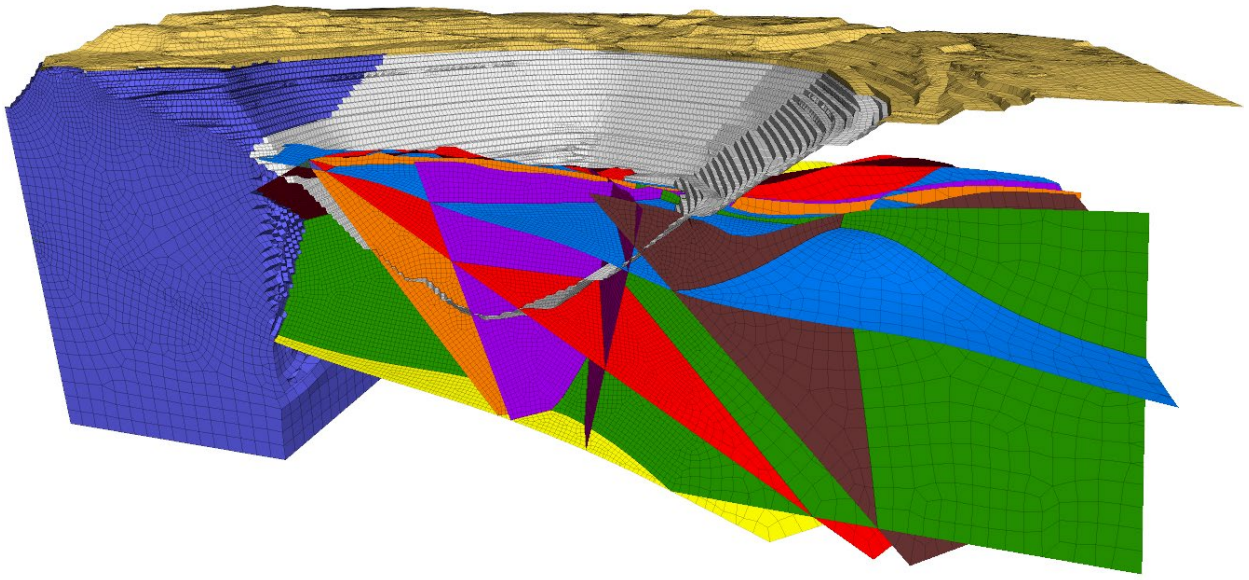
Griddle™

Advanced Grid Generation for Engineers and Scientists

Griddle 2.0 **Tutorial Examples**



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: griddle@itascacg.com
web: www.itascacg.com

Contents

Introduction

The tutorial examples provided in this document are designed to familiarize *Griddle* users with the principles of *Griddle* operation, workflows, and typical *Rhino* commands that allow the creation of structured and unstructured volume meshes suitable for numerical analysis.

The examples presented in the tutorial are not designed to illustrate exact (or even realistic) engineering models and conditions. However, they provide the basis on which *Griddle* users can build their knowledge for working with complex engineering models.

All the examples were developed in *Rhino 6*, but identical (or very similar) operations, commands, and workflows can be done in *Rhino 7*.

When working on the examples from the tutorial, periodically save the model in order to return to it if needed. *Rhino* provides a useful file saving feature, called *Incremental Save* (accessible from the top menu: **File** → **Incremental Save**).

Refer to the *Griddle* User Manual to find more information about *Griddle* commands and options, and the *Rhino* help system (top menu **Help** or **F1**) for *Rhino* operations.

The files corresponding to the tutorial examples can be accessed from the *Windows Start Menu* → *Itasca Griddle 2.0* → *Griddle 2.0 Tutorial Files* link. Clicking on the link opens user writable directory `ProgramData\Itasca\Griddle200` (typically on drive "`C:`") which contains the documentation and the tutorial material. Users can directly work in this directory. A reserve copy of the same material can be found in the `Documentation` subfolder of *Griddle* installation location (typically in `C:\Program Files\Itasca\Griddle200`).

Tutorial 1: A Single Cylinder

Expected work time: 0.5 – 1 hour

This tutorial provides information on how to create unstructured and structured meshes for a simple cylinder using *Griddle*'s ***GVol*** and ***BlockRanger*** volume meshers. Before proceeding with meshing, a cylindrical geometry is created, and this process differs depending on which volume mesher will be used.

Generating unstructured mesh with *GVol*

1. Open *Rhino*, and when the **Templates** splash screen appears, select **Small Objects, Meters**. The same can be done by navigating to **File** → **New** from the *Rhino* top menu.
2. Save the *Rhino* project at a desired location.
3. Type the **_Cylinder** command or select the **Solid** → **Cylinder** menu item. Type **0** and press **Enter** to set the location of the center of the cylinder base at the origin. Next, set the radius of the base to **2**. Enter **10** for the height of the cylinder. Press **Alt**+**Ctrl**+**E** to zoom out all viewports to the extent of all objects (or navigate to **View** → **Zoom** → **Zoom Extents All**). This completes construction of the geometry for a simple vertical cylinder (Figure 1).

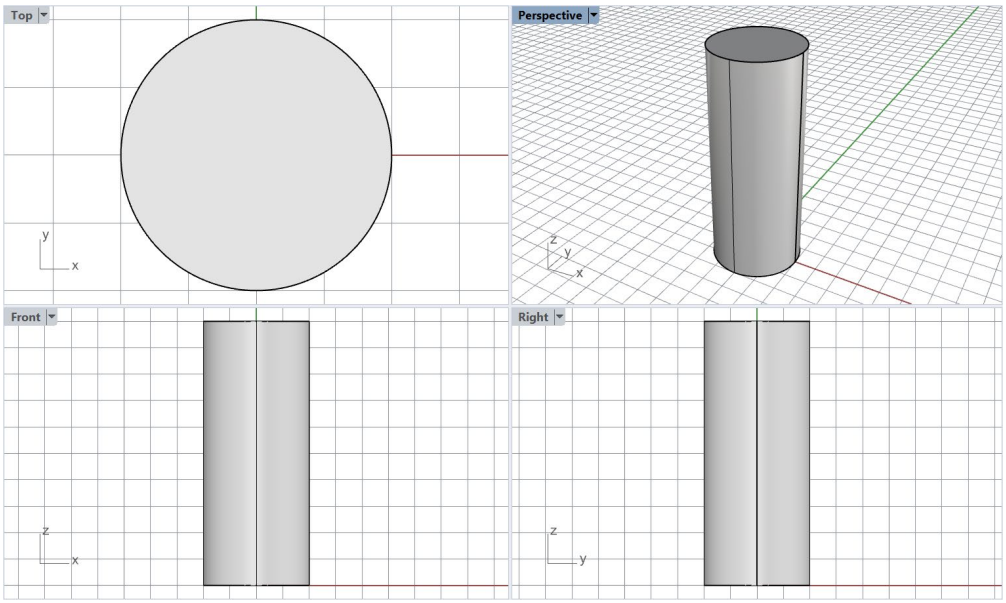


Figure 1: Solid representing a cylinder

The watertight geometry created in the previous steps can be referred to as a cylindrical solid. A solid is a closed surface that defines a volume with a clear interior and exterior. The next step is to triangulate the cylinder surface to create the initial triangular mesh.

4. Select the cylinder and type **_Mesh** in the *Rhino* command prompt. If the **Polygon Mesh Detailed Options** dialog window opens, click on the **Simple Controls…** button in the lower-right corner of the window to open a simplified dialog.
5. In the simplified dialog, move the slider all the way to the right towards **More polygons** and click **OK** to create the initial surface mesh (Figure 2).

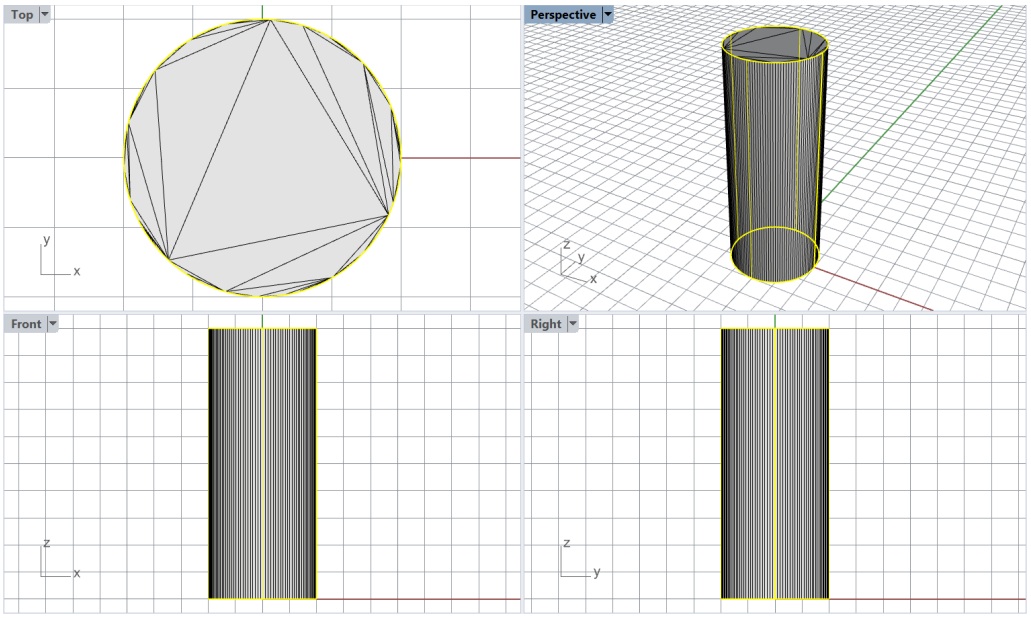


Figure 2: Original highlighted cylindrical surface and the newly created surface mesh superimposed on it.

6. While the original cylindrical surface is still highlighted (seen in light yellow in Figure 2), type the **_Hide** command to hide it, making only the surface mesh visible. This triangular surface mesh serves as input to *Griddle* tools for unstructured meshing.
7. Select the surface mesh and type **_GSurf** or click on the icon in the *Griddle* toolbar to remesh the initial triangular mesh**.** Select *Mode* = *QuadDom*, *MinEdgeLength* = 0.5, *MaxEdgeLength* = 0.5 and keep all other parameters at the default values. The resulting mesh is shown in Figure 3.

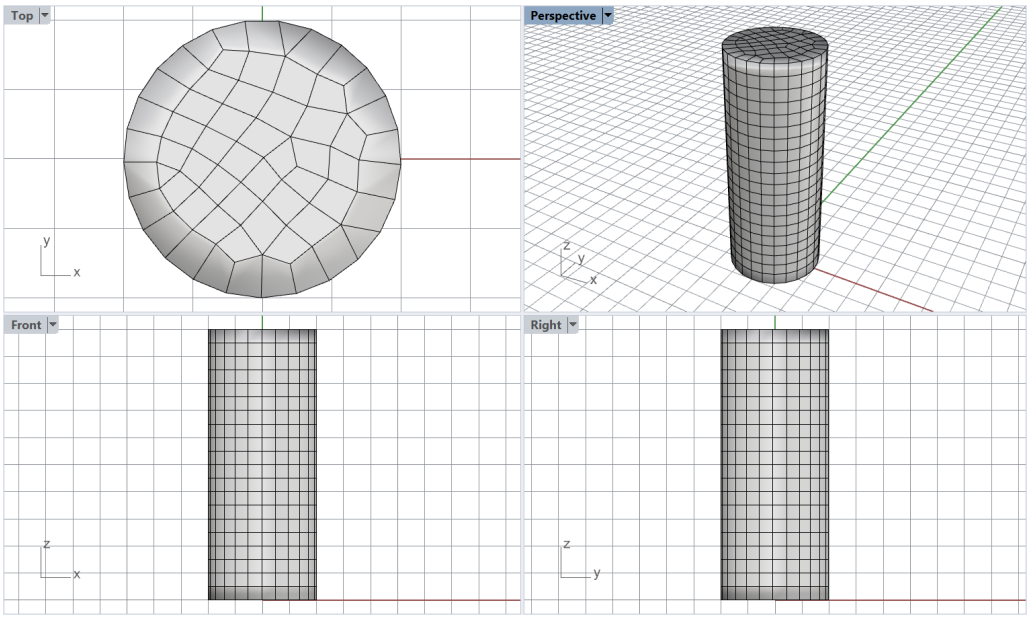


Figure 3: Quad-dominant surface mesh generated by _GSurf.

8. Select the surface mesh generated in the previous step and type **_GVol** or click on the icon in the *Griddle* toolbar. Select **MeshSettings** → **Mode=HexDom**, leave all other parameters at the

default settings and press **Enter** twice. After the **Save As** dialog appears, type a desired name for the output file. ***GVol*** will generate a conformal hex-dominant volume mesh and output it using the *FLAC3D* binary format (the default format) to the selected file. This file can be imported into *FLAC3D* using *FLAC3D*'s **File** → **Grid** → **Import from FLAC3D …** option (Figure 4).

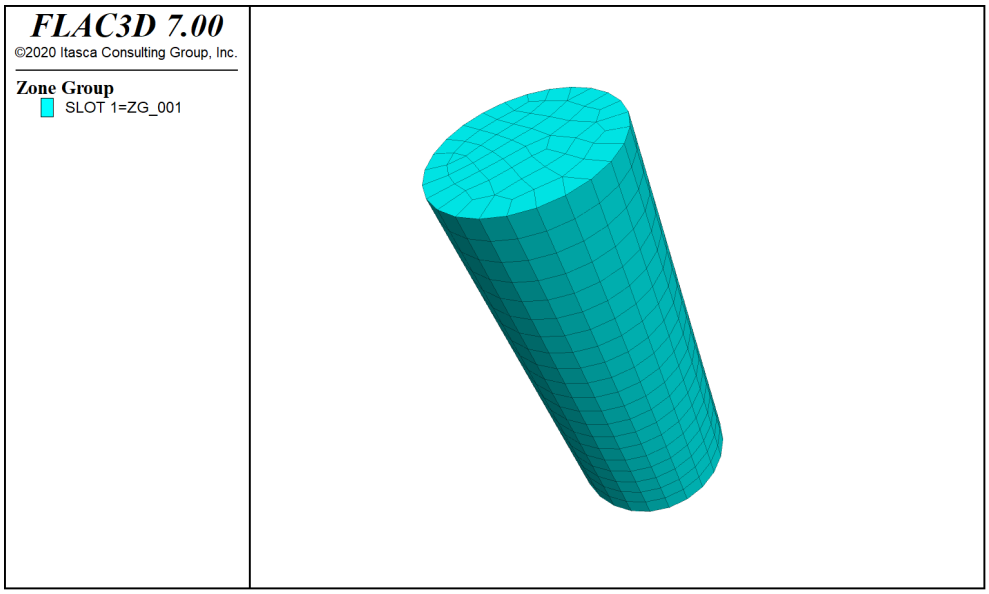


Figure 4: Volume mesh of cylinder imported into *FLAC3D*.

Although the mesh in Figure 3 and Figure 4 may resemble a structured mesh or a pure hexahedral mesh, the ***GVol*** output log file reports that the output mesh contains:

```
number of elements:
  total: 1433
  hexahedra: 1072 (74.81% of total, 94.04% of volume)
  prisms: 67 (4.68% of total, 2.63% of volume)
  pyramids: 187 (13.05% of total, 2.62% of volume)
  tetrahedra: 107 (7.47% of total, 0.71% of volume)
```

This report confirms that the output mesh is hex-dominant as it was selected in ***GVol*** options. Note that users may get slightly different results with regard to the number of elements of each type (this also depends on the particular version of *Griddle*).

Generating structured mesh with *BlockRanger*

BlockRanger is *Griddle*'s all-hex structured volume mesher, and it operates using different principles than the unstructured mesher *GVol*. As such, ***BlockRanger*** operates only on four-, five-, or six-sided solids (not surface meshes). The cylinder created in the previous section is not such a solid as it contains only three distinct sides. Therefore, to create a structured mesh for the cylinder, the initial solid must be subdivided into several pieces suitable for ***BlockRanger***. There are different ways of accomplishing this task; the most commonly used is described below.

1. Open *Rhino*, and when the **Templates** splash screen appears, select **Small Objects, Meters**. The same can be done by navigating to **File** → **New** from the *Rhino* top menu.
2. Save the *Rhino* project at a desired location.
3. Double-click on *Top* viewport to maximize it.
4. In the *Rhino* command prompt, type the following commands to create the initial curves:
   - **_Line** → *Start of line* = 1,1 → *End of line* = -1,1
   - **_Line** → *Start of line* = 1,1 → *End of line* = 2,2
   - **_Line** → *Start of line* = -1,1 → *End of line* = -2,2
   - **_Arc** → *Center of arc* = 0 →*Start of arc* = 2,2 →*End point or angle* = -2,2
     When creating an arc, make sure that the mouse cursor stays above the previously created lines, so the arc goes through the top portion of a circle (Figure 5).

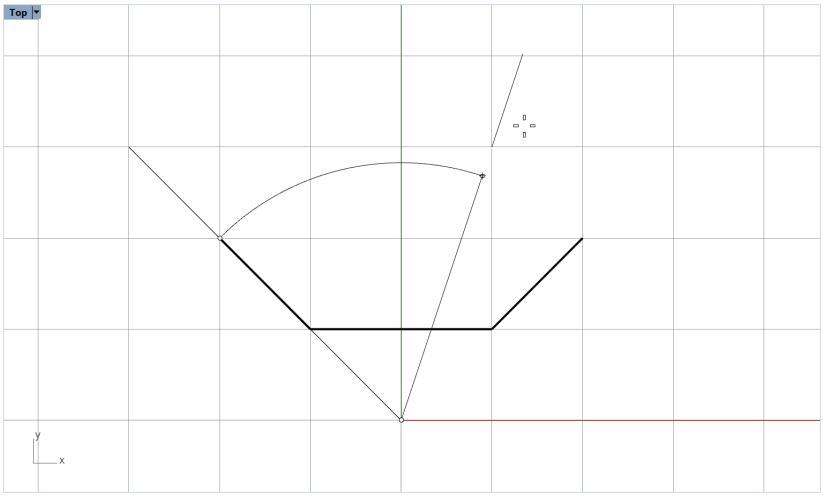


Figure 5: Creating an arc between two points.

5. Select all lines with **Ctrl**+**A** and type the **_Join** command to create a single polyline as in Figure 6.

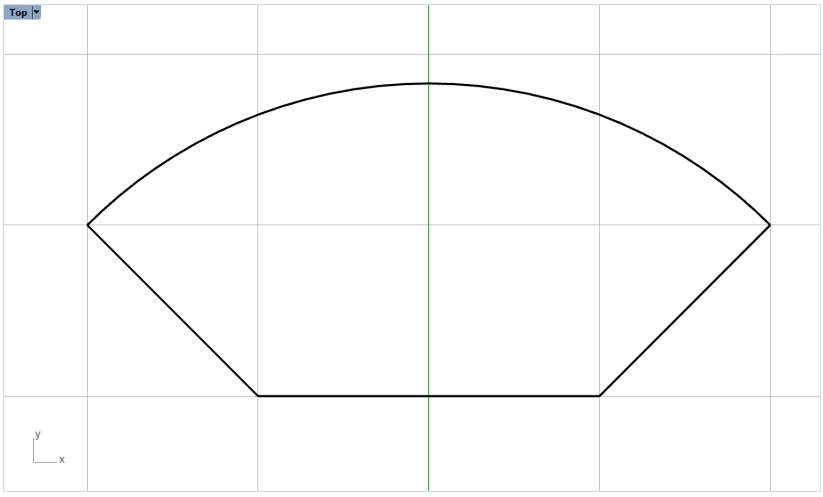


Figure 6: Single polyline consisting of three lines and an arc.

6. While the polyline is selected, type the **_Rotate** command and click on *Copy=Yes*
   - *Center of rotation* = 0

- *Angle of first reference point* = 90 (**Enter**)
- *Angle of first reference point* = -90 (**Enter**)
- *Angle of first reference point* = 180 (**Enter**)
- Press **Enter**

7. Type _Polyline and proceed with:
   - *Start of polyline* = 1,1
   - Next point of polyline = -1,1
   - Next point of polyline = -1,-1
   - Next point of polyline = 1,-1
   - Next point of polyline = 1,1
8. Select all 5 objects and type **_ColorizeObjects** in the command prompt (or click on the  icon in the *Griddle* toolbar). The command will assign a random color to each object (Figure 7).

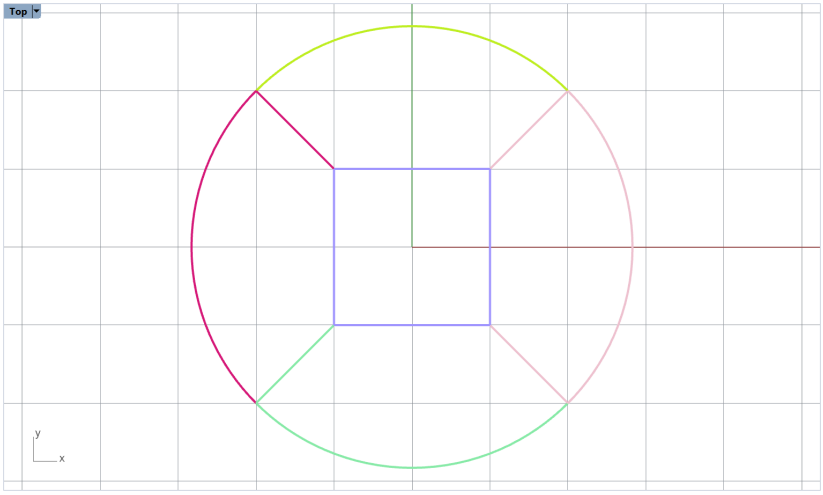


Figure 7: Five colorized polylines (some edges overlap).

9. Switch to *Perspective* viewport by clicking on the icon under the viewport.
10. Select all objects and type the **_ExtrudeCrv** command and set *Extrusion distance* = **10**.
11. Delete the original polylines while they are selected.
12. Select all objects (should be 5 polysurfaces) and type the **_Cap** command to close them and create watertight solids.
13. Select all and type the **_ColorizeObjects command**.

The resulting five solids represent a cylinder, but each solid is topologically equivalent to a rectangular prism. All these solids are suitable for ***BlockRanger***.

14. Select all objects (or type the **_SelPolysrf** command to select only polysurfaces).
15. Type **_BR** or click on the  icon in the *Griddle* toolbar and specify only *GenerateSurfaceMesh* = *BySolid* to visualize the faces on the external and internal boundaries of the solids. Set *MeshSettings* → *MaxEdgeLength* = 0.85. The output volume mesh will be outputted by default in FLAC3D binary format.

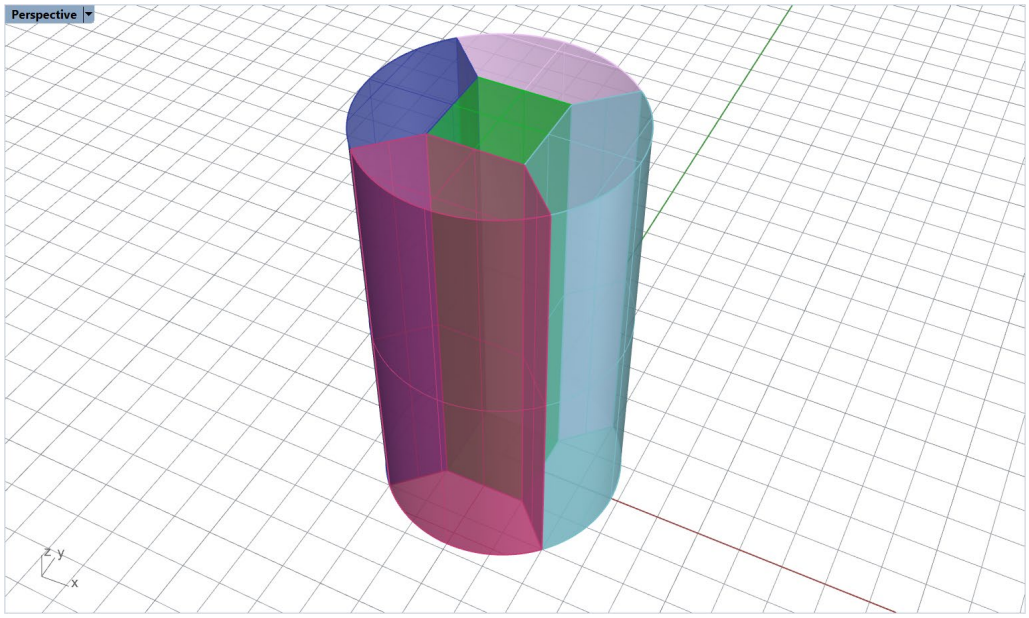


Figure 8: A cylinder subdivided into 5 solids (Ghosted view).

16. Type the **_SelPolysrf** command to select only polysurfaces and delete them.

The resulting surface mesh is provided in Figure 9 and it shows high quality quadrilateral faces on the boundaries. Note that ***BlockRanger*** produces a single mesh; for better representation, the mesh in Figure 9 was split into several submeshes (using ***GExtract*** tool) which were colorized (with ***ColorizeObjects*** tool).

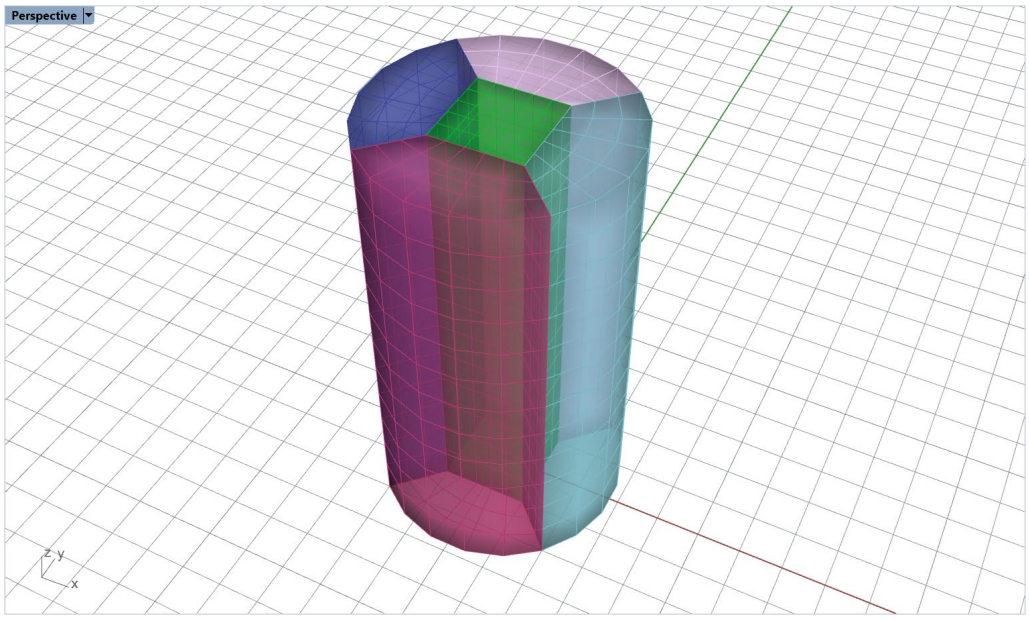


Figure 9: Surface mesh corresponding to the generated volume mesh (Ghosted view).

The output file with a volume mesh can be imported in *FLAC3D*. The zone plot will show five grouped regions composed of a structured pure hexahedral mesh (Figure 10). Note that for this simple extruded

geometry, users can use the *FLAC3D Extruder* to produce similar results. However, ***BlockRanger*** may operate on rather complex 3D solids, which will be shown in further tutorials.

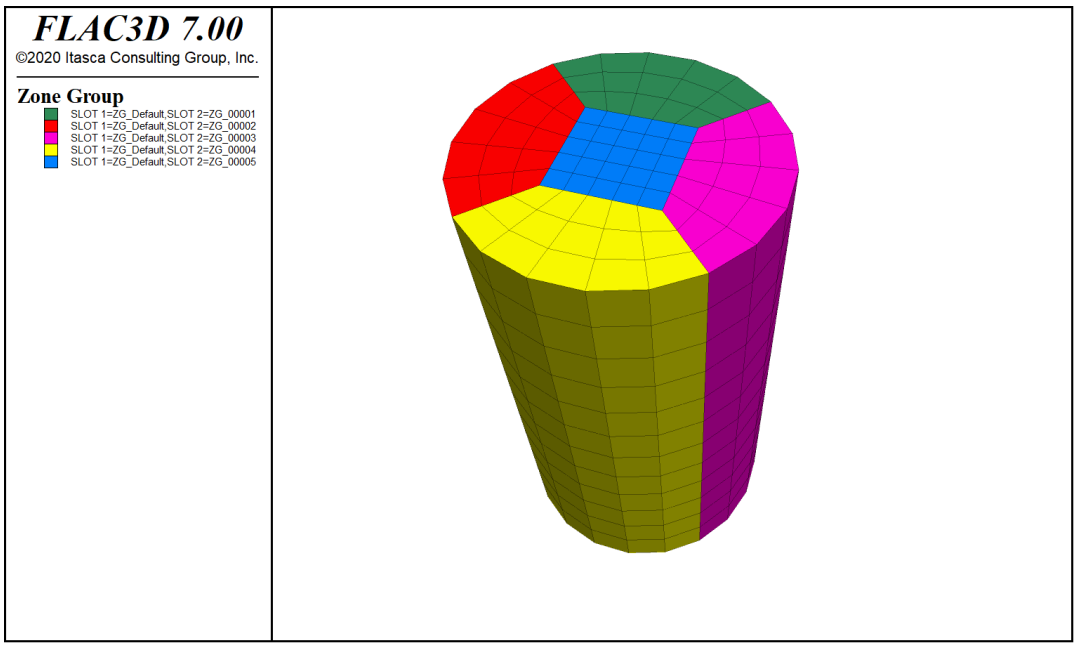


Figure 10: A cylinder subdivided into five solids (Ghosted view).

Tutorial 2: Vertical Shaft Excavation in a Stratified Soil

Expected work time: 0.5 – 1 hour

This tutorial shows an example of constructing the geometry and mesh for a vertical shaft (150 ft deep, diameter of 30 ft) that is to be excavated in stages. The shaft is constructed in two-layered soil with the surface separating the soil layers located at a depth of 50 ft (Figure 11). The model domain is represented as a cube 200 ft × 200 ft × 235 ft. Each excavation stage is 10 ft deep; therefore, there are 15 stages as shown by the red circular pattern in Figure 11.

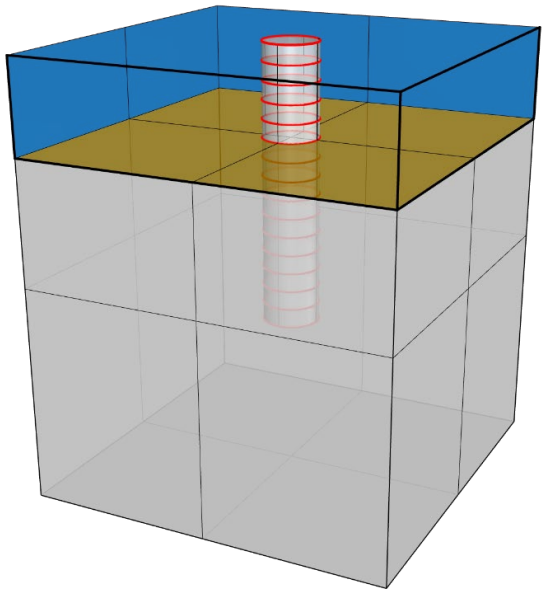


Figure 11: A general view of the model of a vertical shaft in a stratified soil.

Generation of analytical geometry

1. Start *Rhino*, select **Large Objects, Feet** (similar method as step 1 in Tutorial 1) and save the *Rhino* project at a desired location.
2. Select or maximize *Perspective* viewport and switch to Shaded view.
3. Create a cylinder by typing the **_Cylinder** command and specify *Base of cylinder* = 0, *Radius* = 15, *End of cylinder* = -10.
4. Create a line by using the **_Line** command from 0,0,-10 to 0,0,-150.
5. Select the cylinder and type the **_ArrayCrv** command. Then select the line and, in the pop-up dialog, set the *Number of items* to 15 and keep O*rientation* as *Freeform*; press **OK**. The command will copy the initial 10 ft deep cylinder multiple times to create the desired 15 stages of excavations.
6. Type the **_SelCrv** command to select the curve and delete it.
7. Verify that 15 stages were created by selecting everything in the viewport (**Ctrl**+**A**); the command area should display the message:
   `15 extrusions added to selection.`
8. Create three points that will define the plane separating two layers of soil by using the **_Point** command. For the coordinates, specify:
   - -100,-100,-50

- 100,-100,-50
- 100,100,-50

9. Connect all points by a horizontal plane by typing the **_Plane** command and start dragging a plane from one of the "side" points (the first or third in the list above). The objects should look as shown in Figure 12.

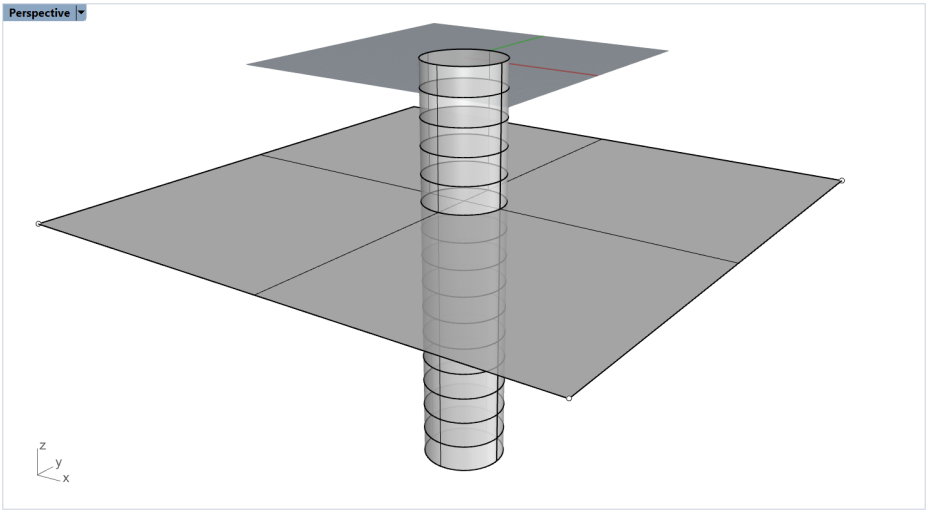


Figure 12: Construction of a shaft and a plane separating two layers of soil

10. Select and delete points using the **_SelPt** command.
11. Create a domain box by typing the **_Box** command (or select in the menu **Solid→Box→Diagonal**) and click on the *Diagonal* option to define the box by 2 points. Enter **-100,-100,-235** for the coordinates of the first point and **100,100,0** for the second point.
12. Switch to Wireframe view and zoom to all extents if needed (**Alt**+**Ctrl**+**E**).
13. Select all objects and type the command **_NonManifoldMerge** (or click on the  icon in the *Griddle* toolbar) to create a single non-manifold polysurface. This complex polysurface will serve as the initial geometry for meshing.

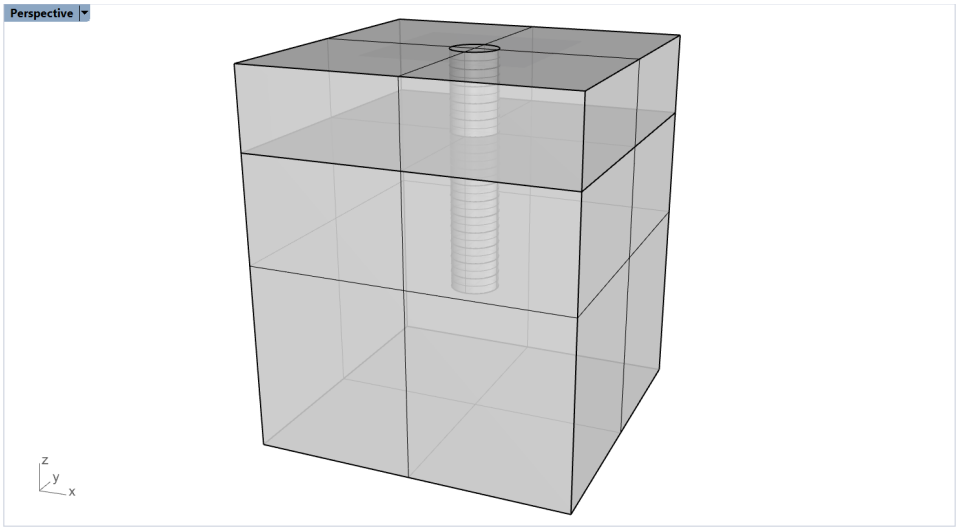


Figure 13: Complete initial geometry of the model of a shaft and two soil layers.

Meshing with unstructured mesh

The geometry created in the previous step is ready for meshing.

14. Select all objects and type the command **_Mesh**. If a simplified dialog appears, click on the *Detailed Controls*… button and enter the values as shown in Figure 14.

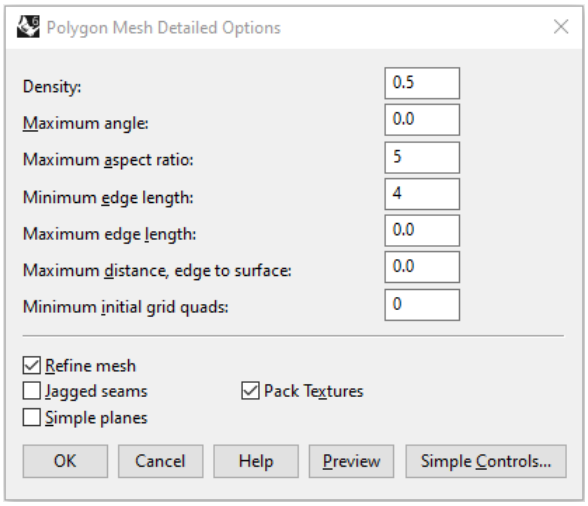


Figure 14: *Rhino* meshing dialog to create an initial mesh.

15. While the polysurface is still selected, hide it with the **_Hide** command. If it is not selected, select it with the **_SelPolysrf** command. Only the mesh should be visible at this point.
16. Select the surface mesh and type **_GSurf** or click on the  icon in the *Griddle* toolbar to remesh the initial triangular mesh**.** Select *Mode* = *QuadDom*, *MinEdgeLength* = 4, *MaxEdgeLength* = 20 and keep all other parameters at the default values. The resulting mesh is shown in Figure 15.

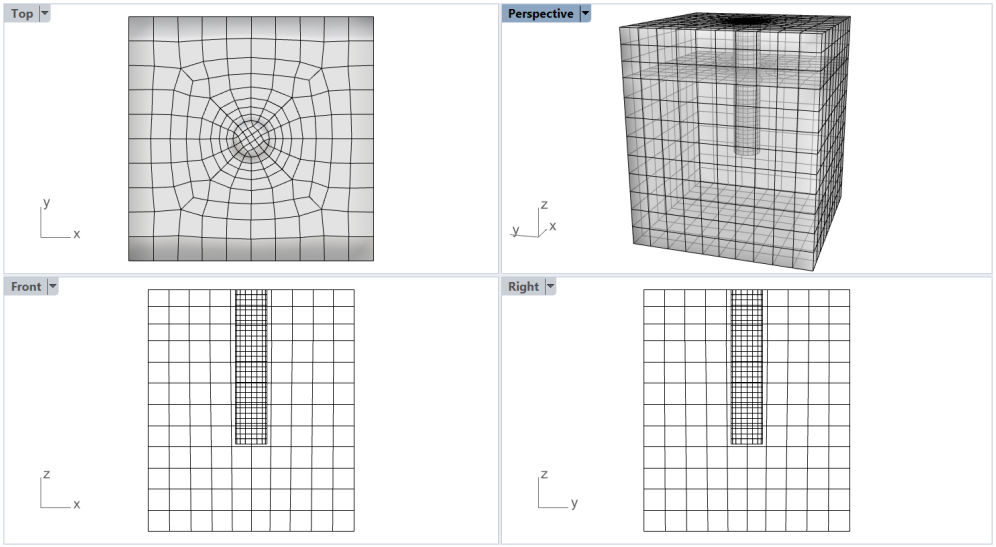


Figure 15: *Rhino* meshing dialog to create the initial mesh.

The surface mesh in Figure 15 looks good and rather symmetric despite the fact that this is an unstructured mesh. ***GSurf*** was able to create such a mesh due to simple symmetric geometry and the choice of meshing parameters. In general, however, unstructured meshes are rarely symmetric.

17. Now the model is ready for volume meshing. To show how the volume meshing parameters influence the final mesh, two cases are considered here.

    - Select the surface mesh and type **_GVol** or click on the  icon in the *Griddle* toolbar. Select *MeshSettings* → *Mode=HexDom*, leave all other parameters at the default settings and press **Enter** twice. Keep the output format as *FLAC3D* Binary.

      The resulting *FLAC3D* grid is shown in Figure 16. The grid is of good quality, but the soil layers have a noticeably sharp change in zone size closer to the shaft. This can be improved by varying the *GVol* parameters.

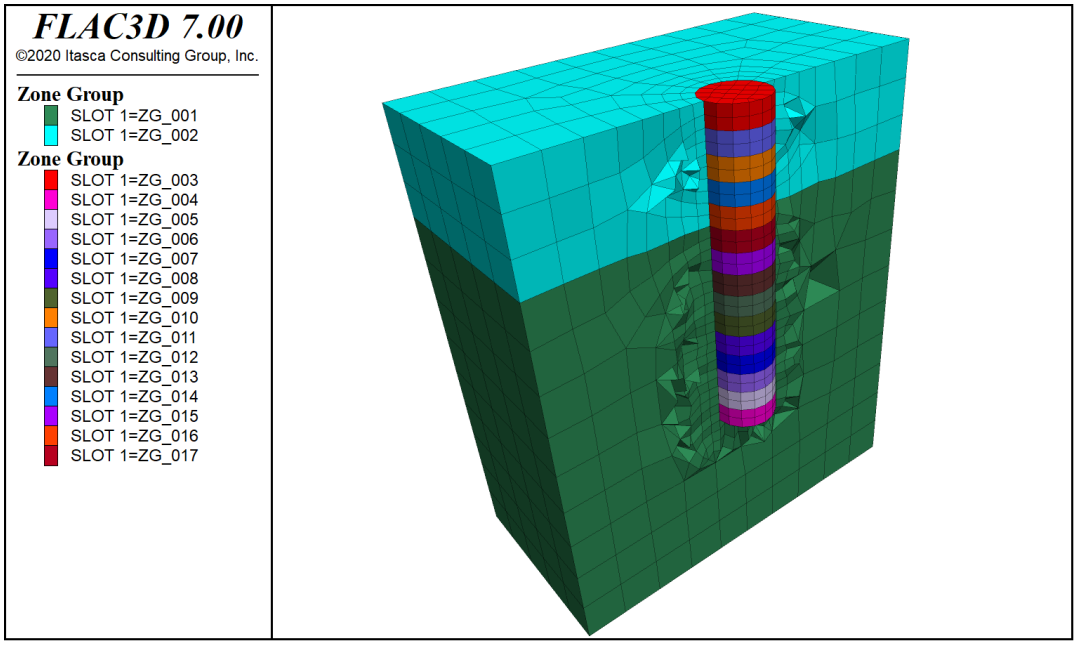


Figure 16: *FLAC3D* grid of the shaft model when using default *GVol* parameters.

    - Create a new volume mesh using the following *GVol* parameters: *MeshSettings* → *Mode=HexDom, MaxGradation*=1*, TargetSize*=5*, Optimization*=10, *ShapeQuality*=1. For the explanation of each parameter, refer to the *Griddle 2.0 User Manual*. The resulting mesh is shown in Figure 17.

      The grid surrounding the shaft in Figure 17 has many more zones, and the transition from coarse to finer zones is more gradual. Typically, such results are achieved when *MaxGradation* is set to a small value (< 0.5), but in this case, as the domain boundaries are relatively close to the shaft, a larger value of *MaxGradation* has to be used to force the mesher to quickly change from a large zone size on the boundary of the domain to a finer size specified by the *TargetSize*. Use of the highest possible values for *Optimization* and *ShapeQuality* allows for improving zone quality.

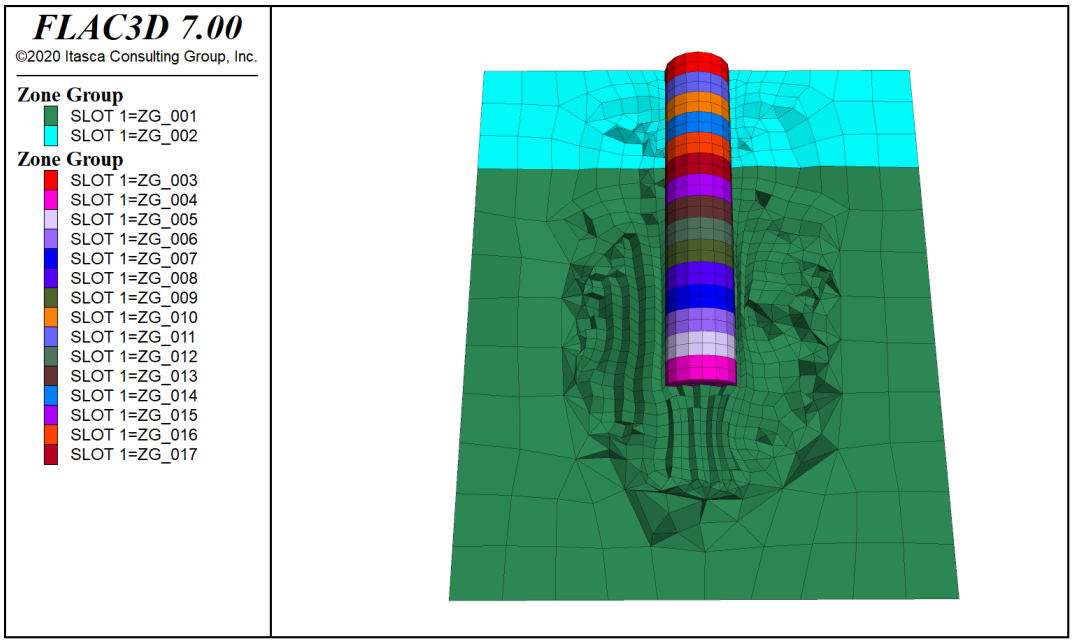


Figure 17: *FLAC3D* grid of the shaft model when using custom *GVol* parameters.

Excavation of the shaft can be modeled by gradual removal of the stages inside the shaft (i.e., zone groups ZG_003 – ZG_017 as in Figure 16 and Figure 17).

Tutorial 3: Meshing a Model with Analytical Surfaces

Expected work time: 1 hour

This tutorial describes two meshing approaches for a model containing analytical objects, such as *Rhino* (poly)surfaces and solids defined by NURBS (non-uniform rational basis spline), SubD (high-precision Catmull Clark subdivision surfaces; only in *Rhino 7*) or BRep (boundary representation).

1. Open project "T3_BridgeSupport.3dm" located in folder "TutorialExamples\3_BridgeSupport". Create a copy of this initial project with a different name at a desired location (use **File → Save As**).
2. The model represents a part of a complex bridge support structure as shown in Figure 18. The model consists of 13 surfaces, seven polysurfaces, and two extrusions represented by BRep objects. Note how the objects are split by each other (e.g., the top light blue surface is split in three pieces); this is done for better meshing results.

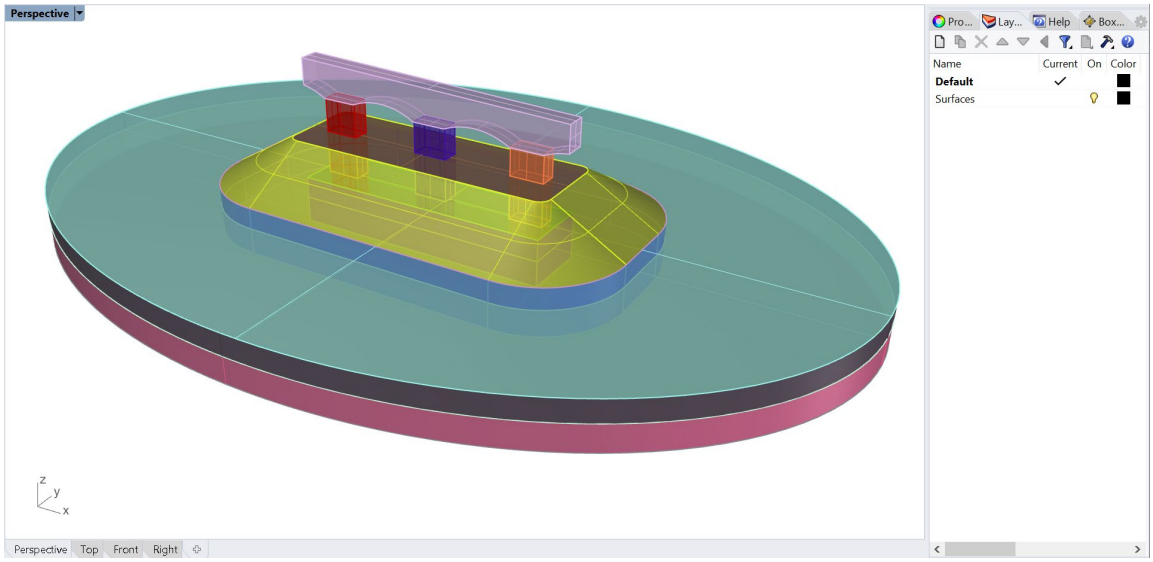


Figure 18: Model of a bridge support structure.

The typical approach to construct meshes when starting with initial (poly)surfaces such as in Figure 18 consists of the following:[1]

- Mesh (triangulate) all the (poly)surfaces to create initial rough meshes; if initial meshes are provided, this step is skipped.
- Intersect the initial meshes with *Griddle*'s surface mesh intersector ***GInt***.
- Remesh all meshes with surface remesher ***GSurf***.
- Create volume mesh with unstructured volume mesher ***GVol***.

Users should follow this general approach in the majority of cases. However, in certain situations, a slight variation of this approach may be more efficient in preparing the model for volume meshing. The standard approach is described in the first sub-section of this tutorial, and potential issues are outlined.

[1] See also *Griddle 2.0 User Manual*, Section "*Using Griddle Tools for Unstructured Meshing*".

The second sub-section presents a variation of the approach, which can be applied to models similar to the one described in this Tutorial.

Mesh generation using *GInt* and *GSurf* (approach 1)

3. Maximize *Perspective* viewport and switch to Shaded view for better visibility.
4. Navigate to the *Layers* pane and create a new layer "Meshes_Approach1". If the pane is not visible, it can be opened from **Edit** → **Layers** → **Edit Layers…**.
5. Select all objects (**Ctrl**+**A**) and type the **_Mesh** command to create the initial mesh of all objects. Use detailed controls with the following settings to construct a fine initial mesh (Figure 19).

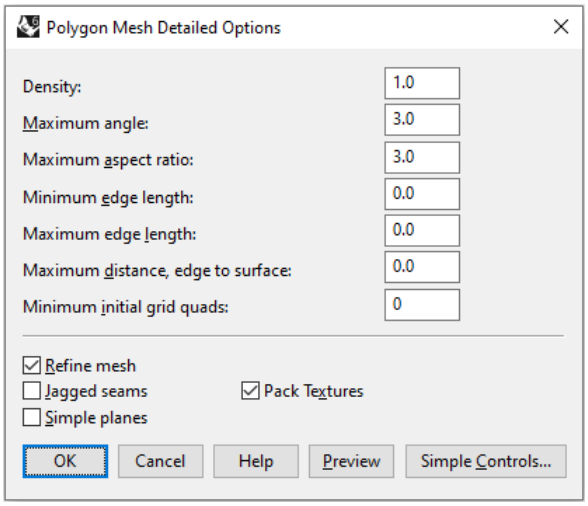


Figure 19: *Rhino* meshing dialog to create fine initial mesh.

6. While all the (poly)surfaces are selected, invert the selection by typing the command **_Invert**, which will select meshes only. Navigate to the *Properties* pane (or press **F3**) and change the mesh layer to "Meshes_Approach1". Open the *Layers* pane and turn off the "Surfaces" layer (click on the lightbulb).

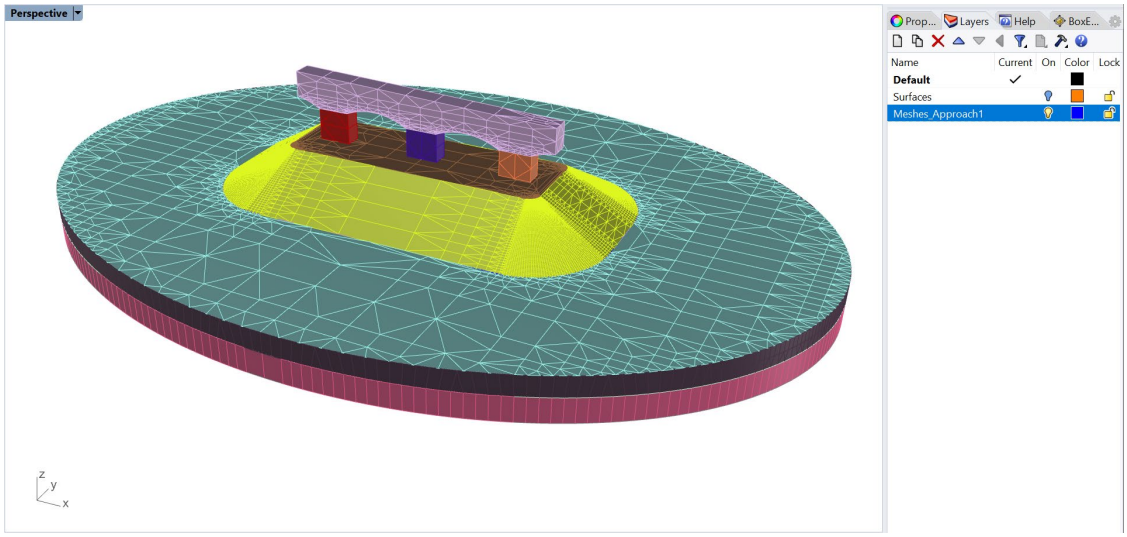


Figure 20: The initial *Rhino* mesh.

The mesh in Figure 20 is not conformal and contains gaps along the curved boundaries (Figure 21). This is due to fact that surface meshes are a discrete representation of the initial analytical (in this case, BRep) surfaces, and the mesh for each object is created independently from any other objects. The gaps and non-conformal faces can be fixed by intersecting meshes using the ***GInt*** surface mesh intersector with a properly selected tolerance. Having a watertight surface mesh domain is a required condition for volume meshing. At the same time, making all model meshes fully conformal is usually needed but is not always required for volume meshing (e.g., separate watertight domains must contain conformal surface meshes but the domains themselves may be non-conformal in the areas of contact).

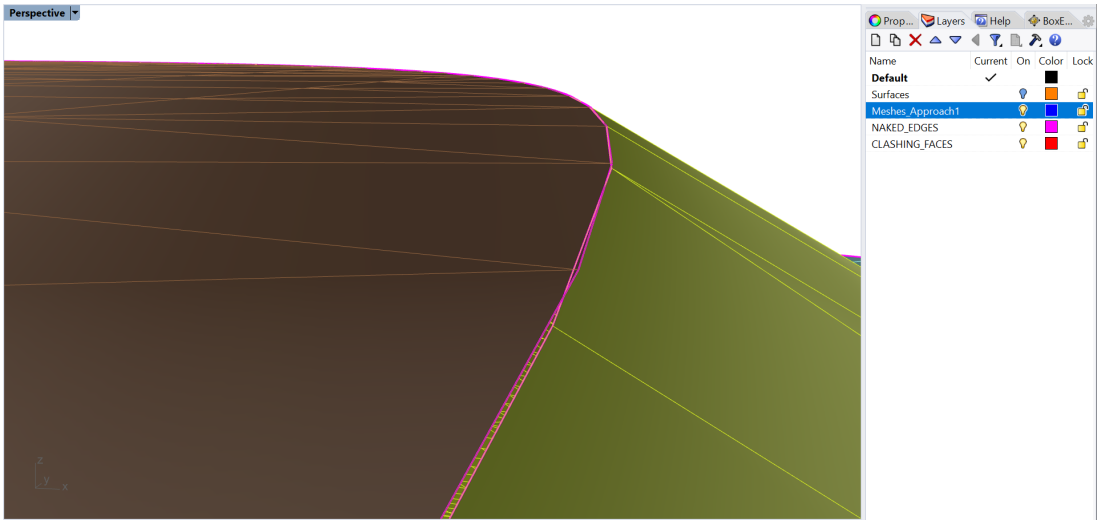


Figure 21: Mismatching mesh boundaries in the initial *Rhino* mesh.

In certain cases, it may be challenging to determine a proper tolerance value when intersecting many meshes. This is especially true for big models with large variations in face (element) sizes and with gaps/mismatches between neighboring faces that are of similar size as some of the mesh faces. The reason for this challenge is that a small ***GInt*** tolerance may not be sufficient to close the gaps and intersect all the faces to make them conformal, while a large tolerance may create undesired intersections and changes of smaller faces that fall within the tolerance. Therefore, it is recommended to do mesh intersections with minimal acceptable tolerance to avoid creating incorrect intersections. Users may need to do several trials starting with 0 tolerance and gradually increase it until all faces are properly intersected. This process is outlined below.

7. Select all surface meshes and type **_GInt** or click on the  icon in the *Griddle* toolbar. For the first trial, keep *Tolerance* = 0, and click on *AdvancedParameters*. By default, ***GInt*** preserves initial mesh shapes and other parameters, which is set by the option *OutputMesh* = *Separated*. In this case, the output mesh needs to be merged to clearly identify if there are any gaps or non-conformal faces between the meshes. Click on *OutputMesh* to set it to *Merged*[2]. Press **Enter** twice to execute ***GInt***.

[2] When setting *OutputMesh* = *Merged*, ***GInt*** intersects all selected surface meshes and creates a single merged mesh assigned to the default (current) layer. Merged meshes have a smaller number of boundary edges and allow for easy identification of holes, gaps, or non-conformal faces within the mesh.

8. To check if there are any gaps or non-conformal edges/faces left, select the output mesh and type **_GHeal** or click on the  icon in the *Griddle* toolbar. Select the *ShowErrors* option, which will create two new layers, “NAKED_EDGES” and “CLASHING_FACES”, containing outlines of open mesh boundaries (e.g., internal holes) and non-conformal faces. The results show that there is a large number of gaps and some non-conformal faces (Figure 22 and Figure 23). This is because ***GInt*** *Tolerance* was set to 0 to intersect only those mesh faces that are in full contact. Naked edges can also be visualized with *Rhino*’s **_ShowEdges** command.

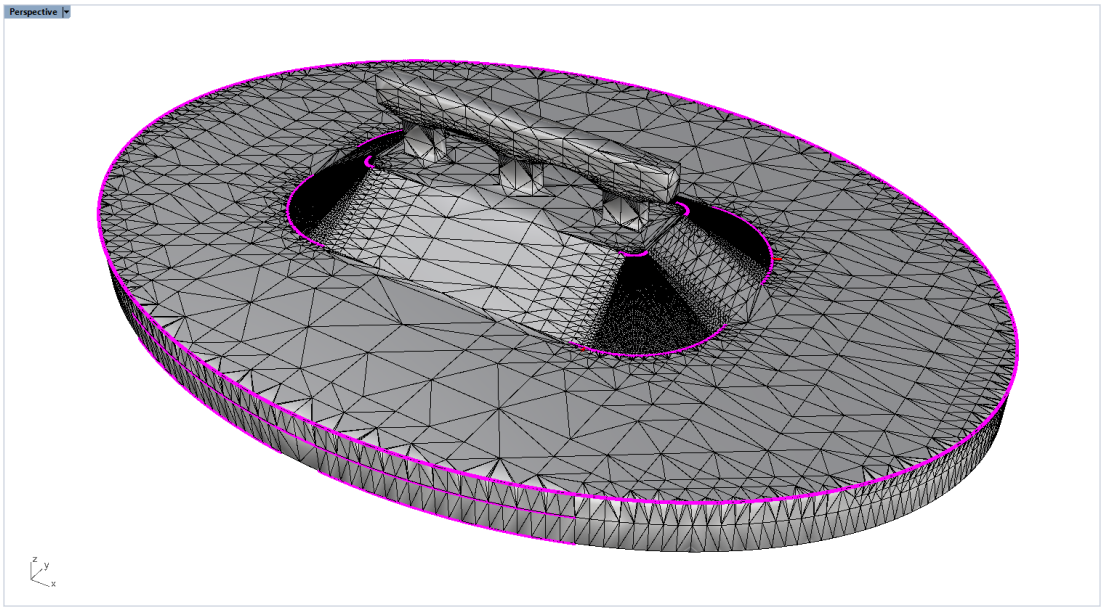


Figure 22: Naked edges (in pink) and clashing faces (red) after using *GInt* with *Tolerance* = 0.

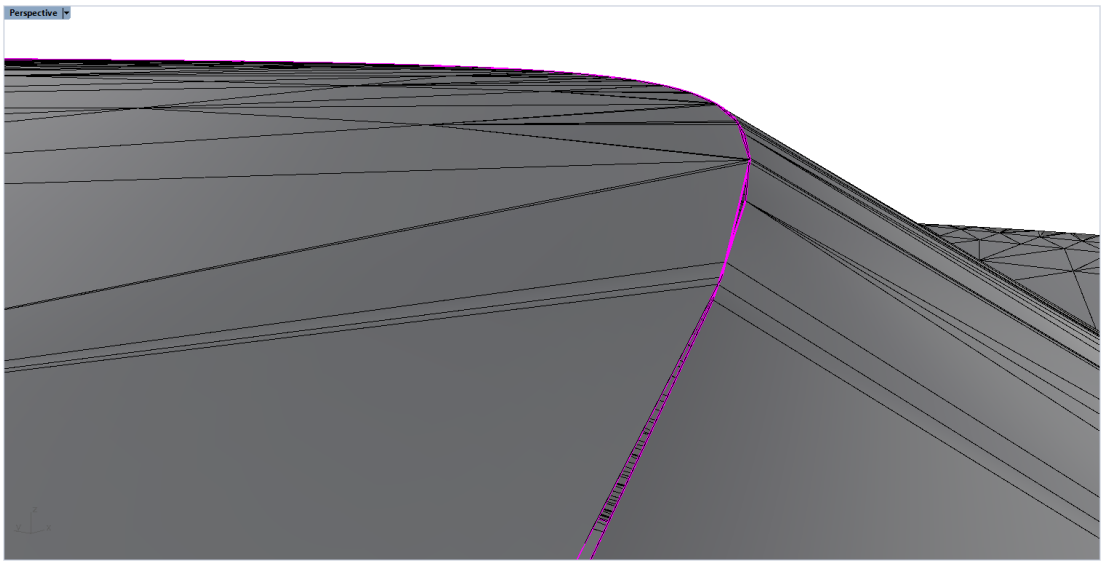


Figure 23: Zoomed naked edges (in pink) after using *GInt* with *Tolerance* = 0. Compare with Figure 21.

9. Undo the last two commands (**_GHeal** and **_GInt**) by pressing **Ctrl**+**Z** twice. Ensure the initial **_Mesh** command is not undone.

10. Select all initial surface meshes and run ***GInt*** again. This time specify *Tolerance* = 0.005 and keep all other parameters the same as in the previous run. After ***GInt*** executes, visualize naked edges and clashing faces as outlined previously in step 8. The results show that there are still some naked edges and clashing faces (Figure 24), which indicates that ***GInt*** *Tolerance* is still not sufficient.

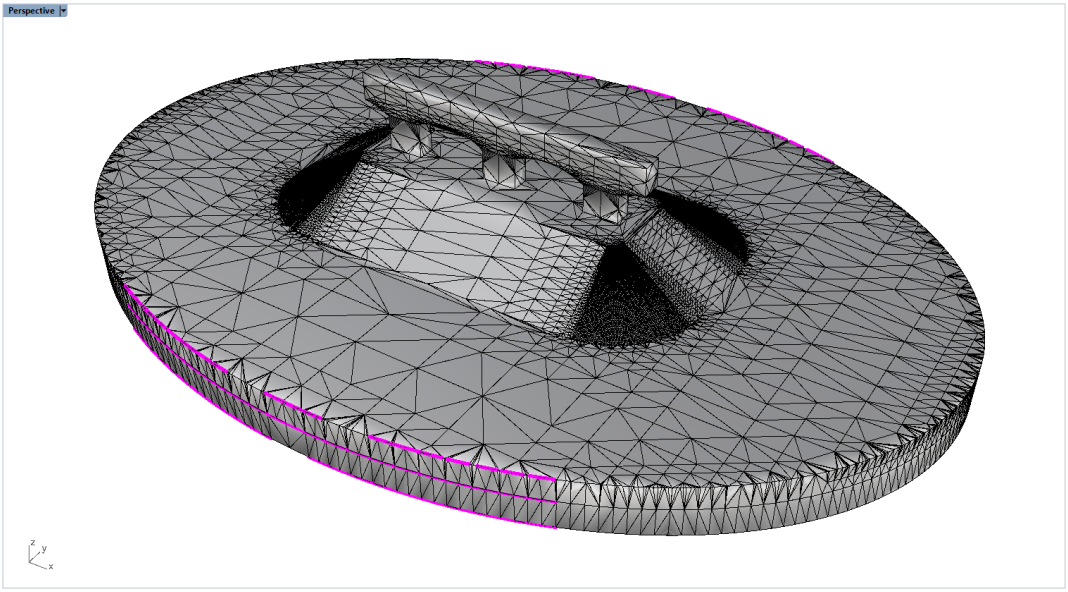


Figure 24: Naked edges (in pink) and clashing faces (red) after using *GInt* with *Tolerance* = 0.005. Compare with Figure 22.

11. Repeat steps 9 and 10 using *Tolerance* = 0.015. The output mesh does not have naked edges or clashing faces, which means that all faces from distinct meshes were properly intersected. To see this, **_Hide** the mesh and select all objects (**Ctrl**+**A**). *Rhino* should report that no objects are selected.
12. Undo the last operations (**_Hide**/**_Show**, **_GHeal**, **_GInt**) by pressing **Ctrl**+**Z** until the initial *Rhino* mesh is shown as in Figure 20.
13. Run ***GInt*** again with *Tolerance* = 0.015 and *OutputMesh* = *Separated* (in *AdvancedParameters*). Note that this time all distinct meshes are preserved, and it would be hard to check for naked edges as each mesh has its own boundary. That is why a merged mesh was used in the previous steps.
14. Select all surface meshes and remesh with ***GSurf*** ( ) using the following parameters: *Mode* = *QuadDom*, *MinEdgeLength* = 0.25, *MaxEdgeLength* = 5, RidgeAngle = 5, AdvancedParameters → MaxGradation = 0.06, and keep all other parameters at the default values. The resulting meshes are shown in Figure 25. These meshes look good overall, but zoomed views of the curved bottom / top embankment sections reveal (yellow-green mesh) that these areas contain small elements (faces) that do not fully conform to the initial shape (Figure 26). This is due to independent meshing of the initial BRep surfaces and due to using a large tolerance in ***GInt***[3].

[3]Another potential issue can be revealed by completing step 14 with *OutputMesh* = *Merged*. The single output mesh may contain a few overlapping (clashing) faces that were introduced by ***GSurf***. However, these clashing faces do not affect the ability to create a volume mesh, as they are between separate watertight sub-domains. In general, clashing faces within a single mesh can be fixed using ***GHeal***'s *FixClashingFaces* function, which locally intersects them.

15. Create a volume mesh using ***GVol*** ( ) and any desired (or default) parameters. ***GVol*** automatically checks input surface meshes for the presence of naked edges (e.g., holes, gaps) and clashing (non-conformal) faces to assure that input meshes create a watertight domain; if any such issues are found, an information message is shown. The message does not prevent ***GVol*** from running (click **Yes** to proceed); it is only a reminder for users to verify that input meshes are valid. In this case, ***GVol*** input consists of multiple surface meshes with boundaries treated as naked edges.

    The initial error check and the message can be turned off by setting *IniErrorCheck* to *No* (in ***GVol***'s *MeshSettings* options).

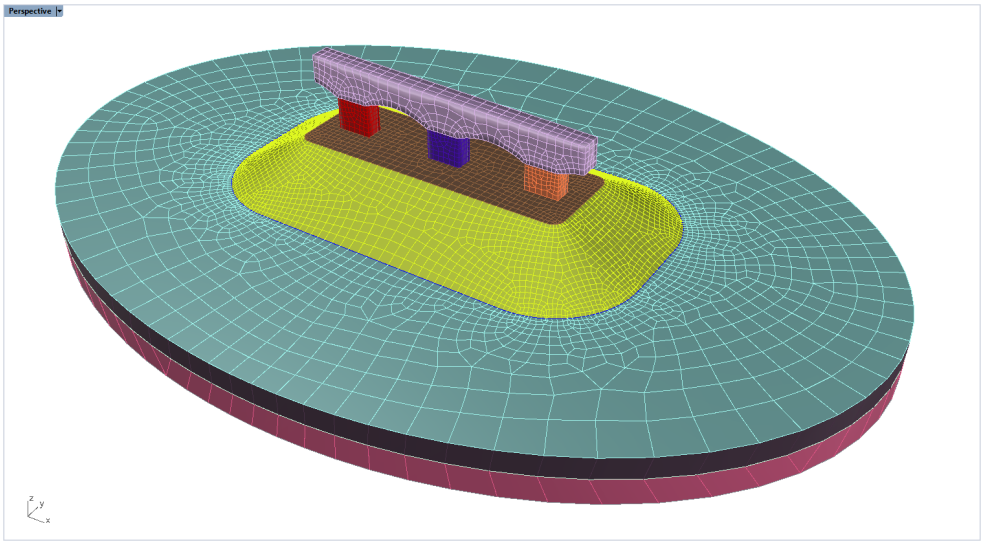


Figure 25: Remeshed surface meshes using standard approach.

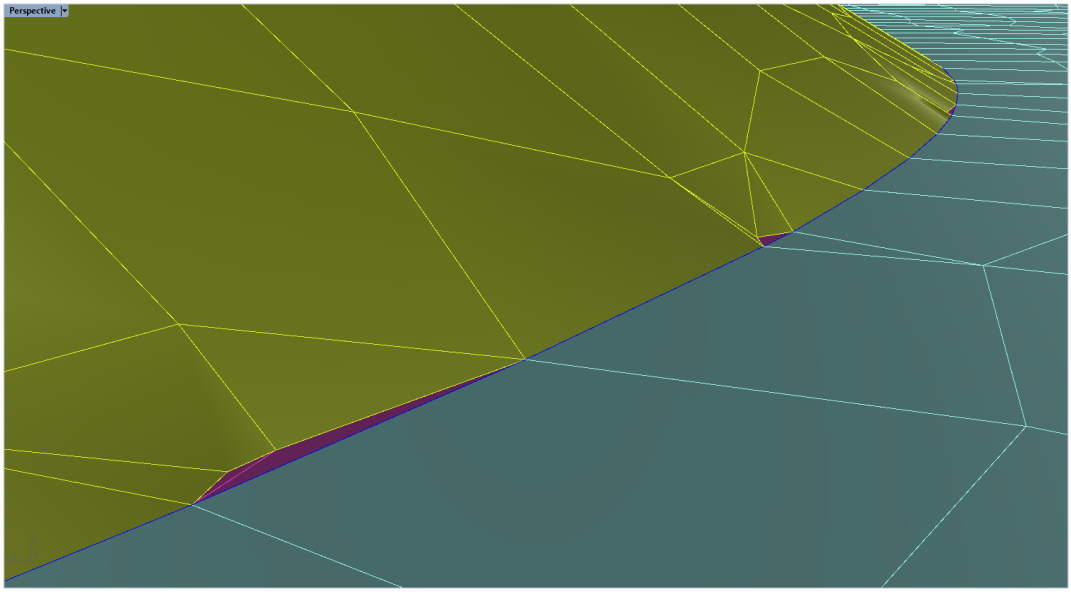


Figure 26: Slightly deformed surface meshes when using standard approach.

Mesh generation using *NonManifoldMerge* and *GSurf* (approach 2)

The issues in the previous approach can be avoided if all initial (poly)surfaces are merged into a single non-manifold polysurface before meshing. The initial meshing (triangulation) of all connecting surfaces within a single object happens in a conformal manner, and no gaps are created between mesh faces. Therefore, with this approach, surface mesh intersection is often not needed. The drawback of this approach is that when the initial objects are merged into a single non-manifold polysurface, user-specified object information is lost (e.g., surface names, colors, etc.). However, if these parameters are not important, the approach outlined below may be much easier, more efficient, and yield better quality meshes.

16. Navigate to the *Layers* pane and create a new layer named "Meshes_Approach2". Turn on layer "Surfaces" to show the initial surfaces.
    Delete layers (if any) "NAKED_EDGES", "CLASHING_FACES", and "MESHING_ERRORS".
17. Select all objects in layer "Surfaces" (as in Figure 18) and use command **_NonManifoldMerge** ( ) to create a single non-manifold polysurface (Figure 27). The new polysurface is placed into the current layer ("Default"); move it back to the "Surface" layer by changing the object layer assignment in the *Properties* pane (**F3**).

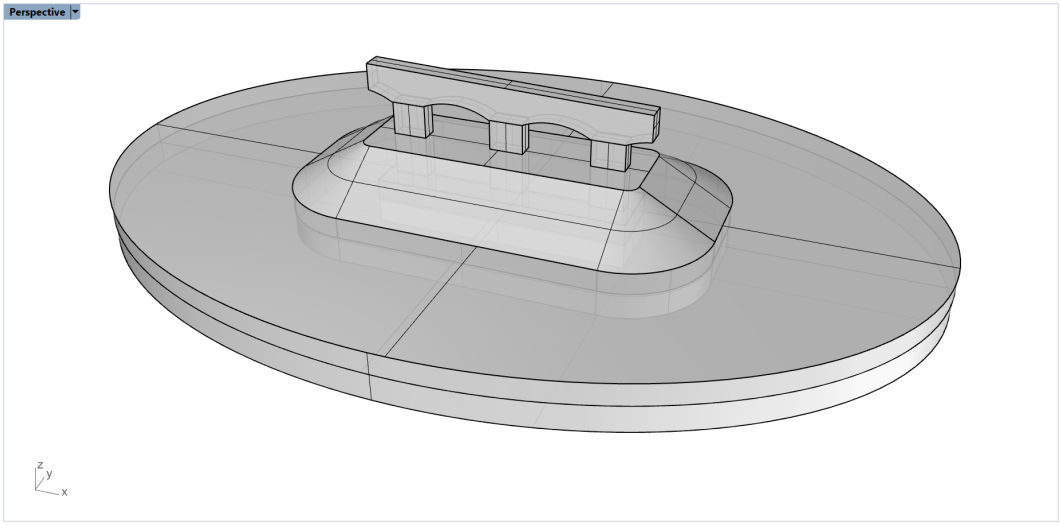


Figure 27: Single polysurface constructed by the NonManifoldMerge command.

18. Mesh the polysurface using the **_Mesh** command with the same settings as in Figure 19. Note that the default mesh settings and *Simple Controls* can be used in this case, as there will be no issues with gaps or non-conformal faces (there is no need to have a highly detailed initial mesh).
19. Assign layer "Meshes_Approach2" to the newly created mesh and turn off the "Surface" layer so only the mesh is present in the viewports.
20. Remesh the initial mesh with ***GSurf*** ( ) using the same settings as in Step 14. The quality of elements (faces) around the curved parts of the embankment is much higher if compared with the mesh in Figure 25 (or with the meshes in layer "Meshes_Approach1").

21. Verify that there are no naked edges (gaps, holes) or clashing faces in the final surface mesh using ***GHeal*** ( ) → *ShowErrors*. There should be no curves outlining any problems in layers "NAKED_EDGES" and "CLASHING_FACES".

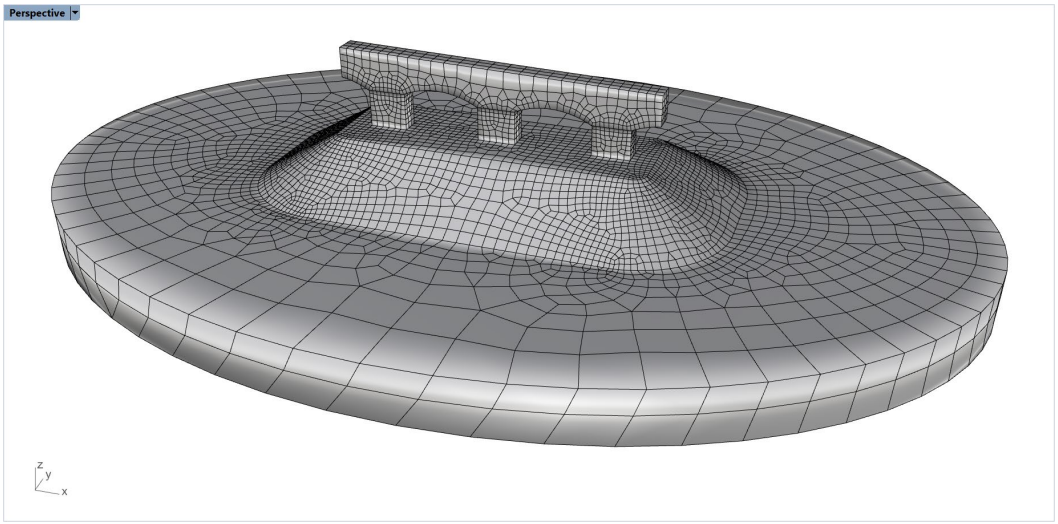


Figure 28: Remeshed surface mesh in the second approach.

22. The mesh is ready for volume meshing. Run ***GVol*** ( ) with any desired (or default) parameters. Due to the higher quality of input surface mesh(es), the total number of elements in the output volume mesh is smaller in this approach than in the previous one (e.g., for Hex-dominant mesh, this approach produces in total ~47,000 elements and the first approach results in ~69,000 elements).

As mentioned previously, the issue of this approach is that all initial (poly)surfaces must be merged into a single nonmanifold polysurface before meshing. If there is a need to assign a name, color, or any other property to a specific surface mesh before volume meshing, the merged surface mesh can be easily split into sub-meshes using *Griddle*'s tool ***GExtract***.

23. Select the merged surface mesh (as in Figure 28) and type the **_GExtract** command or click on the icon in the *Griddle* toolbar. Select option *AllSurfaces* and use *MaxBreakAngle* = 180 to separate meshes only along non-manifold edges (the break angle will not have an effect as it is set to 180°). Alternatively, if break angle = 89° is used, all meshes that connect at an angle > 89° will also be separated (thus, meshes that are perpendicular to each other will be separated).
24. Select all (36) meshes and use *Griddle*'s command **_ColorizeObjects** or click on the icon in the *Griddle* toolbar. This will assign a unique color to each *Rhino* object (Figure 29).
25. Now a name can be assigned to each surface mesh if there is a need to transfer surface mesh names into a volume mesh (available only for *FLAC3D* and *3DEC* output formats in ***GVol***).

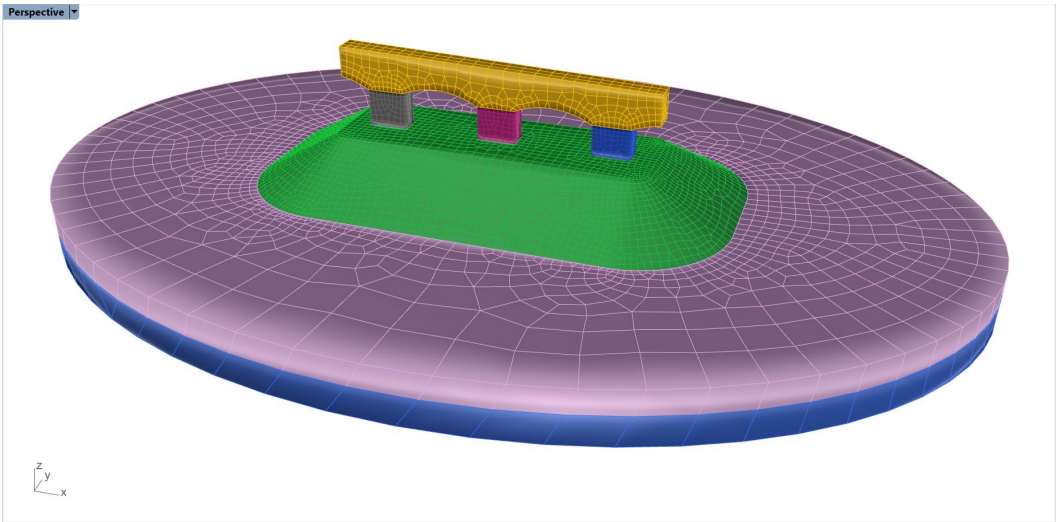


Figure 29: Mesh from Figure 28 “exploded” into sub-meshes, which are subsequently colorized.

Tutorial 4: Creating a Structured Mesh with *BlockRanger*

Expected work time: 1–2 hours

This tutorial provides detailed information on how to build a block-structured hexahedral mesh from an initial CAD model provided by a DXF file. To create a mesh using ***BlockRanger*** (**_BR** command), an assembly of six-, five-, or four-sided *Rhino* solids that conform to the reference model must be created. The creation of such an assembly is the objective of this example. The reference model is shown in the left side of Figure 30. The final hexahedral mesh is depicted to the right. This 3D slope model has a shape similar to a bathtub; hence, it is sometimes referred to as the "bathtub" model.

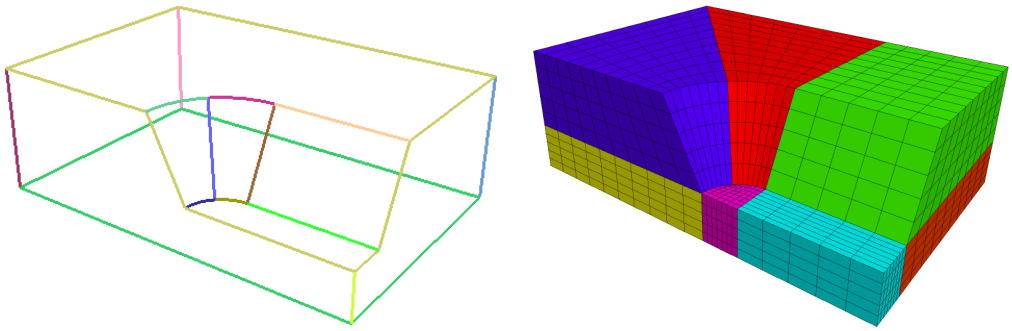


Figure 30: Reference model (left) and *BlockRanger* generated mesh (right).

Importing the geometry

1. Start *Rhino*, select **Large Objects, Meters** (similar method as step 1 in Tutorial 1) and save the *Rhino* project at a desired location.
2. Import the initial geometry from a DXF file by navigating to **File** → **Import** and select "T4_3DSlope.dxf" located in folder "TutorialExamples\4_StructuredMesh_SlopeModel". In the **Import** dialog, select **Model** → **Meters** and **Layout Units** → **Millimeters**, and keep other parameters at the default values.
3. Maximize *Perspective* viewport and make sure Shaded view is selected.
4. In the top menu, navigate to **View** → **Display Options…** This will open the **Rhino Options** dialog; select **View** → **Display Modes** →**Shaded** → **Objects** → **Curves** and set **Curve Width** to 4. This will display all curves thicker in the Shaded view. The initial settings can be always restored by clicking on **Restore Defaults**.
5. Navigate to the *Layers* pane as shown in Figure 31. If the pane is not visible, open it by going to **Edit→Layers→Edit Layers…** In the *Layers* pane, rename the automatically created layer "lines" to "Reference" and delete all layers except "Default" and "Reference". Ensure that all objects are now in layer "Reference" by selecting all objects and navigating to the *Properties* pane (or press **F3**).
6. Select all objects and type the command **_Move**. For the *Point to move from,* select the bottom-left corner of the model and for the *Point to move to*, type 0. This will move all objects and align the bottom-left corner of the model with zero coordinates. Compare the results with Figure 31.

It is always recommended to move/center objects imported from a DXF (or other formats) closer to zero coordinates as it improves the accuracy of *Rhino* operations.

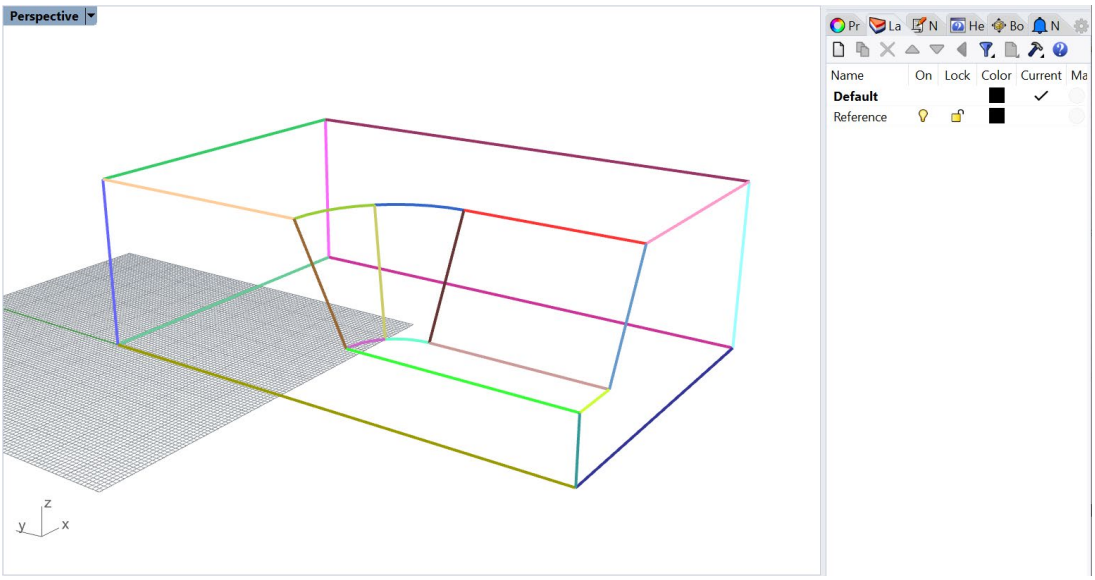


Figure 31: The reference model.

Choosing the right mesh block layout for the model

BlockRanger operates on solids and meshes each solid block individually. The present model may be decomposed into solid blocks in many ways. Some of the possible options are shown in Figure 32 through Figure 35. The arrangement of blocks depicted in Figure 34 will be used in this tutorial.

Each of the blocks shown in Figure 32 through Figure 35 satisfy ***BlockRanger*** requirements for solids, which includes the following:

- Permissible types of solids:
  - four-sided solids made of three-sided faces (a topological tetrahedron);
  - five-sided solids made of two three-sided faces connected through three four-sided faces (a topological triangular prism); and
  - six-sided solids made of four-sided faces (a topological hexahedron).
- Faces must be simple faces that cannot be further “exploded” into simpler faces.
- Face edges must only be simple curves that cannot be further “exploded” into simpler curves.

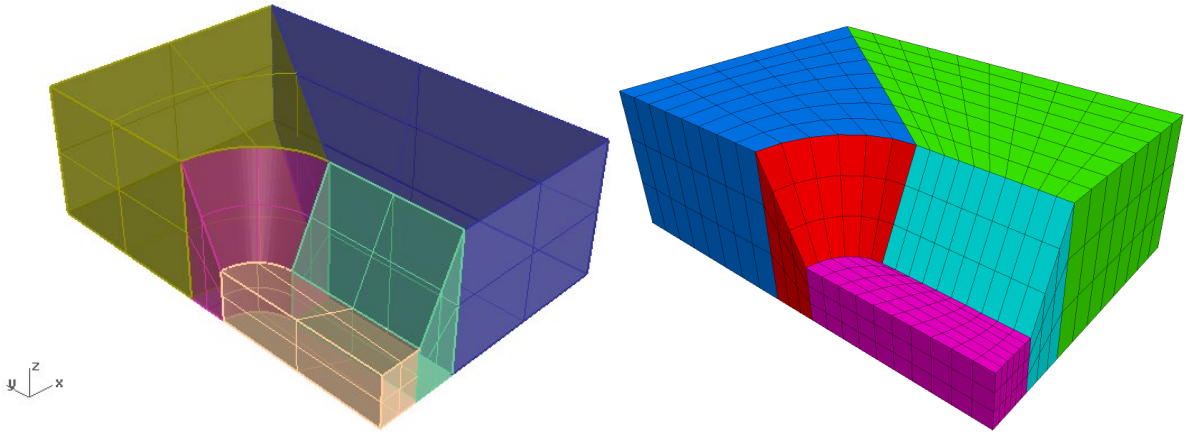


Figure 32: Solid layout resulting in five mesh blocks.

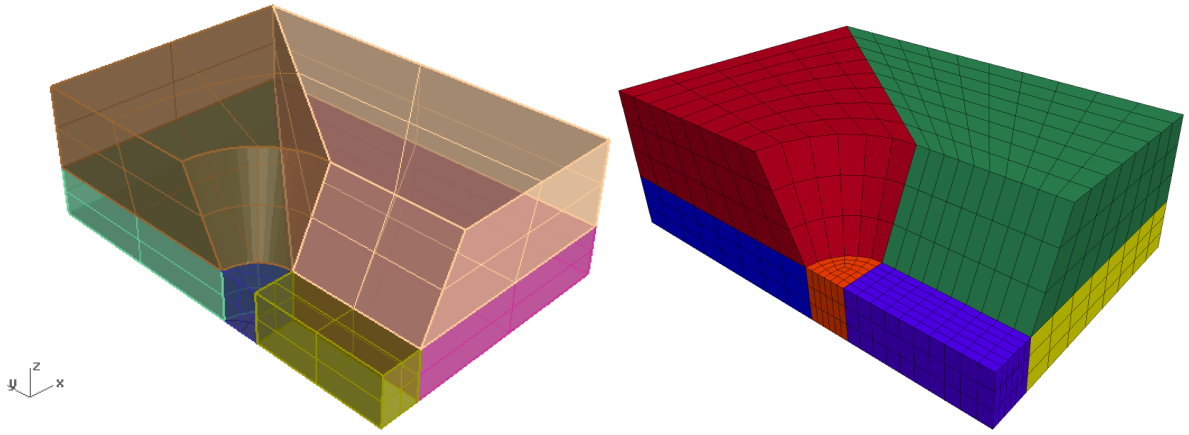


Figure 33: Solid layout resulting in six mesh blocks.

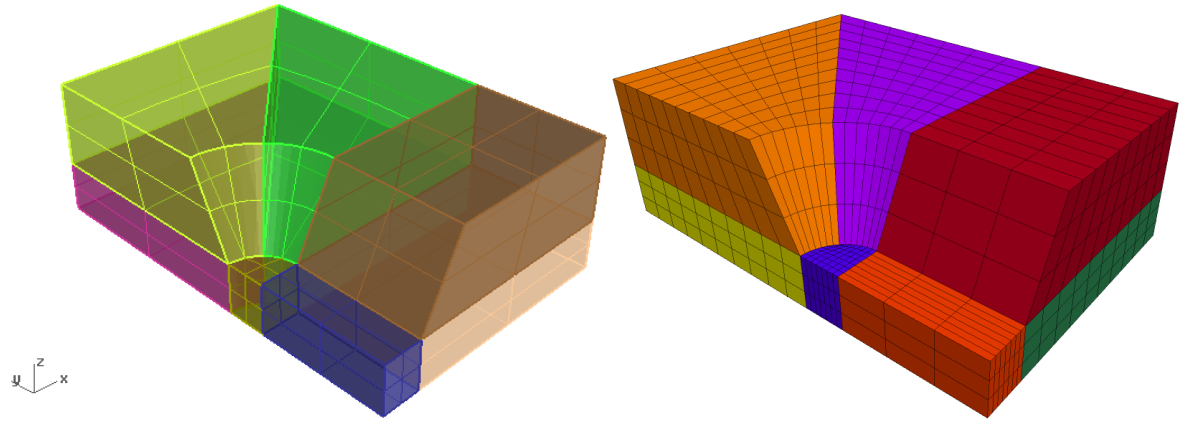


Figure 34: Solid layout resulting in eight mesh blocks.

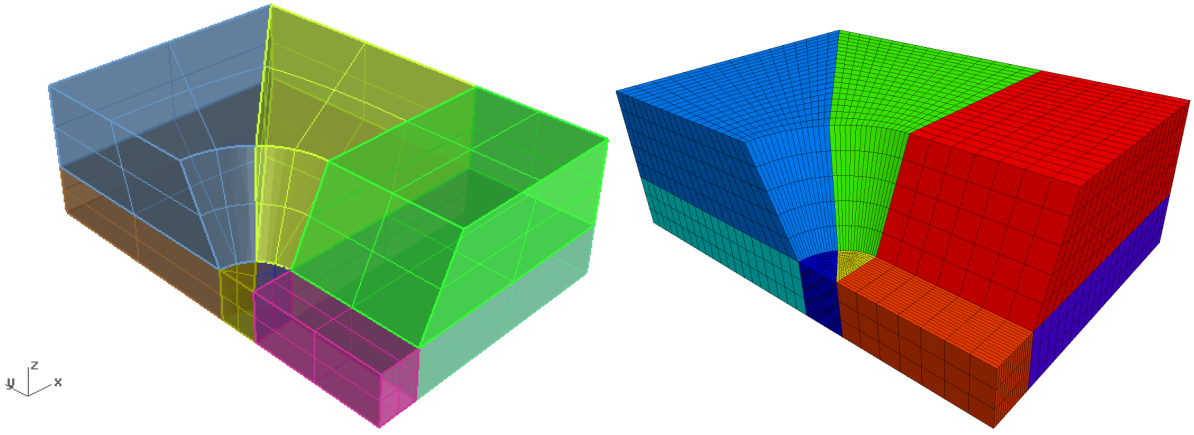


Figure 35: Alternate solid layout resulting in nine mesh blocks.

Building solids from curves

7. Click on **Osnap** (Object Snap) at the bottom of the screen (Figure 36) to highlight it. Make sure that all checkboxes designating what to snap to are checked, as shown in Figure 36. **Grid Snap** may be turned off.

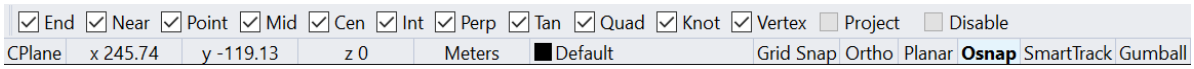


Figure 36: *Rhino* status bar.

8. Starting with the reference model presented in Figure 31, create new construction lines as shown in Figure 38. Use the **_Line** command and start by connecting nodes from the middle of the model (where the curved segments are) to the sides of the model. Pay attention to when the mouse tooltip shows "Perp" or "Perp, Int" (or other types of intersections), which means that the line under construction is perpendicular to (and intersects) another line.

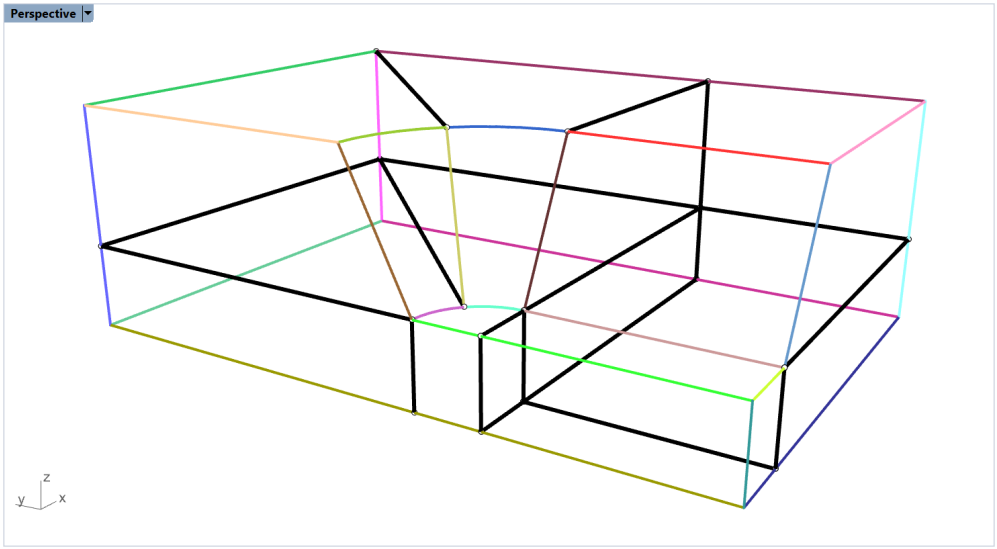


Figure 37: Creating new construction lines (in black).

The curves created in the previous step (Figure 31) are polylines, and some of them are composed of multiple linear segments that cannot be used directly to build surfaces. They must be retraced with arcs (**Curve** → **Arc**) or higher order curves (**Curve** → **Free-Form** → **Interpolate Points**) to create simpler two-noded curves.

9. Zoom in around the curved segments in the model and select in the top menu **Curve** → **Arc** → **Start, Point, End**. Create separate arcs at each curved segment of the model as shown in Figure 38 and Figure 39. In total, four arcs should be created. Pay attention to when the mouse tooltip shows "End, Int, Knot" when connecting to the points in the middle of the curved segments. Such a tooltip means the mouse is at the end of a line, hit an intersection with another line, and is at a knot.
10. After the arcs are created, the initial polylines that have been retraced can be deleted.

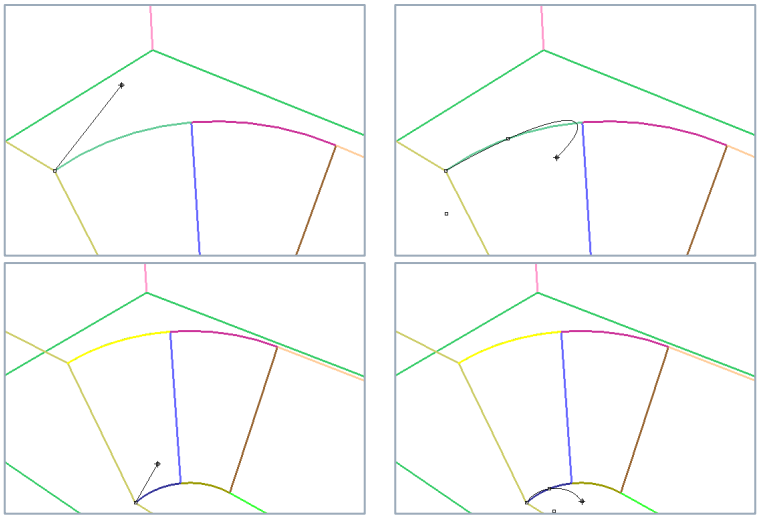


Figure 38: Building the upper (top) and lower (bottom) arcs.

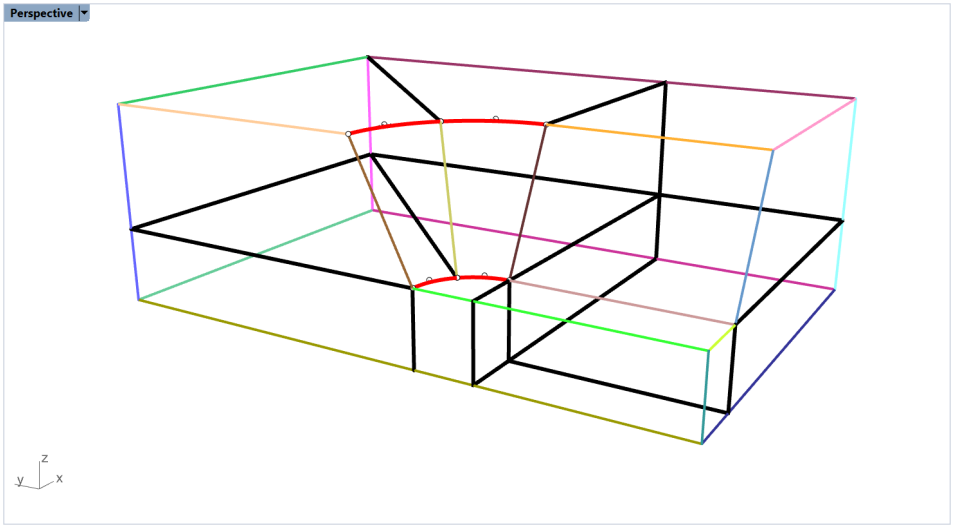


Figure 39: Red arcs retrace the original polylines at the curved section of the model.

11. Select the bottom two arcs and copy them using the **_Copy** command. For *Point to copy from*, select the left corner of the arcs as shown in Figure 40, then, within command options, click on *Vertical* to set it to *Yes*, and drag the corner down until it hits the bottom line and the mouse tooltip is as in Figure 40.

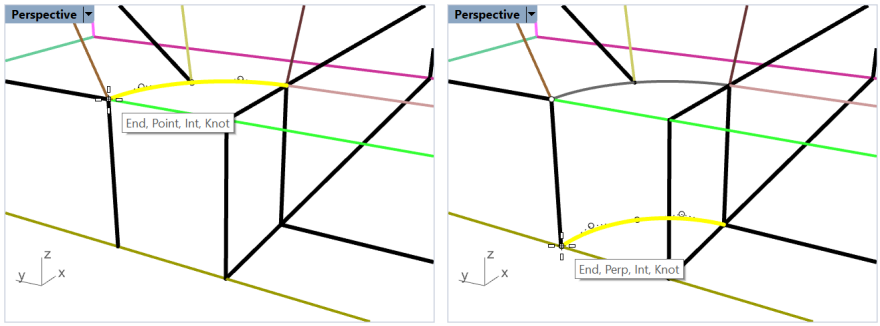


Figure 40: Copying the arcs.

12. Connect the middle nodes as shown in Figure 41 with the dark blue line. Connect the bottom end of the dark blue line to the opposite corner as shown in Figure 41 with the light blue line.

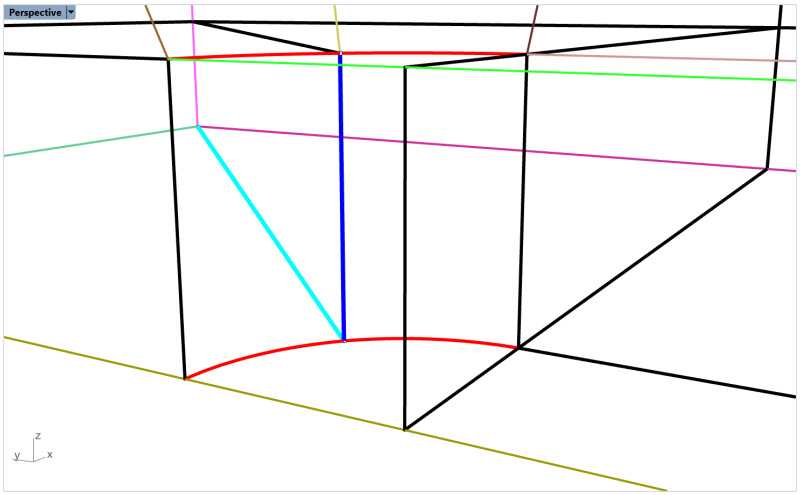


Figure 41: Connecting the middle ends of the arcs (blue lines).

13. Split all intersecting lines (mostly those from the reference model) by using the **_Split** command. For this, first select the line to split and then select a line that intersects or connects to it. For example, the tan line at the bottom of Figure 41 can be split in three pieces by the connecting black lines. The resulting wireframe should contain 56 curves (**Ctrl**+**A** and check the number of selected objects) and is the basis for creating solid blocks (Figure 42).

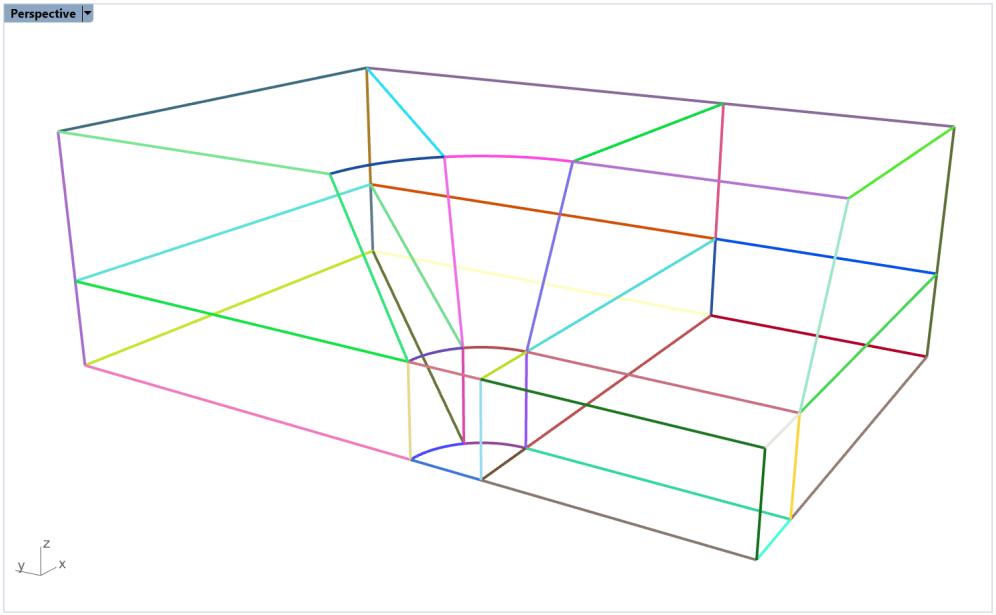


Figure 42: Model wireframe ready to be used for block creation.

14. Start creating a block at the top-left side of the model. Select the two arcs and two lines as shown in Figure 43 (selected lines are in yellow), type the **_Loft** command and use the settings shown in the right side of Figure 43. Click **OK** to build a lofted polysurface between the several curves. Alternatively, a single surface can be built between each pair of curves/lines.

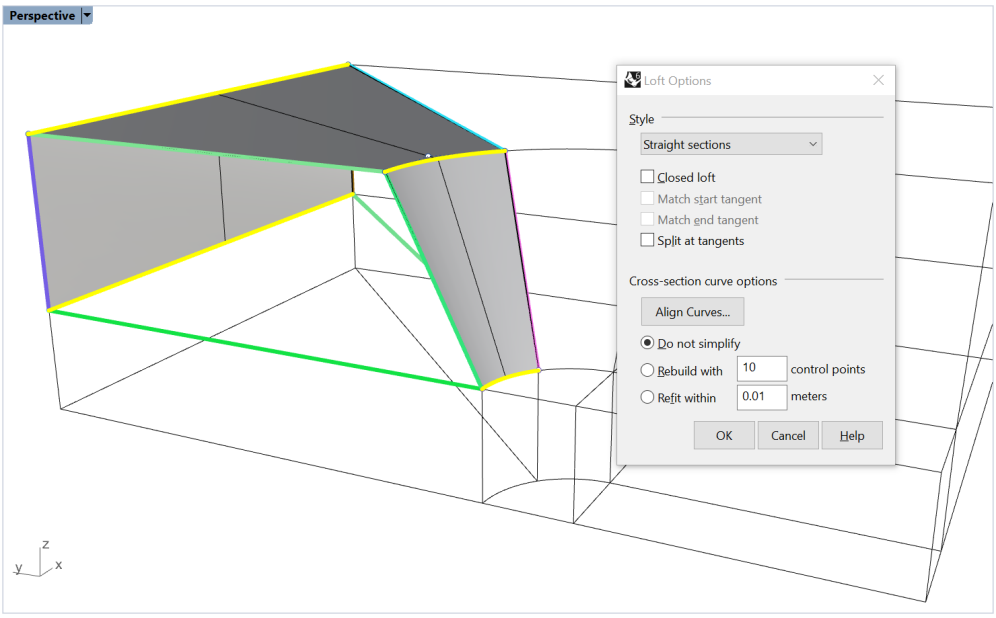


Figure 43: Building the surfaces for the first solid. Note that the grid has been hidden on this image.

15. Now, select only pairs of lines/curves composing the remaining surfaces of the first solid and **_Loft** them. In total, six sides must be created as in Figure 44.

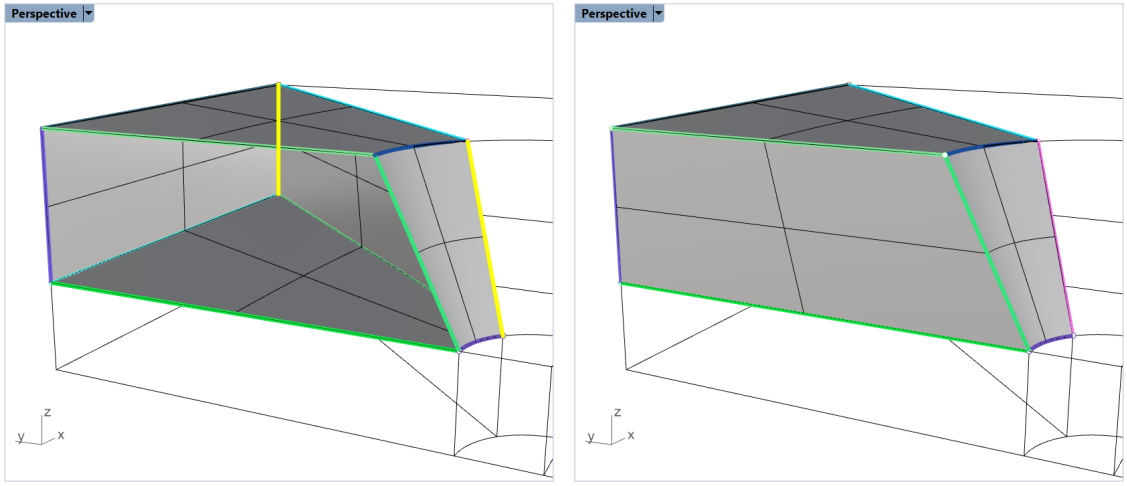


Figure 44: Building surfaces of the first solid.

16. Select all newly created surfaces and a polysurface and join them with the **_Join** command. This is the first solid block. Hide it with the **_Hide** command, as it will be easier to create other solids when nothing obstructs the view.
17. In the same manner, build all other solid blocks except for the small central block, as it requires an additional operation (described below). The stages are outlined in Figure 45.

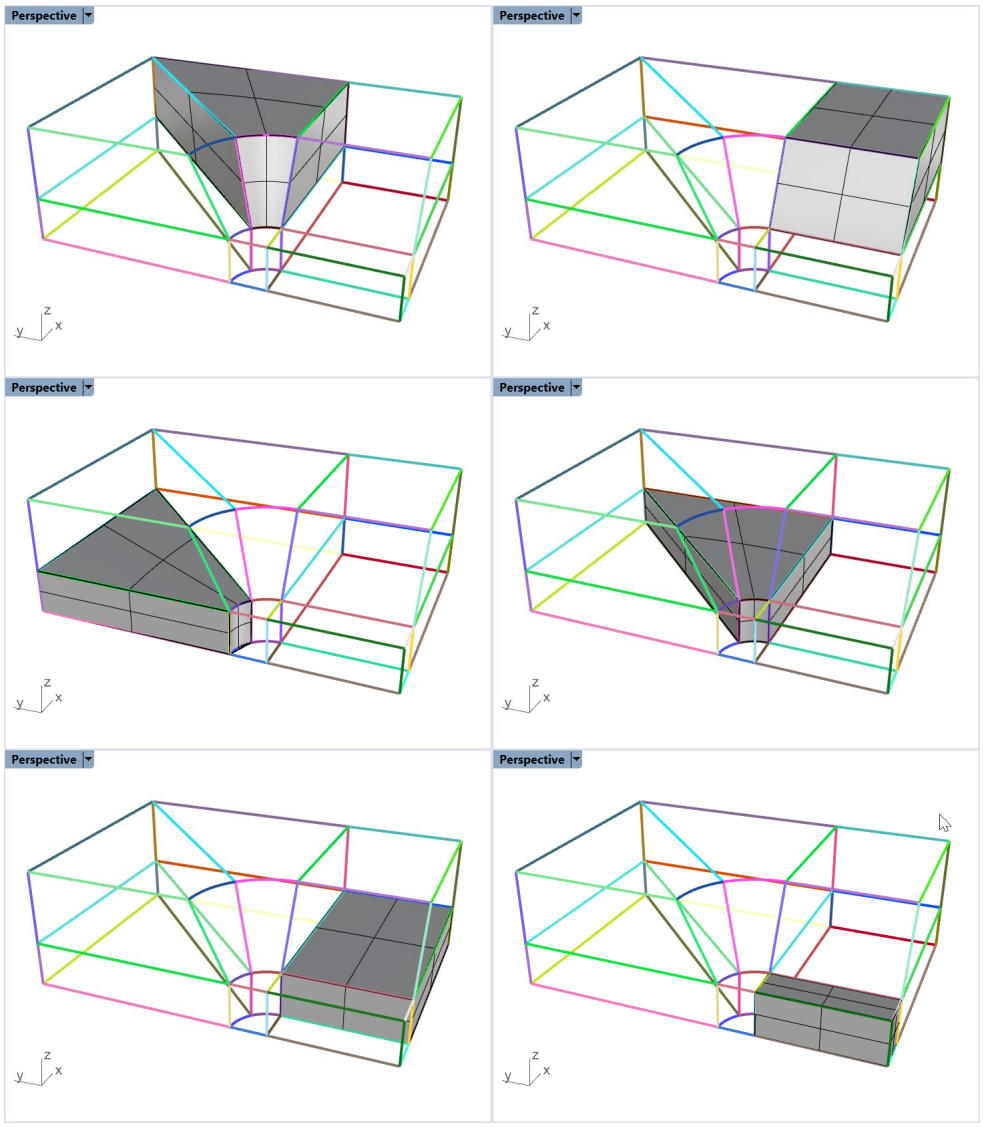


Figure 45: Building solid blocks for the model.

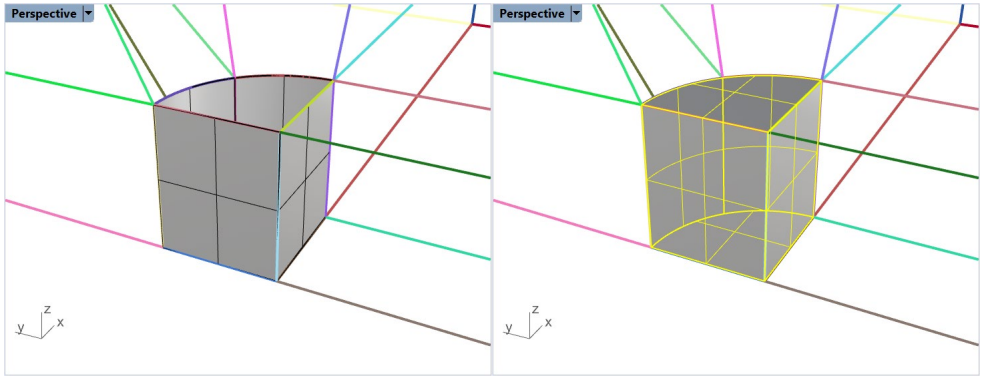


Figure 46: Building the last solid block in the model.

18. For the remaining central block, build only vertical surfaces using the **_Loft** command and join all four of them with the **_Join** command (Figure 46, left).
19. Select a single polysurface as in Figure 46, left and cap the holes with the **_Cap** command (Figure 46, right). The resulting solid is the last block needed before meshing.
20. Show all blocks that were previously hidden with the **_Show** command. There should be eight solid blocks as in Figure 47.

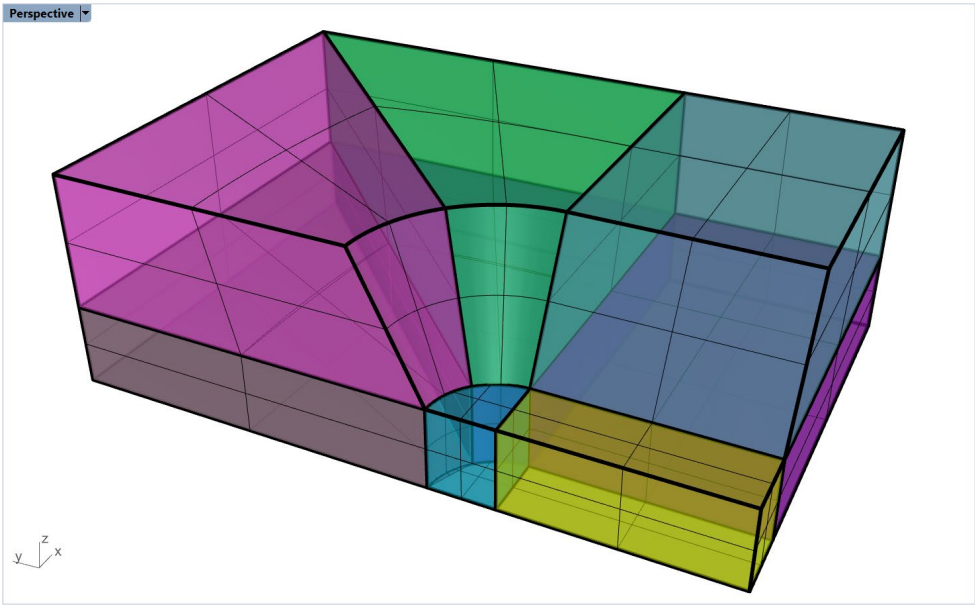


Figure 47: Solid blocks of the model (Ghosted view).

21. Select all curves by typing the **_SelCrv** command and assign layer "Reference" to them (navigate to *Properties* pane or press **F3**). Hide this layer in the *Layers* pane (press on the lightbulb to turn it off).
22. Select all remaining objects with **Ctrl**+**A** and verify that *Rhino* reports only eight polysurfaces have been selected. If any surfaces are reported, this is an indication that they were not joined to a solid block. Find and join them so that only eight closed polysurfaces (solids) are present. While the solids are selected, navigate to the *Properties* pane (or press **F3**) and ensure that object *Type* indicates that they are closed polysurfaces (Figure 48). If the *Type* field says "varies", most likely it means that some solids are not closed (probably incorrect surfaces were created).
23. Create new layers "Upper solids" and "Lower solids" in the *Layers* pane. Assign distinct colors to these layers. Select only upper solids and assign them to the layer "Upper solids" (through the *Properties* pane). Complete the same for the lower solids (Figure 49).

Now all solids are ready for meshing.

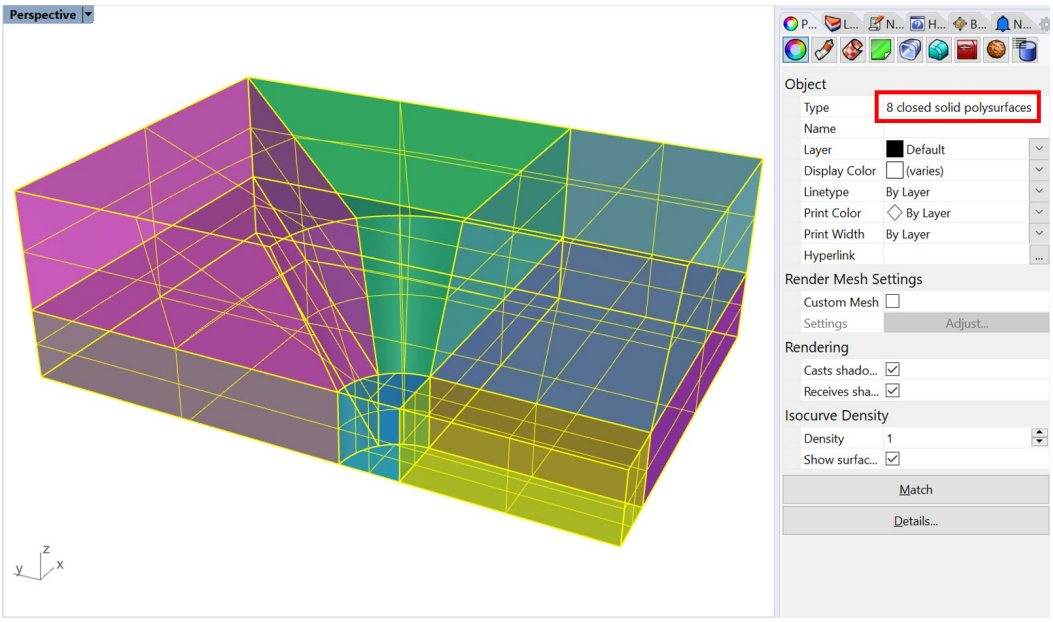


Figure 48: Solid blocks of the model (Ghosted view).

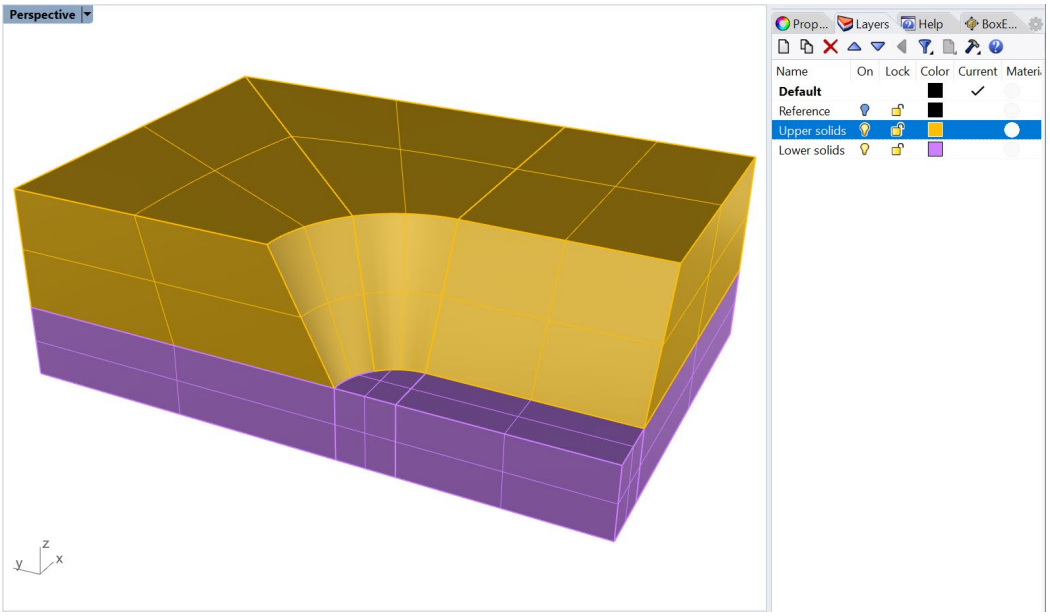


Figure 49: The model after the "Upper solids" and "Lower solids" layers have been specified.

Meshing with *BlockRanger*

24. Select all solids and type the **_BR** command or click on the  icon in the *Griddle* toolbar. Use the default ***BlockRanger*** parameters and save the mesh in *FLAC3D* binary format.

In the *Rhino* command area, ***BlockRanger*** should report: "8 solids processed, 8 solids meshed, 0 errors." If any of the solids do not satisfy the ***BlockRanger*** requirements outlined previously, the corresponding solids will remain highlighted in *Rhino* and ***BlockRanger*** will report that it processed less than 8 solids.

Figure 50 shows a structured *FLAC3D* grid created by ***BlockRanger***, as well as Zone groups from SLOT1. SLOT2 groups correspond to the original solid blocks and are shown in Figure 51.

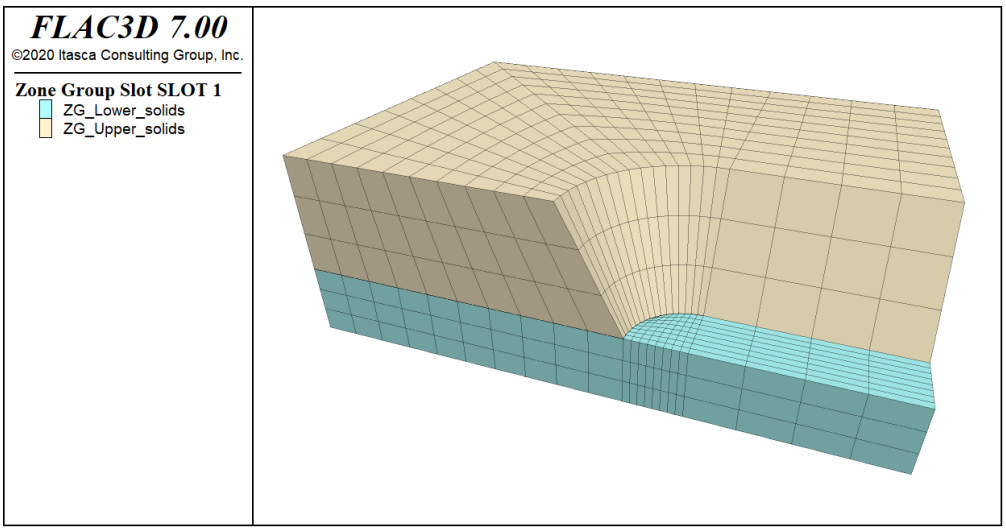


Figure 50: In *Rhino*, solids organized in layers are processed into *FLAC3D* groups in SLOT1.

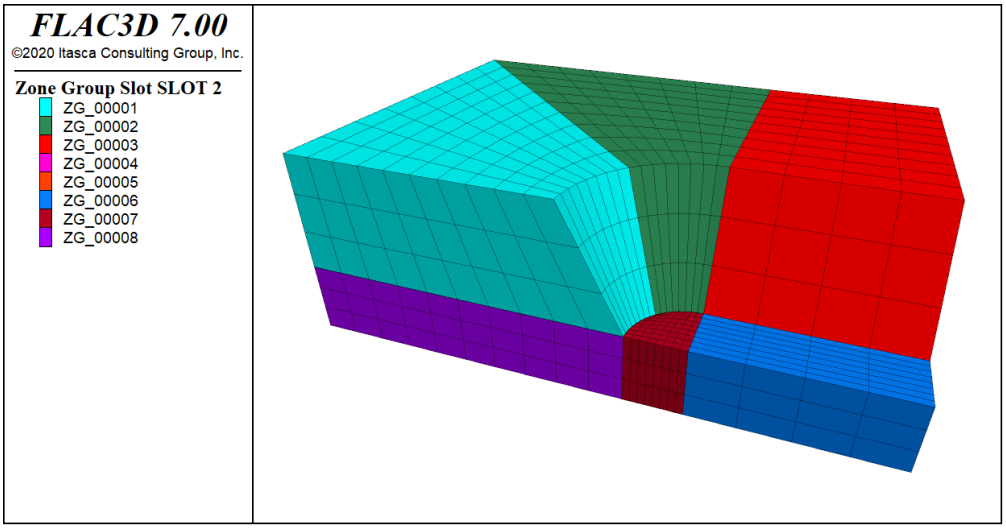


Figure 51: *FLAC3D* SLOT2 groups correspond to the individual *Rhino* solids.

As an additional exercise, change the ***BlockRanger*** meshing parameters and observe how the output mesh changes. For example, try setting *MaxEdgeLength* = 3.0 or *MaxEdgeLength* = 100.0 and *MinEdgeResolution* = 5. The simplest way to check the results is by generating bounding surface mesh with *GenerateSurfaceMesh* option. Use different choices in the option and observe how it changes the bounding surface mesh. The *ByModel* choice results in surface mesh on the boundary of the whole model only, selecting *ByLayer* adds faces the boundary between layers, and selecting *BySolid* adds faces on the boundaries between each solid.

Tutorial 5: Creation of a Hybrid Structured-Unstructured Mesh

Expected work time: 1–3 hours

This tutorial describes a methodology that can be used to create hybrid volume meshes consisting of structured and unstructured meshes. This methodology ensures that both types of meshes are perfectly connected and fully conformal.

Structured meshes often provide high-quality elements and conform better to desired mesh parameters (such as element size, type). These meshes, however, can typically be created only for regular shape objects, for example, man-made structures. On the other hand, unstructured meshes can fill objects of any shape and are well suited for meshing irregular geologic features. In this example, a tunnel excavation is filled with a structured mesh produced by ***BlockRanger***, while the surrounding rock domain containing irregular topographic and fault surfaces is filled with an unstructured mesh.

Importing the geometry

1. Start *Rhino*, select **Large Objects, Meters** (similar method as step 1 in Tutorial 1) and save the *Rhino* project at a desired location.
2. Import the initial geometry from a DXF file by navigating to **File** → **Import** and select "T5_geometry.dxf" from folder "TutorialExamples\5_HybridMesh_Tunnel". In the **Import** dialog, select **Model** → **Meters** and **Layout Units** → **Millimeters**, and keep other parameters at the default values.
3. Maximize *Perspective* viewport, select Shaded view, and zoom out to the extent of the model by pressing **Ctrl**+**Alt**+**E**.
4. In the top menu, navigate to **View** → **Display Options…** This will open the **Rhino Options** dialog; select **View** → **Display Modes** →**Shaded** → **Objects** → **Curves** and set **Curve Width** to 4. This will display all curves thicker in the Shaded view. The initial settings can always be restored by clicking on **Restore Defaults**.

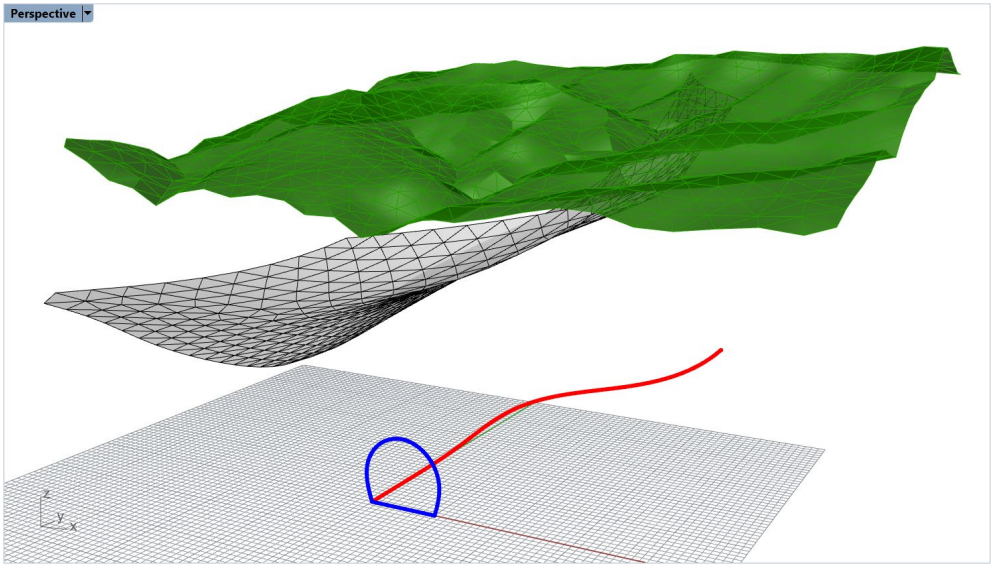


Figure 52: DXF geometry imported to *Rhino* corresponding to the tunnel profile and excavation direction.

5. Navigate to the *Layers* pane and delete all layers except for “Default”, “Direction”, “Topo”, “Fault”, and “TunnelProfile”.

The imported geometry contains two meshes, one corresponding to the topographic surface and one corresponding to a large fault. Note that the fault almost reaches to the topo surface but does not intersect it. This will need to be fixed, as the fault is supposed to intersect the surface. The geometry also contains a horseshoe-shaped curve (in blue) corresponding to the tunnel profile and a red curve corresponding to the tunnel excavation direction. The tunnel must be constructed along the red curve in such a way that its cross-section is always perpendicular to the curve. Note that the front end of the red curve is already aligned with the bottom-left corner of the tunnel profile.

Construction and meshing of the tunnel with structured mesh

6. Turn off layers “Topo” and “Fault” to leave only the profile and direction curves visible.
7. Tunnel profile consists of several curves (in blue). Select all of them and type **_Join**.
8. Select the red curve (excavation direction) and type the command **_Sweep1**. For cross-section curves, select the single blue curve (tunnel profile). When offered to *Drag seam point to adjust*, drag it from the bottom-left corner of the tunnel profile to the topmost point of the profile and press **Enter** (maximize *Front* viewport for ease of dragging the seam point; this can be done while the command is active; Figure 53). It is important to move the *seam point* from the corner to create the proper geometry suitable for ***BlockRanger***. For other options in the **_Sweep1** command, use those shown in Figure 54.

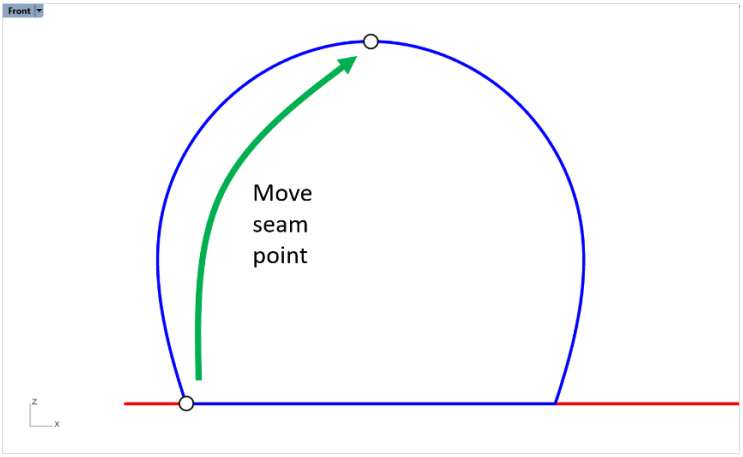


Figure 53: Drag seam point to the top of the tunnel profile.

9. Create new layer “Tunnel” and assign a newly created tunnel surface to it.
10. Select the tunnel profile curve (blue) and create the tunnel cross-section surface out of it by using the command **_PlanarSrf**.
11. Select the initial cross-section surface and use command **_ArrayCrv** to duplicate it along the direction path (red curve) with a spacing of roughly 5 m; this will create excavation stages that can be gradually removed during numerical simulations (Figure 55). In the command options pop-up window, set *Distance between items* = 5, *Orientation* = *Freeform*.
12. Turn off layers “TunnelProfile” and “Direction” as they will no longer be needed.

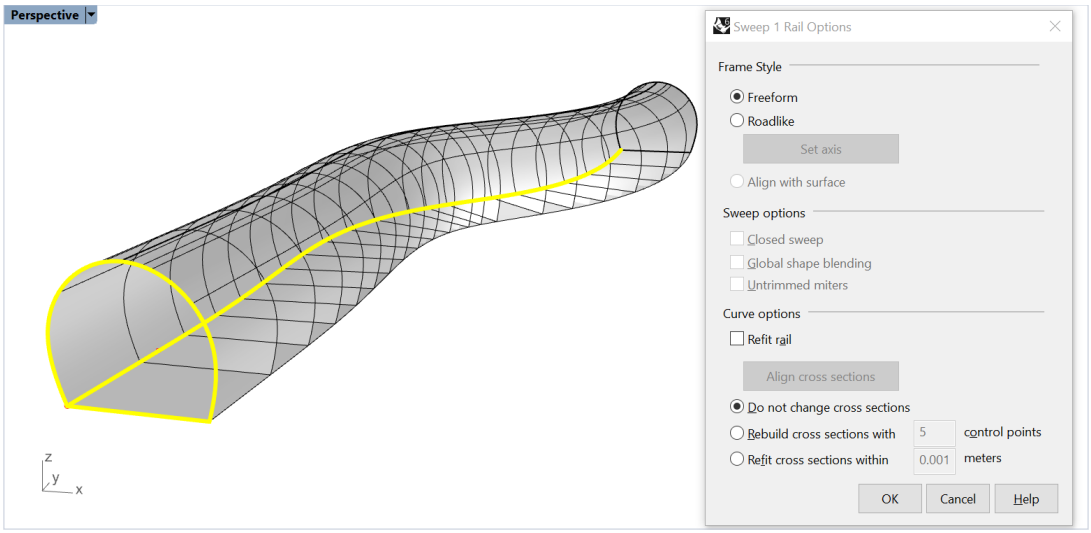


Figure 54: Creation of tunnel surface by sweeping the profile curve along the direction path.

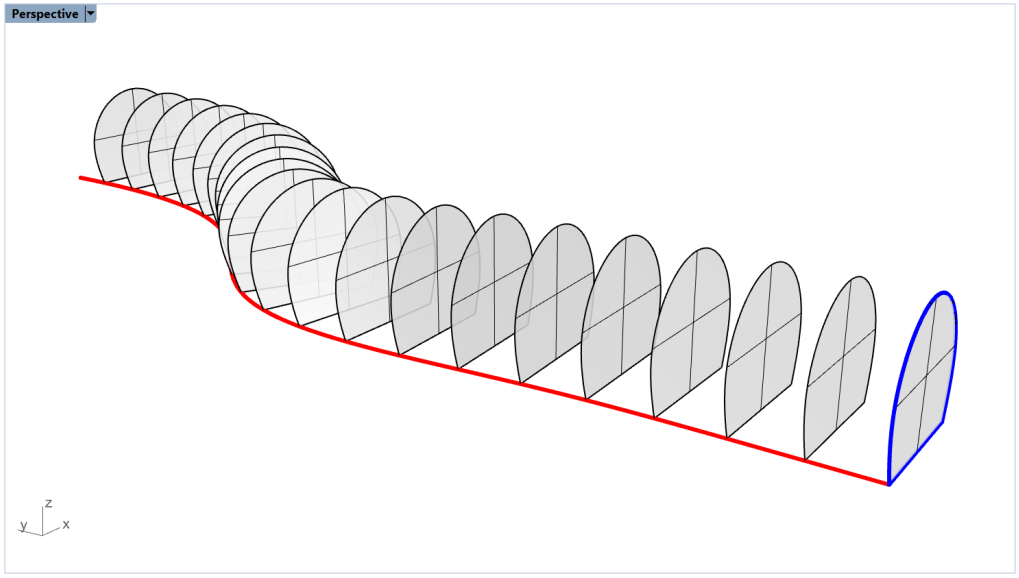


Figure 55: Creation of excavation stages by duplicating initial cross-section surface. Tunnel surface is not shown.

13. Split the tunnel surface with all the cross-section surfaces by selecting the tunnel surface only, then using the **_Split** command and selecting all other surfaces (**Ctrl+A**). A message stating that "One polysurface split into 21 pieces." should appear in *Rhino*'s command area.
14. Turn off layer "Tunnel". Only the tunnel cross-sections should be visible. Delete all of them (21 objects).
15. Turn the "Tunnel" layer back on. Select all 21 polysurfaces and use the **_Cap** command to close the front and back of each excavation stage. A *Rhino* notification should appear that 42 caps have been created. Now each excavation stage is represented by a separate watertight polysurface (or by a solid), which can be easily meshed using ***BlockRanger***.
16. Colorize all stages with *Griddle*'s **_ColorizeObjects** command (Figure 56).

Before proceeding with meshing the tunnel, some additional work is needed to trim parts of the tunnel that extend past the boundaries of the model.

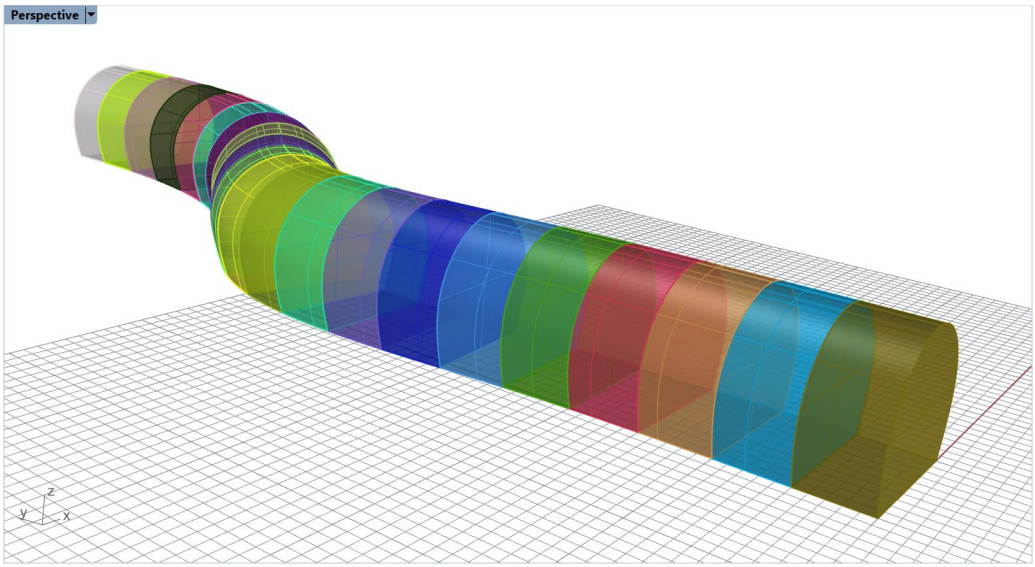


Figure 56: Watertight polysurfaces (solids) representing each excavation stage of the tunnel.

17. Turn off the “Tunnel” layer and turn on the “Topo” layer to show the topographic surface mesh.
18. Create a planar surface underneath the topo mesh by using the **_Plane** command and selecting the *Center* option to specify the center of the plane and its size.
    - *Center of plane*: 0,50,-25
    - *Other corner or length*: 150
    - *Width*: press **Enter**
19. Select the topo mesh and duplicate its border using the **_DupBorder** command.
20. Select the border curve and extrude it downward using the **_ExtrudeCrv** command in such a way that all parts of the extrusion go below the plane (Figure 57). This will create a polysurface that intersects the horizontal plane. Delete the duplicated border.

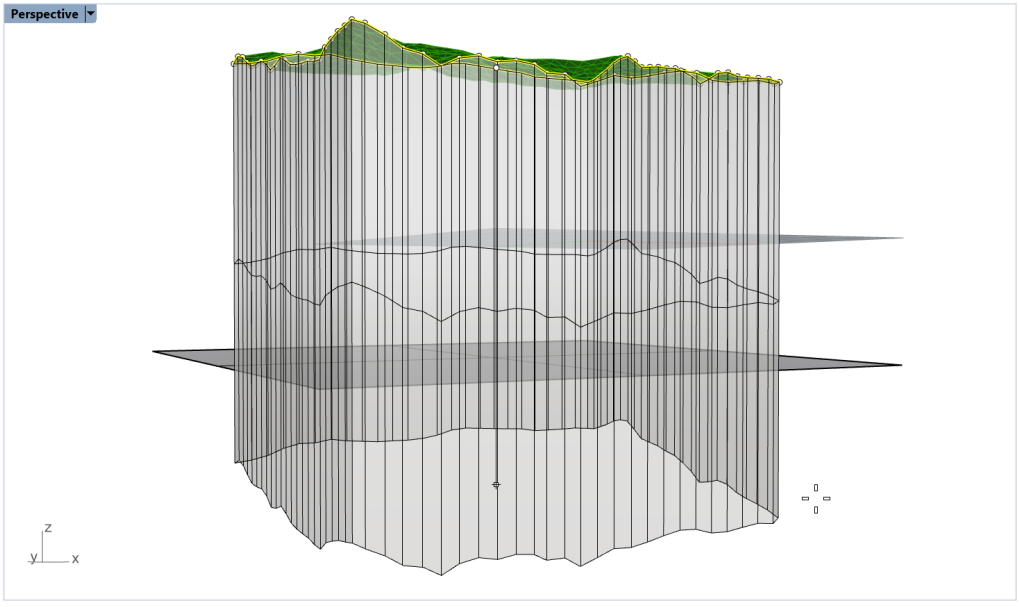


Figure 57: Extrusion of topographic surface border.

21. Select the extruded polysurface and split it by the horizontal plane using the **_Split** command. Delete the bottom piece of the polysurface.
22. Select the planar surface and split it by the remainder of the extruded polysurface. Delete the outer piece of the planar surface.
23. Join the bottom surface with the extruded polysurface (**_Join** command) to create a closed domain (together with the topo mesh), which will serve as the modeling domain after meshing.
24. Create new layer "Domain" and place the newly created polysurface in it (by changing the layer assignment in the polysurface properties).
25. Turn on the "Tunnel" layer and note that the front and back sections of the tunnel stick out from the domain (Figure 58). Split them by the domain and delete excessive parts.

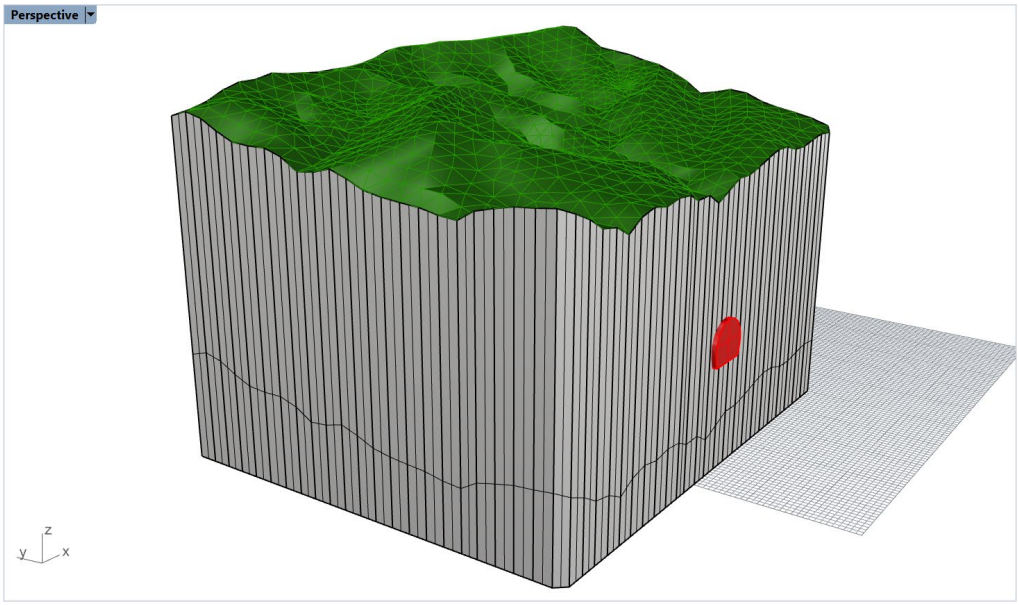


Figure 58: The front part of the tunnel extends through the modeling domain.

26. Hide layers "Topo" and "Domain" so only the tunnel stages are present.
27. Now the front and the back of the tunnel contain openings. Select the front and back tunnel stages and cap them using the **_Cap** command to create watertight solids. At this point, the tunnel is ready for meshing.
28. Select all 21 solids corresponding to the tunnel excavation stages and type the **_BR** command or click on the  icon in the *Griddle* toolbar. Use all default ***BlockRanger*** parameters and set *GenerateSurfaceMesh* = *ByModel*. ***BlockRanger*** will create a structured volume mesh for the tunnel and will output it at the user-specified location (keep *FLAC3D* binary format as output format). ***BlockRanger*** will also create surface mesh corresponding to the tunnel's external surface and will output it into *Rhino* (Figure 59). This surface mesh will be used to create an unstructured volume mesh outside the tunnel.
29. Create new layer "TunnelMesh". Change the layer of newly created surface mesh from "Default" to "TunnelMesh". Now turn off layer "Tunnel", leaving only one layer active as shown in Figure 59.

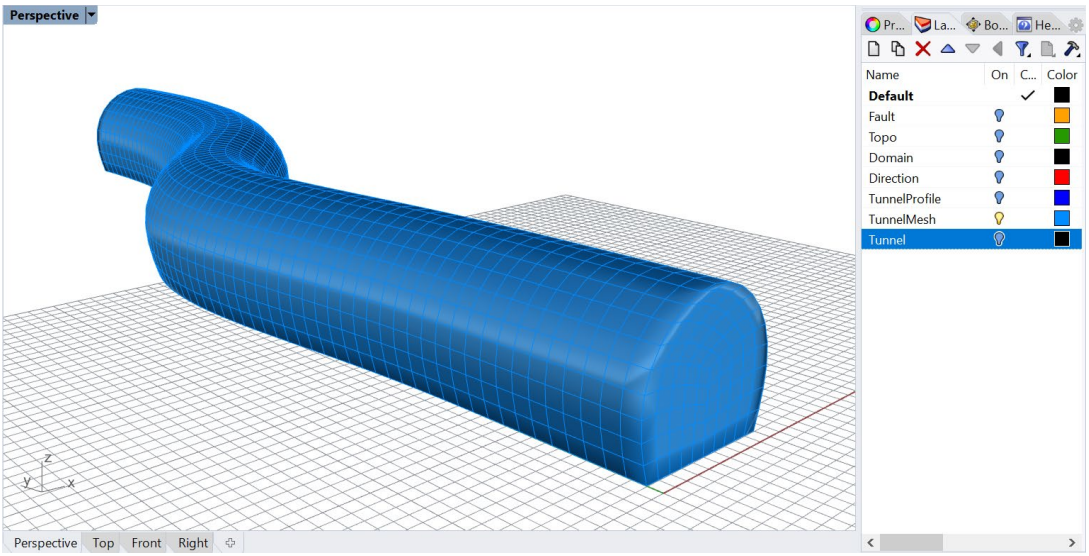


Figure 59: Surface mesh corresponding to the exterior of the tunnel's structured volume mesh.

Meshing the tunnel exterior with unstructured mesh

To create an unstructured domain mesh that is conformal with the tunnel structured mesh, bounding surface mesh presented in Figure 59 will be used.

30. First, the front and back caps of the bounding surface mesh should be removed to leave only the "shell" of the tunnel. Type the **_GExtract** command or click on the  icon in the *Griddle* toolbar. Select option *AllSurfaces* and use *MaxBreakAngle* = 85° to separate the caps. Select and delete them afterwards. The tunnel mesh should be hollow inside.
31. In the *Layers* pane, right-click on layer "TunnelMesh" and select *Duplicate Layer and Objects*. The next few steps will cause changes in the tunnel mesh, but volume meshing (the final step) requires the original mesh. Therefore, the duplicated tunnel mesh is used in the next steps. Turn off layer "TunnelMesh" and keep the duplicated layer on.
32. Select the tunnel mesh and use the command **_DupBorder** to duplicate its open boundary.
33. Turn on layer "Domain". Select the domain polysurface and split it with the duplicated tunnel boundaries by typing the **_Split** command and selecting the curves only.
34. Delete parts of the domain corresponding to the tunnel front and back to create openings for the tunnel (Figure 60).
35. Select the domain and create the initial mesh for it using the **_Mesh** command. Use *Simple Controls* and select the middle position with the slider. While the polysurface is still selected, delete it. The initial mesh is not very good, but it will be remeshed in future steps.
36. Turn on the "Fault" layer. Note that the fault does not fully intersect the domain and the topo meshes (turn "Topo" layer on and off to see that). To extend the fault mesh to the domain boundaries, select it and type the **_GExtend** command or click on the  icon in the *Griddle* toolbar. Select option *ExtendAllBoundaries*, specify *ExtendLength* = 10, and keep *MeshType* = *Merged*. The command will extend the mesh by adding new faces, and the extended mesh will fully intersect the domain and topo meshes. Turn on the "Topo" layer. The result should look similar to Figure 61.
37. Select all objects with **Ctrl**+**A** and use ***GInt*** to intersect all meshes. Use a tolerance of 0.001 and keep all other parameters at defaults.

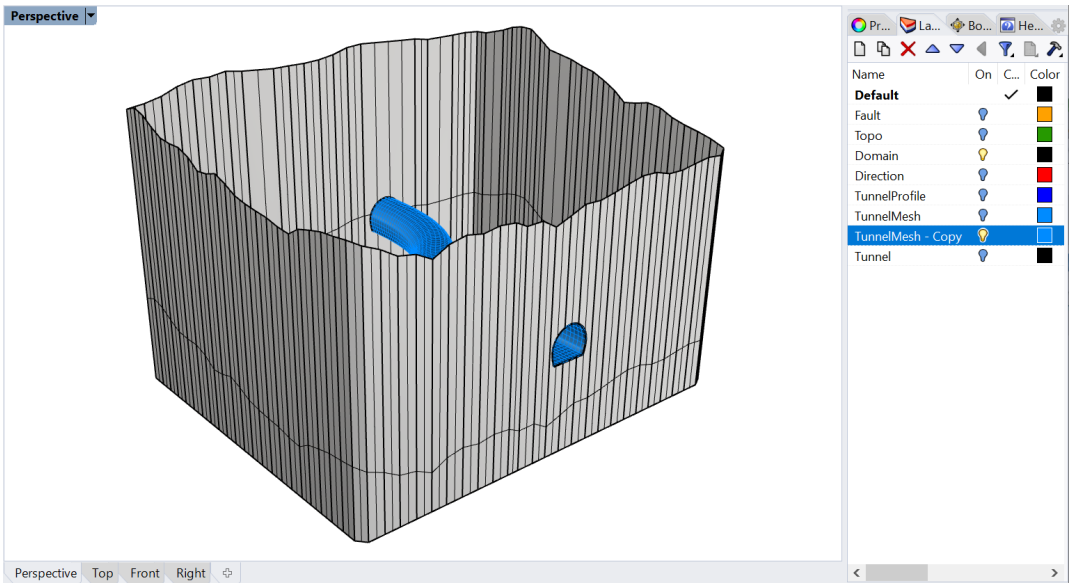


Figure 60: Removing parts of the domain at tunnel front and back.

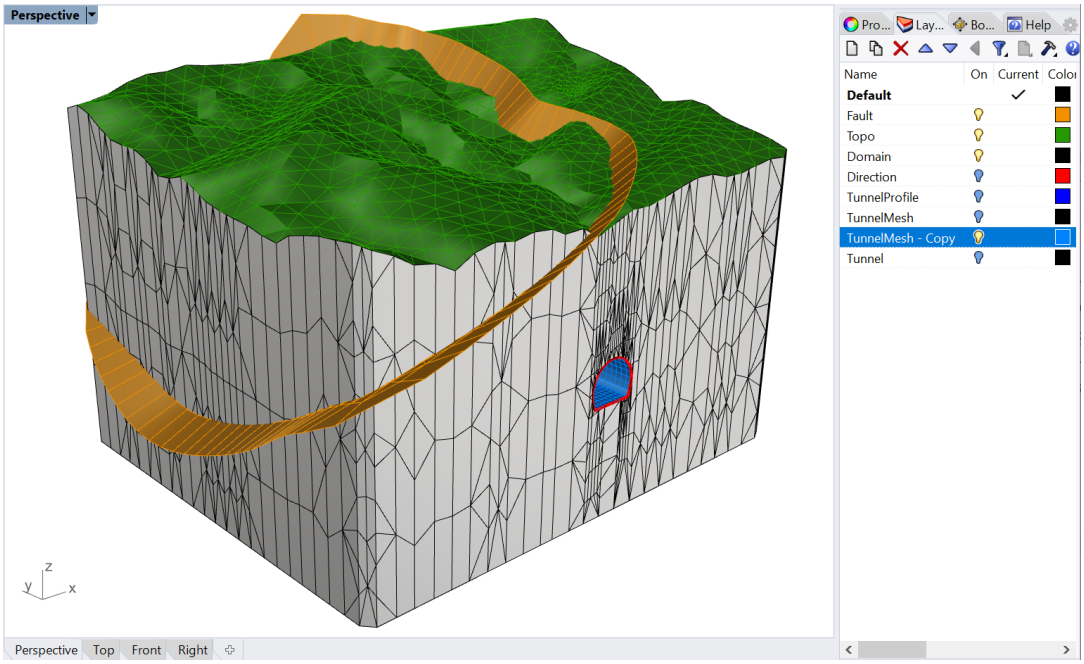


Figure 61: Extended fault mesh intersects domain and topo meshes. Note that the colors of topo and tunnel meshes and duplicated tunnel borders (curves) are changed from default.

38. Delete the copy of the tunnel mesh (blue mesh in Figure 61) or delete the whole layer containing it.
39. Select all objects again with **Ctrl**+**A** and use ***GSurf*** to remesh the meshes. Because the duplicated tunnel borders are selected as well (red curves in Figure 61), they will serve as hard edges to preserve conformity with the tunnel mesh. Use the following parameters for ***GSurf***: *Mode* = *QuadDom*, *MinEdgeLength* = 1, *MaxEdgeLength* = 10, *RidgeAngle* = 40, and keep all other parameters at default values.
    After ***GSurf*** completes remeshing, the duplicated tunnel borders stay selected. Delete them, as they are no longer needed (Figure 62).

Turn on layer “TunnelMesh” and zoom in to the tunnel front or back (Figure 63). It can be easily seen that both tunnel and domain meshes are conformal (nodes and edges match along the border). This is due to using ***GInt*** to create proper initial intersections between the domain and tunnel meshes and due to using the border of the original tunnel mesh as a hard edge during remeshing.

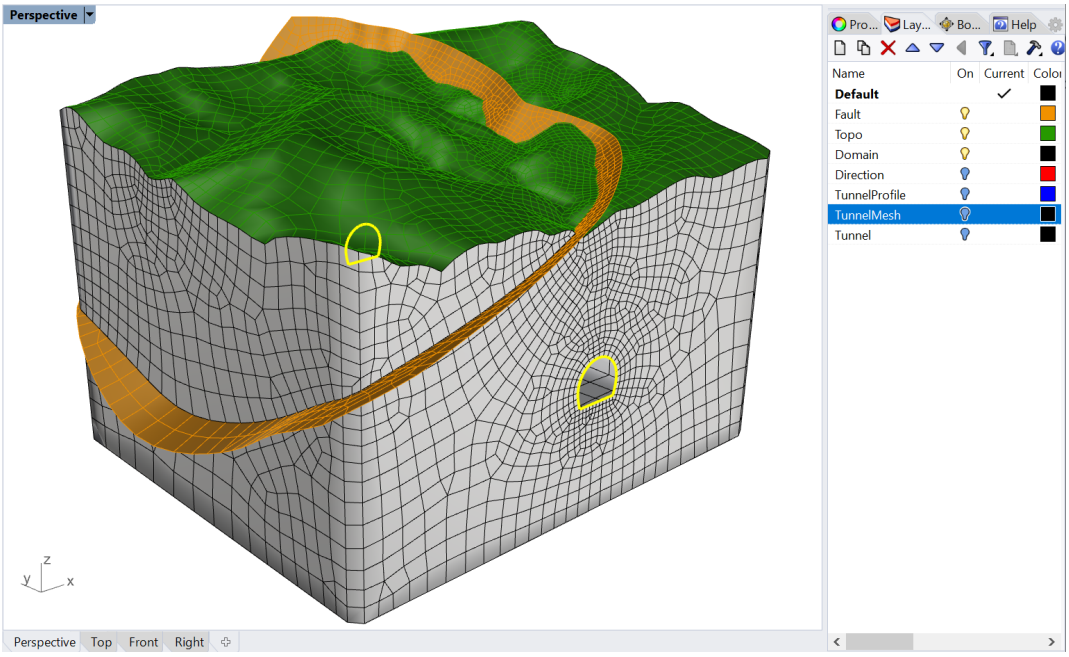


Figure 62: Result of remeshing of all meshes (except for the tunnel mesh) while preserving mesh edges and nodes along the tunnel borders (hard edges).

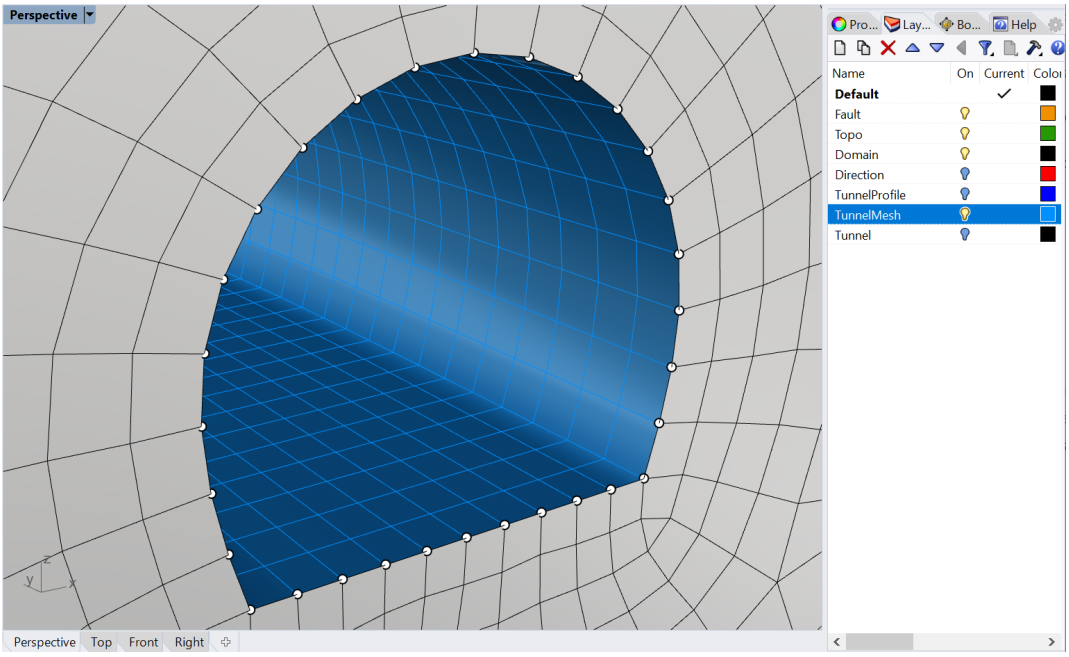


Figure 63: Close view of the tunnel front showing that tunnel and domain meshes are conformal.

40. Assign names to each of the meshes in the object *Properties* pane (click on a mesh and press **F3**), e.g., name fault "Fault", tunnel mesh - "Tunnel", etc. These names will be passed as surface names within the volume mesh.

Now the model is ready for volume meshing with ***GVol***. Even though parts of the fault mesh extend outside the watertight domain, volume meshing can be carried out. ***GVol*** typically marks parts of surface meshes causing meshing errors with red outlines; in this case, however, red outlines will simply indicate that the "external" mesh faces were ignored during the meshing process.

41. Select all meshes and use ***GVol*** to create an unstructured volume mesh for the exterior of the tunnel. For parameters, use *HexDom* mesh and output in *FLAC3D* binary format; keep all other parameters at defaults. Save the output file at the same location as the tunnel structured mesh (do **NOT** overwrite it). Note that the volume mesh will not be created inside the tunnel this time, as the interior part of the tunnel is not a watertight domain; only the exterior of the tunnel constitutes a watertight domain.

Two meshes (grids) have been created in this example: a structured volume mesh with stages corresponding to the interior of the tunnel and an unstructured volume mesh corresponding to the exterior of the tunnel (domain). Both meshes can be loaded in *FLAC3D* (or any other code, if another format was used) and used for numerical modeling. Because the meshes are conformal and faces, edges, and nodes are duplicated along the boundary of the tunnel, *FLAC3D* can merge the meshes to avoid using attach conditions. After importing both meshes, use the merge command:

```
flac3d>zone gp merge
--- 2610 gridpoints merged and 5160 surface faces removed.
```

The result is shown in Figure 64.

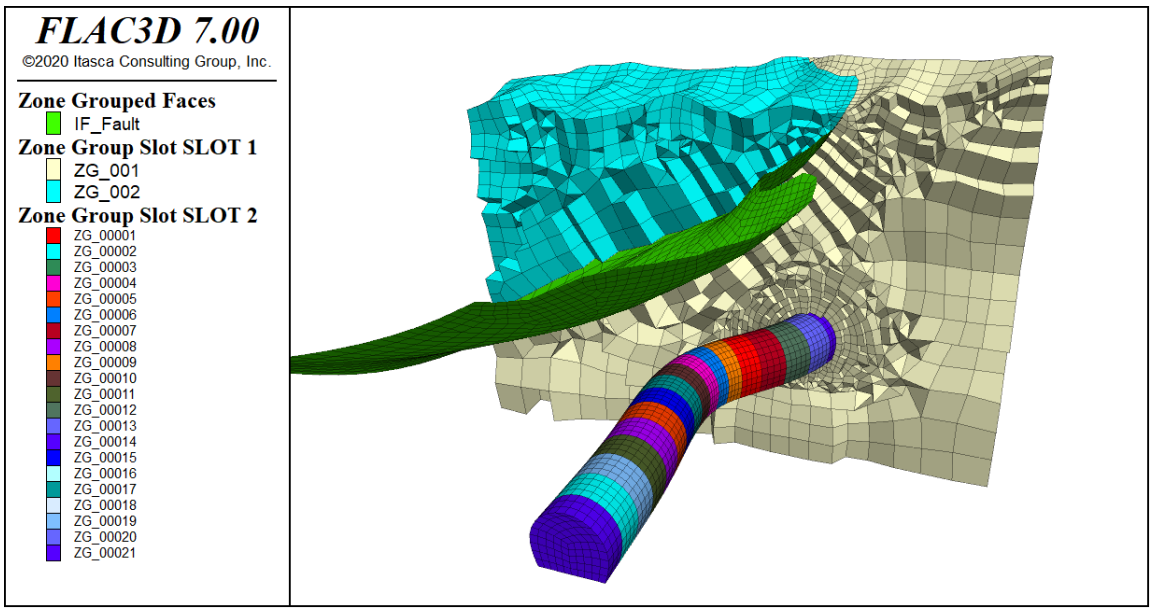


Figure 64: Structured and unstructured meshes loaded in *FLAC3D*.

Tutorial 6: Mesh Clean-up and Rebuilding

Expected work time: 1 hour

This tutorial provides an example of a model that requires initial mesh cleaning and fixing before further model operations can be done. The model provided as a DXF file contains an existing triangular surface mesh defining drifts in a mine and a topographical surface (Figure 65). Such surfaces often come from minimally processed field measurements by various surveillance systems, and they often consist of triangular meshes, which generally are not clean (often non-conformal, contain overlapping and duplicate faces, missing faces, etc.) Thus, they typically need to be cleaned and remeshed to produce surface meshes suitable for volume meshing.

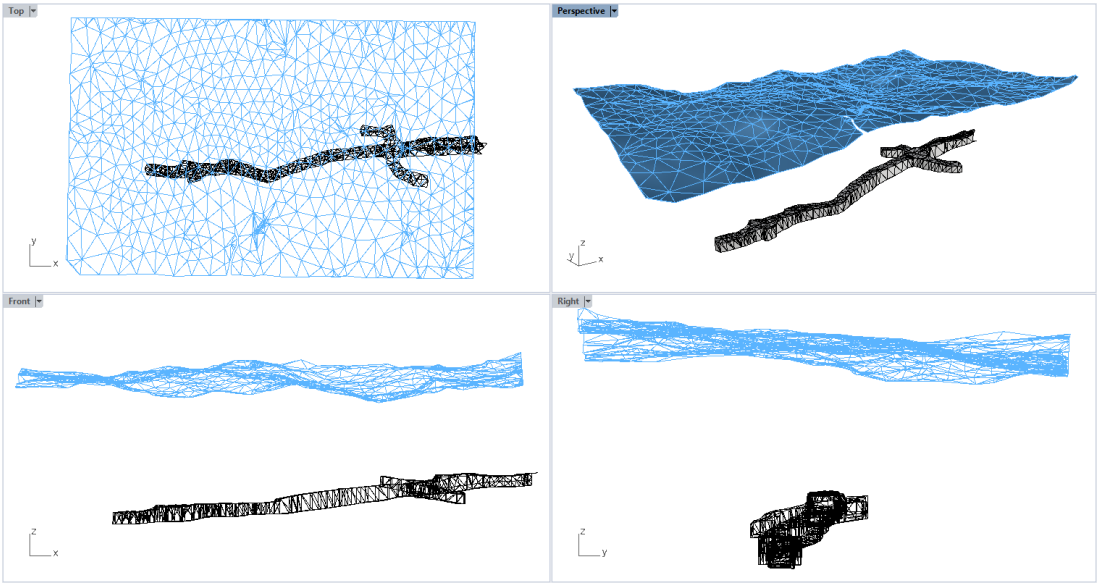


Figure 65: Drift geometry from a mine.

Importing the geometry

1. Start *Rhino*, select **Large Objects, Meters** (similar method as step 1 in Tutorial 1) and save the *Rhino* project at a desired location.
2. Import the initial geometry from a DXF file by navigating to **File** → **Import** and select "T6_drifts.dxf" located in folder "TutorialExamples\6_MeshCleanup_Drifts". In the **Import** dialog, select **Model** → **Meters** and **Layout Units** → **Millimeters**, keep other parameters at default values.
3. If any of the viewport is maximized, double-click on the viewport icon to restore all four views as in Figure 65.
4. Imported objects are located far from the origin and may not be visible. Zoom out to the extent of the model by pressing **Ctrl**+**Alt**+**E**. Click on the topographic and drift excavation meshes to ensure that they are in separate layers. Delete all layers except for "Default", "EXCAVATION", and "TOPO".
5. Hover the mouse around the model and pay attention to the coordinates in the lower-left corner of the information pane: the displayed coordinates are far from the origin (zero coordinates). This is problematic as it limits the number of significant digits available for geometrical manipulations and for numerical analysis. It is good practice to translate the model to locate it around the origin.

6. Select all objects (**Ctrl** + **A**) and type the **_Move** command. In the *Top* viewport select the top-left corner node as the point to move from and type 0 for the point to move to. Press **Ctrl**+**Alt**+**E** to zoom to the new model extents.
7. Maximize *Perspective* viewport and select Shaded view (Figure 66).
8. In the top menu, navigate to **View** → **Display Options…** This will open the **Rhino Options** dialog; select **View** → **Display Modes** →**Shaded** → **Objects** → **Curves** and set **Curve Width** to 4. This will display all curves thicker in the Shaded view. The initial settings can always be restored by clicking on **Restore Defaults**.

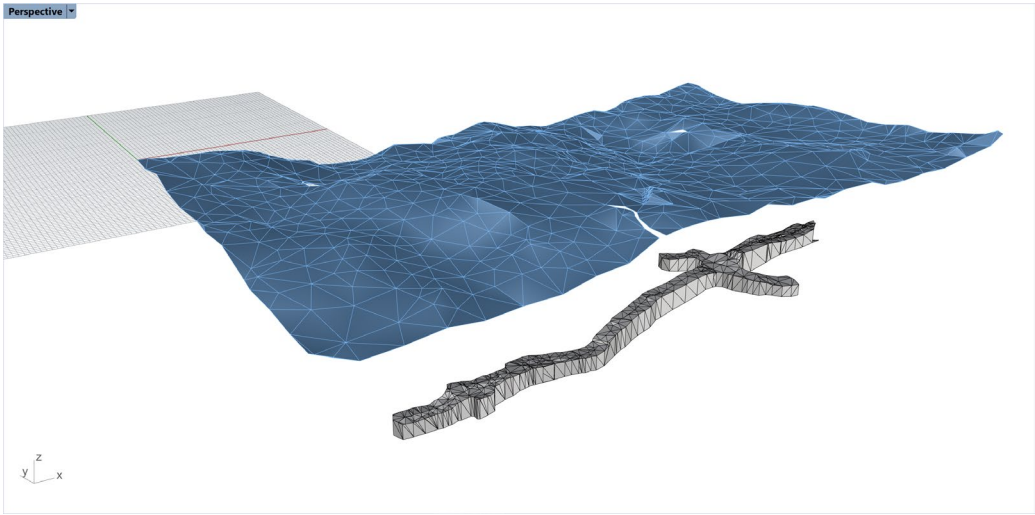


Figure 66: Drift model located near zero coordinates.

Mesh clean-up

Two meshes imported from the DXF file contain a number of issues, such as holes, nonconformal faces, duplicate faces, naked edges, etc. Such surface meshes are not adequate for creating a valid watertight mesh domain suitable for volume meshing. The meshes need to be cleaned up first.

Rhino has some built-in utilities that allow for checking and summarizing surface mesh issues. One such utility can be called by typing the **_Check** command (try this by selecting all meshes; a pop-up window will report a summary of various issues). Another useful command is **_MeshRepair**. However, using that utility for numerous issues may be tedious and cumbersome. *Griddle* offers the command **_GHeal**, which identifies and displays major surface mesh problems (that otherwise would prevent volume meshing) and attempts to fix numerous issues at once.

9. To check the meshes for naked edges (holes, cracks, disjointed faces) and clashing faces (non-conformal faces within a single mesh), type **_GHeal** and select *ShowErrors* (Figure 67). Two new layers will appear: “NAKED_EDGES” (pink curves) and “CLASHING_FACES” (red curves), which contain curves outlining corresponding issues. In this case, the layers contain many curves. After mesh repair operations, the number of issues (and corresponding curves) will decrease, which can be checked by calling **_GHeal** → *ShowErrors* again (the outlines of the issues will be updated). To

clearly visualize only problematic parts, layers “EXCAVATION” and “TOPO” can be turned off in the *Layers* pane (Figure 68).

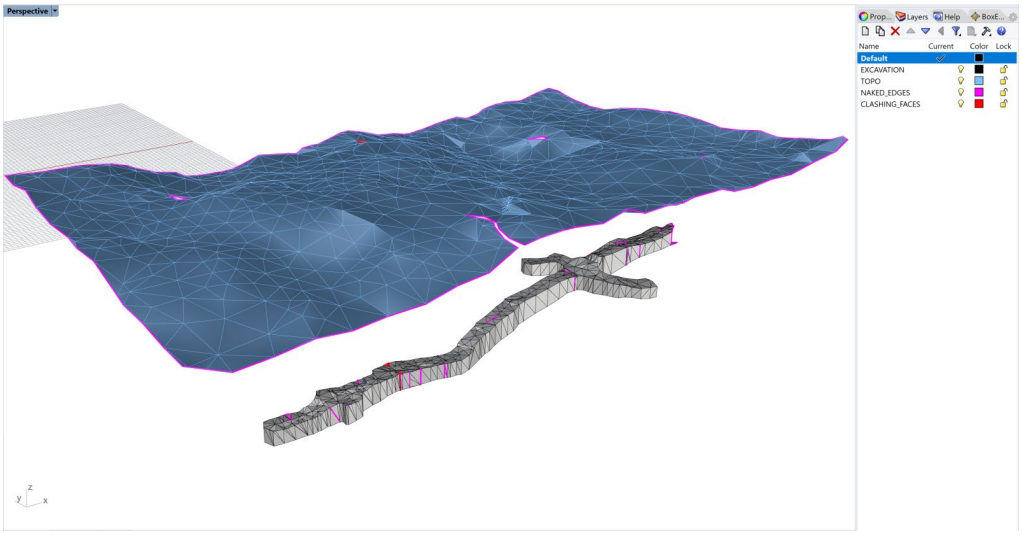


Figure 67: Displaying naked edges and clashing faces.

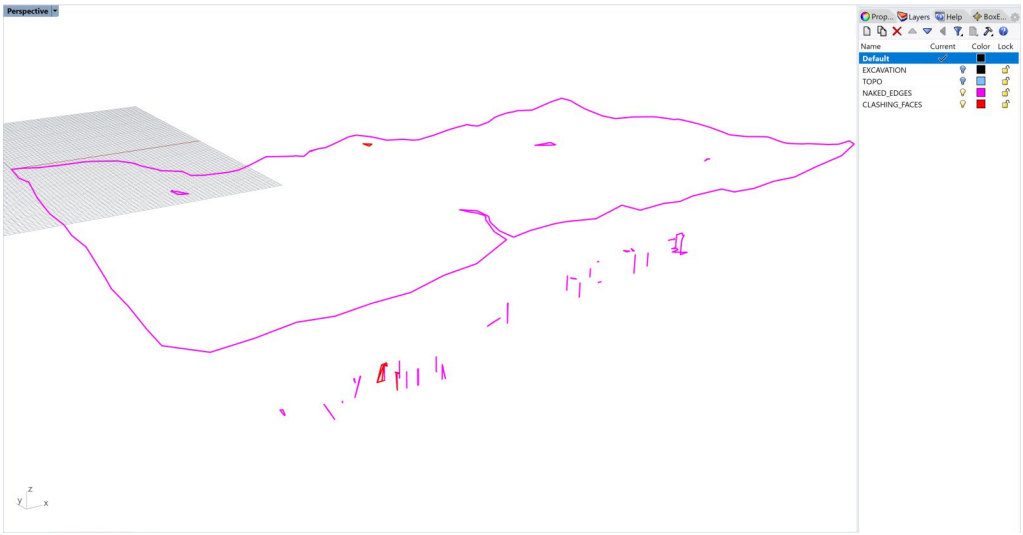


Figure 68: Displaying naked edges and clashing faces with meshes hidden.

10. To start fixing the meshes, first intersect all non-conformal faces within the meshes using ***GInt***. Set tolerance to 0.002 and keep all other parameters at the defaults. After mesh intersections, update the outlines of the issues by calling **_GHeal** → *ShowErrors*. Note that there are no more clashing faces, layer “CLASHING_FACES” is empty, and fewer naked edges remain.

The remaining issues can be fixed using ***GHeal*** in automatic mode with the *AutomaticHeal* option (the manual mode option simply calls *Rhino*’s **_MeshRepair** command). *AutomaticHeal* allows for fixing numerous issues at once, but it is important to note that ***GHeal*** should be applied to problems discretely: only select to fix issues that are expected to be present. In certain cases, it is even better to fix issues one by one to get the best results. For example, first *FillHoles* and *FixClashingFaces* only, then *MendCracks* and then *AlignNormals*. In this model, there are holes and a “crack” in the topographic

mesh, and there are holes and protrusions (single faces with non-manifold edges) in the drift excavation mesh. There are also misaligned normals in both meshes. These issues are fixed in a few steps as outlined below.

11. Select all meshes and use **_GHeal** → *AutomaticHeal* → *IssuesToFix* and set only *FillHoles*, *FixClashingFaces*, and *AlignNormals* to *Yes*, and all others are set to *No.* Press **Enter** and keep *Parameters* = *Automatic*. Press **Enter** again to run the command. All the holes in the topo and drift meshes will be filled, and outlines of the remaining problems will be updated automatically.
12. There is a big "crack" in the topo mesh, and it cannot be readily fixed by the ***GHeal*** command. (If ***GHeal*** is run with *MendCracks* = *Yes*, only half of the crack is "stitched".) The topo mesh must be remeshed first, which will make it easier for ***GHeal*** to identify and fix the problems as the mesh becomes more uniform and regular (the same applies to the drift excavation mesh). Select the topo and drift meshes only (ensure no curves are selected), and remesh them using ***GSurf*** with *Mode* = *Tri*, *MinEdgeLength* = 0.5, *MaxEdgeLength* = 5, while all other parameters remain at defaults. A rather fine mesh is used to preserve the original mesh details.
13. Select the topo mesh only and fix the crack using **_GHeal** → *AutomaticHeal* → *IssuesToFix*, set *MendCracks* = *Yes* and all others to *No*; press **Enter** and keep *Parameters* = *Automatic*. Now the topo mesh contains no issues and has a single naked poly-edge outlining its boundary (Figure 69).

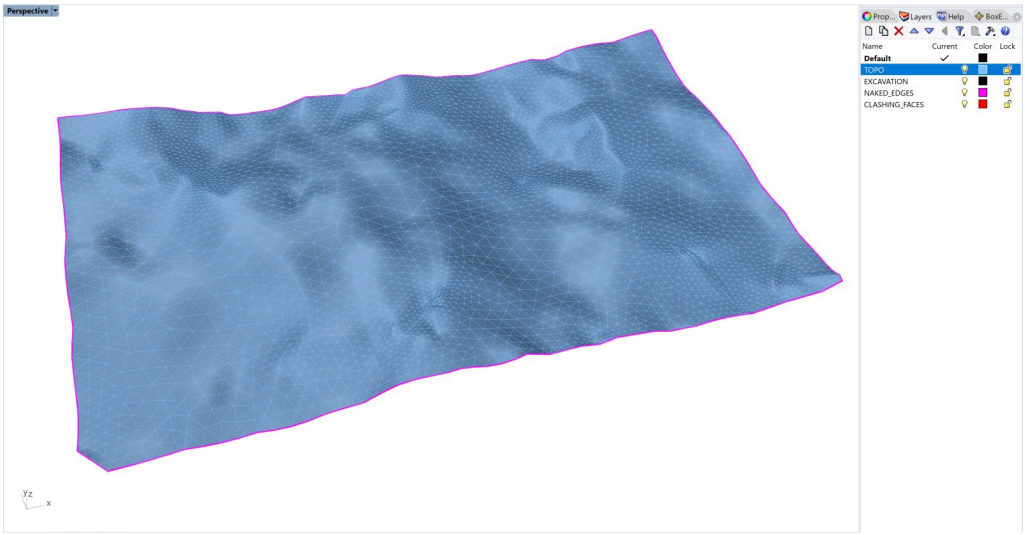


Figure 69: Remeshed and fixed topographic mesh. (The grid is not shown.)

14. Select the drift mesh only and fix the remaining issues by calling **_GHeal** → *AutomaticHeal* → *IssuesToFix*, set RemoveProtrusions = *Yes* and all others to *No*, press **Enter**, and keep *Parameters* = *Automatic*. After fixing the drift mesh, the only remaining problems are a few faces that stick out from the front part of the drift as shown by the pink outlines in Figure 70. They will be resolved later when finalizing model construction.

At this point, all outstanding problems are fixed and the meshes are ready to be used to construct the full model. Note that, in general, ***GHeal***'s ability to fix various mesh problems significantly depends on

the input mesh itself. Thus, results may vary depending on how the initial meshes are intersected and/or remeshed.

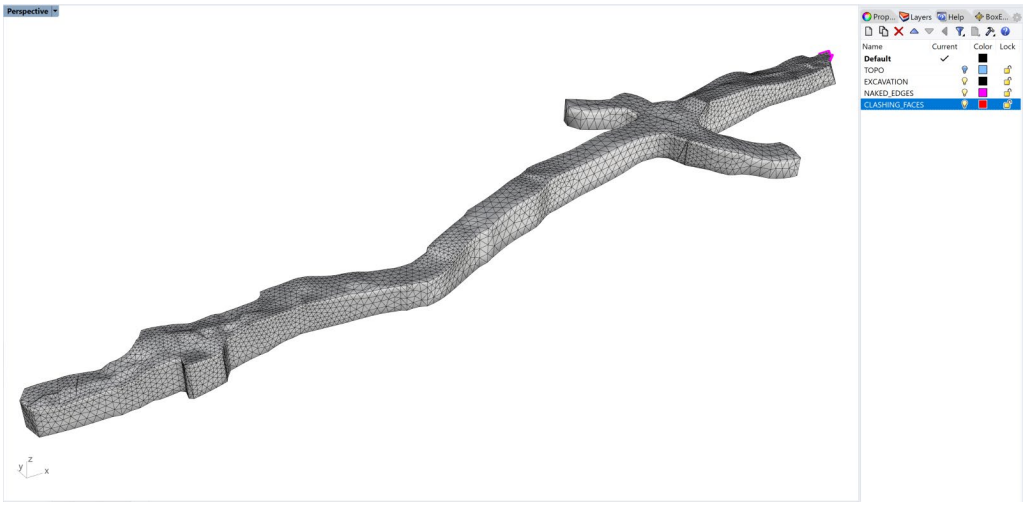


Figure 70: Remeshed and fixed drift mesh. (The grid and topographic mesh are not shown.)

Building model domain and creating volume mesh

Before final remeshing and volume meshing, a watertight modeling domain must be created. This can be done using *Griddle*'s ***GExtrude*** command.

15. Create a plane using the **_Plane** command and *Center* option. Then specify:
    - Center of plane (Deformable): 91,-55,-100
    - Other corner or length (3Point): 250
    - Width. Press Enter to use length: 200

    This will create a plane under the drift mesh as shown in Figure 71.

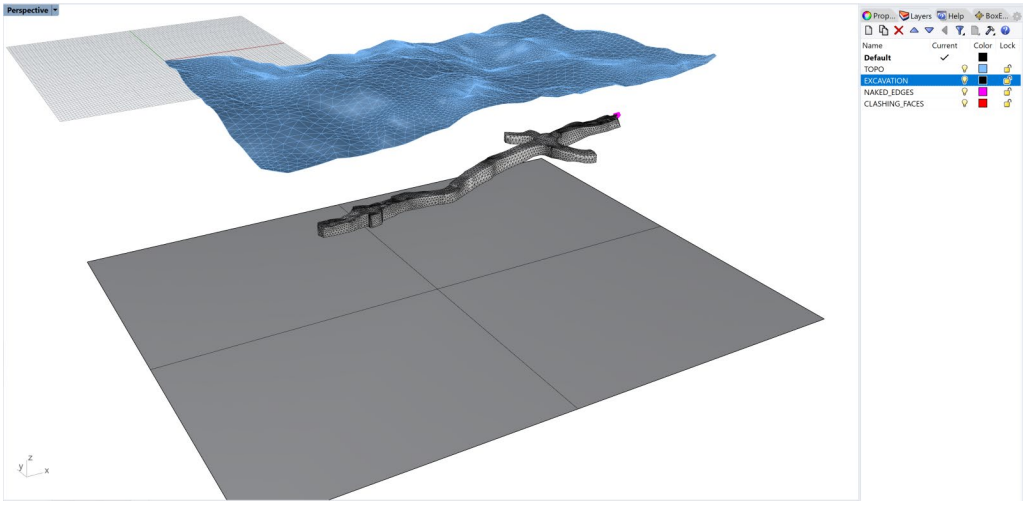


Figure 71: Creating a plane under the drift mesh.

16. Select the topo mesh and the plane and type **_GExtrude** or click on the  icon in the *Griddle* toolbar to extrude the mesh to the plane and create a watertight domain. Use the following parameters: *ExtrMeshType* = *Tri*, *MeshOutput* = *Merged*, *MeshMode* = *Unstructred*, *MinEdgeLength* = 0.5, and *MaxEdgeLength* = 10.
    The command will extrude the topographic mesh along its boundary until the intersection with the plane and will create side and bottom meshes. The result should look as shown in Figure 72.

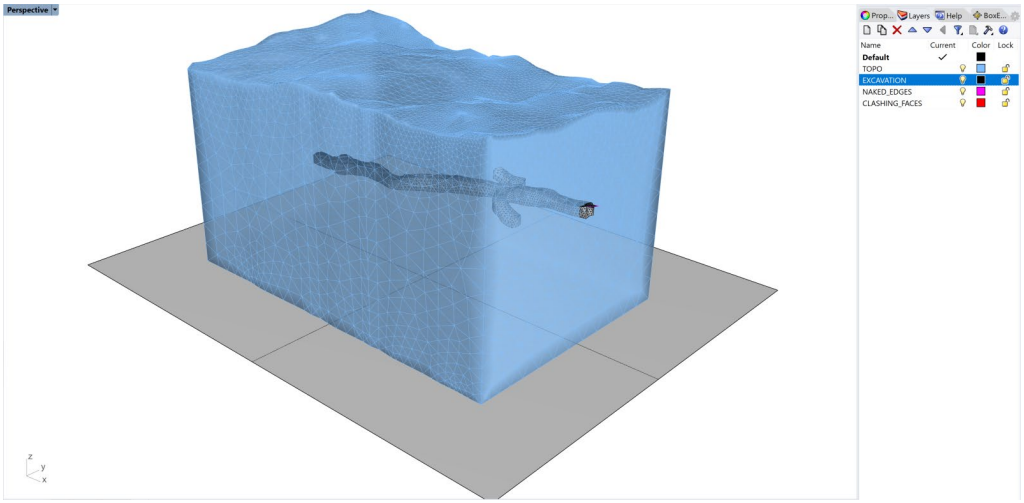


Figure 72: Creating watertight mesh domain with *GExtrude* command (Ghosted view).

17. Select and delete the plane. Turn off layers “NAKED_EDGES” and “CLASHING_FACES”.

Note that a piece of the drift mesh protrudes from the domain mesh. It can be easily separated and removed after intersection of meshes and remeshing.

18. Select all meshes (**Ctrl**+**A**) and intersect them using ***GInt*** with *MeshType* = *Tri*, *Tolerance* = 0, and *AdvancedParameters* → *OutputMesh* = *Merged*. This will create a single conformal mesh.
19. Remesh the mesh with ***GSurf*** using *Mode* = *QuadDom*, *MinEdgeLength* = 1, *MaxEdgeLength* = 10, RidgeAngle=40, and all other parameters at defaults. A large *RidgeAngle* is used to smooth the mesh. Note that the piece of the drift mesh still protrudes from the domain mesh.
20. To separate the drift mesh from the domain mesh and remove the protruding part of the drift, select the whole mesh and type the **_GExtract** command or click on the  icon in the *Griddle* toolbar. Select option *AllSurfaces* and use *MaxBreakAngle* = 180 to separate meshes only along non-manifold edges (the break angle will not have an effect as it is set to 180°).
21. Select all meshes and use *Griddle*’s command **_ColorizeObjects** or click on the  icon in the *Griddle* toolbar. This will assign a unique color to each mesh piece (Figure 73).
22. Delete all parts of the drift mesh that protrude from the domain mesh.
23. If desired, assign specific names to the drift excavation mesh and its front portion (entrance).
24. The surface meshes are ready for volume meshing. Create a *HexDominant* mesh using ***GVol*** with *MaxGradation* = 1, *TargetSize* = 7, and set the output format as desired (*FLAC3D* is used here; note, *Tet* mesh should be used for *3DEC*). If a warning about the presence of naked edges appears, press

Yes to continue with volume meshing (naked edges are present in the final model due to mesh extraction done earlier). Meshing may take some time due to the fine resolution of the topographic and drift meshes. The results of the volume mesh imported in *FLAC3D* are shown in Figure 74.

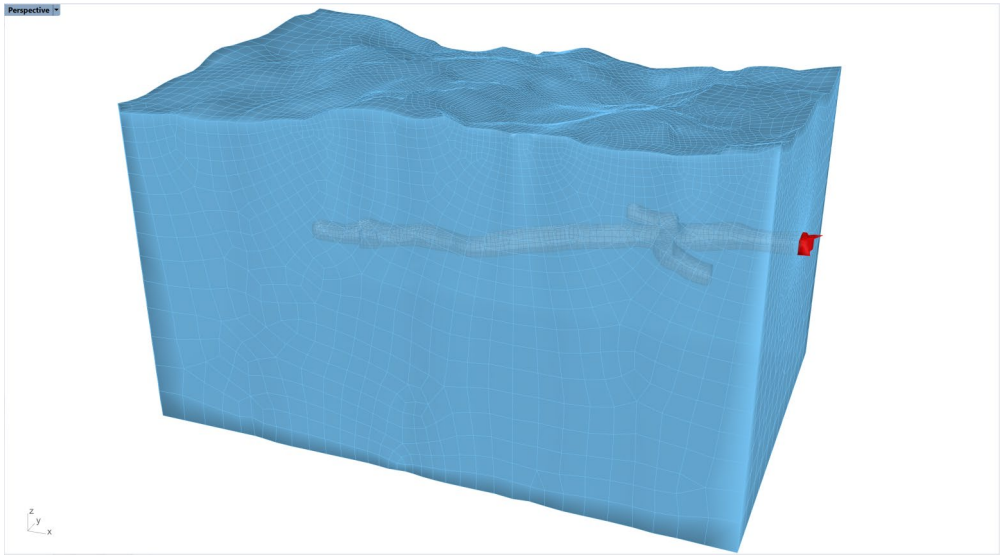


Figure 73: Meshes separated with the _GExtract command.

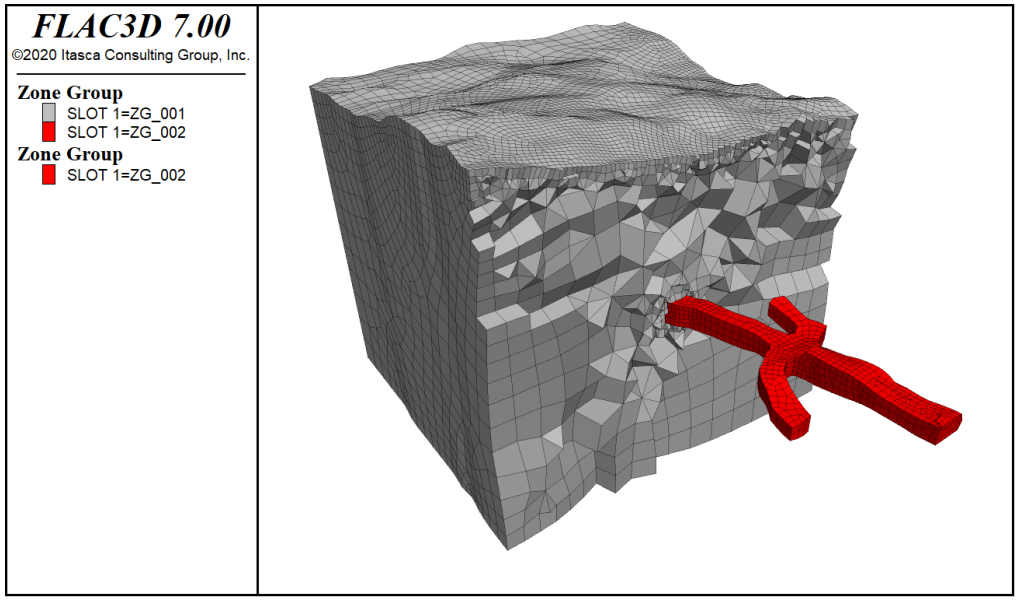


Figure 74: Final volume mesh obtained from the cleaned-up DXF meshes.

Tutorial 7: Large Open Pit Model with Multiple Intersecting Faults

Expected work time: 1–3 hours

Griddle provides powerful tools for cleaning, preparing, and meshing very large models. This tutorial describes a workflow that can be used to prepare a large open pit mine model for volume meshing. The model contains multiple intersecting faults and stratigraphic layers. Even though the model is artificial, the initial surface meshes resemble those that are often obtained from minimally processed field data. Note that due to the complex workflow, the user may get slightly different results than those presented in the tutorial; however, this should not affect the workflow as a whole. Additionally, it is assumed that advanced ***GInt*** and ***GSurf*** parameters are initially configured to the defaults; this can be achieved by restarting *Rhino* or resetting the parameters through the option ***GInt***, ***GSurf*** → *AdvancedParameters* → *Reset*.

Navigate to folder "TutorialExamples\7_LargeOpenPitMine", open file "T7_LargeOpenPitModel.3dm", and examine the model (Figure 75, Figure 76).

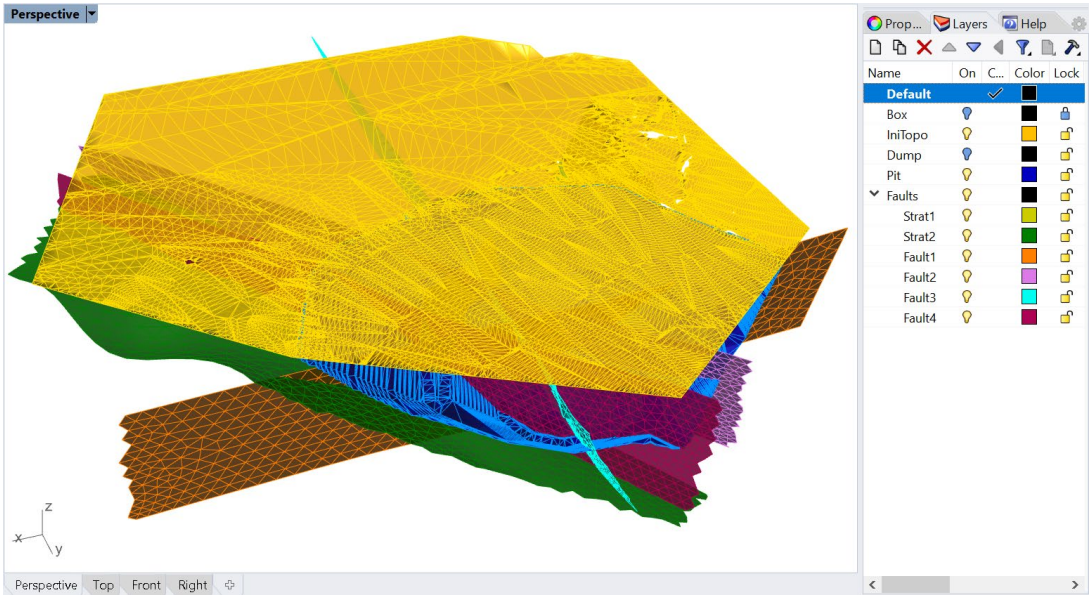


Figure 75: Initial model with topographic surface shown. Ghosted view.

Only a part of the open pit mine is provided in the model. It contains the area of interest (yellow dotted line in Figure 76) and includes the pit wall that may have the potential for instability due to several faults and a stratigraphic boundary crossing it and due to a large pit dump above the wall.

Upon examination of the initial model, following observations can be made:

- There is a single mesh in each layer. Meshes do not have specific names (see *Properties* pane / **F3**).
- Layer "Box" is locked, and it contains a polysurface representing the desired modeling domain. The layer is locked to avoid accidental modification of objects within the layer. To enable selection and modification of the objects, unlock the layer.

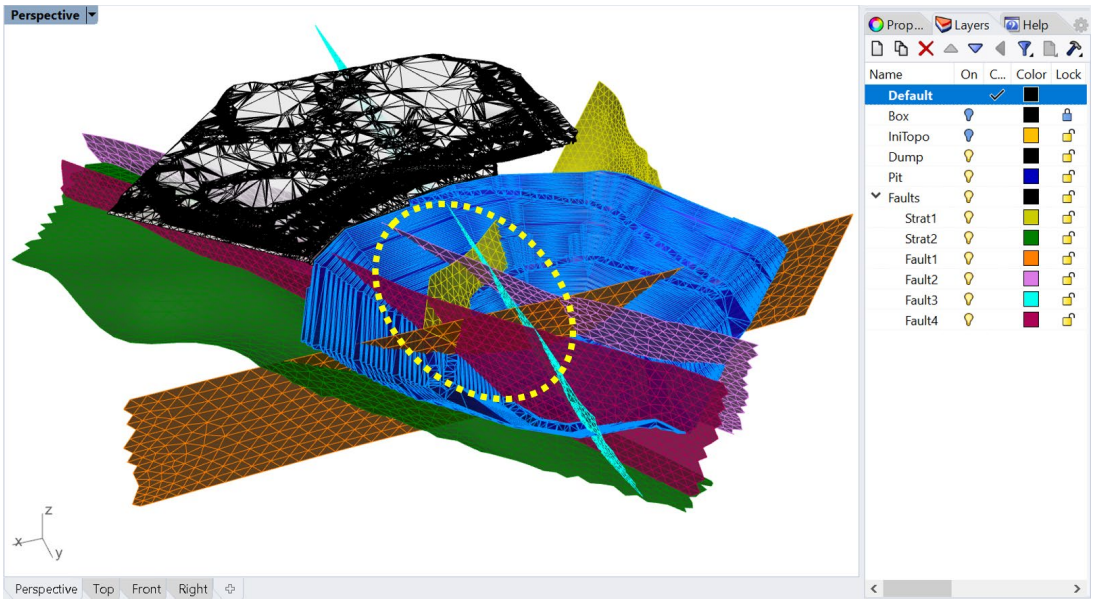


Figure 76: Initial model with large pit dump shown. Ghosted view.

- Surface meshes in layers “Pit”, “Dump”, and “IniTopo” seem to be minimally processed and contain a very large number of faces; face reduction may be needed. All other meshes appear to be remeshed and trimmed to roughly match the size of the modeling domain.
- The surface mesh corresponding to the initial topographic surface (layer “IniTopo”) visibly contains multiple holes and possibly other issues.
- None of the meshes appear to be conformal at the areas of contact or intersection.
- Some of the faults and stratigraphic boundaries seem to terminate on one another. However, it is not clear if meshes are in full contact or if there are small gaps between the meshes.
- Neither “PitDump” nor “Pit” meshes fully connect/intersect to the initial topographic surface. To verify if two meshes intersect each other, use the command **_MeshIntersect**. The command will create polylines tracing the intersecting mesh parts (do not forget to delete these polylines afterwards).
- All meshes are located away from zero coordinates but not so far as to have any effect on the accuracy of calculations. Thus, there is no need to move the model closer to the origin.

Preparation of the pit, pit dump, and topo meshes

1. Turn off all layers except for “IniTopo”. Select the topographic mesh. To make the mesh more uniform, remesh it with **_GSurf**: *Mode* = *Tri*, *MinEdgeLength* = 20, *MaxEdgeLength* = 100, *RidgeAngle* = 20°, and keep other parameters at the defaults.
2. After remeshing, some of the initial mesh problems are fixed (e.g., clashing faces). However, the mesh still contains several large holes. Fill them using **_GHeal** → *AutomaticHeal* → *IssuesToFix*: *FillHoles* = *Yes* and all other functions set to *No*. Keep *Parameters* = *Automatic*. After this, ensure that there are no more holes (turn on/off “IniTopo” layer) and then delete layers “NAKED_EDGES” and “CLASHING_FACES”.
3. Remesh the topo mesh again with the same parameters used earlier.

4. Turn off layer “IniTopo” and turn on layer “Dump”. The mesh representing the pit dump consists of a large number of non-uniform triangular faces and working with it may be difficult (in particular, remeshing). To check the number of faces in a surface mesh, use the command **_PolygonCount**; the command reports that there are 204,360 triangular polygons in the mesh.
5. There is no need for such a highly accurate representation of the pit dump, so the number of faces in the mesh can be safely reduced without impacting the model. Select the pit dump mesh and type **_ReduceMesh**. In the pop-up dialog, specify to reduce the polygon count by 85%. Keep all other parameters at the defaults.
6. Turn on layers “IniTopo” and “Pit”. Zoom in at any region where meshes visually intersect (e.g., Figure 77). One can find that in certain regions, the meshes overlap and protrude through one another or conversely, do not fully connect. This is typically due to obtaining meshes from different sources and initial mesh processing (including mesh separation, intersection, and remeshing).

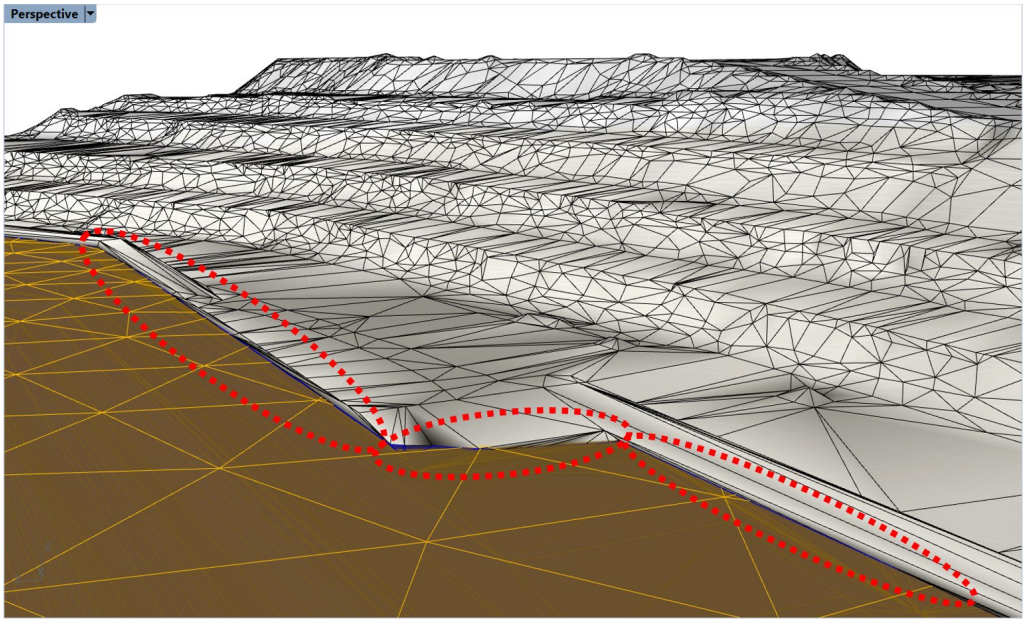


Figure 77: Zoomed view of the intersection between the pit, pit dump, and topo meshes. Marked regions denote some of the areas where intersections are not perfect.

Having several meshes (with rather different face sizes) overlap and/or terminate in close proximity to one another may present difficulties when using mesh intersector ***GInt***. In this case, it can be challenging to find such a ***GInt*** tolerance that would be large enough to close all gaps and small enough that undesirable mesh deformations do not occur. Also, these nearly overlapping faces present similar challenges when remeshing and proceeding with volume meshing.

The simplest way to approach this challenge is to spatially separate meshes, so there are only two meshes intersecting along the same line, making it much easier to select a ***GInt*** tolerance to prepare meshes for final intersection and remeshing.

In order to separate the meshes for this model, a thin boundary layer will be extracted from the pit dump mesh along a part of its boundary. After that, the pit dump mesh will be extended downward and intersected with the topo mesh.

7. Turn off layer “IniTopo” and “Pit” layers.
8. To extract a thin boundary layer, the pit dump mesh must be remeshed such that it contains small elements (faces) along the boundary. Select the pit dump mesh and remesh it with **_GSurf**: *Mode* = *Tri*, *MinEdgeLength* = 5, *MaxEdgeLength* = 5, and *RidgeAngle* = 60°. Remeshing may take a minute. Large *RidgeAngle* value is used to smooth the mesh.
9. Select the pit mesh and use the **_GExtract** → *BoundaryFaces* option. This operation will extract a single layer of faces along the boundary of the mesh. Repeat this operation one more time to extract the next boundary layer of faces such that the two outermost rings of boundary faces have been separated from the dump mesh (Figure 78).

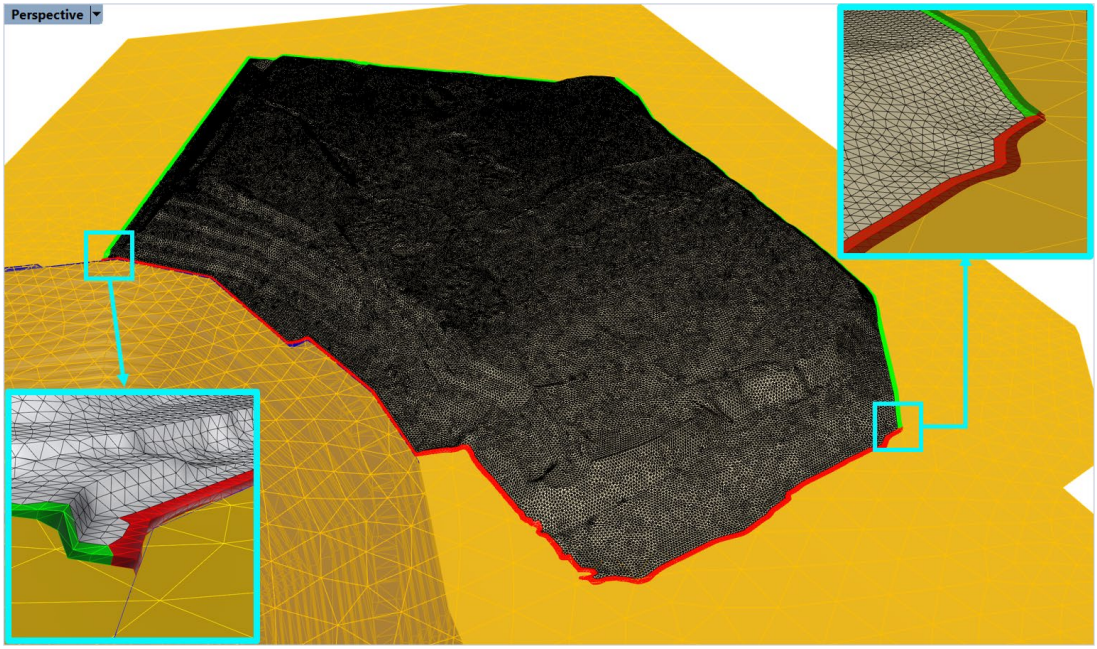


Figure 78: Boundary faces extracted from the pit dump mesh (note that colors are changed from default gray color). Faces in red were separated and will be deleted.

10. Hide the main pit dump mesh using the **_Hide** command. The only objects that should be visible are the two thin boundary layers of the dump mesh.
11. Using the **_ExtractMeshFaces** command, extract the faces from the boundary layers that are closest to the pit boundary and those that almost overlap the topo mesh (shown in Figure 78 in red). While the command is in face selection mode, other layers can be turned on and off and specific areas can be zoomed in/out to help with selection (use the **_Zoom** command while in selection mode). Note that faces can be removed from the selection when pressing **Ctrl**.
12. After extracting faces from the layers, delete them (only faces in red in Figure 78). This will create a thin gap between the pit mesh and the dump mesh and will avoid having nearly overlapping faces between the pit dump and topo meshes.
13. Unhide the rest of the dump mesh with the **_Show** command. First select the remaining thin layers of faces and add the main pit dump mesh to the selection. Type the **_Join** command to join all meshes. Now a small gap between the pit and the dump meshes can be clearly visible as in Figure 79 (turn on “Pit” layer to check).

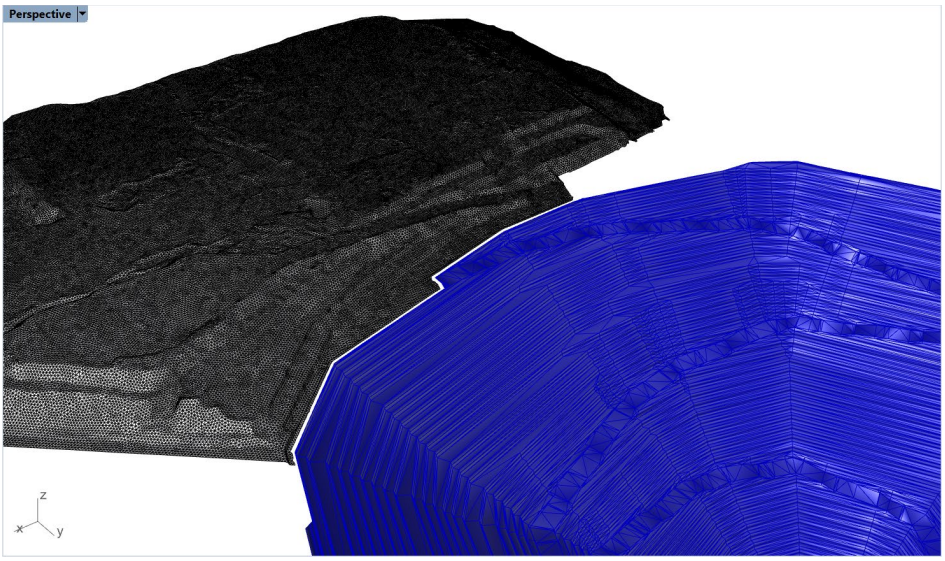


Figure 79: A gap between the pit and pit dump meshes.

14. Select the pit dump mesh and remesh it with **_GSurf**: *Mode* = *Tri*, *MinEdgeLength* = 20, *MaxEdgeLength* = 100, and *RidgeAngle* = 60°. This operation will further reduce the number of mesh faces and smooth it, making it easier to work with the mesh. Remeshing may take a minute.
15. Turn off "Pit" layer and turn on "IniTopo" layer. Double-click on the viewport icon in the top-left corner of the model view (e.g., Top); four standard viewports will be shown.
16. Select the pit dump mesh and use **_GExtend** → *FreeExtend* option. The mesh boundary will be highlighted in magenta; select the whole boundary (**Ctrl**+**A**) and press **Enter**. Now the boundary can be dragged to a desired location, and the mesh will be extended to the new boundary.
    While the cursor is in drag mode, move it to *Front* or *Right* viewports (Figure 80). Then press and hold **Shift** and drag the mouse to position it under the topo mesh as in Figure 80. When the whole boundary is below the topo mesh, release the mouse button. This will extend the dump mesh along its boundary strictly downwards (Figure 80, Figure 81).

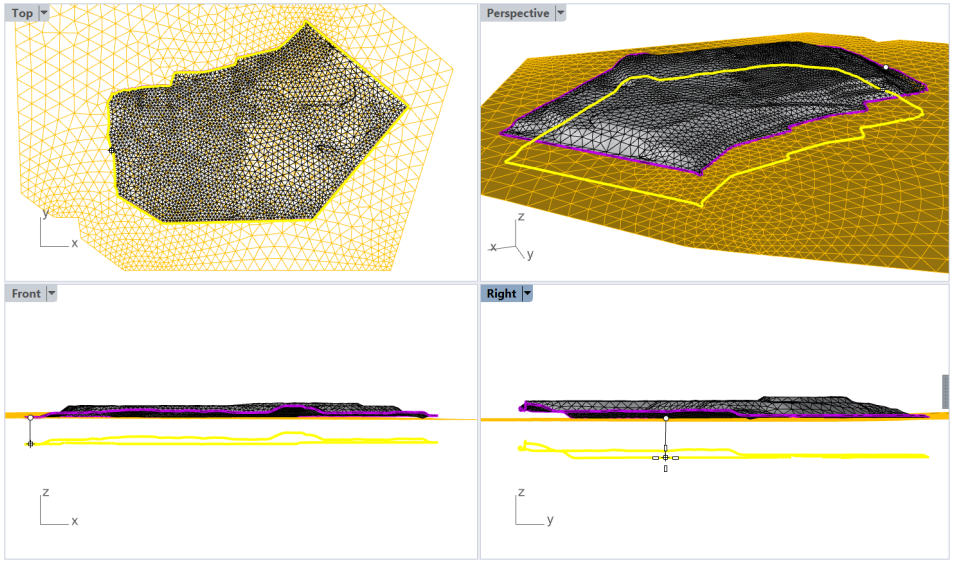


Figure 80: Extension of the pit dump mesh downward along its boundary (*FreeExtend* mode).

The same operation can be done by using **_GExtend** → *ExtendSelectedBoundary* and specifying *ExtendLength* = 150 and *Direction* = *AlongVector*. Select the whole boundary and type 0,0,0 for the beginning of the extension vector and 0,0,-1 for the end. This will extend the mesh downwards.

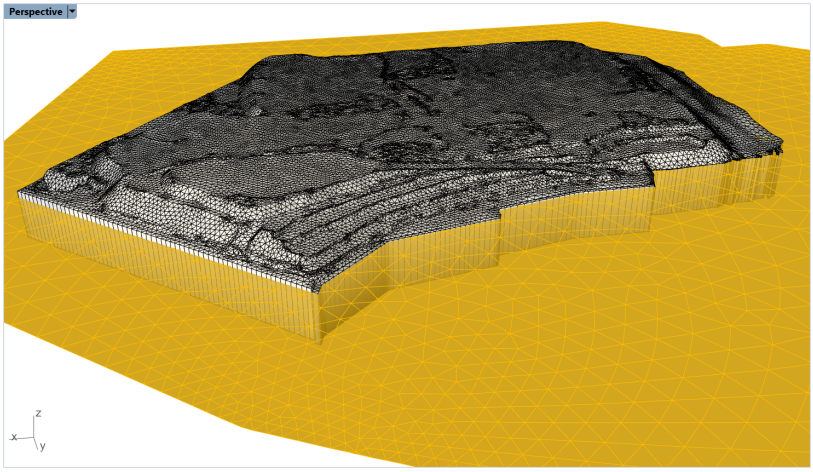


Figure 81: Extended pit dump mesh fully intersecting topo mesh.

Now the dump and topo meshes fully intersect each other. A similar process will be used to extend part of the pit mesh; however, the pit mesh must be remeshed first.

17. Turn on the "Pit" layer and turn off all others. Remesh the pit mesh with **_GSurf**: *Mode* = *Tri*, *MinEdgeLength* = 10, *MaxEdgeLength* = 40, and *RidgeAngle* = 20°.
    Restore all four viewports by double-clicking on the current viewport icon (if any viewport is maximized). Turn on layer "Box" and switch *Top* viewport to Ghosted view. Select the pit dump mesh and use the **_GExtend** → *FreeExtend* option. Select only the top part of the pit boundary in such a way that the selection goes a little beyond the box but not to the corner points of the top boundary[4], as in Figure 82. Like before, move the cursor to *Front* or *Right* viewports and while holding **Shift**, drag the mouse upward, so the extended mesh is sure to intersect the topo mesh (the exact extension length does not matter). The resulting mesh should look similar to Figure 83.
18. Navigate to the *Layers* pane and unlock layer "Box" (click on the lock icon). Turn on the layer if it is turned off. Mesh the box polysurface with the **_Mesh** command, using simple controls and the fewest number of polygons. After meshing, the polysurface is still selected. Delete it.

At this point, the pit, pit dump, topo, and box meshes are in full contact. Now they can be easily intersected and excessive parts removed. The approach described below intersects and merges meshes and then separates sub-meshes along nonmanifold edges[5].

[4] Extending a part of the top pit boundary (not whole) allows for later extraction of sub-meshes along non-manifold intersection lines. If the whole top boundary of the pit mesh is extended, the faces around the corners may connect without forming non-manifold connections.

[5] An alternative approach is to use *Rhino*'s **_MeshSplit** command. However, the command does not work properly in *Rhino* 6; it was fixed in *Rhino* 7. If using *Rhino* 7, the user can easily split one mesh by another provided they are conformal (intersect with ***GInt***, *OutputMesh* = *Separated*). This approach is simpler than the one outlined here.

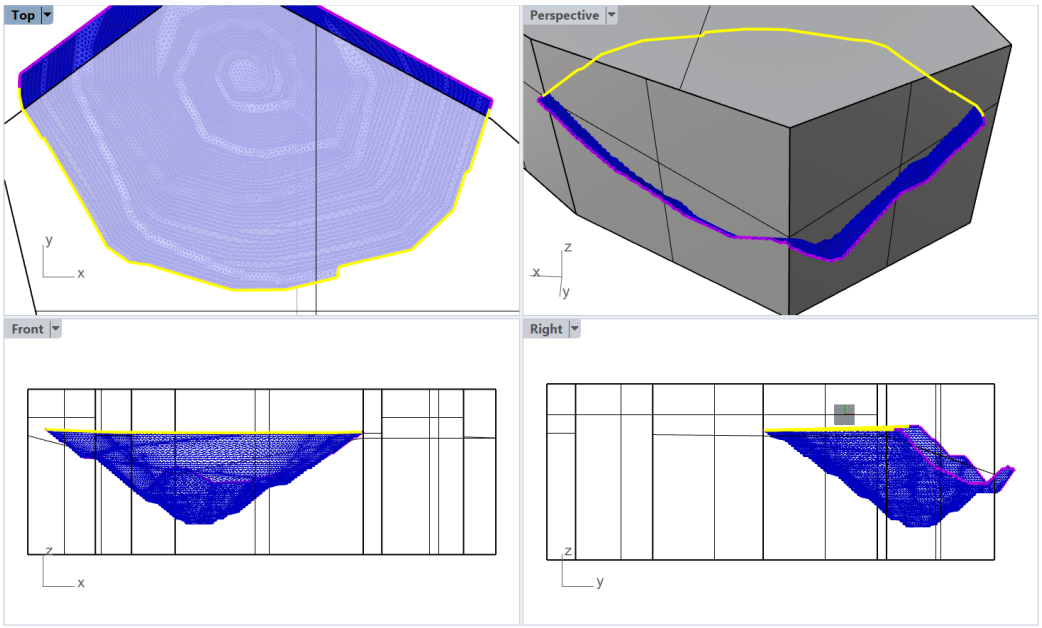


Figure 82: Selection of a piece of the pit boundary for mesh extension.

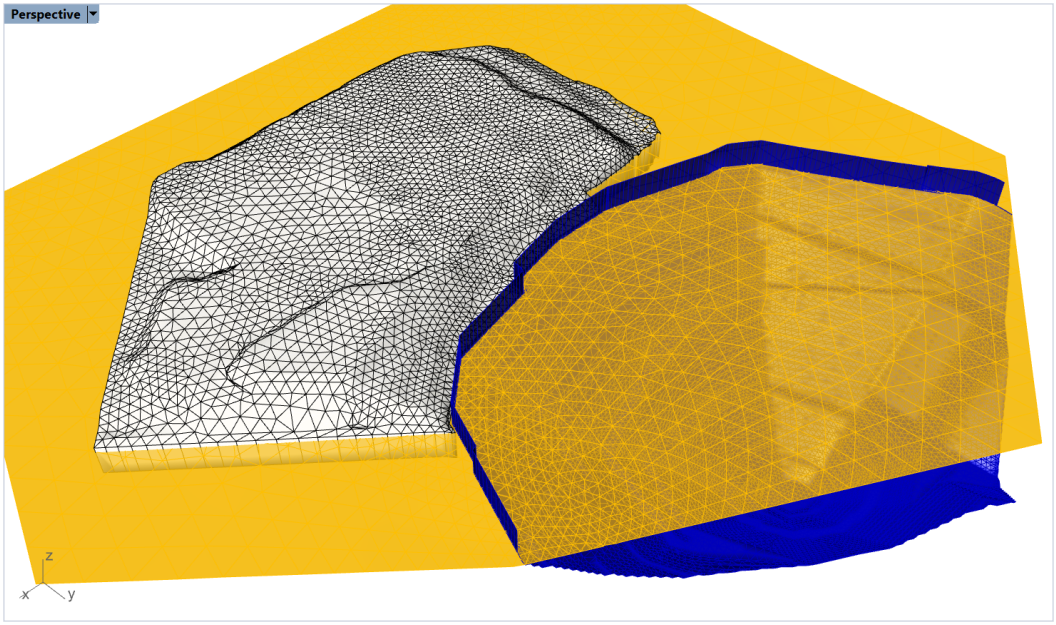


Figure 83: Extended pit mesh. Note that the mesh is not extended all the way to the corner points of its top boundary.

19. Select the pit, pit dump, topo, and box meshes and intersect them with **_GInt** → *Tolerance* = 0, *AdvancedParameters*: *OutputMesh* = *Merged*, and keep other parameters at the defaults. This will create a single conformal nonmanifold mesh within the current ("Default") layer.
20. Select the merged mesh and use **_GExtract** → *AllSurfaces*: *MaxBreakAngle* = 180 to extract meshes along the nonmanifold edges (the break angle will not have an effect as it is set to 180°).
21. Delete the top part of the extracted box mesh, excessive parts of the topo, pit, and dump meshes that extend through the topo and beyond box mesh. Merge all pieces of the box mesh and all pieces of the topo mesh (including the pieces underneath the dump and overtop of the pit) using the **_Join**

command. In the *Properties* pane, reassign layer "Box" to the box mesh, layer "IniTopo" to the topo mesh, layer "Dump" to the dump mesh, and layer "Pit" to the pit mesh.

22. Turn off all layers except for the default layer (e.g., "Default"). Check if there are any small pieces of meshes left after the extraction; press **Ctrl**+**A** to select all of them. Usually, these would be narrow single faces[6] that may not be visible when larger objects are present. Ensure that no more than a few pieces are present (there will most likely be only one piece selected). Delete these pieces. If there are many small pieces, most likely the meshes were not intersected or extracted properly; undo and repeat the last several steps according to the instructions.
23. Turn layers "Box", "IniTopo", "Dump", and "Pit" back on. Select all meshes and use **_GHeal** → *ShowErrors* to check if there are any internal naked edges. Turn off layer "CLASHING_FACES" (they will be fixed by remeshing) and type the command **_SelCrv** to select the outlines of naked edges. The command should report that only four curves are selected. If this is the case, delete the curves while they are selected. If more than four curves are present, there are holes within some of the meshes, which is likely due to incorrect mesh intersection or extraction; undo and repeat the last several steps according to the instructions.
24. All four meshes should be remeshed to make them more uniform and to simplify further work. First, assign a specific element size to each mesh in the hyperlink field of the mesh properties pane (**F3**):
    - The box and topo meshes: *elemsize:75*
    - The pit dump mesh: *elemsize:25*
    - The pit mesh: *elemsize:15*

    Then use **_GSurf**: *Mode* = *Tri*, *MinEdgeLength* = 10, *MaxEdgeLength* = 100, and *RidgeAngle* = 20° to remesh all four meshes. The resulting meshes should look like that in Figure 84.

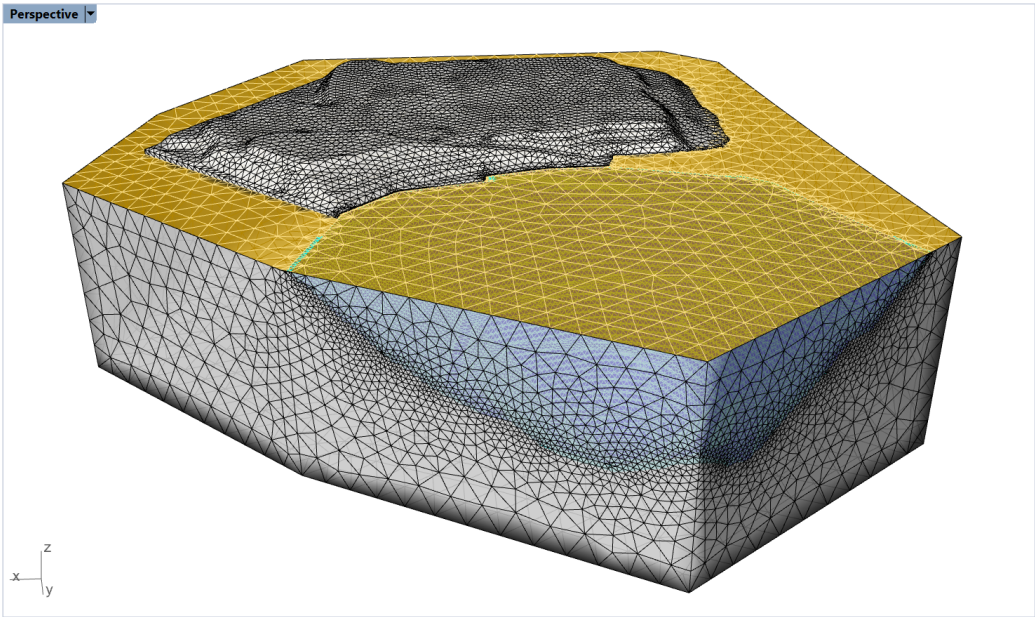


Figure 84: Intersected box, pit, pit dump, and topo meshes (Ghosted view).

[6] If there are extremely small / thin faces around mesh intersection areas (which sometimes happens after intersecting meshes), ***GExtract*** may extract such small pieces and create separate mesh objects from them.

At this point, a volume mesh could be created from the surface meshes shown Figure 84. However, the model does not yet include faults. The next section provides information about extending, intersecting, and trimming faults before including them into the model.

Preparation of the fault meshes

Turn on “Faults” layers and turn off all others. Explore the areas where meshes visually terminate on one another and/or use the **_MeshIntersect** command to see if there is full intersection between meshes. Mesh “Strat1” should terminate on “Fault1” and the rest of the fault meshes should terminate on “Strat2”. Unfortunately, none of these meshes fully intersect/terminate as they should (e.g., see Figure 85). This is a common situation when working with meshes generated from field data. As before, such meshes need to be extended, properly intersected, and then excessive pieces removed.

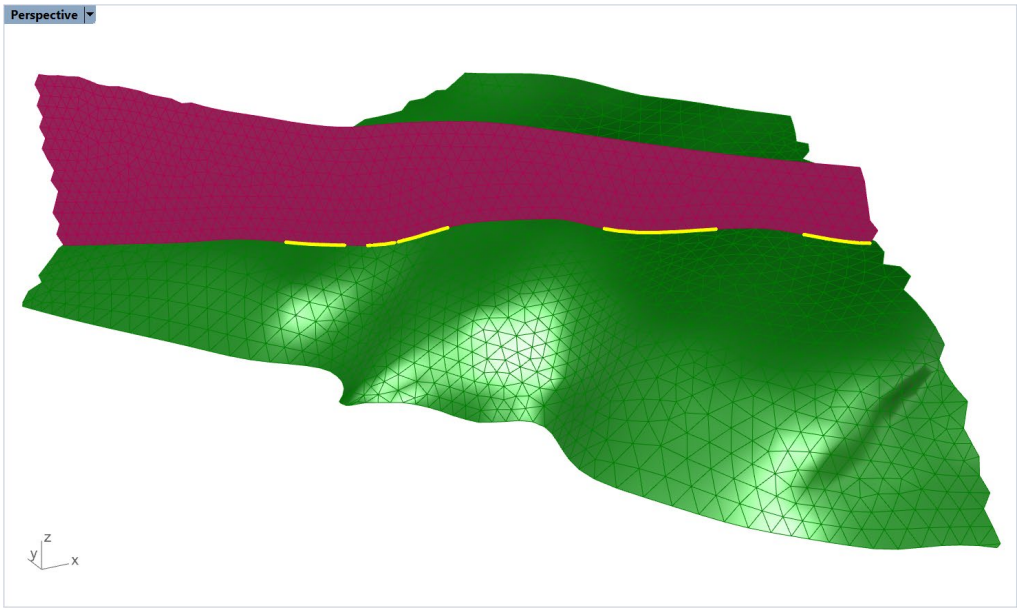


Figure 85: The _MeshIntersect command shows actual intersections (yellow lines) between “Fault4” and “Strat2” meshes. Other parts of the “Fault4” mesh do not intersect the “Strat2” mesh, leaving minor gaps.

25. Turn on only “Strat1” and “Fault1” layers and extend the “Strat1” mesh only along the edge that visually connects to the “Fault1” mesh. Use **_GExtend** → *ExtendSelectedBoundary*: *ExtendLength* = 20, *MeshType* = *Merged*. Additional segments of the boundary can be included in the selection by clicking on them and can be removed when holding **Ctrl** and clicking. Ensure that the extended portion fully intersects and is enclosed within the “Fault1” mesh (as in Figure 86).
26. Repeat this operation to extend meshes “Fault2”, “Fault3”, and “Fault4” so they extend through the “Strat2” mesh. Use the same **_GExtend** parameters as in the previous step. Ensure that the extended parts fully intersect and are enclosed within the “Strat2” mesh (zoom in at intersection corners).
27. Turn on layers “Box”, “Strat1”, and “Fault1” and turn off all other layers. Intersect all three meshes with **_GInt** → *AdvancedParameters*: *OutputMesh* = *Merged*, *Tolerance* = 0, and keep other parameters at the defaults. This will create a single conformal nonmanifold mesh within the current (“Default”) layer.

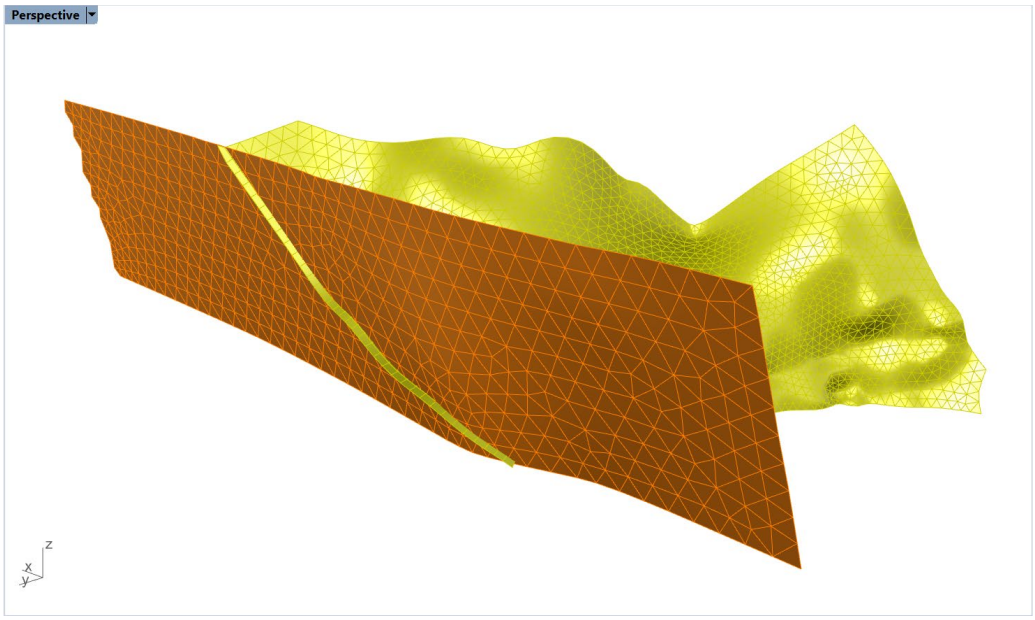


Figure 86: Extension of "Strat1" mesh along the part of the boundary closest to "Fault1".

28. Use **_GExtract** → *AllSurfaces*: *MaxBreakAngle* = 180 to extract meshes along nonmanifold edges (break angle will not have an effect as it is set to 180°).
29. Delete excessive parts of meshes located outside the box. Hide the box with the **_Hide** command and delete the excessive part of the "Strat1" mesh extending through the "Fault1" mesh. In the *Properties* pane, reassign layers "Strat1" and "Fault1" to the corresponding meshes.
30. Show the box using the **_Show** command. The box is still in the "Default" layer, no need to reassign it yet. Turn off layers "Strat1" and "Fault1" and turn on layers "Strat2" and "Fault2". Repeat the operations from the previous three steps to intersect, separate, and remove excessive mesh parts. Note that the box will be split in two pieces during these operations. There is no need to join them and reassign layers "Box" and "Strat2" until all meshes are intersected and excessive parts removed.
31. Repeat the previous four steps for "Fault3" and "Fault4" meshes. Remember to include the box when intersecting (in case it was hidden earlier) and to delete thin mesh pieces underneath the "Strat2" mesh.
32. Join two pieces of the box mesh and place them into the "Box" layer (**_Join**). Join (up to five) pieces of the "Strat2" mesh and place them in the "Strat2" layer. Turn on all "Faults" and "Box" layers only.

Now there should only be seven meshes, as in Figure 87 (check with **Ctrl**+**A**). All these meshes are perfectly intersected between one another and are ready to be intersected with the pit, pit dump, and topo meshes, followed by final remeshing and volume meshing.

Final mesh intersection and remeshing

33. Turn on all the layers. Select all meshes and intersect them using **_GInt** → *Tolerance* = 0.001, *AdvancedParameters*: *OutputMesh* = *Separated*.
34. Remesh all meshes using **_GSurf**: *Mode* = *Tri*, *MinEdgeLength* = 10, *MaxEdgeLength* = 75, *RidgeAngle* = 10° and *AdvancedParameters*: *MaxGradation* = 0.2, *Optimization* = 10, *ShapeQuality* = 0.9, and *OutputMesh* = *Separated*. The results are shown in Figure 88.

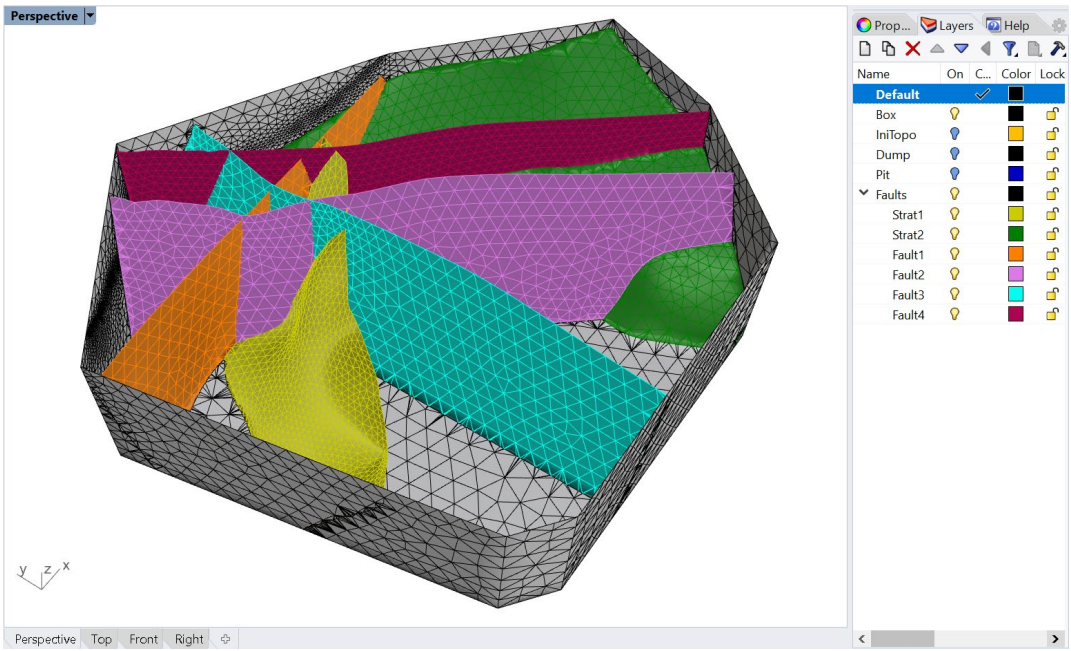


Figure 87: Intersected and trimmed "Faults" meshes.

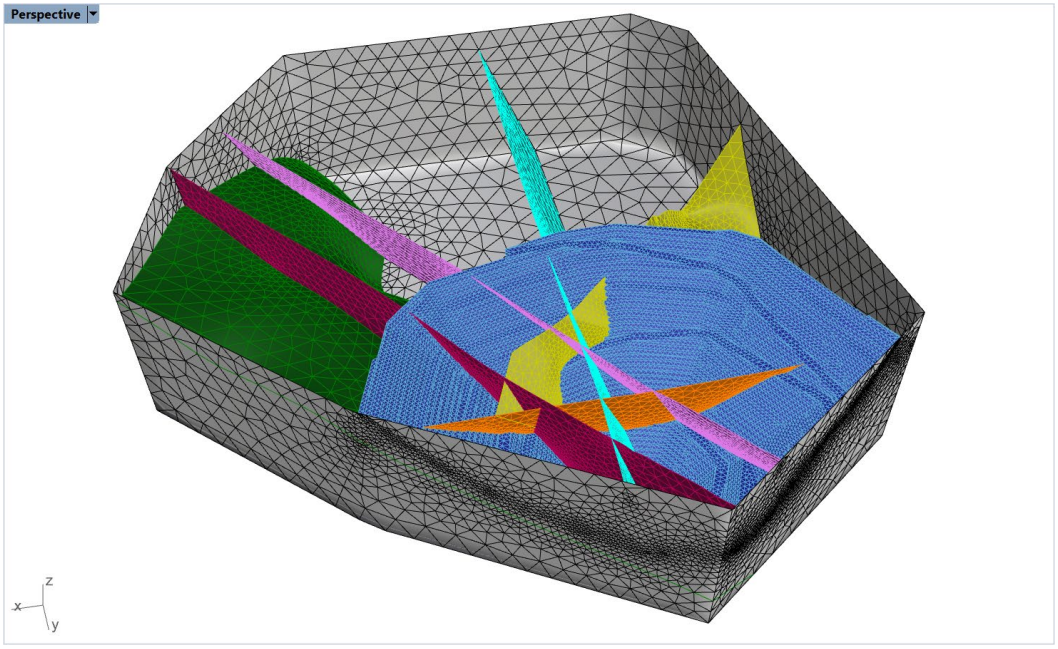


Figure 88: Final triangular surface meshes (the pit dump and topo meshes are not shown).

35. Verify that the final surface meshes do not contain any holes or clashing faces by calling **_GHeal** → *ShowErrors* (select all meshes first). After that, type **_SelCrv**. *Rhino* should report that only 10 curves are selected.
    If more than 10 curves are selected, that indicates the presence of nonconformal faces, internal holes, and/or clashing faces. To locate issues, hide all layers except for "NAKED_EDGES" and "CLASHING_FACES" and start deleting large curves corresponding to mesh boundaries (there should be 10 curves). Any curves left indicate potential issues.

If any issues are present, investigate each problematic area. It may happen that some nearly overlapping faces from different meshes got merged into a single face during remeshing. In such a case, one of the meshes may contain a hole. An example of this is shown in Figure 89, which was obtained when conducting step 24 of this tutorial with *RidgeAngle* = 10°. In this case, nearly overlapping faces from the pit mesh and the topo mesh got merged after remeshing, and the resulting single face got assigned to the pit mesh only. This is not a critical issue as meshes still form a watertight domain and the volume mesh can be generated. The only negative consequence is that the face group designating mesh surface within the volume mesh may contain a hole or the wrong group would be assigned to the "overlapping" face[7]. Such situations are more often encountered when working with models containing a large number of faults intersecting at very small angles (so faces from one mesh are very close to the faces from another mesh near intersections; depending on which remeshing parameters are used, some faces may become merged between meshes).

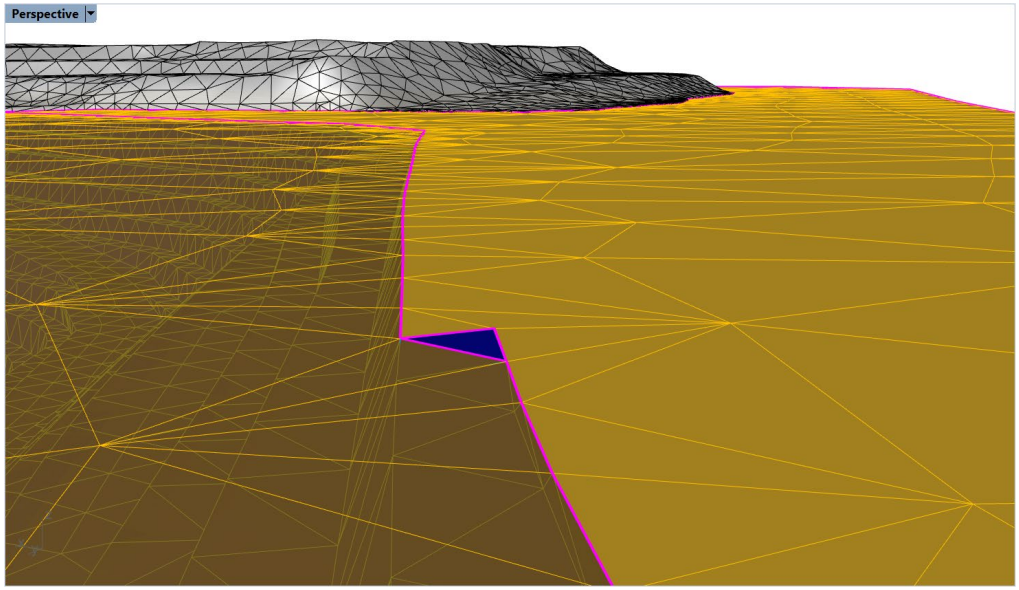


Figure 89: Overlapping faces (in blue) created a hole in the topo mesh.

36. Assign names in the meshes *Properties* pane as desired. These names will be transferred as named face groups (surfaces) to the volume mesh (currently, only for *FLAC3D* and *3DEC*). See details about group naming and surface naming in the *Griddle 2.0 User Manual*.
37. Create a tetrahedral volume mesh in *FLAC3D* binary format. Keep all other ***GVol*** parameters at default values. Tetrahedral volume mesh generation is usually a fast process[8], and it is used here. In case *Griddle* issues a message about the presence of naked edges, read the message and press **Yes** to continue. The final mesh is shown in Figure 90.

[7] Fixing the hole created by overlapping faces will not resolve the situation, as no volume elements are created between the overlapping faces (there is zero volume between them).

[8] In general, it is much easier to generate pure tetrahedral meshes for very complex models, as filling any random volume with tetrahedrons is a simpler process than filling a volume with a combination of various shapes. Besides, the average quality of tetrahedral elements in such cases is typically higher than for hex-dominant meshes.

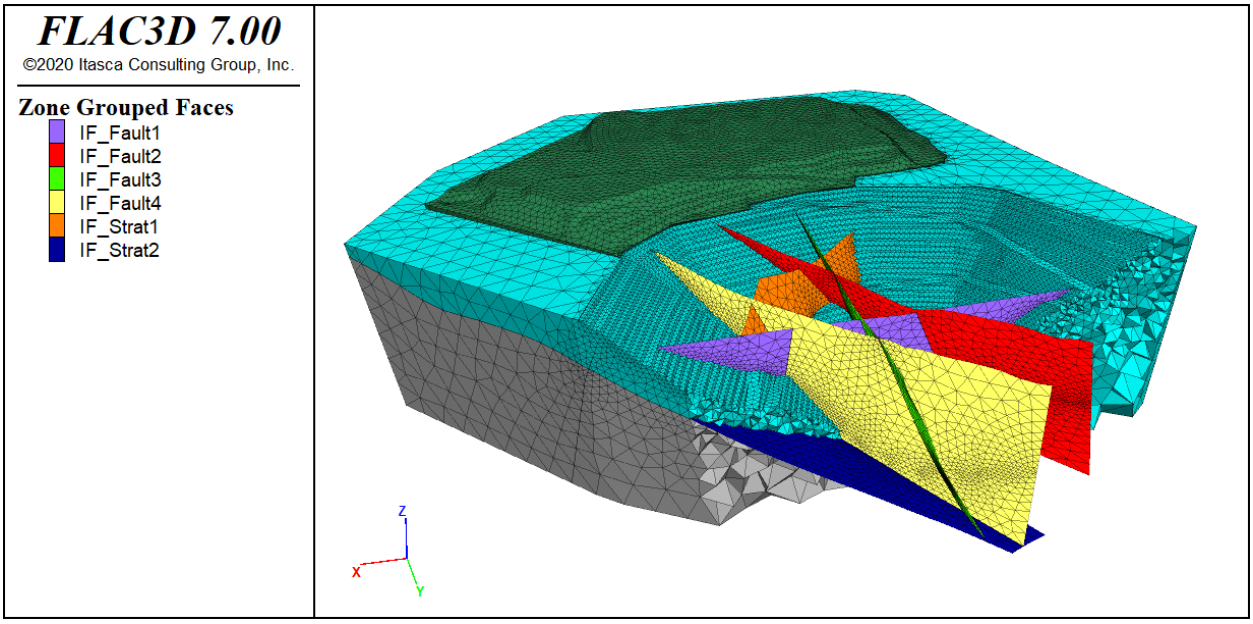


Figure 90: Tetrahedral volume mesh of open pit mine model.

As an optional exercise, remesh all meshes again using the same ***GSurf*** parameters as in step 34 but with *QuadDom* mesh type. Then generate a hex-dominant (HexDom) volume mesh in *FLAC3D* binary format (which may take several minutes for such a large model). Save the file under a different name than the previous file. Compare log outputs from ***GVol*** for both cases to see what kind and how many elements are generated.

When generating the hex-dominant volume mesh for the model, ***GVol*** will produce about half of the total number of elements compared to a tetrahedral mesh. However, a HexDom mesh contains a large number of undesirable pyramidal elements[9]. For such complex models, engineers often generate pure tetrahedral meshes and then split each tetrahedron into several hexahedrons within *FLAC3D* to obtain a pure hexahedral grid (pure hexahedral grids yield significantly more accurate numerical results in *FLAC3D*).

Note that this model is not suitable for outputting in *3DEC* deformable blocks format as faults do not connect to the topo mesh, and, thus, they do not cut blocks within the pit or the rest of the model. *3DEC* rigid block format could be used instead, but the surface meshes will need to be significantly coarsened first to produce a coarser volume mesh that can be loaded into *3DEC* (as *3DEC* will try to detect and create a contact between each rigid tetrahedral element/block, resulting in an enormous amount of block contacts).

[9] *FLAC3D* grid densification becomes rather challenging if pyramids are present.