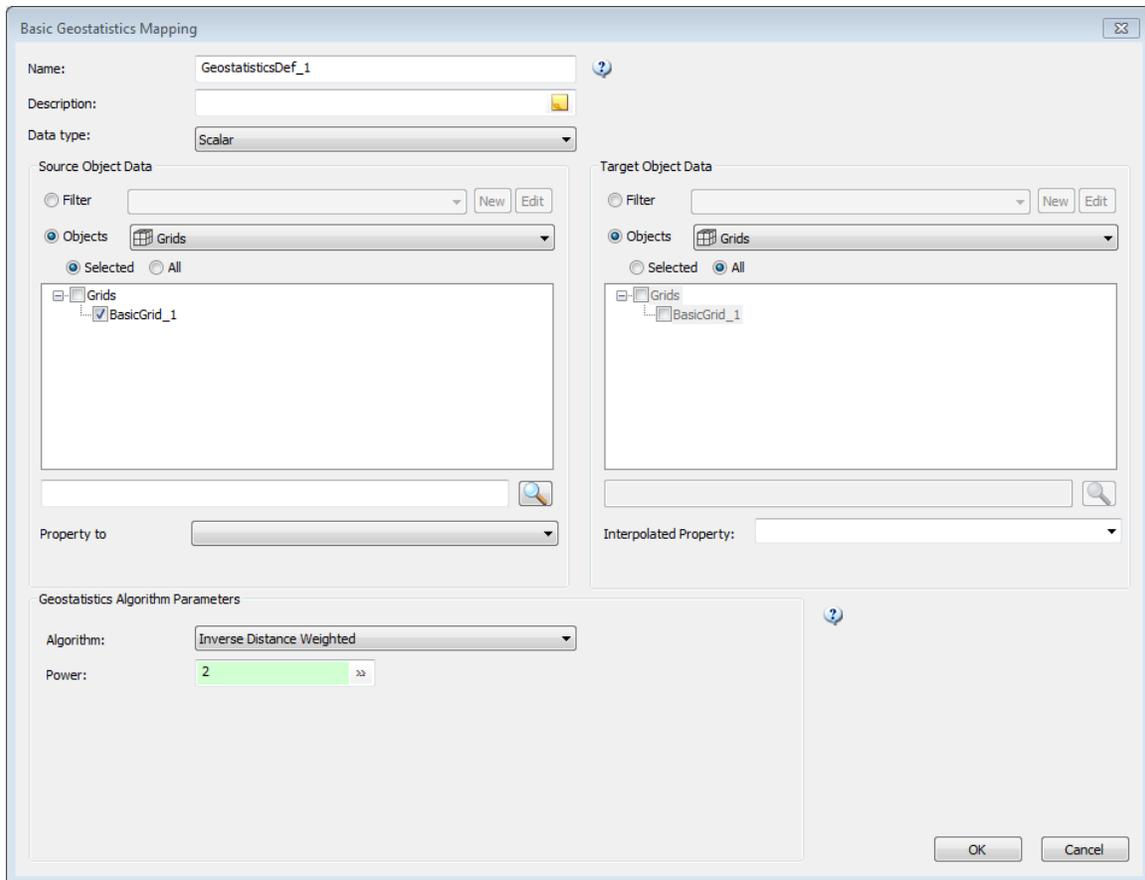


Basic Geostatistics Mapping

FracMan provides basic geostatistics features including:

- Inverse Distance Weighted
- Points On Convex Set (POCS)
- Ordinary Kriging
- Modified Shepard

Additionally FracMan provides the ability to map trend / plunge values together as a 3D orientation.



Inverse Distance Weighted

Points On Convex Set (POCS)

Ordinary Kriging

Modified Shepard

Dynamic Fracture Analysis

Dynamic analyses are among the most complex that FracMan offers. Like static analysis, dynamic analysis seeks to answer the questions you built your DFN to resolve, but it differs from other analyses in that it uses time, pressure, and flow in its calculations – that is, the results differ based on time.

Transient Well Test

Overview

The current version of FracMan can be used to compute transient single-phase flow solutions of slightly compressible fluids, such as oil and water, or highly compressible gas flow through a discrete fracture network. Caution needs to be taken in the case of gas flow. For single-phase well tests, a generic matrix block scheme can be used to approximate fluid exchange between fractures and the surrounding rock matrix.

FracMan supports two kinds of flow calculations:

Well tests, which model the transient response of fluid to injection and production events within the fracture network.

Directional permeability (block K) simulations, which compute equivalent permeability tensors over the cells of a grid for dual-continuum simulation.

To perform a flow calculation, the following model objects are required:

A fracture set with permeability, aperture and compressibility (optional for block K simulations) properties

An axis-aligned grid (optional for well tests)

For a well test, one or more wells and associated well perforation intervals

Fluid flow is computed using the Galerkin finite element method, which subdivides fractures into smaller triangular elements. Overall mesh refinement can be controlled by the user, and extra refinement near wells is automatic. For single-phase flow simulations, both the triangulated fractures and the pressure values can be loaded back into FracMan for visualization and analyses.

Well Test Simulation

Single well test simulations are defined and run from a **Well Test** definition.

On the **Dynamic Analysis** menu, select **Well Test...** to open the Well Test definition page.

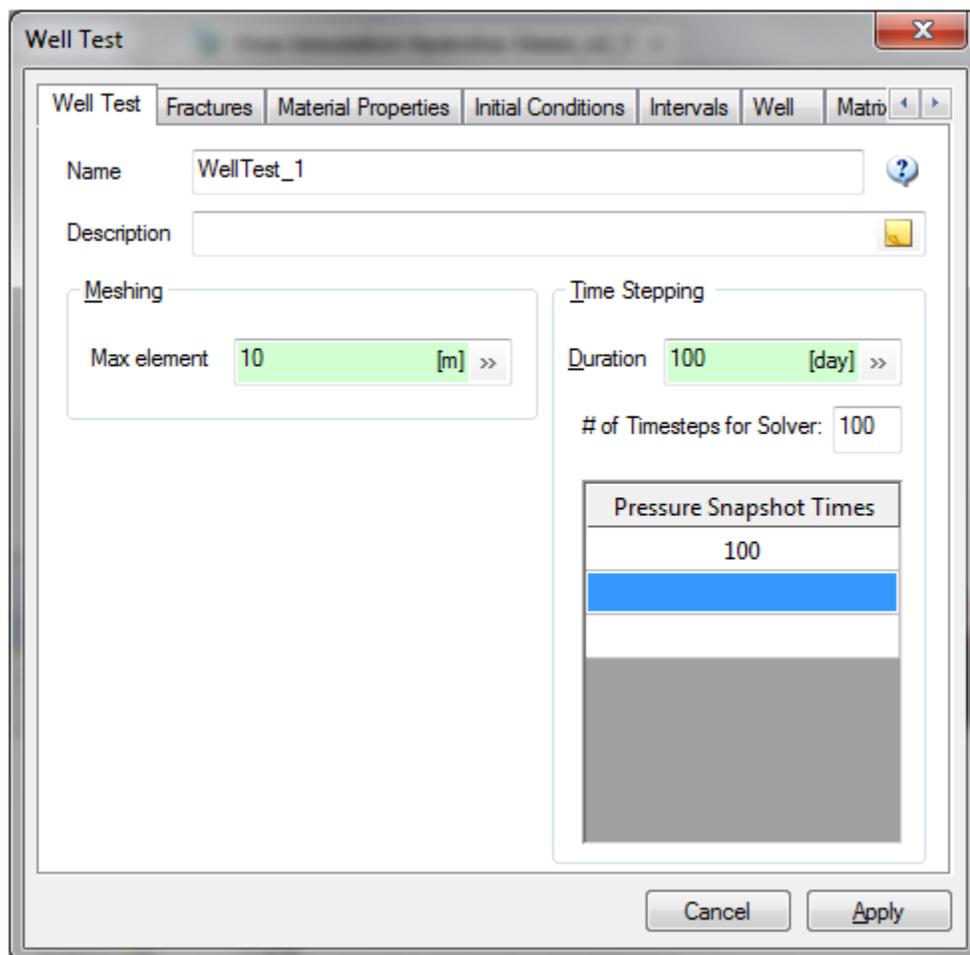
Well Test Tab

Name: label that will be displayed in the **Object Control Window** on the **Definition** tab under the **Flow Simulations** group. This will also be used to name output files for the simulation.

Max element: This is the maximum length in meters that an edge of any finite element in the mesh is allowed to have. This number must be appropriate for the size of the smallest grid cell; otherwise, elements will be too large and no “free” nodes will be produced by the meshing algorithm.

Duration: This is the duration of the well test in the current time units. If this number is greater than the maximum well flow rate duration, then the remainder of the simulation will run with all well flow rates at 0.

Pressure Snapshot Times: These are the times at which pressure data for the entire mesh is stored for visualizing and analyzing. For smoother derivative plots, specify only one output time equal to the duration.



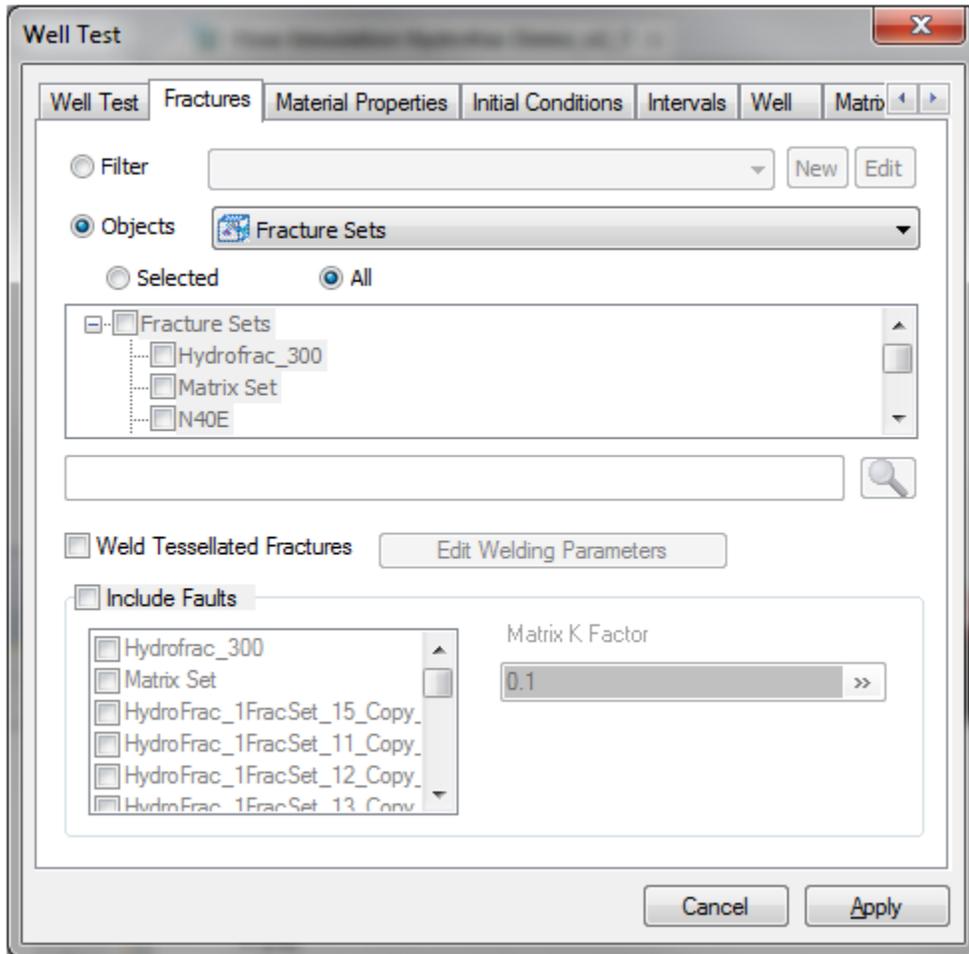
Fractures Tab

On the second tab, select the fractures to include in the simulation.

Note: one can use the default, using all fractures in the model, or a filter can be specified to only include a subset of fractures, but only those fractures which are connected to at least one well will be included in the simulation.

Tessellated fractures can also be welded and included in simulation.

Matrix K-factor: It is specified to adjust matrix K value if the matrix block is within fault region.



Material Properties Tab

Material properties for oil, water or gas under reservoir conditions, i.e. reservoir pressure and reservoir temperature (for gas flow).

If **Use depth-dependent values** is checked, different material properties along specified depth should be provided.

When **Storage aperture multiplier** is checked, a multiplier is applied to the storage aperture to increase or decrease the fracture storage, while permeability/transmissivity is not affected.

Well Test

Well Test | Fractures | Material Properties | Initial Conditions | Intervals | Well | Matr

Fluid

Density 800 [kg/m³] >>

Viscosity 0.2 [cP] >>

Compressibility 3e-005 [1/psi] >>

Use depth-dependent values

Edit Values...

Storage aperture multiplier 1 >>

Cancel Apply

Depth Dependent Fluid Values

Depth	Density	Viscosity	Compressibility

OK Cancel

Initial Conditions Tab

On the fourth tab, set the initial values for the simulation:

If **Use restart file** is checked, a restart file and the time for restarting should be specified

If Gas Flow Rate Conversion to Standard Condition is checked (gas flow only), the reservoir temperature and gas compression factor Z should be specified to convert rate from reservoir condition to surface standard condition (1 bar and 293 K temperature).

The screenshot shows the 'Well Test' dialog box with the 'Initial Conditions' tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with tabs for 'Well Test', 'Fractures', 'Material Properties', 'Initial Conditions', 'Intervals', 'Well', and 'Matrix'. The 'Initial Conditions' tab is active and contains two main sections:

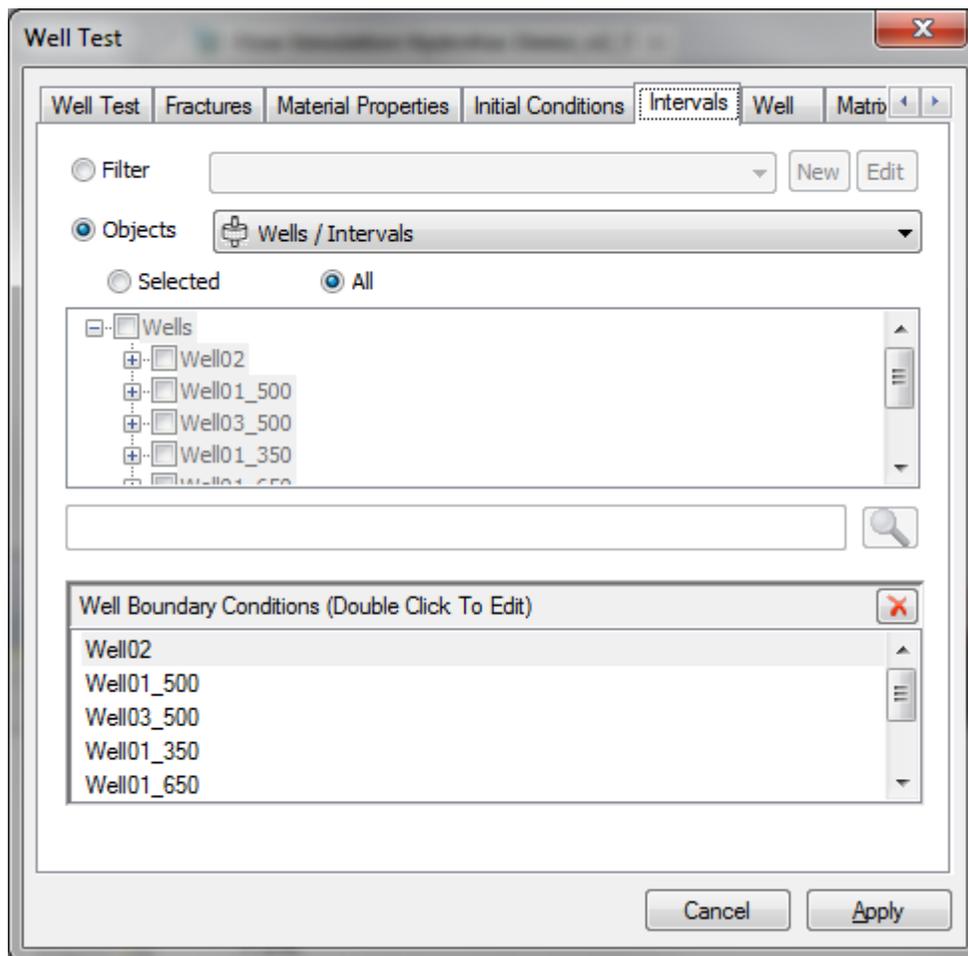
- Reservoir Condition:**
 - Use restart file: A checkbox followed by a text input field and a browse button (...).
 - Time at restart file: A numeric input field containing '0', followed by '[day]' and a right-pointing arrow (»).
 - Initial Pressure: A numeric input field containing '1000', followed by '[psi]' and a right-pointing arrow (»).
- Gas Flow Rate Conversion to Standard Condition:**
 - Reservoir Temperature: A numeric input field containing '10', followed by '[°C]' and a right-pointing arrow (»).
 - Gas Compression Factor Z: A numeric input field containing '1', followed by a right-pointing arrow (»).

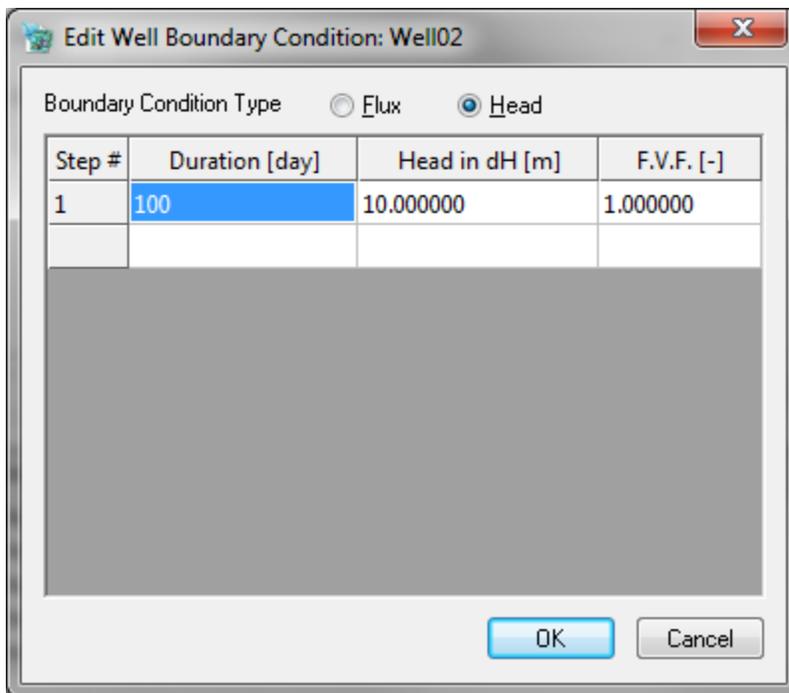
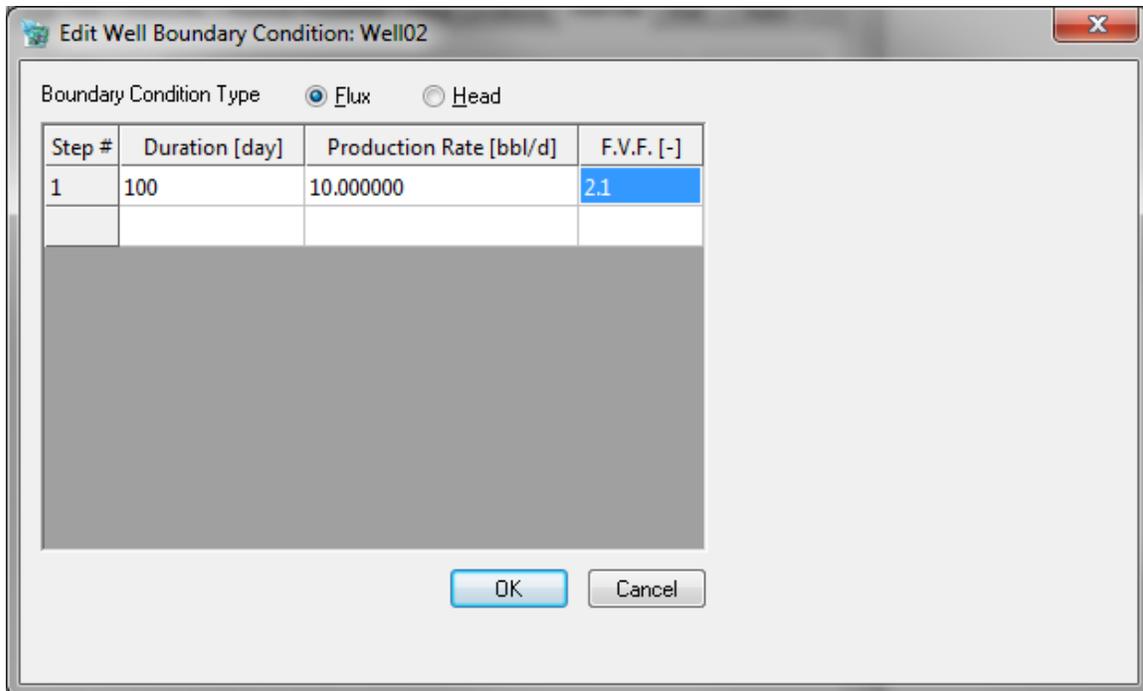
At the bottom of the dialog are 'Cancel' and 'Apply' buttons.

Intervals Tab

On the fifth tab, select the wells / intervals to be included in the simulation. To select a group of wells, click on the first well's label, then shift-click on the second well's label. To select or deselect individual wells, ctrl-click on that well's label.

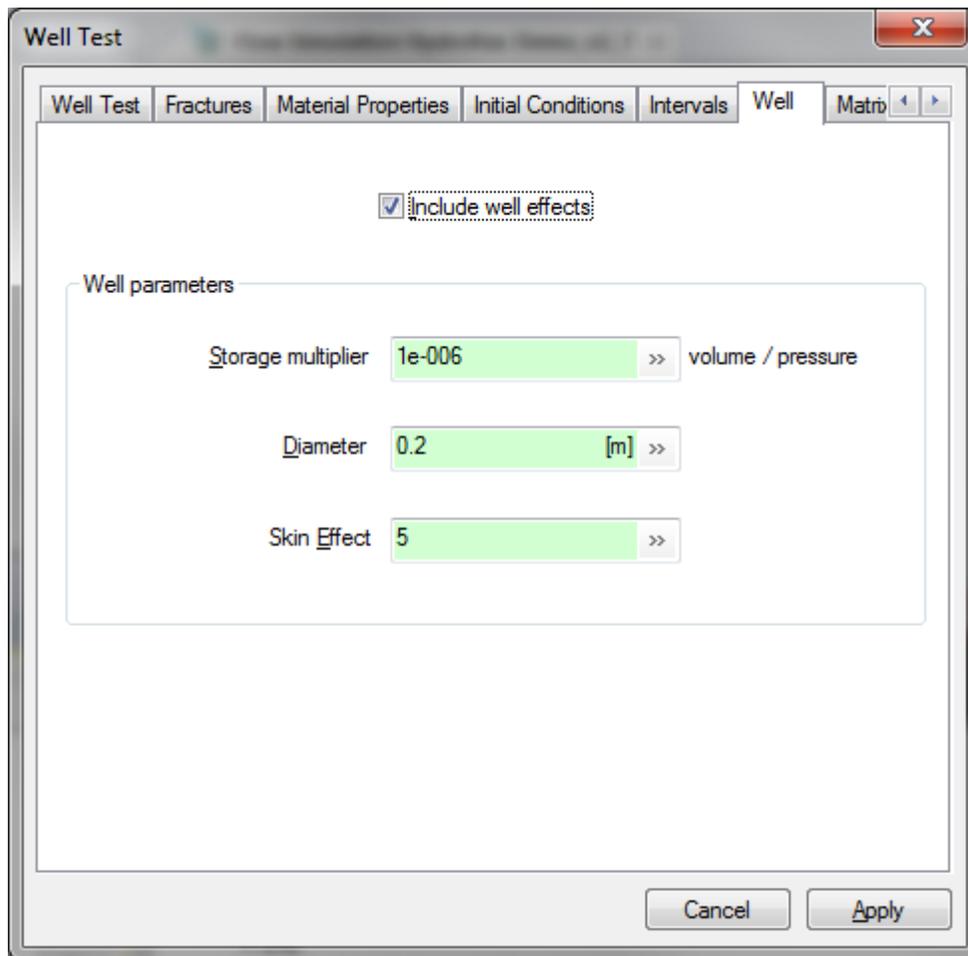
To specify the well boundary conditions, double click the well name to edit the flow duration, production rate (Flux Option)/Head in DH(Head option) and formation volume factor (F.V.F).





Well Tab

On the sixth tab, check Include Well Effects to have the simulation include the effects of skin and borehole volume. Note that negative skin is currently not working in MaficOil. One can manually change the permeability for nodes around the well to achieve stimulated permeability effect (negative skin effect).



Matrix Tab

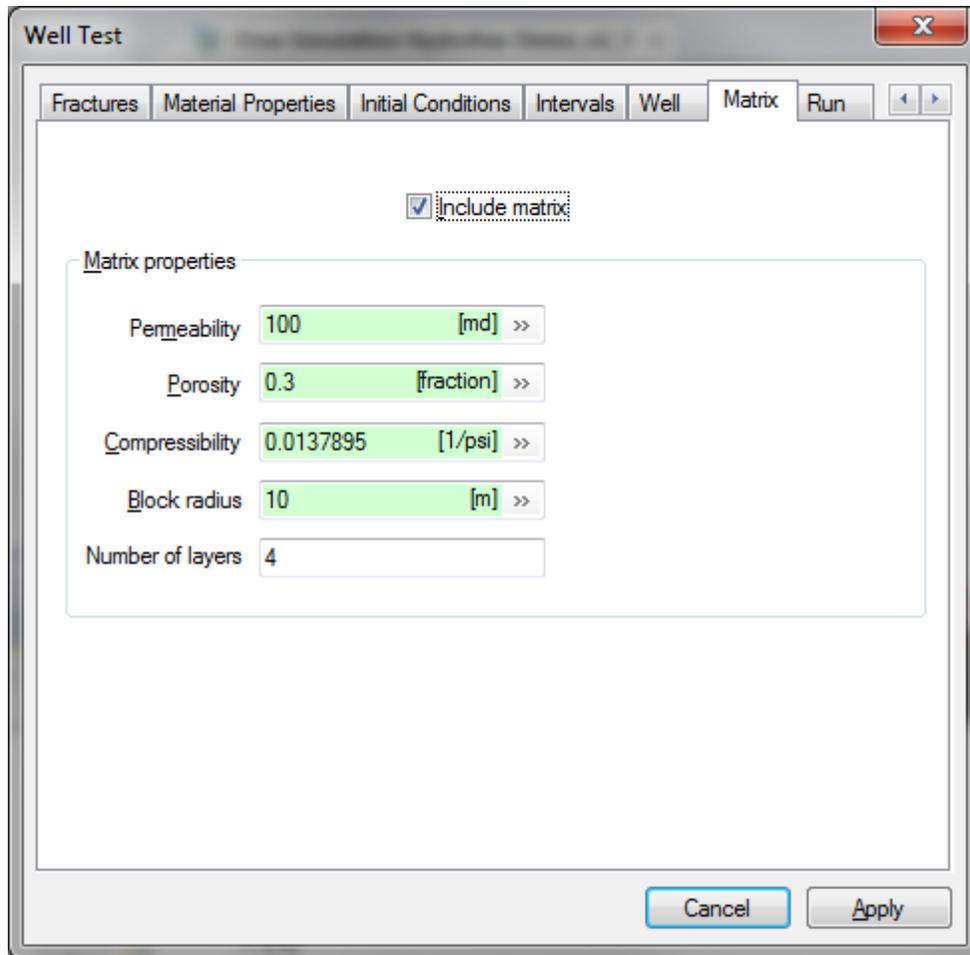
On the seventh tab, check **Include matrix** to include fluid exchange between the surrounding rock matrix and the fractures.

Compressibility on this tab is the matrix compressibility. The total compressibility is a combination of the fluid compressibility from the Materials Properties tab, C_F , and the matrix compressibility, C_M , weighted by the matrix porosity, ϕ_M :

$$C_T = \phi_M * C_F + (1 - \phi_M) * C_M$$

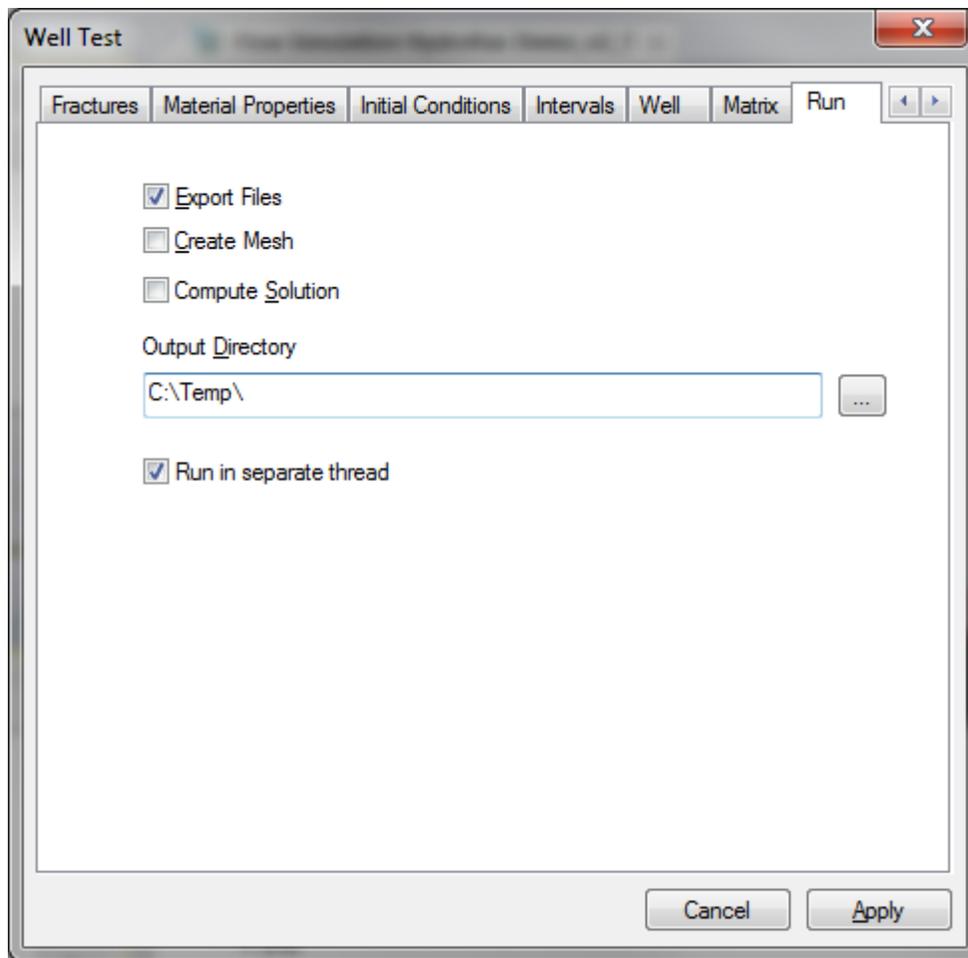
Block radius is specified to determine the proportion of matrix block flow to individual fractures, and this should be specified such that: 1) the total surface drainage area of the matrix blocks equals the available fracture surface area, and 2) the total volume of the region equals the total volume of the matrix blocks. More details on matrix block radius calculation can be found in Mafic Manual.

Number of layers is to specify the layers for finite element discretization.



Run Tab

On the final tab, specify Run specifications

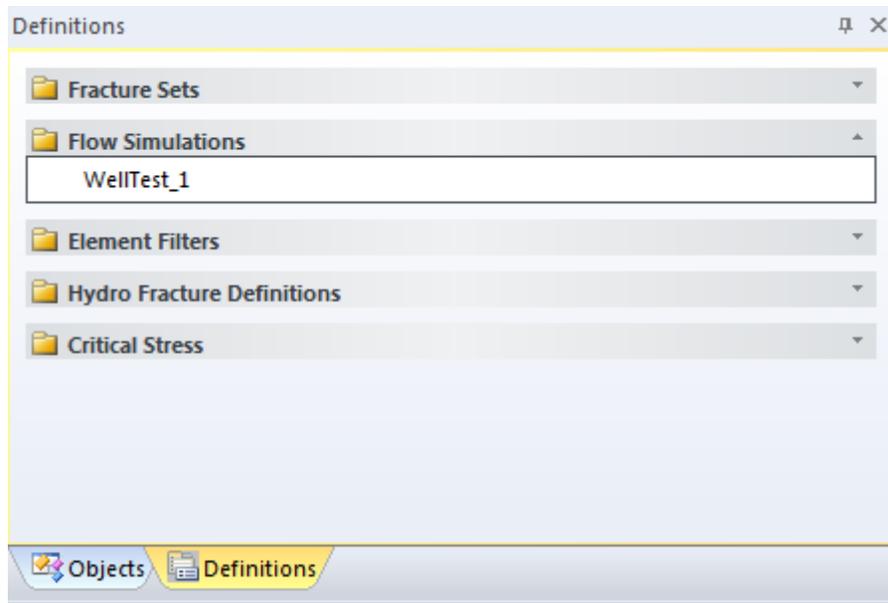


If the **Create Mesh** button is checked, the user must select the directory into which the mesh files are exported. If a path to the mesher has not already been set, then the user will also be prompted to locate that executable.

If the **Compute Solution** button is checked, the user must select the directory in which mesh files corresponding to this experiment must exist. If none exist, then an error will occur. Otherwise, a flow solution will be computed and results will be placed in that directory.

Select an output directory.

Click **Apply** to complete the simulation definition. A new object under **Flow Simulations** on the **Definitions Window** will be added with the Name specified in the **Well Test** tab.



Visualizing Well Test Results

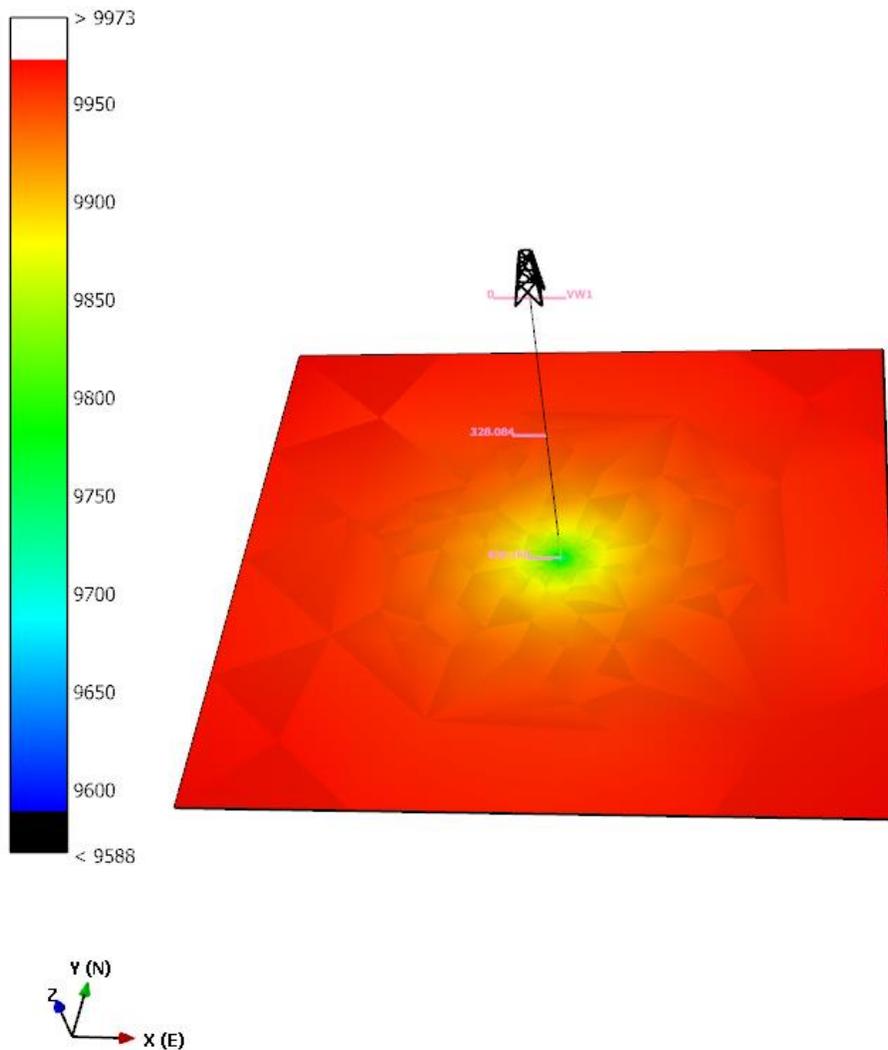
When a flow simulation has finished, each node (vertex) of the mesh has a pressure associated with it for each output time of the simulation. The flow solutions and their associated meshes may be visualized within FracMan.

To Visualize Pressure Distribution from a Single Phase Well Test

In the **Object Control Window**, hide items, such as fracture sets and grids, which could obscure visualization of the pressure data by unchecking the box next to their labels.

Expand the **Well Test Object** by clicking the “+” next to its label.

Right-click on any one of the fracture labels beneath the flow simulation object and hold the mouse cursor over the **Visualize** item from the popup menu.



Pressure - 100

The pressure results at the user-specified output times will be listed in the subsequent popup menu. From these, choose which results to display in the visualization window.

When a particular pressure result is selected, the bottom of the **Visualization Window** will read “Pressure-<output time>” and the fractures involved in the simulation will be displayed in colors corresponding to the pressure values at each node.

To view the pressure values to which each color corresponds, choose **Legend** from the **View** menu.

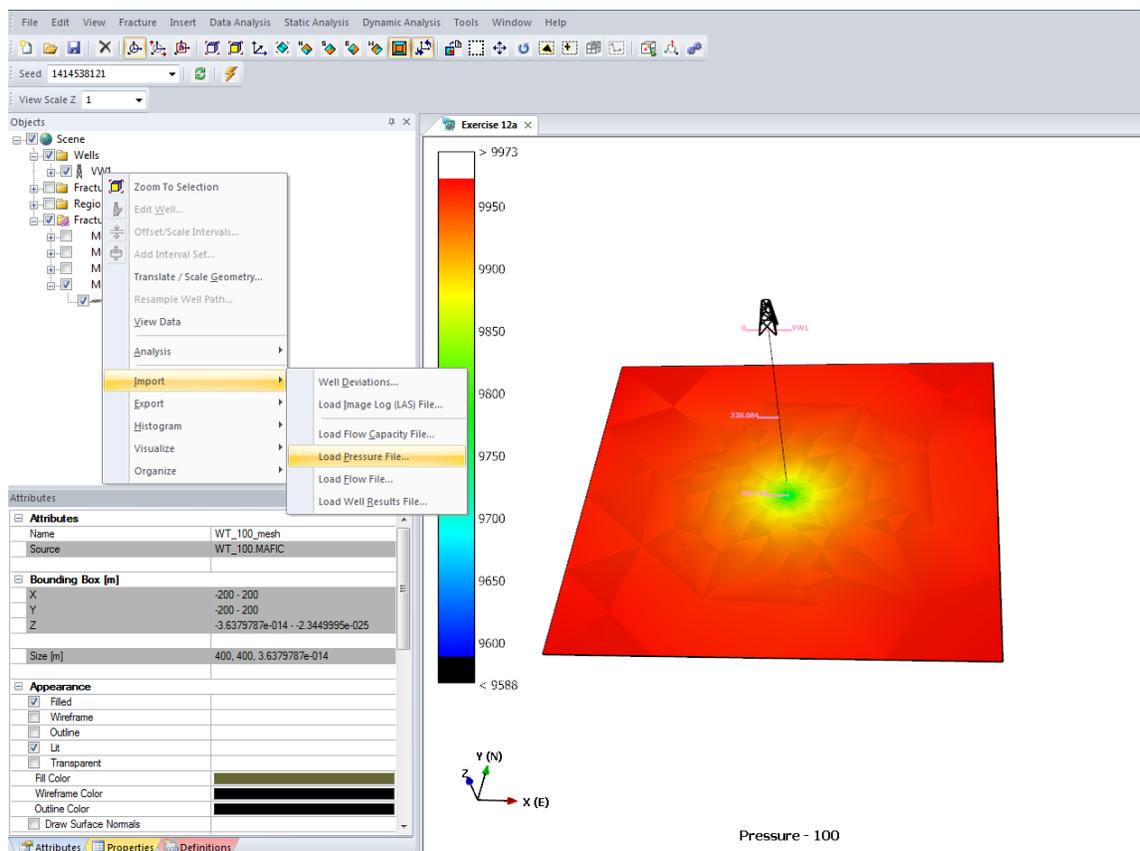
Finite element meshes and pressure data from flow simulations can also be imported into FracMan independent of a well test.

To Import an FEM Mesh and Pressure Data

On the File menu select Import | MAFIC mesh to invoke the Import MAFIC mesh dialog.

Select the desired mesh file. Such files will have the extension .mafic_output, .mft, or .maf.

In the **Objects Window**, a mesh object with the label **Well Test: <filename>** will appear. Right-click on its label and select **Import Pressure File** from the popup menu.



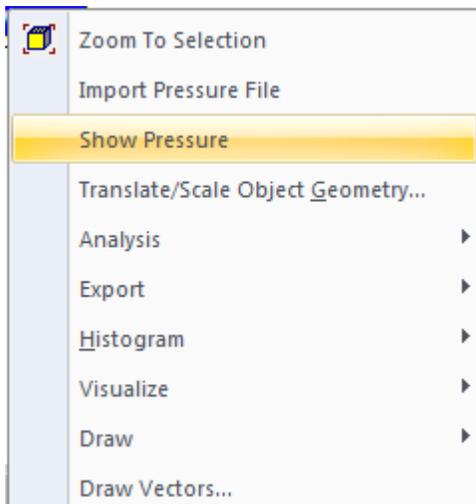
In the file dialog, select the pressure file corresponding to the flow simulation.

Follow the steps described above to visualize pressure results.

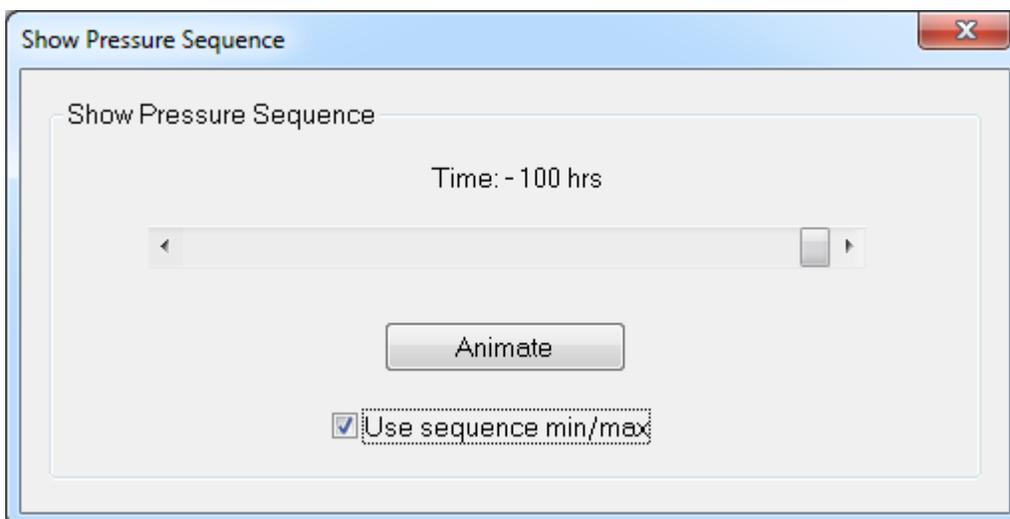
Pressure results from a well test can also be animated in the visualization window.

To Animate Pressure in the Visualization Window

In the **Objects Window**, right-click on the flow simulation object and select **Show Pressure** from the popup menu.



In the **Show Pressure Sequence** dialog, click the **Animate** button to step through the pressure data at each output time.

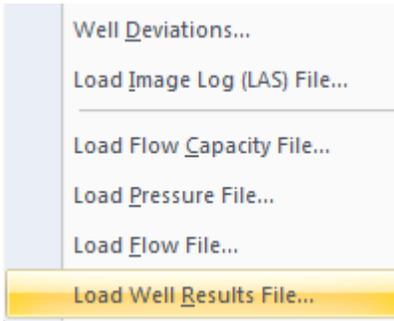


To range the colors according to the pressure data over all times, check the **Use sequence min/max** checkbox. Otherwise, colors will be chosen to match the pressure range at the particular output time.

At the end of a well test, computed pressure values at the well will be loaded automatically; however, flow rates (flux) will not. These values can be plotted as a time series or as a pressure derivative plot.

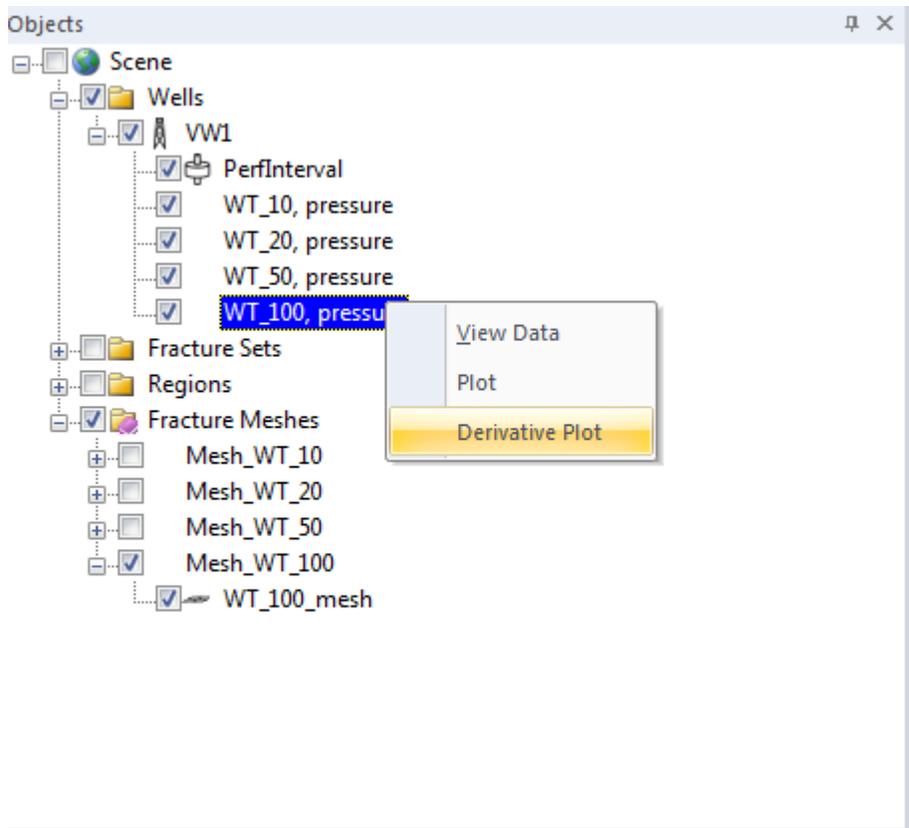
To Import and Plot Pressure or Flow Rate Results from a Simulated Well test

In the **Objects Window**, right-click on the well object and select **Import->Load Well Results File**.



In the file open dialog, select the “*proj*.mafic_P_Well” file to load the simulated pressure values at the well. Select the “*proj*.mafic_Q_Well” file to load the flow rate (flux) values at the well.

Depending on the type of file selected, a child object with “- FlowRate” or “- Pressure” will appear underneath the well object. To view the data in a text window, select **View Data**. To plot the data as a time series, select **Plot**. To view the derivative plot, select **Plot Derivative**.



FracMan 7.50 Beta (2014.10.20) 64bit

File Edit View Fracture Insert Data Analysis Static Analysis Dynamic Analysis Tools Window Help

Seed 1414605308

View Scale Z 1.0

Objects

- Scene
 - Wells
 - VW1
 - PerfInterval
 - WT_10, pressure
 - WT_20, pressure
 - WT_50, pressure
 - WT_100, pressure
 - Fracture Sets
 - Regions
 - Fracture Meshes

Exercise 12a1 | Text Results2

Time	pressure
0	10000
7.71605e-008	9956.49
8.79209e-008	9951.16
1.00182e-007	9945.88
1.14153e-007	9940.69
1.30072e-007	9935.6
1.48211e-007	9930.66
1.68879e-007	9925.86
1.9243e-007	9921.23
2.19266e-007	9916.74
2.49843e-007	9912.41
2.84685e-007	9908.21
3.24386e-007	9904.11
3.69623e-007	9900.12
4.21168e-007	9896.19
4.79902e-007	9892.33
5.46826e-007	9888.51
6.23084e-007	9884.73
7.09976e-007	9880.98
8.08985e-007	9877.25
9.21801e-007	9873.55
1.05035e-006	9869.87
1.19683e-006	9866.22
1.36373e-006	9862.59
1.55391e-006	9858.99
1.77061e-006	9855.41
2.01753e-006	9851.85
2.29888e-006	9848.31
2.61947e-006	9844.78
2.98476e-006	9841.25
3.401e-006	9837.73
3.87529e-006	9834.21
4.41571e-006	9830.69
5.0315e-006	9827.17
5.73317e-006	9823.65
6.53269e-006	9820.14
7.4437e-006	9816.63
8.48176e-006	9813.12
9.66458e-006	9809.62
1.10123e-005	9806.13

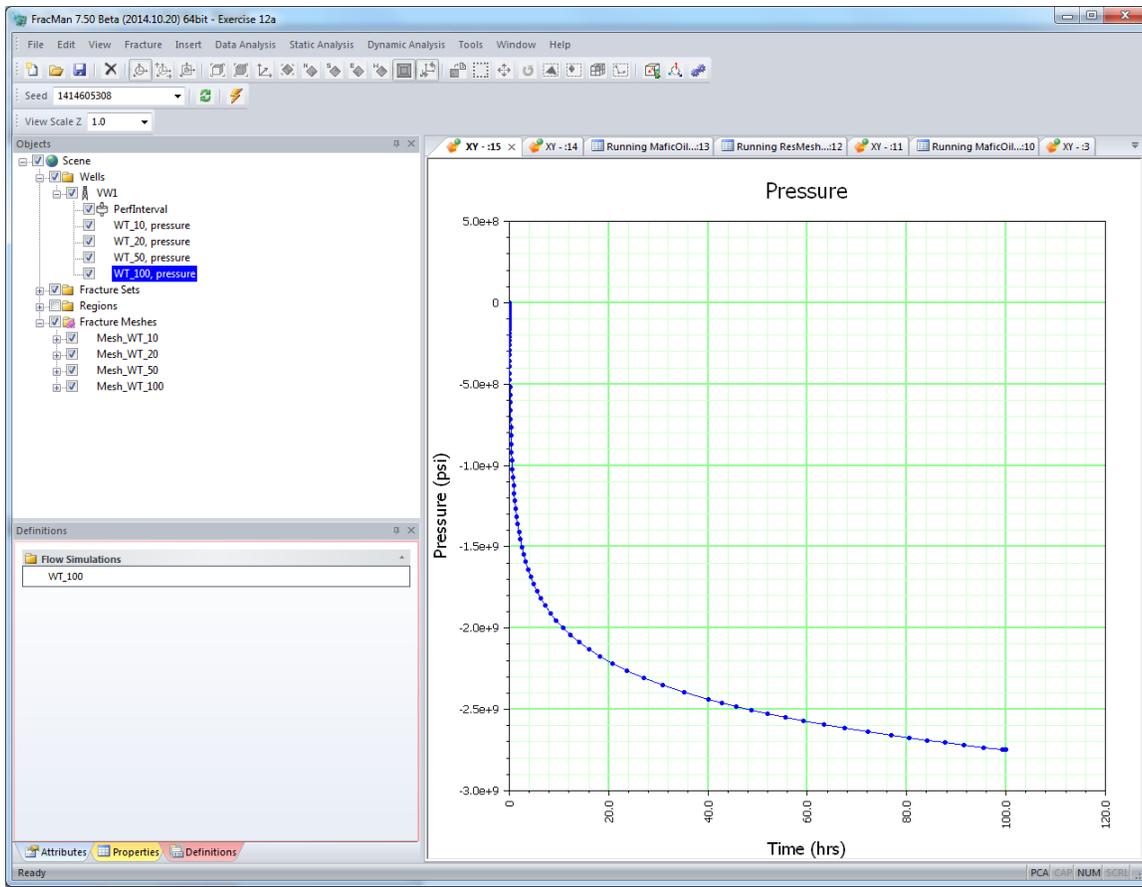
Definitions

Flow Simulations

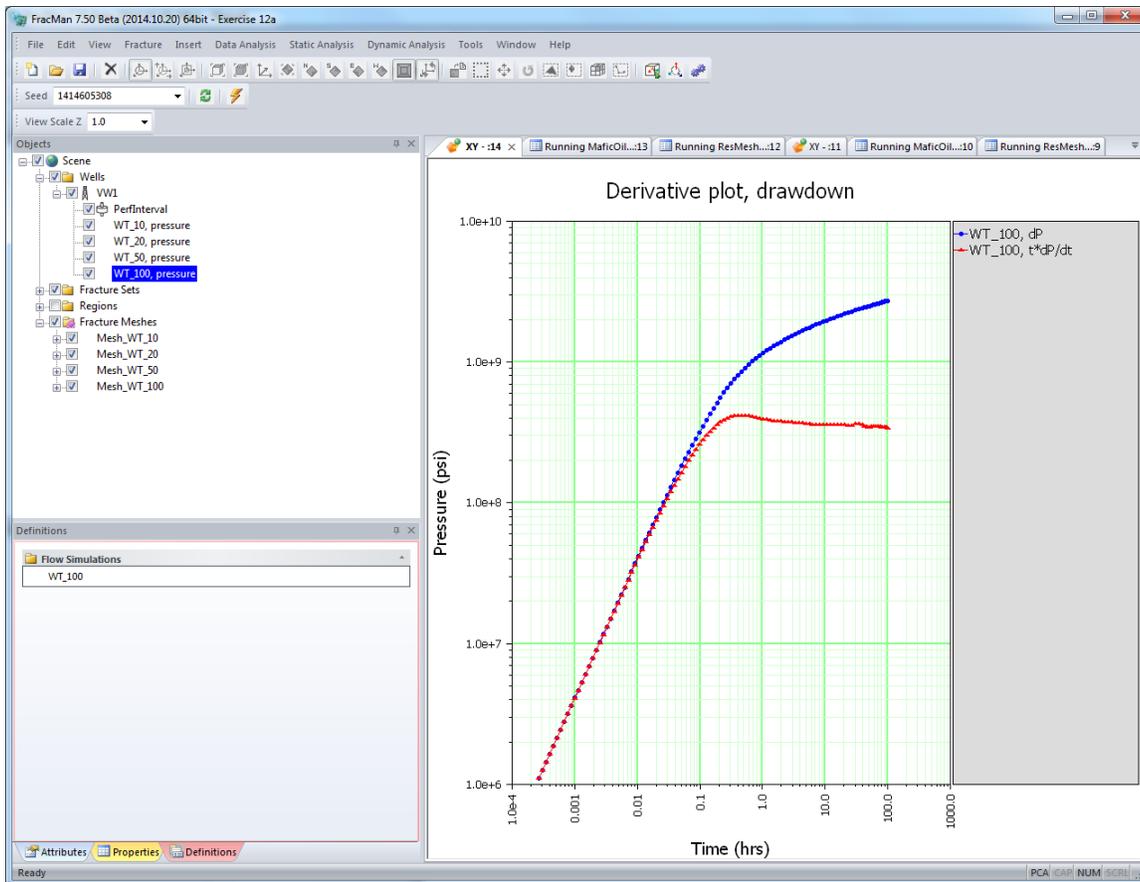
Attributes Properties Definitions

Ready

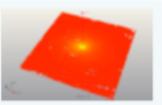
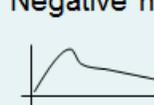
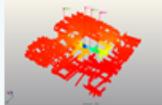
PCA NUM



The well test derivative curve can tell us the fluid conduit mechanism.



From the slope of $\log(t \, dp/dt)$ versus $\log(dt)$ we can judge if fluid flows to well mainly by 1-D fracture, 2-D fracture network or 3-D homogeneous flow. Fracture networks produce well-test curves that are similar to those of standard porous media models but with different properties and geometries.

Derivative Behavior	Fracture	Porous Medium
Half-slope  	One strong fracture set	Vertical hydraulic fracture
Zero-slope  	Space-filling fracture network in planar region	Homogeneous radial flow
Negative half slope  	3-D network or 2-D system with additional vertical connectivity, e.g. faults	Spherical flow, partial penetration
Other slopes  	Partial space filling network; non-homogeneous intensity	Multiple complex boundaries

The flow rate for each fracture is reported in the `"*.proj*.mfaci_Q_well"`:

Try using a different realization of the fracture network.

If the flow simulation is terminated or cannot run successfully, it could be because of the following reason:

The fracture permeabilities and porosities in the model cannot sustain the specified flow rate schedules. Reduce the flow rates or durations, or adjust the fracture properties.

MAFIC Flow Simulation

MAFIC (Matrix/Fracture Interaction Code) was developed by Golder Associates to simulate transient flow and solute transport through three-dimensional rock masses with discrete fracture networks. Flow and solute transport are simulated in both the fractures and the rock matrix. The FracMan package provides a complete PC and workstation based system for analysis of geological and hydrological data, synthesis of hydrogeological conceptual models, and engineering analysis. MAFIC provides flow and transport simulation for hydrogeological conceptual models generated under FracMan. Input files for MAFIC can be generated by FracMan/MeshMaster and edited by FracMan/EdMesh.

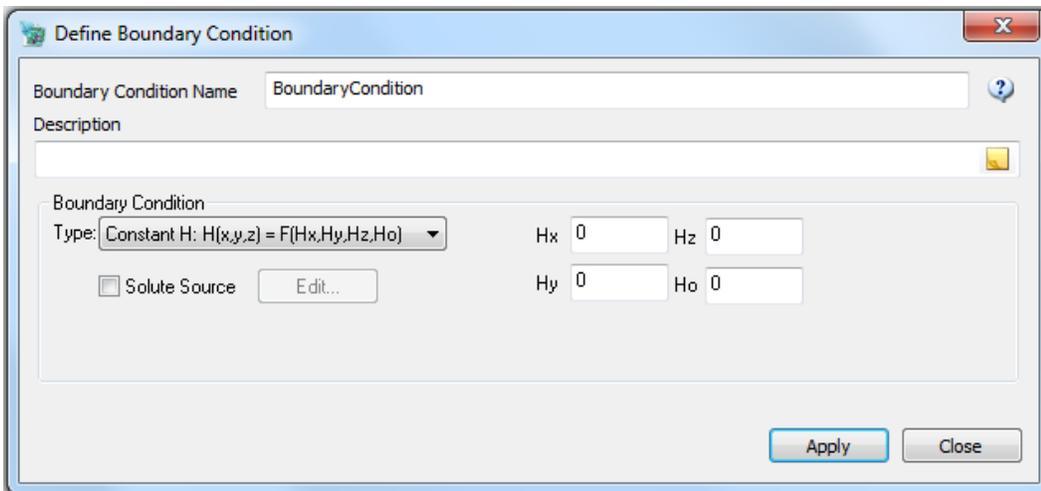
For a complete description of MAFIC flow simulation see the MAFIC Manual, "MAFIC Manual.pdf", located in the FracMan7 Program directory.

Boundary Conditions

Boundary Conditions for the simulation may be defined independently.

Constant Head

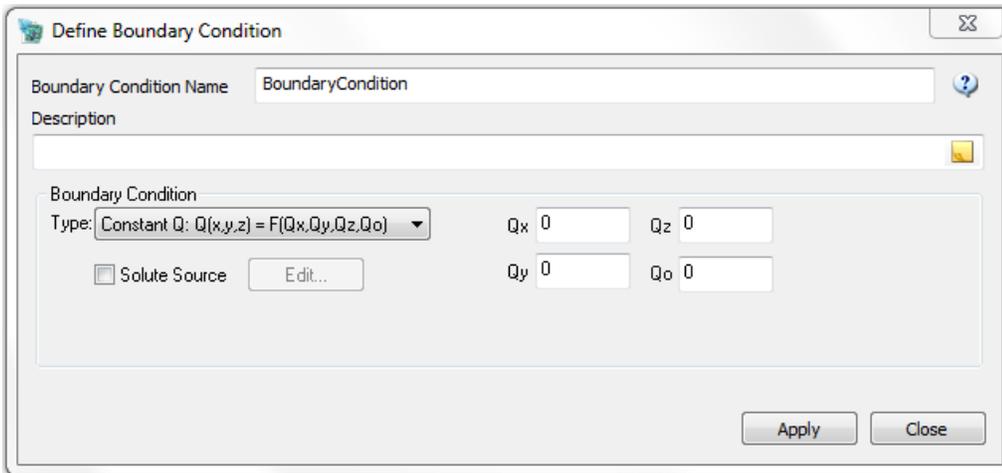
The constant head can be applied as a coefficient to X, Y, and Z dimensions (i.e. if 1 is applied in the Hz box, then as Z increases so will head). The constant head value can also be applied as to an object that does not vary in space (using the Ho box) but as a single value.



Constant Flux

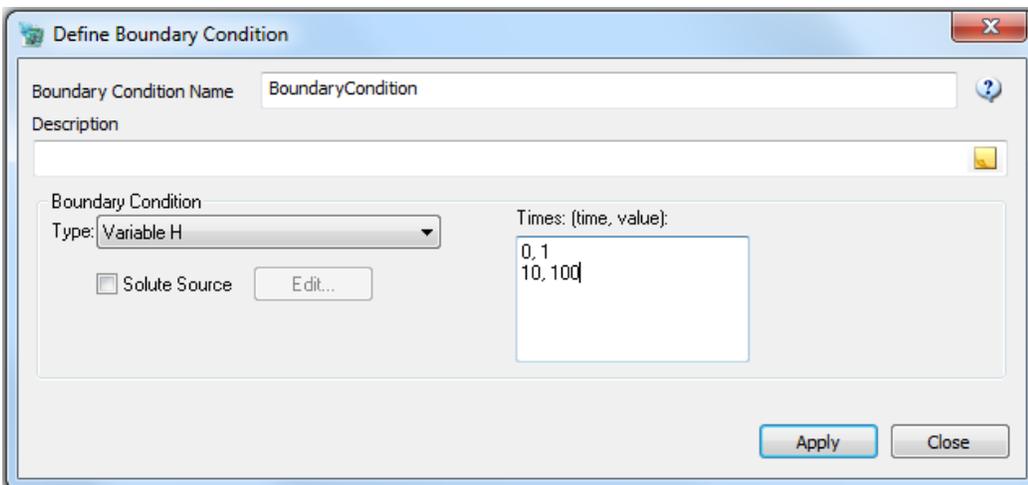
The constant flux can be applied as a coefficient to X, Y, and Z dimensions (i.e. if 1 is applied in the Qz box, then as Z increases so will the flux). The constant flux value can also be applied as to an object that does not vary in space (using the Qo box) but as a single value.

In Mafic, the only way to assign a No Flow boundary is to use the constant flux boundary condition type and assigning 0 to all the boxes in boundary condition definition as show below:

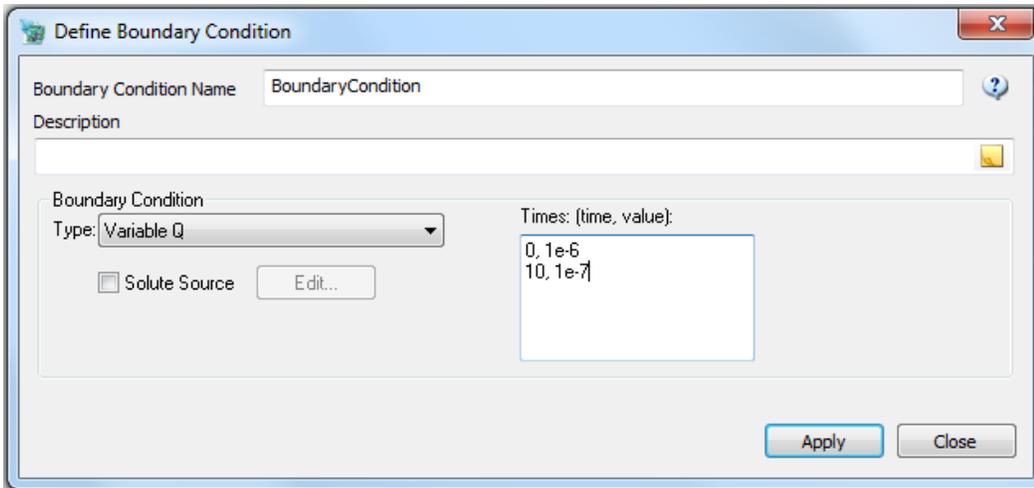


Variable Head

Time-varying boundary conditions are specified for each nodal group as chronologically tabulated data pairs of elapsed time and boundary value. For each time step, the corresponding boundary value is determined from linear interpolation of the tabulated values using the elapsed time at the end of the time step.

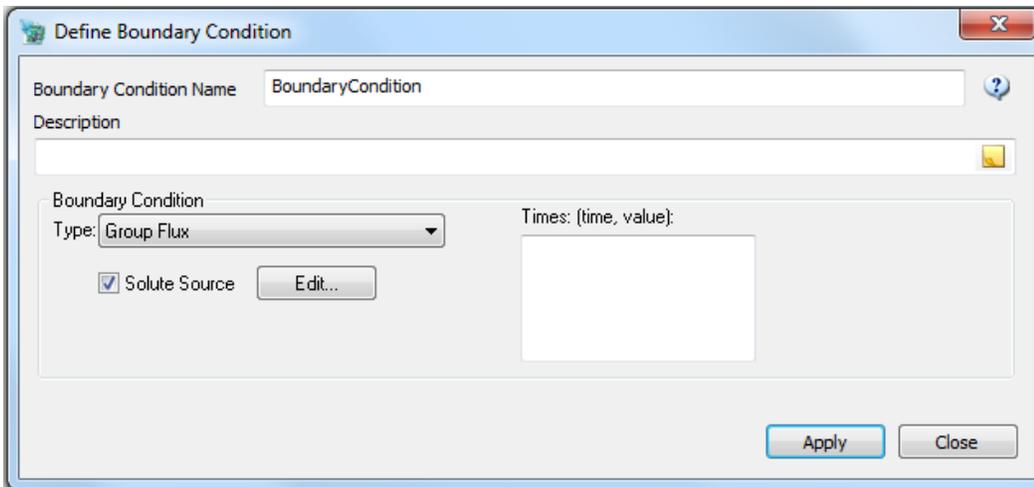


Variable Flux



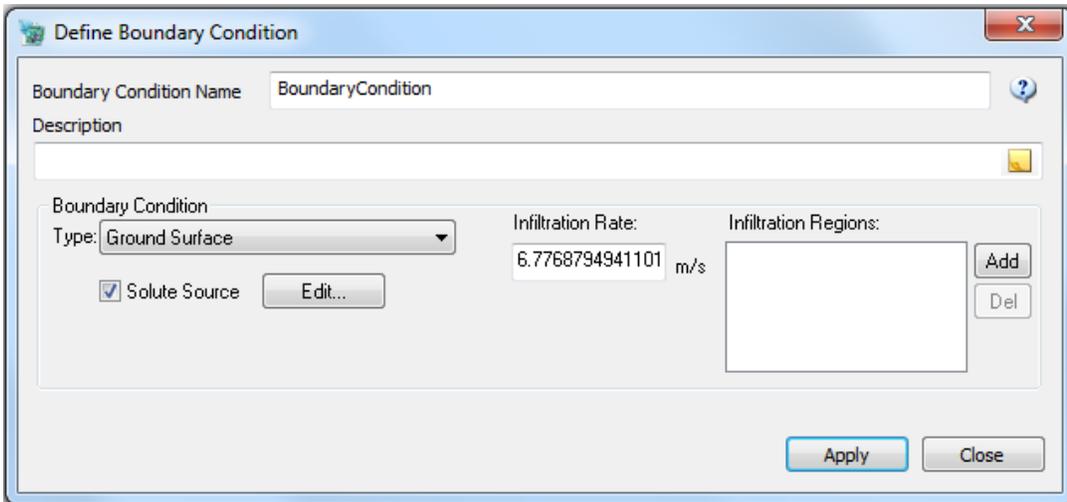
Group Flux

The group flux boundary condition allows for the specification of the total flux from all nodes within a nodal group. This allows for the simulation of a commonly occurring well boundary condition whereby the total flux into or out of the well is known but the production of individual nodes within the well is not.



Infiltration Region (Ground Surface)

Simplified phreatic-surface approximation for modeling infiltration and topographic control of the water table.



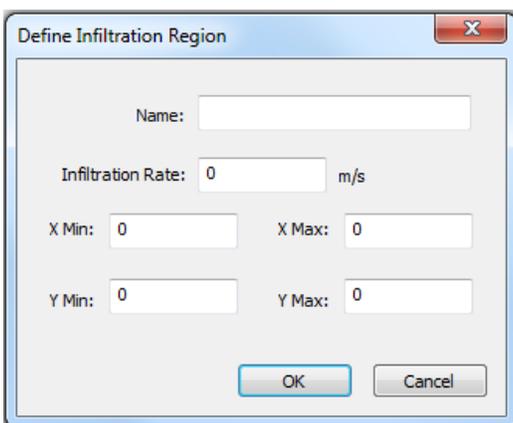
Ground-surface nodes have a variable boundary condition that depends on the location of the phreatic (head) surface with respect to the ground. There are three possible states for ground-surface nodes:

‘Swamped’, where the phreatic surface is at the ground surface and there is either a net outflow from the node to the surface, or the net inflow is less than the available infiltration due to precipitation.

‘Vadose’, where the phreatic surface is within 1 meter of the ground surface. In this case there is a net infiltration rate at the node, which varies from 0 m/s if the phreatic surface is at the ground surface, up to the specified maximum infiltration rate if the phreatic surface is 1m below the ground surface.

‘Free’, where the phreatic surface is more than 1 m below the ground surface, and the specified maximum infiltration rate is applied to the node.

The maximum regional infiltration rate, in m/s, is specified by the user as the INFILT parameter in the \$MAFIC namelist input. This rate is then applied to the surfaces of fracture or matrix elements when all nodes on a surface are specified as having the ground-surface boundary type.

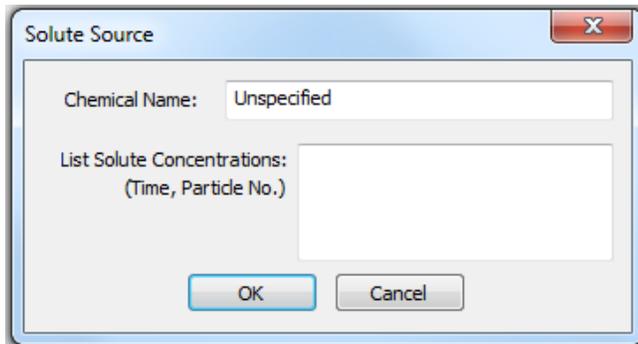


When ground-surface nodes are specified in a problem, MAFIC automatically iterates to find a solution where each ground-surface nodes satisfies one of the above three conditions.

Head-linked nodes are zero-flux nodes that are identified as having the same head as another node. Head-linked nodes allow the user to link nodes that are adjacent in space but that do not have explicit elements connecting them, for example fracture and matrix nodes. The linkage of the heads is equivalent to specifying an infinite connection between the nodes.

Solute Source

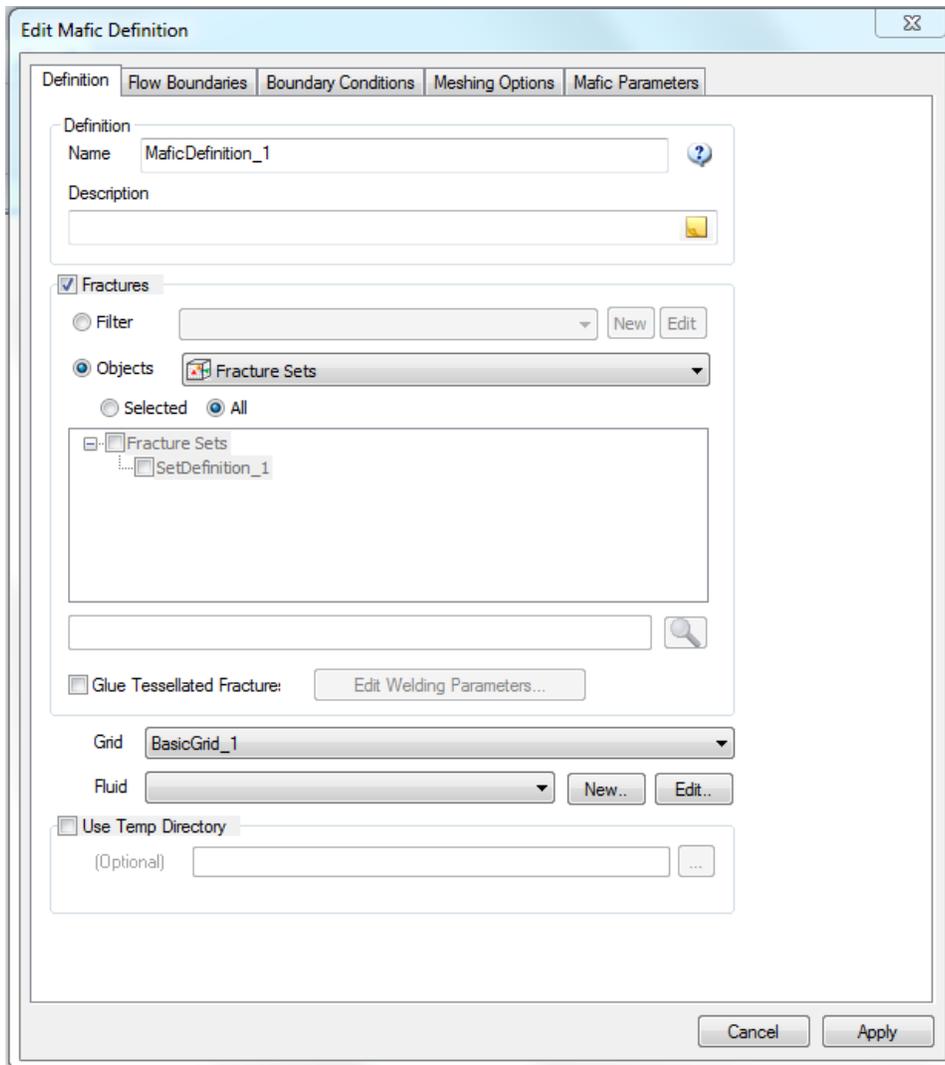
Solute sources are specified as the concentration of a chemical species of the in-flowing water at a nodal group. More than one chemical species may be released at any nodal group.



Defining a MAFIC simulation

The MAFIC interface inside FracMan enables the user to specify boundary conditions, input parameters (i.e. mesh element size), and type of simulation in one place over a series of tabs.

On the first tab, the name of the simulation is given, fractures are assigned (Note: one can use the default, using all fractures in the model, or a filter can be specified to only include a subset of fractures).



Name: Keep it short, this name is also used as a prefix to identify the output files.

Description: Optional text to describe definition or log usage.

Glue Tessellated Fracture: Use this option to optimize performance if tessellated fractures are included in the MAFIC simulation. See Manual section [Weld Fracture Faces](#) for details of feature and parameters specified using the **Edit Welding Parameters** button.

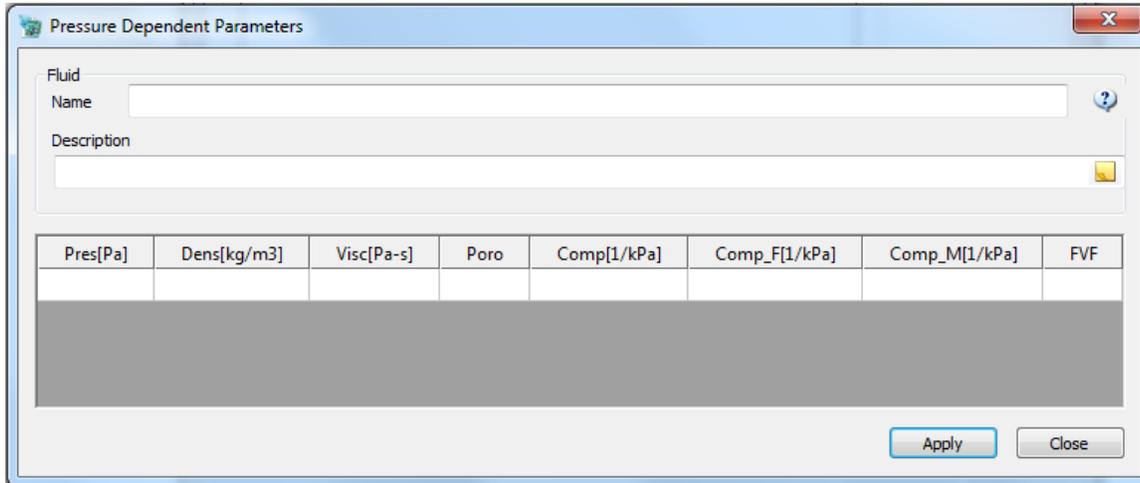
Grid: Select grid for matrix properties if running a dual-porosity simulation *and* you either need spatially varying matrix properties or you want to run the **fully discretized matrix approach** (see Mafic Manual for details). Generally the number of fractures in the model must be relatively small. The grid must have the properties “Porosity” and “Permeability”. It’s also possible to use one which has permeability generated using the [Define Mafic EPM/DFN Model](#) functionality. Dual-porosity simulations may also be run using the **matrix block approach** where you specify matrix properties on the Mafic Parameters tab.

Fluid: Enter pressure dependent fluid and matrix properties if desired. If not used, the constant fluid properties from the [Project Units](#) tab are used. If data is available for fluid properties as a function of pressure, select an existing Fluid definition or click on **New** and enter the values in the [Pressure Dependent Parameters](#) dialog.

Use Temp Directory: Select a directory to hold temporary files. The filename and path must not include spaces. If not entered, the users default temp directory will be used (for example, C:\Users\username\AppData\Local\Temp”).

Pressure Dependent Parameters

Enter fluid property values at known pressures spanning the expected range and MAFIC will interpolate between these data points to determine properties at any given pressure. Units for the columns can be changed by right-clicking or double-clicking on the headers.



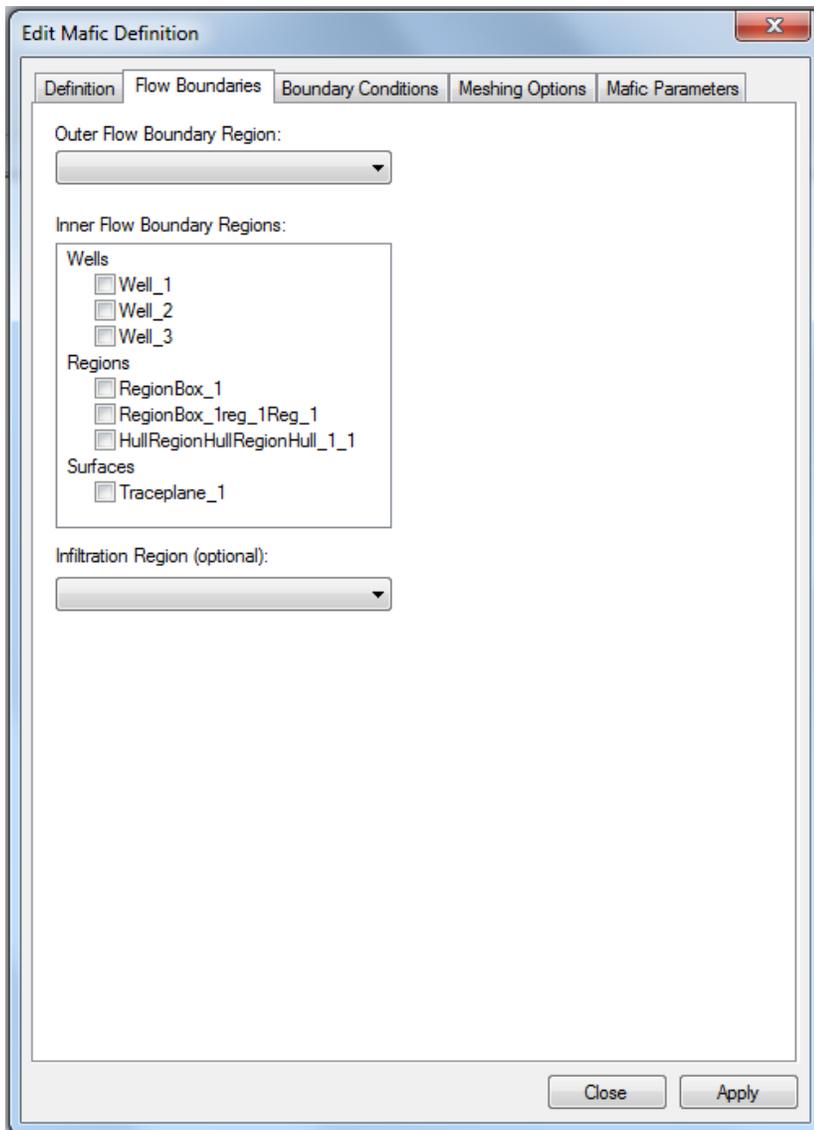
Pres[Pa]	Dens[kg/m3]	Visc[Pa-s]	Poro	Comp[1/kPa]	Comp_F[1/kPa]	Comp_M[1/kPa]	FVF

- **Pres:** Pressure
- **Dens:** Density of the fluid
- **Visc:** Viscosity of the fluid
- **Poro:** Porosity of the matrix
- **Comp:** Fracture compressibility
- **Comp_F:** Fluid compressibility
- **Comp_M:** Matrix compressibility
- **FVF:** Formation Volume Factor

See Manual section [Reserved Properties](#) for details on how these parameters are used to convert between TSA and PCA properties. The Mafic backend uses TSA for all calculations and head instead of pressure.

Mafic Flow Boundaries

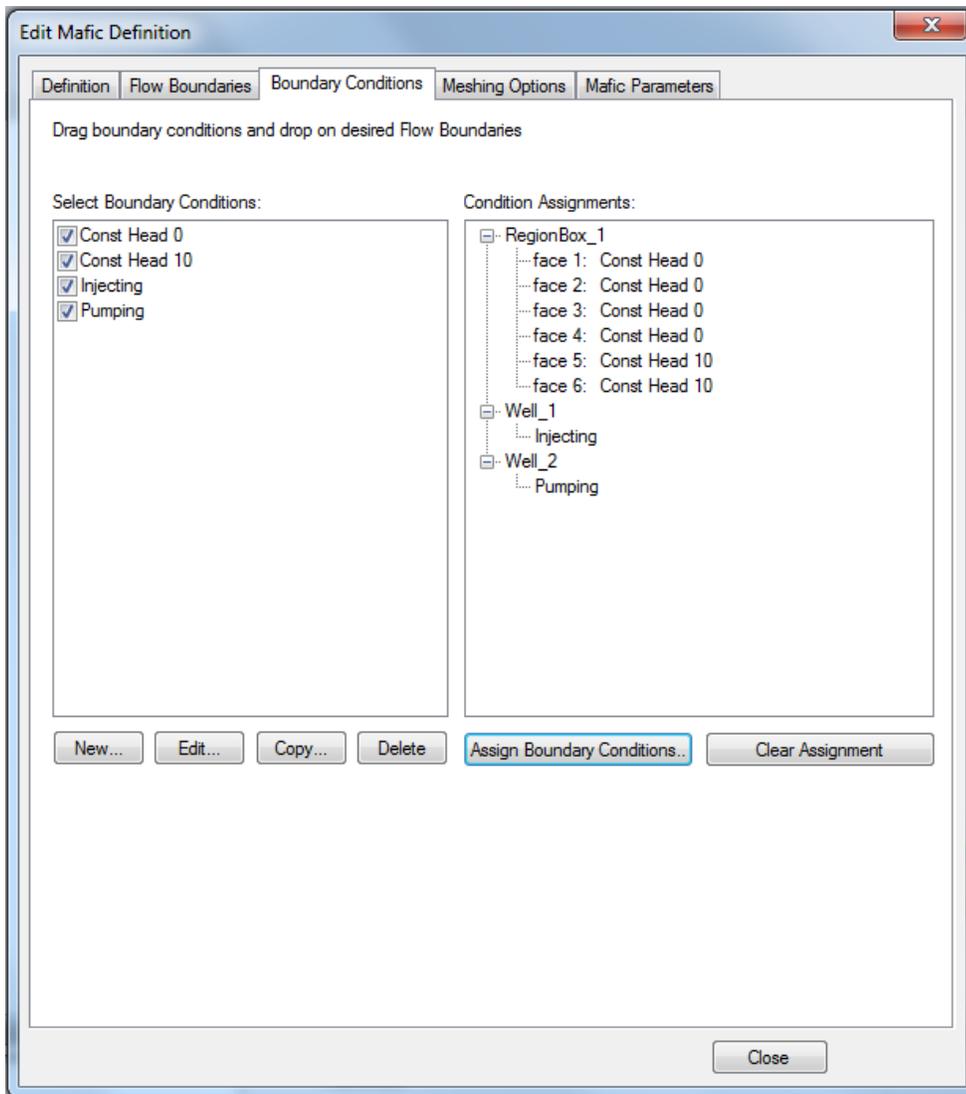
In the “Flow Boundaries” the outer boundary region MUST be selected, in many cases is a region box, and the OPTIONAL inner flow boundaries are chosen using any remaining objects in the FracMan file. It should be noted that Meshing occurs INSIDE outer flow boundaries, BUT NOT INSIDE inner flow boundaries. A surface, such as a surface imported from GoCAD, can also be assigned here if the infiltration feature in MAFIC will be used.



Assigning Boundary Conditions

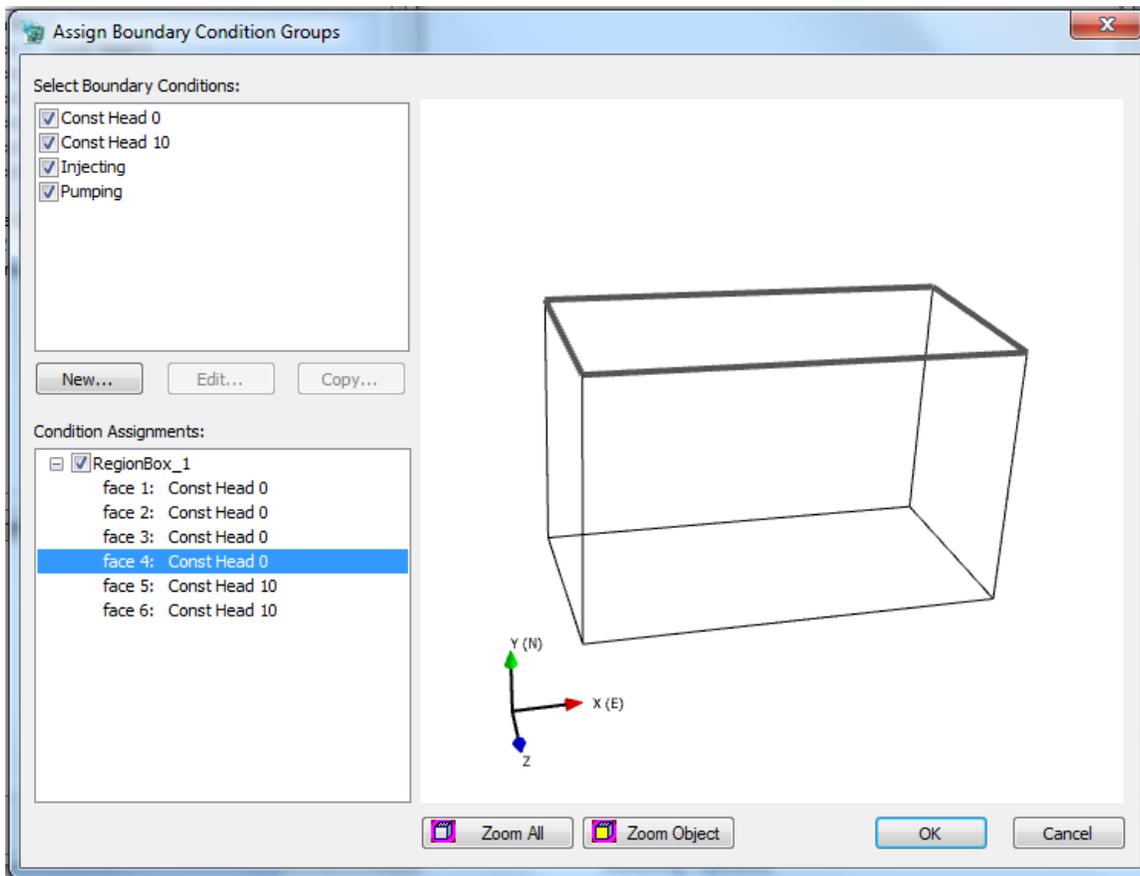
Once the flow boundaries have been specified you must apply boundary conditions to them. You can drag the available boundary conditions from the list on the left and drop them on the boundary objects on the right. You can specify different conditions to different faces of most boundary objects.

To visualize the boundary conditions and the faces they are assigned to, press **Assign Boundary Conditions**. The button **Clear Assignment** removes the selected assignment in the **Condition Assignments** window.



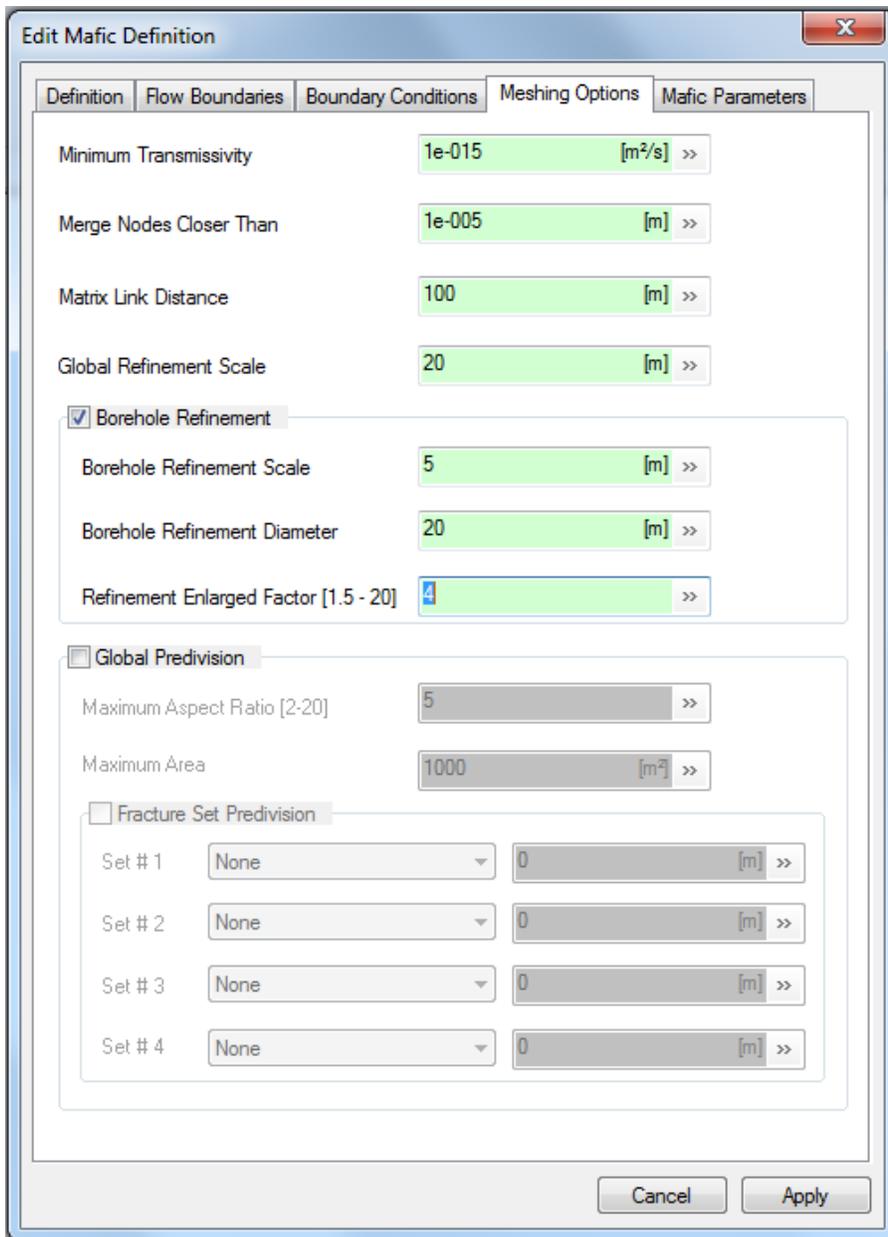
Assign Boundary Condition Groups

This dialog allows a visualization of where boundary conditions are being assigned. You can drag a boundary condition from the **Select Boundary Conditions** window to a location in the **Condition Assignments** window as in the parent boundary Conditions tab, but also see where in the model this is happening. Selecting a boundary condition will cause the faces assigned this condition to appear in yellow. Selecting a face in the **Condition Assignments** window will highlight that face in the visualization window. The mouse may be used to rotate the model in the visualization window. The **Zoom All** and **Zoom Object** buttons are useful for large, complicated models.



Meshing Options

These options are used by the MeshMaster (mesh.exe) program to build a mesh for the flow simulation. For a complete description of all the parameters and capabilities of MeshMaster see the MeshMaster Manual, "MeshMaster.pdf", located in the FracMan7 directory.



- **Minimum Transmissivity** - minimum transmissivity to filter out those fractures whose transmissivity is low enough that their contribution to the set is insignificant and negligible.
- **Merge Nodes Closer Than** - This distance refers to the minimum distance allowed between two nodes. If the distance between two nodes is less than this minimum value, the two nodes are merged. The process of merging nodes is fraught with danger. On one hand, long and skinny elements cannot be solved correctly in the finite element solution. On the other hand, merging the adjacent nodes which produce long and skinny elements can produce other problems such as incorrect boundary condition assignments and occasional holes in the mesh. The distance the user selects should depend on the scale of the problem. For example, in a 10,000 meter scale model, it may be appropriate to merge nodes less than one meter apart. While in a 10 meter scale model the merge nodes criteria might be 0.01 meters. Also, be aware that

the value you assign to merge nodes is sufficiently smaller than the lengths of the borehole sides. This assures that there will be no unwanted merging of the borehole nodes.

- **Matrix Link Distance** –In MAFIC 2D finite elements on fracture-medium are linked to 3D volumetric elements in rock matrix by nodes (vertices of finite elements). The link distance (link_distance) is specified by the user. Its value is recommended as one-half of the diagonal length of the 3D element. This way any nodes in 2D elements near a 3D element can be linked to rock matrix.
- **Global Refinement Scale** –The specified refinement scale (Big_Scale) will be applied to all fracture surfaces. The triangular elements after refinement will have side length no more than “Big_Scale”
- **Borehole Refinement Scale** – The specified distance (Fine_Distance) will be used to determine if the finite elements are close to borehole or not. If it is close to borehole, the refinement scale “Fine_Scale” will be applied. The distance between element and borehole is calculated from the center point of the element to the borehole axis.

Normally “Fine_Distance” should be greater than or equal to “Big_Scale”.

“Fine_Distance” >= “Big_Scale” > “Fine_Scale”

- **Borehole Refinement Diameter** - The degree of mesh refinement near boreholes needs to be watched carefully. In addition, be careful to ensure that the refinement regions for adjacent boreholes do not overlap. Overlapping borehole regions can produce unsatisfactory meshes. The maximum value for adjacent borehole regions should be less than half the distance between the borehole centers.
- **Refinement Enlarged Factor** – To connect every edge and corner of element to its adjacent element we need a scale-up parameter for elements in the “Fine_Distance” range to elements in global. This input parameter (Scaling_Factor) is used to scale up refinement scale from “Fine_Scale” to “Big_Scale”. The larger scale-up factor would reduce the total number of elements, but increases the aspect-ratio of elements. The plot shown below has Scaling_Factor = 2 at the upper-left, 3 at upper-middle and 4 at the upper-right.
- **Maximum Aspect Ratio** – This specified aspect ratio will be applied to determine the maximum aspect ratio after mesh predivision. This ratio should be between 2-20 for accurate finite element solution.
- **Maximum Area** – Specify the maximum mesh area after predivision.
- **Fracture Set Predivision** – Specify the average element size for fracture set predivision.

Mafic Parameters Tab

Edit Mafic Definition

Definition | Flow Boundaries | Boundary Conditions | Meshing Options | **Mafic Parameters**

Output: Summary Full Output Output Last Timestep
 Output All Timesteps

Elements: Linear Quadratic

Flow: Steady Transient 1D Simulation

Transport No Solute Solute [hrs] - 0 timesteps

Parameters for Solute Transport

Random seed: 12321

Track All Particles

L dispersivity: 1 [m] Particle mass: 1e-008 [kg]
T dispersivity: 1 [m] Fluid density: 1000 [kg/m3]
Diffusion coefficient: 1e-009 (e.g., 1.0e-09 for water)

Ref Pressure: 0 [Pa] Ref Pressure Z: 0 [m]
Ref Pressure Gradient: -9792 [Pa/m]

Initial Head: $H(X,Y,Z) = H_xX + H_yY + H_zZ + H_o$ [m]
Hx: 0 Hy: 0 Hz: 0 Ho: 0

Initial Flux: $Q(X,Y,Z) = Q_xX + Q_yY + Q_zZ + Q_o$ [m3/s]
Qx: 0 Qy: 0 Qz: 0 Qo: 0

of nodes to plot: 0 Max Iterations: 5000
Tolerance: 1e-007 [m] Extrapolation [0-1]: 1
Matrix element: Fully discretized tetrahedral element [Edit Properties...]

Cancel Apply

Output – The basic input summary provides information for problem identification and size requirements. Element type, matrix-block details, and group boundary values are also indicated by this output segment. Full Output includes Nodal fluxes and heads at the end of every timestep. Nodal and element input data are printed when full output is specified. This output segment is used to check input data and/or provide identification of problem geometry.

Elements – Choice of linear or quadratic basis functions for approximation of head fields of element edges.

Flow - Solute transport modeling with either steady-state or transient flow conditions.

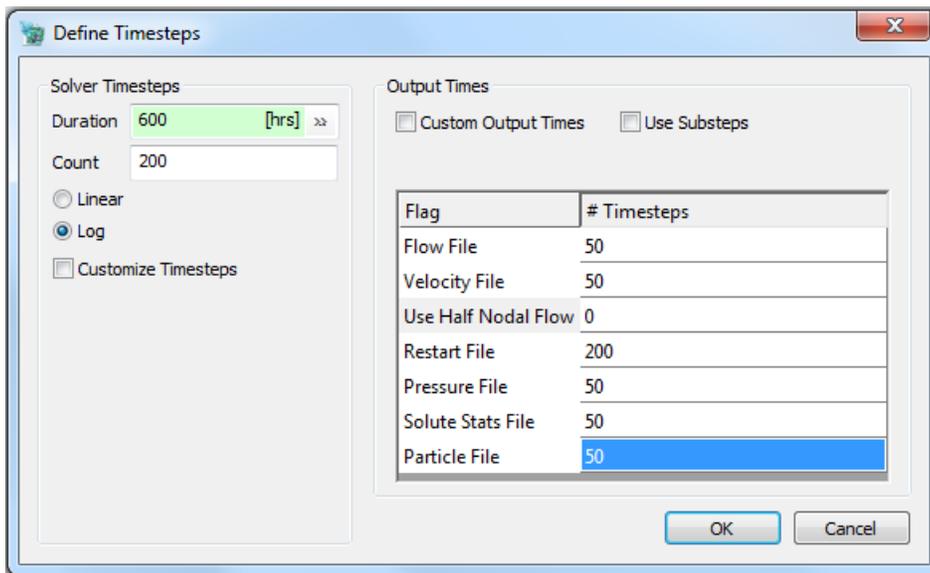
Transport – Select the Solute/No Solute mode for the simulation. If Solute option is specified, the particle properties, such as dispersivity, mass and fluid density, should be provided.

Output Last Timestep – Toggling this will generate an additional mesh that will contain the pressure and flow solution for the MAFIC simulation parameters and boundary conditions that you have defined. Other input parameters (i.e. Aperture, and Transmissivity) will also get written to this mesh. This mesh gets the filename “restart.rst” which will have to be imported manually into FracMan after the flow simulation is run.

Output All Timesteps - This will generate a mesh file for each time step.

1D Simulation – One dimensional pipe flow simulation between fractures.

Timesteps: Define the timesteps using the button below the 1D Simulation checkbox if **Transient Flow** or **Solute Transport** options are selected to define both the timesteps used for the simulation and the frequency of output file creation. The **Define Timesteps** dialog will open:



Solver Timesteps

Duration: Enter the total time.

Count: Enter how many timesteps are desired if timesteps are to be automatically chosen.

Linear or Log: Used to determine spacing between timesteps if timesteps are to be automatically chosen.

Customize Timesteps: Opens extra table on current dialog to manually enter or paste desired timesteps as elapsed time from the start. If this option is checked, the **Count** and **Linear/Log** fields above are ignored.

Output Times

Custom Output Times: Opens separate window to manually enter or paste desired timesteps.

Use Substeps: Enter the number of calculation substeps desired for each timestep. The default is 1.

Timesteps: Indicate how many timesteps between output of the following files:

- **Flow File:** This is the summary output file “FlowRate.out” containing group boundary flow rates and heads.

- **Velocity File:** A velocity file “velocity.dat” is created which contains the time(s) and element centroid positions and fluid velocities. For transient flow simulations, flow fields at different timesteps are included in the same file.
- **Use Half Nodal Flow:** Report both flow into and flow out of internal nodes.
- **Restart File:** A restart file “restart.maf” is created which echoes the input file of the current simulation with initial heads and fluxes given by the heads and fluxes at the run completion. The restart file may be used as a new input file for a new run as is, or may be edited to accommodate changes in boundary condition type and/or location.
- **Pressure File:** Time vs head “plot.dat”.
- **Solute Stats File:** (Solute) Solute particle statistics.
- **Particle File:** (Solute) The file “particle.pcl” is created which includes the time and particle locations.

Parameters for Solute Transport

Random Seed: Seed for the random number generator used for the solute transport functionality.

Track All Particles:

L and T dispersivity: Lateral and transverse dispersivities of the fractures. Can be positive, negative or zero. A positive value is representative of the fracture media as a whole. A zero value means no lateral or no transverse dispersive motion. A negative value indicates the dispersivity specified individually for each element.

Particle mass: Mass of a single particle used in the simulation. The smaller the particle mass is the greater the total number of particles generated at the solute sources. More particles mean greater accuracy as well as greater computational effort.

Fluid density: Density of the fluid which is transporting the solute.

Diffusion coefficient: Diffusivity of the solute in the fluid. For groundwater, the diffusivity value is generally 10^{-9} to 10^{-10} m²/s.

Ref Pressure: Pressure added to head value of nodes.

Ref Pressure Z: Reference pressure depth used in combination with the reference pressure gradient when pressure dependent fluids are used.

Ref Pressure Gradient: Reference pressure gradient used along with the reference pressure depth when pressure dependent fluids are used.

Initial Head: Initial head for nodes which are not included in the boundary conditions.

Initial Flux: Initial flux for nodes which are not included in the boundary conditions.

nodes to plot: Used for reporting well testing results. These are the number of nodes on the well and can be between 0 and 50.

Tolerance: The maximum head change per iteration to signal iterative solution convergence. If a value less than $1e-10$ is entered, it will be reset to $1e-10$. A convergence tolerance of 0.001 to 0.00001 is generally sufficient to achieve proper solution convergence. Convergence occurs when the largest nodal head change is less than the tolerance value times the highest minus the lowest nodal head, and when the largest unbalanced nodal flow rate is less than the tolerance value times the sum of all flows through the system, or when the heads cease to change.

Max Iterations: The maximum number of iterations allowed per timestep to achieve the convergence tolerance. If the convergence tolerance is not reached after this number then a warning message is printed, and program execution continues. The maximum number of iterations per timestep is normally 100 to 1000 iterations; this number increases as the convergence tolerance is decreased, timestep size is increased, or problem size is increased. If this value is set to zero, the simulation will be terminated after checking the data and printing out the memory requirements.

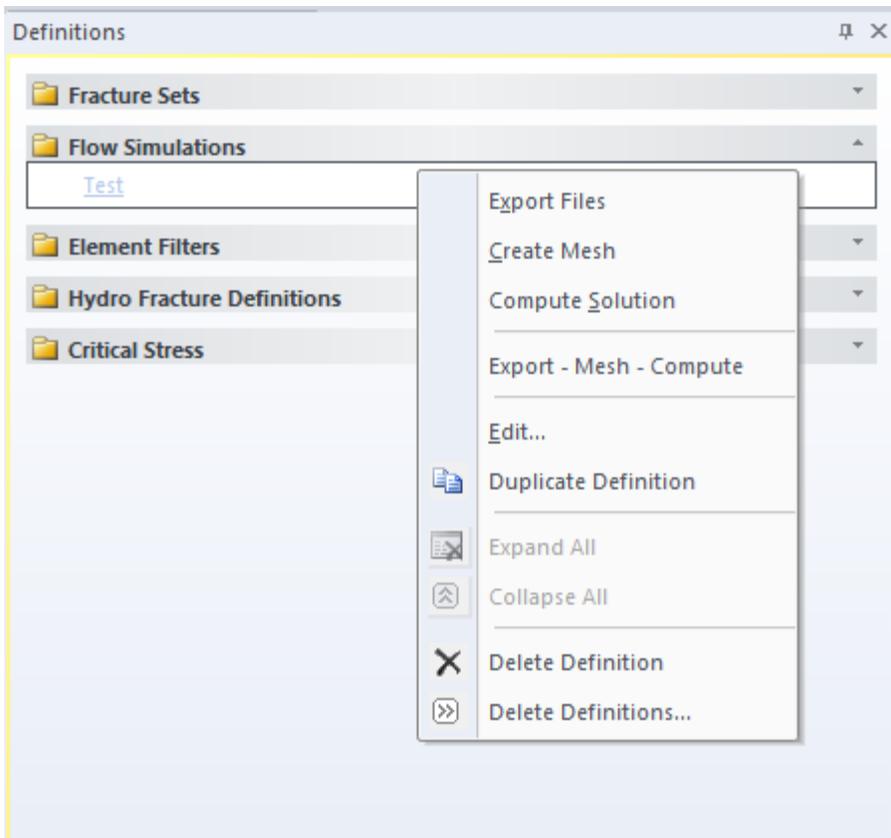
Extrapolation [0-1]: Convergence may be speeded following the first timestep by using a variable extrapolation factor which estimates head values at the end of the current timestep by extrapolating head changes from the previous timestep. When the extrapolation value equals zero no extrapolation is conducted. When the extrapolation value equals 1, the head gradient is extrapolated one timestep. The extrapolation value should be positive and less than or equal to 1.

Matrix element: If a grid was selected on the Well Test tab, then the matrix element will be fixed at “Fully discretized tetrahedral element”. Otherwise, you may choose to ignore the matrix component by leaving the choice at “None”. If you would like to run a dual-porosity simulation using the **matrix block approach**, select either “Spherical matrix block” or “Rectangular slab matrix block”. Slabs are suitable for predominantly parallel fractures and spheres are suitable for networks composed of orthogonal fracture sets or randomly oriented fractures. If a matrix block element is selected, then the matrix properties should be specified using the **Edit Properties** button:

Property	Value	Unit	Lock
Hydraulic Conductance	1e-011	[m/s]	Locked
Specific Storage	1e-007	[1/m]	Locked
Porosity Factor [0-1]	0.04		Locked
Initial Head	0	[m]	Locked
Element Radius / Thickness	4	[m]	Locked
Number of Elements in Block	1		Locked

Running MAFIC

Once you have defined your MAFIC definition along with your boundary conditions, you are ready to run your MAFIC simulation. Select your MAFIC Definition for the following options:



Export Files

This is your first step and this must be redone every time you change anything in the FracMan model or in the definition parameters. This step will export all the data necessary for the remaining steps.

Create Mesh

This step will run Meshmaster (mesh.exe) to convert your model into a MAFIC mesh.

Import Mesh (optional)

You can import and visualize your mesh if you would like to see the results of the meshing step.

Mesh Optimization (optional)

This optional step will run EdMesh on your mesh to merge vertices closer than $1e-4$ and delete triangles smaller than $1e-10$.

Compute Solution

This step will run MAFIC to compute your flow solution

Import Timestep Pressure(s)

The timestep pressure(s) can be imported for visualization by selecting "Import Timestep Pressures". The flow and pressure at each timestep will be available as a mesh property for visualization or export.

Import Particle Track

If the MAFIC simulation included particle tracking the tracks.pcl file can be imported as a point data set for visualization etc.

Troubleshooting MAFIC

This section will attempt to answer, or help with, some of the major problems that one might come across when putting a flow simulation using MAFIC.

Non-convergence

This is one of the most common problems experienced by many flow codes (i.e. MAFIC, MODFLOW) and is mainly caused by inappropriate specifications of boundary conditions and/or input parameters (i.e. transmissivity). In a steady-state solution, non-convergence means that the mass balance (i.e. “what goes in must come out”) was not achieved. Most non-convergence issues can be fixed by making an adjustment in the one of the input parameters or modifying one of the boundary conditions. Most of the time non-convergence is created by not having boundary conditions set up correctly, thus checking boundary conditions is the first thing one should do.

Meshing Errors

Sometimes when dealing with complicated fracture networks or boundary conditions (as this the case with most geologic environments meshing errors tend to occur. Errors will not be detected until MAFIC runs, because almost anything can get meshed, just not meshed in a way where MAFIC can use it. Most of the time, meshing errors is the result of having node elements with little or no area. This along with other meshing errors can be fixed using a program that was included with your installation called EdMESH.

EdMESH is a program that can be used for mesh refinement. Given the example in the paragraph above, one can input the mesh and refine it using EdMESH with just a few steps. See images below for an example of how to use EdMESH to delete ugly meshes..

Disclaimer: EdMESH runs in DOS, so one must be familiar with using the command prompt and navigating to the working directory along with calling on EdMESH from the working directory, which will usually be stored in the FracMan program files folder. If one does not know how to do this, one can copy the EdMESH.exe from the program files FracMan folder and paste it into the working directory. After pasting EdMESH into the working directory then one can run EdMESH by simply double-clicking on it.

Delete Meshes using EdMESH

```
C:\ToddsTest\Trans\edmesh.exe
EdMesh Version 1.830
MAFIC Mesh Editing Utility
Copyright Golder Associates Inc., 2009

Input file type (1=MAFIC .maf, 2=MaficOil .cfg, 3=.sab) > 1

Enter name of MAFIC file to change > TransSim.mff
Problem Title > TransSim.mff

**** MAIN MENU ****

<0> Quit without saving changes!
<1> Set MAFIC solution parameters
<2> Assign fracture properties
<3> Refine, merge or grout fracture meshes
<4> Assign matrix element properties
<5> Create zone or delete matrix elements
<6> Save EdMesh file
<7> Run EdMesh

Selections > 3

**** SUB-DIVIDE/DELETE/MERGE MESHES ****

<0> Quit, return to main menu
<1> Refine meshes
<2> Delete meshes
<3> Merge nasty meshes
<4> Inject grout to meshes connecting boundary
<5> Check nodal group numbers

Selections > 2
```

```
C:\ToddsTest\Trans\edmesh.exe
To delete meshes due to
(1) small property value, (2) small fracture area, (3) small mesh area
Enter choice 1, 2 or 3 > 3
Minimum area > 0

**** SELECT REGION ****

(0) Back to previous menu
(1) All 3-D space
(2) Fracture numbers
(3) Fracture sets
(4) Box

Selection Region > 1

**** MAIN MENU ****

(0) Quit without saving changes!
(1) Set MAFIC solution parameters
(2) Assign fracture properties
(3) Refine, merge or grout fracture meshes
(4) Assign matrix element properties
(5) Create zone or delete matrix elements
(6) Save EdMesh file
(7) Run EdMesh

Selections > 7
Seed value for random numbers > 1
```

After this step, hitting enter, EdMesh will go through the mesh and delete any node elements with areas equal to zero.

Define Mafic EPM/DFN Model

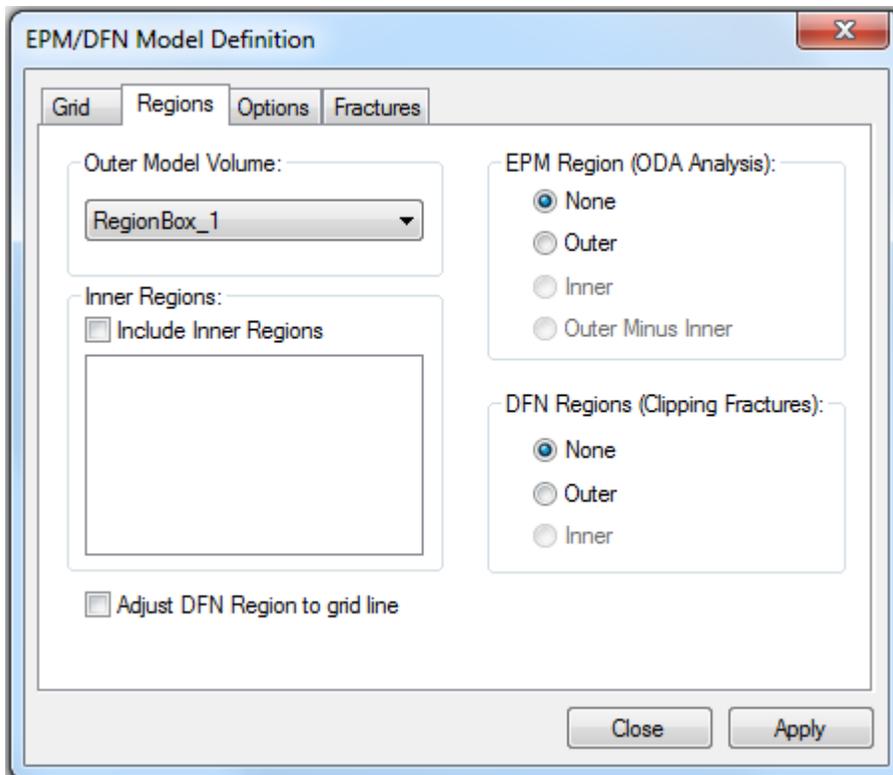
When building a MAFIC flow simulation for a large model it is often advantageous to use a hybrid equivalent porous medium (EPM)/DFN approach. FracMan can help by creating a model that combines a large scale EPM grid with a small scale DFN model. Select **Dynamic Analysis | Mafic | Define Mafic EPM/DFN Model** from the top menu.

The screenshot shows the 'EPM/DFN Model Definition' dialog box with the following fields and options:

- Grid** tab selected.
- Name:** EpmDfn_1
- Description:** (empty)
- Grid:** BasicGrid_1
- Define Z-domain (non-regular grids only):**
 - Original Grid
 - Below original grid
 - New regular grid
- Z Max:** 108.235 [m]
- Z Min:** -108.235 [m]
- Number of Layers:** 0

Buttons: Close, Apply

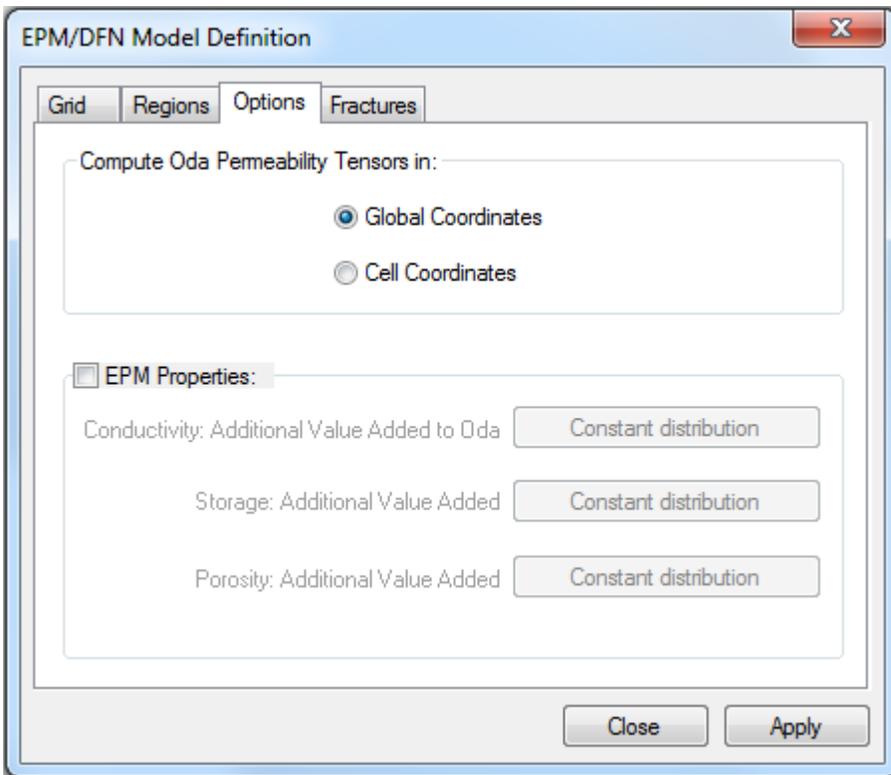
Name the EPM/DFN hybrid definition and select an EPM grid.



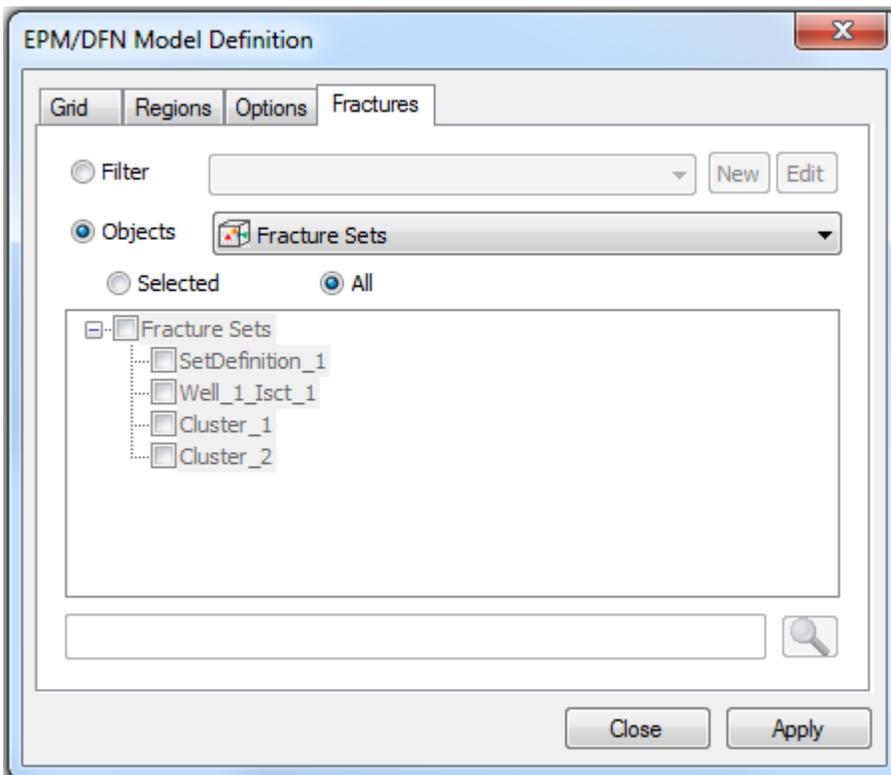
Select an **Outer Model Volume** and (optionally) one or more **Inner Regions**. Then specify the EPM and DFN regions. The DFN region will be used to clip the fractures creating a small scale DFN to be used in MAFIC simulation or other flow simulation. The EPM region will use ODA Upscaling to compute the permeabilities of the grid cells.

Select **Adjust DFN Region to grid line** to adjust the grid cell boundaries so that they line up exactly with the DFN region.

Note: This adjustment is highly dependent on the geometry of the DFN region. It is recommended to make the DFN region extend slightly beyond the boundary of the grid. If your resulting regions don't line up correctly or the clipping is not what you expect, try adjusting the size of the DFN region slightly.



ODA upscaling can produce 0 permeability in cells that do not contain fractures, this can cause problems for flow simulation. Specify a distribution of small values for conductivity, storage and porosity that will be added to the ODA values in order to provide suitable background (matrix) property values.



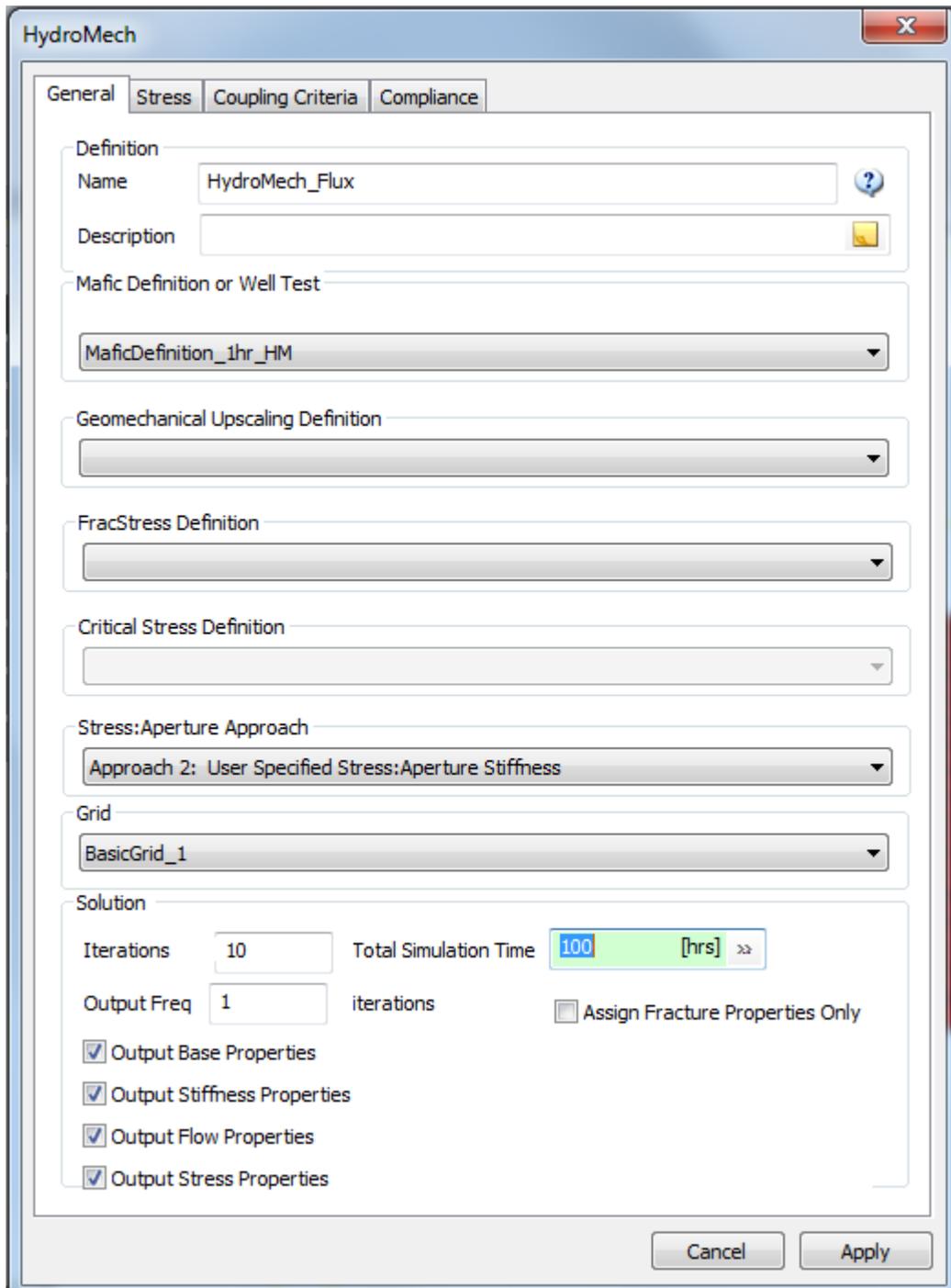
Finally, select the fracture sets to be included. These fracture sets will be clipped to the DFN region.

Fracture Upscaling

See the Manual section for Static Analysis -> Grid-Based Analyses -> [Fracture Upscaling](#).

HydroMechanical Coupling

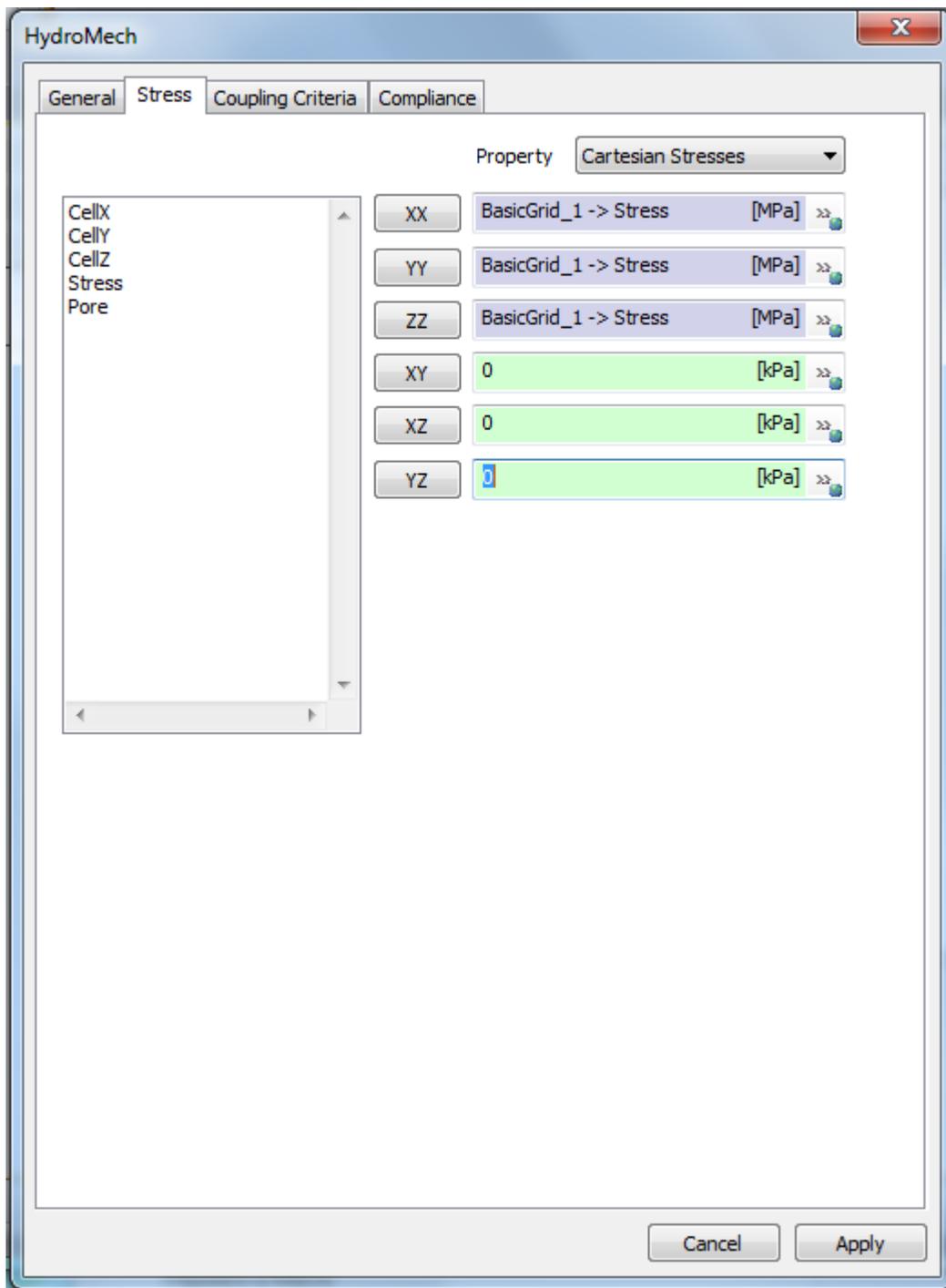
The HydroMech module allows the user to combine MAFIC well testing with evolving stress conditions.



The image shows a software dialog box titled "HydroMech" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Stress", "Coupling Criteria", and "Compliance". The "General" tab is selected and contains the following fields and options:

- Definition**
 - Name: HydroMech_Flux
 - Description: (empty)
- Mafic Definition or Well Test**: MaficDefinition_1hr_HM
- Geomechanical Upscaling Definition**: (empty)
- FracStress Definition**: (empty)
- Critical Stress Definition**: (empty)
- Stress:Aperture Approach**: Approach 2: User Specified Stress:Aperture Stiffness
- Grid**: BasicGrid_1
- Solution**
 - Iterations: 10
 - Total Simulation Time: 100 [hrs] >>
 - Output Freq: 1 iterations
 - Assign Fracture Properties Only
 - Output Base Properties
 - Output Stiffness Properties
 - Output Flow Properties
 - Output Stress Properties

At the bottom of the dialog are "Cancel" and "Apply" buttons.

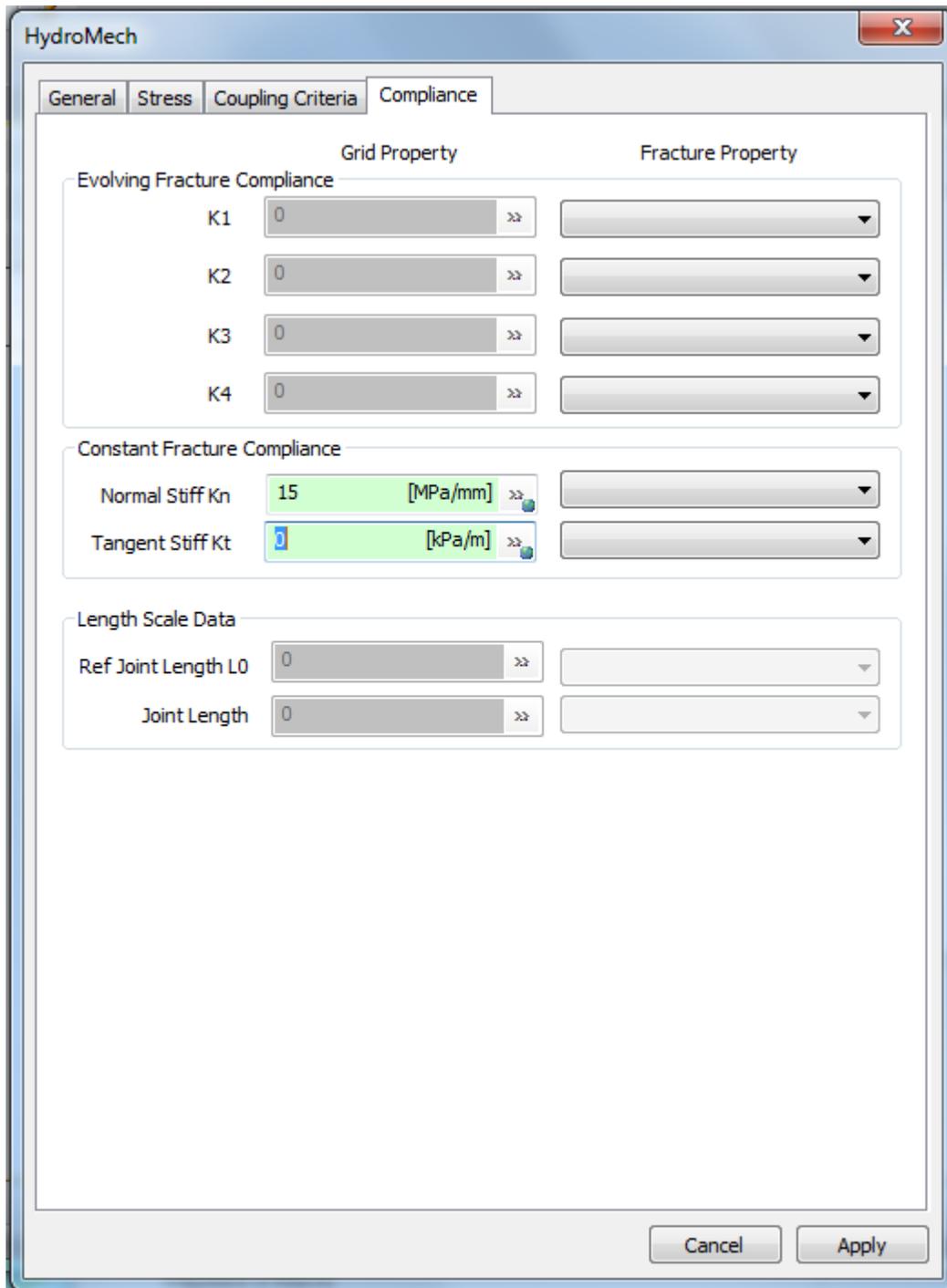


HydroMech

General Stress Coupling Criteria Compliance

	Grid Property	Fracture Property
Frictional Properties		
Friction Angle	30 [deg]	
JRC	15	
JCS	150 [MPa]	
UCS	200 [MPa]	
Fracture Closure Properties		
A	0	
B	0	
C	6.41	
D	-0.72	

Cancel Apply



Using Macros

Introduction to Macros

FracMan macros allow you to execute a series of commands and instructions automatically. Instead of manually performing a series of time-consuming, repetitive actions, you can create and run a single macro that accomplishes a series of tasks for you.

FracMan macros also allow you to loop over a set of commands. This enables you to generate multiple realizations of a fracture network and to conduct Monte Carlo simulations.

The Macro language used by FracMan is continually evolving, so Macros recorded from earlier versions of FracMan may not work correctly with newer versions. Best practice would be to record a new macro using the current software version to use as a basis for creating new or updating any Macros made from earlier versions.

You can always view the Macro output that gets generated during your FracMan session by opening the **Log** window using the **View** menu.

Recording a Macro

Recording of macros allows you to capture a series of FracMan commands and instructions to a text file. With the help of a text editor you can then edit the macro, for example, to loop over a set of commands in the macro.

Note that the macro recording feature does not record all possible actions in FracMan. It focuses on those operations that are normally used to generate and analyze fracture networks and that will typically build the basis for Monte Carlo Simulations. Rendering commands, object attribute changes (e.g. color), etc. are currently not recorded.

To Record a Macro:

On the **Tools** menu, select **Record Macro**.

A standard file browser window will appear that lets you specify the macro file name and location.

Click **Save** to begin recording.

To stop recording the macro, select **Stop Record Macro** from the **Tools** menu.

Editing a Macro

FracMan macro files are standard ASCII text files and can be edited with any standard text editor such as Notepad or WordPad.

A simple example of a macro file is show below. The section on Macro Language Syntax in this chapter of the manual contains a list of commands and their syntax that can be used in FracMan macros.

```
BEGIN SetSeed
  Seed = 1508417028
END

BEGIN CREATEREGION
  ObjectName = "Box1"
  Type = "Box"
  Center = 250.00000000000000, 1200.00000000000000, -1500.00000000000000
  Size = 1000.00000000000000, 1200.00000000000000, 350.00000000000000
END
```

```

BEGIN_LOOP (2, 6, 2)

BEGIN DefineFractureSetDefinition
  DefinitionName = "SetDef1"
  FractureModel = "REGION"
  RegionName = "Box1"
  ClipToRegion = 0
  UseSurfacePoints = 0      #0=centers, 1=surfacePts, 2=cellcenter (geocell only)
  BEGIN Properties
    Name = "Aperture"
    BEGIN Distribution
      ValueType "Aperture" Units "m"
      Type = "CONSTANT"
      Parameter "Value" = 0.001 [m]
      MinValue = 0 [m]
    END
    Name = "Permeability"
    BEGIN Distribution
      ValueType "Permeability" Units "m2"
      Type = "CONSTANT"
      Parameter "Value" = 9.869233e-013 [m2]
      MinValue = 0 [m2]
    END
    Name = "Compressibility"
    BEGIN Distribution
      ValueType "Compressibility" Units "1/kPa"
      Type = "CONSTANT"
      Parameter "Value" = 0.001 [1/kPa]
      MinValue = 0 [1/kPa]
    END
  END
  END
  *** Intensity
  Count = 200
  AdjustedByTruncation = 0
  *** Orientation
  UsePole = 1
  BEGIN OrientationDistribution
    Type = "FISHER"
    Parameter "Mean Trend" = 15
    Parameter "Mean Plunge" = 85
    Parameter "Concentration" = 15
  END
  *** Size
  BEGIN SizeDistribution
    ValueType "Length" Units "ft"
    Type = "UNIFORM"
    Parameter "Mean" = 150 [ft]
    Parameter "Deviation" = 50 [ft]
  END
  *** Shape
  NumSides = 5
  Termination = 0
END

BEGIN GenerateFractureSet
  DefinitionName = "SetDef1"

```

```

    #FractureSetName = "SetName" #Optional name for generated set - (replaces existing objects!)
END

BEGIN ExportFile
    Path = " C:\Temp\SetDef1_%d.fab"
    TypeName = "FAB"
    Object = "SetDef1_1"
END

BEGIN DeleteObjects
    Name = "SetDef1_1"
END

END_LOOP

```

Loops and Monte Carlo Simulation

FracMan allows you to repetitively execute groups of commands. This allows you to generate different realizations of your model and it therefore can be used to carry out Monte Carlo simulations. To loop over a group of commands enter the BEGIN_LOOP keyword before and the END_LOOP keyword after the group.

```

BEGIN_LOOP (1, 5, 1)
    ...
    ...
END_LOOP

```

The BEGIN_LOOP keyword has three parameters that represent the **beginning**, the **end** and the **step size** for the loop. The resulting number can also be used to increment names within the commands that are included in the loop. The placeholder “%d” gets replaced by the current loop number.

Example:

```

BEGIN_LOOP (1, 3, 1)
    Path = "C:\Temp\fracset%d.fab"
END_LOOP

```

becomes:

```

Path = "C:\Temp\fracset1.fab"
Path = "C:\Temp\fracset2.fab"
Path = "C:\Temp\fracset3.fab"

```

If you want to loop over floating point values instead of integer values, use the “%f” placeholder. Here your BEGIN_LOOP statement would look like BEGIN_LOOP(1.5, 10.5, 0.25) and the “%f” in the Macro would get replaced by 1.5, 1.75, 2.0, 2.25, ..., 10.5.

Playback a Macro

FracMan allows you to playback macros from a text file. This runs the Macro and will execute the commands in the project.

To Playback a Macro:

On the **Tools** menu, select **Run Macro**.

From the file browser window, select the macro file you wish to playback.

Ignore Macro Errors

On the **Tools** menu, select **Ignore Macro Errors** to toggle on/off the functionality. If toggled on, it will force the program to attempt to continue even when an error is encountered. This is often useful for very long running macros where an error should not prevent additional steps from running correctly.

Macro Language Syntax

FracMan macros are ASCII text files which specify a series of commands that are executed by the program without user interaction. Macro files allow the user to automate repetitive tasks and perform Monte Carlo simulation. The macro file must be a plain text file that does not contain any formatting. Such a file can be created by a standard text editor, such as Notepad. They generally have a *.fmf extension but that is not required. The FracMan macro language is not case sensitive.

Comments

Lines that start with pound '#' character are comments. For example:

```
# This line contains a comment and is ignored by
# FracMan's macro reader.
```

Comments can also be placed in-line where everything following the '#' sign is ignored:

```
Action = 1    #1=replace, 2=add, 3=rem, 4=toggle
```

Object Blocks

FracMan uses an object-oriented approach for its macro files. The definition of an object begins with the `BEGIN` keyword and the specification of the object type. Objects can be actual objects of a FracMan model, such as a region or a surface, or they can be operations on actual objects (e.g. deletion of a region). The following lines contain all the keywords and parameters that apply to the object. The definition of the object is closed by the `END` keyword. Object blocks can be nested. For example:

```
BEGIN DefineFractureSetDefinition
...
...
  BEGIN SizeDistribution
    ...
    ...
  END
END
```

Keywords and Parameters

The FracMan macro language utilizes keyword parameter pairs to input information. The keyword and the associated parameter(s) have to be on the same line. The keyword and the parameter(s) are separated by an equals sign (=). If more than one parameter is needed, those parameters need to be separated by commas. For example:

```
ObjectName = "Box1"
Center = 250, 1200, -1500
```

Macro Parameter Types

Input parameters need to have a specific type. The fracture set name, for example, is interpreted as text. The mean fracture set trend, on the other hand, must be a number. The FracMan macro language assumes the following data types:

<tag>	FracMan keyword (e.g., FISHER).
<str>	Text string (e.g., "Set 1")
<real>	Number in real format (e.g., 2.5)
<int>	Number in integer format (e.g., 3)
<bool>	Boolean (either 0 or 1)

String parameters must be enclosed in quotes. All other parameters are specified without quotes.

Special Macro Commands

The FracMan macro language supports a number of playback-only utility functions. These functions cannot be recorded but they have been introduced to improve the playback of macros.

Stop Drawing

```
BEGIN DrawingOff
END
```

Start Drawing

```
BEGIN DrawingOn
END
```

Execute a Macro

```
BEGIN RunMacro
  Path = <str>
END
```

Deletion of all Objects

```
BEGIN DeleteAll
END
```

Begin logging to a file

This is similar to selecting "Log To File" from the log bar – it will create a file which will store all progress messages and macro commands.

```
BEGIN LOGTOFILE
FILE = "filename.fml"
END
```

Stop Macro Execution

This is a special macro command which will stop the current macro from being executed. This is useful when debugging a macro, you can stop the macro and observe the state of the model at that point. There is no way to resume a macro from the stopped point.

Note: this macro command is not in the BEGIN/END format

STOP

Exit the program

This will cause FracMan to exit.

Note: this macro command is not in the BEGIN/END format

EXIT

Exit on Errors

This command will cause the program to exit if any errors are encountered.

BEGIN EXITONERROR
END

Appendix A FracMan Supported File Formats

FracMan supports a wide range of files for data import and export. Please note that the examples and descriptions of the file formats here are provided to assist you in interpreting your data files – please consult the various manufacturers for complete descriptions of the file formats.

MAFIC Mesh file (.mafic_input.mft, .rst)	Import / Export: MAFIC Mesh
Legacy MaficOil ITM_FRA (.itm_fra)	Import / Export: Fracture Set
Fracture Data files (.ffb, .fab)	Import / Export: Fracture Set / Grid
Generic Fracture Data (.txt)	Import: Fracture Set
Petrel Well Trace (.txt, .asc, .dat)	Import: Borehole Trajectory, Well Markers
GOCAD Ascii Well (.wl)	Import / Export: Borehole Trajectory, Well Markers, Depth Intervals, Well Logs
GOCAD PLIne (.pl)	Import / Export: Tracemap
GOCAD VSet (.vs)	Import / Export: Point Data
Log ASCII Standard (.las)	Import / Export: Well Logs
SAB files (.sab)	Import / Export: Borehole Trajectory, Well Intervals, Regions
ORS files (.ors)	Import / Export: Well Logs

Borehole definition files (.txt, .csv, .dat)	Import / Export: Well Trajectory
Well Interval (tops) File (.txt)	Import / Export: Well Intervals
Well Intervals Table (.txt)	Import: Well Intervals
ECLIPSE ASCII grid (.grd, .grdecl)	Import / Export: Volumetric Grids
Stress Field File (.str)	Import / Export: Volumetric Grids
3Dmove Grids (.sdb)	Import: Volumetric Grids
Rescue (.txt, .bin)	Import / Export: Volumetric Grids, Boreholes
GoCAD Sgrid (.sg)	Import: Volumetric Grids
MODFLOW Discretized Grid (.DIS)	Import: Volumetric Grids
FeFlow grid (.fpd, .fem)	Import: Volumetric Grids
Zmap+ (.zcor, .dat)	Import: Surface, Tracemap, Point Data Export: Tracemap, Point Data
AutoCAD (.dxf/.dwg)	Import / Export: Surface, Tracemap, Point Data
RMS Surface (.txt, .rms)	Import: Surface
CPS-3 Surface (.cps)	Import: Surface
Earth Vision Surface (.txt, .dat, .ev)	Import: Surface
GoCAD Surface (.ts)	Import: Surface Export: Surface, Fracture Set
Petrel Surface (.txt)	Import: Surface
Charisma Fault Stick (*.*)	Import: Surface
Polygon File Format (.ply)	Import: Surface, Tracemap, Point Data Export: Surface, Fracture Set, Point Data, Region, Well, Tracemap
ESRI Shapefile (.shp)	Import: Surface, Tracemap, Point Data Export: Surface, Fracture Set, Point Data, Region, Well, Tracemap
Maptek Vulcan(.00t_asc)	Import / Export: Surface
Space delimited column data (*.*)	Import: Point Data
Mafic Particle Track (.pcl)	Import: Point Data
Surpac String/Digital Terrain (*.str / *.dtm)	Import Point Data, Surface
Elfen Grid Format (*.mes)	Import Grid
FLAC3D Grid Format	Import / Export Grid

Wavefront OBJ format	Import / Export Point Data, Surface, Tracemap
GiD format (*.gid, *.msh)	Import Grid, Surface, Tracemap, Point Data
GSLib Property (Simplified GeoEAS)	Import / Export property values
CSV / Compressed CSV (Comma Sep Vars)	Import / Export of property values
IRAP Classic Lines	Import / Export Tracemap

MAFIC Mesh File

See the MAFIC manual, "MAFIC Manual.pdf", located in the FracMan7 directory for the complete description of the MAFIC mesh file format.

Legacy MaficOil Fractures

This is a legacy format that is provided only for loading data from legacy projects. Fracture Data Files

See [Appendix C FAB Fracture File Format](#).

Generic Fracture Data

These are space delimited column data containing fracture information. Columns required are:

X, Y, Z: the location of the fracture center

Trend / Plunge or Dip/Dip Direction: Fracture Orientation

Radius: Fracture equivalent radius

NumSides (optional) number of sides of the fracture

Set Name (optional) – fractures will be grouped into sets by this column

Property Values: additional columns will be stored as fracture properties

Lines beginning with '#' will be ignored (comments).

Tomographic Fracture Images – TFI

Petrel Well Trace

Exported well trace files from Petrel each contain a single well object with a header followed by space delimited columns.

```
# WELL TRACE FROM PETREL
# WELL NAME:          WELL_1
# WELL HEAD X-COORDINATE: 1159483.00000000
# WELL HEAD Y-COORDINATE: 1064973.00000000
# WELL KB:           1417.00000000
# WELL TYPE:         OIL
# MD AND TVD ARE REFERENCED (=0) AT KB AND INCREASE DOWNWARDS
# ANGLES ARE GIVEN IN DEGREES
#=====
=====
      MD          X          Y          Z          TVD          DX          DY          AZIM
INCL          DLS
```

```

#=====
=====
0.00000000 1159483.00000000 1064973.00000000 1417.00000000 0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.00000000
35.00000000 1159483.00000000 1064973.00000000 1382.00000000 35.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.69000000
50.00000000 1159483.00200000 1064972.99400000 1367.00001900 49.99998050 0.00161776 -0.00617532
165.32000000 0.16000000 0.66956520
75.00000000 1159483.01100000 1064972.95700000 1342.00036100 74.99963883 0.01139170 -0.04348441
165.32000000 0.42000000 0.61277515

```

GOCAD ASCII Files

All GOCAD files contain a series of objects, each defined by a Header, Body and End tag. FracMan supports reading and writing to GOCAD well, surface (TSurf), line (PLine), and Point (VSet) files. FracMan supports reading only of GOCAD SGrid files.

GOCAD Well (WellPath)

GOCAD well files have a '.wl' extension and contain a standard header followed by one or more objects containing a WREF (well reference point) and one or more VRTX (vertex), PVRTX (vertex with property) or PATH (deviation – measured depth, Actual Depth, DX and DY) entries. FracMan does not currently support the newer TVSS_PATH, TVD_PATH or PATH_CURVE_ format wells.

```

GOCAD Well 1
HEADER {
name:Well_1
}
WREF 0.000000000000 0.000000000000 100.000000000000
VRTX 0.000000000000 0.000000000000 100.000000000000
VRTX 0.000000000000 0.000000000000 -100.000000000000
PATH 300.0000000000 -200.0000000000 10.00000000 10.000000
END

```

GOCAD Stratigraphic Grid (SGrid)

GOCAD SGrids have a .sg extension and come in a number of different sub-formats and flavors. FracMan reads many of these formats as [Grid](#) objects but does not support writing grid geometry in this format.

GOCAD Surface (TSurf)

GOCAD surface files have a .ts extension and contain one or more triangulated surfaces with zero or more vertex properties. GOCAD surface files are read in as [Surface](#) objects and support writing of surface as well as [Fracture Set](#) objects. Stochastic fracture sets are encoded as a single triangulated surface in order to avoid a GOCAD limit on the number of separate surfaces in a single file.

```

GOCAD TSurf 1
HEADER {
name:Traceplane
*solid*color:0 0.4 0.8 1
}
GEOLOGICAL_TYPE fault
PROPERTIES
ESIZES 1 1 1
TFACE
VRTX 1 -53.3867436 -95.32807159 -30.20858765
VRTX 2 146.6132564 -95.32807159 -30.20858765
VRTX 3 146.6132564 95.32806396 30.20858765

```

```

VRTX 4 -53.3867436 95.32806396 30.20858765
TRGL 1 2 3
TRGL 1 3 4
END

```

GOCAD Curve (PLine)

PLine files have a '.pl' extension and contain a set of poly lines organized into segments made up of individual vertices (with or without properties). These objects are imported into and exported from FracMan [Tracemap](#) objects.

```

GOCAD PLine 1
HEADER {
name:Traceplane_traces
}
ILINE
VRTX 1 -53.38674360 0.00000000 0.00000121
VRTX 2 23.30662340 0.00000000 0.00000121
VRTX 3 46.61326403 0.00000000 0.00000121
VRTX 4 132.58427996 -0.00000000 0.00000121
SEG 1 2
SEG 2 3
SEG 3 4
END

```

GOCAD Points (VSet)

VSet files have a '.vs' extension and contain a set of points which are used to create FracMan [Point Data](#) objects. VSet files begin with a standard GOCAD header with the type name "VSet", followed by one or more objects consisting of either VRTX (vertex) or PVRTX (vertex with property) items.

```

GOCAD VSet 1
HEADER {
name:BasicGrid_Points
}
SUBVSET
PVRTX 1 -50.00000000 -50.00000000 0.00000000 -50.000000 -50.000000 0.000000
PVRTX 1 -50.00000000 50.00000000 0.00000000 -50.000000 50.000000 0.000000
PVRTX 1 50.00000000 -50.00000000 0.00000000 50.000000 -50.000000 0.000000
PVRTX 1 50.00000000 50.00000000 0.00000000 50.000000 50.000000 0.000000
END

```

Log ASCII Standard (LAS)

A complete description of the Canadian Well Logging Society LAS standard can be found on their website at <http://www.cwls.org>. SAB Files

SAB files contain wells, box/cylinder/prism regions and MAFIC simulation definitions. A complete description of these files can be found in the manual "MeshMaster.pdf" located in the FracMan7 directory.

ORS Files

Orientation files are ASCII space delimited column based files containing image log, FMI or other well depth dependent entries. Borehole Definition Files

Borehole Definition files contain well trajectories in space delimited column format. A single file can contain multiple well trajectories if a WellName column is specified. The well trajectory can be specified in several different ways:

XYZ – a column each for the x, y and z coordinate of each point

Collar / Direction – a column each for the X₀, Y₀ and Z₀ Collar location as well as a Trend (Azimuth) column, Plunge (Inclination) column and either segment length or Measured Depth.

Lines beginning with '#' will be ignored (comments).

Example: XYZ file

#WellName	X	Y	Z
"Well_1"	0.0000	0.0000	0.0000
"Well_1"	0.0000	0.0000	-100.0000

Example: Collar / Direction

#WellName	X0	Y0	Z0	Trend	Plunge	Length
"Well_1"	0.0000	0.0000	0.0000	0.0000	90.0000	100.0000
"Well_1"	0.0000	0.0000	0.0000	0.0000	90.0000	100.0000

Well Interval (tops) files

Well interval files are space delimited ASCII files containing well interval definitions. Each line should contain the well name, the interval set name, the interval name as well as the starting depth and (optionally) the ending depth. If the ending depth is omitted, the starting depth of the next interval will be used instead. Lines starting with '#' will be ignored (comments). Additional columns can be specified that will be interpreted and stored as interval properties. A single file can contain multiple interval sets for multiple wells.

Example:

#Well	IntervalSet	MD_Top	MD_Bottom	AProperty
Well_1	Well_1_A_Intervals	0	66.69	10
Well_1	Well_1_A_Intervals	66.69	133.3	20
Well_1	Well_1_A_Intervals	133.3	200	30
Well_1	Well_1_B_Intervals	0	100	11
Well_1	Well_1_B_Intervals	100	200	22

Well Intervals Table files

An alternative way to specify well intervals is in a file with a table format where the columns are the intervals and each row represents a well. Interval set name column is optional.

Example:

WellName IvlSet top1 top2 top3 top4

Well_1 Intervals 100 200 300 400

Well_2 Intervals 150 200 350 400

Well Collar / Deviation File

Well collars / trajectories are often stored in separate files. In this case we handle them together, loading the well collars and then handling the well trajectories (often specified in dx/dy/dz terms thus the name deviation) separately.

The collar file is assumed to have one line per well collar and include the x, y, z coordinates of the well collar, other columns can be ignored.

Example (Collar):

```
BHID   XCOLLAR YCOLLAR ZCOLLAR
WELL_1 1050.00 2000.00 260.42
WELL_2 1100.00 2000.00 280.00
WELL_3 1250.00 2000.00 320.50
```

The deviation file is assumed to have one well segment per line and can support a variety of different data formats. It needs to have the well name on each line as well as either measured depth or segment length and either the angular deviation (inclination / azimuth) or spatial deviation (dx / dy)

Example (Deviation):

```
BHID                                     AT BRGDIP
Well_1                                  0  216.25 -70.8478
Well_1                                  6.096  215.7084 -72.0876
Well_1                                  12.192  216.1823 -72.5417
....
Well_2                                  0  216.25 -70.8478
```

ASCII Well Fracture File

This is a space or comma delimited file with well log data (based on depth). Multiple wells / sets can be supported from a single file as well as multiple orientation formats (DIP / Pole etc).

ECLIPSE ASCII Grid

These are also known as “ASCII ECLIPSE keywords files”. FracMan supports both compressed and non-compressed formats for import but exports only in the uncompressed format. In an ECLIPSE grid file, the geometry may be specified in two formats:

Block-Centered: The blocks are horizontal and all eight corners are at right angles. Each block is defined by the dimensions of its three sides and the depth of the top surface

Corner-Point: The corner depths of each grid cell are specified independently of on another in the ZCORN section, and the sides of the cell are determined by projecting along the coordinate lines defined in the COORD section. Cell edges are not necessarily orthogonal.

FracMan only supports the (more general) corner point format with Cartesian (x,y,z) coordinate systems. Radial (cylindrical) coordinates are not supported. In an ECLIPSE file, sections containing particular types of information begin with a keyword, and the end of each section is denoted by a forward slash (/) character. The SPECGRID keyword is followed by three values specifying the number of cells in the x, y, and z directions (nx, ny, nz), followed by another value specifying the number of distinct grids in the model, and whether or not the coordinate system is Cartesian.

Coordinate Lines

The COORD section of an ECLIPSE grid file contains a total of $2 * (nx+1) * (ny+1)$ points. Each pair of (x,y,z) triplets define the end points of a line segment. These segments define the lines on which the corners of the grid cells must lie. If the z-values of these two points are identical, then the coordinate line is assumed to be vertical and to pass through the first point (the second point is ignored). Segment points are listed first along i, then along j.

Grid Cell Corner Depths

The ZCORN section of an ECLIPSE grid file contains a total of $8 * n_x * n_y * n_z$ floating point values, one for each corner point of each cell. A particular corner point is found by intersecting its coordinate line with the horizontal plan defined by its z value. The corner z-values are listed first along i, then j, then k. The notation 24*1000 for compressed file formats indicates that the next 24 z-values are all equal to 1000.

Coordinate Systems

The default ECLIPSE coordinate system has the z-axis pointing downward (depth), the x-axis pointing East and the y-axis pointing South. An ECLIPSE file may also specify a local coordinate system with a MAPAXES section. This section has 6 floating point values which specify the x,y coordinates of three points: p_1 , p_2 and p_3 . The origin of the local coordinate system is given by the second point p_2 . The local x-axis is given by $p_3 - p_2$, the local y-axis is given by $p_1 - p_2$ and the local z-axis is given by the vector cross product $(p_3 - p_2) \times (p_1 - p_2)$. Since only x, y coordinates are specified, the local z-axis will point either straight up or straight down. If a MAPAXES section is found, FracMan will apply the specified transformation unless the GRIDUNIT is set to "MAP".

When importing a grid from an ECLIPSE file, the x,y,z values are converted to the FracMan coordinate system in which the x-axis points East, the y-axis point North and the z-axis points upward (elevation). When exporting a grid to an ECLIPSE file, the coordinates are converted into the ECLIPSE system.

```
NOECHO
PINCH
/
MAPAXES
-100 -101
-100 -100
-99 -100
/
-- NX NY NZ NUMRES QRADIAL
SPECGRID
2 2 1 1 F /

COORD
0 -0 -100 1.49976e-030 -2.44929e-014 100
100 -0 -100 100 -2.44929e-014 100
200 -0 -100 200 -2.44929e-014 100
6.12323e-015 -100 -100 6.12323e-015 -100 100
100 -100 -100 100 -100 100
200 -100 -100 200 -100 100
1.22465e-014 -200 -100 1.22465e-014 -200 100
100 -200 -100 100 -200 100
200 -200 -100 200 -200 100
/

ZCORN
-100 -100
-100 -100
-100 -100
-100 -100

-100 -100
-100 -100
-100 -100
```

-100 -100

100 100

100 100

100 100

100 100

100 100

100 100

100 100

100 100

/

Grid Cell Properties

Properties can be loaded from the same file as the grid or later from a separate file. If loaded from a separate file, the first line contains the property name and the next (nx*ny*nz) lines contain the property value for the grid cell (order is kji). The last two lines must contain a forward slash followed by an empty line.

Stress Field File

See [Appendix E Stress Field and Fault Grid File Format](#)

3DMove Grid

Midland Valley 3DMove format grid files have a .sdb file extension and contain a list of vertices and their associated arranged in a regular grid.

Centerpoint Size Grid

This is a unstructured grid format where each line contains a single cell, defined by the xyz centerpoint of the grid cell along with the xyz size of the cell. Additional columns define additional grid properties.

Example:

XC	YC	ZC	XINC	YINC	ZINC	ZONE	GEOLOGY
-10000	-11800	240	400	400	400	1	DEFAULT
-10000	-11800	640	400	400	400	1	DEFAULT
-10000	-11800	1040	400	400	400	1	DEFAULT
-10000	-11800	1440	400	400	400	1	DEFAULT
-10000	-11800	1840	400	400	400	1	DEFAULT
-10000	-11800	2240	400	400	400	1	DEFAULT

ELFEN Grid

Produced by the ELFEN software, FracMan currently only loads grids from these files.

FLAC3D Grid

The FLAC3D grid file is a simple format with nodes (gridpoints) and elements(zones). Note the * as the comment character.

Example:

```
* FLAC3D grid produced by FLAC3D 5.01 Release 131
* Fri Dec 4 11:11:39 2015
* GRIDPOINTS
G 1  2.000000000000000e+00  0.000000000000000e+00  0.000000000000000e+00
G 2  2.10747101203521e+00  0.000000000000000e+00  0.000000000000000e+00
G 3  2.000000000000000e+00  2.000000000000000e-01  0.000000000000000e+00
G 4  1.99931464995111e+00  0.000000000000000e+00  5.23538966157462e-02
```

```
G 5 2.10747101203521e+00 2.0000000000000e-01 0.0000000000000e+00
G 6 1.99931464995111e+00 2.0000000000000e-01 5.23538966157462e-02
G 7 2.10674883435465e+00 0.0000000000000e+00 5.51671597423867e-02
G 8 2.10674883435465e+00 2.0000000000000e-01 5.51671597423867e-02
G 9 2.22031557467218e+00 0.0000000000000e+00 0.0000000000000e+00
G 10 2.22031557467218e+00 2.0000000000000e-01 0.0000000000000e+00
```

* ZONES

Z B8 1 1 2 3 4 5 6 7 8

GSLib Property File

This is a property file format that is used by the Geostatistical Software Library (www.gslib.com) and is also supported by Petrel. We have found that Petrel is capable of loading much larger properties in GSLib format than in its own Eclipse format.

Example:

Object C - N20W_1_76 - exported from FracMan

2

Area[feet]

Compressibility[1/kPa]

-9999	1e-006
-9999	1e-006
-9999	1e-006

RESCUE Grid

RESCUE (Reservoir Characterization Using Epicenter) is a joint industry project managed by Energestics. Complete information is available on their website at <http://www.energestics.org/reservoir/rescue-standards>

MODFLOW Discretized Grid

MODFLOW format grid files have a '.dis' file extension and come in a number of different internal formats. The first line contains:

Number of Layers

Number of Rows

Number of Columns

Number of stress periods

Time unit

Length Unit (0= undefined, 1=feet, 2=meters, 3=centimeters)

Comment lines are ignored

5 20 10 1 4 0

0 0 0 0 0

0	100.000(10E12.4)	0	DELTA-X or C
0	100.000(10E12.4)	0	DELTA-Y or R
0	100.000(10e12.4)	-1	TOP of Model
0	80.000(10e12.4)	-1	BOTTOM of Layer 1
0	60.000(10e12.4)	-1	BOTTOM of Layer 2
0	40.000(10e12.4)	-1	BOTTOM of Layer 3
0	20.000(10e12.4)	-1	BOTTOM of Layer 4
0	0.0000e+00(10e12.4)	-1	BOTTOM of Layer 5

10.000000 1 1.200000 SS

MODFLOW Layer Property File

The layer property file is a way to export property values (primarily permeabilities) to Modflow. It consists of a single file with a header followed by the cell values for each layer of each property in row/column order.

MODFLOW Zone Numbers File

Zone numbers is essentially a grid property (integer format). The zone numbers can have a range of 1-999999. The file is a simple file with no header lines containing four columns:

Row, Column, Layer, ZoneNumber

FeFlow Grid

FracMan is capable of reading grid geometry and properties from a FeFLOW .fem model file. This file is an ASCII formatted file that contains the geometry as well as properties of the grid object.

Z-Map Plus

FracMan can read surfaces from Z-Map Plus format "Grid" files as well as tracemaps and point data from Z-Map point and line format files

```
!      Comment lines
!
@Grid HEADER,GRID,5
 15,0.1000000E+31, , 7, 1
 700,647,2576850.0000,2609150.0000,4451850.0000,4486800.0000
 0.0000, 50.0000, 50.0000
@
 0.1000000E+31  0.1000000E+31  0.1000000E+31  0.1000000E+31  0.1000000E+31
 -4146.5732    -4152.6299    -4155.7983    -4159.8096    -4164.8843
 -4169.9595    -4176.4189    -4184.2114    -4192.0039    -4197.0254
...continues
 0.1000000E+31  0.1000000E+31  0.1000000E+31  0.1000000E+31  0.1000000E+31
```

DXF Surface

The standard configuration supports ASCII dxf files containing 3DFace records. This format is supported by older versions of AutoCAD and can generally be obtained from a new AutoCAD version by selecting to save as an earlier version.

RMS Surface

The IRAP RMS format handles two distinct types of files. The first, so called "GEOMATIC" format as well as the more common "Classic" IRAP format.

The "GEOMATIC" format consists of a header (delineated by "Begin" and "End") followed by parameter descriptions, followed by 3d points and then triangle vertex indices.

Example:

```
Begin GEOMATIC file header
...
End GEOMATIC file header
...
End parameter
-2.2100933e+03 3.5932527e+03 -8.1879033e+03
-2.1824495e+03 3.5729907e+03 -7.9738809e+03
```

```

-2.1549280e+03 3.5528184e+03 -7.7629033e+03
...
-1.3014900e+03 4.0431472e+03 -6.2816973e+03
0 1 14
1 15 14
1 2 15
...
251 265 264

```

The "Classic" format has a header followed by a block of all the X values, then all the Y values and all the Z values. The x, y, z values form a grid defined by the 3rd and 4th parameters in the header

Example:

```

-996 359 100.000000 100.000000
583239.8750 642739.8750 3638987.5000 3674787.5000
596 0.0000 583239.8750 3638987.5000
0 0 0 0 0 0 0
9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000
9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000
9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000
9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000
9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000 9999900.0000

```

CPS-3 Surface

FracMan can read CPS-3 format grid (mesh) data (surface objects in FracMan). These files are identified by the "FSASCI" on the first line of the file.

The header includes the following keywords:

- **FSASCI** Used to identify the CPS-3 mesh format
- **FSATTR** Not used
- **FSLIMI** The first 4 numbers are X-Minimum, X-Maximum, Y-Minimum and Y-Maximum respectively. The last two numbers are not used.
- **FSNROW** The first number identifies the number of rows, the second number the number of columns.
- **->** Indicates the start of the z values

Example:

```

FSASCI 0 1 "Computed" 0 1000000015047466219876688855040.000000 0
FSATTR 0 0
FSLIMI 1272000.0 1291000.0 230000.00 263000.00 448.54349 3685.0830
FSNROW 166 96
FSXINC 200.00000 200.00000
->Default
0.100000E+31 0.9883727E+03 0.1047147E+04 0.1112168E+04 0.1177649E+04
0.124080E+04 0.1303553E+04 0.1359258E+04 0.1406198E+04 0.1449266E+04
0.149218E+04 0.1533531E+04 0.1572911E+04 0.1610486E+04 0.1647059E+04
0.168304E+04 0.1718736E+04 0.1754464E+04 0.1790438E+04 0.1826841E+04
0.186319E+04 0.1899145E+04 0.1934789E+04 0.1970341E+04 0.2006256E+04
0.204292E+04 0.2080046E+04 0.2117406E+04 0.2154524E+04 0.2190654E+04
0.222484E+04 0.2256072E+04 0.2285586E+04 0.2314498E+04 0.2343419E+04
0.237284E+04 0.2402762E+04 0.2432462E+04 0.2461459E+04 0.2489767E+04
0.251781E+04 0.2546487E+04 0.2575592E+04 0.2604711E+04 0.2633801E+04
0.266300E+04 0.2692280E+04 0.2721548E+04 0.2750791E+04 0.2780021E+04

```

0.280925E+04 0.2838482E+04 0.2867689E+04 0.2896782E+04 0.2925824E+04
0.295488E+04 0.2983842E+04 0.3012370E+04 0.3039971E+04 0.3066538E+04

Earth Vision Surface

Example:

```
# Grid_size: 4 x 3
#
0.0 0.0 1.0    1  1
1.0   0.0 1.0    2  1
2.0   0.0 1.0    3  1
3.0   0.0 1.0    4  1
0.0 1.0 1.0    1  2
1.0   1.0 1.0    2  2
2.0   1.0 1.0    3  2
3.0   1.0 1.0    4  2
0.0 2.0 1.0    1  3
1.0   2.0 1.0    2  3
2.0   2.0 1.0    3  3
3.0   2.0 1.0    4  3
```

Petrel Surface

This is a surface format exported by Petrel Software.

Charisma Fault Stick

A surface format generated by the Schlumberger Charisma Seismic Interpretation software. The file consists of whitespace delimited columns containing ordered xyz points defining the fault pillars along with the indices of the fault pillar. FracMan supports importing surfaces defined in this format.

Example:

```
INLINE- 610 118 75.96975    60.39011    55.28551    Fault_01    1
INLINE- 610 119 54.09193    60.16113    116.55864    Fault_01    1
INLINE- 610 119 49.42254    60.11226    197.78397    Fault_01    1
INLINE- 610 119 57.56576    60.19749    263.37055    Fault_01    1
INLINE- 610 120 19.87941    59.80305    317.3503     Fault_01    1
INLINE- 605 118 63.58165    210.26996    48.28071     Fault_01    2
INLINE- 605 119 59.26404    210.22477    131.96597    Fault_01    2
INLINE- 605 119 40.88854    210.03245    224.13177    Fault_01    2
INLINE- 605 120 6.33591     209.67081    306.6538     Fault_01    2
INLINE- 600 120 8.32834     359.70118    59.23688     Fault_01    3
INLINE- 600 120 5.20732     359.66851    153.91104    Fault_01    3
INLINE- 600 121 -23.11411     359.37209    291.76689    Fault_01    3
INLINE- 595 121 -29.57262     509.31401    94.82768     Fault_01    4
INLINE- 595 121 -13.54983     509.4817     223.28949    Fault_01    4
INLINE- 615 119 61.83485    -89.76734    85.78833     Fault_01    5
INLINE- 615 117 104.52054    -89.32058    189.52743    Fault_01    5
INLINE- 615 117 100.92914    -89.35817    278.18578    Fault_01    5
INLINE- 610 176 -1675.34555    42.06037     43.11076     Fault_02    1
INLINE- 610 175 -1640.21736    42.42803     114.36884    Fault_02    1
INLINE- 610 172 -1536.91666    43.5092      222.40134    Fault_02    1
INLINE- 610 168 -1410.22867    44.83515     384.82094    Fault_02    1
INLINE- 610 163 -1284.36392    46.15249     502.4302     Fault_02    1
INLINE- 605 175 -1639.90679    192.44079    58.16685     Fault_02    2
```

INLINE- 605 175 -1625.71196	192.58935	106.06411	Fault_02	2
INLINE- 605 174 -1611.69424	192.73607	169.76239	Fault_02	2
INLINE- 605 171 -1523.43703	193.65979	248.98921	Fault_02	2
INLINE- 605 165 -1342.70719	195.55136	420.19784	Fault_02	2
INLINE- 600 173 -1588.59926	342.9873	57.81449	Fault_02	3
INLINE- 600 173 -1568.70804	343.19548	116.38309	Fault_02	3
INLINE- 600 169 -1454.37041	344.39217	281.21582	Fault_02	3
INLINE- 600 166 -1370.42943	345.27072	383.51935	Fault_02	3
INLINE- 595 171 -1523.07878	493.68256	75.91235	Fault_02	4
INLINE- 595 170 -1479.15426	494.14229	157.77559	Fault_02	4
INLINE- 595 168 -1425.9093	494.69957	262.22929	Fault_02	4
INLINE- 595 164 -1311.41028	495.89794	398.84775	Fault_02	4
INLINE- 590 169 -1444.37614	644.5158	98.73181	Fault_02	5
INLINE- 590 166 -1375.99988	645.23144	253.65479	Fault_02	5
INLINE- 590 164 -1305.04532	645.97407	357.62368	Fault_02	5

Polygon File

The PLY Polygon file, also known as the Stanford Triangle format is a simple to use ASCII format containing vertices, triangular faces and edges. Complete documentation can be found online.

Example:

```
ply
format ascii 1.0
comment Comment Lines are ignored
element vertex 8
property float32 x
property float32 y
property float32 z
element face 6
property list uint8 int32 vertex_indices
end_header
-1 -1 -1
1 -1 -1
1 1 -1
-1 1 -1
-1 -1 1
1 -1 1
1 1 1
-1 1 1
4 0 1 2 3
4 5 4 7 6
4 6 2 1 5
4 3 7 4 0
4 7 3 2 6
4 5 1 0 4
```

Wavefront OBJ

Object files define the geometry and other properties for objects in

Wavefront's Advanced Visualizer. The format is well documented online.

GiD ASCII mesh

A GiD format mesh file contains a list of elements (vertices) followed by a list of elements that can be lines, triangles, quads, tetrahedra, hexahedra, triangular prisms, pyramids, points, spheres and circles. FracMan will automatically create the corresponding FracMan objects based on the contents of the file. Note that FracMan does not currently support pyramid/sphere/circle type point data.

GiD files starts with a line that defines the contents of the file, followed by a list of vertices and then a list of elements and finally an optional list of “materials” which are currently ignored by FracMan.

```
mesh dimension 3 elemtype tetrahedra nnode 4
coordinates
1 0 0 0
2 3 0 0
3 6 0 0
4 3 3 0
5 3 1.5 4
6 3 1.5 -4
7 1.5 0 2
end coordinates
elements
1 1 2 4 5 1
2 2 3 4 5 1
3 1 4 2 6 1
4 2 4 3 6 1
5 1 2 5 7 1
end elements
```

FracMan also supports a variation of the GiD format as created by the ConnectFlow software.

```
NNODES 7 NELL 5 NMAT 1
NODES
1 0 0 0
2 3 0 0
3 6 0 0
4 3 3 0
5 3 1.5 4
6 3 1.5 -4
7 1.5 0 2
END NODES
ELEMENTS
1 1 2 4 5 1
2 2 3 4 5 1
3 1 4 2 6 1
4 2 4 3 6 1
5 1 2 5 7 1
END ELEMENTS
MATERIALS
```

1 Rock

ESRI Shapefile

An ESRI Shapefile consists of a main file, an index file and a dBASE table. A complete description of the ESRI shapefile can be found at <http://www.esri.com/>

FracMan supports reading the following shape types from ESRI shapefiles:

- Point, PointZ, PointM, Multipoint, MultipointZ and MultipointM – these are loaded into Point Data objects.
- PolyLine, PolyLineZ and PolyLineM – are loaded into tracemap objects

Note: Connected polyline parts are joined into a single trace, dis-connected polyline parts are loaded into separate traces.

- Polygon, PolygonZ and PolygonM are loaded into a single surface object. If you are loading surfaces from ESRI files, make sure to include only a single surface per ESRI file in order to get separate objects in FracMan. Additionally complex polygon outlines should be triangulated in order to simplify their geometry before importing into FracMan.

FracMan supports writing the following types into ESRI shapefiles:

- Point Objects, Point Data Sets
- Fracture Sets, Surfaces and Region Objects
- Wells and Tracemaps

Maptek Vulcan

Surpac Digital Terrain / String Files

Surpac data files are broken into two pieces. The “String” (.str) file contains the xyz coordinates of points along with additional properties. The “Digital Terrain Model” (.dtm) files define elements (triangles) that reference the vertices in the string file.

Example .dtm file:

```
pointsfilename.str,5057543000;algorithm=standard;fields=x,y
0, 0.000, 0.000, 0.000, END
OBJECT, 1,
TRISOLATION, 1, neighbours=yes,validated=true,closed=yes,direction=solid
1, 359, 4, 3, 238, 2, 3,
2, 3, 4, 1, 1, 81, 3,
...
440, 298, 314, 302, 397, 439, 385,
END
```

Example .str file:

```
pointsfilename,01-Jan-01,Description of file contents,
0, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000
1, 6793783.404, 1527263.851, -312.428,
1, 6792977.970, 1527030.044, -592.570,
...
0, 0.000, 0.000, 0.000,
```

0, 0.000, 0.000, 0.000, END

FracMan Tracedata Files

These are space delimited column data containing trace data information. FracMan can support a number of variations of columns:

- X1, Y1, Z1, X2, Y2, Z2, [TraceIndex]: Each line specifies a start and end point of a trace segment. Successive lines with either a matching (optional) TraceID column or matching start/end points are linked into traces.
- X1, Y1, Z1,[TraceIndex]: Each line specifies a single point. Successive lines are linked together into traces. An optional TraceID column will split into separate traces.
- If z coordinates are omitted and a traceplane is specified (with the first line of the file containing the name of a [.SAB file](#) containing a traceplane). The XY points are computed in the traceplane relative to the plane origin.
- Additional columns can be handled and will be stored as segment properties
- Lines beginning with '#' are ignored (comments).

A sample F2D file containing xyz start and end points:

#	x1	y1	z1	x2	y2	z2
236.95	339.785	121.21	486.46	-197.935	121.21	
486.46	-197.935	121.21	849.66	-231.785	121.21	
2523.03	-206.805	121.21	2486.23	-240.665	121.21	
3675.291	753.265	121.21	3638.491	719.415	121.21	
5201.45	167.955	121.21	5164.651	34.095	121.21	
4762.991	955.565	121.21	4726.21	921.715	121.21	

An F2D file containing xy points with a plane from a .sab file

```
XYPlane.sab
# x1, y1, x2, y2
486.46 -197.935 236.95 339.785
486.46 -197.935 849.66 -231.785
2523.03 -206.805 2486.23 -240.665
3675.291 753.265 3638.49 1719.415
5201.45 167.955 5164.65 134.095
4762.991 955.565 4726.2 1921.715
```

CPS-3 Lines

IRAP Classic Lines

MAFIC Particle Track

SEGY Format Data

We support several interpretations of the SEG Y rev 1 Data Exchange format as presented by the SEG Technical Standards Committee, May 2002 (http://www.seg.org/documents/10161/77915/seg_y_rev1.pdf). This standard is used to store seismic reflection data. If you have difficulty loading your SEG Y formatted data, contact FracMan support and we will assist you. Different data providers are continually adjusting their format.

Appendix B Troubleshooting USB Security Key Issues

You can use the driver installer, install.exe, to remove all drivers and dongle-related software.

To do so, do the following, making sure that you are logged in as an Administrator on the machine:

1. Remove all dongles from the system.
2. Run install.exe. When the opening screen comes up, make sure that the "Uninstall" checkbox is selected. Click OK.
3. When the uninstall displays a message saying that it has finished, all files have been removed.
4. To reinstall the drivers after an uninstall, do the following, making sure that you are logged in as an Administrator on the machine:
5. Reboot the system.
6. Re-run install.exe. Select the USB checkbox.
7. Make sure the install has displayed a message saying that it is finished before you attach the dongle. When you do so, the Windows New Hardware wizard should start and ask to install the drivers for a USB Security Protection device. Accept all the defaults and let the wizard finish.

Appendix C FAB Fracture File Format

FAB files are ASCII formatted text files containing stochastic (planar polygon) fractures, tessellated (triangulated surface) fractures and matrix blocks (regular grids).

Line #	Parameter/ Keyword	Valid Values/ Keyword	Example
1	BEGIN Format	Keyword, the beginning of format specification	BEGIN Format
2	Format =	Specification of whether the file is in Ascii or (Ascii) or Babylonian (base 60) notation (Babylonian)	Ascii
3	Scale =	scale of the box containing all fractures in meters (float)	100.
4	XAxis =	coordinate system (right handed)	East or South
5	NumFractures =	number of fractures in the file (integer)	200
6	NumTessFractures =	number of non-planar (tessellated) fractures (integer)	5
7	NumRockBlocks =	number of cells defining continuum elements (integer)	2000

8	NumNodes =	number of nodes fractures (integer), tessfracture, and rock block	800
9	NumProperties =	number of properties to be assigned to fractures and blocks (integer between 3 and 20)	3
10	NumNodeProps =	number of properties to be assigned to nodes (integer between 0 and 20)	0
11	NumSets =	number of fracture sets (integer)	5
12	END Format	Keyword, the end of format data block	END Format
		Blank line	
13	BEGIN Properties	Keyword, the beginning of fracture property specification	BEGIN Properties
14	Prop1 = (Real*4)	Property type (keyword), property name (text of up to 20 characters enclosed in quotes). Repeat this record NumProperties+NumNodeProps times to specify all property names and types. Valid types are (Real*4), (Integer*4), and (Text*20)	Prop1 = (Real*4) "Transmissivity "
15	END Properties	Keyword	END Properties
		Blank line	
16	BEGIN Sets	Keyword, the beginning of fracture set specification	BEGIN Sets
17	Set1 =	The set name enclosed by a pair of double-quotes, # of characters up to 20. Repeat this record NumSets times.	Set1 = "Conjugate Shear A"
18	END Sets	Keyword	END Sets
		Blank line	

Line #	Parameter/ Keyword	Valid Values/ Keyword	Example
19	BEGIN Fracture	Keyword for beginning of fracture specification	BEGIN Fracture
20	Fracture #, # of vertices, Set #, Properties	Fracture number (integer), Number of vertices (integer), set number (integer), Properties (NumProperties float)	1 4 1 2.300E-06 2.700E-05 1.100E-03
21	x,y,z coordinates for the vertex, node properties	Vertex index (integer), x,y, z coordinates (float), node properties if any (float). Repeat for each vertex of the fracture	1 -1.000000E+02 9.999999E+01 - 1.000000E+01
22	x,y,z components of fracture unit normal vector	0 (integer), then, three direction cosines for the unit normal (three float)	0 2.244704E-01 6.784559E-01 - 6.995074E-01
23		Repeat 20,21,22 for each fracture	
24	END Fracture	Keyword	END Fracture
		Blank Line	
25	BEGIN TessFracture	Keyword for beginning of non-planar (tessellated) fractures	BEGIN TessFracture
26	# FractureNum NumNodes NumFaces SetNum Prop1,Prop2, ..., PropN	Fracture Number (integer), number of nodes (integer), number of faces (integer), Set number (integer), properties (NumProp float numbers)	1 4 2 3 1.000E-06 1.000E-07 1.000E-04
27	index, x,y,z coordinates for the first node, node properties	Index (integer), x,y, z coordinates (three float), node properties if any(float). Repeat NumNodes times.	-1.000000E+02 9.999999E+01 -1.000000E+01
28	FaceNum NumNodes Node1 Node2.	Index (integer), number of nodes defining the face (integer, usually 3), indices for the nodes defining the cell (integers)	1 3 4 3 2
29		Repeat 26, 27, 28 for all TessFractures	
30	END TessFrac	Keyword	END TessFrac
		Blank Line	

Line #	Parameter/ Keyword	Valid Values/ Keyword	Example
31	BEGIN RockBlock	Keyword for beginning of rock blocks	BEGIN RockBlock
32	BlockNum NumNodes NumFaces, SetNum, Prop1,Prop2, ..., PropN	Index (integer), number of nodes defining the block (integer), number of faces defining the block (integer), set number (integer), block properties (NumProperties float numbers)	1 4 2 3 1.000E-06 1.000E-07 1.000E-04
33	index, x,y,z coordinates for the nodes, node properties	Index (integer), x,y, z coordinates (three float), node properties (float, if any). Repeat NumNodes times for this block	1 -1.000000E+02 9.999999E+01 - 1.000000E+01
34	BlockNum, NumNodes Node1 Node2.	Block Number (integer), number of nodes on this face (integer), node numbers (integers). Repeat for each face of this block.	1 6 4 3 2 7 8 9
35		Repeat 32, 33, 34 for all RockBlocks	
36	END RockBlock	Keyword	END RockBlock

Example Data File:

BEGIN FORMAT

```
Format = Ascii
Scale = 100.
NumFractures = 7
NumNodes = 28
NumProperties = 3
```

END format

BEGIN Properties

```
Prop1 = (Real*4) "Transmissivity "
Prop2 = (Real*4) "Storativity "
Prop3 = (Real*4) "Frac Thickness "
```

END Properties

BEGIN Fracture

```
1 4 1 1.000E-06 1.000E-06 1.000E-06
1 -1.000000E+02 9.999999E+01 -1.000000E+01
2 9.999998E+01 1.000000E+02 -9.999999E+00
3 1.000000E+02 -9.999999E+01 -9.999999E+00
4 -9.999999E+01 -1.000000E+02 -1.000000E+01
0 8.742278E-08 7.642743E-15 -1.000000E+00
2 4 1 1.000E-06 1.000E-06 1.000E-06
```

```

1 -1.000000E+02  9.999999E+01  9.999999E+00
2  9.999998E+01  1.000000E+02  1.000000E+01
3  1.000000E+02 -9.999999E+01  1.000000E+01
4 -9.999999E+01 -1.000000E+02  9.999999E+00
0  8.742278E-08  7.642743E-15 -1.000000E+00
3  4  2  1.000E-08  3.000E-09  1.000E-07
1 -8.532802E+00  7.138195E+01 -4.576039E-01
2 -8.532801E+00  5.000718E+00 -5.917314E+01
3 -8.532790E+00 -5.371481E+01  7.208108E+00
4 -8.532791E+00  1.266644E+01  6.592362E+01
0 -1.000000E+00 -8.742278E-08  7.549790E-08
4  4  2  1.000E-08  4.000E-09  1.000E-07
1  3.489493E+00  4.877172E+01 -4.762970E+01
2  3.489500E+00 -3.982861E+01 -4.962060E+01
3  3.489507E+00 -4.181949E+01  3.897973E+01
4  3.489500E+00  4.678084E+01  4.097061E+01
0 -1.000000E+00 -8.742278E-08  7.549790E-08
5  4  2  1.000E-08  5.000E-09  1.000E-07
1  7.627331E+00  5.515668E+01 -1.291166E+01
2  7.627334E+00 -1.791647E+01 -6.305441E+01
3  7.627343E+00 -6.805923E+01  1.001874E+01
4  7.627341E+00  5.013939E+00  6.016149E+01
0 -1.000000E+00 -8.742278E-08  7.549790E-08
6  4  2  1.000E-08  6.000E-09  1.000E-07
1  5.342791E+00  1.790385E+01 -6.814629E+01
2  5.342801E+00 -5.795161E+01 -2.232127E+01
3  5.342803E+00 -1.212658E+01  5.353417E+01
4  5.342793E+00  6.372886E+01  7.709132E+00
0 -1.000000E+00 -8.742278E-08  7.549790E-08
7  4  2  1.000E-08  7.000E-09  1.000E-07
1  3.506775E+00  3.207570E+01 -6.095095E+01
2  3.506785E+00 -5.082361E+01 -2.961909E+01
3  3.506789E+00 -1.949173E+01  5.328023E+01
4  3.506779E+00  6.340758E+01  2.194833E+01
0 -1.000000E+00 -8.742278E-08  7.549790E-08

```

END Fracture

BEGIN TESSFRACTURE

```

1  4  2  3  1.000E-06  1.000E-06  1.000E-06 # FractureNum  NumNodes  NumFaces
SetNum  Prop1  Prop2...
1  2.820940000e+002  1.274650000e+003  -1.372050000e+003 #  VertNum  X  Y  Z
2  9.132780000e+002  8.731020000e+002  -1.343660000e+003
3  7.110740000e+002  1.231640000e+003  -1.692540000e+003
4  5.312290000e+002  1.317930000e+003  -1.684960000e+003
1  3  4  3  2 #  FaceNum  NumNodes  Node1  Node2...
2  3  4  2  1
2  4  1  4  2.000E-06  1.500E-06  1.010E-05
1  2.999999E+01  1.100000E+01  2.947445E+00
2  2.000000E+01  -1.100000E+01  -2.947445E+00
3  -3.000000E+01  -1.100000E+01  -2.947445E+00
4  -3.000000E+01  1.100000E+01  2.947445E+00
1  4  4  3  2  1

```

END TESSFRACTURE

BEGIN ROCKBLOCK

```

1  4.041470000e+002  7.807370000e+002  -1.477100000e+003

```

2	3.713240000e+002	8.219120000e+002	-1.480100000e+003
3	4.346420000e+002	8.495630000e+002	-1.479020000e+003
4	3.680230000e+002	9.038510000e+002	-1.482390000e+003
5	4.422810000e+002	9.498270000e+002	-1.481290000e+003
6	3.868270000e+002	1.009080000e+003	-1.483400000e+003
7	4.533280000e+002	1.050740000e+003	-1.482510000e+003
8	5.189910000e+002	1.107690000e+003	-1.482170000e+003
9	4.544300000e+002	1.149910000e+003	-1.483270000e+003
10	4.527270000e+002	1.297540000e+003	-1.483910000e+003
11	5.142860000e+002	1.230460000e+003	-1.482670000e+003
12	5.596560000e+002	1.322350000e+003	-1.481860000e+003
13	5.024640000e+002	1.326520000e+003	-1.482850000e+003
14	5.260910000e+002	1.392950000e+003	-1.482090000e+003

```

1  8  6  5  1.000000000e-006  1.000000000e-007  1.000000000e-009 # BlockNum
NumNodes NumFaces SetNum Prop1 Prop2...
  1  4  1  2  3  4 # FaceNum NumNodes Node1 Node2...
  2  4  5  6  7  8
  3  4  1  2  6  5
  4  4  2  3  7  6
  5  4  3  4  8  7
  6  4  4  1  5  8
2  6  5  6  1.600000000e-006  5.340000000e-007  1.280000000e-009 # BlockNum
NumNodes NumFaces SetNum Prop1 Prop2...
  1  3  9  10  11
  2  3  12  13  14
  3  4  9  10  13  12
  4  4  10  11  14  13
  5  4  11  9  14  12
END ROCKBLOCK

```

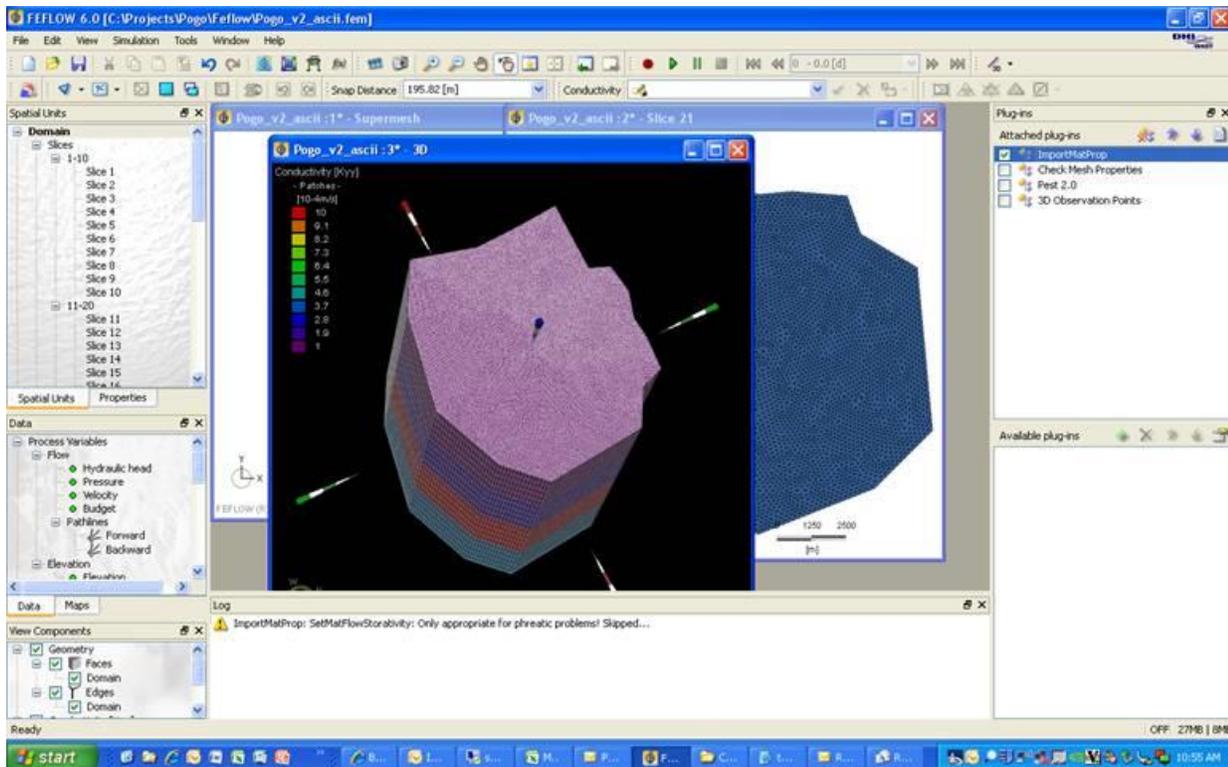
Appendix D FEFLOW Interface

FEM import from FEFLOW

ASCII model definitions files (*.fem) can now be imported directly from FEFLOW select “Grid” from the “File” -> “Import” menu. Model geometry as well as matrix properties are accessible directly from the FracMan model.

Property Export with FEFLOW plugin

Grid properties (including ODA results) can be exported in a single file that is directly importable by the FEFLOW plugin “ImportMatProp”. This plugin is available from WASY (<http://www.feflow.info/>). To install the plugin, simply copy the plugin files into the FEFLOW directory and install it via the “Plug-ins” panel. Next, attach the plugin to the current model via the plug-ins panel and hit the “Edit” button in the upper part of the panel. In the file selection box, select the file to import.



The model must be set to “unconfined” for the import to work.

Appendix E Stress Field and Fault Grid File Format

Basics

The FracWorks95 grid file uses an object oriented file structure. It consists of three parts: the *Header*, the *local coordinate system* object and the *region* object. Each object is defined in a definition block, which has the general format

```
BEGIN object_type
object constants
object definition
END
```

The file may also contain *comments*. Comments may be inserted anywhere in the input file using the comment character, '#’.

Some lines in a FracWorks95 file may require a large number of columns to specify the required data. Computer or terminal screens with fixed 80-character widths may be unable to display long input lines without “wrapping” part of the text back to the left-hand margin. Because files with wrapped text are often difficult to read, the grid file format provides a line continuation character, ‘\’. Use of this character at the end of a line instructs FracMan to merge the following line with the present one.

Because an extra space (‘ ’) is inserted between lines as they are merged, individual words cannot be continued across lines.

Example: The lines

```
This is\
really just\
one line.
```

are interpreted as:

This is really just one line.

Note that the line continuation character, '\', has special meaning only when it appears as last character on the line. Thus, FracMan will read and process these lines separately:

The \ character is not in the right position for these lines to be merged.

Intensity Parameters

The grid file can contain up to two user-defined scalar fracture intensity parameters. The number of user-defined intensity parameters and their names must be specified at the start of the file, before any BEGIN-END definition blocks. Two assignment statements are required:

```
num_intensity_params = num
```

```
intensity_param_names = name1 ... namenum
```

where num is the number of intensity parameters, and name₁ through name_{num} are their names. If num is zero, then the intensity_param_names assignment statement can be omitted altogether. An example is provided below:

```
! FILE: C:\temp\BPF_B3_small.ste
! DATE: 8/21/2000

num_intensity_params = 1
intensity_param_names = YoungMod
```

Local Coordinate Systems

The grid file format allows you to define *local coordinate systems* of arbitrary origin and orientation relative to the FracMan global coordinates. Local coordinate systems make it much easier to define objects with complex geometry and to import data from other programs. A local coordinate system is defined by specifying the location of its origin and orientation of its axes relative to a *parent* coordinate system. Any previously defined coordinate system, including the FracMan global coordinates, may serve as the parent. There is no limit on the number of local coordinate systems that can be defined. The FracMan global coordinate system is named 'gbl'.

The orientation of a local coordinate system's axis is determined as follows: start with a set of axes identical in orientation to the parent coordinate system. Apply one clockwise rotation of the requested magnitude about each local coordinate system axis in the order requested. For example, a rotation order of zyx produces a rotation first about the Z-axis, then about the Y-axis, and finally about the X-axis. One or more coordinate systems can be defined in a single definition block. The example below defines a new coordinate systems new_gbl in which X points North, Y points West, and Z points up.

```

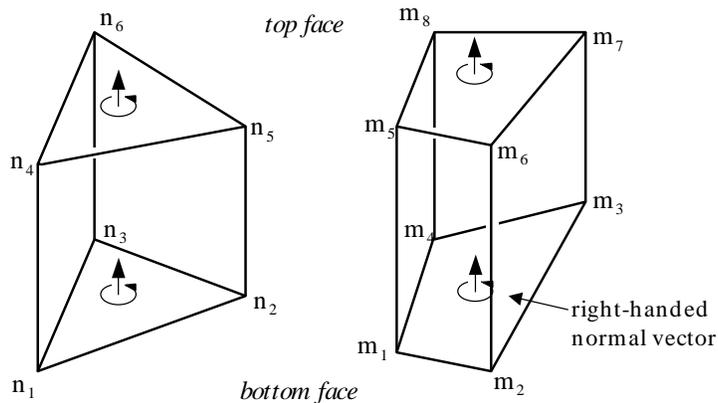
BEGIN
# name parent  xo  yo  zo  rot_x  rot_y  rot_z  rot_order
#-----
new_glb  glb  0.0  0.0  0.0  0.0  0.0  -90.0  xyz
END

```

Grid Structure

The stress field or fault grids are composed of prismatic elements with three- or four-sided cross-sections. A triangular cross-section prismatic element contains three side faces composed of four-sided polygons, and top and bottom faces composed of three-sided polygons. A quadrilateral cross-section prismatic element is composed of six four-sided polygons. The top and bottom faces of the prismatic element are not required to be parallel.

Prismatic elements are defined by specifying six or eight nodes in the order they should be connected to create the bottom and top faces. For triangular cross-section prismatic elements, nodes 1 through 3 form the bottom face while nodes 4 through 6 form the top face



Similarly, for quadrilateral cross-section prismatic elements, nodes 1 through 4 form the bottom face while nodes 5 through 8 form the top face. The nodes are listed in an order such that the right-handed normal vector to the bottom face points into the element while the right-handed normal vector of the top face points outside of the element. Hence, the prismatic elements shown above would be defined in a grid input file using the format:

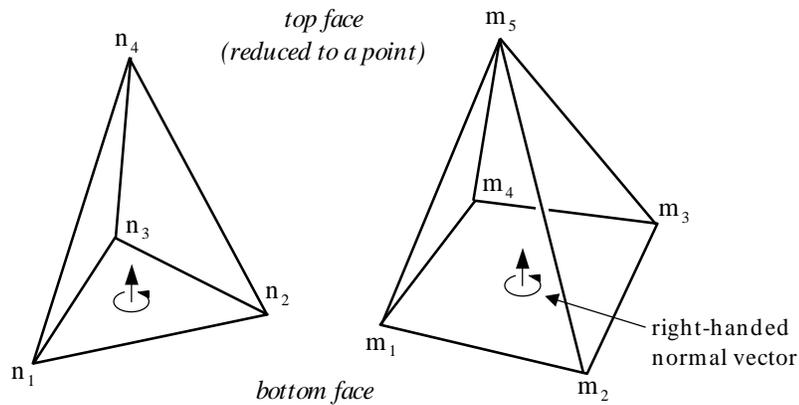
other element parameters.. n1 n2 n3 n4 n5 n6 # 3-sided

other element parameters.. m1 m2 m3 m4 m5 m6 m7 m8 # 4-sided

Tetrahedral or pyramid-shaped elements can be defined by using nodes repeatedly in an element definition. For example the elements shown below would be defined using the format:

other element parameters.. n1 n2 n3 n4 n4 n4 # tetrahedron

other element parameters.. m1 m2 m3 m4 m5 m5 m5 # pyramid



Stress Region Definition

The definition block begins with a BEGIN statement of the form:

```
BEGIN stress_region
```

This statement is immediately followed by the definition of three constants. Each constant is defined in a three-word assignment statement in the format

```
name = value
```

where name is the constant name and value is the assigned value. The three constants may be defined in any order. Appropriate values are listed below:

name: The name to be used for this stress region. The name can be of any length. Multiple word names can be assigned by enclosing the name in double-quotes.

df_csyz: The default coordinate system in which nodal coordinates and stress orientation vectors will be specified (see below). Valid choices are the default global coordinates (gbl) or the name of any local coordinate system previously defined in the input file.

dir_vec_fmt: The mode used to define principal stress direction vectors. Valid choices are trend_plunge, alpha_beta, and x_y

Once the stress region constants have been defined, the geometry of the stress region is specified in terms of nodes and elements. Nodes are defined first in a section beginning with the keyword nodes on a line by itself. FracMan assumes that each subsequent line defines a node, until it encounters the keyword elements on a line by itself. Five parameters are required to define a node, as listed below.

Column	Suggested Column Title(s)	Description	Units
1	name	Node name.	N/A
2	csys	The coordinate system in which the node location is specified. "DF" = use the default coordinate system named by the constant df_csys.	N/A
3-5	x,y,z	Location of the node in the coordinate system specified in column 2.	distance

The element keyword on a line by itself terminates the nodal definition section and begins the element definition section. A stress element is defined by specifying six or eight nodes which are joined together to form a prismatic element with a triangular or quadrilateral cross-section. For each newly-created stress element, values must be provided for the magnitudes and directions of principal stress and the user-defined intensity parameters named by the constant intensity_param_names. Depending on the number of prism sides and the number of user-defined stress intensity parameters, a stress element definition statement requires fourteen or more parameters, as shown below:

Column	Suggested Column Title(s)	Description	Units
1	csys	The coordinate system in which the principal stress directions are specified. "DF" = use the default coordinate system named by the constant df_csys	N/A
2-4	sig1,sig2,sig3	Principal stress magnitudes.	stress
5-6	sig1_dir	The direction of the most compressive principal stress component, θ_1 .	degrees or distance
7-8	sig2_dir	The direction of the least compressive principal stress component, θ_3 .	degrees or distance
(>8)	ip1...ipn	One value for each user-defined fracture intensity parameter	(varies)
(>8)	n1-n6 or n1-n8	The six or eight nodes which are to be joined together form the prismatic element	N/A

The END keyword on a line by itself terminates the definition of elements. It also marks the end of the stress region definition block.

Appendix F File Formats for Well Test Simulation

Appendix G File Formats for Flow Simulation

This chapter covers the file formats of the various input files used for flow simulation.

Parameter Control File (Welltest Simulation)

The parameter control file is the main input file defining the global parameters of the simulation and the name and location of other input files. An example of a parameter control file with comments is shown below.

```
Version 1.20
wtest1           # Simulation ID
WELL_TEST       # Simulation type
well.cfg        # Well configuration file
1               # No. of fracture files
FREDFractures.fab # fracture file name
0               # No. of fault files
none            # Matrix type
eclipse.dat     # Reservoir petrophysical property data file
1000            # Initial pressure
998             # Fluid density
1.5            # Fluid viscosity
1e-006         # Fluid compressibility
3.4            # Block radius
2e-006         # Matrix compressibility
20             # Max. element size
```

Parameters, which are grayed out are currently not used in FracMan.

Parameter Control File (Permeability Calculation)

The parameter control file is the main input file defining the global parameters of the simulation and the name and location of other input files. An example of a parameter control file with comments is shown below.

```

Version 1.20
perm1          # Simulation ID
BLOCK_PERMEABILITY # Simulation type
block_k.cfg    # Block_k configuration file
1             # No. of fracture files
demo.itm_fra   # fracture file name
0            # No. of fault files
none          # Matrix type
Test.grd      # Eclipse grid file
1000         # Initial pressure
998          # Fluid density
1.5          # Fluid viscosity
1e-006       # Fluid compressibility
3.4          # Block radius
2e-006       # Matrix compressibility
20           # Max. element size

```

Parameters, which are grayed out are currently not used in FracMan.

Well Control File

The well control file is used to define well interval locations and rate schedules. An example of a Well Control File with comments is shown below.

```

1          # No. of Wells
Well      # Well name
1         # No. of perforation intervals
5 0.1     # No. of points, perforation radius
15 50 0
15 50 -25
16 50 -26
17 50 -27
19 50 -27
1         # No. of rates, then duration, rate, formation volume factor
10 0.01 1

no_restart

3         # No. of pressure snapshots
    1
    2
    4

100      # No. of timesteps

```

Permeability Calculation Control File

The permeability calculation control file is used to define the cell range and directions. An example of a Permeability Calculation Control File with comments is shown below.

```

0          # of fault files
0          # fault file name
CALC_STAT # Keyword for calculating block statistics
NOLGR     # Local Grid Refinement ?
RANGE     # Keyword for grid cell range definition
1,4       # I1, I2
1,10      # J1,J2
1,1       # K1,K2
-1        # Directions (1=X, 2=Y, 3=Z, -1=all)|
NO_FLOW   # Keyword to specify boundary condition type

```

Parameters, which are **grayed out** are currently not used in FracMan.

Units Control File

The units control file is used to define units to be used in the simulation. It can also be used to specify the ISTORE runtime parameter used by the flow solver.

An example of a units control file with comments is shown below.

```

# Units configuration - coordinates always 'meter'

#23456789012345678901234567890

time          hours
density       kg/m^3
FlowRate      m^3/d
pressure      psi
compressibility 1/psi
Viscosity     cP
permeability  mD
FractureAperture m
porosity      fraction

```

Valid choices for parameters and their units in FracMan are:

Time	"sec", "day", "hour", "hrs"
Density	"kg/m^3", "kg/m3", "kgm-3", "g/cm^3", "g/cm3", "gcm-3"
Rate	"m^3/s", "m3/s", "m^3/d", "m3/d", "stb/d", "litre/min", "liter/min", "l/min"

Pressure	"m", "bar", "psi", "Pa", "kPa", "atm"
Compressibility	1/Pa, "Pa-1", "Pa^-1", "1/bar", "bar-1", "bar^-1", "1/psi", "psi-1", "psi^-1", "1/kPa", "kPa-1", "kPa^-1", "1/atm", "atm-1", "atm^-1"
Viscosity	"Pa.s", "Pas", "Pa s", "cP"
Permeability	"m/s", "ms-1", "ms^-1", "mD", "Darcy", "D"; "m^2", "m2"
Aperture	"m", "mm", "ft", "feet", "inch"
Porosity	"fraction", "-", "percent", "%"

ISTOR is used to specify the storage space required for registering the non-zero terms in the global linear algebraic equation system for solving the unknown nodal pressures. If there are 100,000 nodes in the finite element mesh, the solver has to calculate the nodal pressures for all these nodes at every time step. There will be 100,000 linear equations for this purpose, and there will be 100,000 by 100,000 (or 10 billion) terms in the global linear algebraic equation system.

To reduce memory requirements, the solver only stores half of the terms (by utilizing the symmetric nature of the problem), and only the non-zero terms. This dramatically reduces the total number of terms in the equation system, in this particular example, from 10 billion to ISTOR terms which the solver has to remember and calculate.

ISTOR is a function of the number of nodes in the finite element mesh, and the connectivity of the fracture network. The latter influences the number of non-zero terms greatly. Unfortunately, the solver cannot estimate the connectivity of a network until it starts to build the non-zero terms. By then, it is too late, and if ISTOR is not large enough, the solver will terminate.

ISTOR is estimated by the solver based on the numbers of nodes and elements. If the user has specified an ISTOR in the units.config file (optional), the solver will use whichever number is greater.

If the ISOTR value is not large enough, the solver will terminate and inform the user about the minimum required. The user may have to increase it several times (by trial and error) until it is large enough (it is not recommended to use a too large number, e.g., 100 million, because this will effect other processes on your computer).

Fortunately, the user will not have to deal with ISTOR very frequently. In the majority of cases, the number estimated by the solver is sufficient. The user does not have to specify an ISTOR number in the "units.config" file (specification of the parameter is optional).

Mesh File

There are primarily two parts in a mesh file:

- Nodal Coordinates and Boundary Definitions;
- Fracture (Triangular) Element Definitions; and

```

NODE      X      Y      Z      iType      H      Q      iGrp
  1  4.6568151855E+02 -9.8826373291E+02  9.1204917431E-01  0  0.0000E+00  0.0000E+00  0
  2  4.6515899658E+02 -9.8807928467E+02  0.0000000000E+00 -1  0.0000E+00  0.0000E+00  6
  3  4.5000000000E+02 -9.8326397705E+02  0.0000000000E+00 -1  0.0000E+00  0.0000E+00  6
.
.
.
  0  0.0000000000E+00  0.0000000000E+00  0.0000000000E+00  0  0.0000E+00  0.0000E+00  0
FracElem Node1 Node2 Node3 Frac# Set# Kf Cfr Apert
  1      2    4596    4597     1     1  1.2583E+05  1.0000E-06  1.0000E-03
  2    4596    4597    4600     1     1  1.2583E+05  1.0000E-06  1.0000E-03
  3    4599    4600    4601     1     1  1.2583E+05  1.0000E-06  1.0000E-03
.
.
.
  0      0      0      0      0      0  0.0000E+00  0.0000E+00  0.0000E+00
MatrixElem Node1 Node2 Node3 Node4 Trans Stor
  0      0      0      0      0      0

```

Well Pressure Results File

The well pressure file, which is generated by the flow solver, contains well pressures for each timestep.

```

| TIME          Well
| (HRS)                (PSI)
0.0000000000E+00  9.9999993896E+02
2.7777778450E-04  -5.3524902344E+03
3.0916504329E-04  -5.6659985352E+03
3.4409886575E-04  -5.9016264648E+03
3.8298004074E-04  -6.1022167969E+03
4.2625452625E-04  -6.2825742188E+03
4.7441883362E-04  -6.4497407227E+03
5.2802538266E-04  -6.6076650391E+03
.
.
.

```

Well Fracture Flow Results File

The fracture flow results file is generated by the flow solver and contains fracture production (+) or injection (-) rates for each timestep.

```

| TIME | Well
|      | TOTAL | TOTAL | FRAC 1 | FRAC 2 | FRAC 3 | FRAC 4
| (MATRIX & | (FRACTURE | 1.500E+01| 1.505E+01| 1.716E+01| 1.508E+01
| FRACTURES) | ONLY) | 4.991E+01| 4.995E+01| 4.997E+01| 4.996E+01
| (HRS) | (M^3/D) | (M^3/D) | -1.077E+01|-1.146E+01|-2.697E+01|-2.311E+01
-----
1.000E+00  1.000E-02  1.000E-02  2.363E-03  3.142E-03  2.219E-03  2.275E-03
2.000E+00  1.000E-02  1.000E-02  2.369E-03  3.152E-03  2.208E-03  2.271E-03
4.000E+00  1.000E-02  1.000E-02  2.372E-03  3.156E-03  2.203E-03  2.269E-03

```



The three numbers under the fracture label indicate the x,y,z coordinates of a the fracture intersection with the well.

Appendix H Third Party Software

Here is a description of the third party software and libraries used in FracMan

Geometric Tools (Wild Magic)

Link

<http://www.geometrictools.com/index.html>

Version 5.6

License (Boost License)

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Haru

Link

<http://libharu.org>

Version 2.2.1

License

Copyright (C) 1999-2006 Takeshi Kanno

This software is provided 'as-is', without any express or implied warranty.

In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

HDF5 (Hierarchical Data Format 5)

Link

<https://support.hdfgroup.org/HDF5/>

Version 1.8.16

License

Copyright Notice and License Terms for
HDF5 (Hierarchical Data Format 5) Software Library and Utilities

HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright (c) 2006, The HDF Group.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright (c) 1998-2006, The Board of Trustees of the University of Illinois.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.
3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.
4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by The HDF Group and by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and credit the contributors.
5. Neither the name of The HDF Group, the name of the University, nor the name of any Contributor may be used to endorse or promote products derived from this software without specific prior written permission from The HDF Group, the University, or the Contributor, respectively.

DISCLAIMER:

THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall The HDF Group or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

Limited portions of HDF5 were developed by Lawrence Berkeley National Laboratory (LBNL). LBNL's Copyright Notice and Licensing Terms can be found here: COPYING_LBNL_HDF5 file in this directory or at http://support.hdfgroup.org/ftp/HDF5/releases/COPYING_LBNL_HDF5.

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois, Fortner Software, Unidata Program Center (netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip), and Digital Equipment Corporation (DEC).

Portions of HDF5 were developed with support from the Lawrence Berkeley National Laboratory (LBNL) and the United States Department of Energy under Prime Contract No. DE-AC02-05CH11231.

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

DISCLAIMER:

This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately- owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

HDF5 is available with the SZIP compression library but SZIP is not part of HDF5 and has separate copyright and license terms. See [Szzip Compression](#)

in HDF Productsâ€™ (www.hdfgroup.org/doc_resource/SZIP/) for further details.

MUParser

Link

<http://muparser.sourceforge.net/index.html>

Version 2.10

License (MIT License):

Copyright (c) 2011 Ingo Berg

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

PCRE - Perl Compatible Regular Expressions

Link

<http://www.pcre.org/>

Version 10.20

License

PCRE2 LICENCE

PCRE2 is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Release 10 of PCRE2 is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE2, supplied in the "doc" directory, is distributed under the same terms as the software itself. The data in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also included in the distribution is a just-in-time compiler that can be used to optimize pattern matching. This is an optional feature that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS

Written by: Philip Hazel

Email local part: ph10
Email domain: cam.ac.uk
University of Cambridge Computing Service,
Cambridge, England.
Copyright (c) 1997-2015 University of Cambridge
All rights reserved.

PCRE2 JUST-IN-TIME COMPILATION SUPPORT

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu
Copyright(c) 2010-2015 Zoltan Herczeg
All rights reserved.

STACK-LESS JUST-IN-TIME COMPILER

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu
Copyright(c) 2009-2015 Zoltan Herczeg
All rights reserved.

THE "BSD" LICENCE

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
- * Neither the name of the University of Cambridge nor the names of any
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

Proj.4

Link

<http://proj4.org>

Version 4.9.3

License

Copyright (c) 2000, Frank Warmerdam

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Rescue

Link

RESCUE (Reservoir Characterization Using Epicenter) is a joint industry project managed by Energestics. Complete information is available on their website at:

<http://www.energestics.org/reservoir/rescue-standards>

Version 37.8

License

Copyright 1995 Petrotechnical Open Software Corporation

POSC grants permission to copy or reproduce this material in its original form for internal use only.

This software is subject to the provision of the "POSC Software License Agreement" which states in part:

- 1) Licensee accepts a non-exclusive, non-transferable, license to use, display, modify and distribute works derived from the licensed documentation and Software Product.
- 2) Licensee shall have no right to distribute in its original form any Software Product or documentation licensed under this agreement.

Scintilla

Link

<http://www.scintilla.org>

Version: 1.76

License

License for Scintilla and SciTE

Copyright 1998-2002 by Neil Hodgson <neilh@scintilla.org>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

ShapeLib

Link

<http://shapelib.maptools.org/>

Version 1.2

License

Copyright (c) 1999, Frank Warmerdam

This software is available under the following "MIT Style" license, or at the option of the licensee under the LGPL (see LICENSE.LGPL). This option is discussed in more detail in shapelib.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

TinyXML

Link

<http://www.grinninglizard.com/tinyxml/index.html>

Version 2.53

License

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

The OpenGL Extension Wrangler Library (GLEW)

Link

<http://glew.sourceforge.net/>

Version 1.7.0

License

The OpenGL Extension Wrangler Library

Copyright (C) 2002-2008, Milan Ikits <milan ikits[]ieee org>

Copyright (C) 2002-2008, Marcelo E. Magallon <mmagallo[]debian org>

Copyright (C) 2002, Lev Povalahev

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * The name of the author may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

VegaFEM

Link

<http://run.usc.edu/vega/>

Version 2.2

License

BSD 3-clause license

Zlib

Link

<http://zlib.net/>

Version 1.2.5

License

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.