

Griddle™

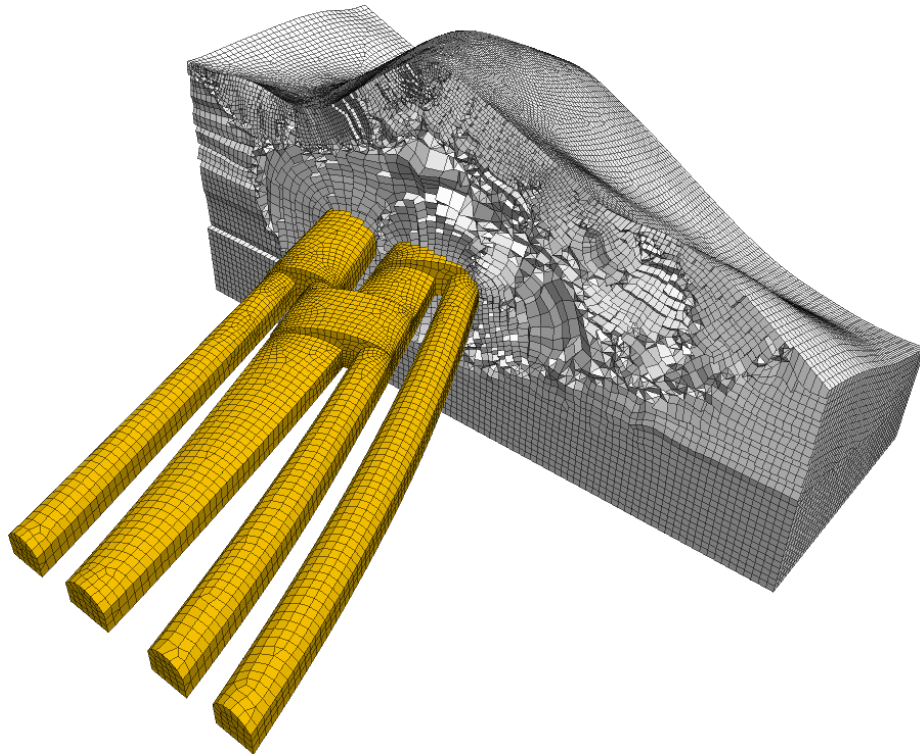


BlockRanger™

Advanced Grid Generation for Engineers and Scientists

Griddle and BlockRanger

User's Guide



©2017

Itasca Consulting Group Inc.
111 Third Avenue South, Suite 450
Minneapolis, Minnesota 55401 USA

phone: (1) 612-371-4711
fax: (1) 612-371-4717
e-mail: software@itascacg.com
web: www.itascacg.com

Contents

Contents.....	1
Quick Start for Griddle, BlockRanger, and Rhino	2
About this manual.....	3
Setting up your work environment.....	3
Using Griddle.....	9
Using BlockRanger	20
FLAC3D and 3DEC Groups in Griddle and BlockRanger.....	23
Tutorial 1: A Single Cylinder (Griddle).....	28
Tutorial 2: Vertical Shaft in a Stratified Soil (Griddle)	32
Tutorial 3: 3D Slope (BlockRanger).....	44
Tutorial 4: Intersecting Circular Tunnels (Griddle).....	60
Tutorial 5: Mesh Cleanup and De-Featuring (Griddle).....	69
Tutorial 6: Open Pit Model from Contour Lines (Griddle)	79
Tutorial 7: Open Pit with Intermittent Faults (Griddle).....	93
CAD Representation of surfaces and volumes.....	111
Tips and tricks of the trade	112
References.....	116

Quick Start for *Griddle*, *BlockRanger*, and *Rhino*

Griddle and *BlockRanger* are mesh generation tools which plug into *Rhino* 5 (64 bit). A single installer (Griddle_xxxx.msi) will install both *Griddle* and *BlockRanger* products on your computer. Your license will determine which components are enabled. *Rhino* is a CAD system (installed separately from *Griddle* and *BlockRanger*), used for constructing model geometry. Instructions are provided below to get started with *Rhino*, *Griddle* and *BlockRanger*.

Installation

1. Download ***Rhino 5 for Windows (64 bit)*** from www.rhino3d.com/download and install it.
2. Download the ***Griddle*** installer (only available in a 64-bit version) from here: www.itascacg.com/software-demo and double click on Griddle_xxxx.msi to install it.
3. Reboot your machine.
4. From the Windows Start menu, find the Itasca group, Griddle 1.0, and click on the “Griddle 1.0 User Files” shortcut. This will move you into a directory with *Rhino* plugins that need to be installed.
5. Double click on **BlockRanger.rhi** and follow the instructions, to install *BlockRanger*.
6. Similarly, double click, each in turn, on **Gint.rhi**, **GSurf.rhi**, **GVol.rhi**, **G_NMExtract.rhi** to install *Griddle* components.
7. Open *Rhino* (64-bit) and select **rvb** and **rui** files from the same directory as above and **drag and drop** these files into the ***Rhino*** viewport.
8. You are done.

Griddle and *BlockRanger* license key

Griddle and *BlockRanger* require a license key to run with full functionality enabled. A license key can be obtained by contacting www.itascacg.com/sales. A *Griddle* license automatically includes a full access to *BlockRanger*. *BlockRanger* can also be purchased separately from *Griddle*. A *BlockRanger* license does not allow access to *Griddle* functionality. *BlockRanger* will also work with a *FLAC3D* v6.0 or a *3DEC* v5.2 license key. If a license key is not present, *Griddle* and *BlockRanger* will operate in a demonstration mode. In demonstration mode, the *Griddle* surface remesher and volume mesher output a maximum of 600 surface or volume elements respectively. The *Griddle* surface mesh intersector does not operate in a demonstration mode. *BlockRanger* will only output VRML format in demonstration mode.

If a network key was purchased, it must be placed in a USB port of a network accessible server. The server requires key server driver software to be installed which can be obtained from here: <https://sentinel.gemalto.com/support-downloads/sentinel-drivers/> (download and install the latest Sentinel Protection Installer). *Griddle* should be installed on a network accessible workstation (not on the server) as described above. To set up access to the server network key from the workstation, start *Rhino* on the workstation, and then select the GSurf icon or type _GSurf on the command line. A dialog box will appear which allows you to specify the server location (the machine with the *Griddle* key).

Griddle and BlockRanger User's Guide and Example Files

A pdf version of the *Griddle* and *BlockRanger* User's Guide and associated example files can be downloaded from here:

www.itascacg.com/software/products/meshing-solutions/griddle-blockranger-manual

or they are accessible from the Windows Start menu if you installed *Griddle* or *BlockRanger*. The user's guide provides step-by-step instructions for creating models in *Rhino* and generating computational grids for *FLAC3D* (Itasca, 2012) and *3DEC* (Itasca, 2013). Output to other formats (*ANSYS*, *ABAQUS*, ...) is as simple as a key press.

If you have questions, please email them to either griddle@itascacg.com or blockranger@itascacg.com.

About this manual

This document describes the use of *Griddle* and *BlockRanger* grid generation tools in conjunction with the *Rhino* CAD system, through detailed tutorials focused on geomechanical applications. *Griddle* and *BlockRanger* feature the latest innovations in CAD-based automatic and interactive grid generation.

Griddle and *BlockRanger* offer engineers both automatic and interactive grid generation capabilities that cover a wide range of volume grid generation needs.

- Interactive solid-based mapped hexahedral meshing (*BlockRanger*).
- Automatic tetrahedral and hex-dominant meshing.
- Meshing of complex geometries featuring internal discontinuities.

Setting up your work environment

Displaying useful *Rhino* Toolbars, buttons and Plug-ins.

Following the Quick-Start Procedure described on page 2, the added toolbars, scripts and plug-ins will be registered and ready for use in *Rhino*.

The *Rhino* command area

The command area is the field (generally on top) where *Rhino* displays text information. The location of the command area can be moved by dragging. Figure 1 shows the *Rhino* window after the *Griddle* and *BlockRanger* toolbars and plug-ins have been installed.

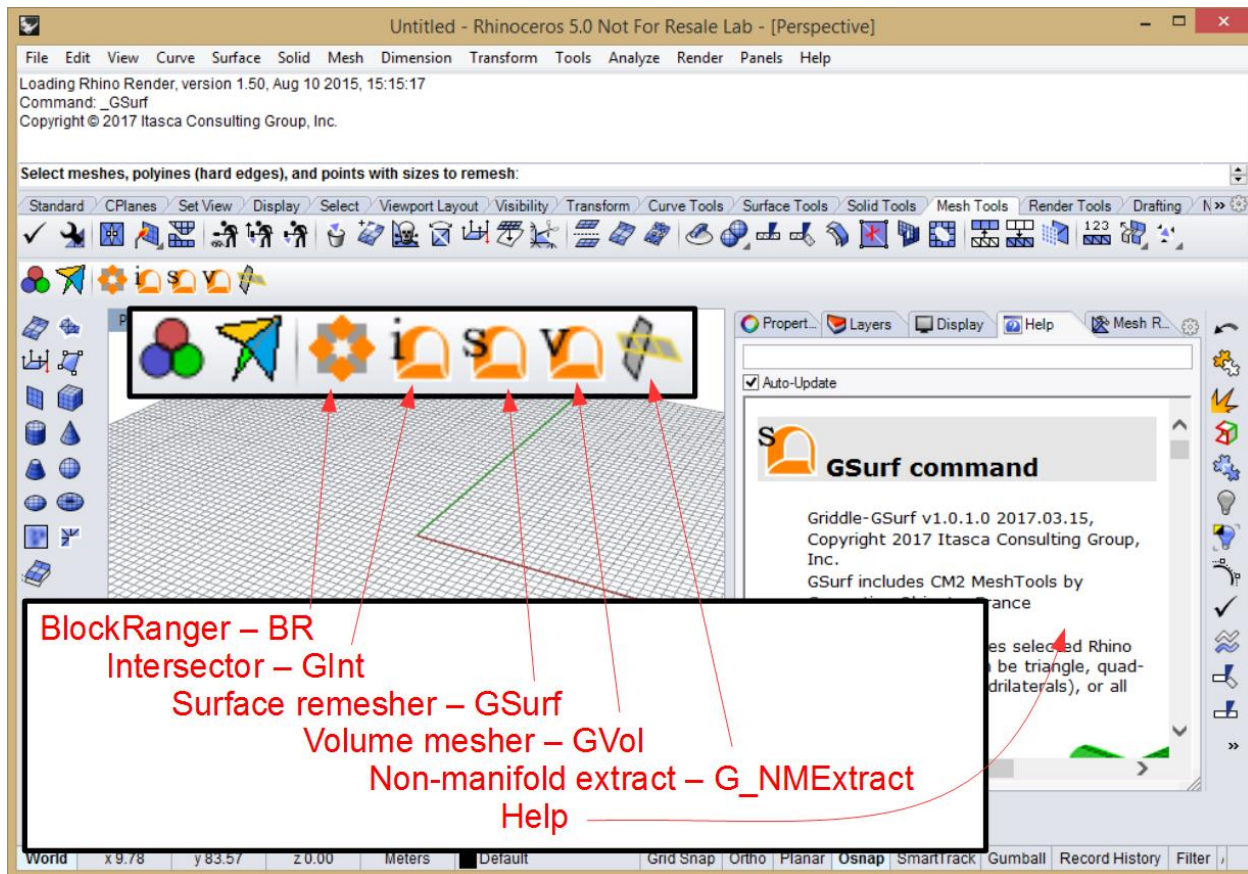


Figure 1: The *Rhino* window featuring the *Griddle-BlockRanger* toolbar (zoomed view).

Icons, menu items, commands and language

Anything you do in *Rhino* is a **command** that you can enter in the command line. For instance, **Shade** is a command which will change the display to shaded. If you enter **_Shade**, it will assume that you refer to the **English** command **Shade**. Therefore, **regardless** of the language in which *Rhino* is set up, **_Shade** creates a shaded image whereas the command **Shade** will only be understood if the installation language is **English**.

During this tutorial, we sometimes refer to a menu item and at other times, we may refer to a command (as in **_Shade**). In all cases, whether you click an icon, select a menu item or enter a command, a command is always executed. **Help | Command Help** opens the Help tab (Figure 2). If Auto-Update is enabled, and the help pane is open, help will immediately appear for *Rhino* commands as they are entered. This help also includes information about where to locate a specific command in the *Rhino* menus, toolbars, and shortcuts (click on the link “Where can I find this command?” in the help shown in Figure 2).

Throughout this manual we refer to *Rhino* functions mostly through commands that are entered in the *Rhino* command line. Sometimes *Rhino* functions are referred to by their icon name or through a menu item. The name of an icon is displayed in a tooltip that pops up when you hover your mouse over the

icon. Often, an icon has two related functions depending on whether you click the left or right button. After you click on an icon, the corresponding Rhino command-line is issued in the command line area.

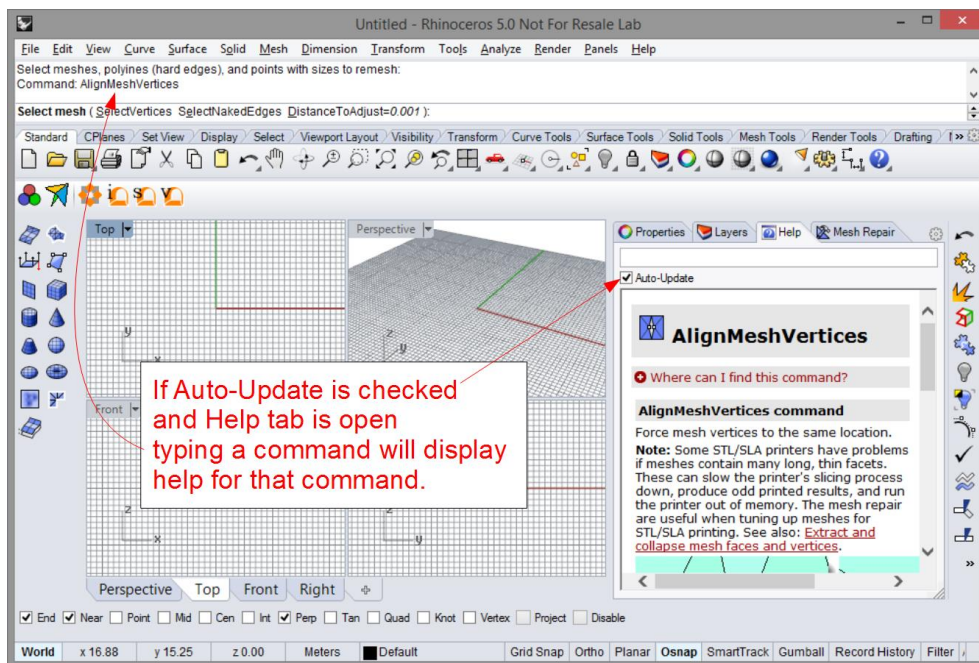


Figure 2: Enabling Auto-Update for Rhino Help.

Rotating and panning views

In the **Perspective** view, you can rotate a view by **right-clicking** and holding the mouse down and moving it around. You can **pan** a view using **<SHIFT>** and the **right** mouse button held down.

Colorizing objects in the model

In general, all newly-created objects in Rhino appear the color of the current Layer. To differentiate objects from each other by color, **select** several objects in Rhino and click on the icon marked **ColorizeObjects** to assign different colors to each selected object (Figure 3). If you want to revert to a colorization **by layers**, select objects, press **F3** to open up the Properties tab, and in the item labelled Display Color, select color by Layer.

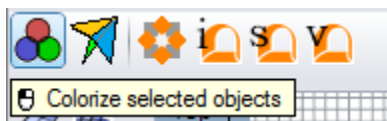


Figure 3: The ColorizeObjects icon and its tooltip in the Griddle-BlockRanger toolbar

Joining non-manifold surfaces

_NonManifoldMerge joins several manifold or non-manifold surfaces into a single non-manifold PolySurface (Figure 4). In its latest implementation, **_NonManifoldMerge** builds a single non-manifold

Polysurface from a selected set of intersecting surfaces and Polysurface. You can subsequently use **_ExtractSrf** to get rid of unneeded surfaces.

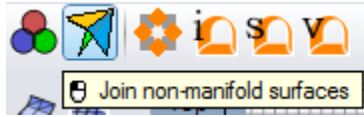


Figure 4: The NonManifoldMerge icon and its tooltip in the *Griddle-BlockRanger* toolbar

Useful *Rhino* shortcuts when working with *Griddle* or *BlockRanger*

The following shortcuts are pre-programmed into Rhino:

F3	Properties
F8	Ortho
F9	Snap

The following **shortcuts may be added to Rhino** to streamline many arduous tasks:

- **AlignMeshVertices:**

The **_AlignMeshVertices** command is very useful for cleaning up meshes. The most common use of this tool involves the following preliminary actions:

1. setting the **DistanceToAdjust** Parameter
2. clicking on **SelectVertices**
3. clicking on the **first** vertex that will **host** all the subsequent vertices
4. Clicking on the vertices that will **collapse** onto the host vertex
5. pressing **<ENTER>**

You can set custom **shortcut, F4**, that does all this. Left-click on the icon marked **Options**, on the left pane, under Rhino Options select **Keyboard**. If the slot in front of F4 is available, type in **_AlignMeshVertices SelectVertices**

- **SplitMeshEdge:**

Similarly, set **F6** to **_SplitMeshEdge**

Tolerances

Tolerances matter especially when intersecting meshes or doing Boolean operations on meshes. In all cases, make sure that your model is never too far away from the origin and, in case it is, **Move** it closer

to the origin. Moving objects closer to the origin improves accuracy and helps with the graphics both in Rhino and in the analysis software.

When starting a new project, use the initial Rhino **template** to specify whether the model will be a "Large object in "meters", "small" object in "feet", etc., then import DXF, STL or even existing 3dm files into the new project. In this fashion you **control the tolerance** instead of using any tolerance **inherited** from the DXF file.

Avoiding accidental object dragging

It may happen that you **inadvertently drag** a highlighted object with the left mouse button. To **avoid** this, open menu item **Tools|Options**. In the left pane of the Rhino **Options** dialog box, click on **Rhino Options|Mouse**, and in the **Click and drag** section, set the **Object drag threshold** to **100** pixels. Now, dragging a highlighted objects requires a minimum of 100 pixels of mouse movement before it takes effect.

SetWorkingDirectory or "where are my files?"

When you **start** Rhino by double-clicking a 3dm file located in a folder, generally Rhino considers that folder to be the **working directory**. *Rhino* will look for input files in this directory and *Griddle* or *BlockRanger* will place the resulting meshes in this directory. But in certain circumstances *Rhino* may be working out of a protected Windows directory, and when you run *Griddle* or *BlockRanger*, Rhino will mention this to you. When in doubt, in Rhino, use the **_SetWorkingDirectory** command to tell Rhino **where** to read and write its files.

Rhino AutoSave

Rhino will automatically save your work after set time interval. This AutoSave feature interrupts *BlockRanger* or *Griddle* meshing operations. If you are creating a huge mesh (millions of elements), the meshing tools may require longer to compute than the AutoSave interval. To avoid interrupting meshing operations, the AutoSave interval can be increased in *Rhino* in **Tools|Options|Rhino Options|Files**. Increase the number of minutes in the **Save every** field.

Object snap options

Two categories of snapping options are available in *Rhino*: **Grid Snap** and **Osnap**. **Grid Snap** is similar to Snap to Grid in *PowerPoint*. **Osnap** or Object Snap is the *Rhino* equivalent of *PowerPoint*'s Snap to Objects. To be able to snap lines, polylines, corners of objects, etc. to existing objects, **Osnap** must be active. Click on the word **Osnap** at the bottom of the *Rhino* window to activate it. Next, you can specify to which particular point of an existing object you want to snap to by checking any of the words: End, Near, Point, Mid, etc. appearing at the bottom the *Rhino* window.

Orthogonal restriction of mouse movement

You can **restrict** the movements of the **mouse** to the x, y and z directions **permanently** by clicking on the word **Ortho** appearing at the bottom of the graphic window or by hitting **<F8>**. You can restrict the movement of the mouse **temporarily** by holding down **<Shift>** during mouse movement.

Getting rid of the background grid

You can hide the **background** grid in any window by hitting <F7> while that window is active. To get rid of the grid in all windows, go to **Tools|Options**. In the left pane of the *Rhino Options* dialog box, click on **Document Properties|Grid**, and in the **Grid properties** section, **uncheck** Show grid lines and Show grid axes.

Thicker lines

The **default** representation of **curves** (1 pixel wide) may be too **thin** and difficult to see when a *Rhino* window is projected on a screen. Figure 5, left, shows the default representation. The following procedure shows how to draw curves **3 pixels-wide** whenever **Shaded View** is selected (Figure 5, right).

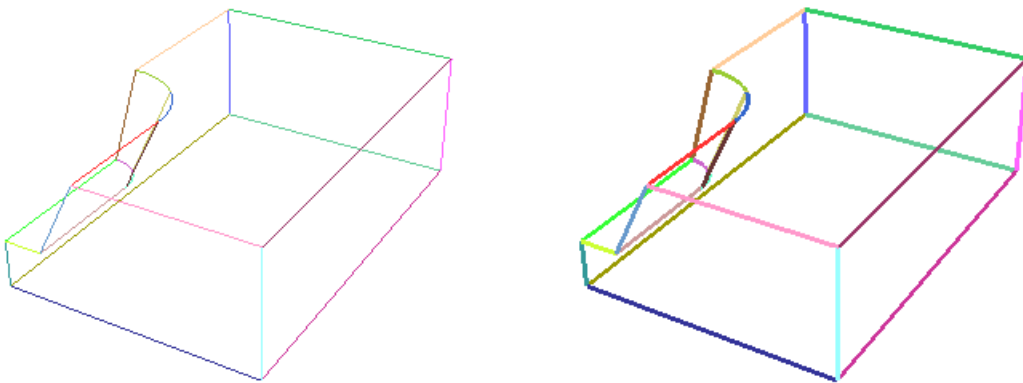


Figure 5: A regular (left) and "thick line" (right) representation.

Select the **Tools|Options** menus item. In the **Left** pane of the **Rhino Options** dialog box, under **Rhino Options**, select **View|Display Models|Shaded|Objects|Curves**. In the Curve Settings pane, to the right, Change the default setting of **Curve width** from 1 to **3** and click **OK** (Figure 6).

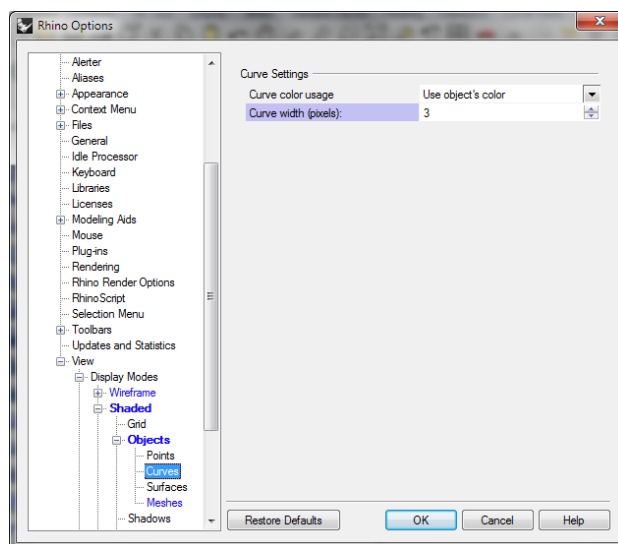


Figure 6: Setting line width to 3 pixels in Shaded Views.

Using Griddle

Griddle is an automatic grid generator accessible from within *Rhino* 5. You can access *Griddle* functions by clicking on the icons shown in Figure 7.



Figure 7: The main Griddle icons in the Griddle toolbar

Griddle consists of three main functions: i) a triangle surface mesh intersector, ii) a surface remesher, and iii) a volume mesher. These three functions are represented by the icons above going from left to right. Each *Griddle* function can be accessed from the *Rhino* command line with **_GInt**, **_GSurf**, and **_GVol** commands respectively for intersector, surface remesher, volume mesher. *Griddle* also has a utility function called **_G_NMExtract** for extracting non-manifold mesh faces from a mesh and this is described later.

All three of the *Griddle* functions operate directly on or with *Rhino* surface meshes appearing on your *Rhino* screen. The meshing workflow with *Griddle* is summarized with the steps shown in Figure 8.

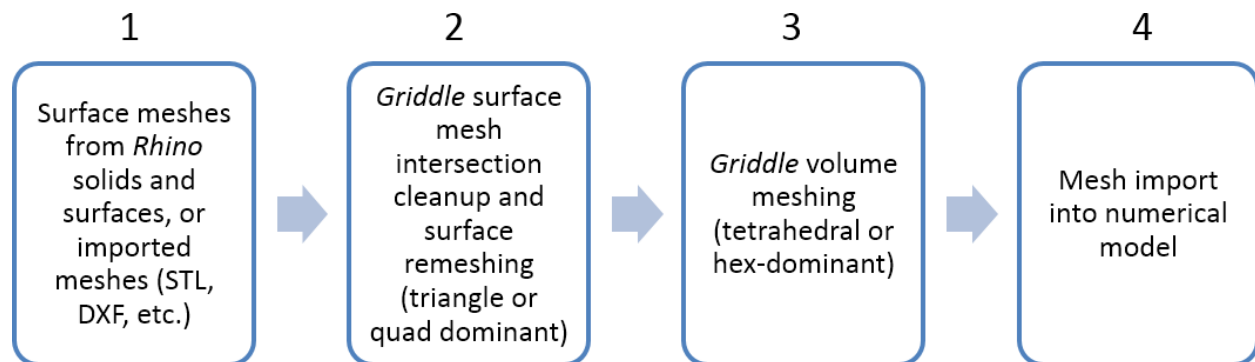


Figure 8: *Griddle* meshing workflow.

Rhino has a rich set of surface meshing tools which allow you to create and edit meshes on NURBS (non-uniform rational basis spline) surfaces and BReps (boundary representations) of volumes. These meshes, although good for machining and prototyping purposes, are usually not suitable for numerical computations. These *Rhino* generated (or imported) surface meshes can be cleaned (properly intersected) and remeshed with **_GInt** and **_GSurf** respectively. Once a desirable surface mesh is obtained, it can be used (along with other surface meshes) as input to **GVol** which fills the interior regions bounded by the surface meshes with tetrahedra or hex-dominant (hexahedra, prisms, pyramids and tetrahedral) elements (or *3DEC* blocks) for use in numerical programs such as *FLAC3D* or *3DEC*. The options for **_GInt**, **_GSurf**, and **_GVol** are described below (and in the *Rhino* help window). Detailed examples follow later in this manual.

GInt Options

_GInt is the Rhino command corresponding to *Griddle's* surface intersector also callable with the icon above. **_GInt** is used to clean up surface meshes that are not properly intersected. The two other *Griddle* functions (**_GSurf** and **_GVol**) require conformal surface meshes as input. That is, triangles and quadrilaterals forming the surface meshes must share edges and points at their intersections. Figure 9 shows two meshes that are input to **_GInt** to produce a single conformal mesh.

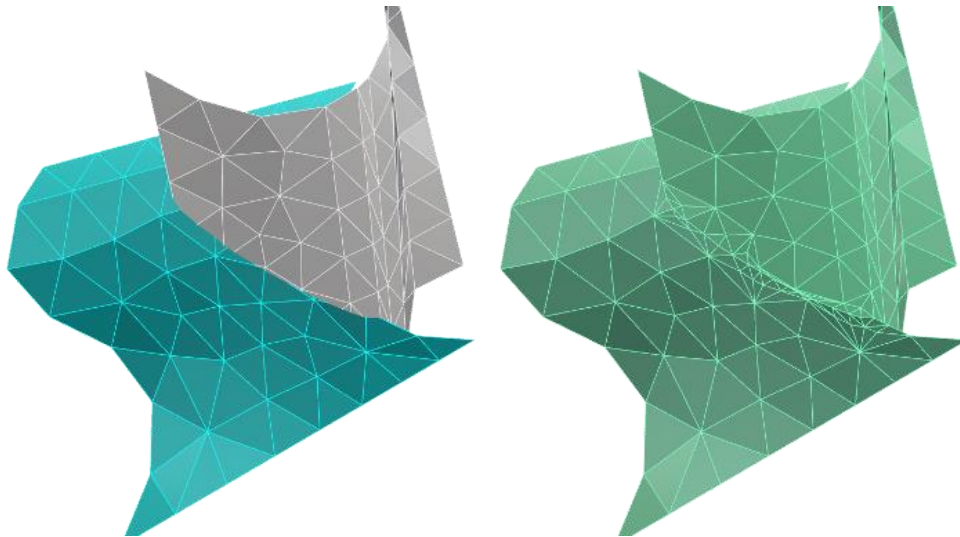


Figure 9: Non-conformal surface meshes (left) and conformal mesh after intersecting with **_GInt** (right).

_GInt takes as input any number of *Rhino* surface meshes. If the surface mesh contains quadrilaterals, these quadrilaterals are automatically converted into triangles (by arbitrarily inserting one of the two possible diagonal edges splitting the quad face) prior to **_GInt** running.

_GInt will not intersect parallel triangles. This includes duplicate triangles. These triangles should be cleaned up prior to running **_GInt**. Check your mesh for duplicate faces and fix it with the *Rhino* **_MeshRepair** panel prior to running **_GInt** (type **_MeshRepair** on the command line to access this functionality).

If you generated your surface meshes from *Rhino* NURBs or BReps then it is better to intersect these mathematical entities with *Rhino's* Boolean functions or with the **_NonManifoldMerge** command and then **_Mesh** these intersected mathematical surfaces with *Rhino*. Intersections of NURBs and BReps are more accurate than intersecting triangle approximations to these surfaces.

As can be seen in Figure 9, **_GInt** does not remesh everything. It only remeshes triangles that are intersected. The entire mesh can be remeshed to a desired element size with *Griddle's* **_GSurf**.

_GInt has three options described below.

Tolerance

Tolerance is an absolute distance (in model units) used to determine whether triangle faces intersect each other. If any negative value is specified, a small (near to zero) default tolerance value is used.

ShowIntersections

The default for this option is No. If Yes is selected, then intersecting areas of meshes are shown as Rhino curves (Figure 10). Use the Rhino **_SelCrv** command to select the intersections curves and highlight them in your view. When **ShowIntersections=Yes**, the **DeleteInput** option (below) is ignored; your selected surface meshes will not be altered in any way. This option is useful for finding isolated problem areas that occasionally remain after remeshing and can usually be cleaned up manually using Rhino's mesh editing tools.

DeleteInput

This option will delete your original selected meshes if Yes (default) is selected. If No is selected your original meshes remain intact.

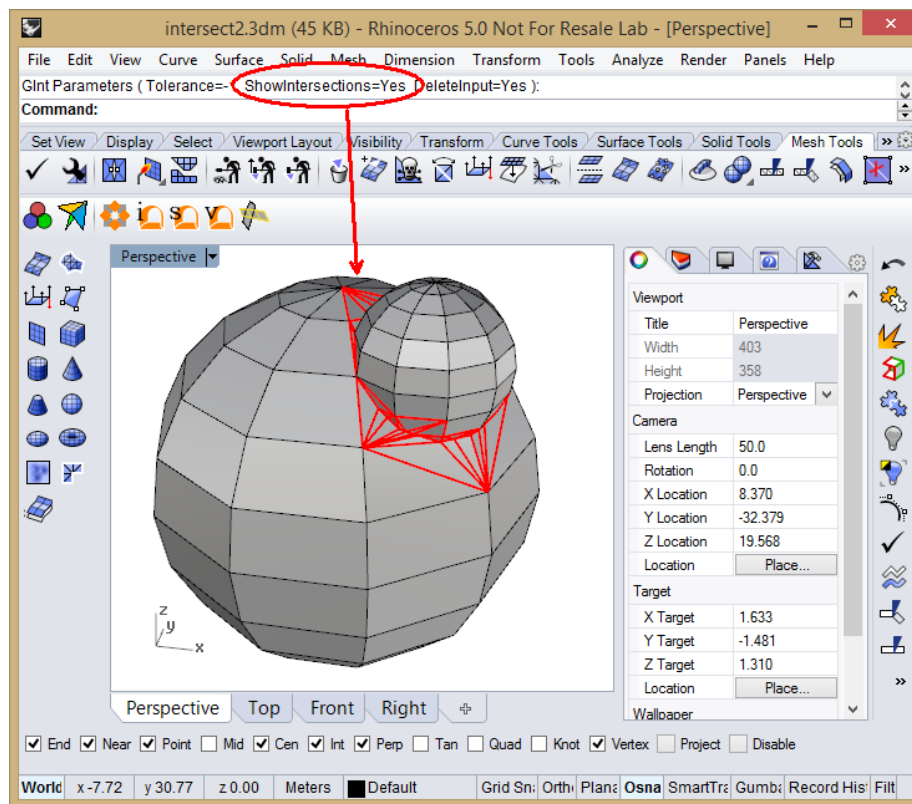


Figure 10: Traces of intersecting faces shown with GInt's ShowIntersections option.

GSurf Options

_GSurf is the *Rhino* command corresponding to *Griddle's* surface remesher also callable with the icon above. **_GSurf** is used to remesh selected surface meshes to desired element edge size and type (triangle, quad-dominant, or all-quad). Surface meshes are used as input into the *Griddle* volume mesher **_GVol**. The volume mesher uses the surface meshes to determine solid element size and type. Input surface mesh faces will appear as faces of solid elements in the volume mesh so it is important to spend the time generating good surface meshes. Input to **_GSurf** must be one or more conformal, properly intersected, meshes (or meshes that do not intersect at all).

You can work on meshes piecemeal. For example, if a finer mesh is required in one portion of your model, you can remesh those surfaces independently of the surfaces where a coarser mesh is desired. In the end, as long as all mesh intersections are conformal, and meshes form a closed volume, these meshes will be suitable as input to the *Griddle* volume mesher.

GSurf options are described below.

Mode

GSurf has three modes:

1. **Tri** (default) produces an all-triangle surface mesh.
2. **QuadDom** produces a quad-dominant surface mesh (contains a mix of triangles and quadrilaterals).
3. **AllQuad** produces an all quad surface mesh.

MinEdgeLength, MaxEdgeLength

These two parameters control the resulting minimum and maximum edge size in the final surface mesh. To get uniform sizes, minimum and maximum edge size can be set to the same value. Edge size is specified in model units.

RidgeAngle

This parameter, specified in degrees, controls the level of detail (the number of ridge lines) in the resulting mesh. The angle between mesh faces sharing an edge is termed a ridge angle (angle=0 degrees if faces are parallel, 90 degrees if faces are perpendicular). Ridge lines can be traced through the surface mesh by joining edges of faces that have ridge angles greater than the specified RidgeAngle. A higher RidgeAngle results in less ridge lines (less detail) included in the final mesh. A lower RidgeAngle results in more detail included in the final mesh. Generally, RidgeAngle should be kept below 45 degrees. 20 degrees (the default) is a good compromise between mesh size and fidelity.

MaxGradation

A positive value, default=0.1, controls the gradation of element sizes. A value close to 0 leads to a more gradual variation of mesh size (smoother) while higher values lead to more abrupt changes in element size.

DeleteInput

This option will delete your original selected meshes if yes (default) is selected. If no is selected your original meshes remain intact.

Additional Edge Size Control Options

In addition to the min and max edge size specified in the parameters above, local overrides can be made to the edge sizes.

Local size of a portion of a mesh can be specified by setting that mesh's name (in the Properties tab for that mesh) to a numeric value representing the required edge size as shown Figure 11. If a mesh does not have a size specified in its Name field, then the mesh size will be determined by the MinEdgeLength and MaxEdgeLength parameters.

Similarly, local edge size can be specified at a point (Figure 12), by creating a point (with the *Rhino* **_Point** command) and assigning to that point an edge size (a numeric value) to the point name (in the Properties tab). These points must be coincident with existing mesh vertices (enable vertex snapping in Rhino and snap your point to a mesh vertex). The resulting mesh will include this original point.

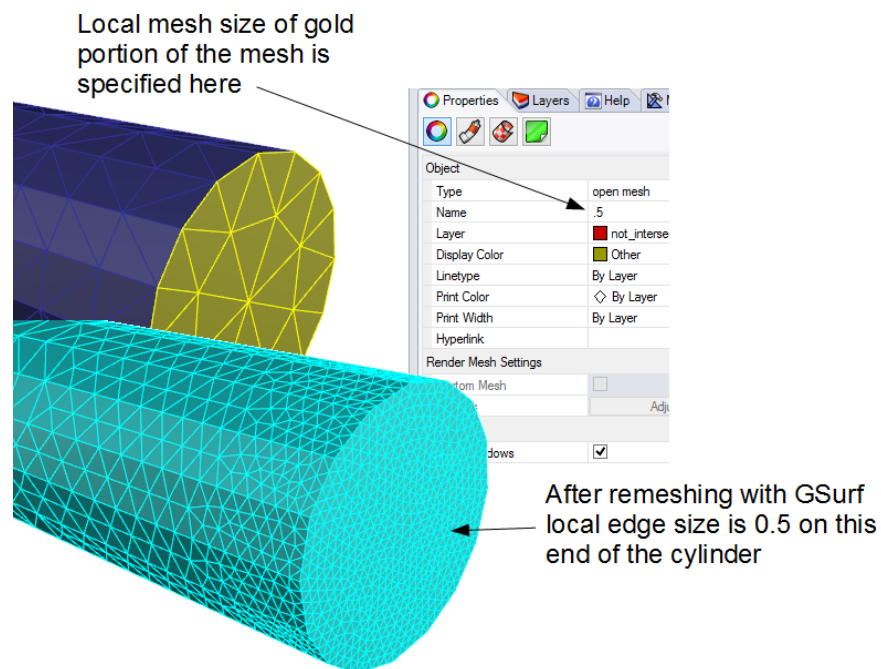


Figure 11: Specifying local edge size for a portion of a mesh.

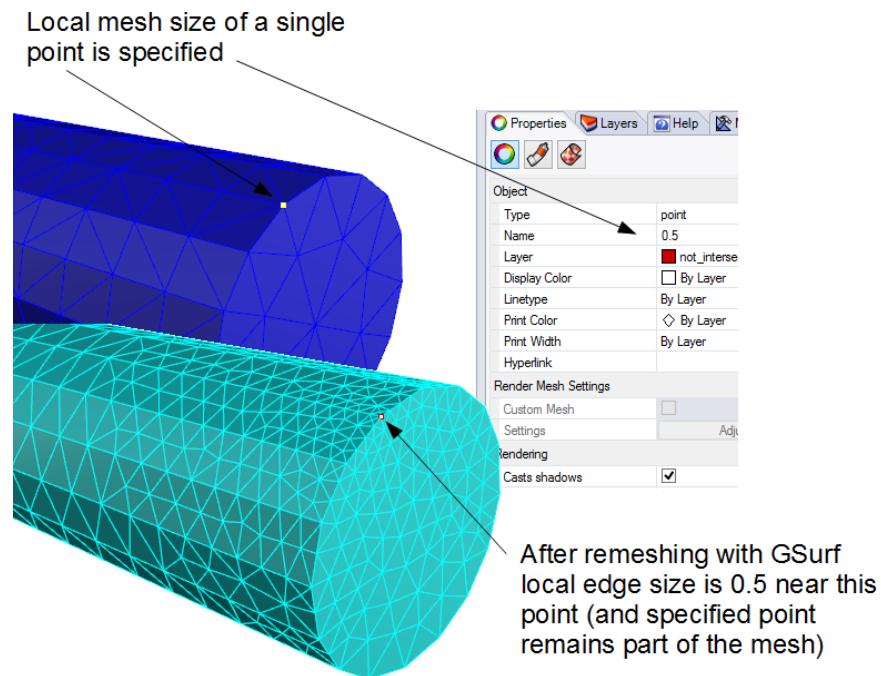


Figure 12: Specifying local edge size at a point.

Hard Edges

Hard edges are edges that are preserved in a mesh during the remeshing process. Referring to the left side of Figure 13, we would like the green mesh to remain intact and have a finer red mesh, but we want both meshes to connect properly without remeshing the green mesh. 1) use the *Rhino _PolyLine* command to draw a polyline, with segments corresponding to surface mesh edges you wish to preserve (this is shown by the yellow line, which contains 5 segments). 2) Hide the green mesh. 3) Select the red mesh and the polyline as input to *_GSurf* and press Enter. *_GSurf* will treat the polyline edges as edges to preserve. The new red mesh is finer, but still conforms to the green mesh (right side of Figure 13). The *Rhino _DupBorder* command is also useful for duplicating a mesh outline with connected line segments.

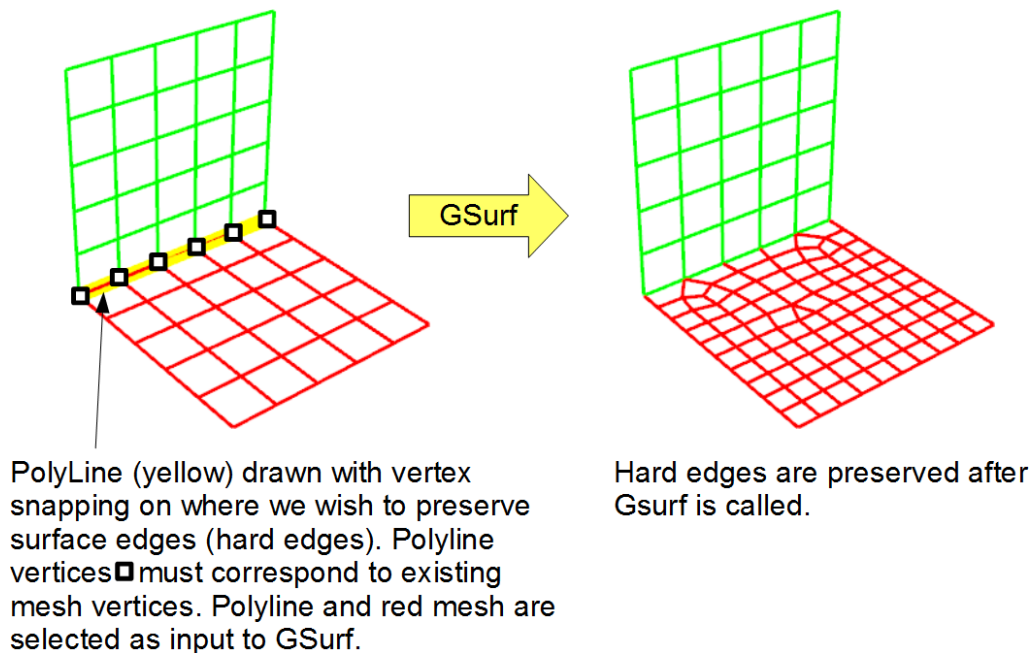


Figure 13: Specifying hard edges.

GVol Options

_GVol is the *Rhino* command corresponding to *Griddle's* volume mesher also callable with the icon above. The **_GVol** command creates a tetrahedral or a hex-dominant mesh using selected surface meshes as boundaries (Figure 14). Surface meshes can be composed of triangles, a mix of quadrilaterals and triangles, or all quadrilaterals. The latter two must be used for generating hex-dominant grids. Tetrahedral grids can be generated from triangle, triangle-quad, or all-quad surface meshes (quadrilaterals are arbitrarily split, along one of their diagonals, into triangles). A combination of selected surface meshes must form a watertight boundary surrounding the entire volume of interest. Surface meshes can also separate discrete volumes within the larger volume. Surface meshes can also “float” inside the volume of interest or be partially connected to other surface meshes. All surface mesh faces, including “floating” surface meshes inside a volume are included as “hard faces” in the final volume mesh. That is, you will see input surface faces as faces of elements in the resulting volume grid.

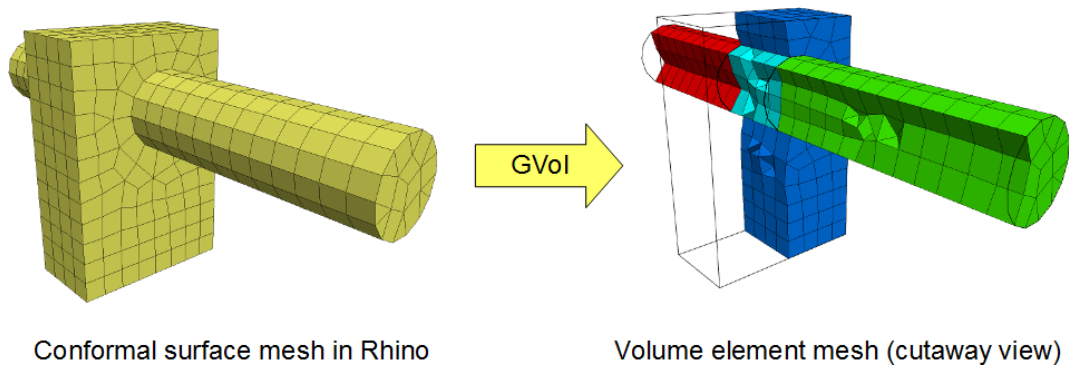


Figure 14: Volume mesh creation from surface meshes.

_GVol requires input surface meshes to be properly connected. Duplicate, overlapping, intersecting triangles and quadrilaterals are not allowed. Check your mesh for errors with the *Rhino* _MeshRepair panel prior to running _GVol (type **_MeshRepair** on the command line). If your mesh contains intersections, these areas can quickly be located using _GInt with the ShowIntersections option set to Yes. If _GVol encounters problems during meshing, traces of the problem areas will be placed into a Rhino layer called MESHING_ERRORS (Figure 15). This layer will contain Rhino curves and points, outlining areas of your surface meshes that caused problems.

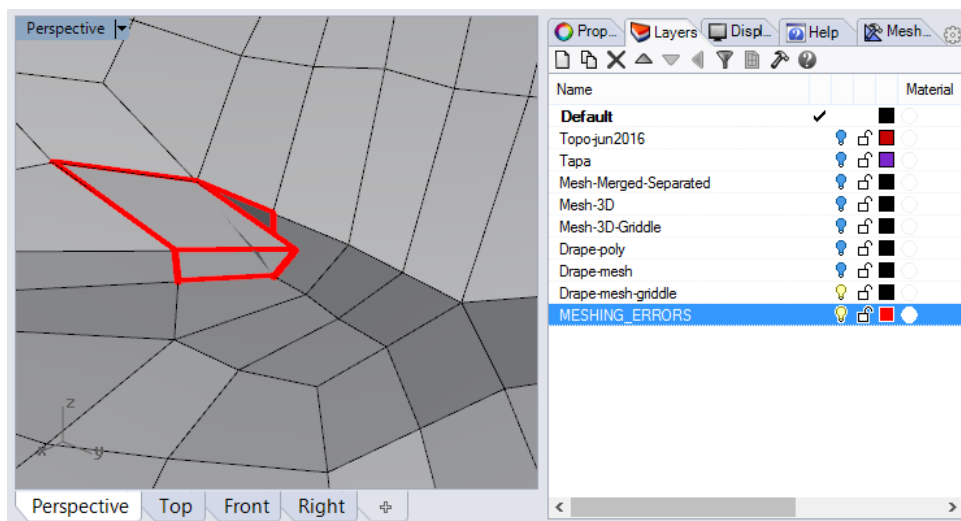


Figure 15: Traces of problem areas placed in MESHING_ERRORS layer.

_GVol options are described below.

Mode

_GVol has two modes:

1. **Tet** (default) produces an all-tetrahedral volume mesh. Any quadrilaterals on surface meshes are converted to triangles in this case.
2. **ConHexDom** produces a conformal hex-dominant volume mesh. Input surfaces must contain quadrilaterals for this option.

OutputFormat

Format in which the resulting volume grid file should be saved. Choices are: FLAC3D, 3DEC, ABAQUS, ANSYS, and NASTRAN. If running interactively, GVVol will prompt for an output file name and location, otherwise output will be sent to a file named GVVol.xxx in the current working directory (the xxx extension being f3grid, 33dat, inp, cdb, or bdf corresponding to the volume grid options mentioned above).



G_NMExtract

_G_NMExtract is the Rhino command corresponding to a *Griddle* utility function that is useful in selecting portions of non-manifold surface meshes. The **_G_NMExtract** command extracts a set of faces from a joined Rhino mesh. This command is similar to the Rhino command **_ExtractConnectedMeshFaces** except the **_G_NMExtract** command allows a consistent manner of selecting mesh faces of non-manifold meshes as shown in the example below. The selected face in Figure 16 is connected to a non-manifold edge that is common to four faces. The **_G_NMExtract** command allows you to extract the faces connected to this highlighted face based on a break angle and whether you wish to stop the extraction at non-manifold edges.

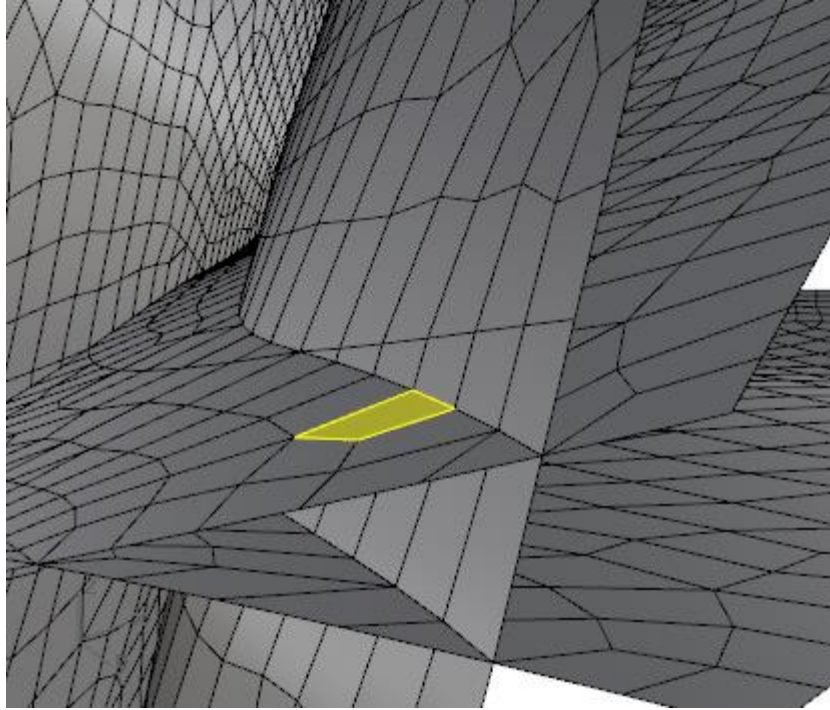


Figure 16: Selected mesh face that contains a non-manifold edge.

The `_G_NMExtract` command has two options: 1) a maximum break angle and 2) break at non-manifold edges. The first option specifies the angle between joined faces (in degrees). 0 degrees will give you joined faces that are coplanar with the face you picked. At a non-manifold edges, `_G_NMExtract` will propagate its selection of faces to those faces having the smallest angle with the currently processed faces (i.e., at a non-manifold branch point, `_G_NMExtract` will follow the smoothest path). The second option allows you to stop the selection at non-manifold edges. The figures below show the result when break at non-manifold edges is on (Figure 17) or off (Figure 18).

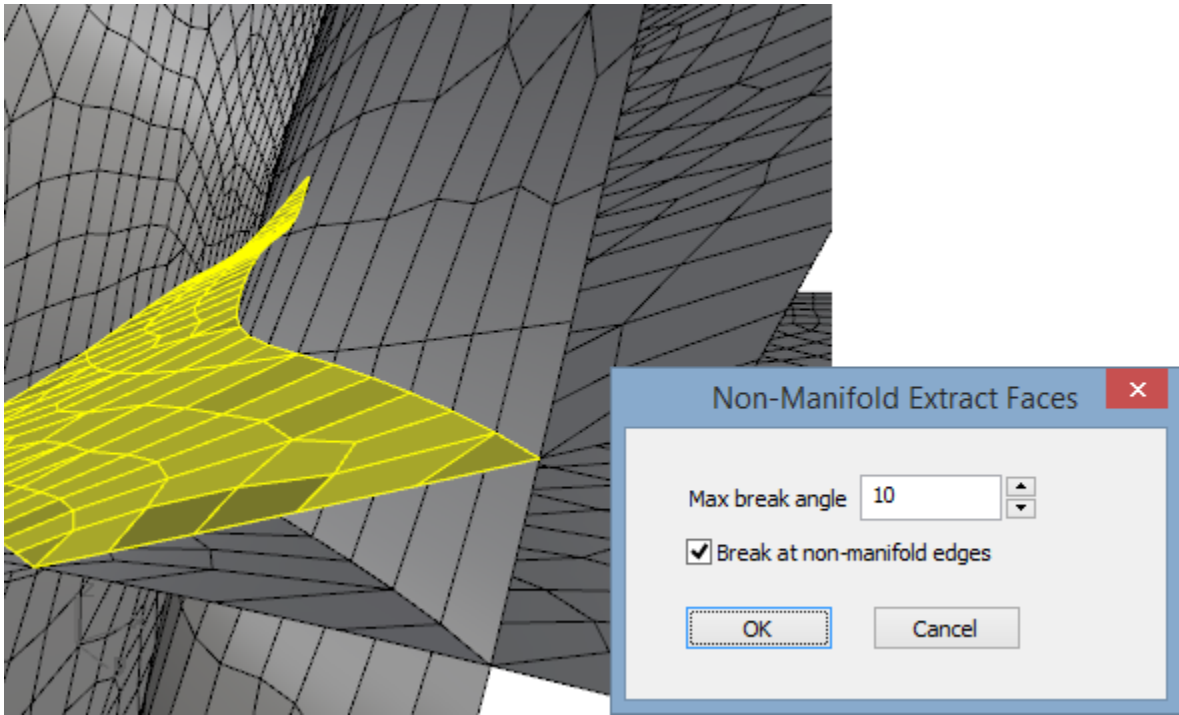


Figure 17: Mesh face selection, breaking at non-manifold edges.

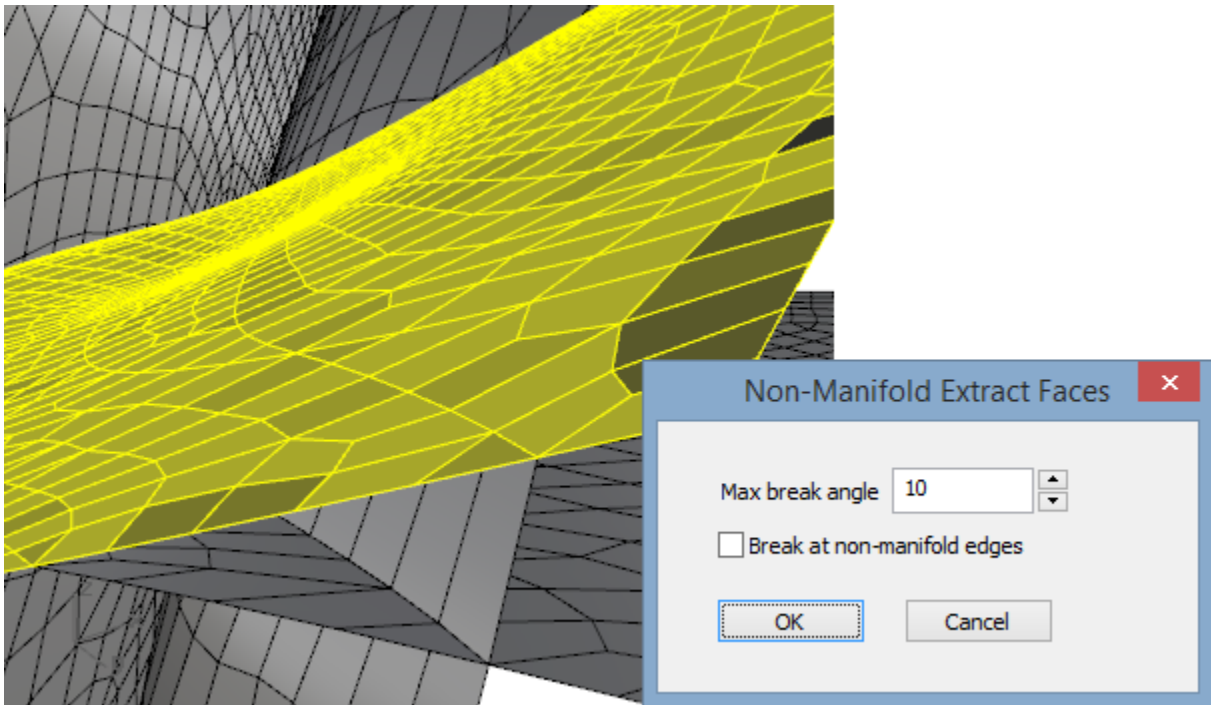


Figure 18: Mesh face selection, ignoring non-manifold edges.



Using *BlockRanger*

BlockRanger is an interactive all-hex mapped mesher. The *Rhino* command to run *BlockRanger* is **_BR** and the icon to activate it is shown in Figure 19.



Figure 19: The *BlockRanger* icon in the *Griddle* toolbar

The **_BR** command converts selected 4, 5 or 6-sided solids into blocks of high-quality hexahedral (brick) elements for use with many engineering analysis packages, including Itasca's *FLAC3D* and *3DEC*, and saves the resulting mesh as an ASCII file in your (*Rhino*'s) Current Working Directory. Tutorial 3: 3D Slope (*BlockRanger*) is devoted to the use of *BlockRanger*.

Admissible *BlockRanger* solids (Figure 20) are:

- 6-sided solids (hexahedron-like) composed of 6 surfaces each bounded by 4 curves.
- 5-sided solids (prism-like) composed of 2 triangle-like surfaces bound by 3 curves and 3 quad-like surfaces bound by 4 curves.
- 4-sided solids (tetrahedron-like) composed of 4 triangle-like surfaces each bound by 3 curves.

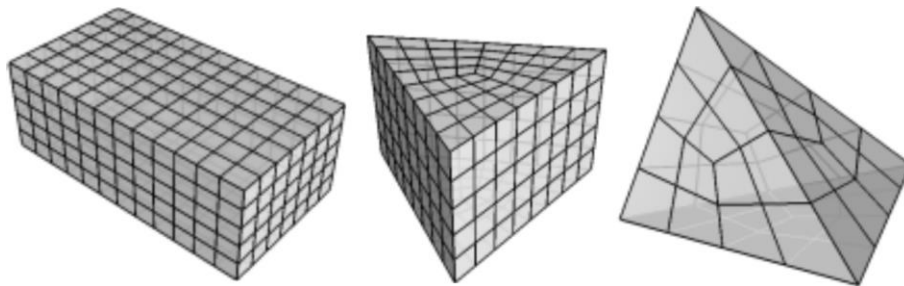


Figure 20: *Rhino* solids recognized by *BlockRanger* with example *BlockRanger* mesh patterns for each.

If the selected solids are contiguous, *BlockRanger* ensures that the resulting grid maintains grid conformity and continuity across block corners, edges and faces so that no dangling nodes will result.

All solids successfully processed by *BlockRanger* are saved as grid files in *Rhino*'s Current Working Directory (see *Rhino*'s **_SetWorkingDirectory** command) as an ASCII file of the form **BlockRanger.xxx**. Solids that did not qualify or could not be successfully meshed remain highlighted on your screen. Use **_Invert** and **_Hide** commands make these solids visible and correct them.

BlockRanger Options

MaxEdgeLength

Maximum element (zone) edge length in model coordinates. By default, this number is calculated as one tenth of the length of the longest edge in your model.

MinEdgeResolution

Minimum number of elements across each grid block edge. Its default value is 3.

TargetNumElements

Use this option if you want to reduce the maximum element aspect ratio in the grid. *BlockRanger* will initially build a grid based on the prescribed MaxEdgeLength and MinEdgeResolution. In a typical block-structured grid, as *BlockRanger* reduces the maximum aspect ratio, the number of elements (zones) increases. *BlockRanger* stops when the number of elements exceeds TargetNumElements.

OutputFormat

Format in which the resulting volume grid file should be saved. Choices are: ABAQUS, ANSYS, FLAC3D, 3DEC, LS-DYNA, NASTRAN, and VRML.

Local Edge Resolution Control

Once a grid has been built using the above global parameters, you can control the local mesh resolution by manually setting the edge resolution you want. A summary of usage is provided below using bathtubSolids.3dm model which you can find in your Tutorial Files (Figure 21).

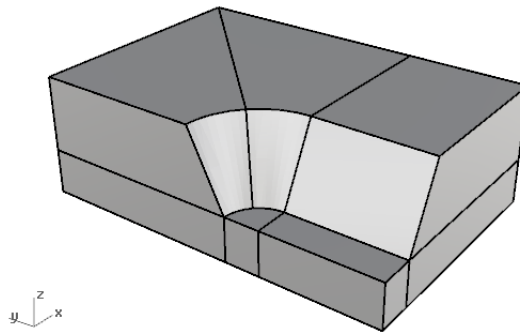


Figure 21: 3D slope example.

Running *BlockRanger* using the default parameters results in the grid below (Figure 22).

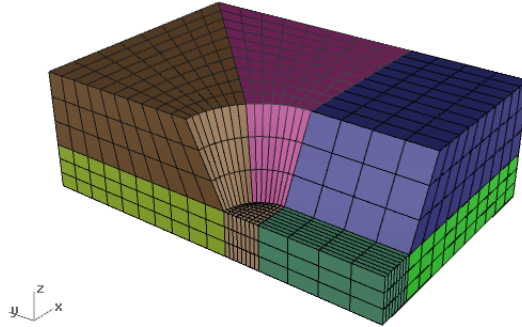


Figure 22: *BlockRanger* generated mesh with default parameter values.

Use the command **_DupEdge** to extract an edge from a *Rhino* solid and set the number of subdivisions for that edge. For example, in Figure 23, the highlighted edge on the right side is extracted first. Once this edge is extracted and highlighted, F3 is pressed to open its Properties dialog box, and in the Name field for the selected edge a number 8 is entered. 8 will be the resolution of that edge. A second edge is extracted using **_DupEdge** (left side of picture below) and the number 5 is entered in its Name field.

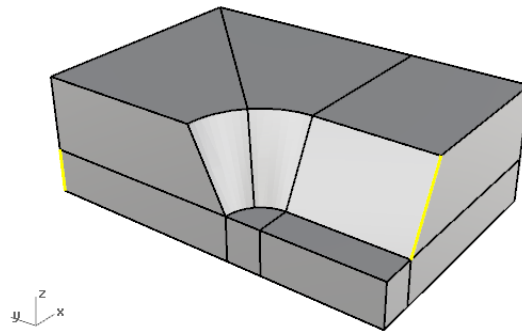


Figure 23: Duplicated edges (highlighted) which had subdivisions specified.

BlockRanger is run again, this time making sure to include the edges in the selection with the solids. This results in the grid below (Figure 24).

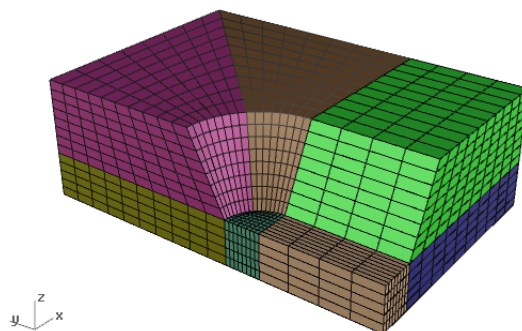


Figure 24: *BlockRanger* generated mesh with specified edge subdivisions enforced.

FLAC3D and 3DEC Groups in Griddle and BlockRanger

FLAC3D and *3DEC* have the ability to associate names with volume elements (zones, blocks) and names or numbers with faces (face groups, joint id). These named entities make it easier to reference areas of the model for material or boundary condition assignment.

Griddle and *BlockRanger* deal with groups in a slightly different manner. *Griddle*, when generating a volume grid, requires a closed set of boundary surfaces. The inside can be composed of “floating” surfaces, i.e., surfaces which are partially or completely disconnected from other surfaces. With *Griddle*, it is not immediately clear if a set of surfaces enclose a watertight volume. For this reason, volumes cannot be directly named (only surfaces can be named). On the other hand, *BlockRanger* requires solids as input. There is no ambiguity about what is inside or outside a solid. For this reason, *BlockRanger* allows you to name the solids. *BlockRanger*, does not allow “floating” surfaces and surfaces in *BlockRanger* do not use names.

Griddle Generated Names

A hypothetical section through a *Griddle* volume mesh generated for *FLAC3D* is shown in Figure 25. After filling the interior of the model domain with zones (elements), *Griddle* assigns a number to each watertight volume. These numbers are shown as the large white numbers below. These numbers are used to generate group names for zones in *FLAC3D*. To distinguish zone groups from other group names, zone groups are given the prefix “ZG_” by *Griddle*. The suffix is the white number, in Figure 25, preceded by a number of zeros. For, example, area 1 can be referenced as ZG_001 in *FLAC3D*. The “ZG_” groups are stored in zone slot 1. Zone faces (represented by the thick black and red lines below) are placed in face groups. *Griddle* classifies the face groups as external faces, which are given an “EF_” prefix and internal faces, which are given an “IF_” prefix. An external face has a zone on one side only. An internal face has zones on both sides of the face. External face names are suffixed with the zone group number they butt up against. Internal face names are suffixed with a combination of two zone group numbers representing the two groups on either side of it (the larger number is always placed second in the name). You will notice that some of the surfaces do not completely divide the volume into distinct pieces. These “floating” faces are represented by the thick red lines. Their face group name contains a duplicate number, e.g., IF_002_002 has two ZG_002 zones on either side of it.

The similar geometry, only this time with a *3DEC* generated model, is shown in Figure 26. *3DEC* block groups are represented by character strings containing numeric digits. The block groups are 10000 + the color index of that block, converted to a string. Joint set ids are integer values. *Griddle* outputs all external faces as joint set 1 (these are ignored by *3DEC*). Joint set 2 refer to all the joints inside a colored region. Joint sets 3, 4, 5, ... are given to the surfaces separating the various block groups as well as the “floating” surfaces. Other than joint id 1 and 2, the joint set id numbers are arbitrarily assigned (there is no relation between joint set id and the block group names surrounding it).

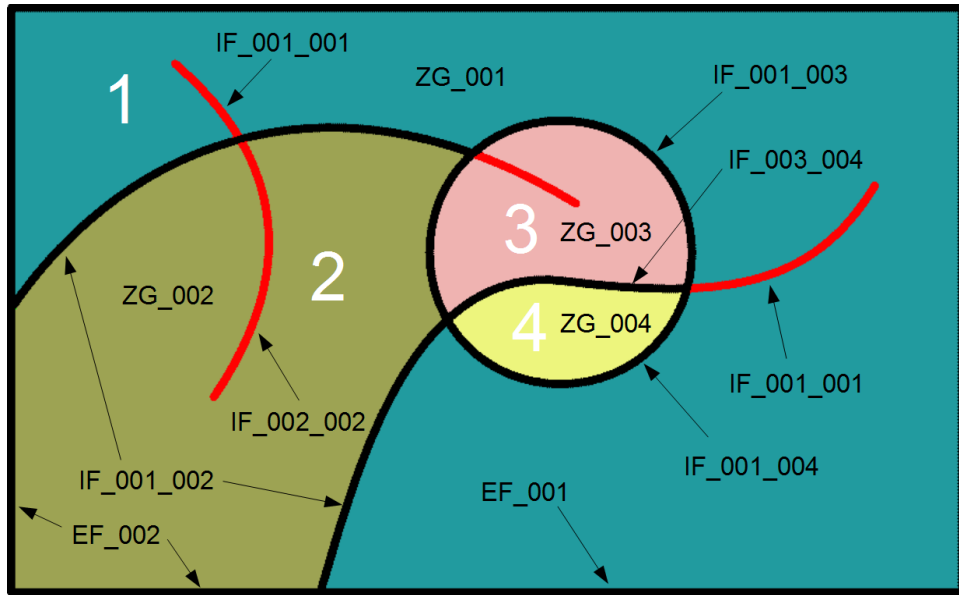


Figure 25: Zone and face groups from a *Griddle* generated *FLAC3D* grid.

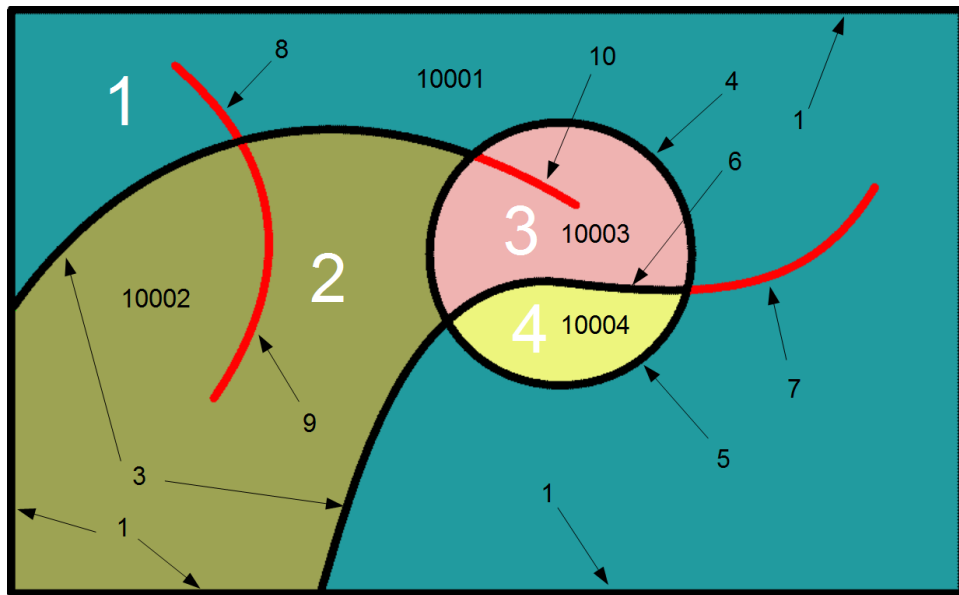


Figure 26: Block groups and joint set ids from a *Griddle* generated *3DEC* model.

Once you have a conformal surface mesh that is ready for volume meshing, you can use *Griddle's* **_G_NMExtract** command or *Rhino's* **_ExtractConnectedMeshFaces** to select mesh faces and unjoin them from the mesh. These separated faces can be named in that mesh's Name field in its Properties tab (Figure 27). After you have extracted and named the various meshes you can select them all and volume mesh them with **_GVol**. **_GVol** will use these names for assigning face group names and joint id numbers. In the example below, an internal mesh is named *rev_fault_1A*. The resulting *FLAC3D* grid would have a face group called "IF_rev_fault_1A" (note, *Griddle* adds an "IF_" or "EF_" prefix to your name assignment). These names take precedence over the automatically assigned names shown in Figure 25. If all the faults were assigned the same name, then you would end up having a single internal face group with that name. These named meshes get assigned a unique *3DEC* joint set id number for all the faces with that name.

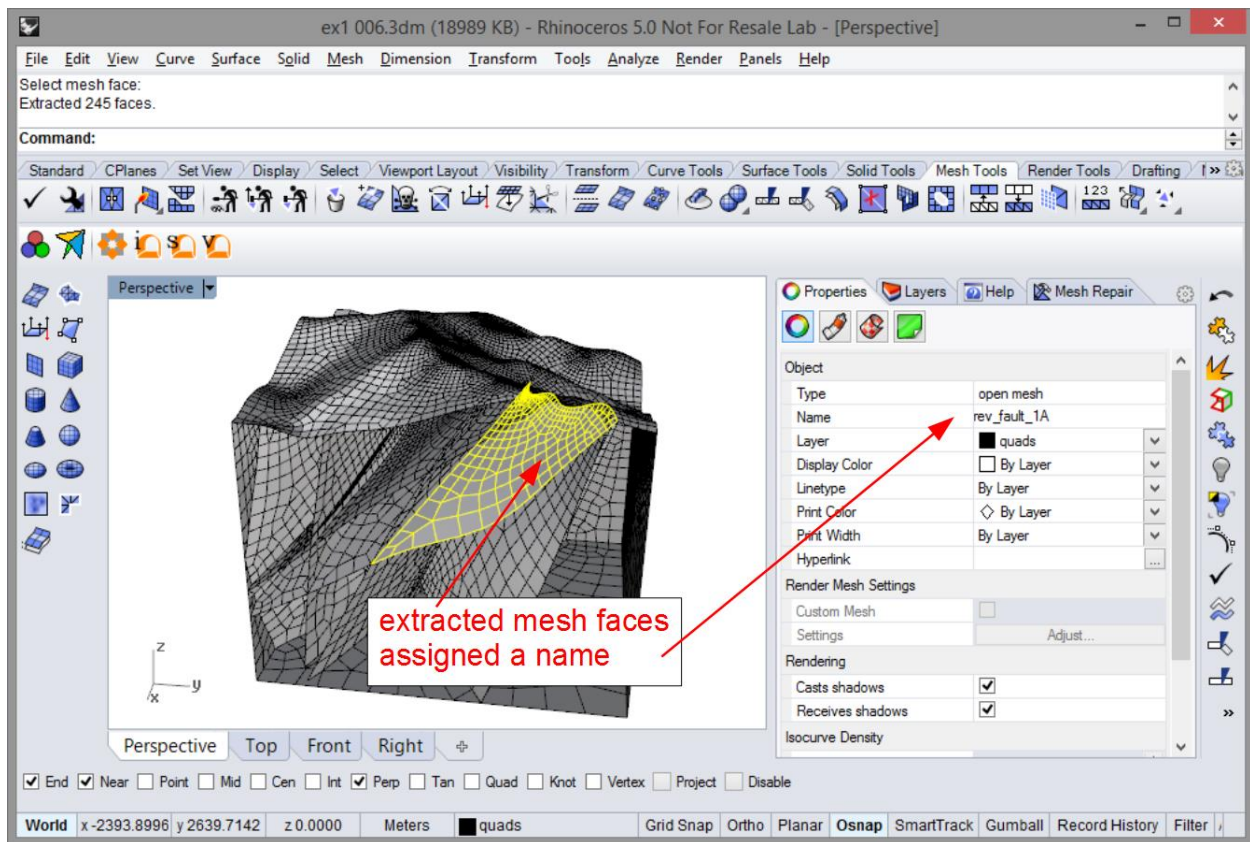


Figure 27: Assigning a name to extracted mesh faces.

BlockRanger Generated Names

Figure 28 and Figure 29 are the *BlockRanger* equivalents of Figure 25 and Figure 26 respectively. *BlockRanger* solids can be named in two ways: 1) through the layer name that the solid belongs and through the solid Properties Name field (Figure 30). The layer name character string is used to generate zone group names for *FLAC3D* in slot 1 and block names for *3DEC*. The Properties Name string is used to generate zone group names for *FLAC3D* in slot 2 and is not used for *3DEC* output. The white text names in Figure 28 and Figure 29 are assumed to be *Rhino* layer names. In Figure 28, the black text prefixed with “ZG_” indicates slot 1 zone groups. The yellow text indicates slot 2 zone groups. Slot 2 group names are automatically generated if the Properties Name field is left blank. A *Rhino* layer name cannot be blank. Example internal (“IF_” prefix) and external (“EF_” prefix) face group names are also shown in Figure 28. Note, a face group is not generated for the seam between the two bedrock solids or the seam between the two soil solids.

For *3DEC* (Figure 29) *BlockRanger* uses the layer name as the *3DEC* block group name. *Rhino* layer “soil” would result in a *3DEC* block group called “soil”. Joint sets are referenced by integer id number. As before, external faces are assigned joint set id 1, internal faces that have the same block group on either side are assigned a joint set id 2. The remaining internal faces (that separate different named layers) are assigned joint set ids of 3, 4, 5, ...

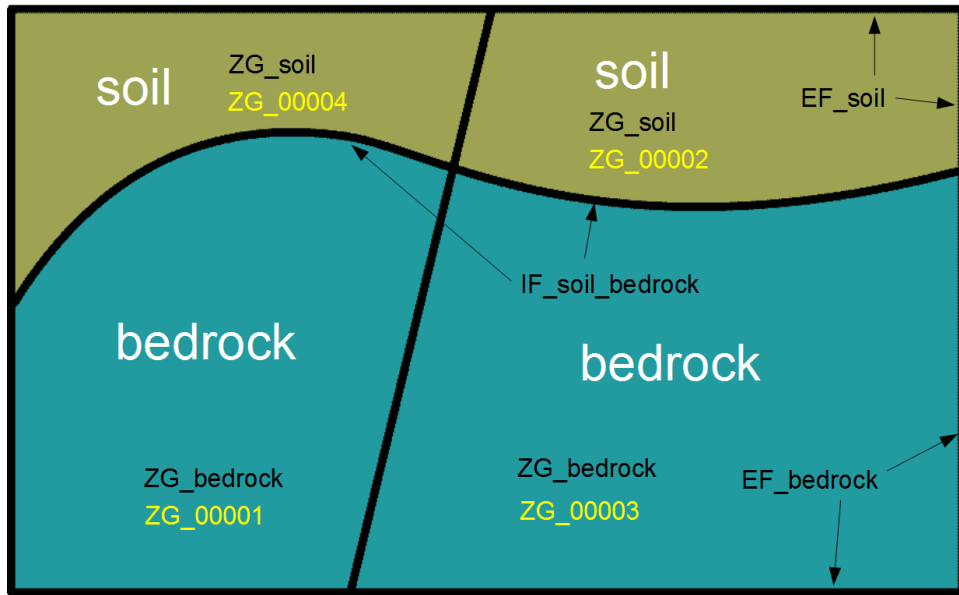


Figure 28: Zone and face groups from a *BlockRanger* generated *FLAC3D* grid.

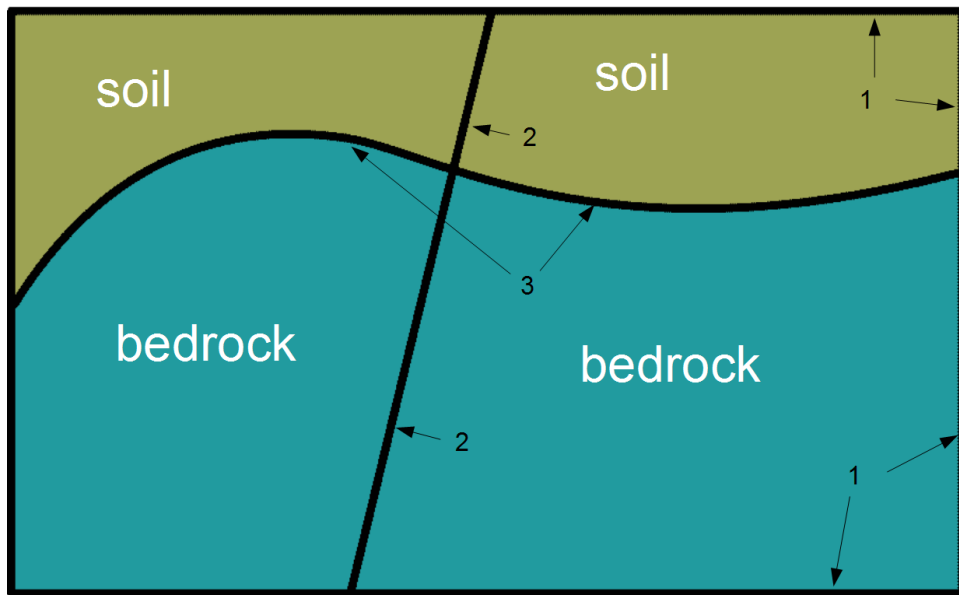


Figure 29: Block groups and joint set ids from a *BlockRanger* generated *3DEC* model.

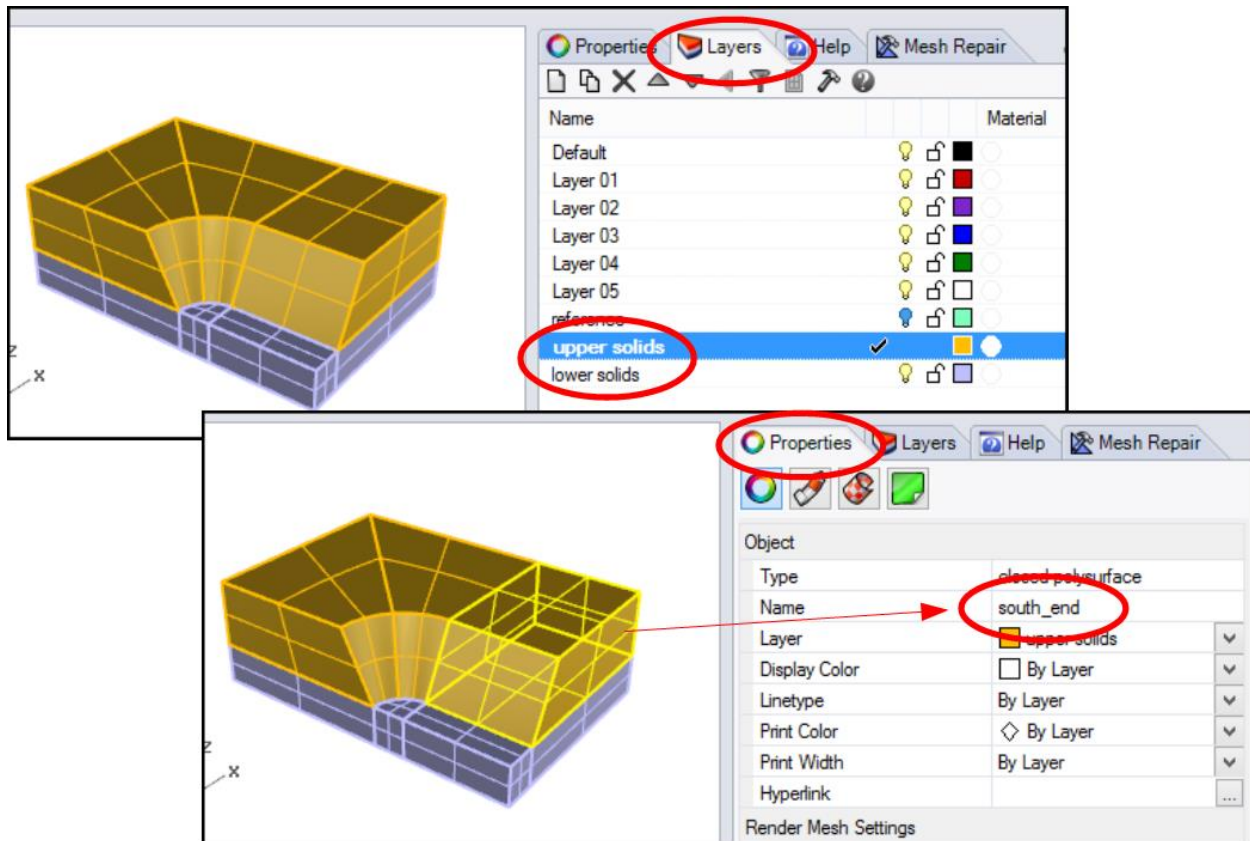


Figure 30: Naming solids by layer (top) and through Properties Name field (bottom).

Note: *Griddle* and *BlockRanger* replace all occurrences of whitespace characters (blanks, tabs, etc.) in name fields with an underscore character “_” to avoid string interpretation problems in *FLAC3D* and *3DEC*. Names are case insensitive: “Upper_Weathered_Rock” is treated as the same name as “upper_weathered_rock”. *Griddle* and *BlockRanger* use the capitalization of the name’s first occurrence (arbitrary what is first occurrence) for the name used in *Griddle* and *BlockRanger* output.

Tutorial 1: A Single Cylinder (*Griddle*)

In this tutorial, you will become familiar with the basics of grid generation for *FLAC3D* and *3DEC* using *Griddle*.

Creating a single-material model

1. Start *Rhino* and when the Template dialog box opens, select **Small Objects-Meters**. If the template dialog does not open, select **File|New** from the *Rhino* menu.
2. Type **_SetWorkingDirectory** on the command line to point *Rhino* to where output files should be written.
1. Click on the label of the window pane Perspective (upper left of the pane) to make it active. Select the **Solid|Cylinder** menu item. Enter **0** followed by **<ENTER>** to center of the *base* of the cylinder at the origin. Enter **2** followed by **<ENTER>** to set the Radius of the base to **2**. Enter **10** followed by **<ENTER>** to set the Height of the cylinder to **10** and complete the construction of a vertical cylinder (Figure 31).

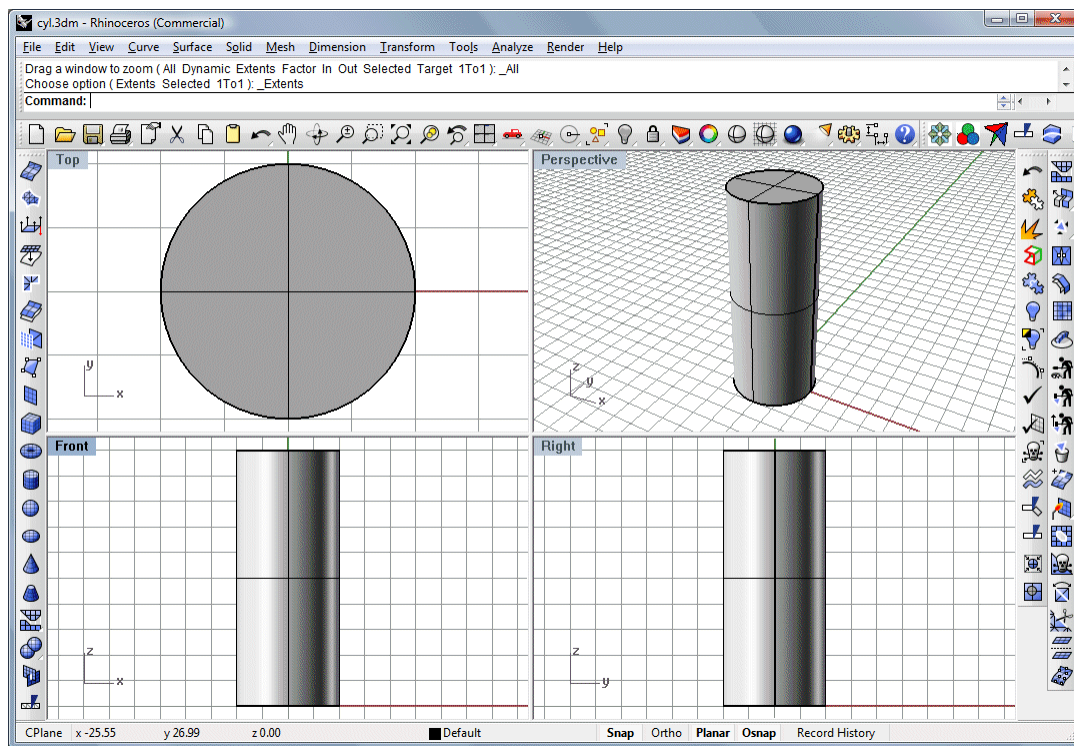


Figure 31: Solid representing a cylinder

You have created a cylindrical solid. A solid is essentially a closed surface. It has a clear interior and exterior. You are now going to create a triangular mesh representing the surface of the cylinder. Creating a triangular surface mesh is a necessary step to creating a volume mesh of the cylinder.

3. Select the **File|Save As** menu item and save your model as **cyl.3dm** to save your work up to this point.

4. Select the cylinder and type **_Mesh** on the Rhino command line. The **Polygon mesh detailed options** dialog box opens. If you see a button in the lower-right corner of the box marked **Simple control**, click it to see a simplified version of this dialog box.
5. In the simplified dialog box, move the slider all the way to the right towards **More Polygons** and click on **Preview** to see a preview of the resulting surface mesh. Click on **OK** to create a surface mesh (Figure 32).

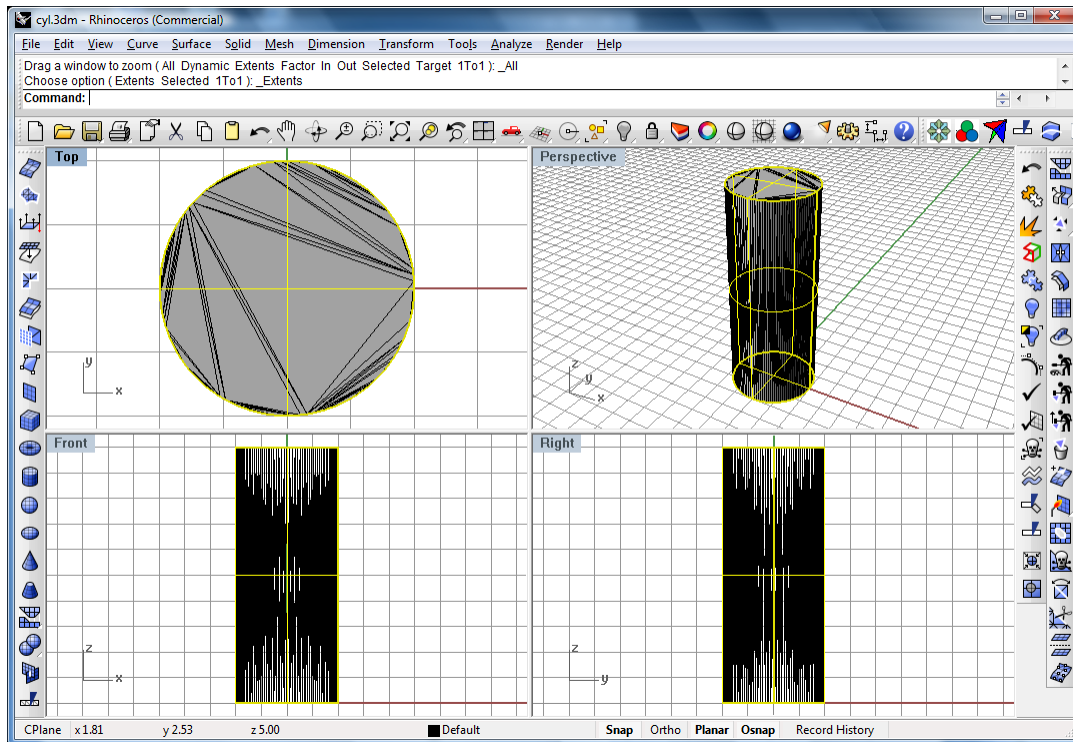


Figure 32: Original highlighted cylindrical surface and the newly-created surface mesh superimposed on it.

6. While the original cylindrical surface is still highlighted (seen in light yellow in Figure 32), **_Hide** it to make only the surface mesh visible. The triangular surface mesh on your *Rhino* screen serves as input to the *Griddle* tools.
7. Select the surface mesh and type **_GSurf**. Select **Mode:QuadDom** and set **MinEdgeLength** and **MaxEdgeLength** to **0.5**. Then press **Enter**. You should see a mesh on your screen similar to that shown in Figure 33.
8. Select the mesh generated in the above step and type **_GVol**. Select **Mode:ConHexDom** and **FLAC3D** output. Press **Enter**. **_GVol** will generate a conformal hex dominant and will output, in your current working directory, an ASCII file named **GVol.f3grid**. This file can be imported into *FLAC3D* using *FLAC3D*'s **File|Grid|Import** (Figure 34).

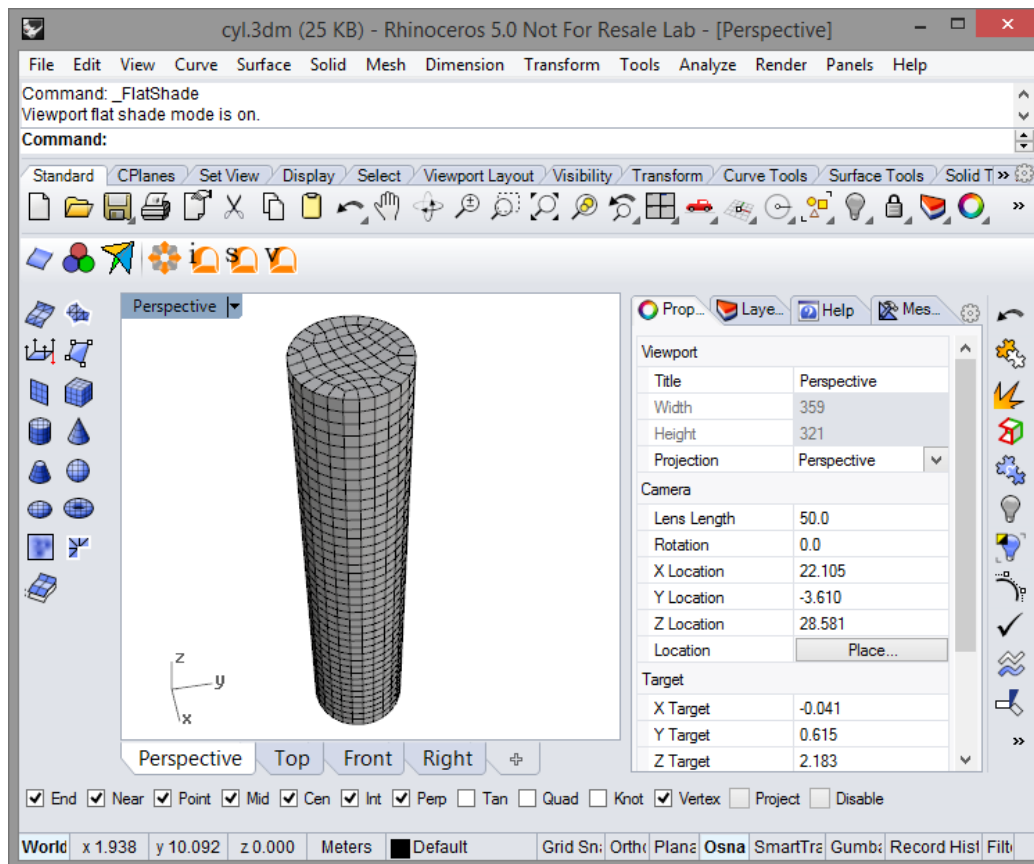


Figure 33: Quad-dominant surface mesh generated by _GSurf.

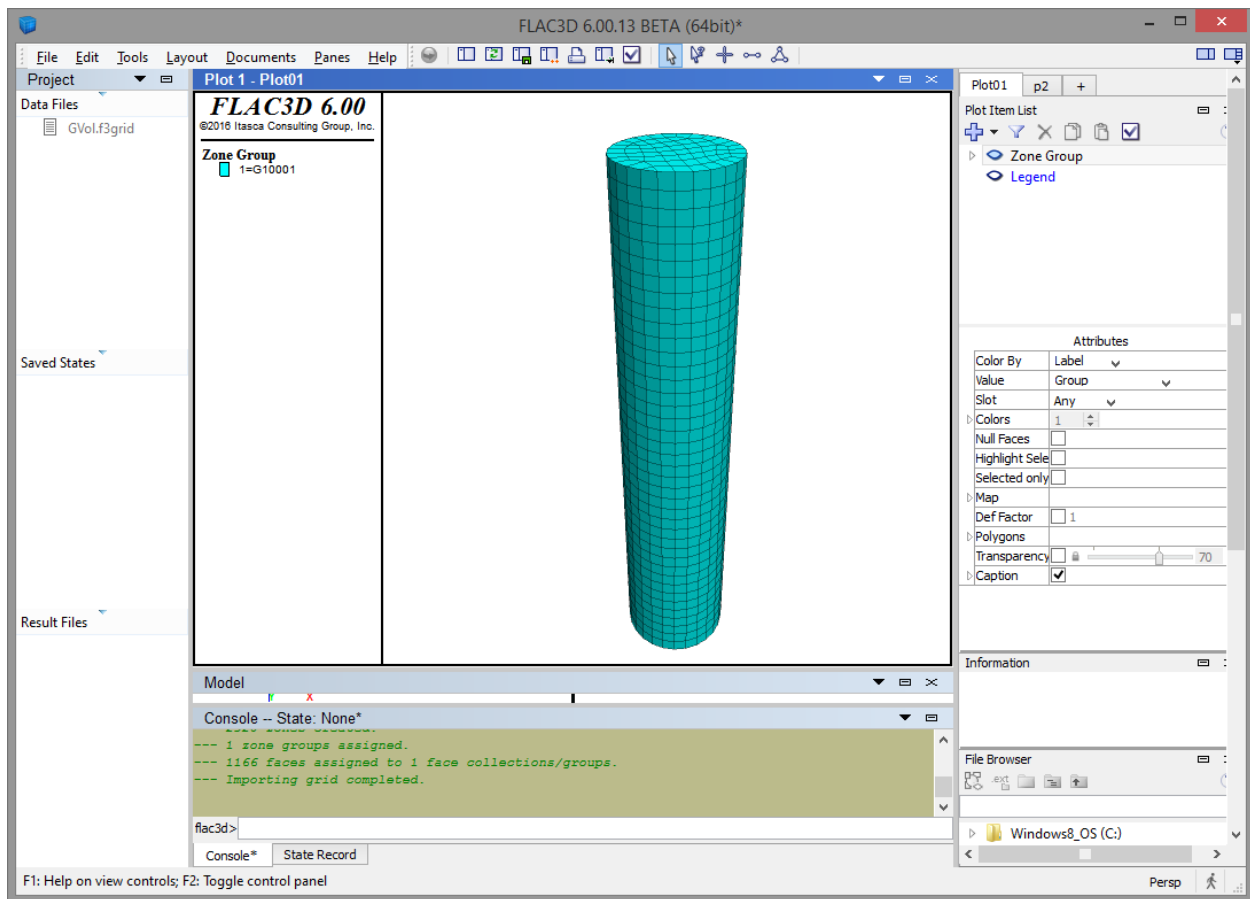


Figure 34: Volume mesh of cylinder read into *FLAC3D*.

END OF TUTORIAL 1

Tutorial 2: Vertical Shaft in a Stratified Soil (*Griddle*)

In this tutorial, you will create a vertical shaft (140 ft deep, 20 ft diameter) inside a cubic block of soil (200 ft × 200 ft × 200 ft) composed of two materials. The surface separating the two soil layers is located at a height of 50 ft (Figure 35).

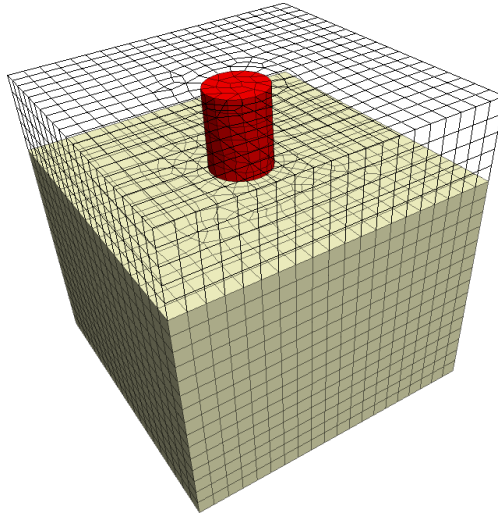


Figure 35: A FLAC3D model of a vertical shaft in a stratified soil

Startup and excavation of the shaft

1. Start *Rhino*, **_SetWorkingDirectory** and select Large Objects, Feet (similar method as step 1 in Tutorial 1).
2. Select **Solid|Box|Diagonal** to define a Box by 2 points. Enter **-100,-100,-100** for the coordinates of the first point, followed by **<ENTER>**. Enter **100,100,100** for the coordinates of the second point followed by **<ENTER>**.
3. Right-click on the **Zoom Extents all viewports** button to make the box fit inside each window. This completes the creation of a Box (Figure 36).

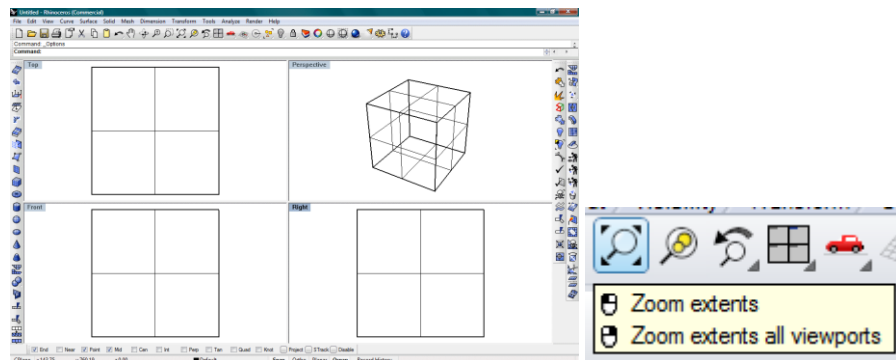


Figure 36: Four views of a box.

4. Double-click the **Perspective** viewport title to maximize it (Figure 37).

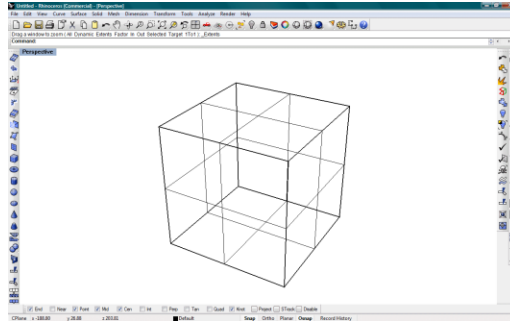


Figure 37: Wireframe view of the box in maximized Perspective viewport.

5. Double-click the **Perspective** viewport title to return to a 4-view window and click on the title of the **Top** viewport to activate it.
6. Select **Solid|Cylinder**. Enter **0,0,0** followed by **<ENTER>** to specify the coordinates of the center of the cylinder base (in the x, y, z coordinate system). Enter **20** followed by **<ENTER>** for the radius. Enter **200** followed by **<ENTER>** to specify the center of the top of the cylinder.

Please note that *Rhino* accepts both **0,0,200** and **200** as the 3rd parameter of the **_Cylinder** Command. Since we are in a **Top** view, **Rhino** correctly assumes that **200** means **0,0,200**.

2. Left-click on the button marked **Shaded Viewport** to see the shaded view of box and cylinder (Figure 38).

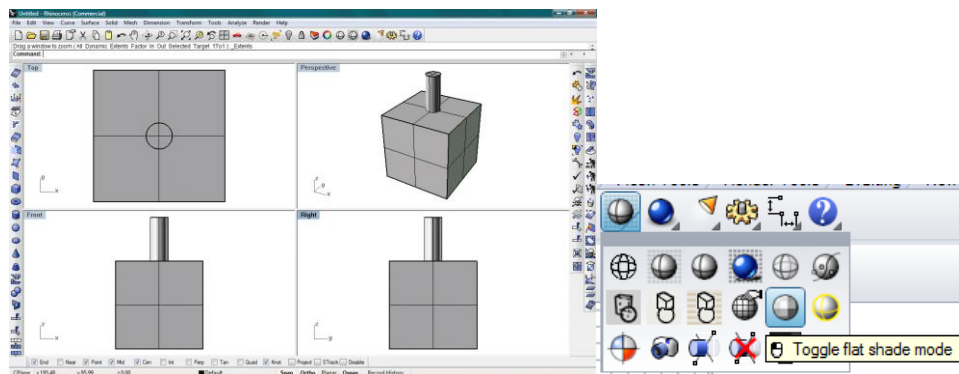


Figure 38: Box and cylinder shaded views.

3. To move the cylinder down by 40 feet, select the cylinder and select the menu item **Transform|Move**. Enter **0,0,0** followed by **<ENTER>**. Enter **0,0,-40** followed by **<ENTER>** (Figure 39).

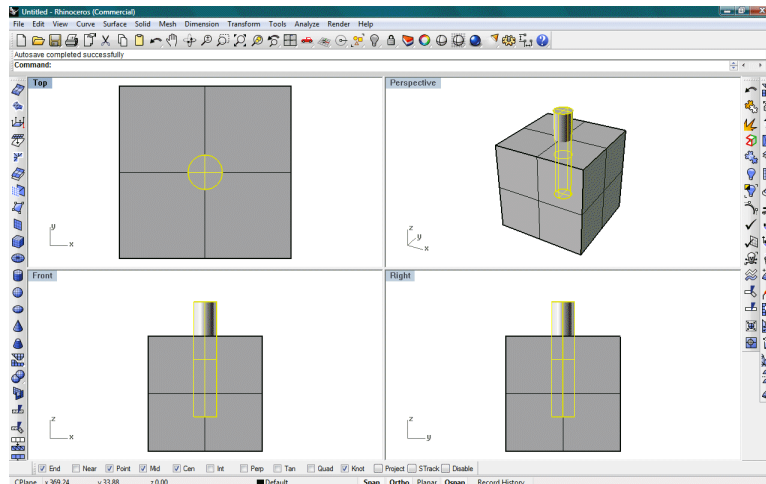


Figure 39: Cylinder (highlighted) partially inside the box.

4. To excavate the cylindrical opening, we must subtract (in the Boolean algebra sense) the cylinder from the box. To do so, select the menu item **Solid | Difference**. First select the box followed by **<ENTER>**, then select the cylinder followed by **<ENTER>** (Figure 40).

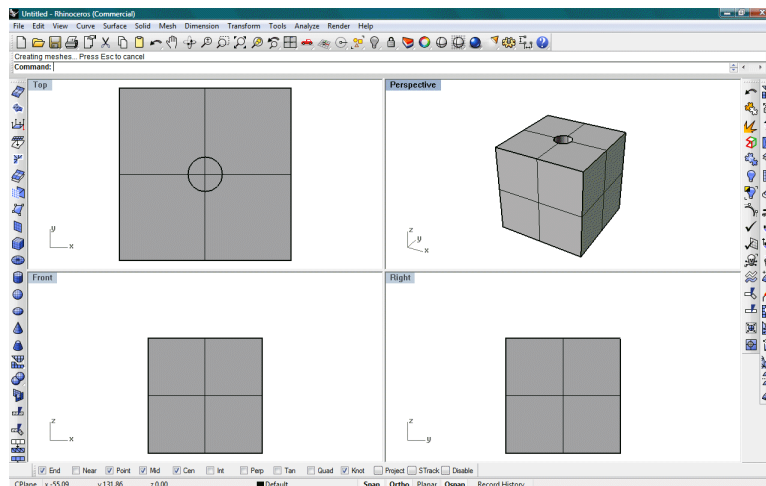


Figure 40: The Boolean subtraction of the cylinder from the box represents the excavation.

Creating a hex-dominant volume mesh of the model

Griddle mesh generation starts with creating a closed surface representing the surface of the object in which we want to create the model.

1. Prior to creating a surface mesh based on a solid model, you should save the *Rhino* model. Select **File | Save As** and when the **Save** dialog box opens, enter **t2_0.3dm** for the file name.
2. Select the model and type **_Mesh** on the *Rhino* command line. The **Polygon mesh detailed options** dialog box opens. If you see a button in the lower-right corner of the box marked **Simple control**, click it to see a simplified version of this dialog box.

5. In the simplified dialog box, move the slider control to the **middle** and click on **Preview** to see what the resulting surface mesh will look like (Figure 41).

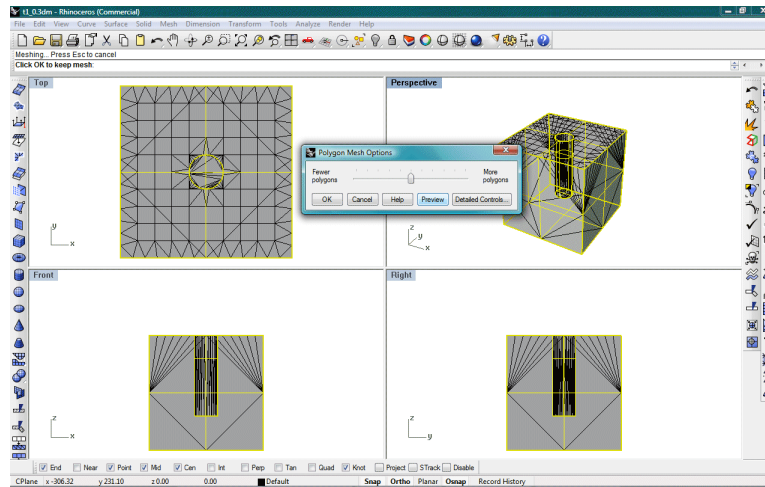


Figure 41: Preview of the surface mesh

6. Click **OK** to accept the surface mesh. Note that the original solid is still highlighted (in yellow) while the mesh is drawn in black.
7. Type **_Hide** to hide the (highlighted) solid. What is left is the surface mesh.
8. Select the mesh and type **_MeshRepair**. Click on the Check Mesh button at the bottom of the Mesh Repair tab. Rhino responds with a message box providing global information about the mesh indicating that, among other qualities, the mesh contains no naked edges. Naked or Free edges are edges attached to only one polygon. Their presence indicates that the mesh is not closed.
9. Select the mesh and type **_GSurf**. Select **Mode:QuadDom**, **MinEdgeLength 5**, and **MaxEdgeLength 10**. Press **Enter**.
10. Select the quad-dominant mesh you just created and type **_GVol**. Select **Mode:ConHexDom** and **FLAC3D** output. Press **Enter**.
11. In *FLAC3D*, import GVol.f3grid using File|Import Grid. Your grid should be similar to that shown in Figure 42.

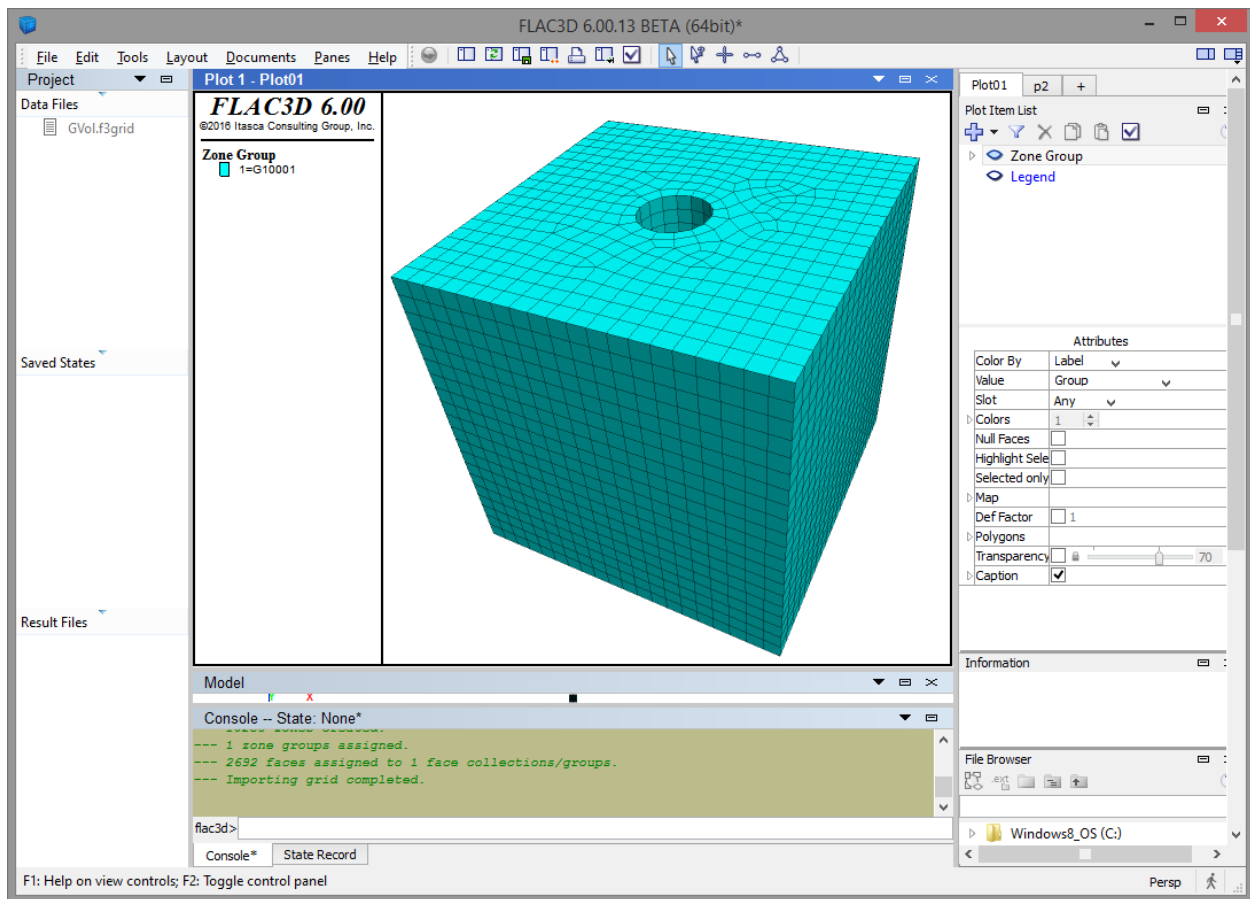


Figure 42: *FLAC3D* grid of a cylindrical shaft in a box.

Creating a model of both the inside and outside of the excavated shaft

1. In Rhino, select the quad-dominant mesh you just created and delete it (with the delete key). Type Show on the command line to show your solid model. Double-click the title of the Perspective viewport to maximize it.
12. Select the menu item **Surface | Planar Curves** and click on the circle representing the rim of the shaft opening (Figure 43). Press <ENTER> to complete the command and create a planar surface on the top of the shaft.

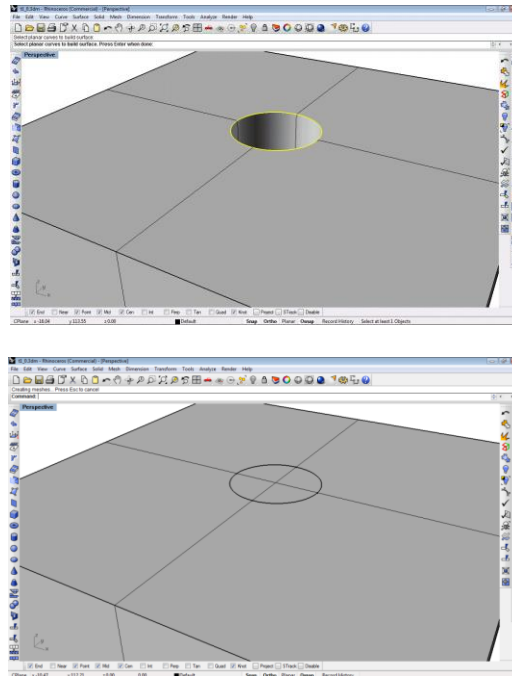


Figure 43: Highlighted curve representing the shaft rim (top) and the capped shaft (bottom)

So far you have been dealing with solids. Solids are closed surfaces that have an unambiguously defined interior and exterior. The surfaces defining such solids are called manifold surfaces. In contrast, consider a solid with a plane cutting through it. Consider the outer surface of the solid and the wall separating the two halves of the solid. This combination of surfaces is closed but doesn't have a clear interior. To be exact, it has two interiors. Such surfaces are called non-manifold surfaces.

So far our model was a perfect solid. Its surface was a manifold surface. The addition of the cap to the shaft creates two separate "interior" regions: inside the well, and outside the well but inside the box. The set of surfaces representing this object constitute non-manifold surfaces. Joining surfaces into a non-manifold surface in Rhino requires a special command. This command joins several manifold or non-manifold surfaces into one non-manifold surface. You can join multiple surfaces into a single non-manifold surface with the **_NonManifoldMerge** icon or command.

1. Select all the surfaces you want to join and click on the **_NonManifoldMerge** icon. The resulting polysurface is a non-manifold polysurface. This can be verified by highlighting the model and pressing **<F3>** to display the Properties of the selected surface and clicking on **Details**.
13. Prior to creating a surface mesh based on a solid model, you should save the Rhino model. Select **File|Save As** and when the **Save** dialog box opens, enter **t2_1.3dm** for the file name.
14. Select the model and type **_Mesh** on the Rhino command line. If the **Polygon Mesh Detailed Option** dialog box opens, click on the **Simple controls** button to bring up the simpler Polygon Mesh Options dialog box.
15. Move the slider to the middle of the scale and click OK to create the surface mesh (Figure 44).

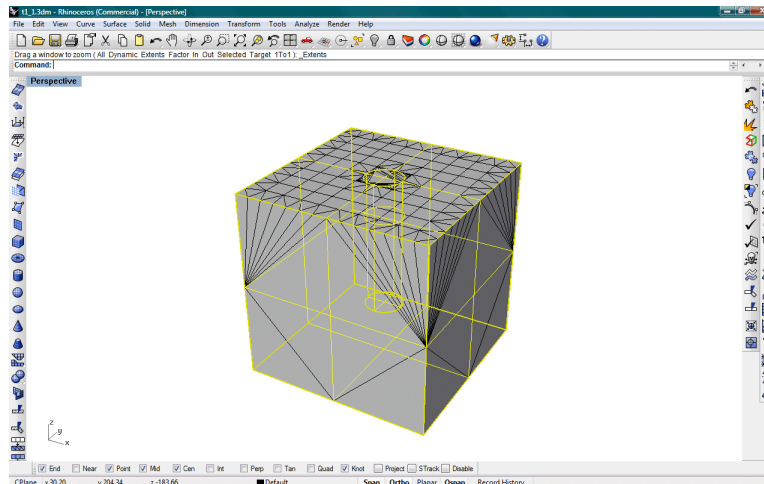
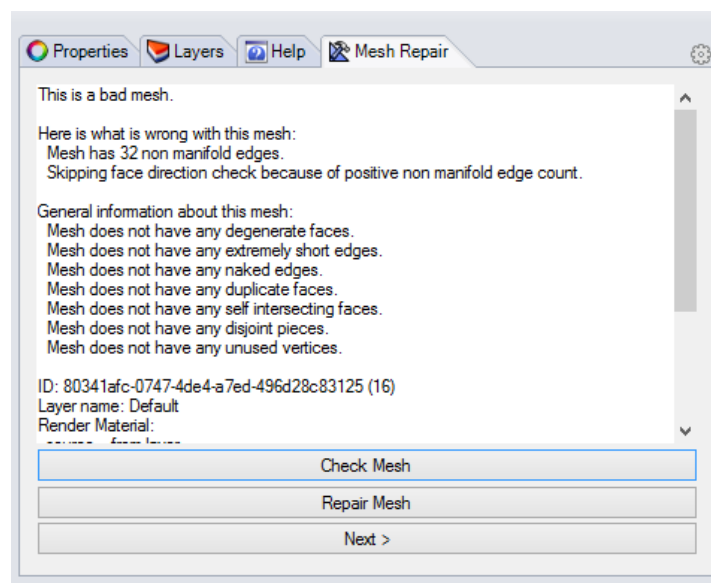


Figure 44: Mesh of the capped model showing the highlighted solid model

16. While the solid model is still highlighted type **_Hide** to just show the mesh.
17. As before, check the mesh with **_MeshRepair**. Rhino responds with the following message box:



Rhino qualifies this mesh as bad because it contains non-manifold triangles (some edges are shared by 3 or more triangles or quadrilaterals), but this was intended because the solid model itself was non-manifold so as to allow the representation of both the inside and outside volumes.

18. Select the mesh and type **_GSurf**. Select **Mode:QuadDom**, **MinEdgeLength 5**, and **MaxEdgeLength 10**. Press **Enter**.
19. Select the quad-dominant mesh you just created and type **_GVol**. Select **Mode:ConHexDom** and **FLAC3D** output. Press **Enter**.
20. In FLAC3D, import GVOL.f3grid using File|Import Grid.
21. The resulting grid is composed of two groups. All the zones located outside the shaft belong to one group and zones inside the shaft to another group (Figure 45).

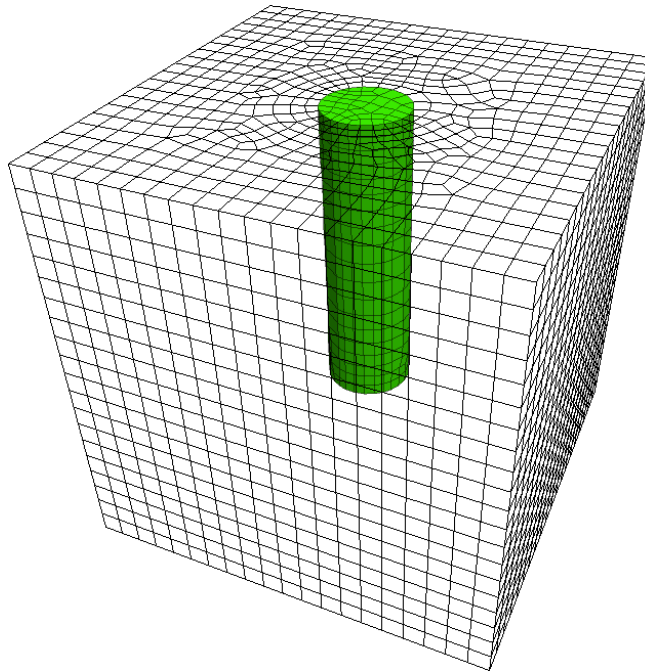


Figure 45: *FLAC3D* grid of the capped model. This grid contains two groups of zones, one group representing inside the cylinder and the other the outside (shown as transparent).

Creating an all-hexahedral mesh of the capped model, with stratified soil

1. In your current folder, double-click the file **t2_1.3dm** you saved earlier. This file contains the Rhino solid model prior to surface meshing. Delete the mesh if one is present and type **_Show** if you hid the polysurface. If the Perspective viewport is maximized, double-click the title of the **Perspective** view port to return to a 4-Viewport view.
22. You are now going to represent the stratification by a cut at height **z=50**. To do so, double-click the **Front** viewport title to maximize the Front view, and select the **Curve|Line|Line Segments** menu item.
23. Enter **-150,50** followed by **<ENTER>** to set the first point, then **150,50** followed by **<ENTER>** to set the second one (Figure 46).

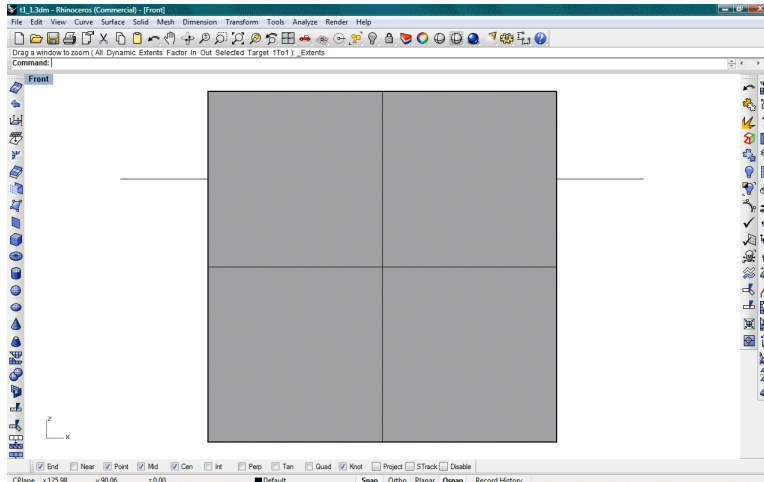


Figure 46: View of the model with horizontal line segment at z=50.

24. Select the menu item **Edit|Split**. Click on the box and hit **<ENTER>**. Click on the horizontal line and hit **<ENTER>**. Delete the horizontal line and note that the model is now split into 2 parts.

Please note that the Split operation splits the surface of the solid into 2 surfaces (that are not closed). To split a solid into two solids you must use Wire cut. This function will be discussed in a later section.

25. Select the top part and **_Hide it**. Double-click the title of the viewport to return to a 4-viewport view and double-click the Perspective view port to maximize it (Figure 47).

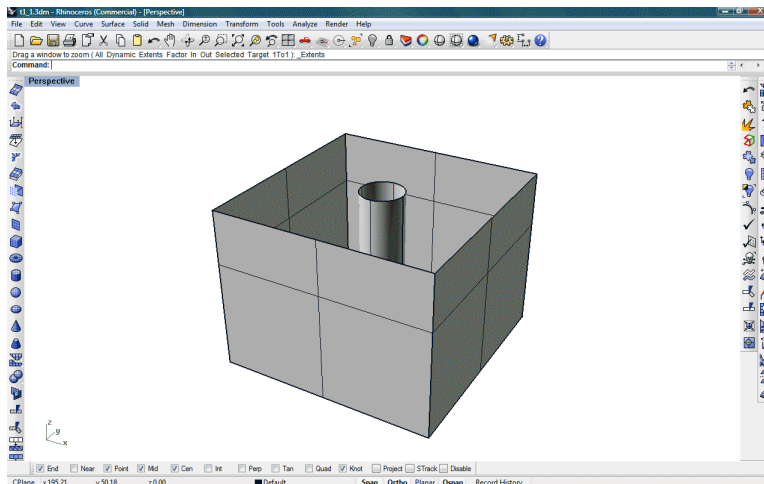


Figure 47: Perspective view of the lower part of the split model

26. Select the **Curve|Curve From Object|Duplicate Border** menu item. Click on the box and then on the cylindrical part of the model followed by **<ENTER>**. This operation extracted a square and a circular curve from the model. Hide the box and the cylinder to only see the two extracted curves shown in Figure 48.

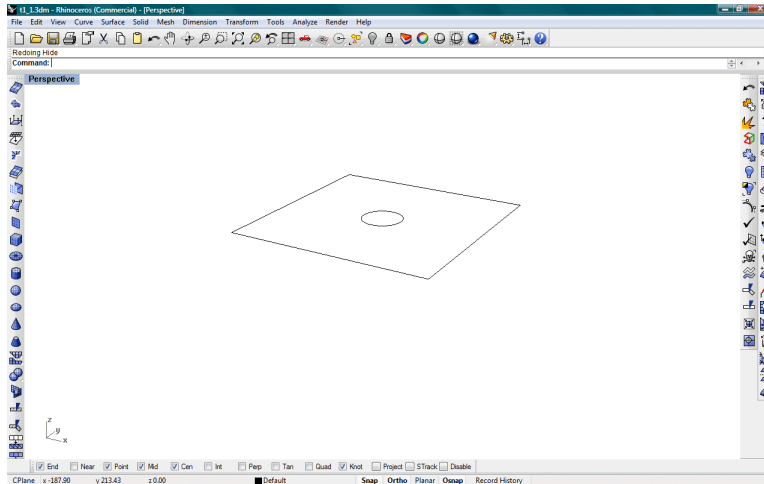


Figure 48: The highlighted square and circular curves represent the Naked Edges of the selected Polysurfaces.

You are now going to create 2 horizontal walls based on these curves: one inside the cylinder and the other outside. These walls will act as a partition between the top and bottom of the model.

27. Select the **Surface | Planar Curves** menu item and click on the square curve to fill its interior with a square surface. Double-click the Title of the **Perspective** Viewport to return to 4-viewport. Double-click the title of the **Top** view port to maximize it (Figure 49).

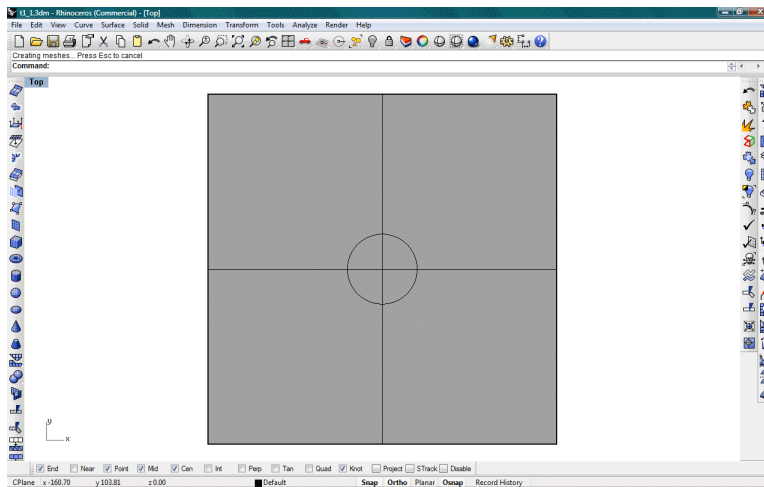


Figure 49: Top view of the square surface patch

28. Select the menu item **Edit | Split** and click on the surface of square, then hit **<ENTER>**. Click on the curve representing the circle and press **<ENTER>** to split the square into two parts. Figure 50 shows the two surfaces with the outer surface highlighted.

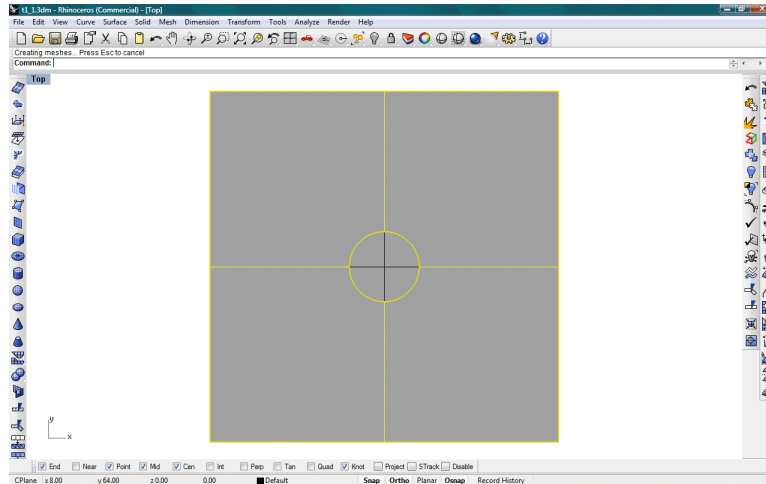


Figure 50: The two surfaces resulting from the split with the outer surface highlighted

29. While in the **Top** view, select the circular and square curves and delete them.
30. Right-click on the button marked **Show Objects** to unhide all the surfaces, select a **Perspective** view of the model and resize it by left-clicking the **Zoom Extents** button.
31. Select the **Edit|Select Objects|All Object** menu item and note that Rhino responds with the message “3 polysurfaces, 2 surfaces added to selection.” on the command-line.

You must now join all the surfaces and polysurfaces into one non-manifold surface.

32. Select all the surfaces and click on the **_NonManifoldMerge** icon to create one single non-manifold polysurface.
33. Prior to creating a surface mesh based on the solid model, you should save your *Rhino* model. Select **File|Save As** and when the **Save** dialog box opens, enter **t2_2.3dm**.
34. Select the model and type **_Mesh**. If the **Polygon Mesh Detailed Option** dialog box comes up, click on the **Detailed Controls** button to bring up the simpler Polygon Mesh Options dialog box.
35. Move the slider to the **middle of the scale** and click **OK** to create the surface mesh. Delete the solid model which has remained highlighted (Figure 51).
36. Remesh the surface and create a **FLAC3D** volume mesh as before (similar to the steps done for the model in Figure 45).

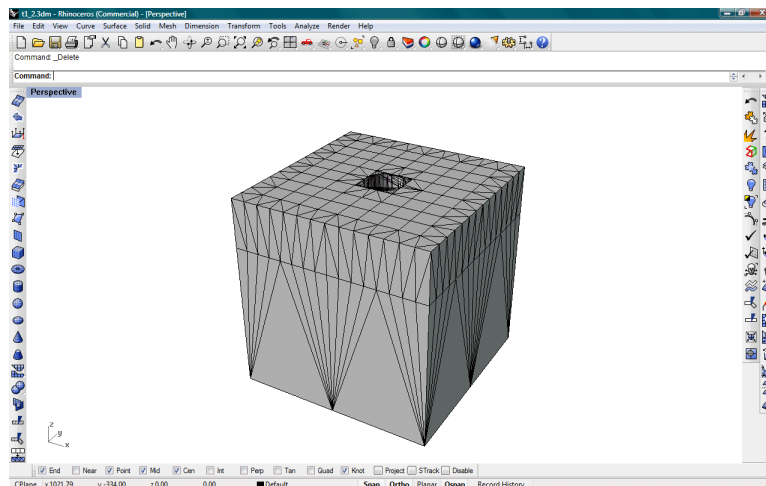


Figure 51: Surface mesh of capped shaft in stratified soil

The resulting grid is composed of 4 groups. Two groups represent the shaft interior and two different groups represent the soil layers, exterior to the shaft (Figure 52).

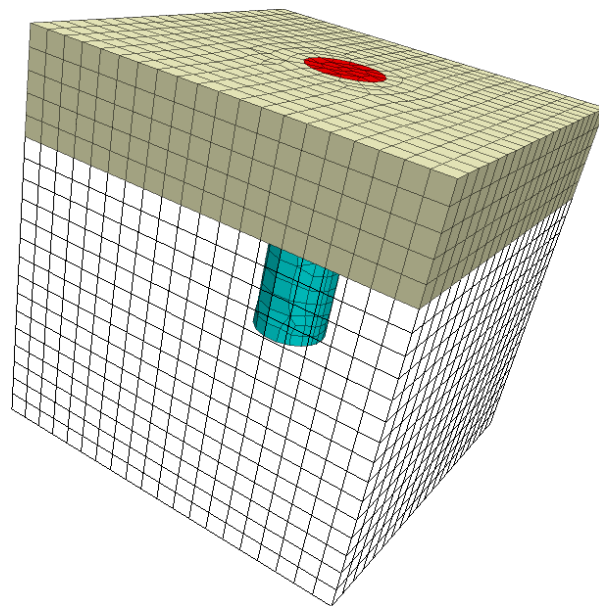


Figure 52: *FLAC3D* model of the capped shaft in a stratified soil with bottom soil group transparent.

END OF TUTORIAL 2

Tutorial 3: 3D Slope (*BlockRanger*)

Objective

In this tutorial you will learn, in detail, how to build a block-structured hexahedral mesh from a CAD model described by a DXF file, the reference model. To create a mesh using *BlockRanger* (`_BR` command), you must create an assembly of 6, 5 or 4-sided *Rhino* solids that conform to the reference model. The creation of such an assembly is the objective of this tutorial. The reference model is shown in Figure 53 left. The final hexahedral mesh is depicted to the right. This 3D slope model has a shape similar to a bathtub and hence it is sometimes referred to as the “bathtub” model.

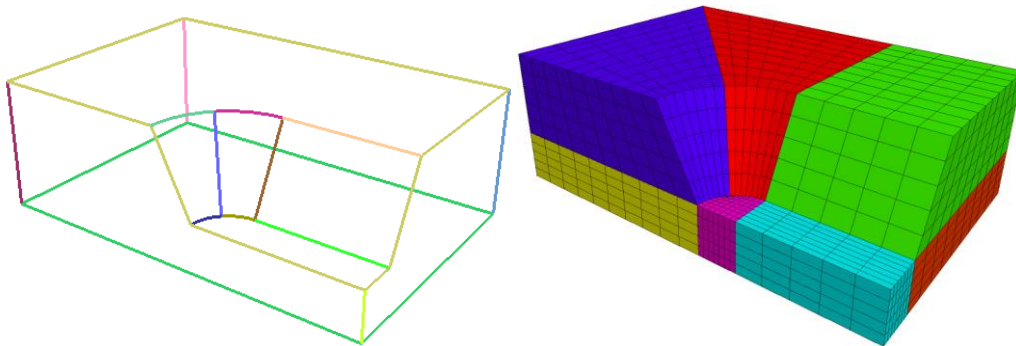


Figure 53: Reference model (left) and *BlockRanger* generated mesh (right)

Preparation

1. Start **Rhino** and select **Large Objects, Meters** as your template. If *Rhino* does not ask you for a template, **File|New** and select **Large Objects, Meters**.
2. In *Rhino*, type `_SetWorkingDirectory` and navigate to your current **working directory**. In this fashion *Rhino* knows where to read, import from and save files.

Importing the geometry

1. **File|Import bt.dxf**, and if you are asked about the model **units**, enter **meters**. Enter **millimeters** for the **layout** units. You can hide the **construction plane grids** in all the viewports by clicking on the **Options** icon to open the **Rhino Options** dialog box. In the **left** pane, in **Document** Properties, select **Grid** and turn Show grid line and Show grid axes **off**.
2. **Edit|Select Objects|All Objects** to select everything, and **Edit|Layers|Change Object Layer**. In the **Layer for objects** dialog box that opens, click on **New**, and in the **New Layer** dialog box enter **reference**. Click **OK**, and **OK** again to finish moving the selected lines into a new layer called **reference**.
3. Click on the **Edit Layers** icon to open the **Layers** pane to the right of your *Rhino* window. In the **Layers** pane, click on the **black square** indicating the current **color** of the **reference** layer and select **Aquamarine** to change the color of this layer to aquamarine. Select Shaded View (see subsection Thicker lines in section Setting up your work environment) and maximize the Perspective Viewport (Figure 54).

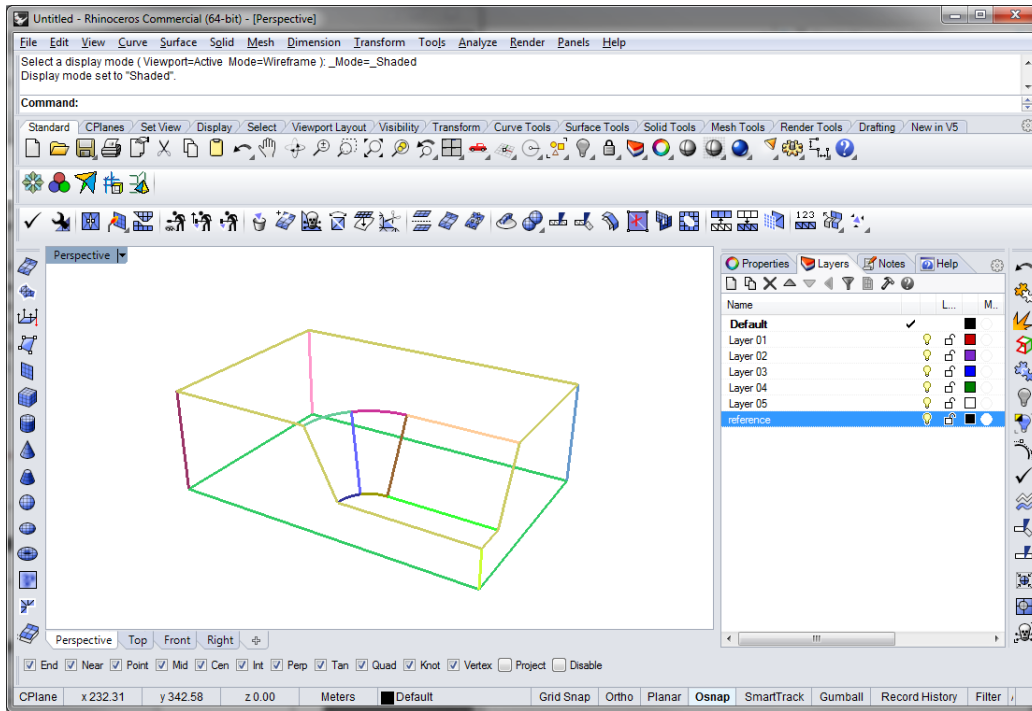


Figure 54: The reference model.

Choosing the right mesh block layout for your model

BlockRanger processes a layout of solids into a layout of mesh blocks. The present model may be decomposed into solid blocks in many ways. The figures below (Figure 55 - Figure 59) show 5 different *Rhino* solid layouts (left) resulting in 5 mesh blocking configurations. In this tutorial you will build the layout depicted in Figure 57 which is composed of **eight** six-sided **solids**.

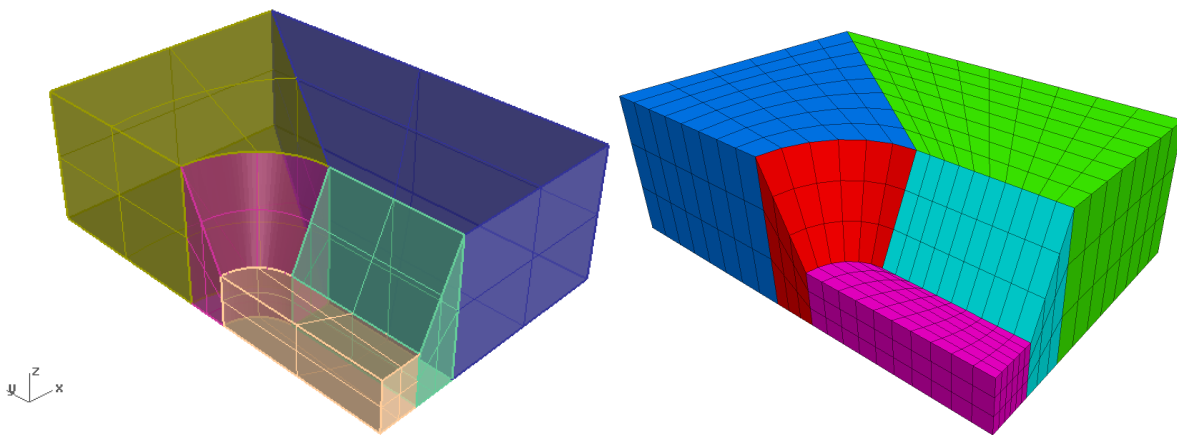


Figure 55: Solid layout resulting in 5 mesh blocks

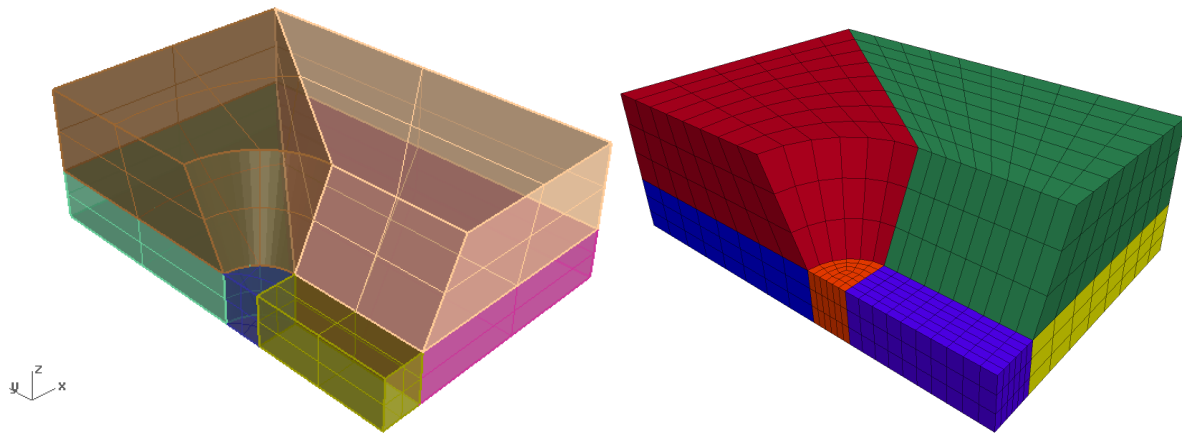


Figure 56: Solid layout resulting in 6 mesh blocks

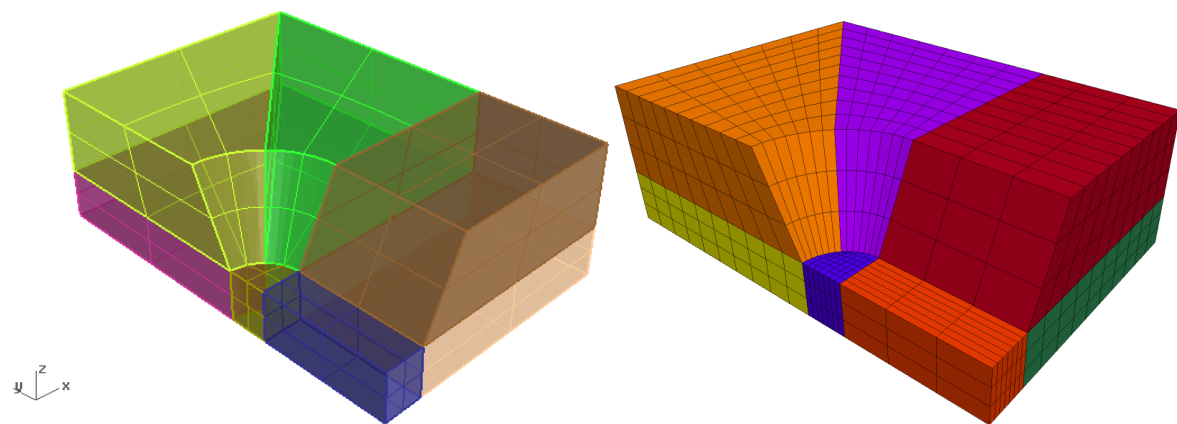


Figure 57: Solid layout resulting in 8 mesh blocks

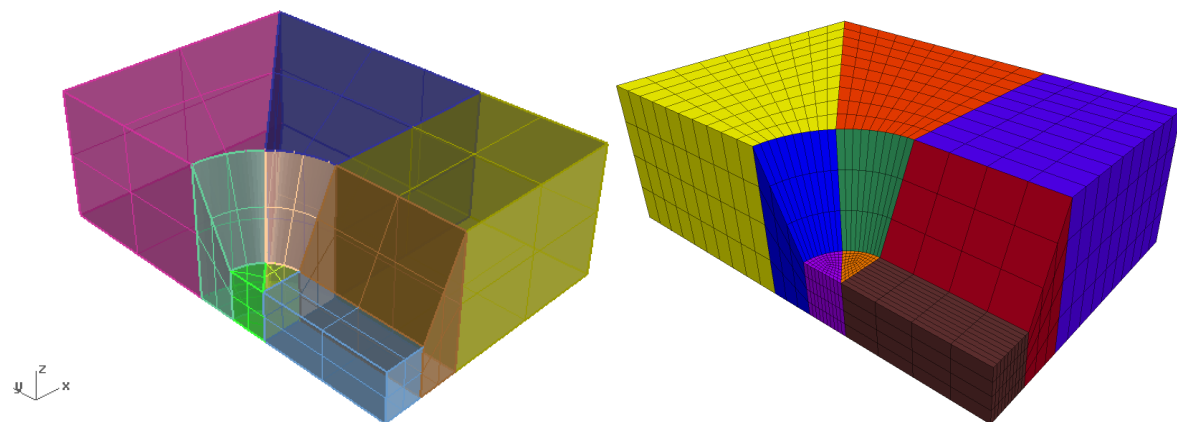


Figure 58: Solid layout resulting in 9 mesh blocks

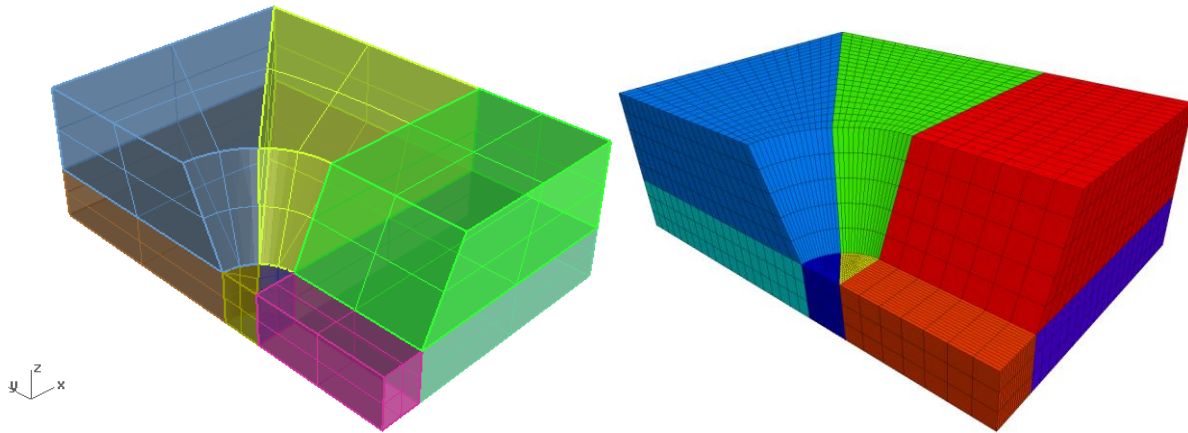


Figure 59: Alternate solid layout resulting in 9 mesh blocks

Building the first solid

Below is a quick summary of the **rules** to follow in building solids for *BlockRanger*.

- Solids must be:
 - 4-sided solids made of 3-sided faces (a topological tetrahedron)
 - 5-sided solids made of two 3-sided faces connected through three 4-sided faces (a topological prism)
 - 6-sided solids made of 4-sided faces (a topological hexahedron)
- Faces must be simple faces that cannot be further exploded into simpler faces
- Face sides (edges) must only be simple curves that cannot be further exploded into simpler curves

The curves in the reference model are all **Polylines**. Some are made of single line segments and the rest are composed of **multiple** line segments and **may not be used directly** to build surfaces. You will need to retrace them with arcs (**Curve|Arc**) or higher order curves (**Curve|Free-Form|Interpolate Points**, for instance) which are simple curves.

1. Click on **Osnap** (Object Snap) at the bottom of the screen (Figure 60) to highlight it. Make sure that **End, Near, Point, Mid, Cen, Int, Perp, Tan, Quad, Knot** and **Vertex** are **checked**, as shown below. **Grid Snap** may be turned off.

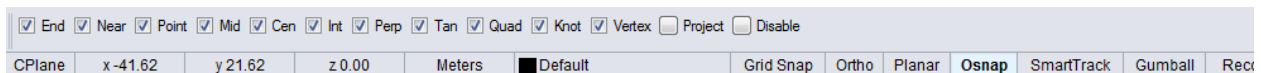


Figure 60: Rhino Osnap bar

The first solid you are going to build is shown in gray in Figure 61. The details follow.

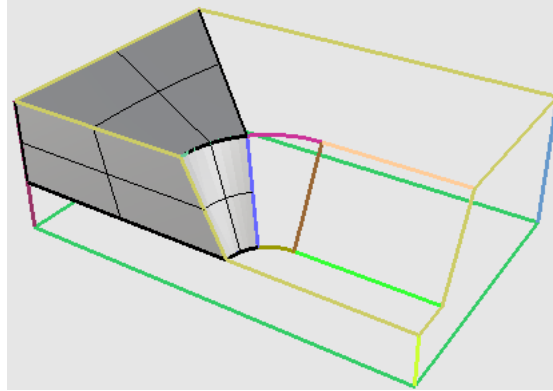


Figure 61: First solid to be built

2. Select **Curve|Arc|Start, Point, End** and **Zoom in** around the curved area and click on the left end of the left upper eight-circle (Figure 62, left) to set the **Start of arc**. Note that a bubble saying **End, Int, Knot** appears momentarily near the corner vertex, before you click on it.
3. **Following** along the same curve, click on a **second** vertex somewhere near the middle of the curve (Figure 62, center). This sets the 2d point of the arc. For the **End** point of the arc, click on the **vertex** at the upper T junction to finish the arc highlighted in Figure 62, right.

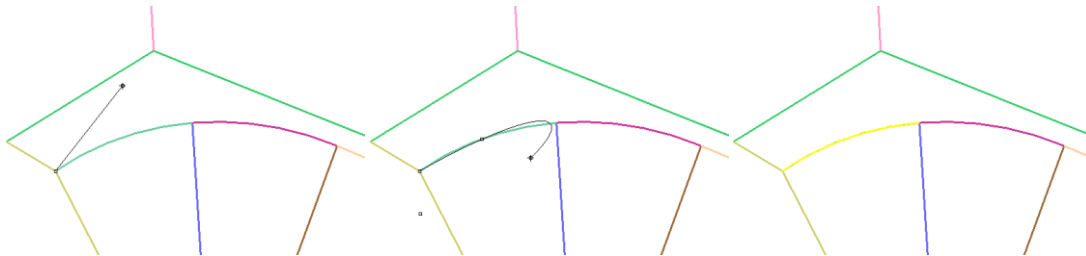


Figure 62: Building the upper curve

For **complex** curves, **InterpCrv (Curve|Free-form|Interpolate Points)** may be used. Essentially, any curve tool in *Rhino* is acceptable as long as it produces a single curve, i.e. when exploded, it results in one curve.

4. Similarly, use **Curve|Arc|Start, Point, End** to retrace the lower eighth-circle that parallels the curve you just built (Figure 63).

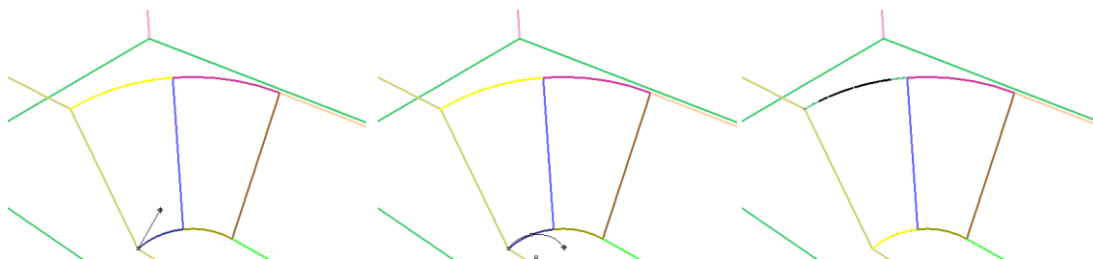


Figure 63: Building the lower curve

5. Select the **two** arcs you have just built (highlighted in Figure 64, left), select **Surface|Loft**, use the **settings** shown Figure 64, right, and click **OK** to build a lofted **surface** between the two curves (Figure 64, right).

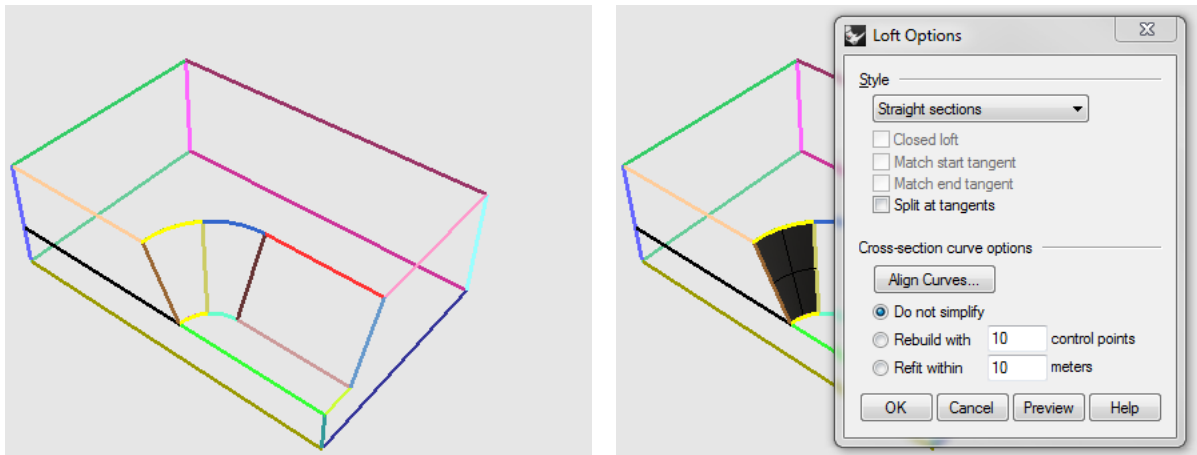


Figure 64: Building the first surface for the first solid

In order to build the opposite surface, you first need to build some construction lines...

6. Select **Curve|Line|Single Line**. Click on the lower-left **corner** of the surface you just built to set the first point of the first construction line (Figure 65, left). Move your cursor **over** the vertical line opposite the first point (Figure 65, left). As you **move** your cursor up and down the vertical line, stop when a bubble with the word **Perp** pops up. This is the location on the vertical line where the line you are building will be **perpendicular** to it. Click when **Perp** pops up to set the second point of the line (Figure 65, right).

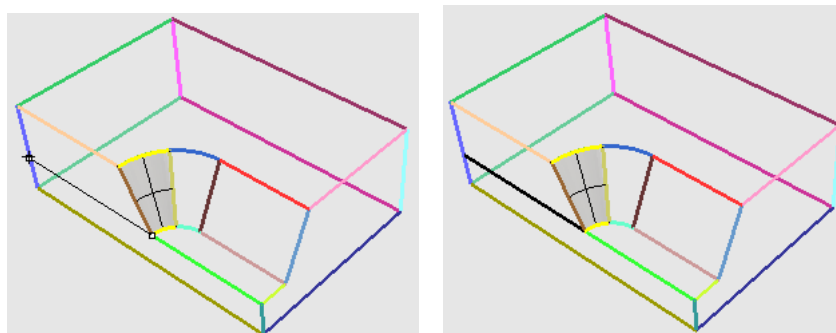


Figure 65: Building the first construction line

7. Select **Curve|Line|Single Line** and click on the end of the last line, then move your cursor over the opposite vertical line and click when the bubble **Perp** appears Figure 66.

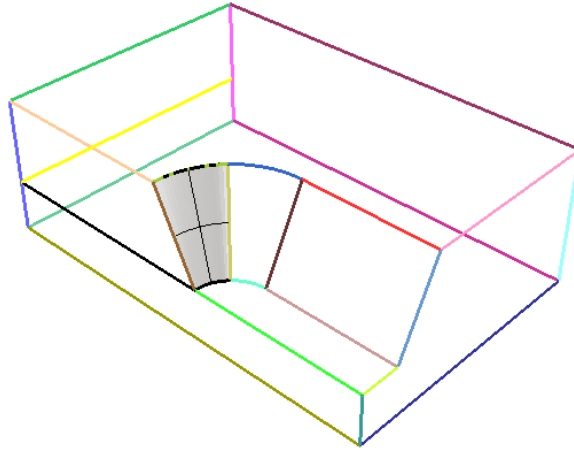


Figure 66: Completing the second construction line

You now have what you need to build the opposite surface.

8. To build the rectangular surface **opposite** the curved face you built earlier, select the two horizontal lines highlighted in Figure 67 and use **Surface|Loft**.

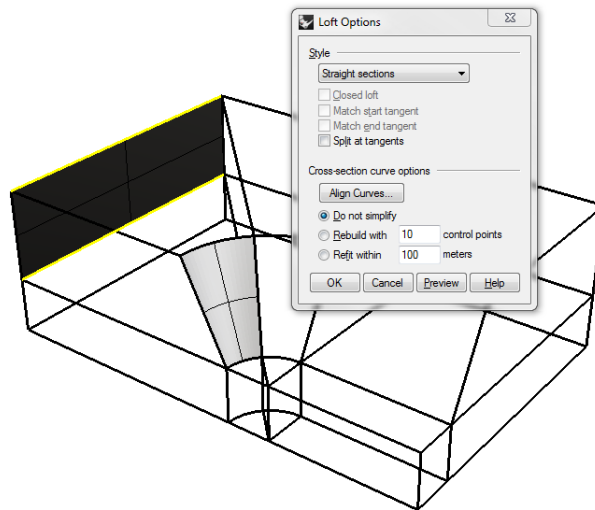


Figure 67: Building the opposite surface

Note that the flat **rectangular** surface could have been created using) **_EdgeSrf**, **_PlanarSrf** or by **Extruding** one of the horizontal lines.

The following sequence of operations builds the solid between the two surfaces and will be used repeatedly throughout this tutorial.

9. Select **both surfaces**, then **Curve|Curve from Objects|Duplicate Border** (Figure 68, left). While the two borders are highlighted, select the **Surface|Loft** menu item.
10. Note that *Rhino* responds with: **Drag seam point**, and the Loft **preview** may indicate that Rhino will connect the **red** vertex to the **green** one (Figure 68, left). If this were the case, the lofted surface

would be **twisted**. To force Rhino to connect the **red** vertex with the **blue** one instead, click on the **green** vertex, then on the **blue** one to move the destination point from green to blue (Figure 68, right).

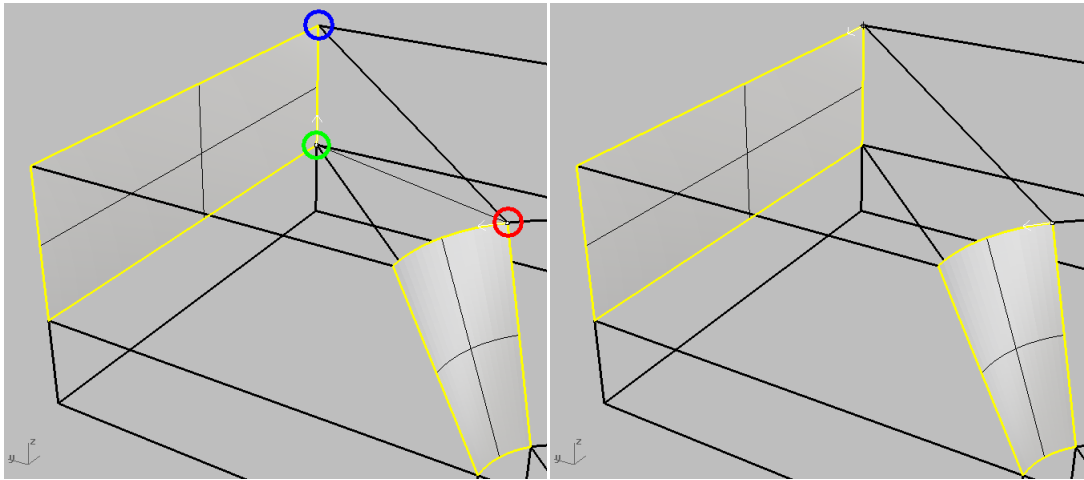


Figure 68: Lofting a surface between the borders of two existing surfaces.

11. Press **<ENTER>** and the **Loft Options** dialog box opens. Make sure that **Straight sections** and **Do not simplify** are selected (Figure 69, left), and click on **OK** to complete the loft operation (Figure 69, right).

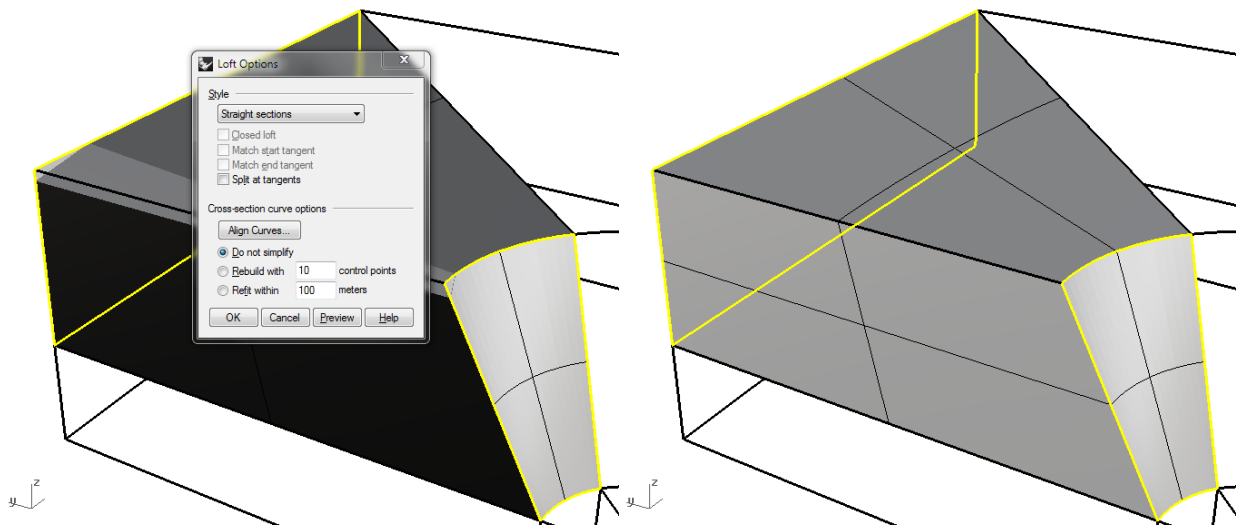


Figure 69: Completing the Loft operation

12. While the two **borders** are still highlighted, **Delete** them. You now have the **two** initial **surfaces** and a **polysurface** representing the lateral walls of the solid you want to build. Select the **two** surfaces and the **polysurface** and **Edit|Join** to complete the creation of your first solid (Figure 70).

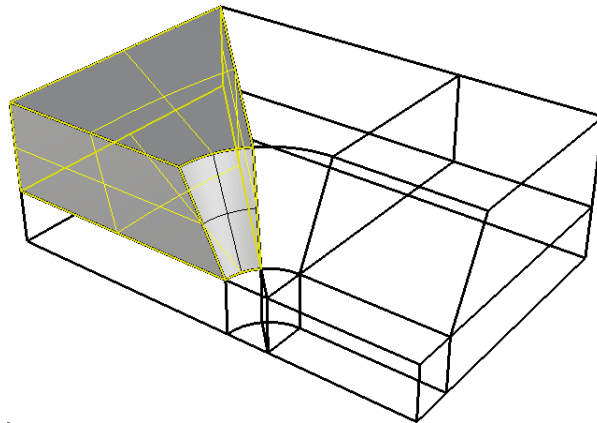


Figure 70: Completing the first solid

The last few steps represent essentially the sequence of operations that you will use to build the remaining 7 six-sided solids.

Building the remaining upper solids

1. Retrace the remaining 2 **eighth circles** (highlighted in Figure 71, left) using **Curve|Arc|Start, Point, End**, then use **Loft** to build the **surface** between the two (Figure 71, left).
2. Build 3 construction lines. The first starting at the far lower back corner of the solid you built, ending on the vertical line across and perpendicular to it (left). The second starting at the lowest, forward right corner of the surface you just built and perpendicular the construction line you built in the previous step (center). The third, starting at the end of the last line and perpendicular to the horizontal line above it (right).

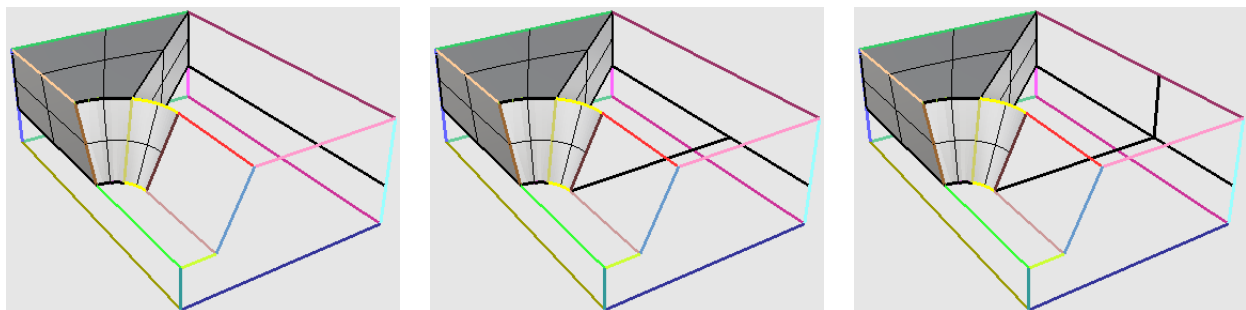


Figure 71: Building a curved surface (left) and additional construction lines

3. Make sure nothing is selected, and select **Surface|Loft**. Click on the **far-right vertical** edge of the Solid and when the **Selection Menu** appears (Figure 72, left), click on any of the **Polysurface edge** items to select the first curve of the Loft operation (right): you have selected one curve of the Loft.

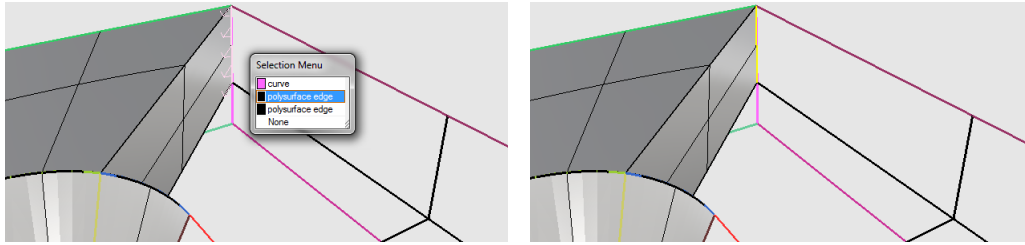


Figure 72: Selecting the correct vertical line

4. Click on the vertical line opposite the highlighted edge to specify the second curve of the Loft. Hit **<ENTER>** to bring up the **Loft settings** dialog box. Click **OK** to complete the Loft (Figure 73).

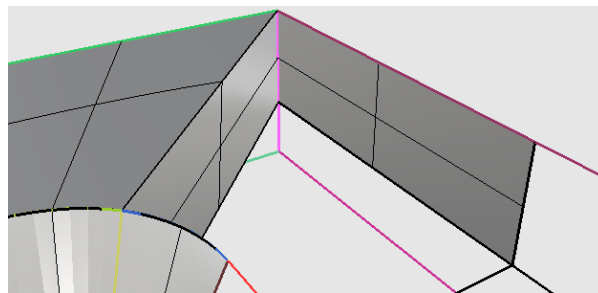


Figure 73: Completing the Loft operation

Building the solid between the two surfaces follows the procedure used earlier.

5. Select the opposing **faces**, followed by **Curve|Curve from Objects|Duplicate Border**, and while the two borders are highlighted, **Surface|Loft**.
6. Again, the guide line in the Loft **preview** indicates that Rhino will connect the **wrong corners** (Figure 74, left). Click on any of the two **white points** at the ends of the guide line connecting the two surfaces, then click on the **appropriate corner** so that the guide line connects the right corners (right).

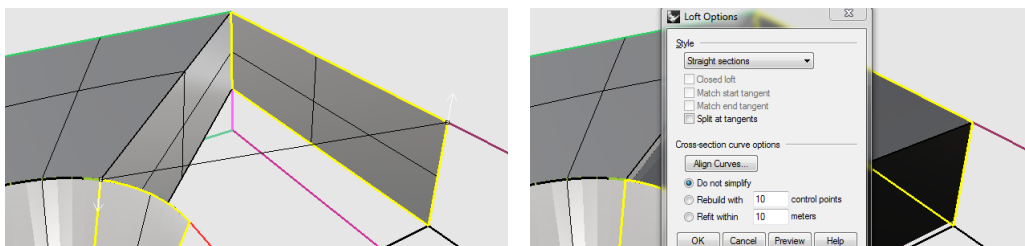


Figure 74: Lofting the two surface borders

7. Click **OK** to complete the loft. **Delete** the **borders** (which have remained highlighted). Select the **two** end faces and the newly created lofted **polysurface**, and **Edit|Join** to complete your second **solid** (Figure 75, left). Complete the **third** solid in the same fashion (Figure 75, right).

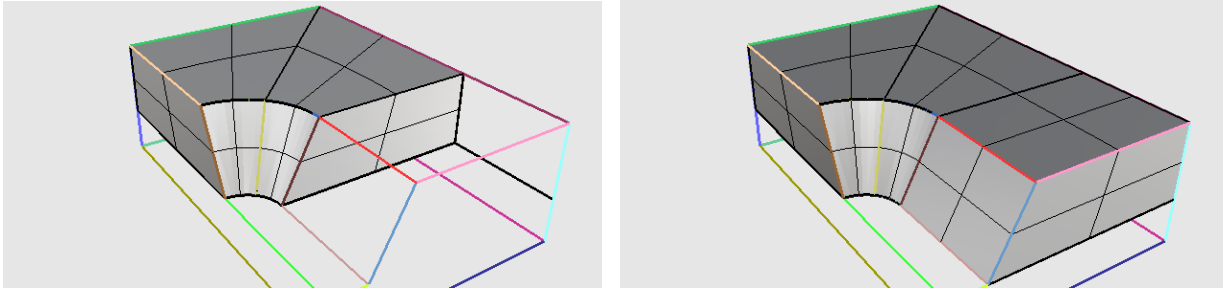


Figure 75: Completing the second (left) and third (right) solids

Building the lower layer solids

Three solids located immediately below the ones you just built may be created at once by extracting the lower surfaces of the existing 3 solids and extruding them downward.

1. **Look below** the model and right-click on the **Extract Surface** icon. On the resulting command line, click on **Copy=No** to change it to **Copy=Yes**. Click on each of the lower surfaces of the 3 solids, then hit **<Enter>** to complete the surface extraction process (Figure 76).

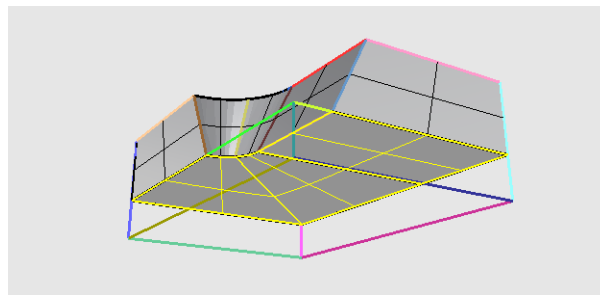


Figure 76: Extracting copies of the 3 lower surfaces of the solids

2. While the surfaces are highlighted, **Solid | Extrude Surface**. **Drag** the cursor down to any of the 4 lower **corners** of the model (Figure 77, left) and click. Note that after the extrusion is complete the surfaces remain highlighted (right). Hit **<Delete>** to delete them.

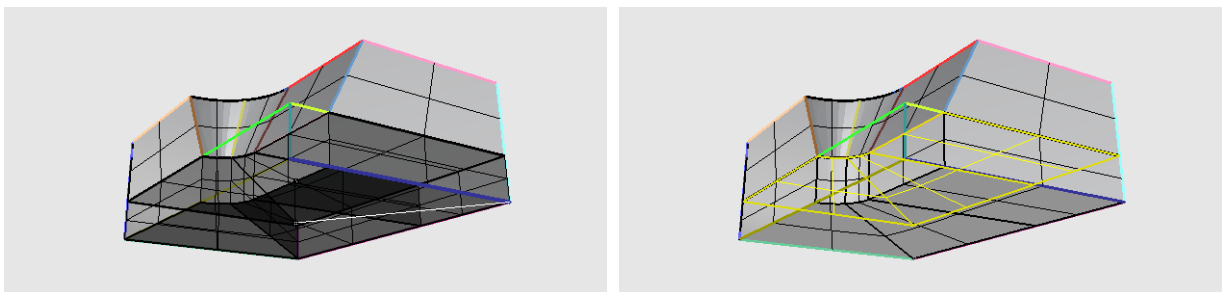


Figure 77: Extruding 3 surfaces at once

Only two more solids remain to be built.

3. Use **Curve|Line|Single Line** to build the two highlighted horizontal line segments shown in Figure 78, left. Use **Curve|Line|Single Line** to build a vertical line segment joining the two T-junctions (Figure 78, right).

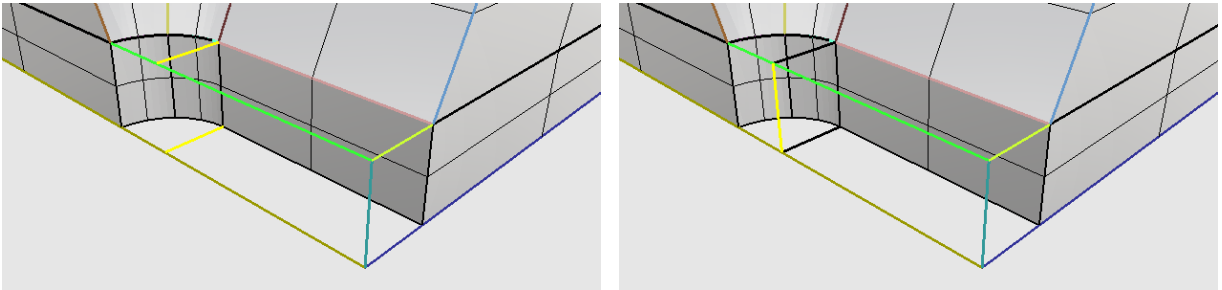


Figure 78: Building additional construction lines

4. Use **Loft** to build the vertical surface between the two highlighted horizontal construction lines shown in Figure 79, left. Use **Loft** to build an additional vertical surface shown in Figure 79, right.

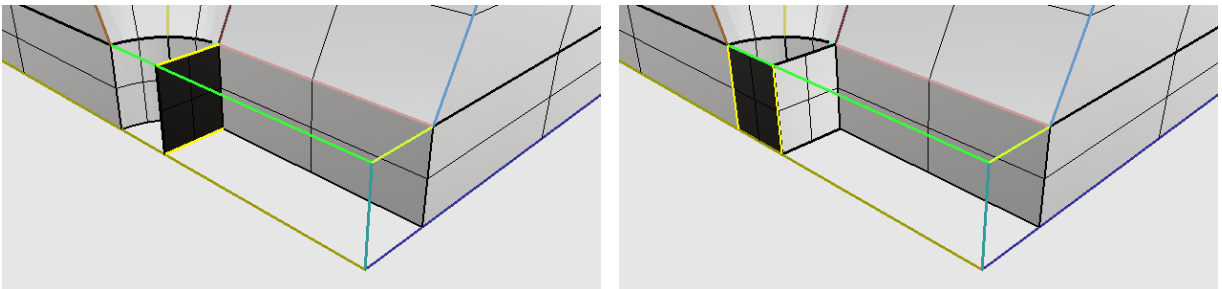


Figure 79: Building 2 additional surfaces

5. Use **_ExtractSrf** and click on the 2 cylindrical surfaces, then press **<ENTER>**. While the two surfaces are selected, hold the **<Shift>** button down and select the **two** small rectangular surfaces you just built. With these **4** surfaces selected, **Edit|Join** them into a **Polysurface** (Figure 80, left). With the **Polysurface** selected, select **Solid|Cap planar holes** to turn it into a solid (right).

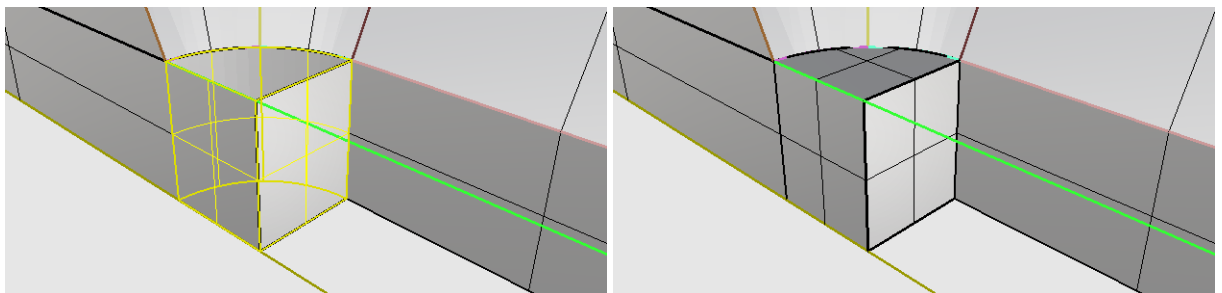


Figure 80: Building a quarter-cylinder solid. Left, the lateral surfaces. Right, the completed solid.

6. Left-click on **Extract Surface** and click on the **larger** of the two rectangular surfaces bordering the remaining **empty** space (Figure 81, left). Select **Solid|Extrude Surface|Along Curve** and click on any

of the smaller horizontal lines perpendicular and adjacent to the extracted surface, and delete the highlighted surface (Figure 81, right).

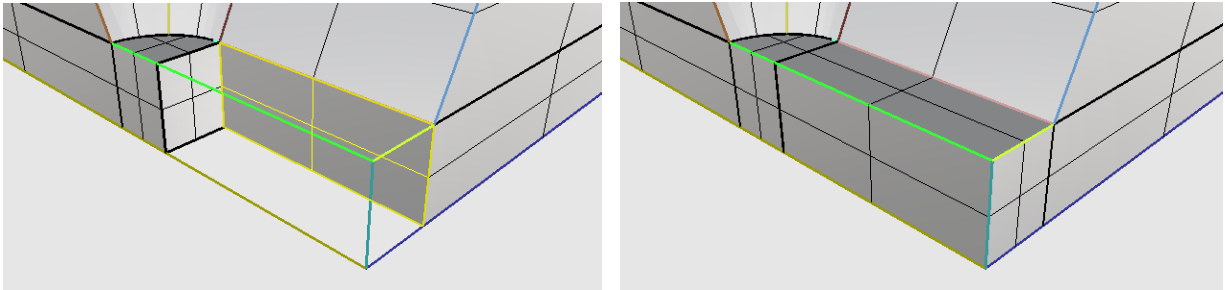


Figure 81: Building the last solid

Your solid model is now complete.

Verification and Cleanup

1. **<Ctrl> A** to select **everything** in your model (Figure 82). Rhino should respond with the message "**8 polysurfaces, added to selection**". If a number of **surfaces** also appear in the list, this is an **indication** that as you were building the solids, you did **not Join** certain **surfaces** into solids. **Find** them and **join** them so that you have **8 closed Polysurfaces (solids)**.

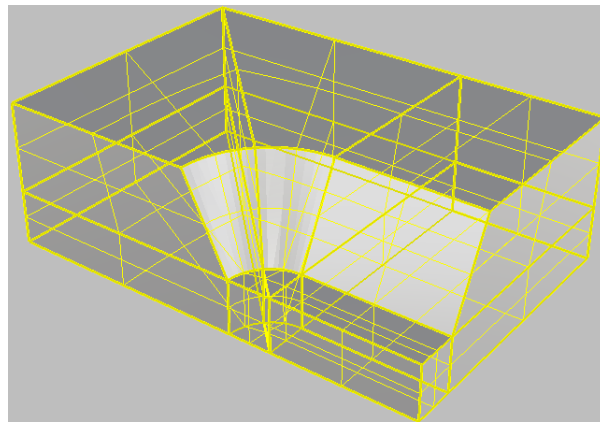


Figure 82: Completed model before verification

2. Select **all 8 solids** (by **clicking** on them or using **Edit|Select Objects|Polysurfaces**) , and select **Edit|Object Properties (F3)**. The **Property** dialog box indicates that the selected **polysurfaces** constitute **8 closed polysurfaces** (Figure 83). Again, **if any of the polysurfaces is not closed** (i.e. not a Solid), join them into solids.

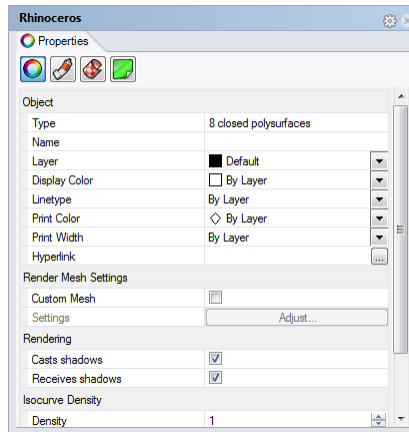


Figure 83: Property dialog box indicating that 8 closed Polysurfaces have been selected

Now, you can assign solids to various layers so that when the mesh is generated elements belong to groups with the same names as the layers.

3. **_Hide** all **solids** and **delete** everything else (curves, points, etc..). Unhide (**_Show**) everything (right-click on the light bulb icon) to **recover** all **solids**. The model now contains **only solids**.
4. Select the **3 upper solids**, then select the **Edit|Layers|Change Object Layer** menu item. When the **Layer for objects** dialog box opens, click on **New**, enter **Upper solids** in the **New Layer** dialog box, click **OK**, followed by **OK**, again in the **Layer for objects** dialog box to complete the moving of the selected solids to the **Upper solids** layer. Similarly, move the remaining lower solids to a new layer called **Lower solids**.
5. If the Layer pane is **not open**, click on the **Edit layers** icon to **open** it. Click on the black square next to the **Upper solids** layer and when the **color chooser** pops up, select **Gold**. Similarly, set the color of the **Lower solids** layer to **Lavender** (Figure 84).

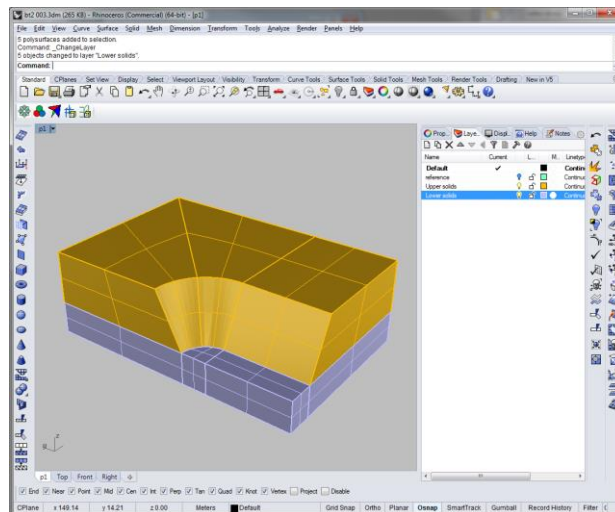



Figure 84: The model after the Upper solids and Lower solids layer have been specified.

Running BlockRanger



1. **Select All solids**, and click on the *BlockRanger* icon: , or just type **_BR** in the command line.
2. Rhino responds with the current BlockRanger Parameters. Press **<ENTER>** to **accept** these parameters and to run *BlockRanger*.

If your **OutputFormat** parameter was set to **FLAC3D**, Rhino should respond with "**8 solids processed, 0 errors. FLAC3D grid file was output to BlockRanger.f3grid.**" If your working directory is properly set, the resulting mesh file with the appropriate extension should appear in your working directory. If you have not set it, type **_SetWorkingDirectory** in the Rhino command line, navigate to your target directory and set it.

If a solid does not comply with the 6, 5 or 4-sided **work rule** or if any of the solid faces is not a 3 or 4-sided face, the corresponding **solids** will **remain highlighted** in your Rhino model.

3. If your **OutputFormat** parameter was **FLAC3D**, start **FLAC3D** and select the **File|Import|Grid** menu item in **FLAC3D**. Select **BlockRanger.f3grid** to **view** your model in **FLAC3D**. For **other** output formats, run your **application** and import the grid. Note that *Rhino* layers become groups (slot 1) in **FLAC3D** (Figure 85).

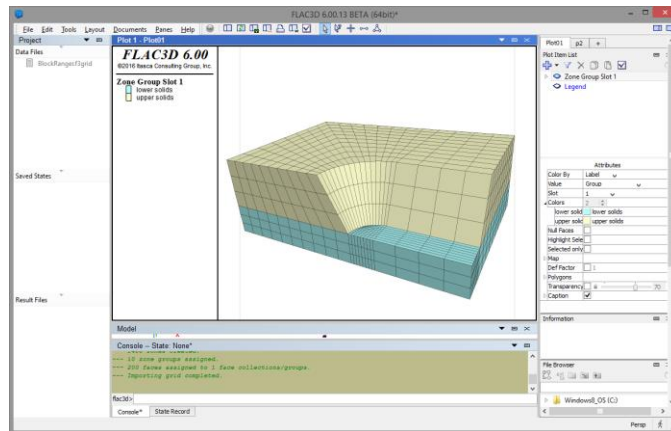


Figure 85: In *Rhino*, solids organized in layers are processed into **FLAC3D** groups.

4. In **FLAC3D**, slot 2 groups correspond to your original solid blocks (Figure 86). That is, a zone is given a slot 2 group name according to the original solid it is contained in.

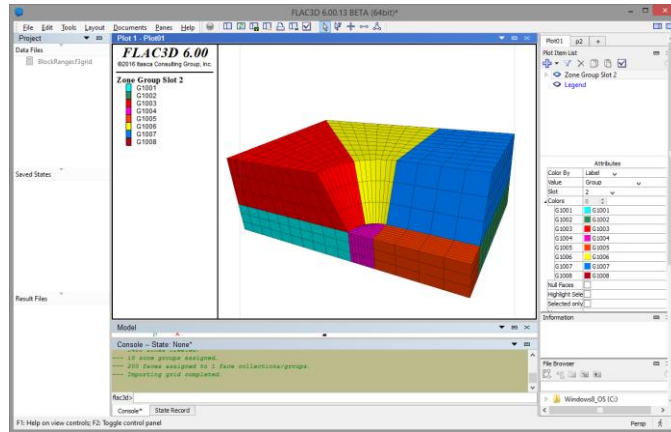


Figure 86: *FLAC3D* slot 2 groups correspond to the individual *Rhino* solids containing the hex meshes.

5. Selecting all **solids**, rerunning **BR** and setting **MaxEdgeLength** to **3.0** results in the grid in Figure 87.

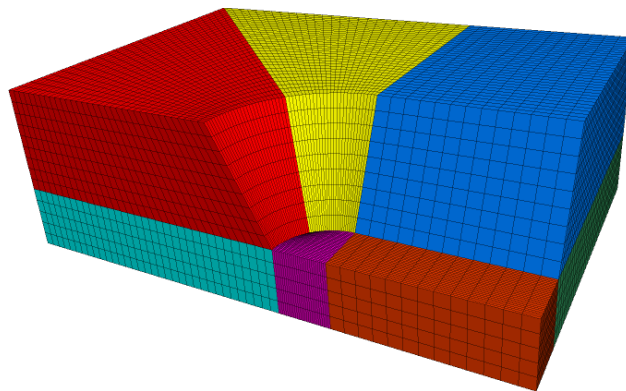


Figure 87: *BlockRanger* run with **MaxEdgeLength** = 3.0.

6. Rerunning with **MaxEdgeLength** 100.0 and **MinEdgeResolution** set to 5 results in the grid shown in Figure 88.

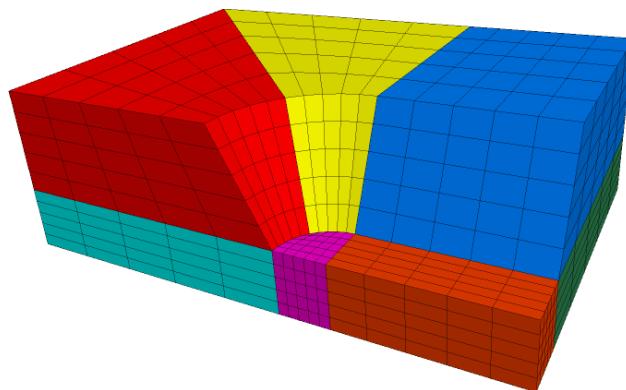


Figure 88: *BlockRanger* run with a minimum edge resolution of 5 instead of the default value of 3.

END OF TUTORIAL 3

Tutorial 4: Intersecting Circular Tunnels (*Griddle*)

In this tutorial, you will learn to create a circular tunnel bifurcating into two circular tunnels making an angle of 20 degrees between them and located inside a block of soil. A *FLAC3D* hex-dominant mesh and a *3DEC* tetra block model will be built (Figure 89).

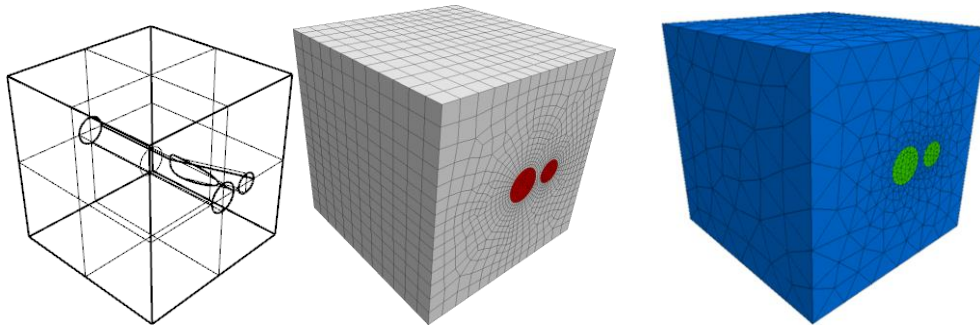


Figure 89: A *Rhino* model, *FLAC3D* hex-dominant mesh, and *3DEC* tetra rigid block model of the bifurcating tunnels.

Startup and creation of a horizontal tunnel of diameter 10 and length 200

1. Start *Rhino*, select the **Large Objects - Meters** template, and double-click on the label of the **Right** view to maximize it. Click on the word **Grid Snap** at the bottom of the screen to activate background grid snap.
37. Select the **Curve|Circle|Center, Radius** menu item and click on the coordinate system **origin** in the **Right** view. This sets the center of the circle. Type **10** on the command window to specify its **radius**
38. Double-click the label of the **Right** view to once again bring up the four views. Select the **Surface|Extrude Curve|Straight** menu item and select the circle in the **Perspective** view, followed by **<ENTER>**. The curve extrusion parameters appear in the command window.
39. If you need to modify any option, simply click on the option in the command window. **Direction** should be, by default, already be set to **1,0,0** since the circle was built in the Right viewport's Construction Plane. Set the remaining options as follows: **BothSides=Yes**, **Cap=Yes (Solid=Yes in RHino 5)**, **DeleteInput=Yes**.
40. Enter **100** on the command line followed by **<ENTER>** to complete the construction of a closed 200 m long horizontal cylinder. Right-click on **Zoom Extents All Viewports**¹ to get a full view of the model so far (Figure 90).

¹ The commands associated with left or right-clicking on each icon can be seen by hovering your mouse over the icon for a second.

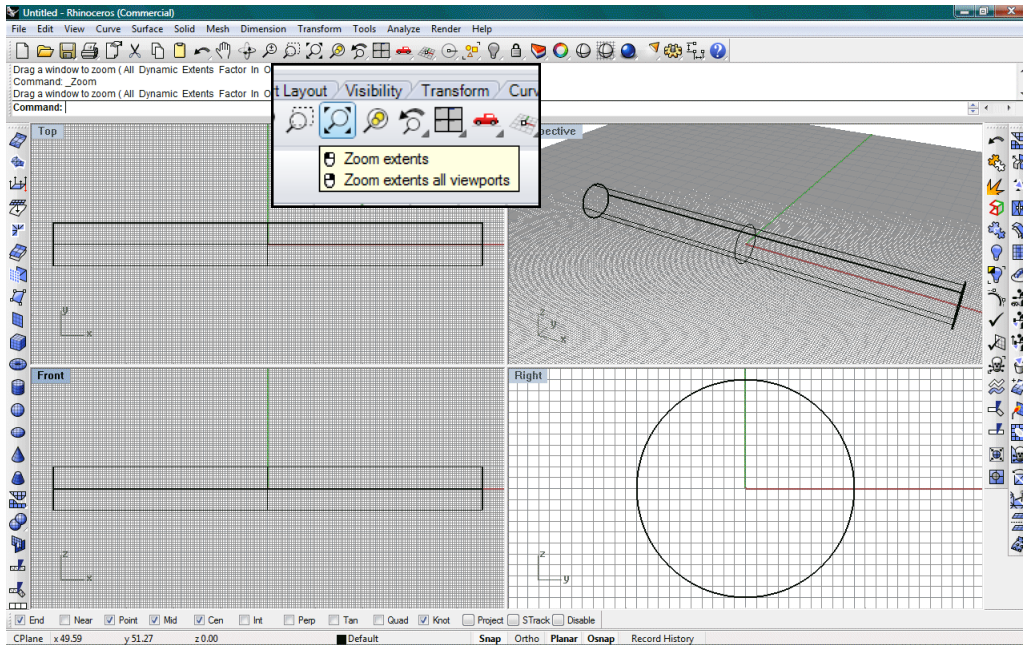


Figure 90: First horizontal cylinder

Creating the bifurcation and the box surrounding it

1. Select the cylinder in any view, select **Edit|Copy** followed by **Edit|Paste** to duplicate it in place. While the copy is still selected, left-click on **Hide Objects** to hide it.
41. In the **Right** viewport, select the remaining visible cylinder and select **Transform|Scale 2D**, and, click on the **origin** of the coordinate system. Type **0.7** followed by **<ENTER>** to scale the cylinder down to a radius of 7 m.
42. While the cylinder is still selected, select **Transform|Rotate**. In the **Top** view, click on the **origin** of the coordinate system. Enter **20** followed by **<ENTER>** to complete the rotation of the smaller cylinder by **20°** around the **z-axis**. Right-click on the button marked **Show Objects** to render both cylinders visible (Figure 91).

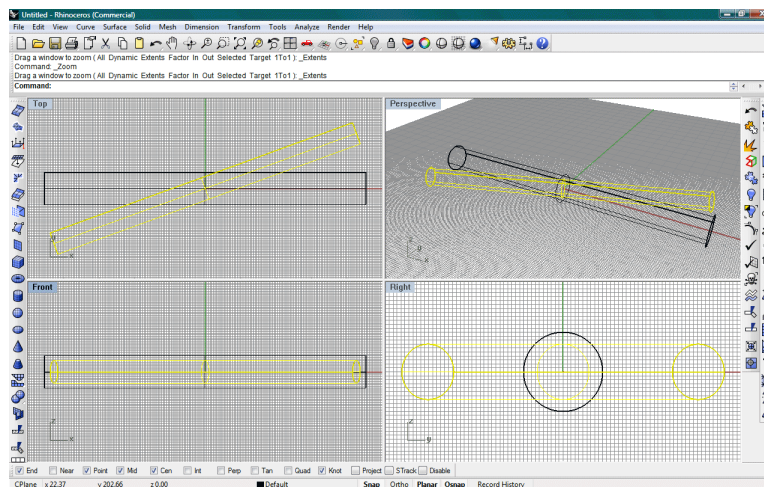


Figure 91: Two intersecting tunnels

You must now split the smaller cylinder with the larger one and delete a portion of the smaller cylinder to create the bifurcation.

43. Hit the **<ESC>** button to unselect everything. Select the **Solid | Boolean Split** menu item. For the Polysurface to split, select the smaller cylinder, then type **<ENTER>**. Select the larger cylinder for the **Cutting polysurface**, followed by **<ENTER>**. This operation splits the smaller cylinder into 3 sections.

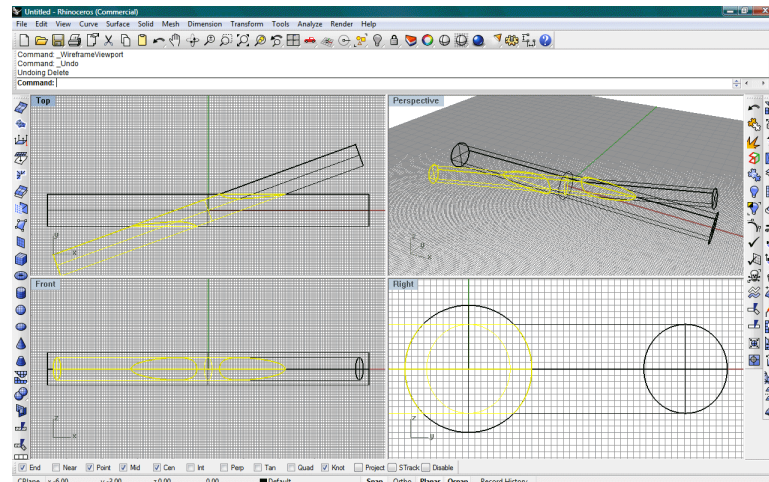


Figure 92: In the Top view, the highlighted section of the smaller cylinder must be deleted to create the branching.

44. In the **Top** view, select the **lower-left** and the **middle** sections of the **smaller** cylinder (Figure 92) and **delete** them. This leaves only one section (upper right) of the smaller cylinder.
45. Select the **two** remaining polysurfaces, and select **Solid | Union** to complete the creation of one single closed solid representing the bifurcation. Click on the button marked **Shaded Viewport** to display a shaded view of the completed bifurcation (Figure 93).

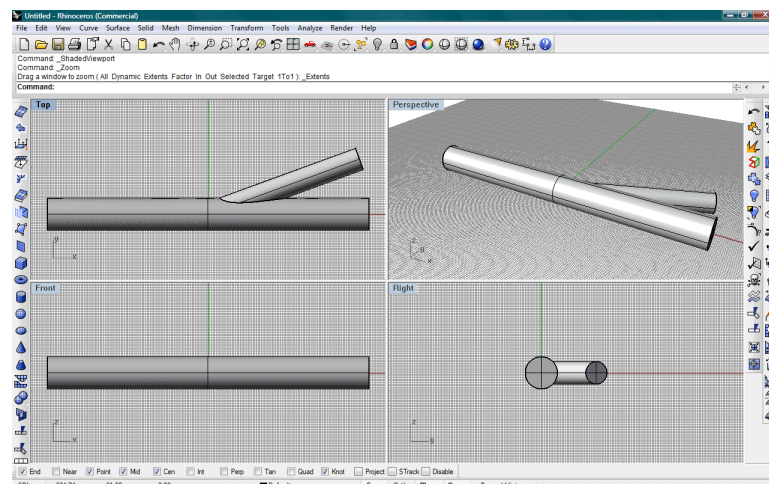


Figure 93: Bifurcation resulting from the union of the larger and the remaining section of the smaller cylinder.

To create a model of the soil surrounding the tunnels, you must create a cube representing the volume in which the tunnels are excavated, and subtract the tunnels from it.

46. To create a **parallelepiped**, select the **Solid|Box|Diagonal** menu item. Enter **-60,-60,-60** followed by **<ENTER>** to specify one end of the diagonal, and **60,60,60**, followed by **<ENTER>** for the other end to create a cube of side 120 centered at the origin.
47. To **subtract** the tunnels from the cube, select the **Solid|Difference** menu item. For the **First set of polysurfaces**, select the box, then type **<ENTER>**. For the **second set**, select the bifurcation, then **<ENTER>** to complete the Boolean difference operation (Figure 94).

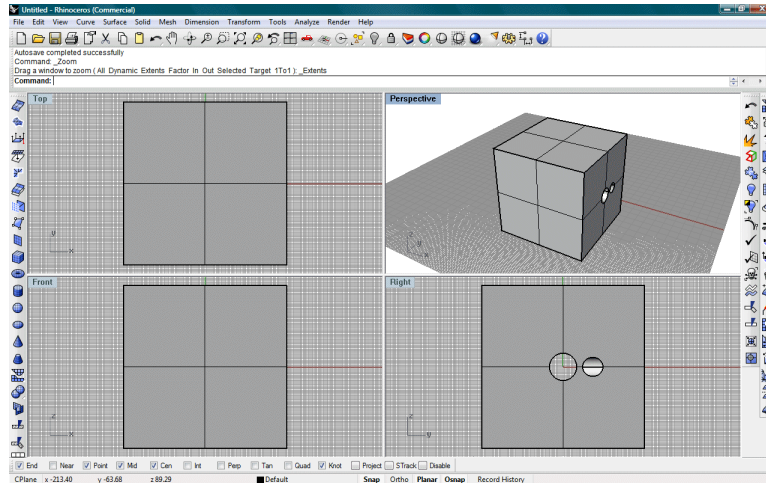


Figure 94: Result of the Boolean difference.

48. We wish to have the tunnels filled with elements, therefore we need to put planar caps on the ends of the tunnels where they intersect the box. Do this with the **_PlanarSrf** command. Type **_PlanarSrf** on the command line and select a tunnel perimeter where it intersects the box boundary and press Enter (Figure 95). Do this for all three openings.

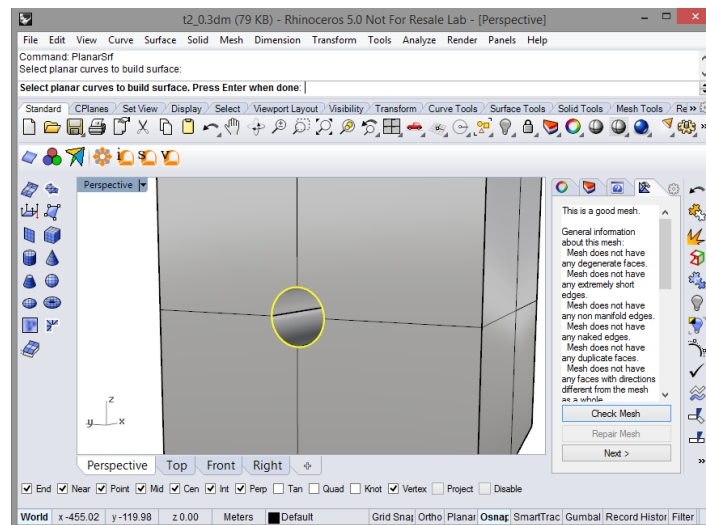


Figure 95: Putting a planar end-cap on a tunnel.

49. Select everything on your screen and type **_NonManifoldMerge** to merge everything into a single BRep.
50. Save your Rhino model. Select File | Save As and when the Save dialog box opens, enter t4.3dm.

Creating a Surface Mesh of the Model

Creating a volume mesh with Griddle requires first generating a conformal surface mesh of the model.

1. To creating a surface mesh based on the solid model, select the model and type **_Mesh** on the Rhino command line. Select the Detailed Controls and fill in the values in the dialog box as shown below in Figure 96. Move the slider to the right side of the scale and click OK to create the surface mesh.

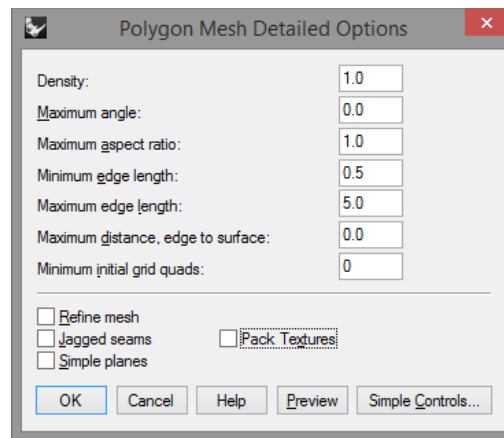


Figure 96: Detailed mesh control options.

2. While the original solid model is still selected, type **_Hide** so the mesh is the only object visible in Rhino model (Figure 97).

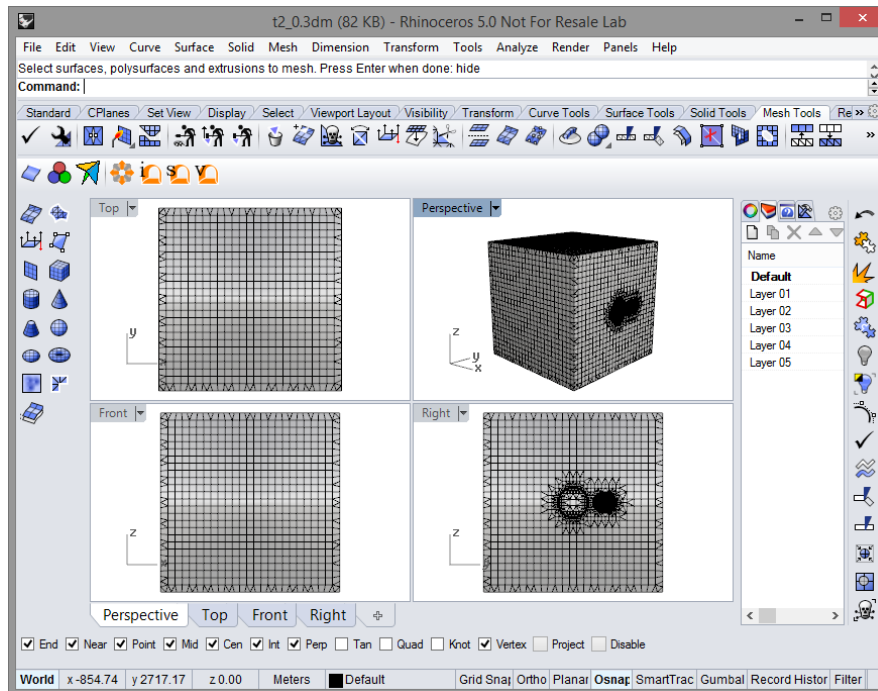


Figure 97: Preview of the surface mesh

3. Select the mesh and type **_Check**. A message box opens and provides global information about the mesh indicating that, among other things, the mesh contains no naked edges, but contains non-manifold edges. **Naked** or Free edges are edges attached to only one polygon. The presence of naked edges indicates that the mesh is not closed. The present mesh is closed. We introduced non-manifold geometry when we capped the tunnel ends (which is fine).
4. We will request Griddle to create a finer mesh near the tunnel and a coarser mesh on the box surfaces. To do this we will Explode the mesh into several components and assign an edge size to the tunnel mesh components. Select the mesh and type **_Explode**. Your mesh will be exploded into 11 components (meshes).
5. Select the mesh of the top of the box and type **_Hide**. Do the same for the other 5 box sides. Do not select or hide the circular mesh caps. All that should be remaining on your screen is the tunnel mesh with the end cap meshes as shown in Figure 98.

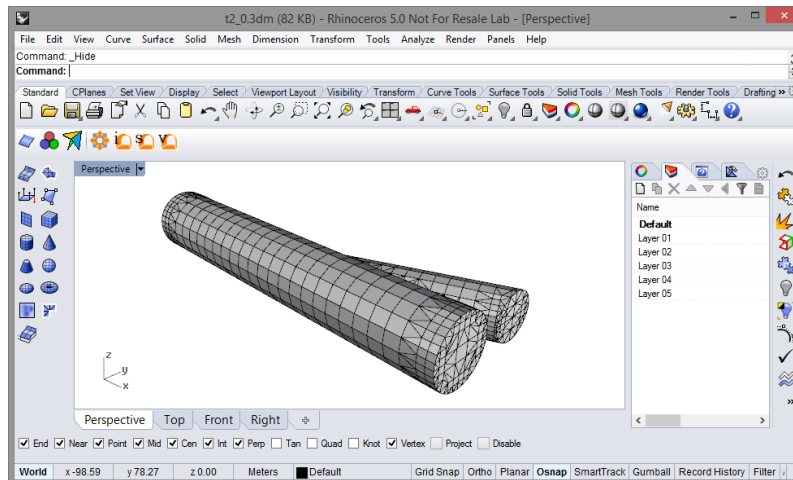


Figure 98: Tunnel meshes.

6. Select the tunnel meshes and type **_Join**. Rhino will join the 5 tunnel meshes into a single mesh.
7. Select the above joined tunnel mesh and type **_Properties** on the command line. In the mesh's Properties Name field enter the number **1.5** to represent the edge size we would like for the tunnel meshes. The reason for using the Detailed Meshing controls in step 1 is to generate smaller surface elements on the cylinders with a good distribution of nodes on the surface. The simple mesh controls would create elongated triangles on these cylindrical tunnels without many intermediate points. It is the surface mesh nodes that are used by *Griddle* to define local mesh size (the 1.5 m we just specified in the Name field). If we used elongated triangles, then we would not have an element size specified for long distances along the tunnels and mesh size would be interpolated from global mesh parameters as well as the nearest mesh nodes (which would be on the end caps of the tunnels).
8. Type **_Show** on the command line so everything is visible. Type **_SelMesh** to select the surface meshes.
9. Type **_GSurf** on the command line to start *Griddle*'s surface remesher with the selected meshes. Select **Mode:QuadDom**, **MinEdgeLength 10**, **MaxEdgeLength 10** and press Enter. *Griddle* will create a quad dominant surface mesh with edge size approximately 10 m on the box boundaries and approximately 1.5 m on the tunnel walls as shown in Figure 99.

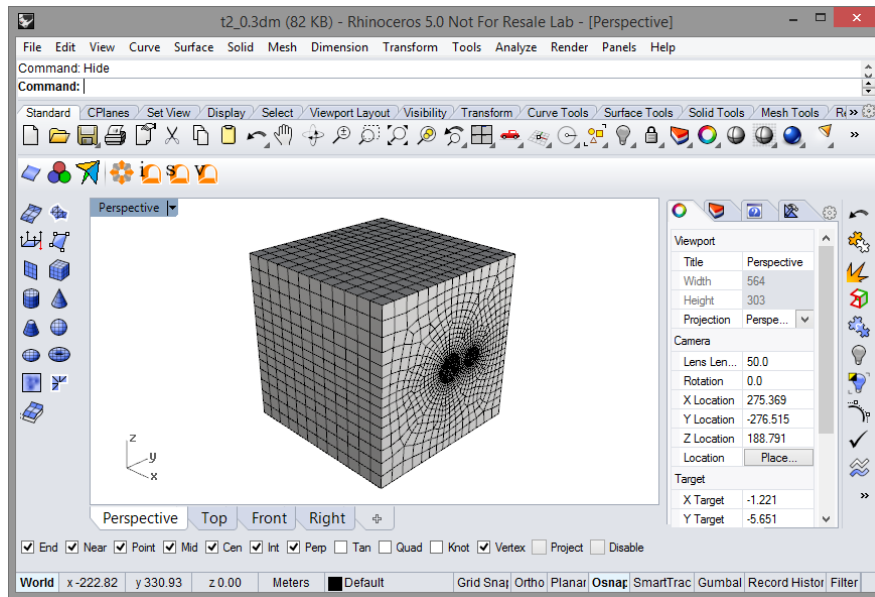


Figure 99: Quad dominant surface mesh generated by *Griddle*.

10. Select the above quad-dominant surface mesh and type **_GVol** on the command line to start the *Griddle* volume mesher. Select **Mode:ConHexDom** and **OutputFormat: FLAC3D** and press enter to create a conformal hex-dominant *FLAC3D* grid (output to GVol.f3grid in your current working directory). Details of the *FLAC3D* grid are shown in Figure 100.

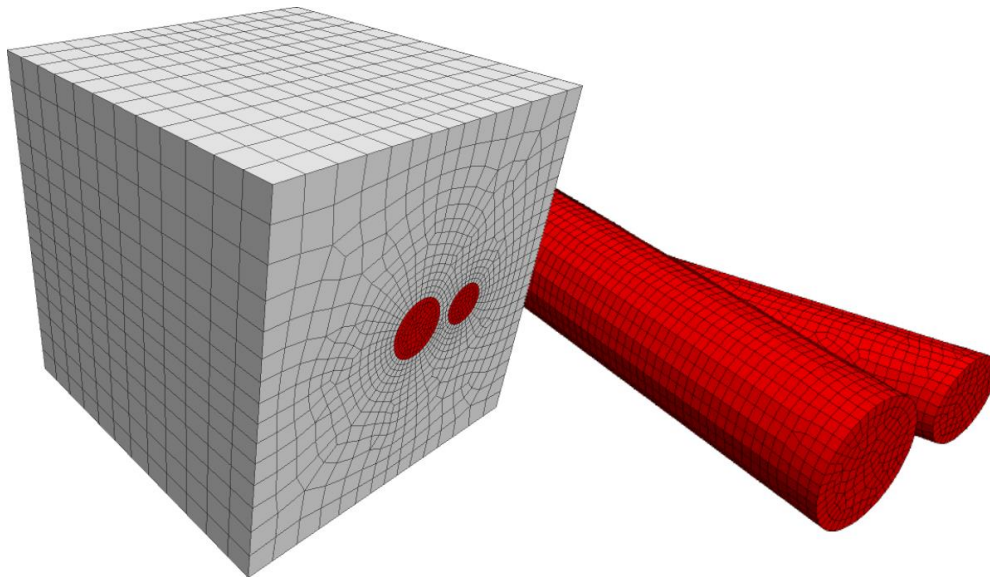


Figure 100: *Griddle* generated *FLAC3D* grid. Entire grid (left) and tunnel detail (right).

Creating a convex blocking for 3DEC

1. Open t4.3dm, select the polysurface and type **_Mesh** to generate a surface mesh similar to what was done above.

2. Instead of generating a quad-dominant mesh, we will generate a tet mesh. Set tunnel mesh size to **3** m (in the tunnel mesh **Properties Name** field). Select all the meshes and type **_GSurf**. Select **Mode:Tri** for a triangle surface mesh. Set **Min** and **MaxEdgeLength** to **20** m in **_GSurf**. Set **MaxGradation** to **1** (to make the sizes of elements change more rapidly from a fine mesh to a coarse mesh), otherwise the mesh will grade too slowly from fine to coarse to obtain the 20 m edge size requested at the box boundaries.
3. Select your resulting triangle surface mesh and type **_GVol** with **Mode:Tet** and **OutputFormat:3DEC**. Your resulting block model (GVol.3ddat in your current working directory) can be called into **3DEC**. Figure 101 shows the resulting block model.

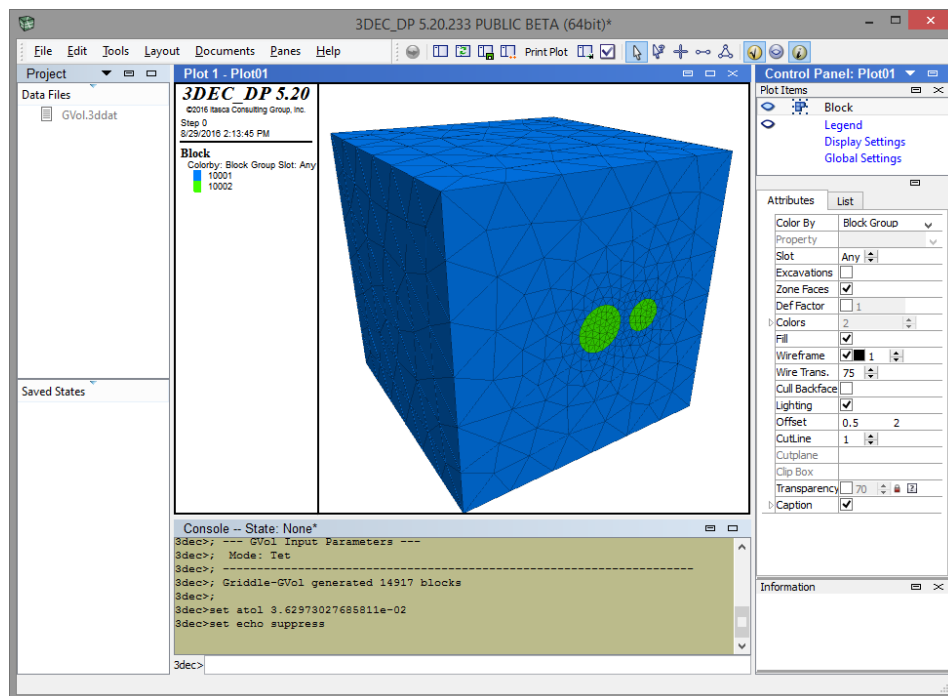


Figure 101: Griddle generated rigid block model for 3DEC.

END OF TUTORIAL 4

Tutorial 5: Mesh Cleanup and De-Featuring (*Griddle*)

In this tutorial, you will learn to clean up an existing triangular surface definition of drifts in a mine available as a DXF file (Figure 102) and create *FLAC3D* and *3DEC* models. Output from many mining systems that are used for ore-grade calculations, surveying, and other planning often consists of triangular surface meshes which generally aren't clean (they are often non-conformal, contain overlapping triangles, duplicates, spurious triangles). These often need to be remeshed in order to be used in numerical modeling software such as *FLAC3D* or *3DEC*.

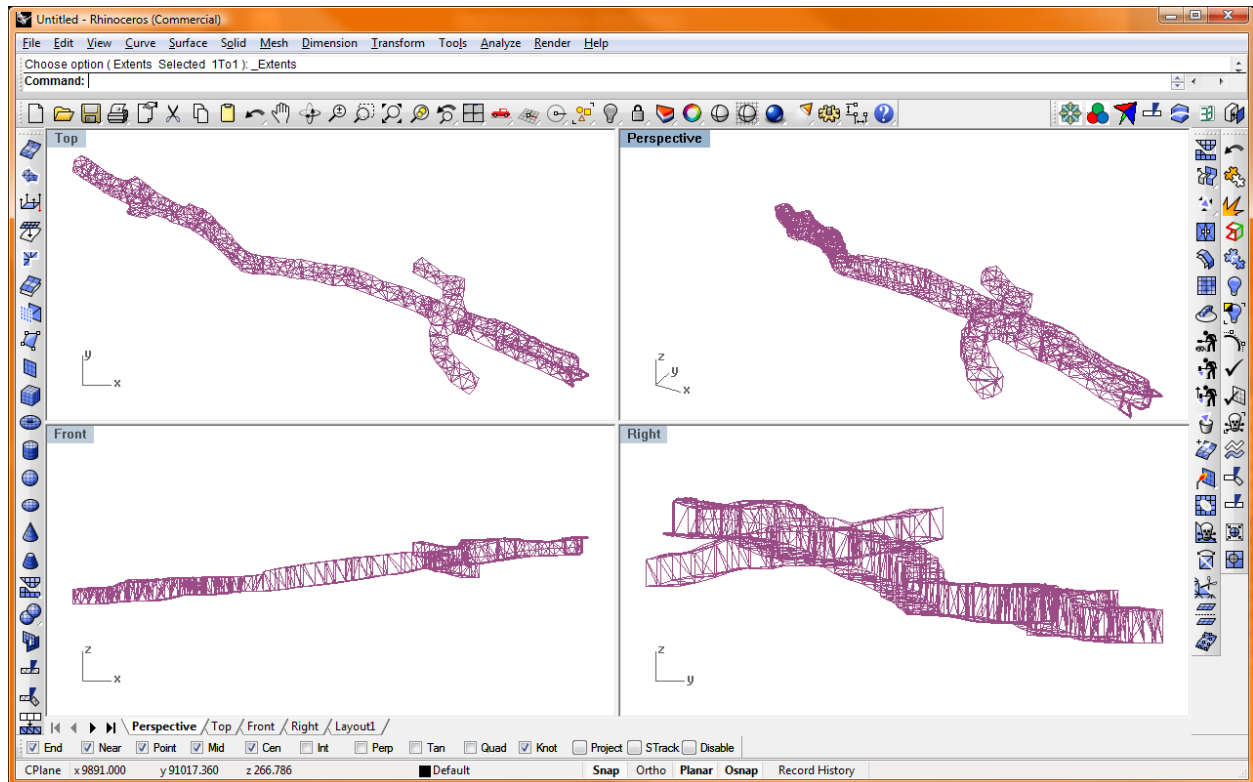


Figure 102: Some drift geometry from a mine.

Startup and reading, joining and centering the model around the origin

4. Start *Rhino*, select **File | New, Large Objects, Meters. _SetWorkingDirectory** to where you have drifts.dxf located and **File | Import drifts.dxf**. Maximize the Top view.
5. Moving the mouse around the screen and looking at the coordinates in **lower-left** corner of the screen (Figure 103) you will notice that the model is far from the origin. This is problematic since it limits the number of significant digits available for geometrical as well as numerical-analysis calculations. It is good practice to translate the model and **center it on the origin** of the coordinate system. This is done in the next step.

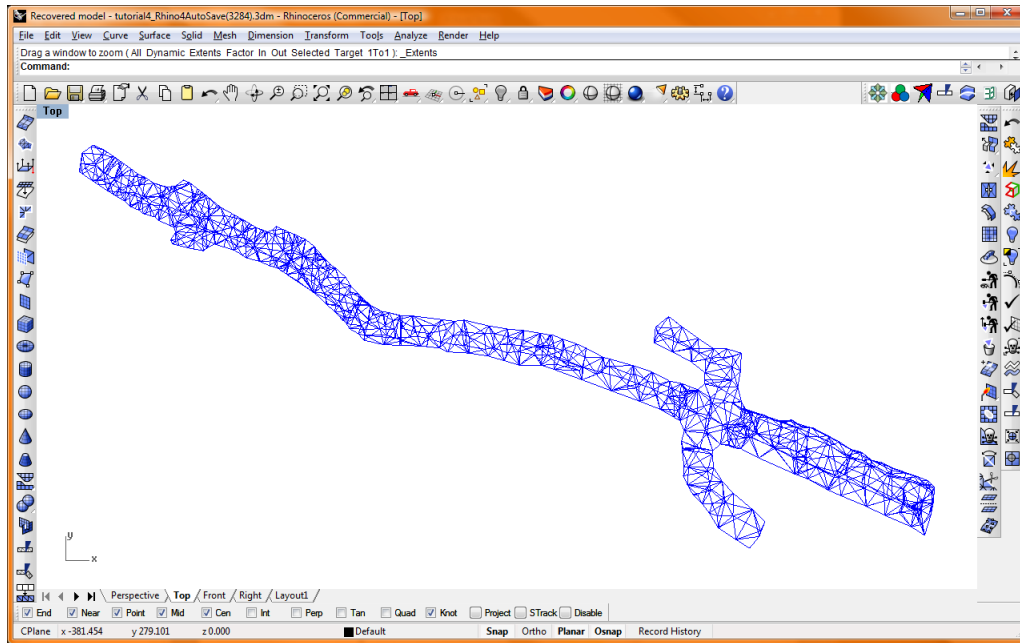
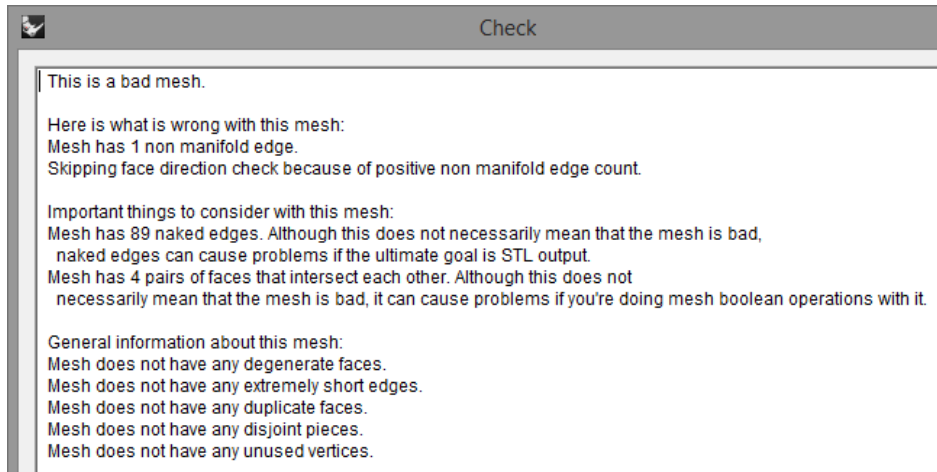


Figure 103: Top view of the model.

6. **Curve|Line|Single Line**, enter **10400,90800,0** for the for the **Start of line**, and **0** for the **End of line**. **Zoom out** in the **Perspective** view to see the **line** you have just created. Select the **line**, left-click on **Edit layers**, hold the button **down** until a new set of icons open. Select the icon **Change object layer**. This opens the **Layer for objects** dialog box. Click **New**, enter **TranslationLine** for the name of the new layer and click **OK** to place line in the layer TranslationLine.
7. Select the mesh and select **Transform|Move** in the menu bar. Click on the first end of the line for the coordinates of the **Point to move from**, followed by **<ENTER>**. Click on the second end of the line for the coordinates of the **Point to move to**. The mesh is now nearly centered on the origin. And, for your own record, you have the translation vector appearing as a line in its own layer called TranslationLine.
8. In the **Layers** Panel, turn **off** the light bulb in front of the **TranslationLine** layer. In the **Perspective** view, click on the **Zoom Extents** to find the model at its new location.

Creating a conformal watertight surface mesh

1. To check the mesh for pathological triangles, type **_Check** in the *Rhino* command line. *Rhino* reports the following problems:



Below are a few definitions to help interpret this message:

Degenerate faces are faces made up of 0-length edges, resulting in a null-area face.

Naked edges or free edges are edges attached to a single triangle. The presence of naked edges means that the volume defined by the surface mesh is not watertight.

Duplicate faces are just that: extra copies of the same face at exactly the same location. Automatic mesh generation with Griddle requires the input surface to be free of duplicate faces.

Non-manifold edges are edges attached to more than two triangles: in other words, more than 3 triangles attached to the same edge. Non-manifold edges may exist by design (for example, when a wall partitions a tunnel in two) but, in the present example, their presence indicates problems with the geometry.

2. Select Analyze|Distance from the *Rhino* menu and select two mesh vertices that across the width of one of the drifts. Similarly measure the height of the drifts. Drifts are approximately 5 m wide and high. This will give us an idea of how large we should make our final elements and the tolerances we might use if merging nodes.
3. the mesh and click on the icon marked **Match mesh edges** in the **Mesh** Tools toolbar.
4. The command **_MatchMeshEdges** has 3 options. Type **d** or **move** your cursor over the word **Distance** to highlight it, and then **click** on it to activate the **Distance to adjust** option. Enter **0.1** and hit **<ENTER>**. Hit **<ENTER>** again to complete the merging of matching edges that are at most separated by 0.1 meters. This is a fairly large tolerance but it operates on edges that are fairly parallel to each other. High values of the merge distance tolerance may introduce distortions in the mesh.
5. To check the mesh again, type **_Check** and you will see that the number of **naked** edges has dropped from 89 to 17.
6. Maximize the perspective view and type **_FlatShade** to get a shaded view of the mesh. You will notice that the one of the drifts is open (Figure 104). We need to close this to eventually obtain a closed volume. We will trim this ragged end of the drift with a line and close the end with a mesh.

Maximize the Top view. Type **_Line** and enter **-308,249,0** for the start of line and enter **-314,236,0** for the end of line. Type **_MeshSplit**, select the mesh as the object to split and press Enter, then select the line segment you just created as the cutting object and press Enter. The mesh is now split into two pieces. Select the line segment and **_Delete** it. Select the ragged portion of the mesh (see highlighted mesh in Figure 105) and **_Delete** it.

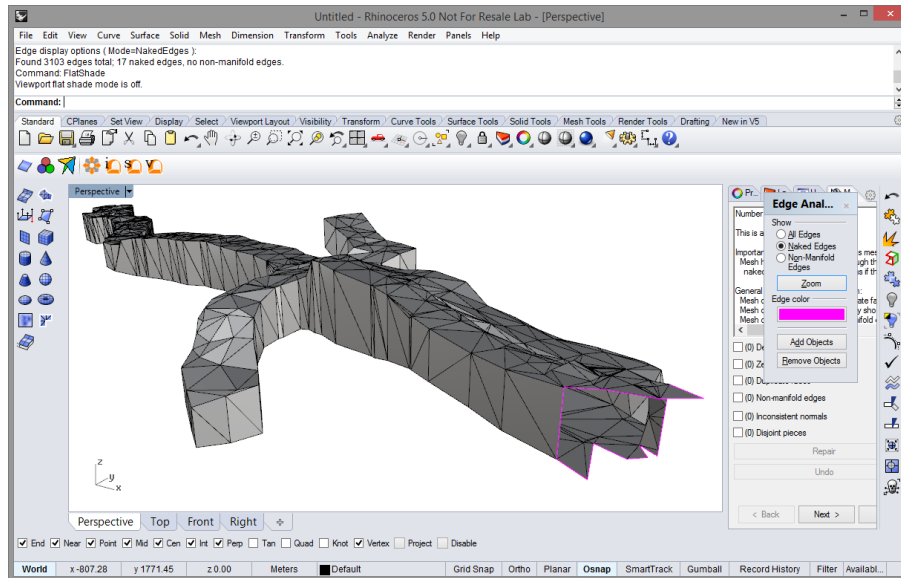


Figure 104: Highlighted open end of drift.

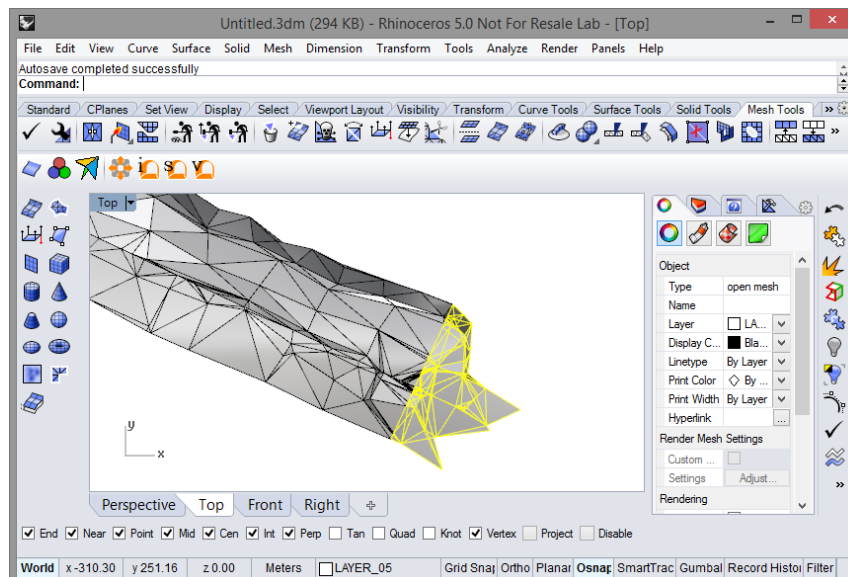


Figure 105: Cleaning up the mesh at the open end of the drift.

7. Maximize the perspective view (double click on the Top view label, then double click on the Perspective view label). Rotate and zoom your view so you see the open end of the drift you just

cleanly split off. Type **_FillMeshHole** and select one of the open edges on the end of your drift. Rhino will automatically fill this with triangles. Some of the triangles in your mesh may not be shaded while others are shaded. This is caused by the mesh having inconsistent normal directions. You can remedy this with **_UnifyMeshNormals** and the **_Flip** command. If your shaded view does appear transparent (you can see the inside), then try the **_Flip** command a second time. Your repaired end of the drift should look similar to Figure 106.

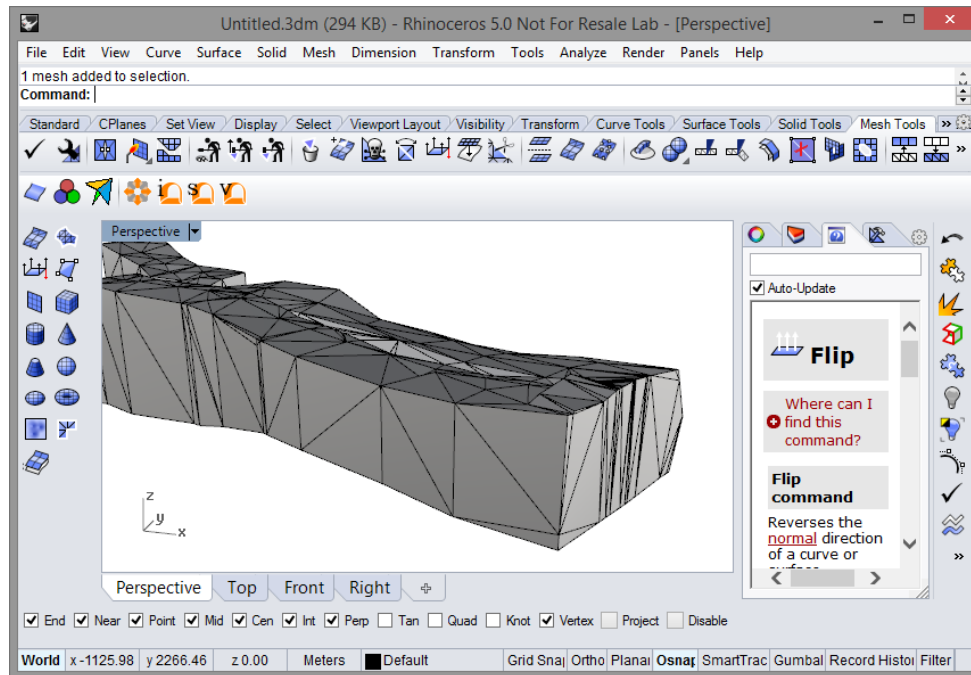


Figure 106: Open end of drift after filling with triangles.

8. At this point we could continue to use *Rhino*'s built-in mesh editing tools summarized below in Table 1 to remove degenerate, duplicate, non-manifold and isolated triangles, or we could generate an almost clean mesh with *Griddle*, using the current mesh as input. Select the mesh and type **_GSurf**. Select **Mode:Tri, MinEdgeLength 0.5, MaxEdgeLength 5, MaxGradation 1** and press Enter. We would like our resulting mesh to have edges no larger than the drift height and width (approximately 5 m) and we would also like to preserve some of the smaller scale features with the MinEdgeLength of 0.5 m (you should experiment with these and other **_GSurf** parameters). Your remeshed mesh should look similar to that shown in Figure 107.

Table 1: A few of the mesh clean-up tools available in Rhino.

To remove all degenerate faces, if there are any, select the mesh and click on the icon marked Cull degenerate mesh faces in the Mesh tools toolbar.
To remove all duplicate faces, if there are any, select the mesh and click on the icon marked Extract mesh toolbar located in the Mesh tools toolbar. Hold the button down to open the Extract mesh toolbar. Click on the icon marked Duplicate faces. All duplicate faces will be highlighted. Press <DELETE> to eliminate them.
To remove all isolated triangles directly connected to non-manifold edges, in the Rhino command window, type <code>_ExtractNonManifoldMeshEdges</code> , followed by <ENTER>. When the <code>ExtractHangingFacesOnly</code> option appears, click on it and set it to YES. Click on the mesh to highlight them. Hit <DELETE> to get rid of all isolated triangles attached to non-manifold edges.
To remove all triangles connected to non-manifold edges, select the mesh and in the Rhino command window, type <code>_ExtractNonManifoldMeshEdges</code> , followed by <ENTER>. Delete the highlighted triangles if there are any.
Select the mesh and click on Split disjoint mesh in the Mesh Tools toolbar. This operation splits disconnected triangles into separate mesh entities so you can select them individually.
If the splitting operation results in more than one mesh, select the main mesh, left-click on the Visibility icon, hold the mouse down and select Invert selection and hide objects. This selects the smaller disjoint pieces. Press <DELETE> to eliminate them.

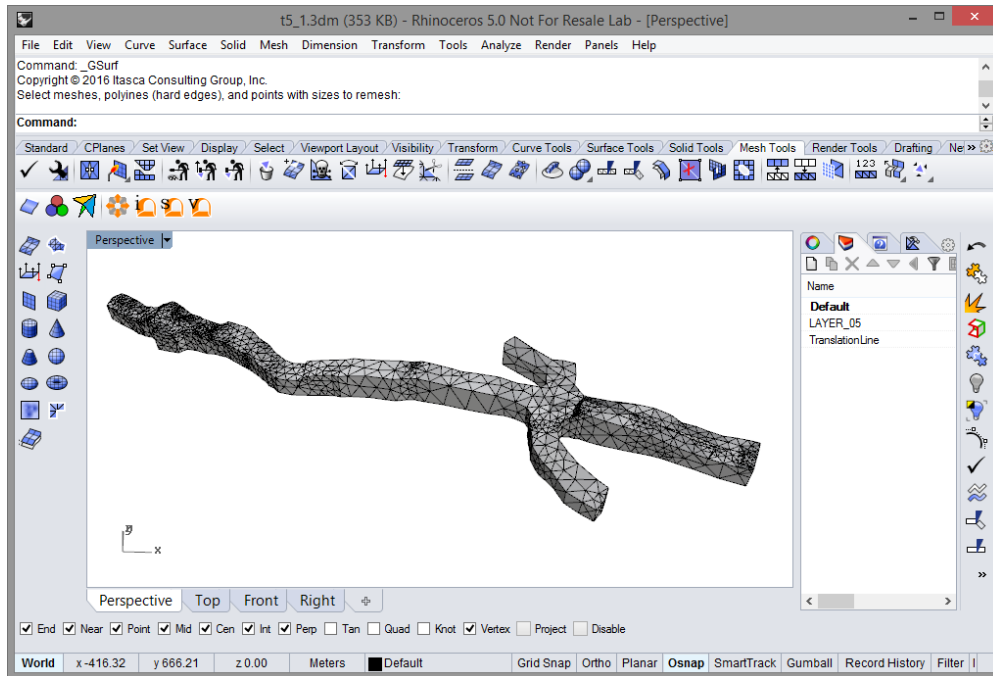


Figure 107: Triangle mesh after remeshing with GSurf.

9. Select the mesh and type **_Check** (or **_MeshRepair**). *Rhino* will report, among other things, that this mesh has faces that intersect each other. We could try using the Griddle surface mesh intersector (GInt) to clean up the intersecting faces, however, this mesh contains a fold, with almost parallel faces (a pathological case which GInt does not handle). To see where the intersections occur type **_TestMsx** (and select the mesh), then type **_SelCrv** (to select curves that represent where the mesh intersects). The **_TestMsx** is a test function in *Rhino* that calculates the intersection lines of meshes. While the curves are selected, in the menu **View | Zoom | Zoom Selected** to zoom into the problem area of the mesh (Figure 108).

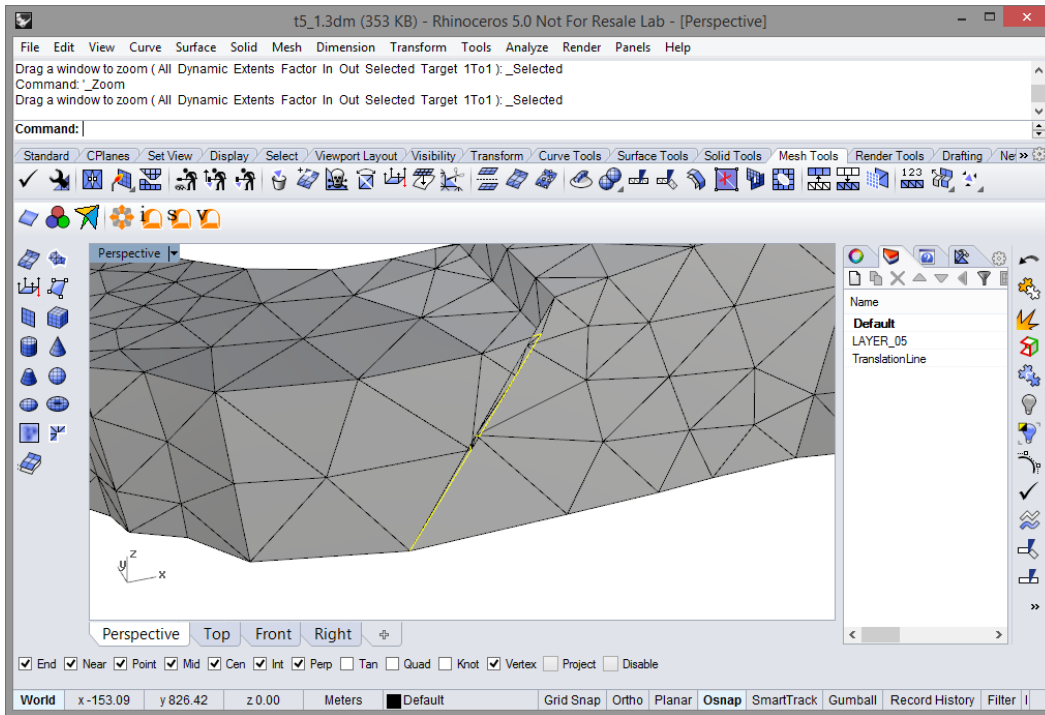


Figure 108: Area, highlighted in yellow, where mesh intersects itself.

10. We will fix this intersecting area by manually deleting triangles and filling the resulting hole with new triangles. **_ExtractMeshFaces** and select the triangles near the highlighted curves, press Enter to finish the extracting and then **_Delete** these triangles. Repeat this last step until you have a hole in the side of the drift similar to that shown in Figure 109.

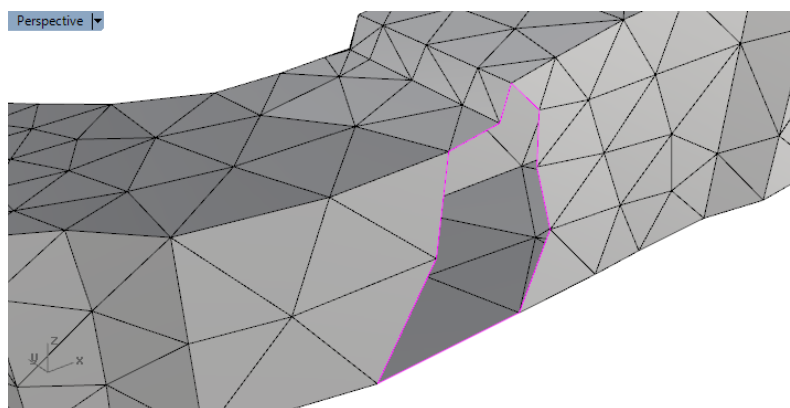


Figure 109: Mesh with overlapping triangles removed (remaining hole is outlined for clarity).

11. Type **_FillMeshHole** and select an edge on the hole boundary to fill this hole. Select the mesh and type **_Check** (it should be a good mesh). Save your *Rhino* model. This clean conformal surface mesh will be the basis for creating other meshes with specified element types and size.

Creating a box representing the computational volume

Now that we have a clean surface mesh that can be easily remeshed, we will create a box surrounding the drifts so there are zones both inside the drifts and in the surrounding rock.

1. To create a box containing the model, select the mesh and type **_BoundingBox** followed Enter.
2. Type **_SetDisplayMode** and select **Wireframe** so you can see the outline of the bounding box and mesh of the drifts. Select the mesh and **_Hide** it.
3. Select the bounding box and type **_VolumeCentroid**. This puts a *Rhino* point in the center of the box. We will use this point as an origin to scale the box around.
4. Type **_ScaleNU**, select the box, and press Enter. The non-uniform scaling function next asks for an origin. **Select the center point** you just created (make sure *Rhino* point snapping is turned on).
5. Type **1.5, 1.5, and 2.0** for the x-, y-, z- axis scaling factors. The scaled box no longer touches the edges of our drifts. Delete the center point of the box. It is no longer required.
6. Select the box and type **_Mesh**. Use the Simple Controls and create a coarse triangular mesh. Select this mesh and type **_GSurf**. Use **Mode:QuadDom, MinEdgeLength 10, MaxEdgeLength 10** to create a mesh for our boundary box.
7. Type **_Show** so to make all objects visible. Select the triangle mesh of the drifts and type **_GSurf**. Use **Mode:QuadDom, MinEdgeLength 1, MaxEdgeLength 5, MaxGradation 0.1** to obtain a quad-dominant surface mesh of the drifts.
8. Type **_SelMesh** to select all the meshes. Type **_GVol** and use **Mode:ConHexDom** with *FLAC3D* output to generate a volume mesh. The resulting volume grid *GVol.f3grid* can be read into *FLAC3D* and used for numerical simulations (Figure 110).

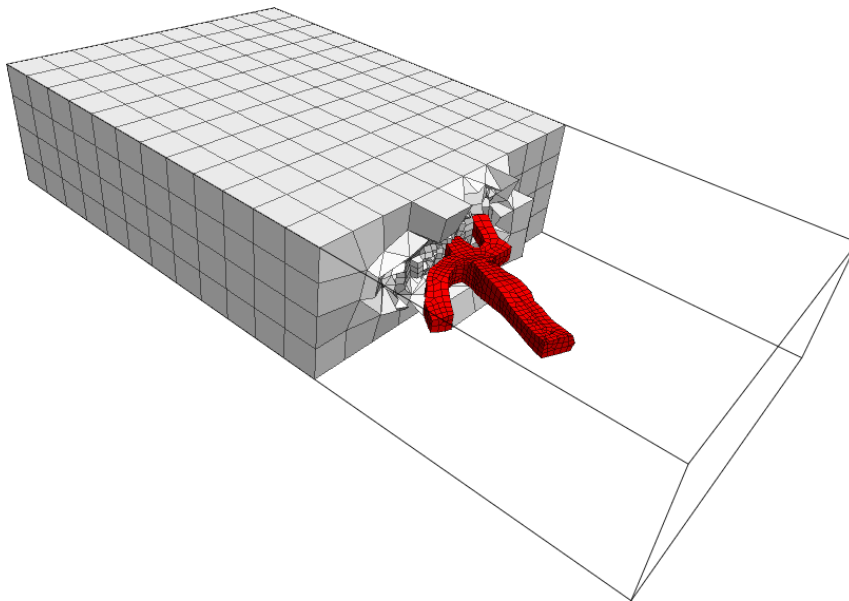


Figure 110: Griddle generated hex-dominant volume grid read into *FLAC3D*.

9. We can also **_SelMesh**, and run *GVol* with **Mode:Tet** with *3DEC* output to obtain a rigid block grid which can be read into *3DEC* (Figure 111). Note, *3DEC* output consists of a data file with *3DEC*

POLYHEDRON commands which create rigid blocks in *3DEC* (these are not deformable zones). If you wish to make the blocks deformable in *3DEC* you need to use the *3DEC* GENERATE command to populate the rigid blocks with zones.

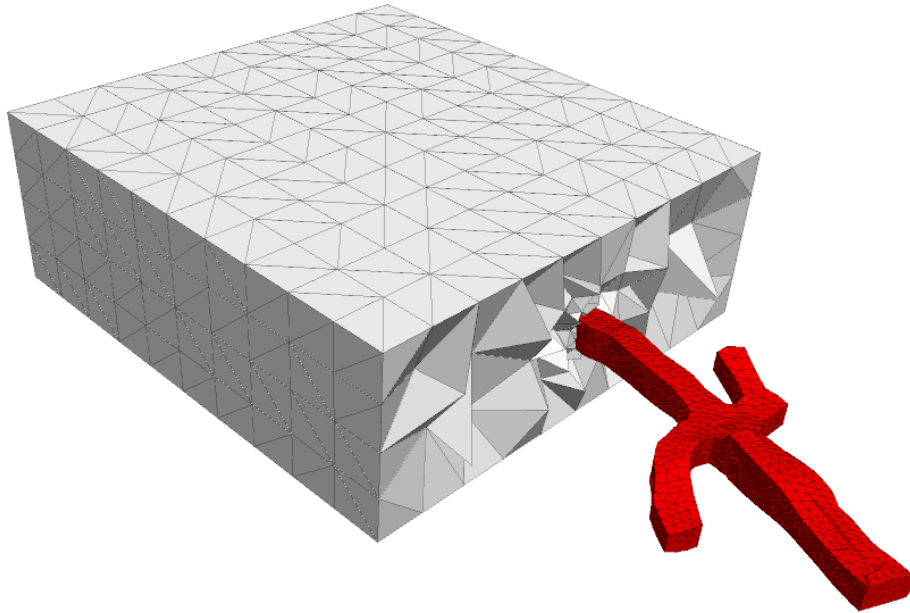


Figure 111: *Griddle* generated rigid tetrahedral blocks read into *3DEC*.

END OF TUTORIAL 5

Tutorial 6: Open Pit Model from Contour Lines (*Griddle*)

In this exercise, you will construct a model of an open pit by manually tracing over topographic contour lines to simplify the geometry (Figure 112). We could drape a *Rhino* surface over the contours which would approximate the sloping sides of the pit, however, in this case, we would like preserve some of the bench details in our model. In order to do this, we will manually simplify each bench and use it directly in our model.

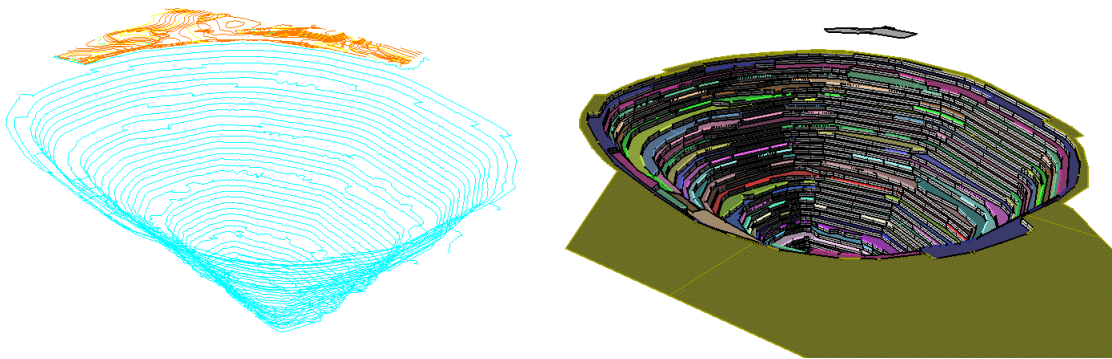


Figure 112: Simplified model (right) obtained by tracing over contour lines shown at left.

Startup and manual retracing of contour lines

7. Start *Rhino*, select **Large Objects, Meters** for your template, and select **File|Import** to read in **pit_contours.dxf**.
8. Type **_SetWorkingDirectory** in the command line and set your working directory to your current working directory.
9. In the Front or Right view, zoom in to the pit bottom, and select the contour line representing the floor of the pit (Figure 113).

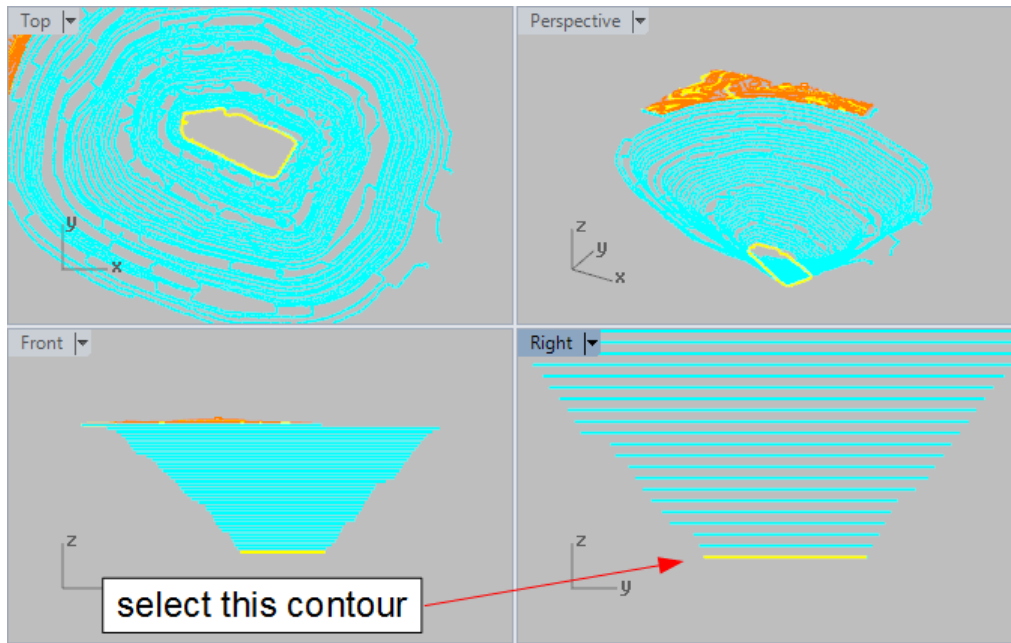


Figure 113: The highlighted contour line represents the floor of the pit.

10. While the “floor” contour (the innermost) is selected, type **_Invert** followed by **_Hide** to hide everything except for what was highlighted.
11. Make sure that **Object Snap** is on by clicking over the word **Osnap** at the **bottom** of the screen, and make sure that **Near**, **Point**, **Mid** and **Cen** are **checked** (Figure 114). This ensures that while tracing over contour lines, Polylines can snap to existing points.

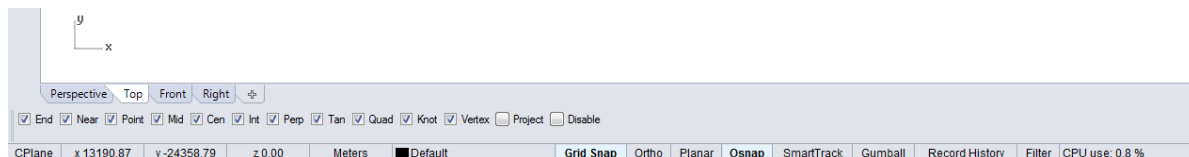


Figure 114: Turning on osnap as well as End, Near, Point, Mid & Cen

12. Maximize the Top view and **View|Zoom|Zoom Extents** to maximize the floor contour on your screen. Type **_Polyline** and click on a starting point somewhere on the contour line in the **Top** view (Figure 115).

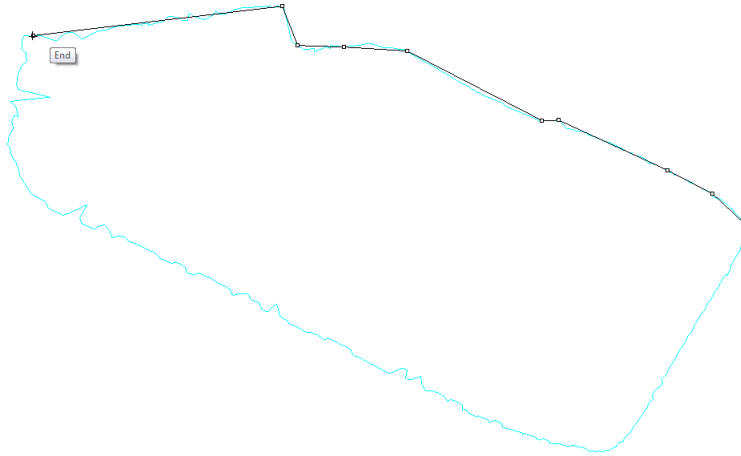


Figure 115: Building a polyline starting from a point on the floor contour.

13. Continue clicking on points, approximating the contour as you go, all the way around. For the **last** point, click exactly on the **first** point to ensure that the resulting Polyline is **closed** (Figure 116).

Please note that, as you approach the starting Point, and if you have Point checked as objects to snap to in Osnap, the word **Point** pops up among all the other options such as End, Int, Knot, etc. When the word Point pops up, it is an indication that you are over the first Point you created, so you can click it to close the curve.

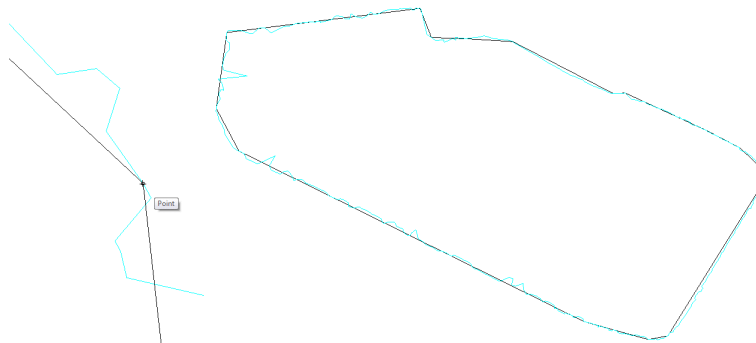


Figure 116: Close-up showing the closure of the Polyline by clicking on the starting point (left), and overall view of the completed polyline representing the floor of the pit (right)

14. **_Delete** the original (light blue) contour line, select the Polyline you just created and type **_PlanarSrf** to create a surface spanning across the Polyline. Delete the Polyline (Figure 117). Rename "Layer 01" to "surfaces". Select the polysurface, right click on the "surfaces" layer and select Change Object Layer.

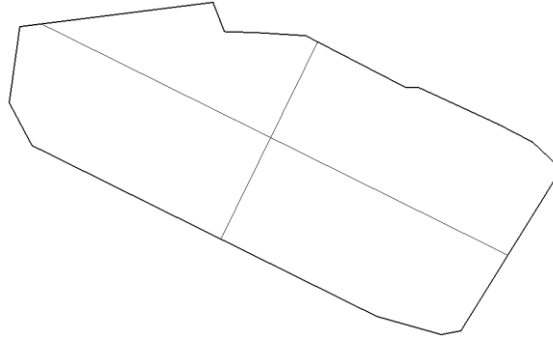


Figure 117: Planar surface representing the floor of the pit

15. Type **_Show** to show all objects (all contour lines), and select the next (next level up) contour line. While holding the **<SHIFT>** button down, select the surface representing the bottom of the pit (Figure 118).



Figure 118: Surface (representing the floor) and the next bench contour line selected.

16. Type **_Invert** and **_Hide** to hide everything except the surface and the next contour line (Figure 119).

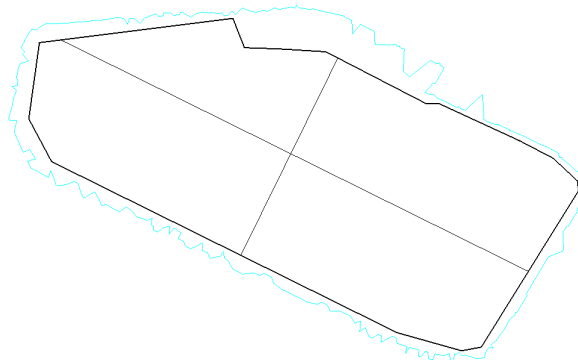


Figure 119: After Invert Selection and Hide Object, the surface and the next contour line remain.

17. Type **_Polyline** and start tracing the next contour line, in the Top view, all around until the polyline is closed (Figure 115). Make sure your polyline does not touch or cross the boundary of the lower level bench you just outlined in the previous step.

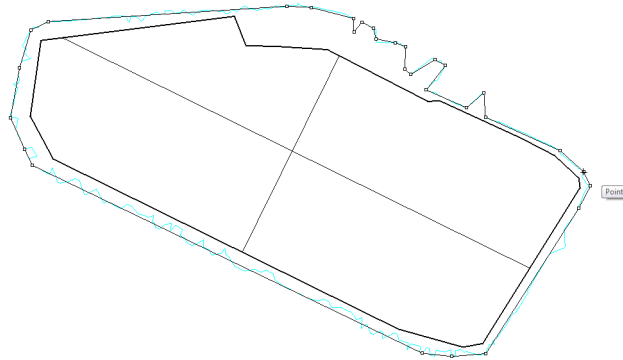


Figure 120: The second retraced contour.

18. Delete the original contour line, select the Polyline you just created and type **_PlanarSrf** to create a surface spanning across the Polyline. Delete the Polyline and change the layer of your planar surface to “surfaces”. Instead of manually tracing the contour lines you could also let Rhino create an approximate fitting contour line by 1) selecting the contour line, 2) typing **_Divide** and dividing the curve into **50** segments, 3) move (or delete) the points you are not satisfied with, snap them to new positions on the original contour line 4) **_SelPt** to select the points, 5) **_CurveThroughPoint** (select **Closed=Yes** option), 6) **_PlanarSrf** through the curve you just created, 7) delete original contour and the new contour line you just created, 8) **_SelPt** and **_Delete**.
19. Continue this process with the remaining pit contours. Once again, when tracing a contour line, ensure that your polyline does not cross or touch previously traced contours.
20. After all the contours have been traced and turned into surfaces, type **_Show** (to show all hidden objects). Select everything (**Ctrl-A**) and **_ColorizeObjects** to colorize the selected objects (Figure 121). Save the model as **Tut6A.3dm**. **3dm** is the native Rhino file format.

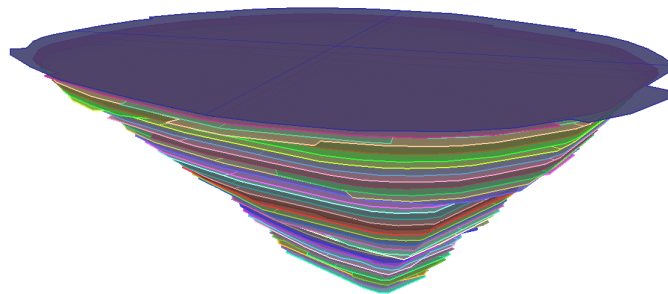


Figure 121: Perspective view of all the planar surfaces (colorized) representing the benches.

Creating the benches

21. Select the **Top** view, select all surfaces and type **_DupBorder** to extract the curves representing the boundaries of all the surfaces you have created. These curves are essentially the polylines you created by tracing over the contours (Figure 122).

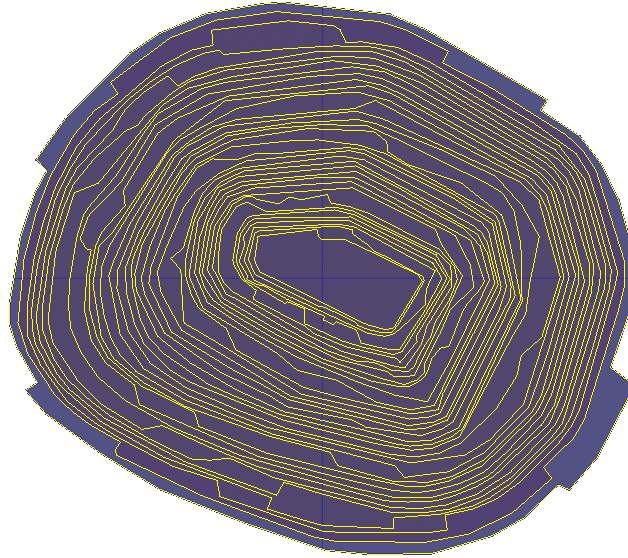


Figure 122: Top view of all the surfaces with all the border curves highlighted.

22. We will now trim each surface with the border line of the surface directly below it. Make sure all four views are open. We will work in the Perspective and Right views. Make sure your surfaces are shaded in your perspective view to make them easier to see (click somewhere in the Perspective window and type **_FlatShade**). To trim the highest bench surface, first select the curve immediately below the top surface in the Top view as shown in Figure 123. While the curve is selected, type **_Trim**, and select a point well inside the top surface in your Perspective view to trim away the part of the top surface located **inside** the curve. You should end up with a trimmed surface similar to that shown in Figure 124.

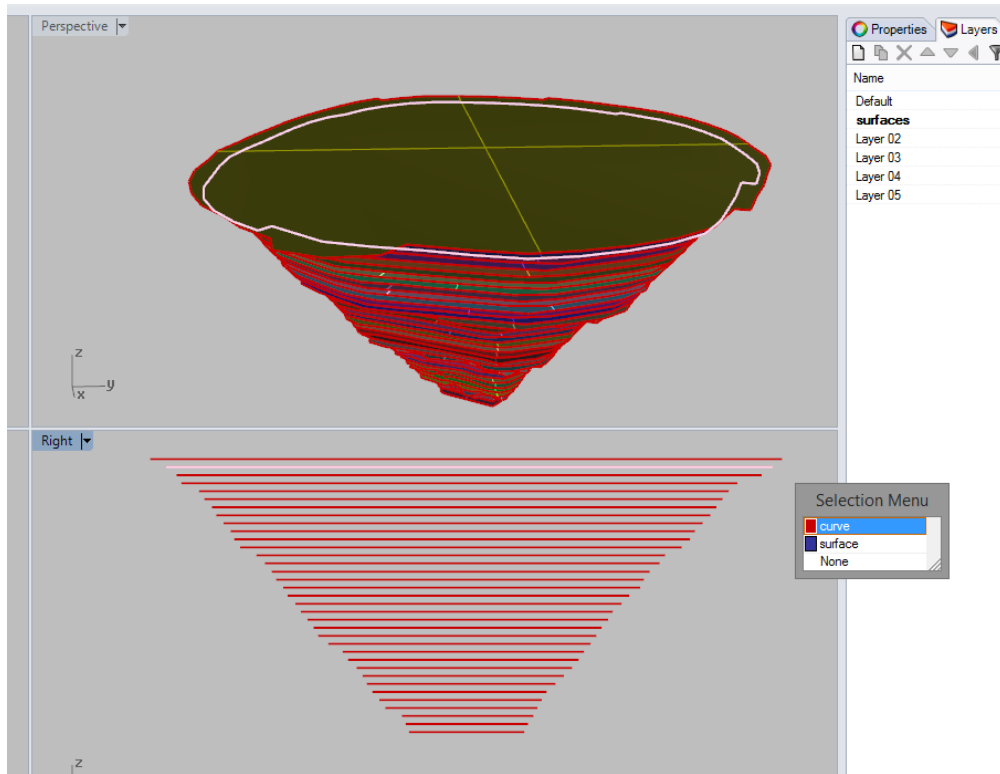


Figure 123: Perspective and Right views showing trim curve selection for trimming surface above it.

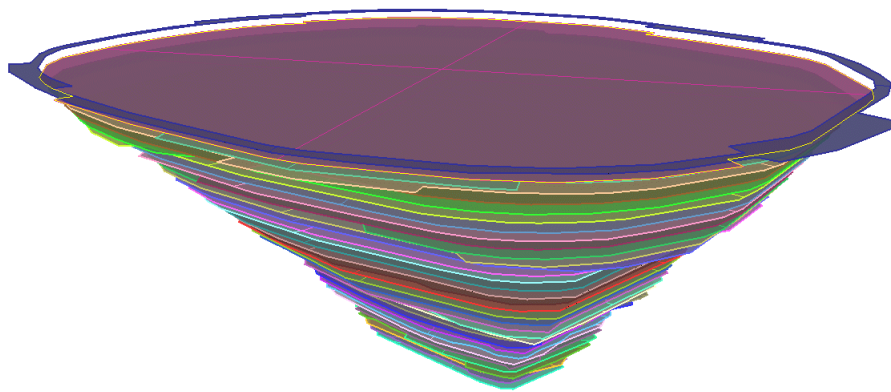


Figure 124: Perspective view showing trimmed top surface obtained by trimming the top surface with the outer border of the surface immediately below it.

23. In the top view, select the topmost surface and curve and **_Hide** them.
24. Repeat the above two trimming steps with the currently visible uppermost surface and the curve directly below it. Work your way down the entire stack of surfaces. Continue this sequence of operations until you reach the floor of the pit, which is a surface which doesn't need any trimming (Figure 125).

25. **_Show** everything. Save your work as **Tut6B.3dm**.

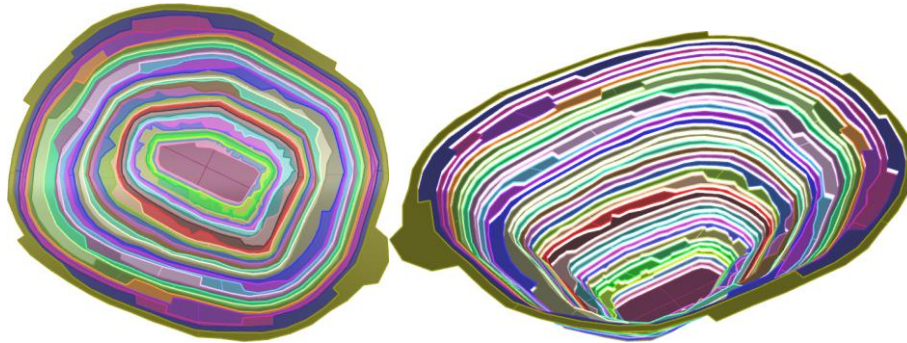


Figure 125: Top and perspective view of all the benches. The benches are obtained by trimming each surface with the border of the surface immediately below it, starting from the topmost surface representing the ground surface.

Creating the bench faces

26. To create the bench faces, we must extrude each bench boundary curve vertically thus creating a surface joining each bench with its immediate neighbor above it. In the Perspective view, zoom in close to the top bench, type **_Line** and create a line by connecting any point of the inner boundary of the top bench with the corresponding point on the outer boundary of the bench immediately below it (Figure 126, left). Select this line and **_Hide** it.
27. Since we know that the benches are at regular height intervals, instead of creating each bench face separately, we will extrude all the curves at once along the vertical line we just created. Type **_SelCrv** to select the boundary curves we created in an earlier step. Type **_Show** to show our hidden line segment (which we don't want in our curve selection) Select **Surface | Extrude Curve | Along Curve**, and click on the lower part of the vertical line segment. You should obtain vertical surfaces similar to those shown in Figure 126, right. Please note that by clicking on the lower part of the line segment, you indicate the end of the curve that represents the starting point of the extrusion.

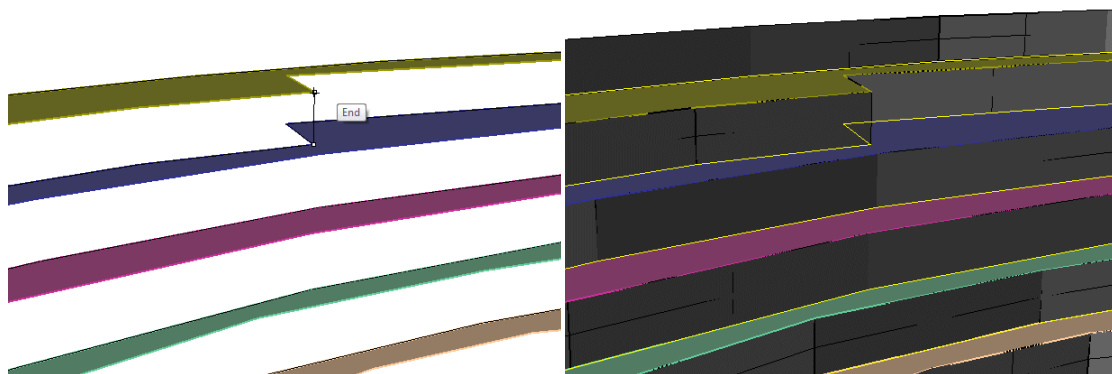


Figure 126: Creating a vertical line joining the outer boundary curve of a bench with the inner boundary curve of the bench immediately above it (left), and extruding all boundary curves along it (right)

28. In extruding all the curves, we may have inadvertently extruded the outer curve of the topmost bench. The topmost bench is at the same level as the ground surface and doesn't need to be extruded. Select the Polysurface resulting from the extrusion of the curves of the topmost bench and delete it (Figure 127).

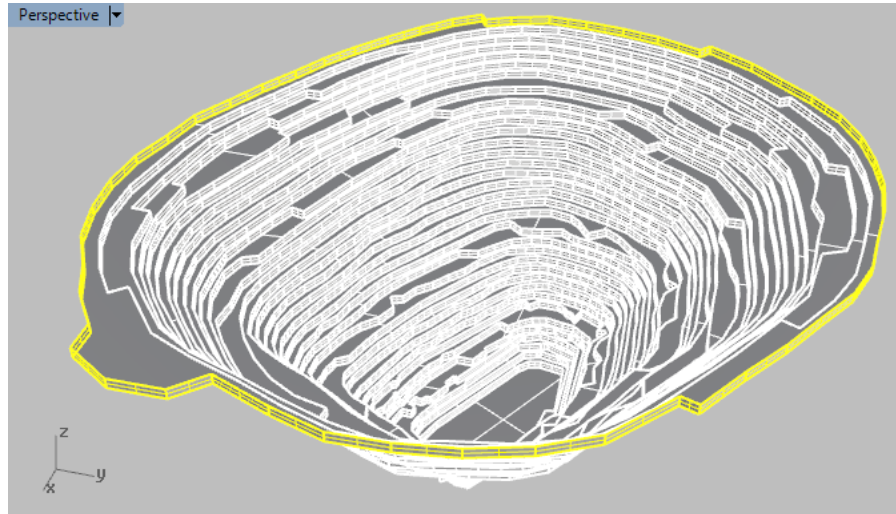


Figure 127: Topmost bench outline curve does not need to be extruded upward.

29. Type **_SelSrf** and **_SelPolySrf** to select all surfaces, then type **_Join** to join all the surfaces and polysurfaces into a single polysurface (Figure 128). Save your *Rhino* model as **Tut6C.3dm**.

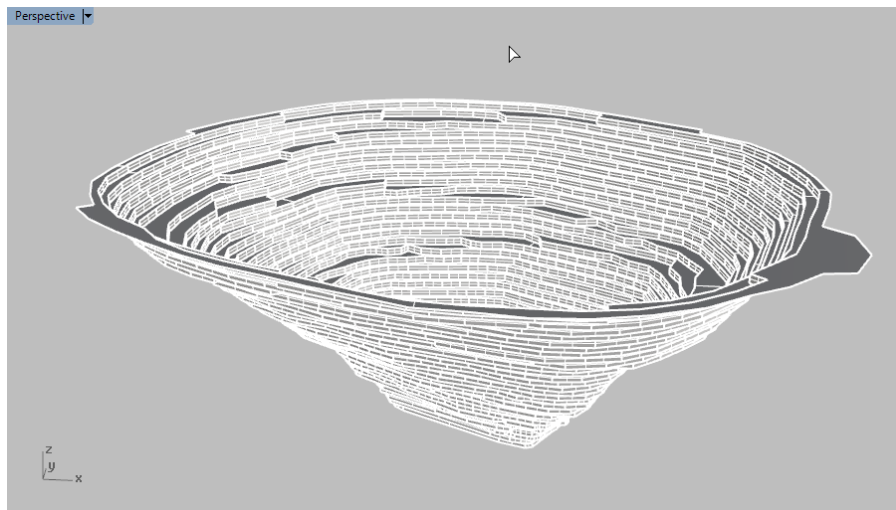


Figure 128: A single polysurface resulting from joining surfaces (benches) and polysurfaces (the vertical faces).

An alternative method of creating the benches and faces

Rhino offers a command called **_MeshPatch** which can be used to create an approximate surface defined by a number of curves. Using **_MeshPatch** directly with the starting DXF lines would result in a highly detailed “approximate” surface and would take a long time to mesh (because of all the fine detail in the original DXF). You can use **_MeshPatch** in a variety of ways. You could, for example, select all

boundary curves you created earlier (by tracing around the DXF contours) and typing **_MeshPatch**. This will result in a triangle mesh joining all the contour lines. An example is shown in Figure 129.

Instead of retracing the original DXF contour lines you could select detailed DXF lines you wish to use and type **_Divide**. Don't worry about the seam points, but select the option to divide the curves by length (try 50 m). You will end up with points spaced approximately 50 m apart on all the contour lines. Type **_SelPt** to select the points and type **_MeshPatch** to get a triangle mesh approximating the DXF contours.

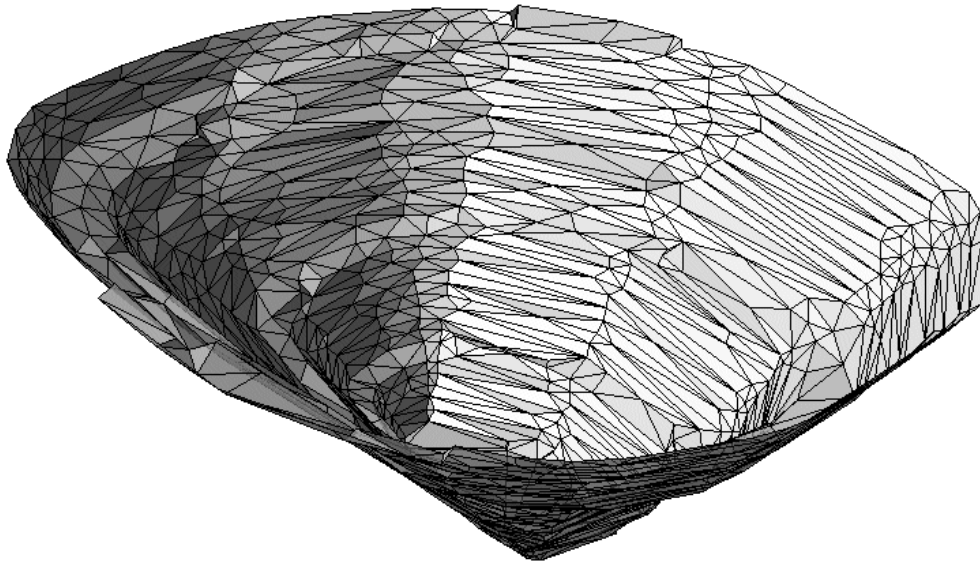


Figure 129: Approximate surface representation resulting from the use of the **_MeshPatch** command.

Please note that the resulting surface does not properly capture the critical features of the pit which we wish to preserve in this case (the bench and bench faces).

Creating an outer box

30. Restore **Tut6C.3dm** to go back to the original more accurate representation of benches and faces.
31. Type **_Box** and select the **Diagonal** option. Enter **-1500,-1500,-1500**, press **<ENTER>** and enter **1500,1500,0** followed by **<ENTER>** to create a box of width 3000, length 3000 and height 1500 centered around the x, y origin. In the **Top** view, select the box, **left-click** on it and drag it on top of the pit (Figure 130).

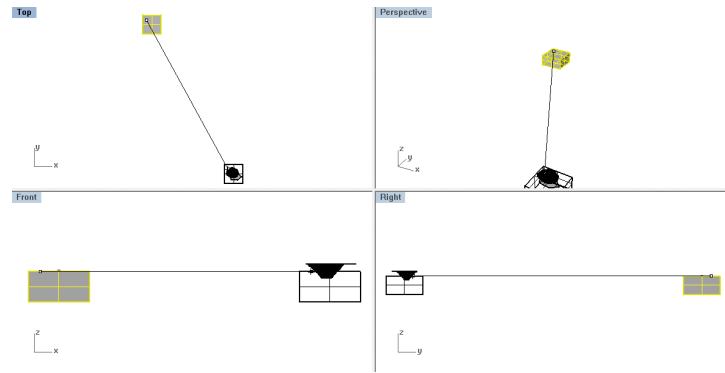


Figure 130: Dragging the box on top of the pit.

32. From the *Rhino* menu, **View|Zoom|Zoom Extents All** to fit all the objects in all the windows at once. In the Top view, move the box so the pit is approximately in the center. Click on the label of the **Front** view to set it as the current view, select **Transform|Align**, select the box first, then the pit and press **<ENTER>**. On the **command-line**, you will see several alignment options. Click on the **Top** option to align the top of the box with the top of the pit (Figure 131).

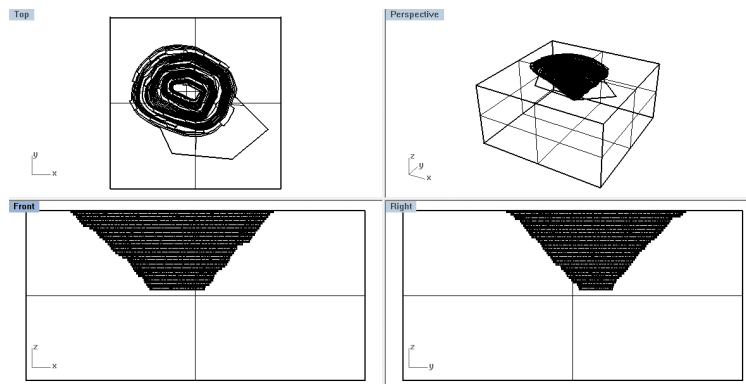


Figure 131: Box and pit after alignment.

33. Select the box and type **_Explode** to break the box up into 6 surfaces. Select the pit and the top surface of the box, type **_Invert** and then **_Hide**. Only the pit and the top surface of the box remain visible Figure 132.

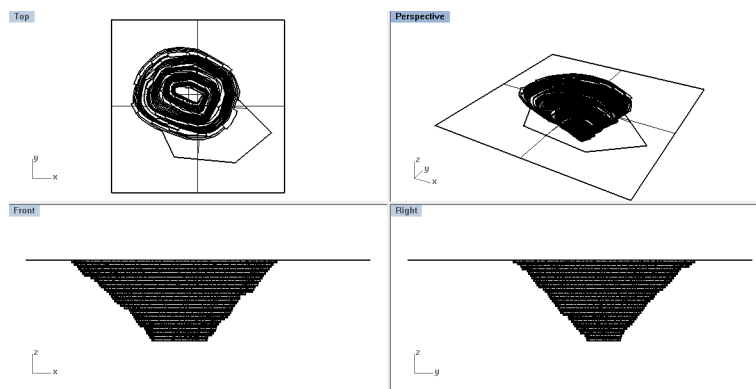


Figure 132: The pit and the surface representing the top of the box.

34. Double-click the label of the **Top** view to maximize it. Select everything (**Ctrl-A**) and **ColorizeObjects** to colorize the box top and the pit in two different colors (Figure 133, left). Select the pit and type **_DupBorder** to extract the curve representing the top surface boundary of the pit. **_Hide** the pit to just leave the top surface and the pit boundary (Figure 133, center). Select the curve and **_Trim**. Click inside the curve to trim off the interior of the curve (Figure 133, right).

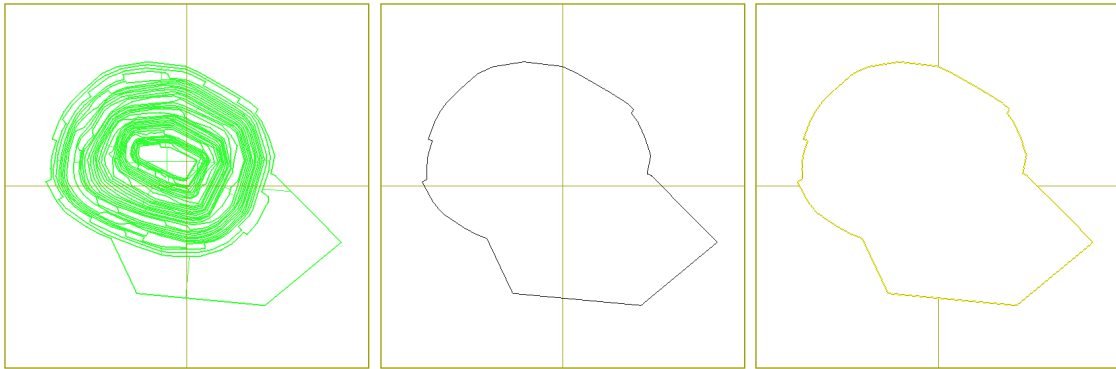


Figure 133: The pit and the top surface of the box (left), the pit boundary and the top of the box (center) and the result of trimming the top surface with the pit boundary (right)

35. **_Show** everything. **_SelCrv** and **_Delete** the selected curves. Select all remaining surfaces and polysurface and **_Join** them into a single polysurface. (Figure 134). Select **File|Save** and save the current model to file **Tut6D.3dm**.

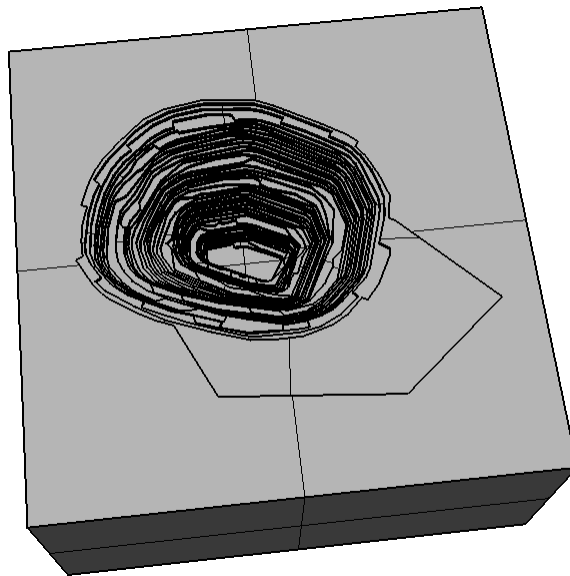


Figure 134: Rhino model of the entire pit.

Creating a surface mesh

36. Select the polysurface you created and type **_Mesh**. Use the Simple controls and drag the slider all the way to the right. Click on **OK** to confirm, and while the polysurface is selected, press the **<DELETE>** key to delete the polysurface.
37. Check your mesh by typing **_Check** or **_MeshRepair**.

Creating a *FLAC3D* model

38. Select your *Rhino* mesh, type **_GSurf** (for the Griddle surface remesher) and remesh this surface with **Mode:QuadDom**, **MinEdgeLength 20** (the bench height). Leave the other values as defaults and press Enter. You should obtain a mesh similar to that shown in Figure 135.
39. Select the quad dominant surface mesh, type **_GVol** (for the Griddle volume mesher) and select **Mode:ConHexDom** and **FLAC3D** output format (remember to do a **_SetWorkingDirectory** if *Rhino* outputs the mesh to the incorrect directory). The resulting grid GVol.f3grid can be imported into *FLAC3D*.

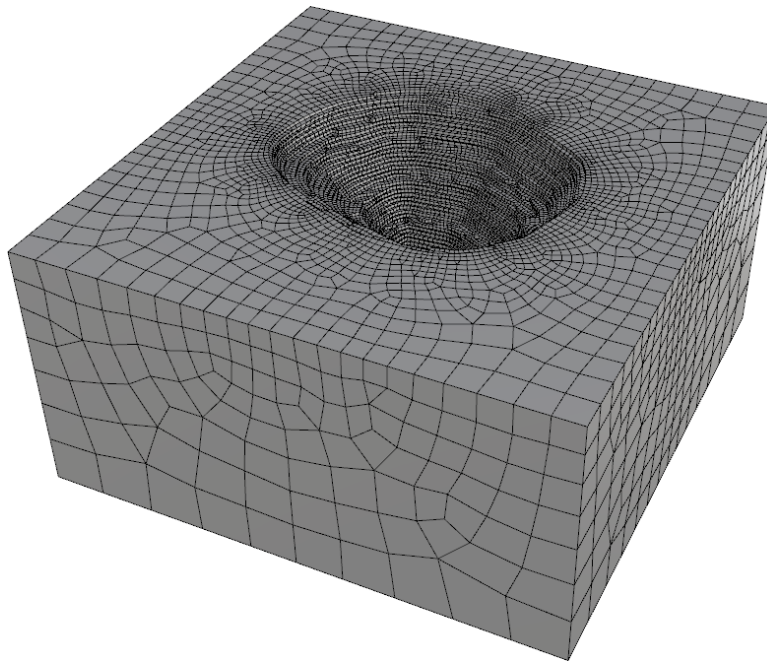


Figure 135: Quad dominant surface mesh produced by *Griddle*.

Creating a *3DEC* model

40. We will now create a tet block grid for *3DEC*. Open Tut6D.3dm and create a surface mesh using the *Rhino* **_Mesh** command as you did for *FLAC3D* in the steps above.
41. Select your *Rhino* mesh, type **_GSurf** (for the Griddle surface remesher) and remesh this surface with **Mode:Tri**, **MinEdgeLength 20** (the bench height) and **MaxGradation 2**. Leave the other values as defaults and press Enter. You should obtain a mesh similar to that shown in Figure 136.

42. Select the the triangle surface mesh, type **GVol** (for the Griddle volume mesher) and select **Mode:Tet** and **3DEC** output format (remember to do a **_SetWorkingDirectory** if *Rhino* outputs the mesh to the incorrect directory). The resulting block model file GVol.3ddat can be called into *3DEC*.

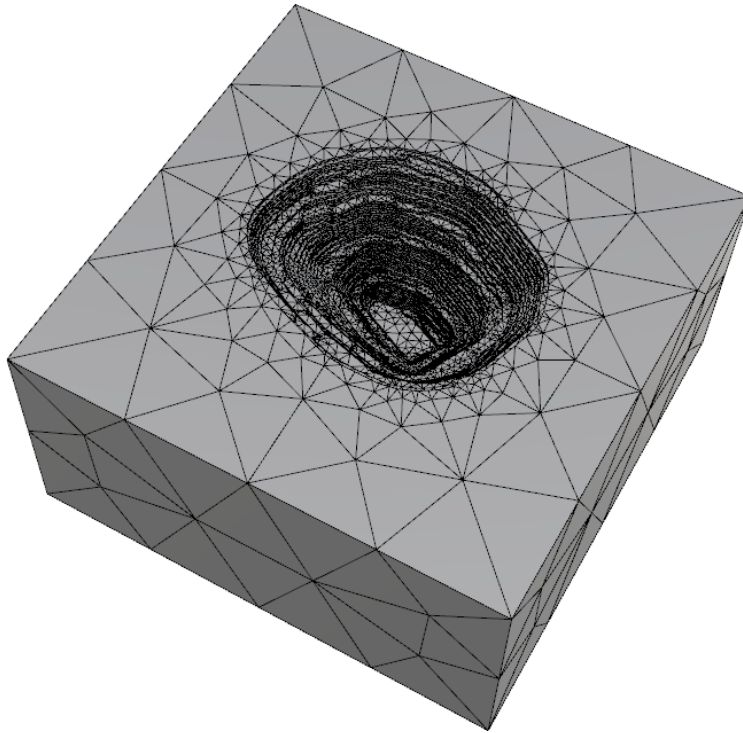


Figure 136: Triangle surface mesh generated by *Griddle*.

END OF TUTORIAL 6

Tutorial 7: Open Pit with Intermittent Faults (*Griddle*)

In this tutorial we will create *FLAC3D* and *3DEC* models of an open pit with two faults. The two faults that intersect each other do not completely divide the modeling domain into separate pieces.

Reading a DXF file containing pit contours, topography and faults

1. Start *Rhino* and select the **Large Object: Meters** template and then **_SetWorkingDirectory** to navigate to the folder in which you will be working.
2. **File|Import open_pit.dxf**. Use **Model units: Meters**, **Layout units: millimeters** and **Mesh Precision: Double Precision** (Figure 137).

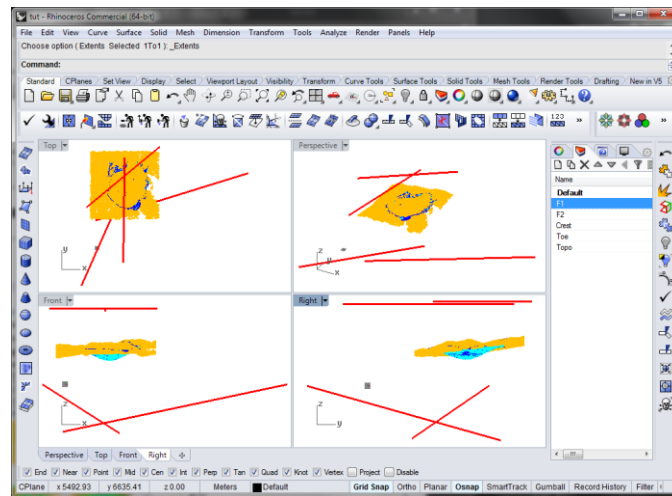


Figure 137: Rhino views after importing open_pit.dxf.

3. Type **_Layer** to open the Layers Panel. Select all layers and click on one lit light bulb (in the Layers Panel) to **hide all** layers. Select the **Toe** and **Crest** layers, click on the light bulb (in the Layers Panel) to turn these 2 layers on (Figure 138).

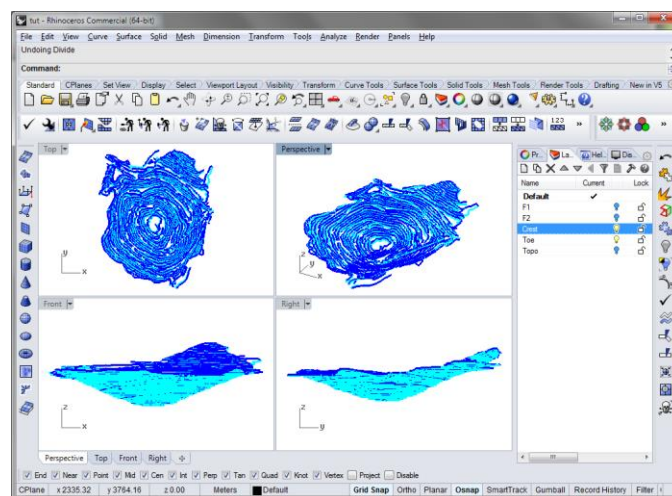


Figure 138: Crest and Toe layers turned on.

Building a mesh representing the pit

1. Double-click the **Perspective** label to maximize the Perspective view. **<CTRL> A** to select all the visible contour lines. **Curve|Point Object|Divide Curve by|Length of Segments**, followed by **<ENTER>**, and set the **length of segments** to 5 (Figure 139).

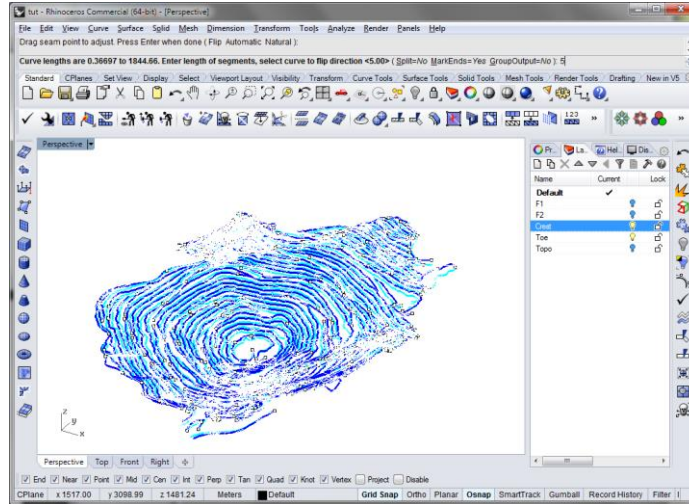


Figure 139: Dividing curves into equal length segments.

2. Hit **<ENTER>** to complete the creation of points every 5 meters on the selected curves (Figure 140).

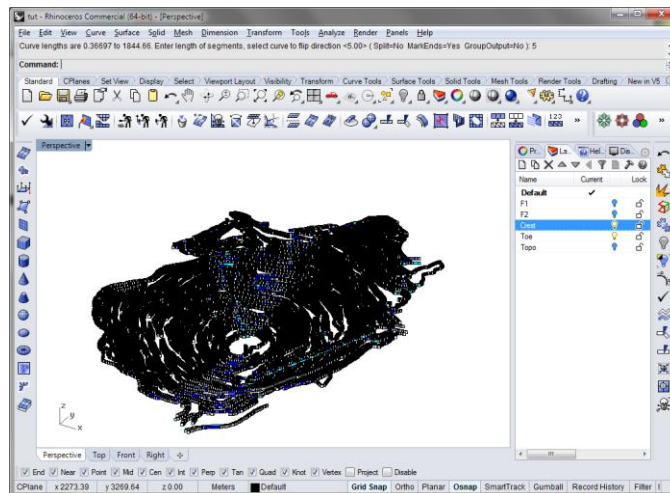


Figure 140: Points spaced 5 m apart.

- **Edit|Select Objects|Points** to select all points, followed by **_MeshPatch**. Press **<ENTER>** to accept the default options, and press **<ENTER>**, again, to obtain a triangular mesh resulting from the points. **Edit|Select Objects|Points**, followed by **<DELETE>** to get rid of the points which are no longer needed. **Hide** the Crest and Toe layers by turning their layer light bulbs off.
3. Type **_Shade** and, in the command options, select Display Mode **Shaded** (Figure 141).

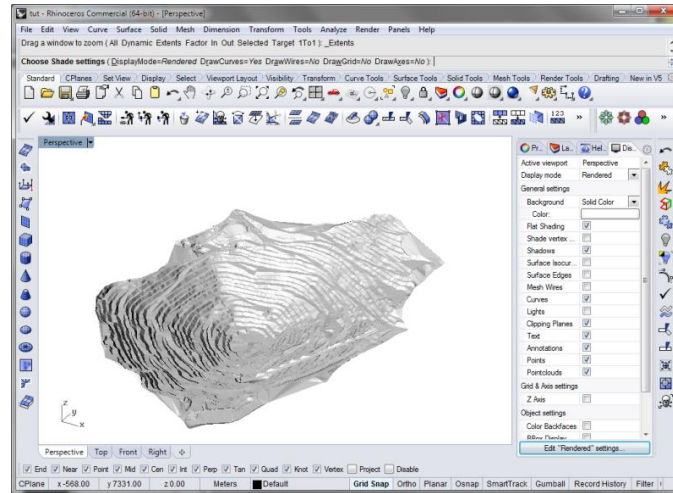


Figure 141: Shaded perspective view.

4. Sometimes smooth shading can make the benches difficult to see. To avoid this, turn flat shading on. First, make sure that **Panels|Display** is checked. In the **Display** Panel, check **Flat Shading** to see the image shown above
5. Click on the **Shaded Viewport** icon to see the mesh with the grid lines (Figure 142).

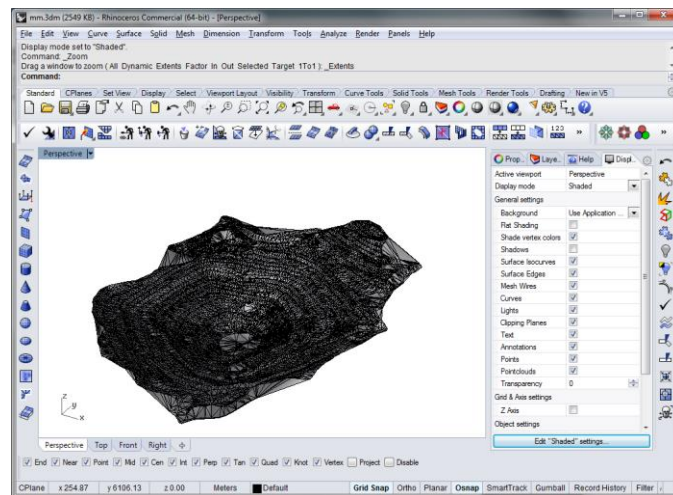


Figure 142: Mesh with grid lines.

6. Again, if flat shading it not on, turn it on with command **_FlatShade** (or with the icons) to better see the details of the mesh. Zoom in to see the image below (Figure 143).

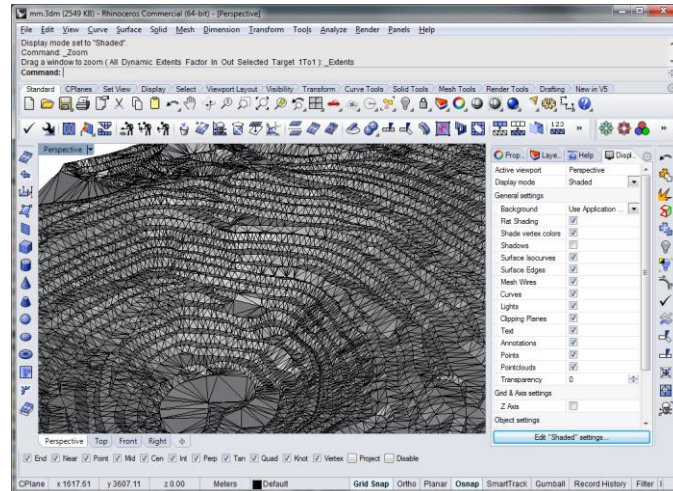


Figure 143: Zoomed in perspective view of the pit.

7. Select the mesh, then **_ChangeLayer** which opens the **Layer for Objects** dialog box (Figure 144).

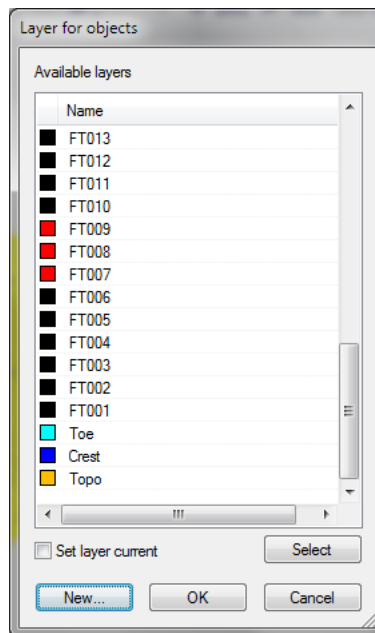


Figure 144: Layer for Objects dialog box.

8. Click **New** which opens the **New Layer** dialog box. Enter **PitMesh** for the name of the new layer and click **OK**. Click **OK** to close the **Layer for Objects** dialog box. In the **Layer** Panel, **hide PitMesh** by turning off its light bulb.

Building a mesh representing the initial topography

1. In the **Layers** Panel, turn off all layers except **Topo**. Click **Zoom Extents** (Figure 145).

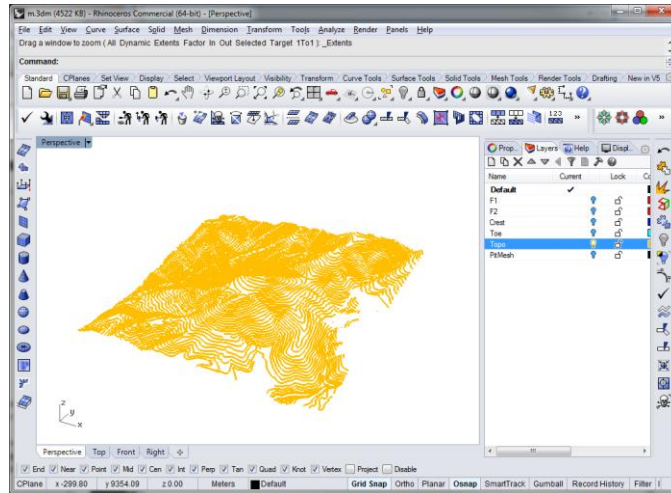


Figure 145: Only the Topo layer is visible.

2. Select all the curves belonging to the **Topo** layer. **Curve|Point Object|Divide Curve by|Length of Segments**, followed by **<ENTER>**. Set the **length of segments** to **5** and hit **<ENTER>**. This results in the creation of points, 5 m apart, on the selected curves.
3. **Edit|Select Objects|Points**, followed by **_MeshPatch**. **<ENTER>** to accept the command defaults, followed by another **<ENTER>** to build a mesh representing the topography. **Edit|Select Objects|Points**, followed by **<DELETE>** to get rid of the point. Hide **Topo** in the **Layers** Panel. **Shade** icon and select **Display Mode=Shaded** (Figure 146).

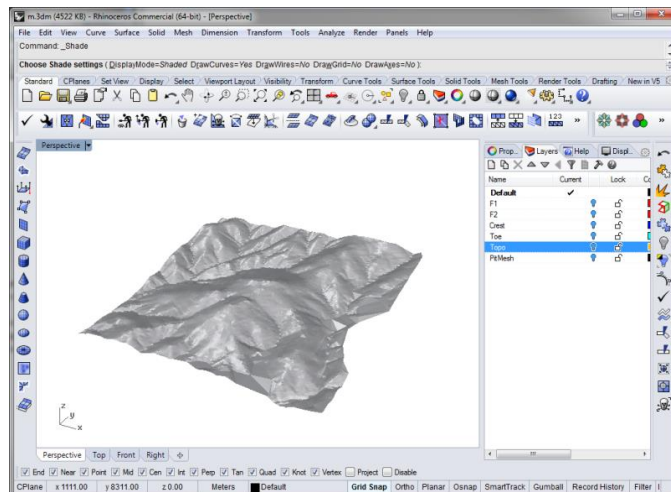


Figure 146: Shaded perspective view of mesh surface on Default layer.

4. Select the mesh, then **_ChangeLayer** which opens the **Layer for Objects** dialog box.
5. Click on **New** which opens the **New Layer** dialog box. Enter **TopoMesh** for the name of the new layer and click **OK**. Click **OK** to close the **Layer for Objects** dialog box. In the **Layer** Panel, **hide TopoMesh**.

Combining the pit and the topography mesh

1. **Hide** all layers. In the **Layers** Panel, turn on the **PitMesh** layer. Select the mesh and, **Curve|Curve from Objects|Duplicate Border**. This is the boundary of the pit (Figure 147).

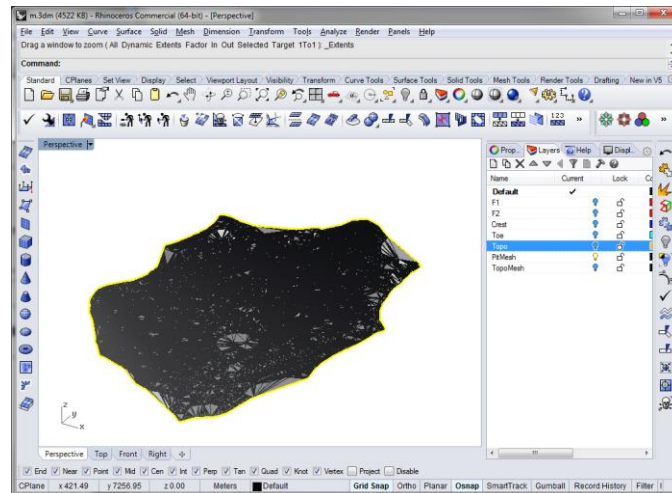


Figure 147: Duplicating the pit border.

2. **Hide PitMesh** in the Layers panel, leaving only the boundary **curve** visible. Turn on the **TopoMesh** layer and select the **Top view**. **Edit|Select Objects|Curves** to highlight the boundary curve (Figure 148).

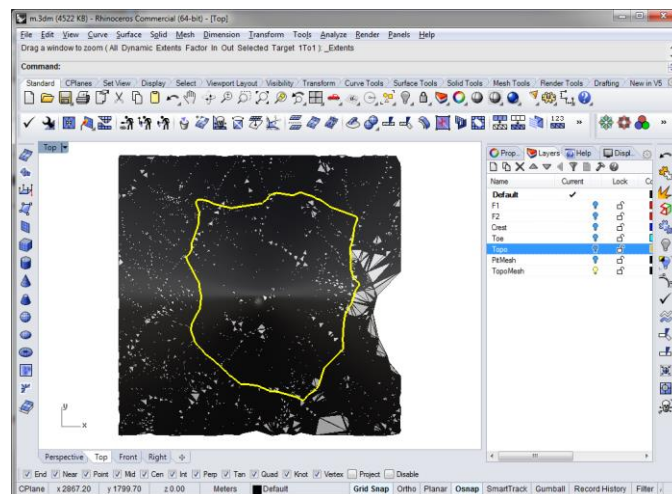


Figure 148: TopoMesh and pit boundary curve.

3. **Mesh|Mesh Edit Tools|Mesh Trim** and **click inside** the curve to trim off the portion of the mesh that is inside the curve. Don't forget **<ENTER>** to complete the Trim operation. **Edit|Select Objects|Curves** followed by **<DELETE>** to get rid of the boundary curve (Figure 149).

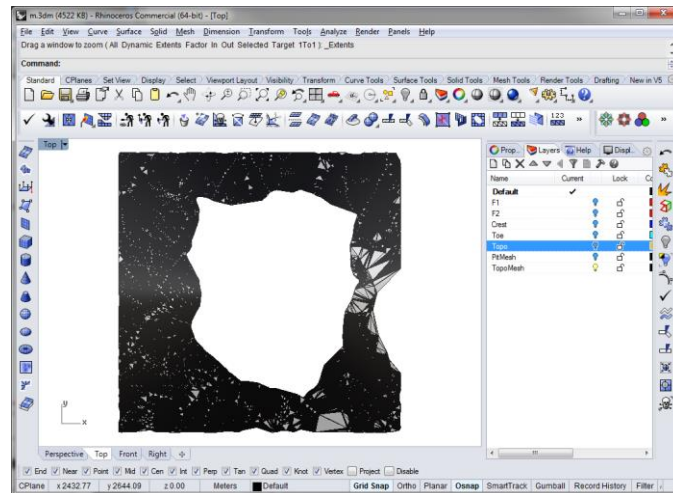


Figure 149: Trimming the TopoMesh.

4. Unhide the **PitMesh** layer so **both** meshes are visible. Select both meshes and **_ColorizeObjects**. Click **Shade** to see the view below (Figure 150).

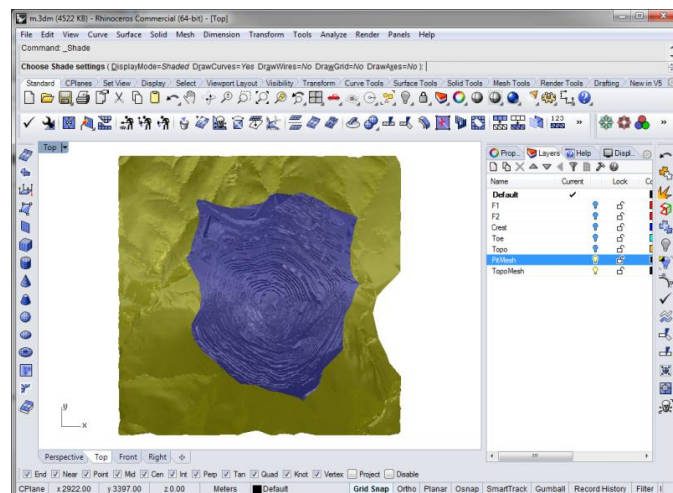


Figure 150: PitMesh and TopoMesh shown shaded in Top view.

Building a single surface extending over the pit and the topography

1. In **Top** view, select **Surface|Drape**. Click somewhere at the **top, left** of the mesh, then **drag** the mouse towards **lower left** corner of the window drawing a **rectangle** as large as possible while still contained **within** the mesh (Figure 151).

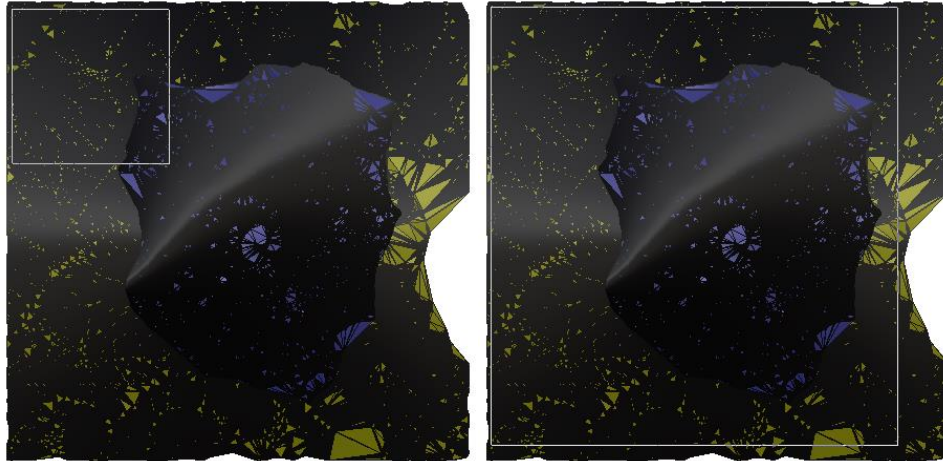


Figure 151: Draping the surface.

2. **Release** the mouse **button** to complete the **Drape**. **Hide PitMesh** and **TopoMesh** in the Layers Panel, select the **Perspective View** and click on the **Shaded Viewport** icon (Figure 152). If you notice the surface dipping below realistic heights at certain points, this may be due to spilling through the gap between **PitMesh** and **TopoMesh**. Try draping again.

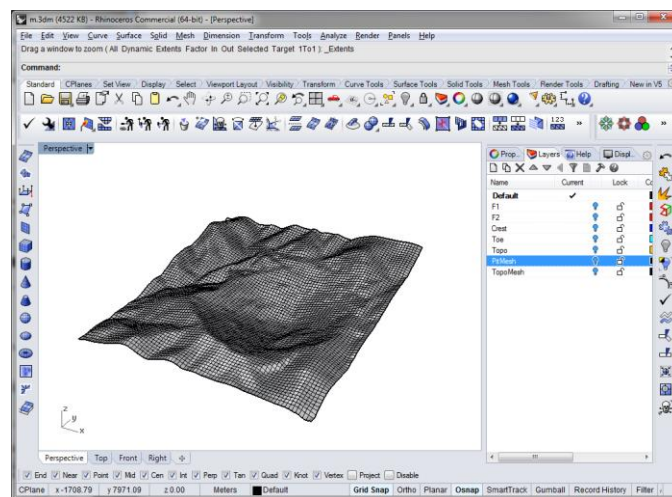


Figure 152: Draped surface.

Trimming of the topography and adding of vertical walls

1. Go back to **Top View**. **Curve|Rectangle|Corner to Corner**. Enter **-120,1310** for the first and **2100,3750** for the second corner. Note, no blanks. **Edit|Select Objects|Curves** to highlight the rectangle. **Edit|Trim** and click on the surface **outside** of the rectangle (don't forget **<ENTER>** to finish Trim), then **delete** the rectangle itself (Figure 153).

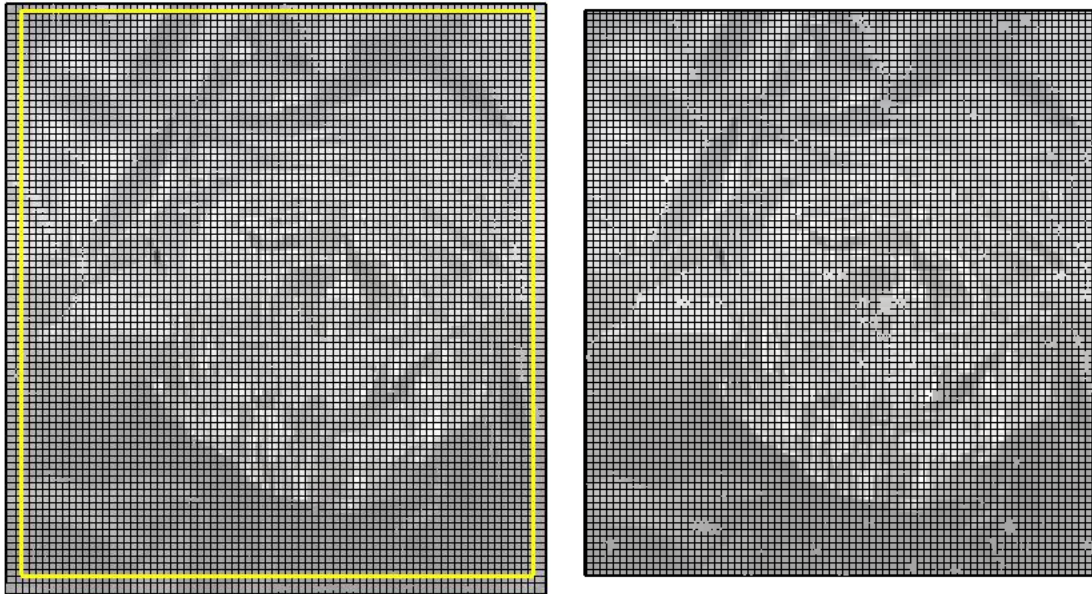


Figure 153: Trimming the draped surface.

2. Select the **Perspective** view. Select the **surface**, then **Curve | Curve from Objects | Duplicate Border**. You can also **right-click** in the **command** area and find it there, in your recent history.
3. While the border **curve** is selected, **zoom away** to give yourself more room, then **Surface | Extrude Curve | Straight (Option Both sides=No)** and **drag** the curve **down** as much as possible, then click the left mouse button to stop the extrusion right there. You have created the vertical walls of the model (Figure 154).

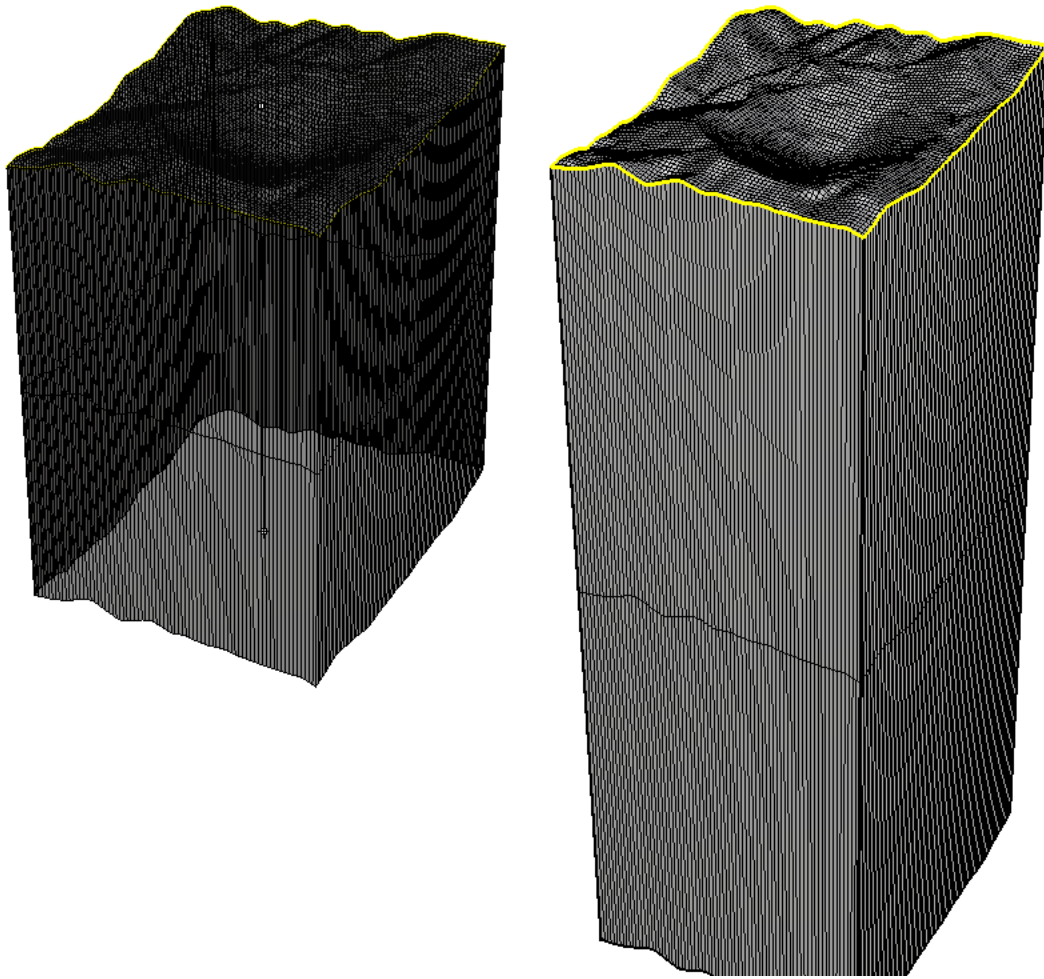


Figure 154: Extruding the surface.

Building a closed volume

1. **Delete** the TopoMesh boundary **curve**. Select the **Front View**, and click on the **Zoom Extents**. **Curve|Line|Single Line**, enter **-400,-500** for the first point of the line. Then, **hold** down **<SHIFT>**, and click somewhere far right of the window to create a perfectly **horizontal** line (Figure 155).

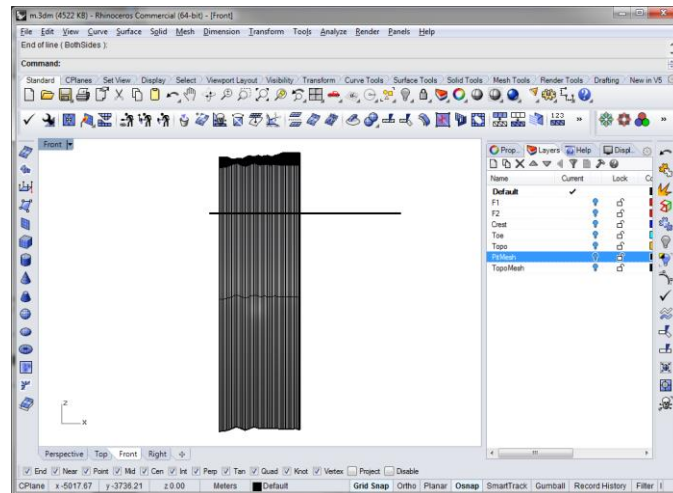


Figure 155: Creating a horizontal trimming line.

2. Select the horizontal line, **Edit|Trim** and click **below** the line to trim away the lower part of the vertical **walls**. **<ENTER>** to **finish** trimming, then **<DELETE>** the horizontal line.
3. The vertical walls are now cut horizontally at $z = -500$ m. Select the **top** (topography) surface and the vertical **walls**, then **Edit|Join** to create a **single Polysurface** open at the bottom. Select the **Polysurface**, then **Solid|Cap Planar Holes** to create a closed Polysurface also called a **Solid** (Figure 156).

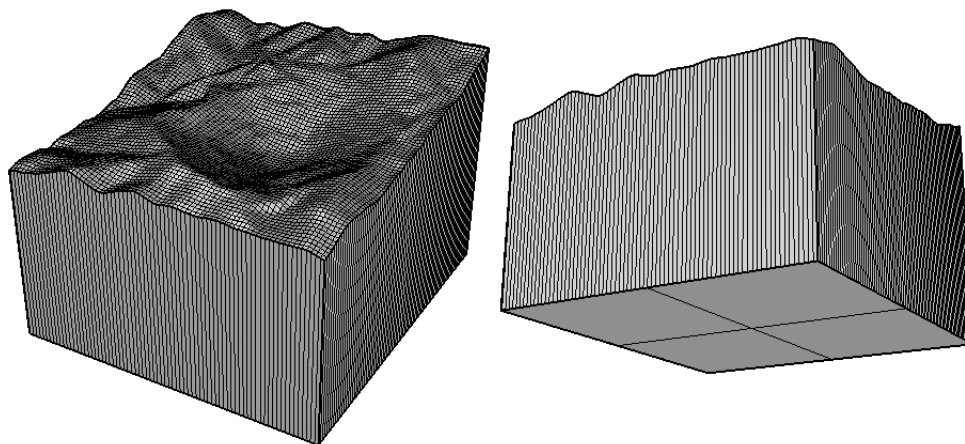


Figure 156: Capping the bottom of the model.

4. Select the solid, then **_Changelayer**. In **Layer for objects** click on **New**. In **New Layer**, type in **Solid**, then click **OK**. Click **OK** to close **Layer for objects** box.
5. **File|Save As**, and save your model as **tut7A.3dm**.

Adding Faults

1. Turn off the **Solid** layer in the Layers Panel and turn Layer **F1** on. Double-click the label of the Perspective view to return to a 4-view mode to see two straight lines used to define fault F1 (Figure 157).

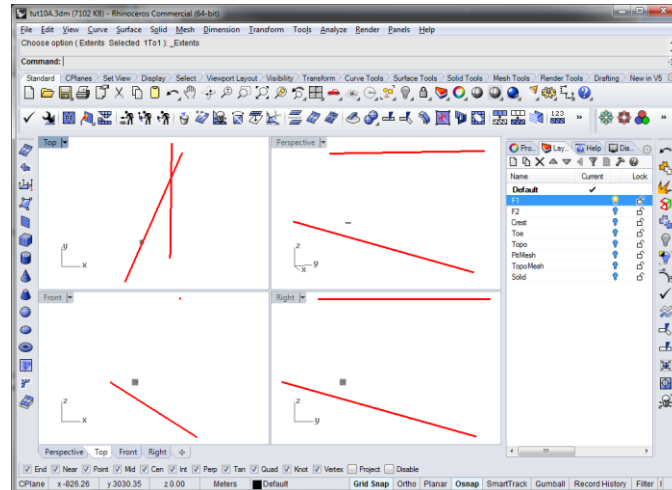


Figure 157: Fault F1 lines.

1. In the menus use, **Surface|Loft**. In any of the views click on **one curve**, then the **next**, followed by **<ENTER>**. This opens the **Loft Options** (Figure 158).

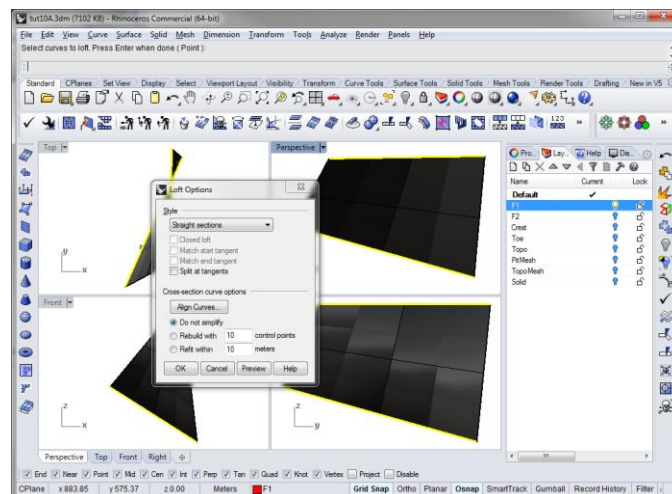


Figure 158: Lofting the fault lines.

2. Select **Straight Section** for the **Style** and **Do not simplify** for Cross-section curve options. If your lofted fault plane has a twist in it, select **Align Curves** and follow the prompt. Click **OK**. **Edit|Select Objects|Curves**, then **<DELETE>** the two lines. Select the surface, then **ChangeLayer** and place the selected surface in layer **F1** (Figure 159).

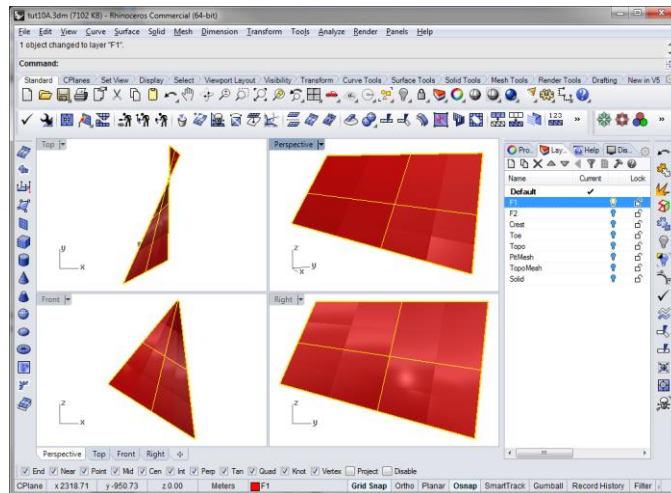


Figure 159: Lofted surface placed into layer F1.

3. Turn off F1 and turn F2 on (Figure 160).

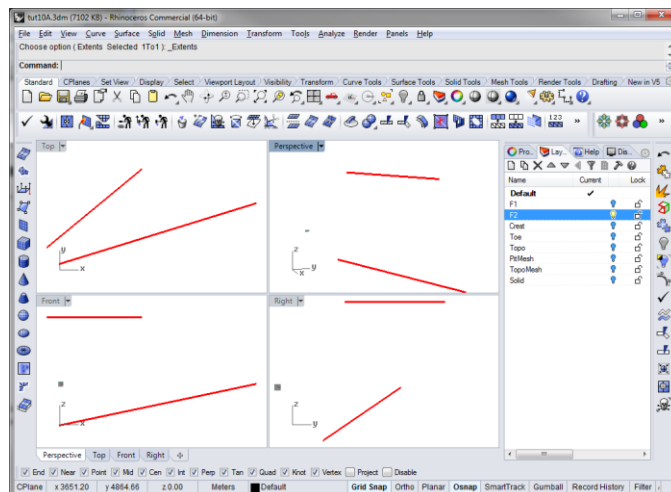


Figure 160: Fault F2 lines.

4. **Surface|Loft.** In any of the views click on **one curve**, then the **next**, followed by **<ENTER>**. This opens the **Loft Options**.
5. Select **Straight Section** for the **Style** and **Do not simplify** for **Cross-section curve options**, then click **OK**. **Edit|Select Objects|Curves**, then **<DELETE>** the two lines. Select the **Surface**, **ChangeLayer** and place the selected surface in layer **F2** (Figure 161).

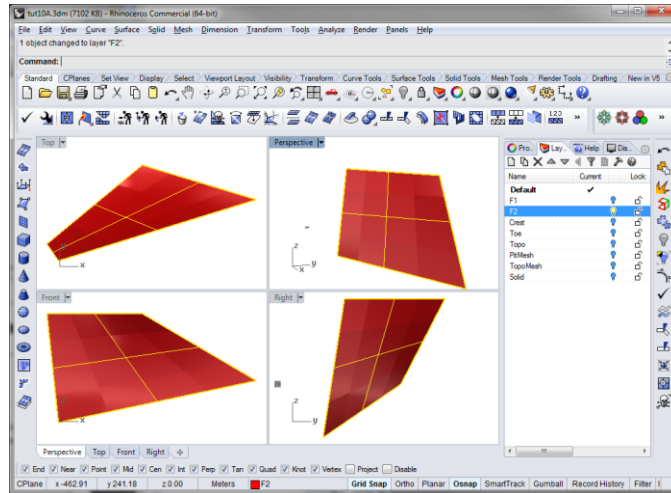


Figure 161: Lofted surface placed into layer F2.

Assembling the faults and the solid into a single non-manifold polysurface

We will use *Rhino's* non-manifold merge function to assist us in trimming the portions of the faults outside the solid.

1. Turn **Solid, F1** and **F2** on in the Layers Panel. Double-click the label of the Perspective view to maximize it. Select the solid and the fault surfaces and copy them into a new layer called Merged (similar steps to what you did previously when creating new layers). Make the Merged layer the only active layer (Figure 162).

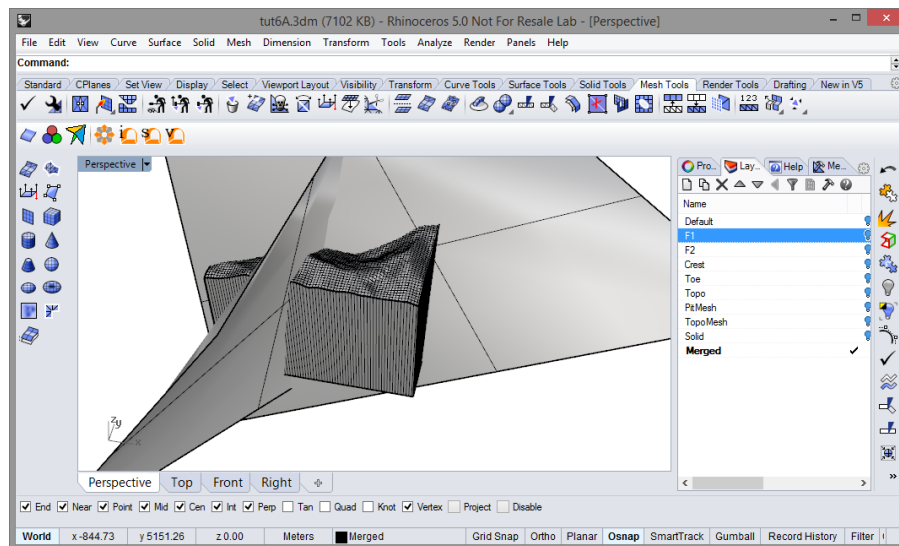


Figure 162: Faults and pit geometry in layer Merged.

2. Select the 2 surfaces representing the faults and the solid and **_NonManifoldMerge**. This creates a single non-manifold polysurface. Type Check to verify this polysurface (this polysurface is too complex for Rhino's engine to handle as a non-manifold object). If we extend the cuts in the surfaces, then Rhino can handle the non-manifold meshing operations we will eventually do.

3. Select the non-manifold Polysurface and type **_testExtendSlits** in the command line. This will yield a valid non-manifold Polysurface. **_Check** to verify that it is indeed valid (Figure 163).

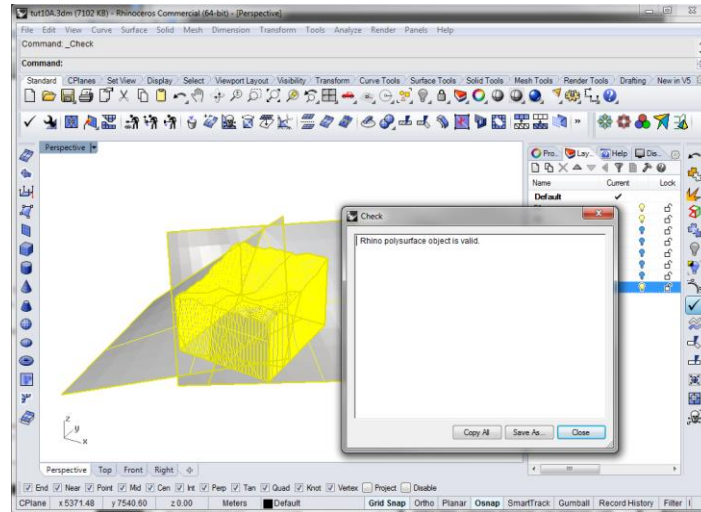


Figure 163: Check after tetExtendSlits is run.

4. **_ExtractSrf, (option Copy = No)**, and click on any portions of the faults sticking outside the computational domain, followed by <ENTER> to complete the surface extraction (Figure 164).

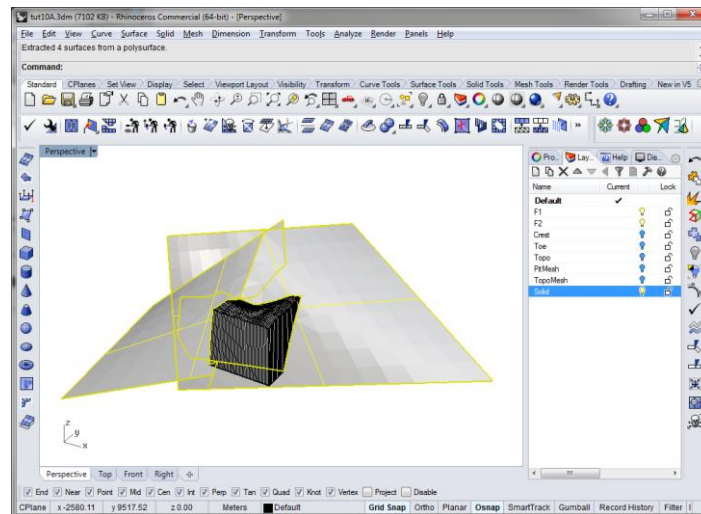


Figure 164: Extracting the exterior portions of the faults.

5. Delete the portions of surfaces sticking outside the computational domain (these are the surfaces you just extracted above).
6. You now have a single non-manifold polysurface describing the entire computational domain and the faults (Figure 165).

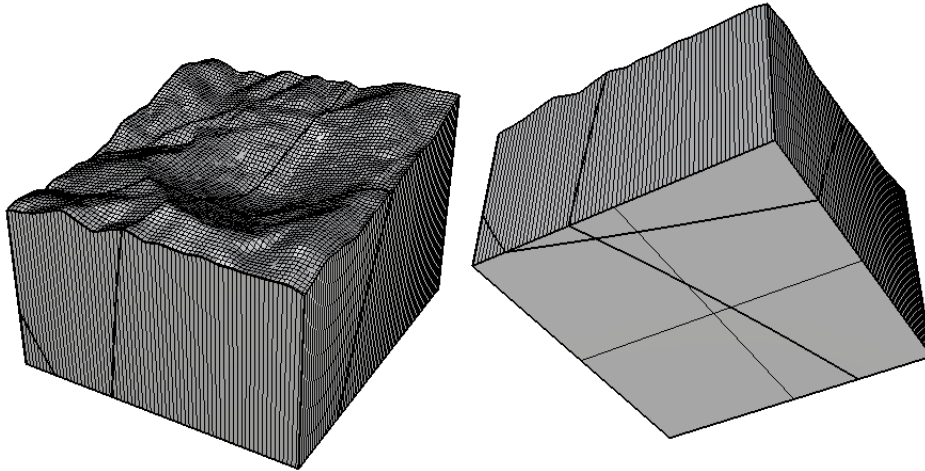


Figure 165: Two views of single polysurface representing pit and cross-cutting faults.

7. File|Save As, and save your model as tut7B.3dm.
8. Select the polysurface and **_Mesh** with Detailed Mesh parameters shown below (Figure 166).

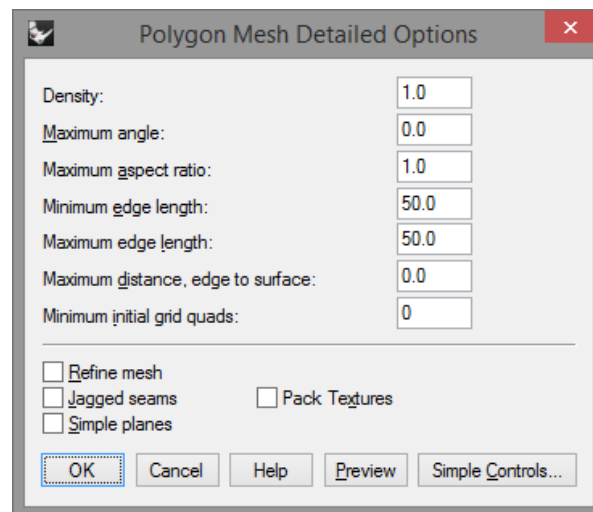


Figure 166: Meshing options.

9. **_Check** this mesh to make sure it is good. Move this mesh to a new layer called **Meshes**.
10. File|Save As, and save your model as **tut7C.3dm**.

Creating a hex-dominant *FLAC3D* mesh

1. Select the mesh in layer Meshes and type **_GSurf**. Use **Mode:QuadDom**, **MinEdgeLength 50**, **MaxEdgeLength 50** to generate a surface mesh similar to that shown below (Figure 167).

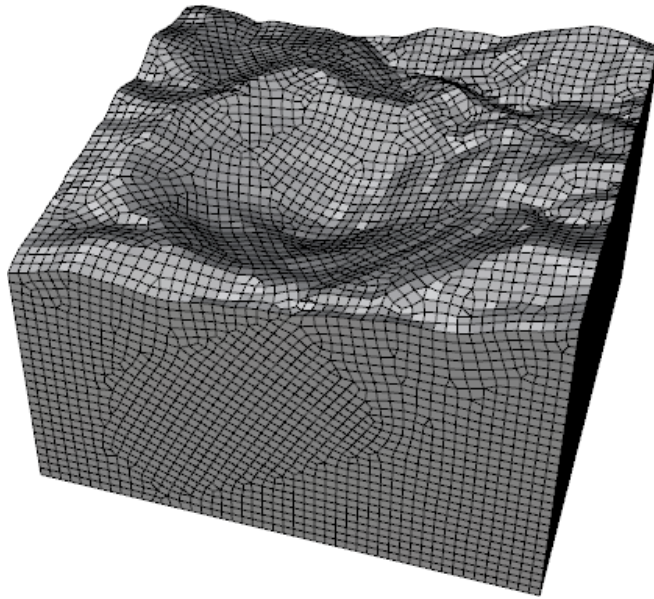


Figure 167: Quad dominant surface mesh of pit and faults.

2. **_Check** the mesh to make sure it is valid. Select the mesh and type **GSurf** with **Mode:ConHexDom** and *FLAC3D* output.
3. Import *GVol.f3grid* into *FLAC3D*. Your mesh should appear similar to that shown below (Figure 168, left: zones, right: face groups generated by *Griddle*).

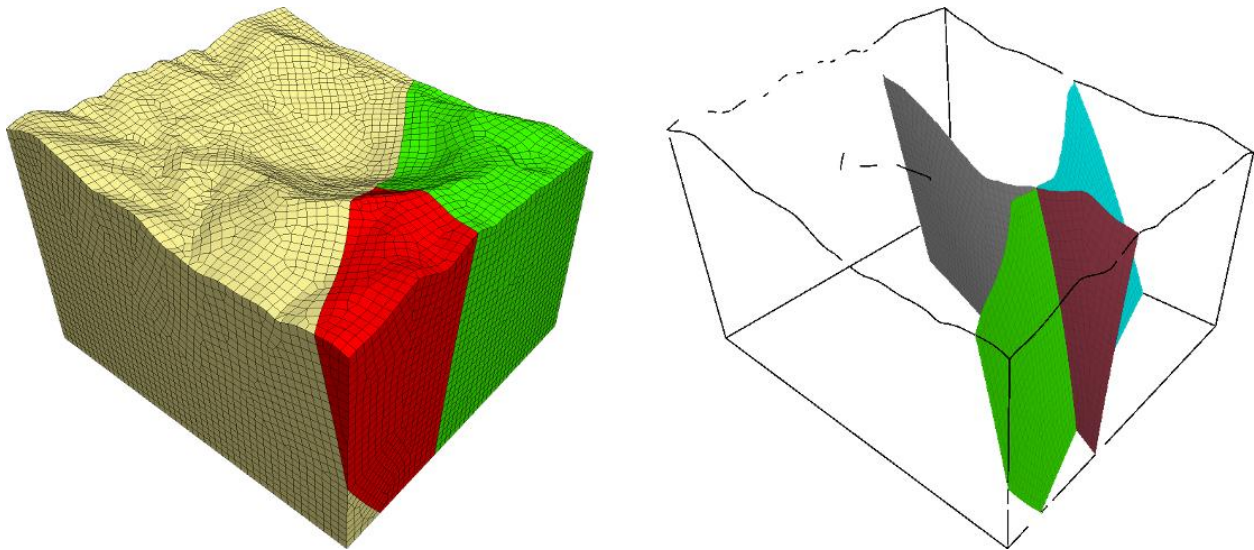


Figure 168: *Griddle* generated mesh read into *FLAC3D*.

Creating a tet block model for 3DEC

1. Open tut7C.3dm in *Rhino*.
2. Select the mesh in layer Meshes and type **_GSurf**. Use **Mode:Tri**, **MinEdgeLength 50**, **MaxEdgeLength 500**, **MaxGradation 1**.
3. Select the mesh and type **_GVol** with **Mode:Tet** and **3DEC** output format.
4. Call GVOL3ddat into **3DEC** (Figure 169). Your rigid block model should be similar to that shown below left. The joint sets are shown below right. Note, **3DEC** output consists of a data file with **3DEC POLYHEDRON** commands which create rigid blocks in **3DEC** (these are not deformable zones). If you wish to make the blocks deformable in **3DEC** you need to use the **3DEC GENERATE** command to populate the rigid blocks with zones.

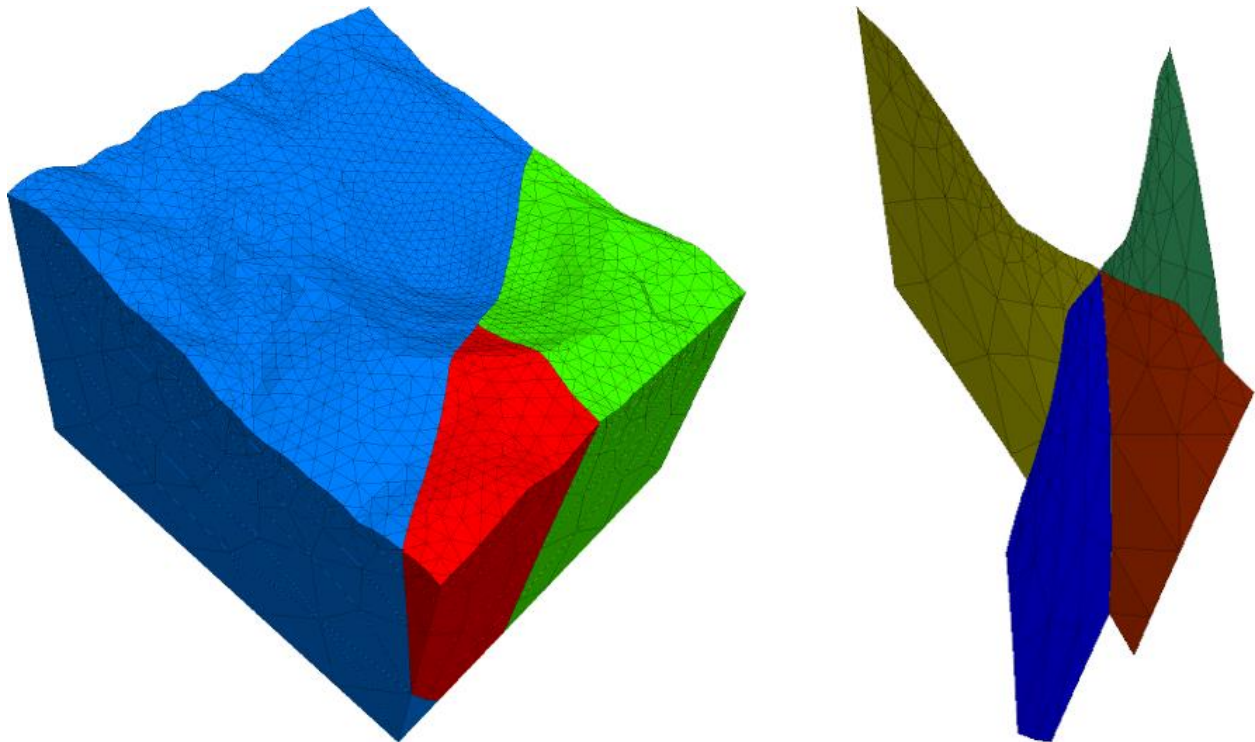


Figure 169: Griddle generated 3DEC block model.

END OF TUTORIAL 7

CAD Representation of surfaces and volumes

Data transfer between different CAD tools is achieved through files. Files may differ by the entities they contain and by their format. For instance, an AutoCAD DXF and a VRML file may both describe a mesh of triangles.

General Guidelines

CAD data is generally either a surface/solid/line (geometrical) model or faceted description of surfaces (discretized polygons). Solid/Surface/line models are mathematically exact definitions of the geometries they describe whereas faceted data represents solid/surface models that have been discretized into a collection of points and polygons which approximate the “exact” mathematical surface.

Surfaces/Solids must be imported as IGES or STEP files into *Rhino*, properly intersected, and cleaned-up prior to meshing with *Rhino*, *Griddle* or *BlockRanger*. For example, if you had a large model with a 5000 m x 5000 m surface with some mine workings (e.g., a 3 m cubic room) that was 2 mm above the surface, it might appear from looking at the *Rhino* viewport that these two entities actually touch (even when you zoom in). However, they likely will not be intersected when doing a Boolean operation or a **_NonManifoldMerge** operation and your resulting surface mesh may not be what you expected. You need to trim and merge the various entities so they conform to each other.

Faceted (triangulated) data may be imported in the STL, VRML, DXF or 3DS formats into *Rhino* for further processing. In *Rhino*, faceted data will be represented as a mesh. The mesh must be first cleaned-up, that is checked for quality (no free edges, degenerate or duplicated faces) and de-featured prior to generating a volume mesh with *Griddle*.

Solids & Surfaces

Solids and surfaces may be imported from the IGES, STEP or ACIS formats into *Rhino*. In *Rhino* the data will be represented as polysurfaces which should be checked for naked edges and other anomalies. A **_NonManifoldMerge** and subsequent clean-up of extraneous pieces (e.g., dangling surfaces outside your model domain) should be done prior to meshing them.

Manifold and non-manifold Surfaces

Solids are assemblies of surfaces, called Polysurfaces in *Rhino*, that have a clearly-defined interior and exterior. Surfaces defining solids are called manifold surfaces. In this context, manifold means that a surface edge is shared at most by two surfaces. In contrast, consider a solid cut in half. The resulting solid has now one interior surface. It is closed but doesn't have a clear interior. To be exact, it has two interiors. Such surfaces are called non-manifold surfaces.

Wireframes

Wireframes may be imported as IGES, STEP, VRML 2.0, DXF or DWG files into *Rhino*. In *Rhino*, the data will appear as lines. Lines should be used as a guides to create closed polysurfaces. It is good practice not to use imported curves directly but to retrace them by creating Polylines (using points on the curves). Often, curves produced by AutoCAD contain many degenerate line segments which, if used

directly in the construction of a surface, may result in invalid surfaces. Use the retraced Polylines or curves to create surfaces. The resulting closed polysurfaces can be meshed.

Points

In *Rhino*, points should be used as a guide to create lines and closed polysurfaces.

Faceted surfaces: polygonal surfaces

Polygonal surfaces may be imported as IGES, STEP, VRML 2.0, 3DS, STL, DXF or DWG files into *Rhino*. In *Rhino*, polygons should be split into triangular meshes. Meshes should be closed and checked for anomalies (degenerate or duplicate elements) and de-featured prior to meshing with *Griddle*.

Faceted surfaces: triangular surfaces

Triangular surfaces may be imported as IGES, STEP, VRML 2.0, 3DS, STL, DXF or DWG files into *Rhino*. In *Rhino*, meshes should be closed and checked for anomalies (degenerate or duplicate elements) and de-featured prior to meshing with *Griddle*.

Tips and tricks of the trade

The SetWorkingDirectory command in Rhino 5

When you start *Rhino* by double-clicking a file located in a folder, generally *Rhino* considers that folder to be the working directory, so when you click on *Griddle* or *BlockRanger* icons in *Rhino*, those grid generation programs will output files to that directory. In certain cases, *Rhino* may not know where the working directory is. In *Rhino*, the command **_SetWorkingDirectory** may be used to tell *Rhino* (thus *Griddle* and *BlockRanger*) where to read and write its files.

Shimmering triangles and off-center models

Have you noticed that in certain situations, as you examine a *Rhino* (or *FLAC3D* or *3DEC*, for that matter) model, as you slowly rotate/zoom towards a minute detail, at certain angles a "shimmering" effect makes it nearly impossible to see which surface covers which one? Essentially, you can't visually inspect the intersection of two triangles especially when they make a shallow angle with each other. Clearly, this is critical while figuring out whether triangles intersect properly.

This is a graphic effect due to truncation errors in calculating triangle normals and the corresponding lighting effects.

From experience, this is often due to the model being excessively off-center with respect to the origin of the coordinates system. If you move the entire model to center it near the origin, this annoying graphic effect often disappears and often the resulting intersection calculation (Booleans, split, trim etc.) are more accurate.

Surface Meshes from Contours

Did you ever receive a huge number of contour lines from your client wishing they'd sent you the actual triangulated mesh the contours came from? *Rhino 5* can mesh this for you.

In Rhino 5, the **_MeshPatch** command which builds a Delaunay triangulation out of curves or point clouds.

To control how fine the resulting surface mesh will be, it is suggested not feeding the curves directly to **_MeshPatch**. Instead, select your contours, then **Curve|Point Object|Divide Curve by|Length of Segments**. Specify a segment length that half the vertical distance between consecutive contour lines (this, to prevent aliasing in the resulting triangulation). Now, select all the points, followed by **_MeshPatch** and <ENTER>, <ENTER>, et voila!

If the surface mesh is too noisy, just **Surface|Drape** it in a Top view. To get a nice looking mesh from a draped surface try **Mesh|From NURBS Control Polygon**.

Surface quality diagnostics in Rhino

There are a number of surface quality checks in Rhino that ensure good quality output.

- **_Check** or **_MeshRepair**. A clean bill of health from the mesh doctor is a necessary condition to obtain a good mesh from Griddle.
- **_ExtractMeshFacesByAspectRatio** with an aspect ratio of 10,000 or more often points to tolerance mismatches on the surface. Deleting these faces and using **_MatchMeshEdges** or other means often saves you a lot of trouble.

Tolerances and off-center models

Tolerances can be the cause of problems especially when dealing with intersection and Boolean operations on meshes and geometric entities.

This is particularly critical when the model center is far from the origin or when the model is very large. In all cases, make sure that your model is not too far away from the origin and in case it is move it closer to the origin. Moving the object closer to the origin also helps with the graphics both in Rhino and in your engineering analysis software.

When starting a new project, use the initial Rhino template to specify whether the model will be a "Large object in "meters", "small" object in "feet", etc., then import DXF, STL or even existing 3dm files into the new project. In this fashion you control the tolerance and not the default tolerance specified in the DXF. By default, **Option|Document Properties|Units** should say an absolute tolerance of 0.01, a relative tolerance of 1.0. and an angle tolerance of 1°.

When preparing a solid, in general you have to join multiple surfaces. Too small a tolerance prevents successful joining. However, when intersecting or doing a Boolean operation on polysurfaces or meshes, a smaller tolerance such as 0.0001 absolute, 0.01 relative and 0.01° in angles may be more appropriate.

Simplifying Complex & Thin Geological Structures?

PROBLEM:

You have a complex geological structure represented by a triangle mesh that has a finite small thickness but spans across a wide area. It really should be modeled as an interface in *FLAC3D* or a joint in *3DEC* but you need to somehow reduce it down to a single, smooth, low-triangle-count, good quality, average median triangular surface.

SOLUTION:

Select the mesh structure, then **_ExtractPt** to extract all its vertices as a cloud of points. While the points are selected, use **_MeshPatch** to triangulate these points. The result is a triangular mesh that is extremely choppy (like a waffle). This is because the triangulation tends to join points across the thickness of the geological structure connecting points on the footwall and hanging wall sides of the structure.

Now, select the choppy mesh and **Transform | Smooth**, check smoothX, Y and Z and Fix Boundaries. Set the Factor to 1 and OK. Repeat several times. As you do this, the mesh converges towards a smooth surface that is an average of surfaces on both sides of the thickness. Select the resulting mesh and use **_ReduceMesh** with 90% reduction to reduce it to a low-count mesh and you are done!

Troubleshooting. Where are my files?

Griddle and *BlockRanger* send file output to *Rhino*'s current working directory. When writing a mesh to an output file, *Griddle* and *BlockRanger* will indicate where the mesh file was written (Figure 170). If this is not your intended directory, then do a **_SetWorkingDirectory** in *Rhino*.

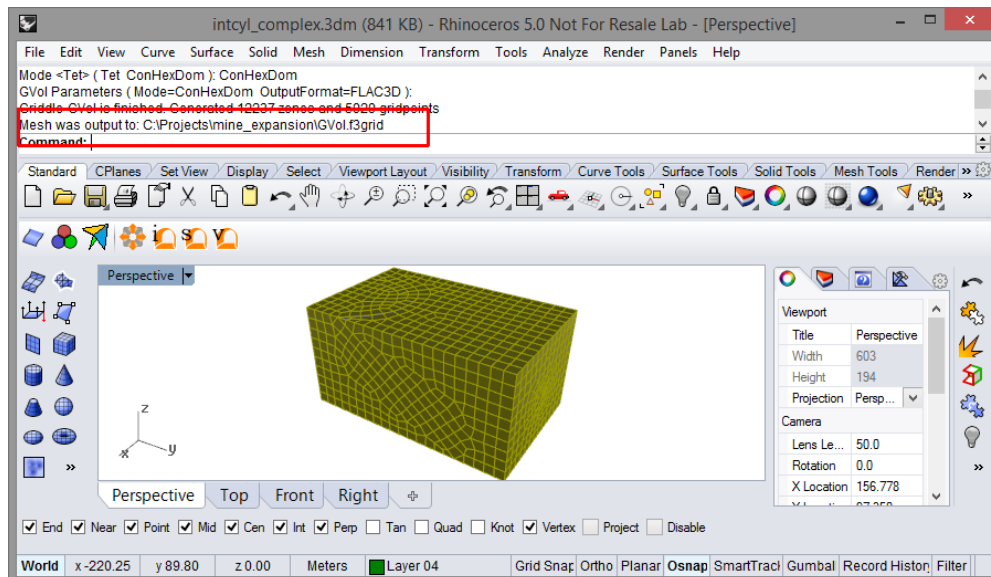


Figure 170: Output file and directory indicated by *Griddle*.

Griddle and *BlockRanger* will complain if they are working in a protected Windows directory as in Figure 171. **_SetWorkingDirectory** will fix this. If you are starting *Rhino* with a shortcut and that shortcut's, "Start in" field points to a protected directory you will often encounter the protected Windows directory message. Clear the "Start in" directory on your shortcut, by right-clicking on the shortcut, selecting Properties, and then the Shortcut tab. In this tab, delete the contents of the Start in field as shown in Figure 172. Doing this will result in *Rhino* using the shortcut's directory as the "start in" directory. This can be done with other Windows shortcuts.

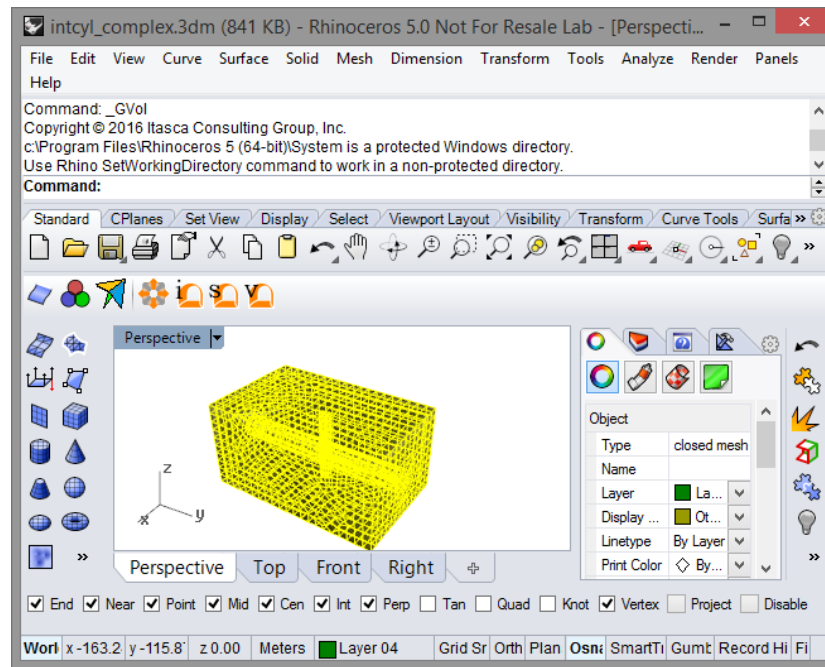


Figure 171: Message when working in a protected Windows directory.

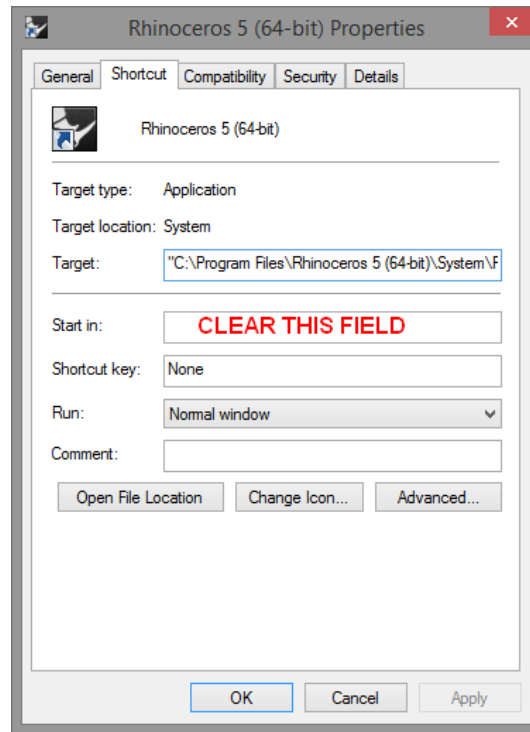


Figure 172: Clearing the "Start in" field of the *Rhino* shortcut.

References

Itasca Consulting Group Inc. (2012) *FLAC3D — Fast Lagrangian Analysis of Continua in Three Dimensions*, Ver. 5.0. Minneapolis: Itasca.

Itasca Consulting Group Inc. (2013) *3DEC — Three-Dimensional Distinct Element Code*, Ver. 5.0. Minneapolis: Itasca.