

Pre-processing and meshing FE-models with *Griddle/Rhino*: Applications on complex geometries from salt mine drifts

G. Maniatis & S. Fahland

**Department B3.5 Geotechnical Safety Analyses
Federal Institute for Geosciences and Natural Resources (BGR)
Hannover, Germany**

BGR, Department B3.5 Geotechnical Safety Analyses

Several Projects on Final Nuclear Waste Disposal in salt rock, claystone, crystalline rock and the relevant geotechnical barriers

FEM applications for:

- Modeling complex geological structures
- Complex underground mine geometries
- Large 3D Models
- Coupled Thermal, Hydraulic and Mechanical processes
- Long modeling times

Solving FEM with several codes:

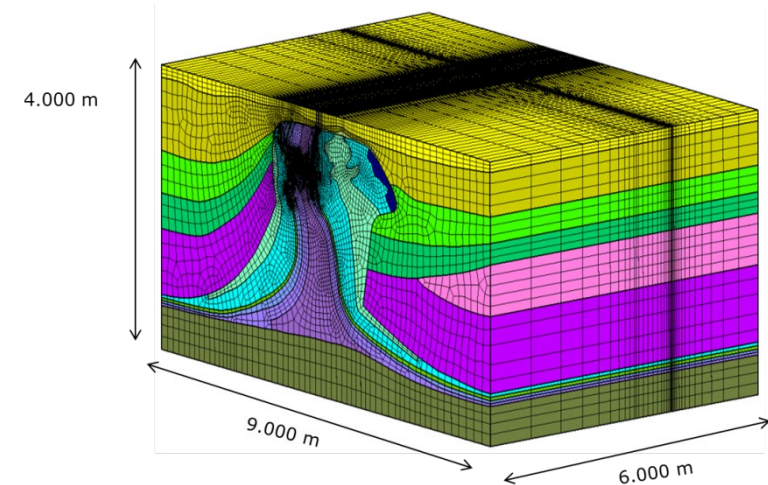
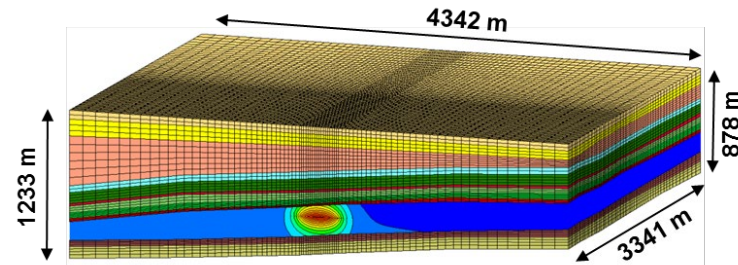
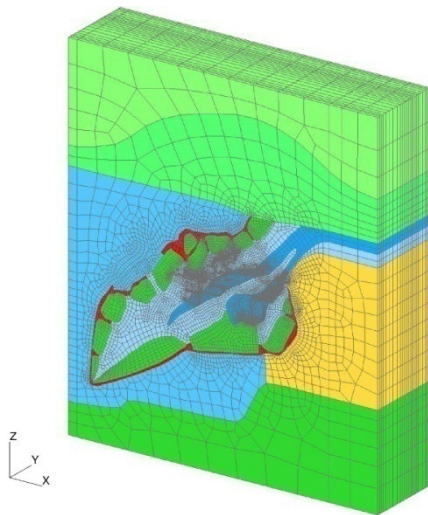
- JIFE (BGR in-house developed)
- Flac3D
- OGS
- Roxol

Preprocessing FEM Models

Depending on the size of the model, preprocessing can be time consuming and often requires compromises on the amount of geometrical detail that the final numerical model will contain.

In most of the cases of R&D so far, the models are designed by hand in CAD software as

- complex 2D geometries extruded to 3D or as
- complex 3D geometries



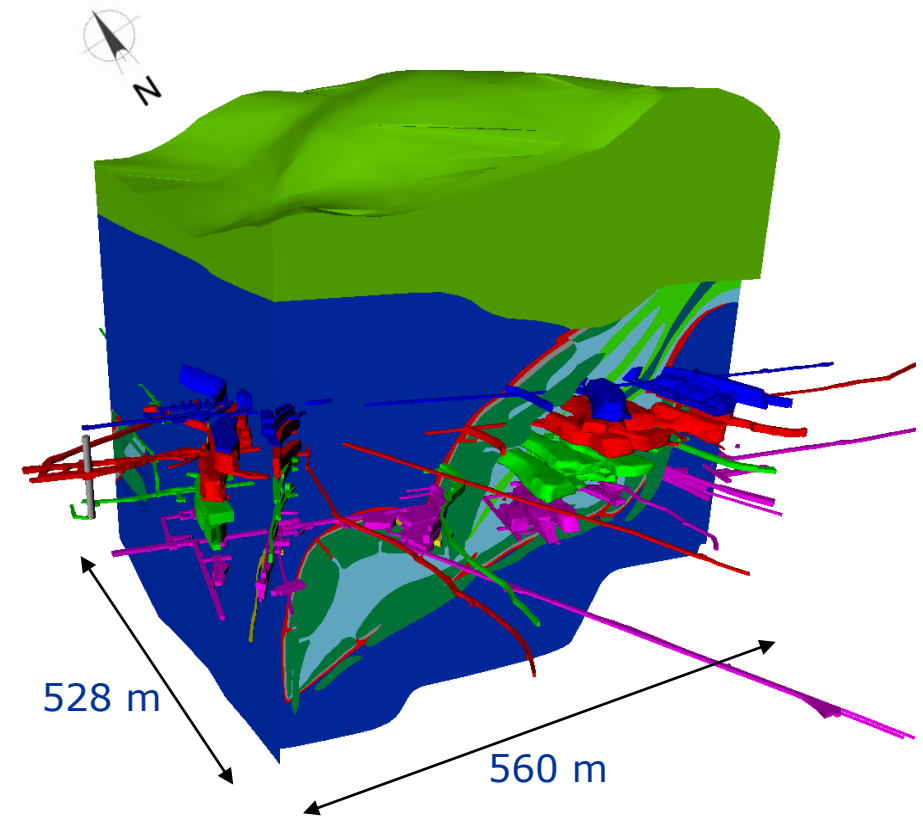
However, most of these models consist of solids defined by flat or relatively simple and smooth surfaces.

Preprocessing FEM Models

Prospective, more elaborate safety analysis of potential nuclear waste disposal sites requires the implementation of more detailed geometries in FE Models such as:

- shafts
- drifts
- mining rooms
- sealing elements
- surrounding geological structures

These requirements increase the 3D complexity of the mesh dramatically. In such cases, more powerful preprocessing and meshing tools are required.



Preprocessing and meshing FE models with *Griddle/Rhino*

Preferably one software package for both geometry and mesh generation

- Easy to use
- Capability to import real geometries and polygon meshes derived from 3D scans and surveying campaigns in the salt mine (different file types)
- Capability to analyze, repair and idealize imported geometries and/or polygon meshes
- Easily produce and refine a good quality volume mesh that can be exported to different FEM software for further preprocessing (e.g. BCs and material properties) and solving

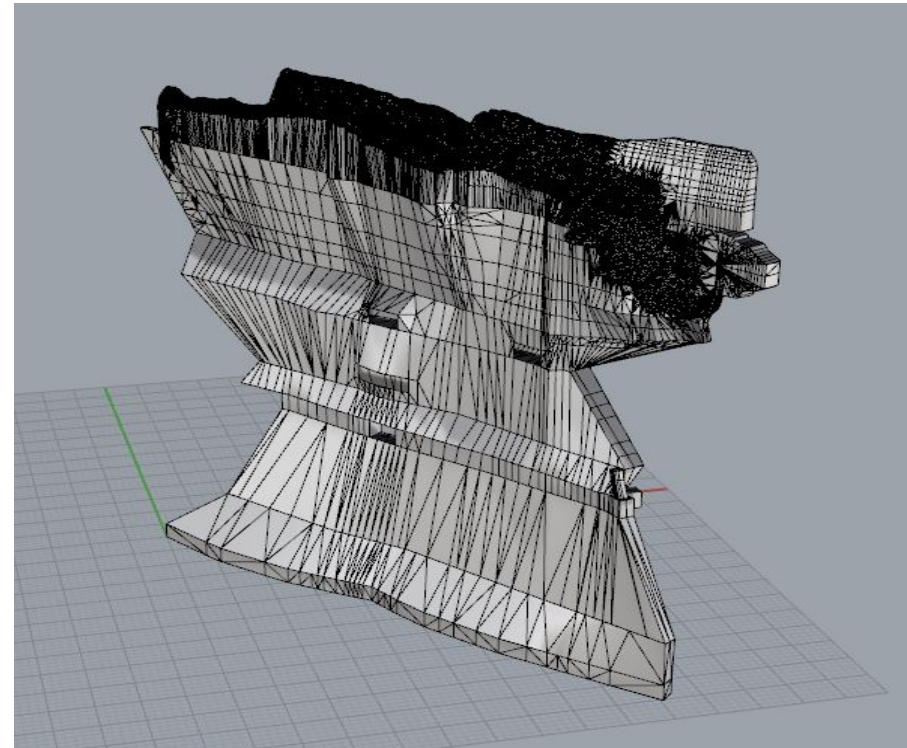
Imported Data

The drift geometries from a German salt mine that were used for this application were available in the common .dwg file format.

Prior to importing the .dwg file into *Rhino* the appropriate length units have to be set for the model and the layout in the properties menu.

The initial polygon meshes were not built for numerical modeling purposes derived from 3D scans and surveying campaigns in the salt mine.

The consist of several small parts not joined with each other and contain several types of errors and defects, which have to be found, analyzed and repaired with the check-mesh and mesh-repair tools in *Rhino*.



Typical mesh errors and defects

- Pairs of surface element faces that intersect each other, duplicate surface elements faces and degenerated faces (zero area)

Commands

Testmsx (*Rhino 5.0*)

CullDegenerateFaces

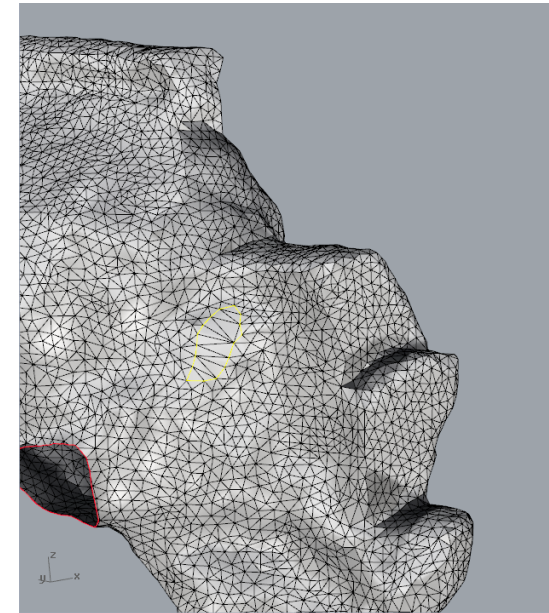
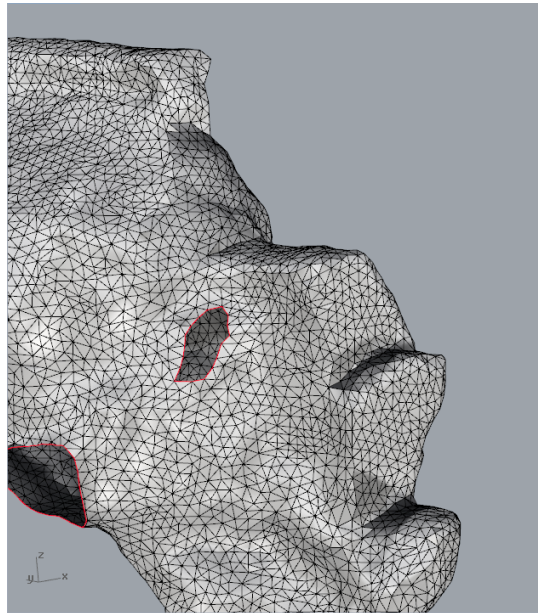
ExtractDuplicateMeshFaces

- Openings (i.e. missing surface elements) in the mesh and “naked edges” at the edges of open polygon meshes

- Commands

FillMeshHole

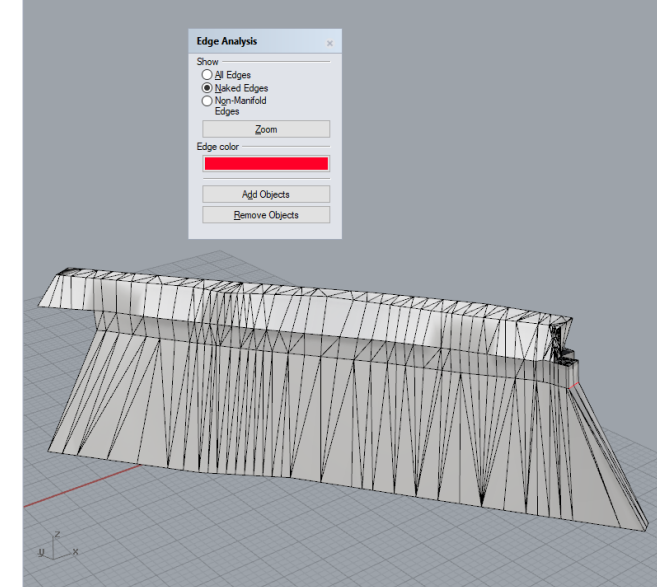
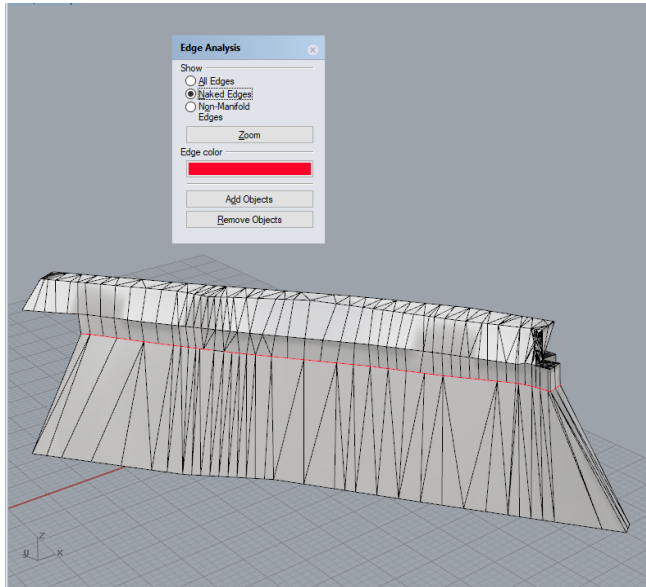
PatchSingleFace



Typical mesh errors and defects

Non-conformal mesh locally or along edges

- Local non-conformal mesh can be repaired by collapsing local vertices, faces or edges (**CollapseMeshVertex**, **CollapseMeshFace**, **CollapseMeshEdge**) or by locally deleting the mesh faces and reconstructing the mesh by filling the mesh hole or patching single mesh faces.
- Larger non-conformal edges (naked edges) can be easily corrected with the **MatchMeshEdge** command



Typical mesh errors and defects

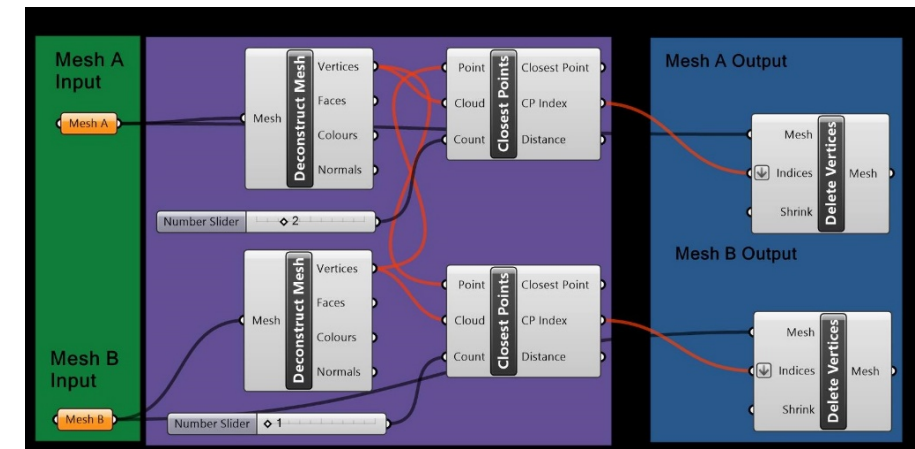
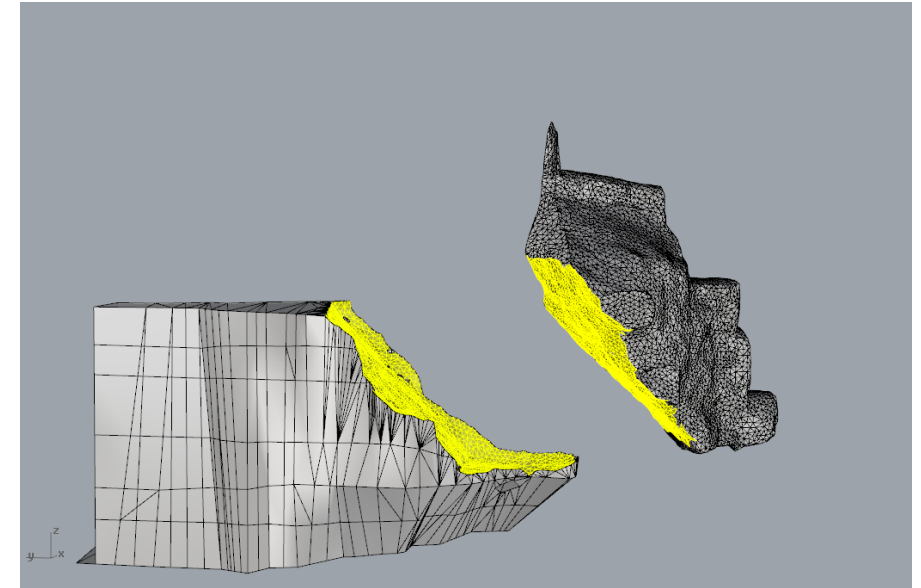
Duplicate mesh areas and overlapping surface mesh

Areas consisting of duplicate (i.e. identical) surface meshes are another usual defect in the initial mesh geometries that can be solved mostly by using mesh Boolean operations in *Rhino*. (**MeshBooleanDifference**)

However, there are extensive, complex and tightly overlapping mesh surfaces that share no identical nodes and therefore cannot be removed by using standard Boolean operations.

Use of an algorithm within the Grasshopper plugin, which is available for *Rhino*.

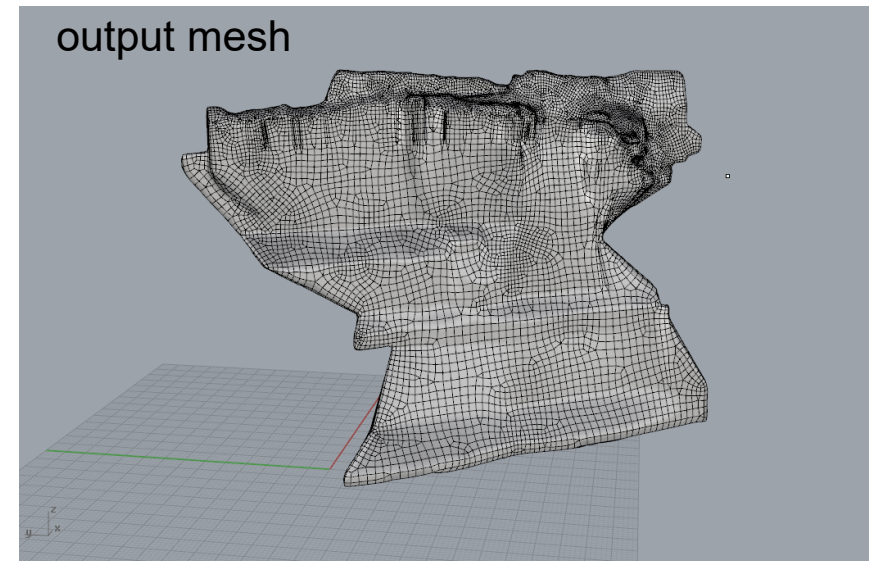
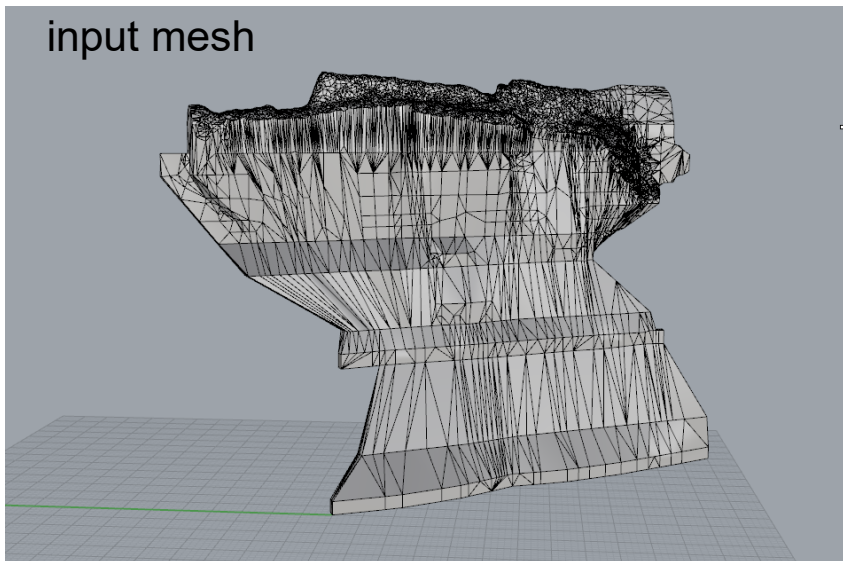
This algorithm searches for the closest neighboring vertices of two partially overlapping meshes and removes or separates them from the rest of the model.



Surface mesh with *Griddle* GSurf

Polygon meshes are joined to create closed ones, while avoiding or correcting non-conformal meshes along the matched mesh edges and finally unifying all mesh normals on the element surfaces.

Remeshing the previously joined surface meshes with the *Griddle* **GSurf** command



Options:

Element Type:

Tri, QuadDom, AllQuad

Element edge length:

Max/MinEdgeLength

Level of mesh detail:

Ridge Angle

Variation of element size:

MaxGradation

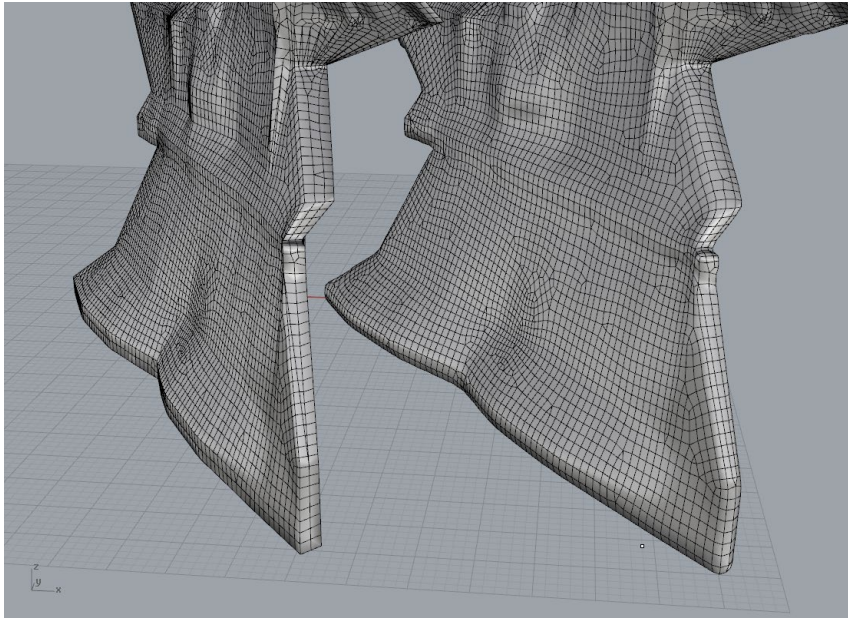
Surface mesh of entire area of interest
(19734 polygons, model size 100m x 71m x 15m)

The generated surface mesh can be checked again for any errors with the *Rhino* **MeshRepair** command.

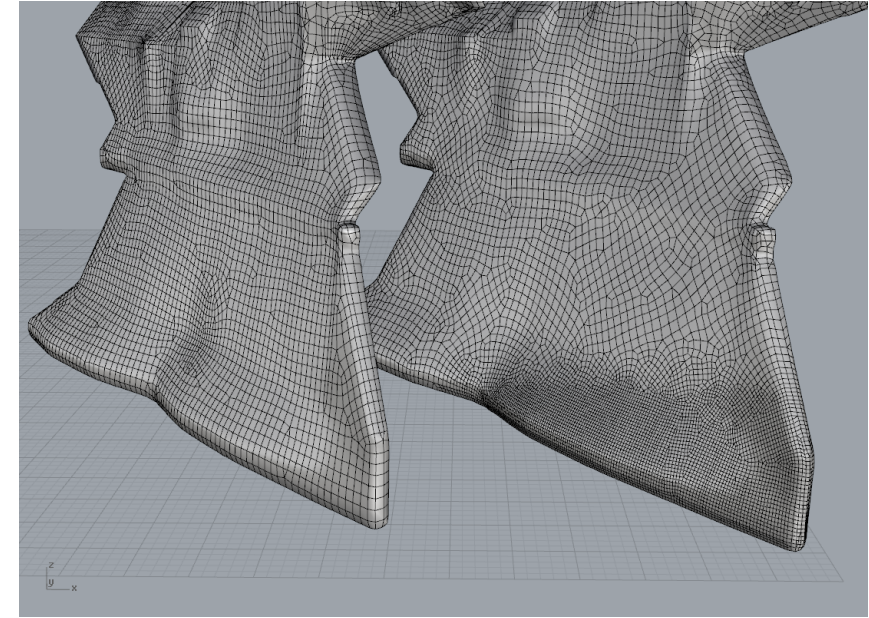
Surface mesh with *Griddle* GSurf



Mesh refinement



Smoothing out sharp edges globally or locally on the surface generated with **Gsurf** with *Rhino's* (**Smooth** command).



Assigning local mesh densities on surfaces, around points or along lines.

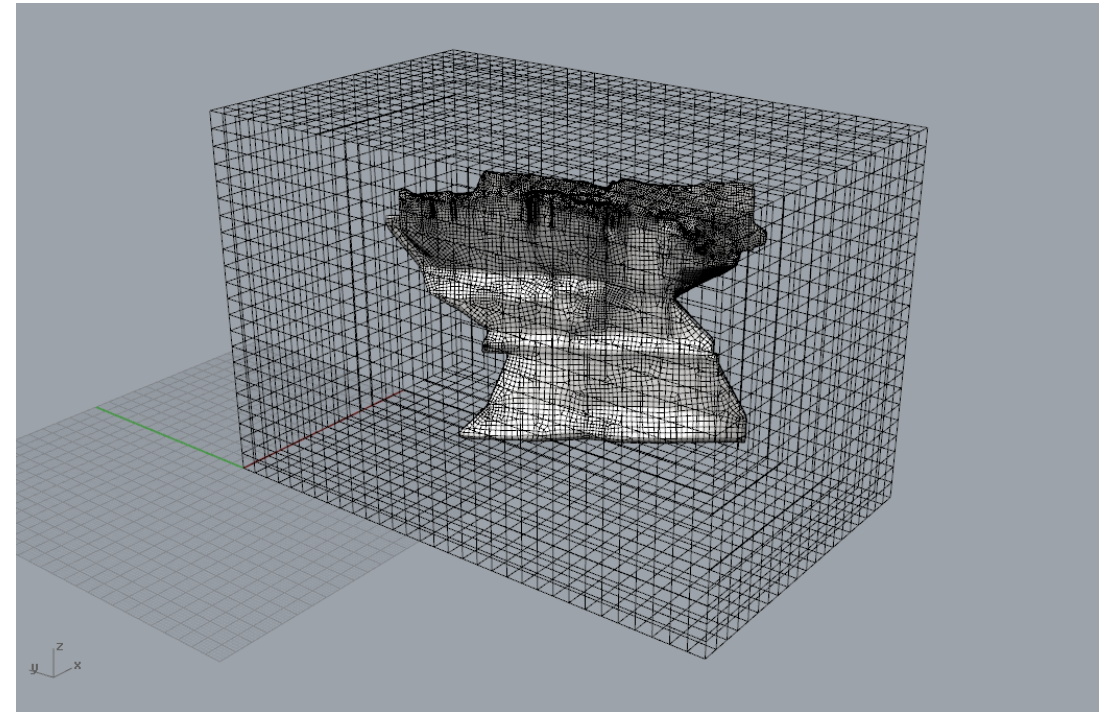
Surface mesh with *Griddle* GSurf

All the surface mesh parts of the model must form structures with watertight boundaries to be used for the generation of the final volume mesh with *Griddle*'s **GVol** command. The output may consist of either a Tet (all-tetrahedral) volume mesh or a conformal hex-dominant volume mesh.

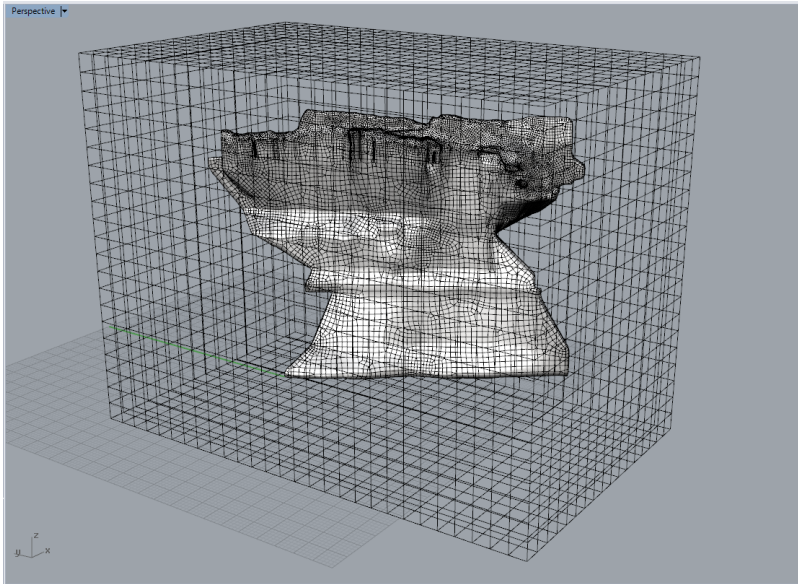
Possible output file formats: FLAC3D, 3DEC, ABAQUS, ANSYS, NASTRAN

The model boundaries can be constructed as a surface geometry with the command **BoundingBox** around the modeled mine room and scaled to the appropriate size (**ScaleNu** command).

The boundary surface of the model can be meshed with *Griddle*'s **GSurf** command in the same way as the mining room.



Volume mesh with *Griddle* GVol



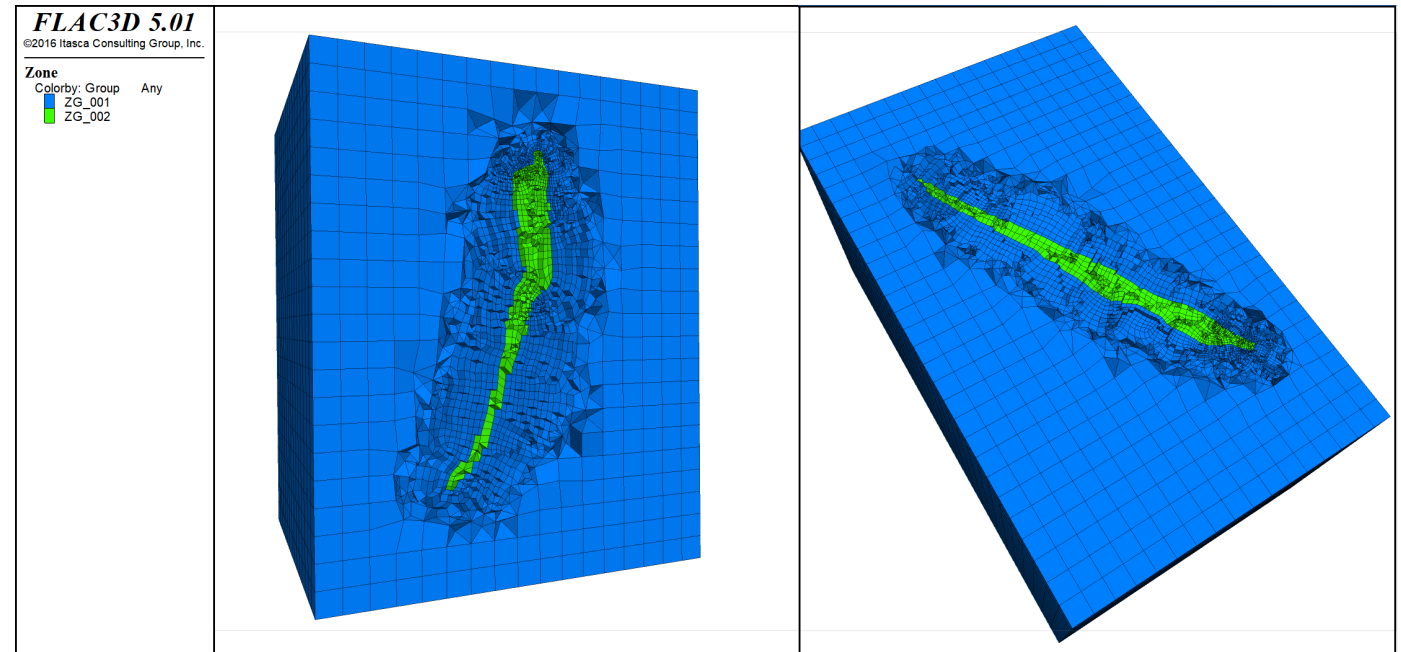
input mesh

- number of surface elements: 22576
- number of nodes: 21633

GVol Mode: ConHexDom

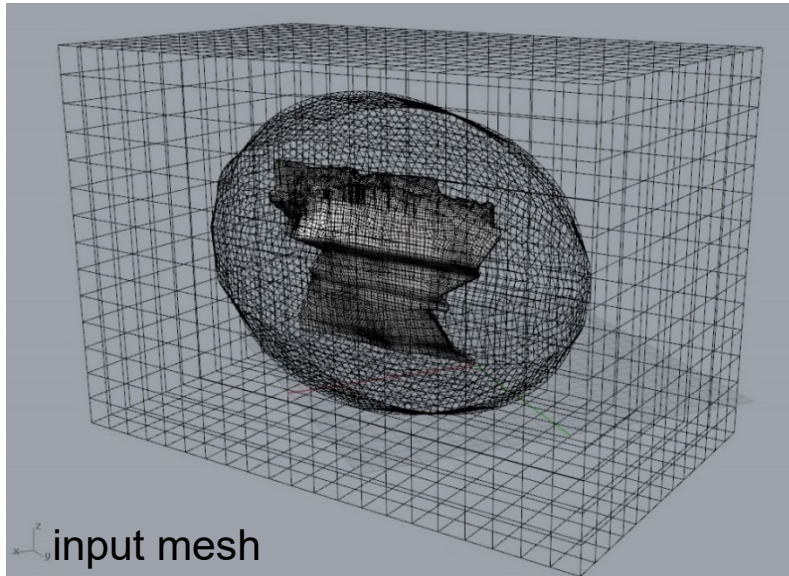
output mesh

- total number of elements: 362406
- hexahedra: 91.94% of volume
- prisms: 0.92% of volume
- pyramids: 5.40% of volume
- tetrahedra: 1.75% of volume
- number of nodes: 176110



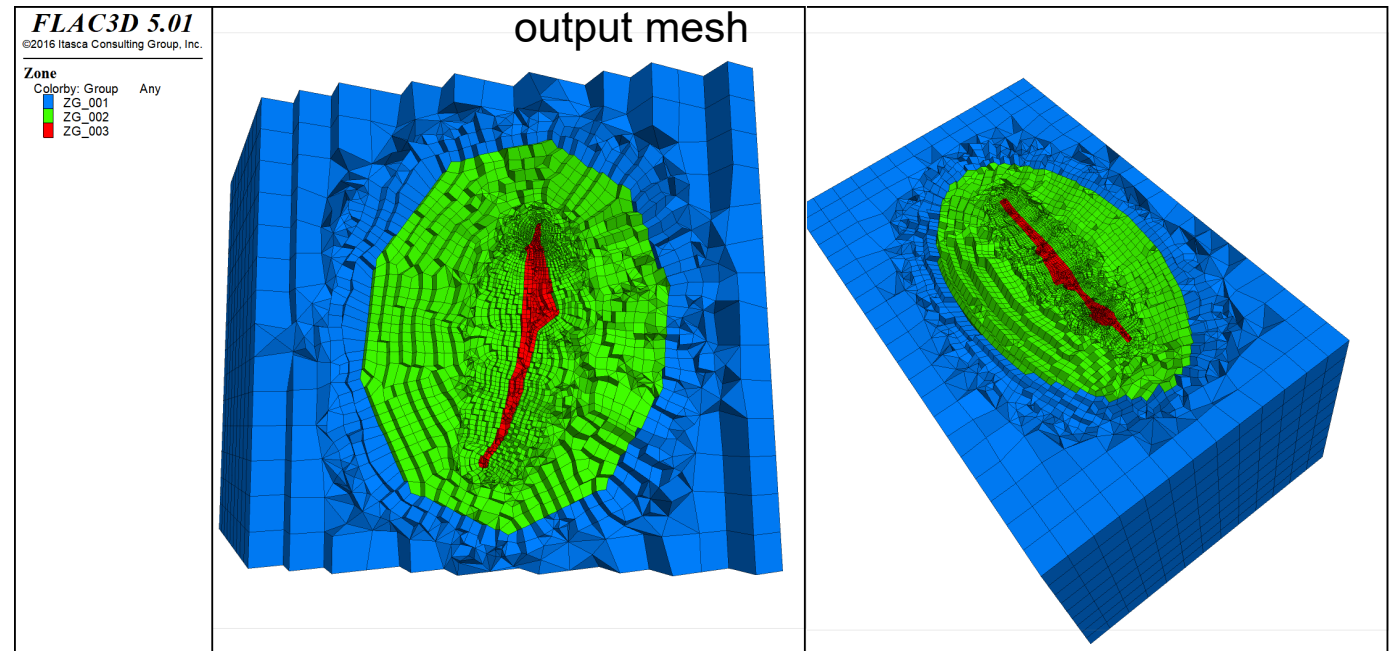
Volume mesh with *Griddle* GVol

A practical solution for the rather abrupt change in the mesh density could be the use of concentric surfaces between the center and the boundaries of the model with gradually increasing mesh density.



total number of elements: 479631
hexahedra: 79.81% of volume
prisms: 2.62% of volume
pyramids: 13.15% of volume
tetrahedra: 4.42% of volume
number of nodes: 236490

number of surface elements: 26198
number of nodes: 25205



Summary & Conclusions

3D preprocessing and mesh generation with *Griddle* in *Rhino*

- Effective, quick and straightforward method to produce a volume mesh of complex geometries
- Existing data can be easily imported in *Rhino* and remeshed with *Griddle* to produce volume meshes that can be moved across different platforms of numerical software
- The most time consuming tasks are initial mesh corrections and defeaturing
- Volume mesh may show a localized transition of element size instead of a smooth gradation



Thank You for Your Attention!