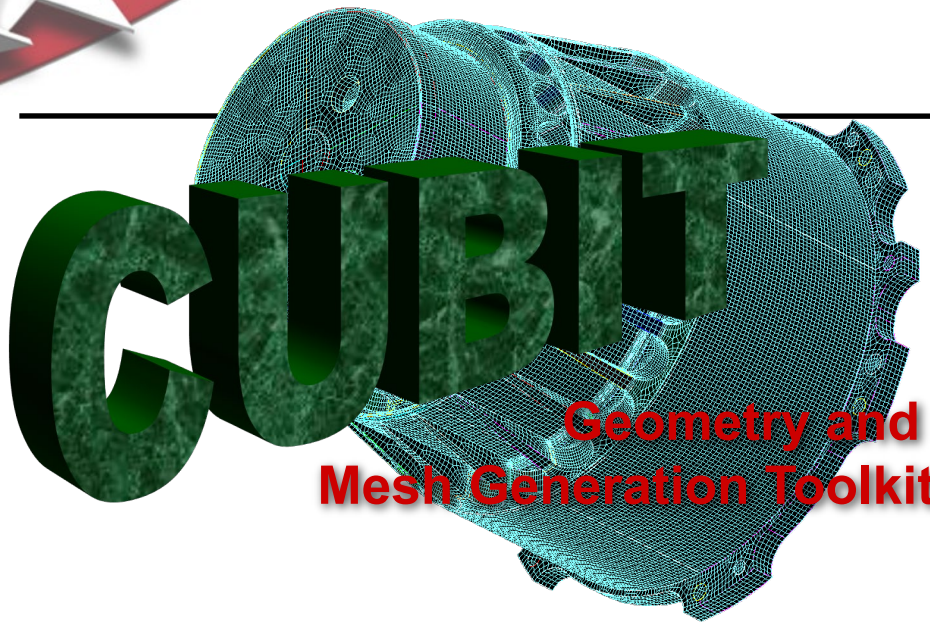


Simulation Modeling Sciences

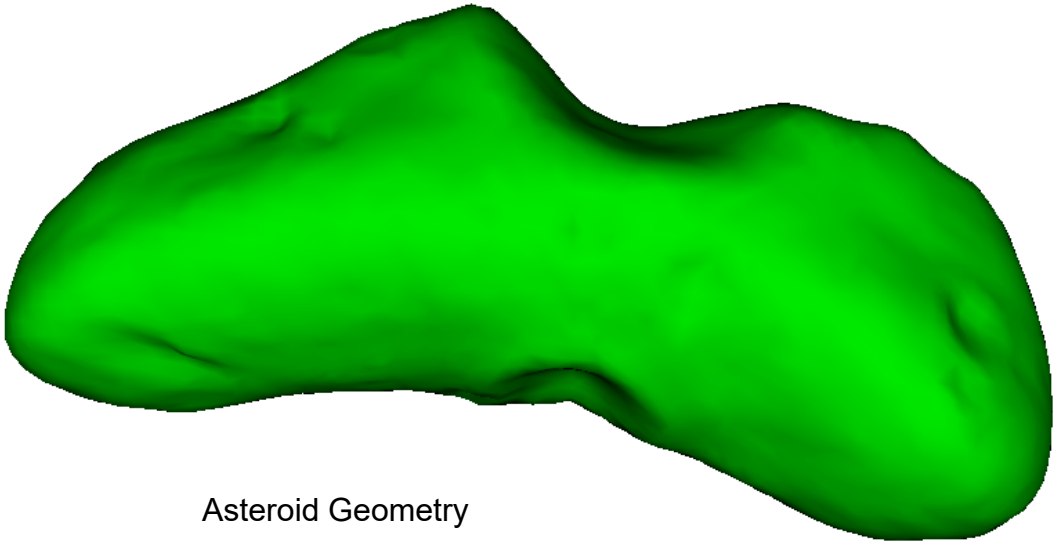
CUBIT Fast-Start Tutorial

21. Sculpt

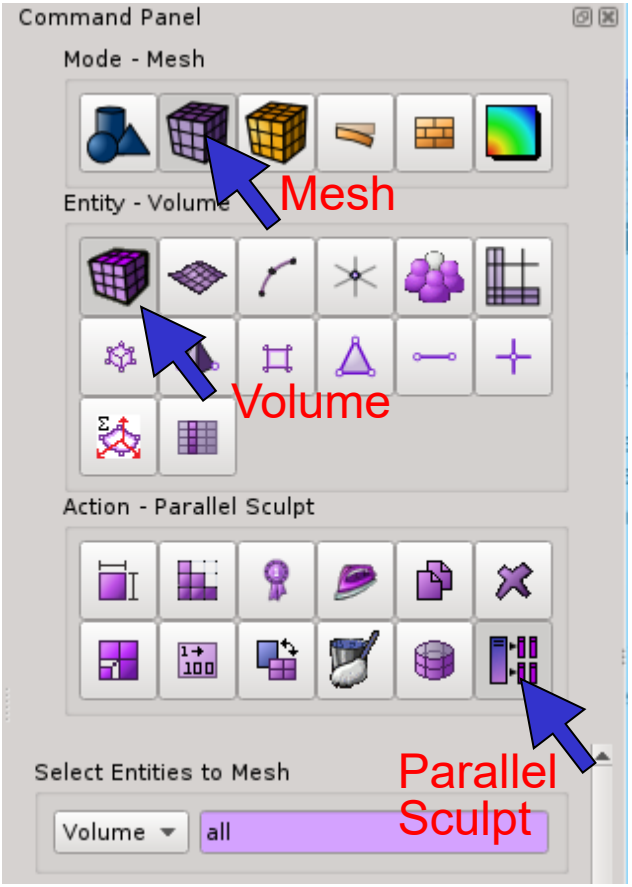
Sandia National Laboratories

Example 1. Simple Sculpt Mesh

1. Import STL file “eros_asteroid.stl”
2. Navigate in Command Panel Window to Parallel Sculpt Panel
3. Select the Parallel Sculpt button



Asteroid Geometry

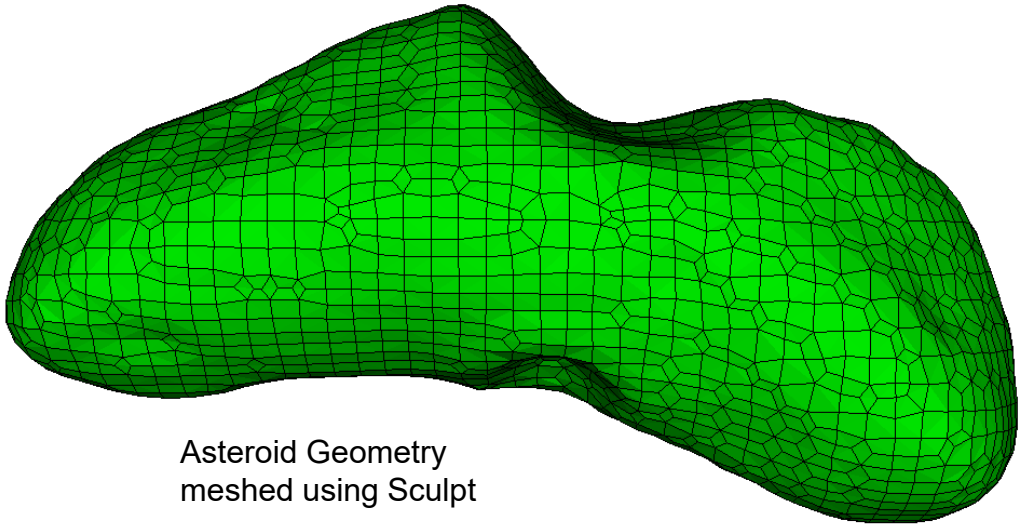


Example 1 Simple Sculpt Mesh

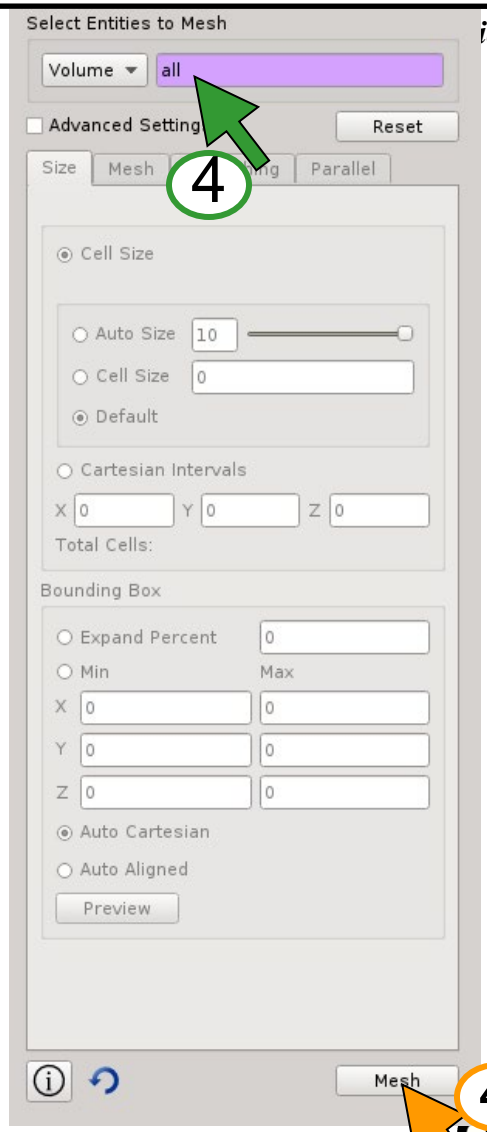
4 Make sure “All” or “1” is entered next to Volume

5 Click the Mesh Button

6 In Command Window issue the command **Draw Block All**



Asteroid Geometry
meshed using Sculpt



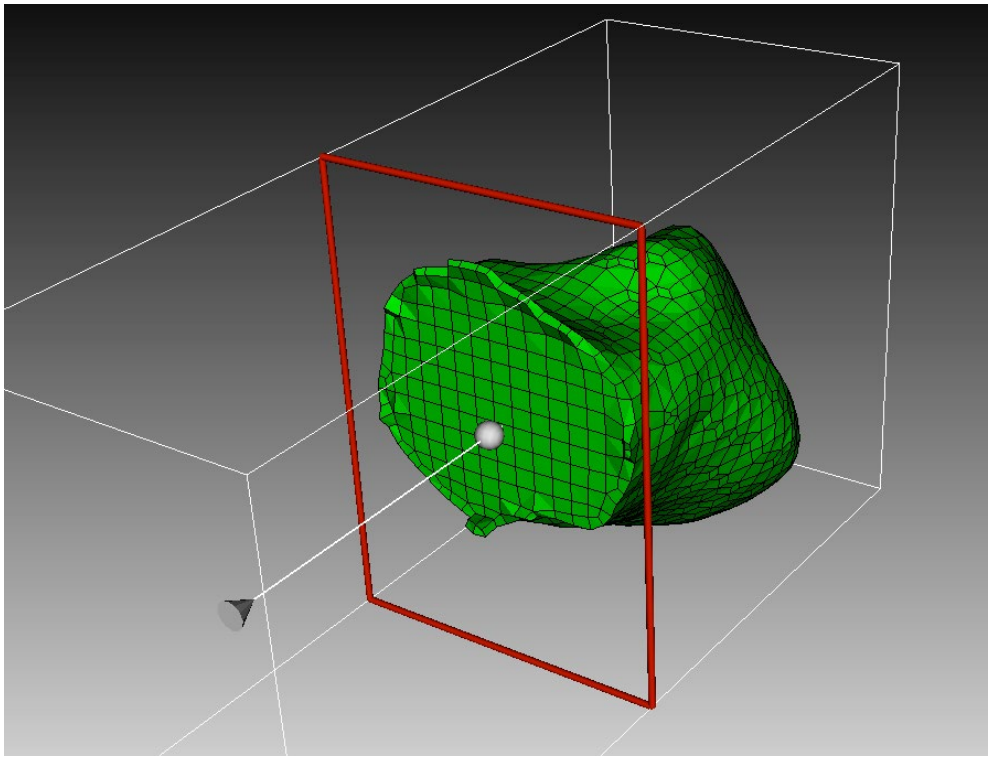
Sandia National Laboratories

Example 1 Simple Sculpt Mesh

6 Select the slice tool to view the interior of the mesh

Sculpt generates a uniform grid of hexes on the interior, but fits a layer of hexes at the boundary

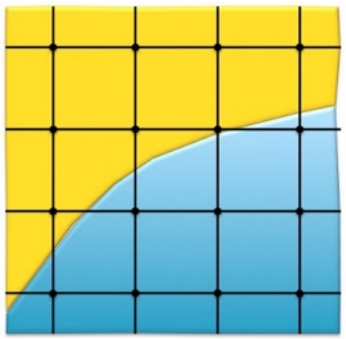
Sculpt creates a “Free Mesh” with no association to the geometry



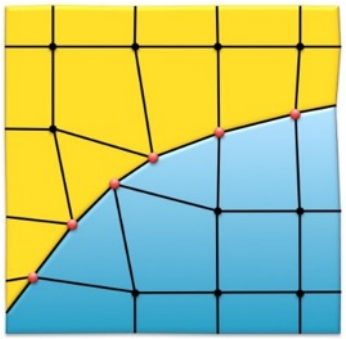
Slice through Asteroid Mesh

Sculpt Procedure

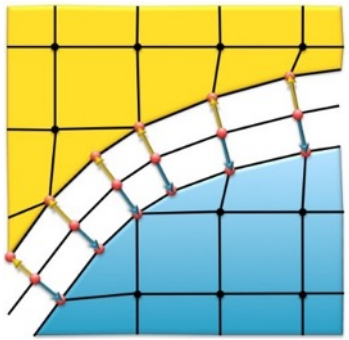
Overlay Grid Procedure used by Sculpt



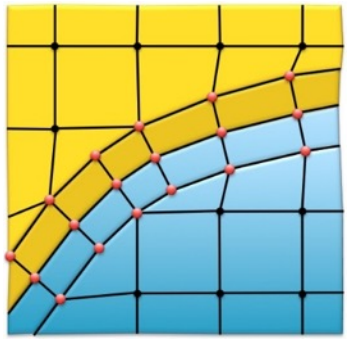
Cartesian Grid overlaid on geometry



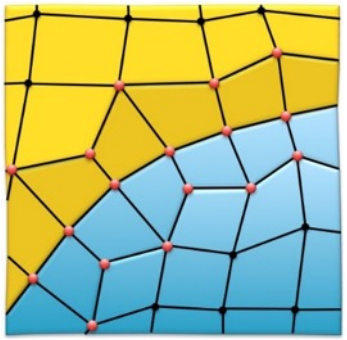
Nearby nodes projected to boundaries



Mesh pulled away from the boundaries



And Layer of hexes inserted



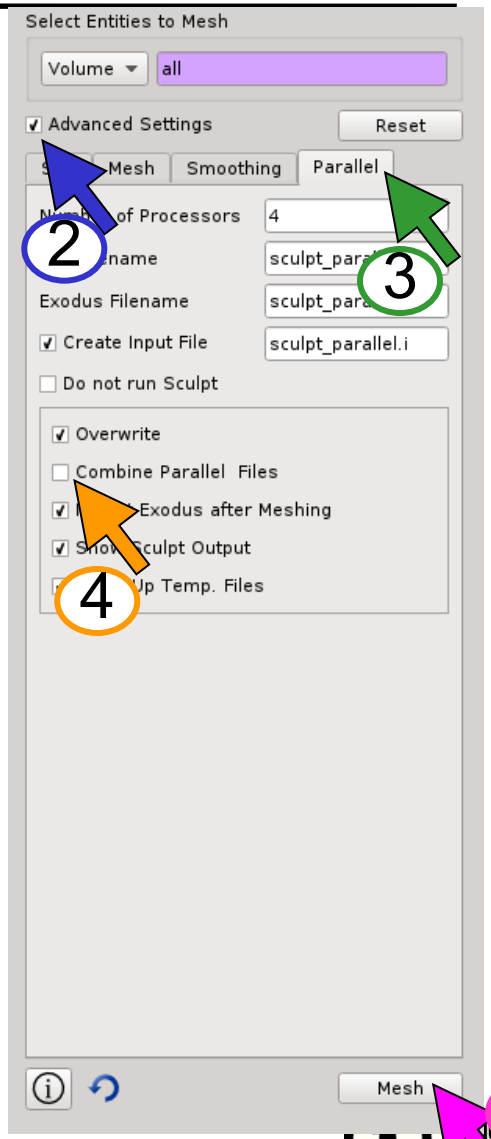
Smoothing performed to improve element quality

Example 2. Sculpt Application

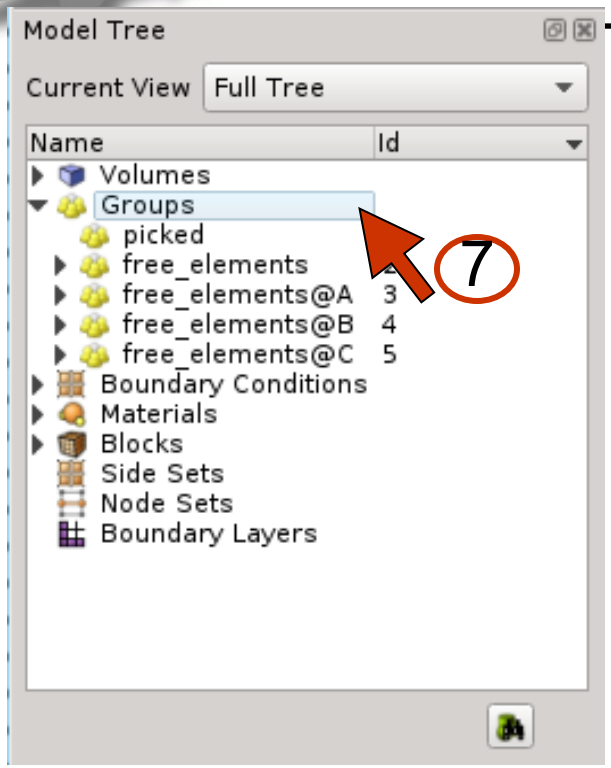
Sculpt is a separate application from Cubit that runs in parallel on multiple processors

1. First, delete the current mesh by typing **Delete Mesh** in the Command Line Window
2. Click on Advanced Settings
3. Select the Parallel Tab
4. Unselect the option to Combine Parallel Files
5. Click the Mesh Button
6. In Command Window issue the command **Draw Block All**

In this case we have generated 4 separate meshes. One mesh from each processor



Example 2. Sculpt Application



Observe the Tree View in the Power Tools Window

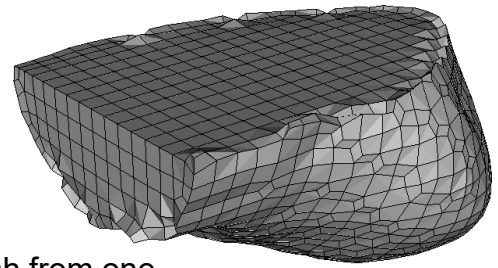
7 Select the Groups to reveal the current groups in Cubit

Note that several groups with the base name "free_elements" are created.

Each group represents the set of elements generated on one processor. Nodes and faces are not shared between groups (because **combine** option was not used)

8 Draw each group independently to see the mesh each processor generated. From the command line use:

Draw group 3
Draw group 4
Draw group 5
Draw group 6

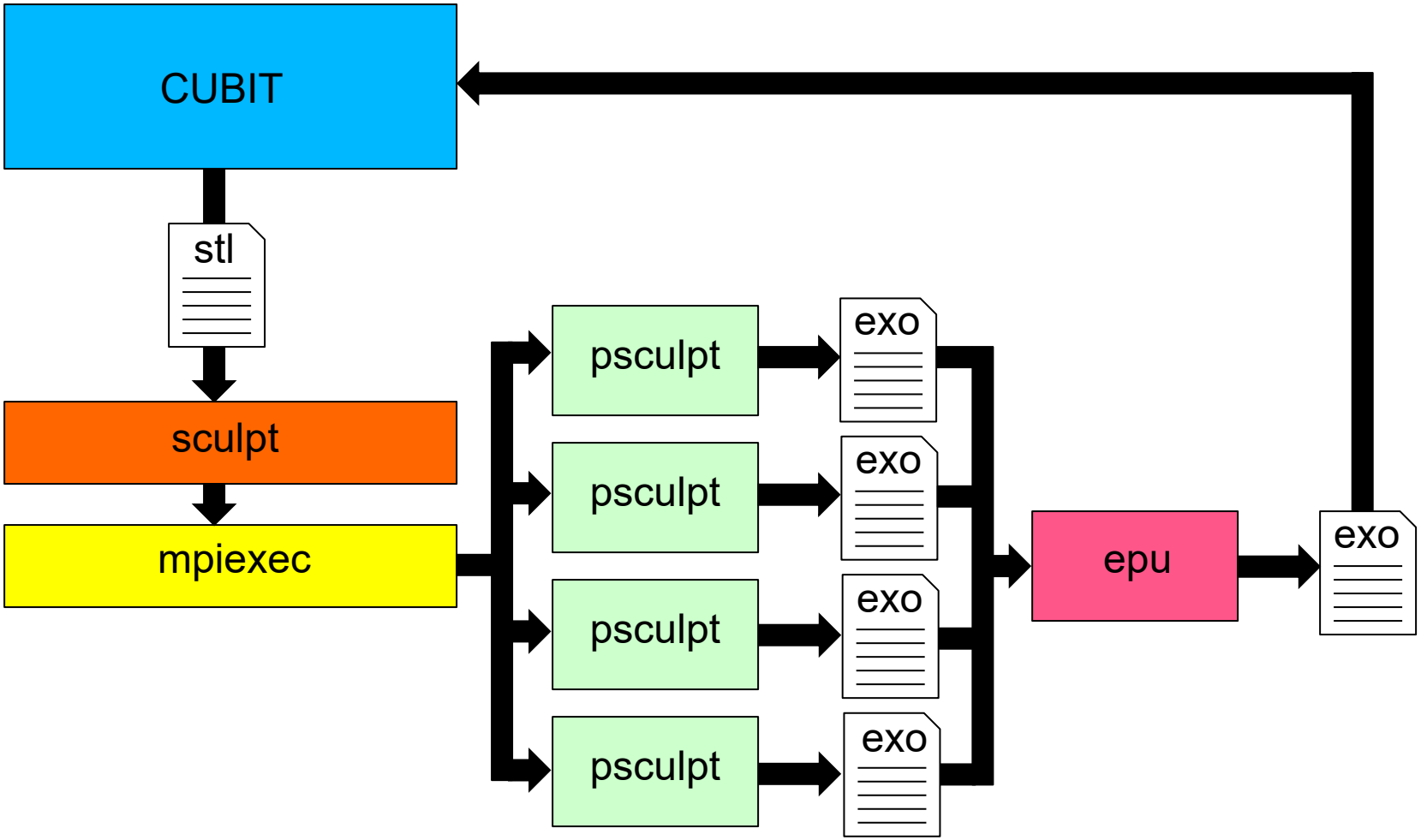


Mesh from one processor

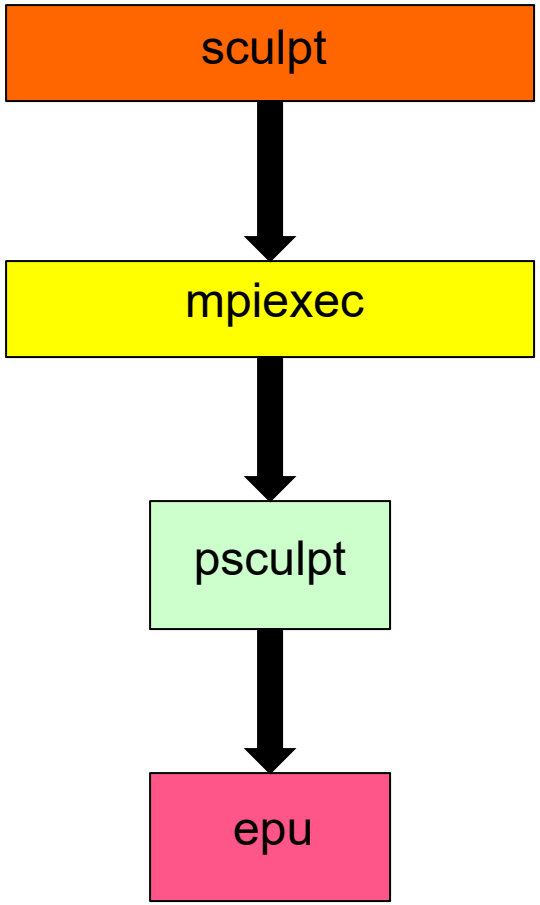
Tip: Groups and Blocks are not deleted when the mesh is deleted. Use **delete block all** and **delete group all** to get rid of them.

Sculpt Application

Simulation Modeling Sciences



Sculpt Application



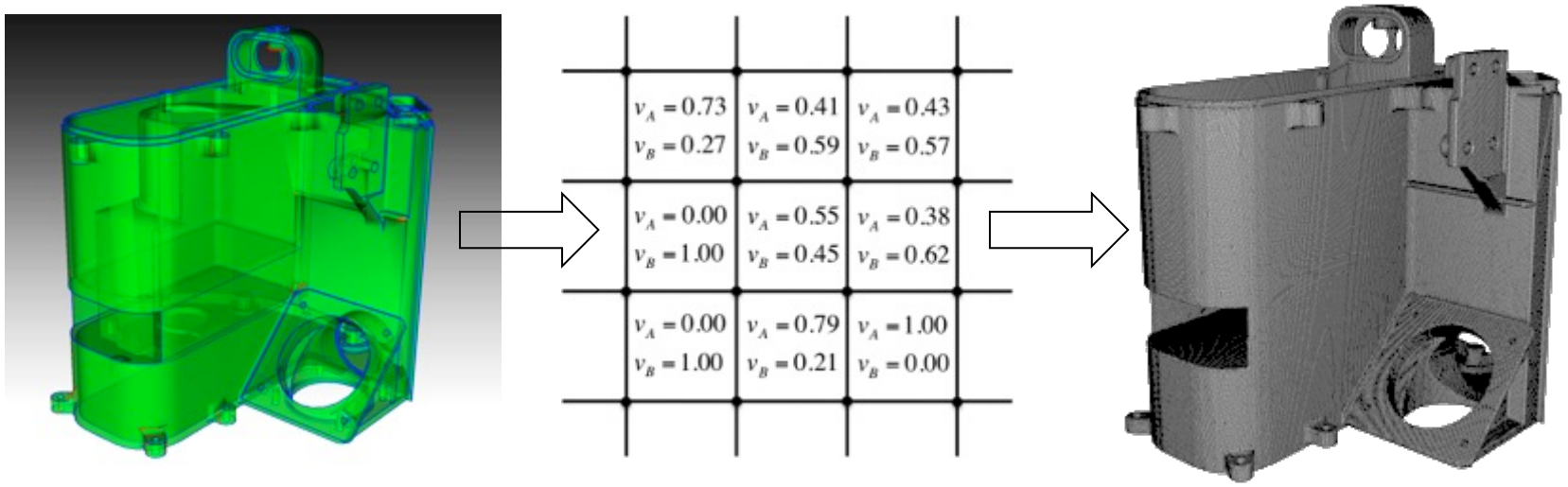
sculpt: Application that controls start-up of **mpiexec** and **psculpt**. Main entry point from Cubit, that checks for the existence and compatibility of either the system **mpiexec** application or will use a local cubit installation of **mpiexec**.

mpiexec: Standard application available on most linux-based operating systems for starting up mpi-based applications on multiple processors. This should be available with your Cubit installation, but is also available from open-mpi.org

psculpt: The main mpi-based Sculpt application. Requires **mpiexec** to run.

epu: Used for combining multiple exodus files, generated with Sculpt, into a single exodus file. This executable is optional, but is useful for importing the resulting mesh into Cubit for viewing. It is part of the SEACAS tool suite developed by Sandia National Laboratories and is also included with your Cubit installation. It can also be obtained in open source form from sourceforge.net.

Sculpt Application



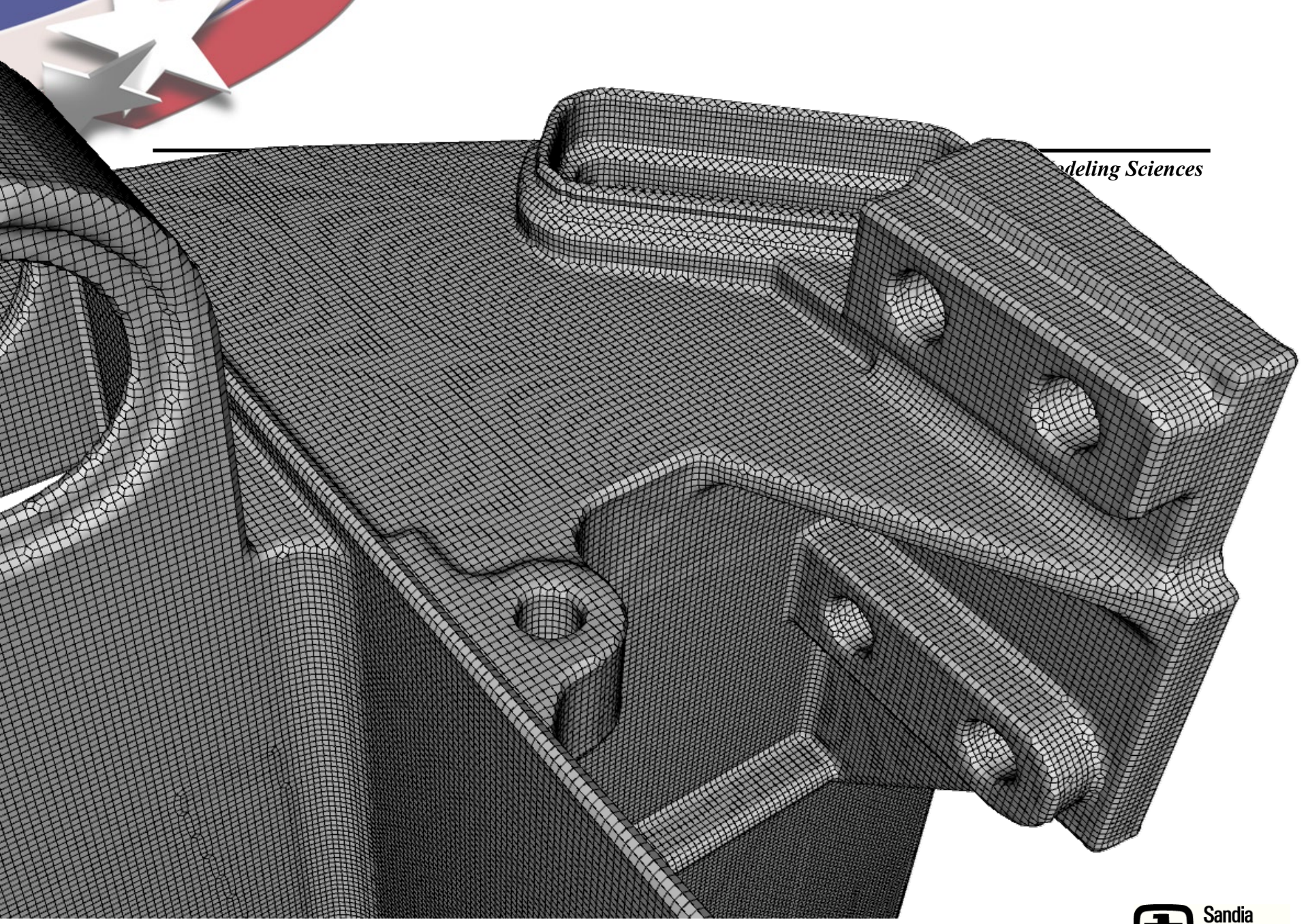
Geometry is first converted to “Volume Fraction” Data before meshing.

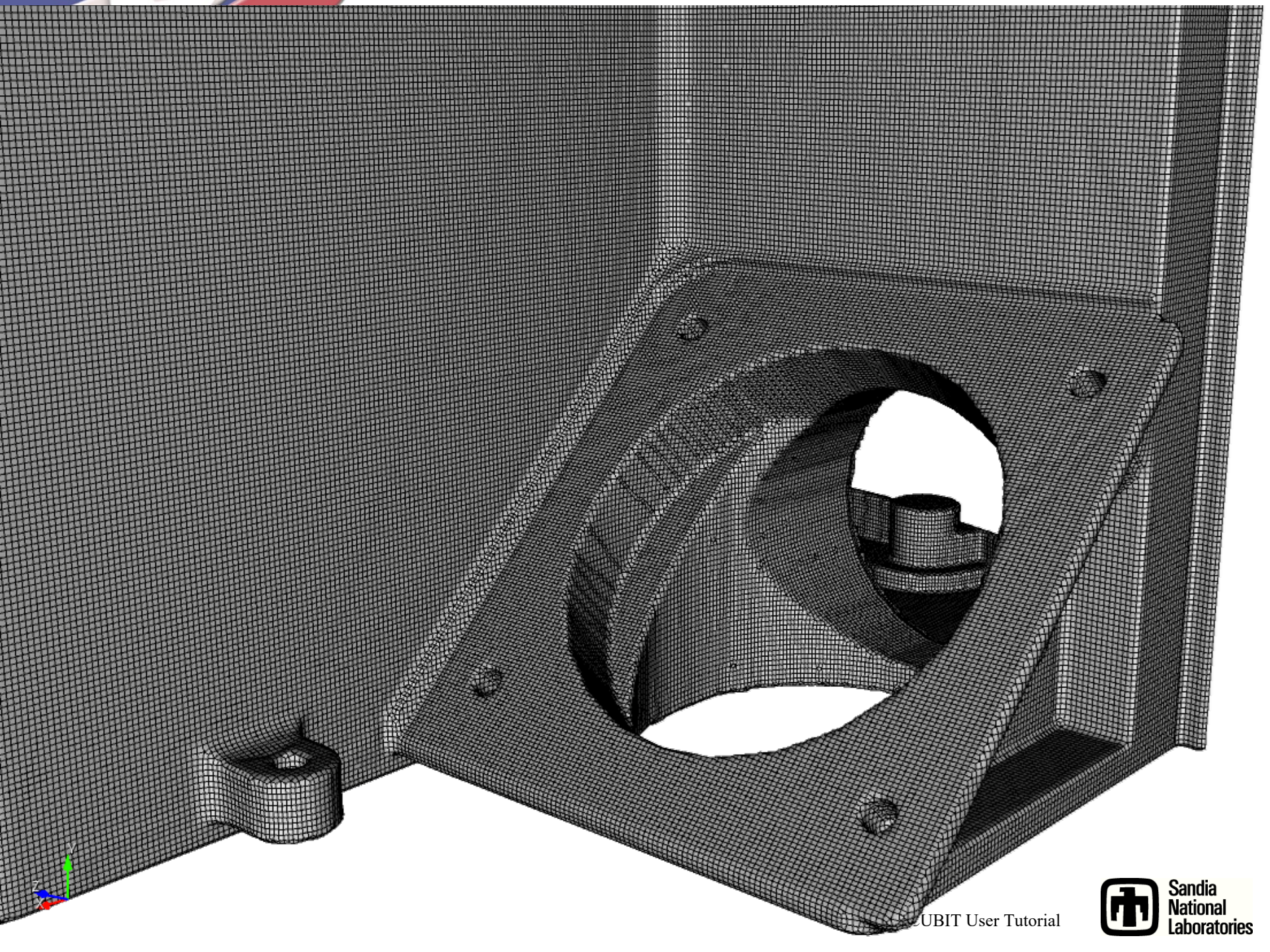
Advantages

- Geometry Tolerant
- No decomposition needed
- No imprint/merge needed
- Scalable to massive parallel

Disadvantages

- No Geometry Associativity
- Does not capture corners
- “Small” features not resolved
- May require large mesh

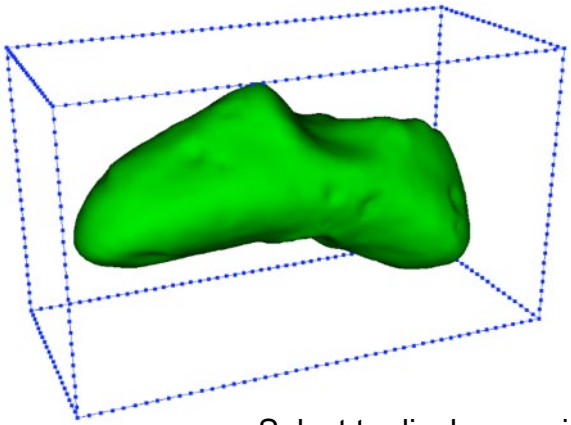




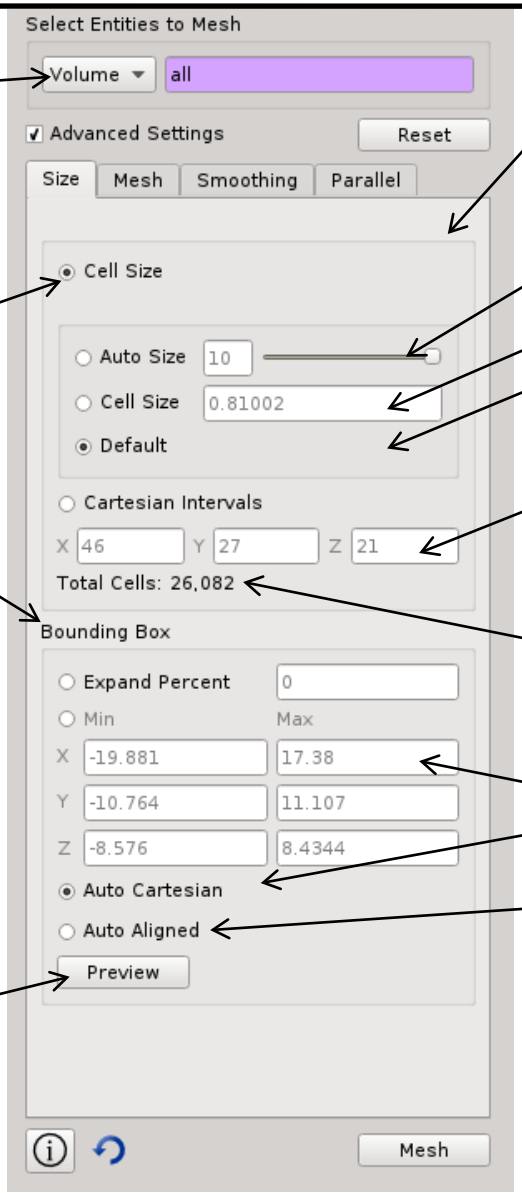
Sculpt Sizing

Select Blocks or Volumes
Volumes: Each volume treated as a separate mesh. One Block created per volume
Blocks: treat all volumes in block as continuous mesh

Base Grid Requires
1. Cell Size
2. Bounding Box



Select to display preview of bounding box and size



Resets all values in panel back to default. Use this if volumes/blocks change

4 ways to define size

1. Autosize – value 0-10. 0=fine, 10=coarse
2. Cell size – exact size
3. Default – use the size defined on the volume. If none then autosize=10
4. Cartesian Intervals – set the number of intervals in x-y-z directions. Tries to maintain constant aspect ratio

Number of cells in Cartesian Grid

3 ways to define box

1. Set exact x-y-z bounds
2. Auto Cartesian – Complete model enclosed in axis-aligned box
3. Auto Aligned – Sculpt attempts to find a natural orientation of the model with the a fitted box

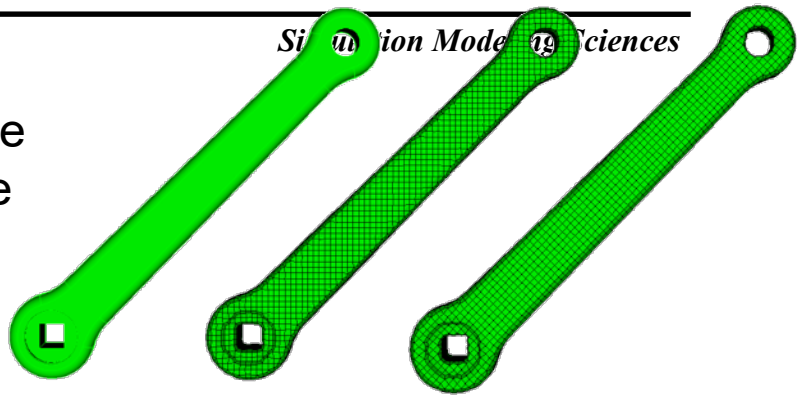
Note: Fields automatically synchronize to reflect updated user selections

Example 3 Sculpt Sizing

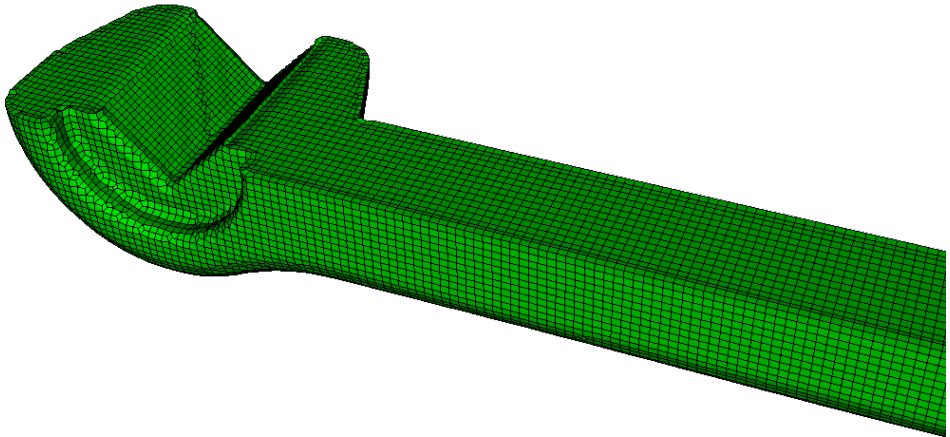
1. Import the ACIS file “crank.sat”
2. Experiment with different mesh sizes to see its effect. Observe how well it captures the features.
3. Use the Auto Align feature. Observe the difference in the resulting mesh
4. How large of a mesh can you generate in under a minute or so.
5. Create an approximate ½ symmetry model using the bounding box feature.

   Hint: First rotate the model

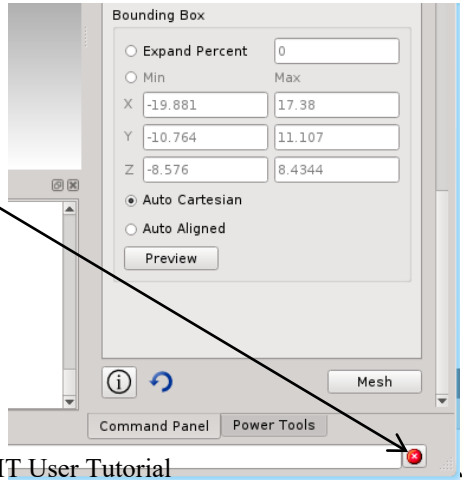
   Vol 1 rotate -45 about z



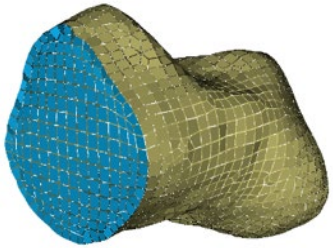
Note the **Total Cells** estimate. This is total number of cells in the base grid (not mesh size) based on current input. Larger numbers indicate longer meshing times.



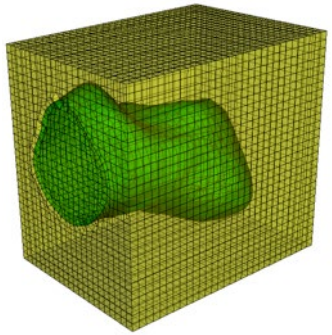
Use the Stop button to kill sculpt if meshing time gets too long.



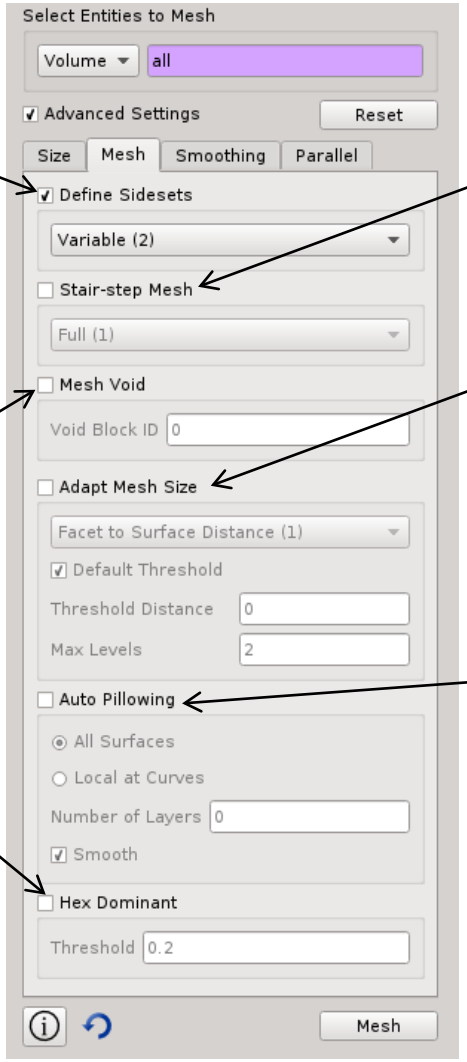
Sculpt Mesh



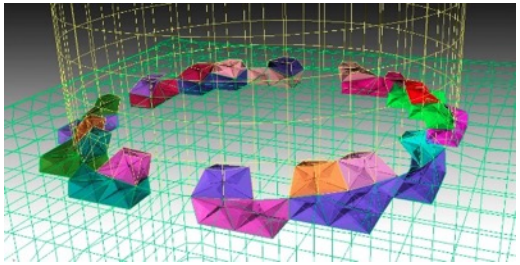
Sidesets: Options for generating sidesets

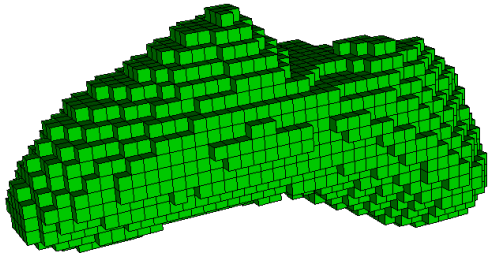


Mesh Void: Generate mesh in surrounding void space. Void will be treated like another material



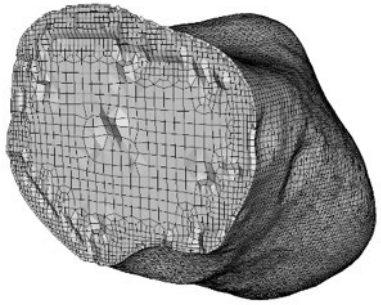
Hex Dominant: Replace any hex whose Scaled Jacobian falls below the threshold with 24 tets



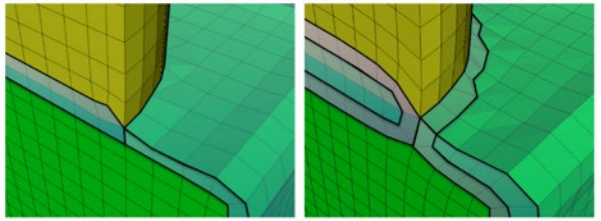


Stair-step mesh: Don't insert buffer layers or smooth to boundaries. Mesh will be 100% cubes

Adapt Mesh Size: Automatically refine at features and boundaries



Auto Pillowing: Insert additional layer(s) of hexes to improve mesh quality at multi-material interfaces (curves)



Note: Turn this ON when using multiple materials

Sculpt Smoothing

Note: In most cases smoothing defaults should be sufficient

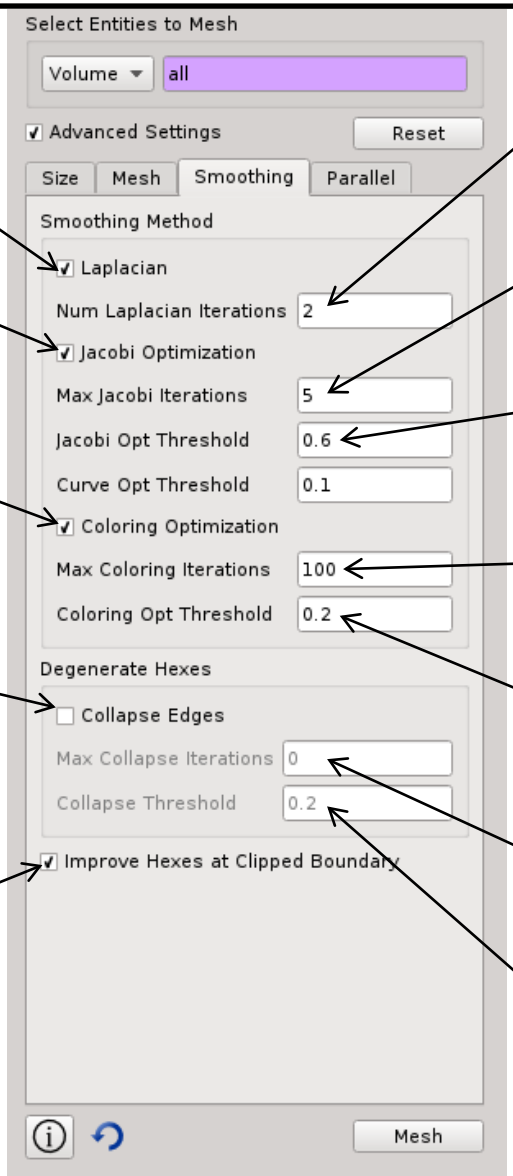
Laplacian: Fast/Cheap method for mesh quality improvement. Used alone can sometimes make quality worse.

Jacobi Optimization: Targeted, more effective in improving quality. Less efficient.

Coloring Optimization: Spot smoothing that treats each node and surrounding hexes independently. Applied only at worst quality elements. Most expensive.

Collapse Edges: Allow Sculpt to collapse edges to create degenerate hexes in order to improve mesh quality.

Improve Hexes at Clipped Boundary: If checked, Smoothing will attempt to improve hexes that are at the clipped domain boundary without regard to constraining to the bounding planes. If not checked, boundaries will be planar.



Num Laplacian Iterations: Number of Laplacian Iterations to perform. (Each iteration effects all nodes in the mesh)

Max Jacobi Iterations: Number of Jacobi optimization Iterations to perform. Stops after this number of iterations or when all elements exceed threshold.

Jacobi Opt. Threshold: The lowest scaled Jacobian value in which Jacobi optimization smoothing will be applied.

Max Coloring Iterations: Number of Coloring optimization iterations to perform. Stops after this number of iterations or when all elements exceed threshold.

Coloring Opt. Threshold: The lowest scaled Jacobian value in which coloring optimization smoothing will be applied.

Max Collapse Iterations: Number of edge collapse iterations to perform. Stops after this number of iterations or when all elements exceed threshold.

Collapse Threshold: The lowest scaled Jacobian value in which edge collapses will be applied.

Sculpt Parallel

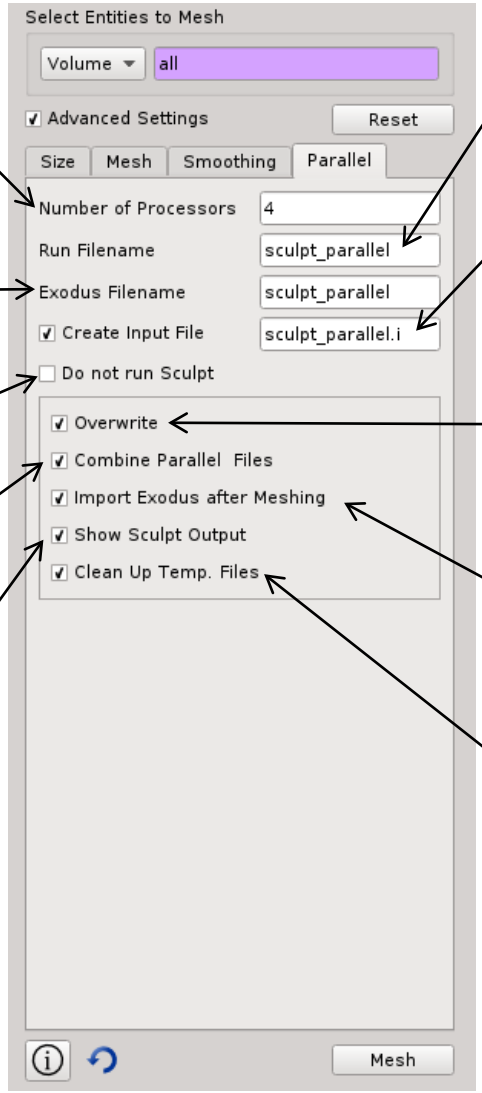
Number of Processors: Sculpt will decompose the domain into the specified number of subdomains for meshing. It is normally the total number of available compute cores (ie. 4, 8, etc.), however it can be any integer value.

Exodus Filename: Optionally define the base name of the exodus files to be written, otherwise will use the run filename.

Do not run Sculpt: Cubit will only generate the necessary files to run sculpt in batch. It will not generate a mesh.

Combine Parallel Files: Automatically run EPU on the resulting exodus mesh files to combine them into a single mesh file.

Show Sculpt Output: While running Sculpt from Cubit, the text output from Sculpt will be displayed to the Cubit Command Window



Run Filename: Create an executable file (eg. sculpt_parallel.run) containing the OS command line for running sculpt in batch.

Create Input File: Create a sculpt input file containing all defined parameters in a readable form. This is for running sculpt in batch from the OS command line. For example:
sculpt -i sculpt_parallel.i -j 4

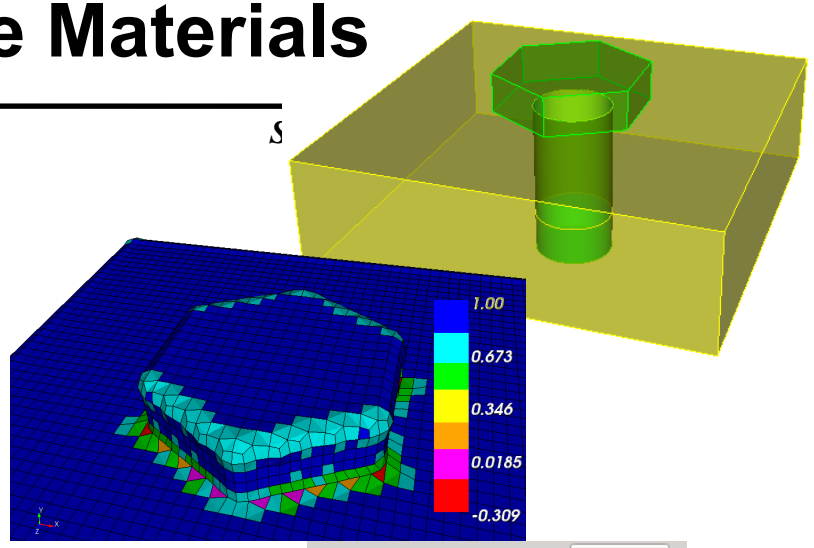
Overwrite: If not checked will prevent execution if *run*, *input* or *exodus* files of the same name are already present in the working directory.

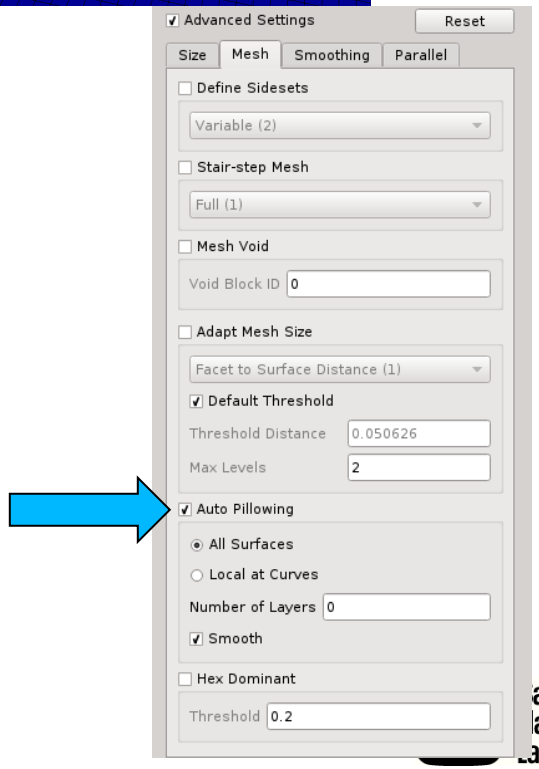
Import Exodus After Meshing: After running Sculpt, Mesh will be imported into Cubit as “Free Mesh”.

Clean Up Temp. Files: After successfully generating the mesh with Sculpt, all temporary files will be deleted. ie. run file, input file, log file, mesh files and STL files.

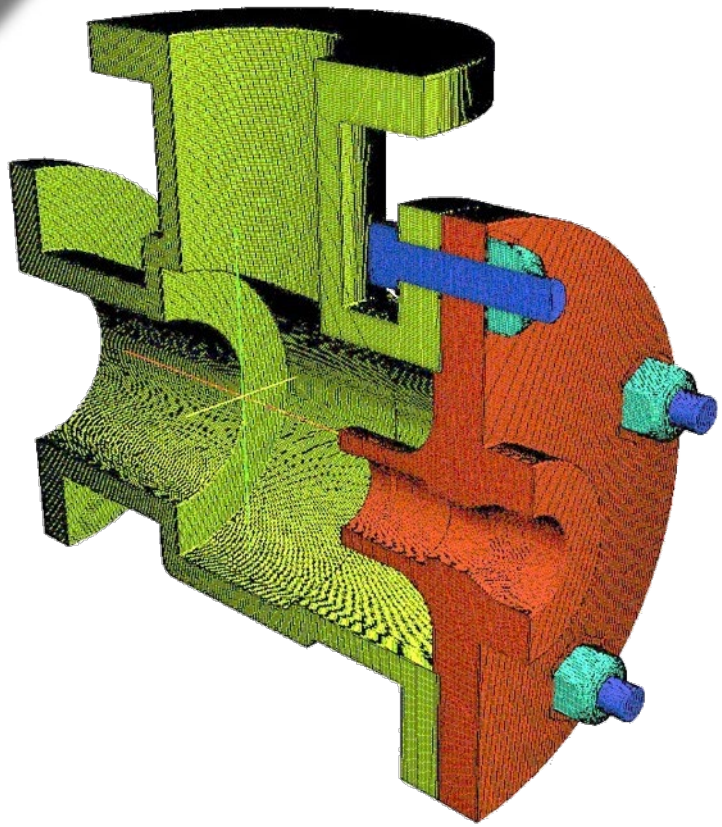
Exercise 4 Multiple Materials

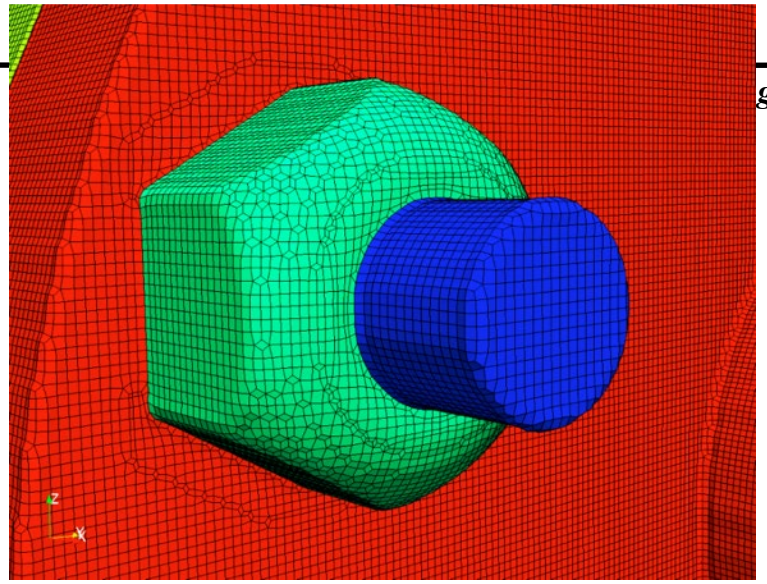
1. Import the model “bolt-plate.sat”
2. Generate a mesh with a cell size of about 0.2
3. Display the mesh quality using the Scaled Jacobian Metric. Note where the worst quality elements are concentrated.
4. Delete the mesh
5. Select the Auto Pillowing option and mesh again.
6. Check Mesh Quality. What is the difference?
7. Use the slice tool to examine the interior of the mesh
8. Try using the **All Surfaces** and the **Local At Curves** option. What is the difference in the resulting mesh?

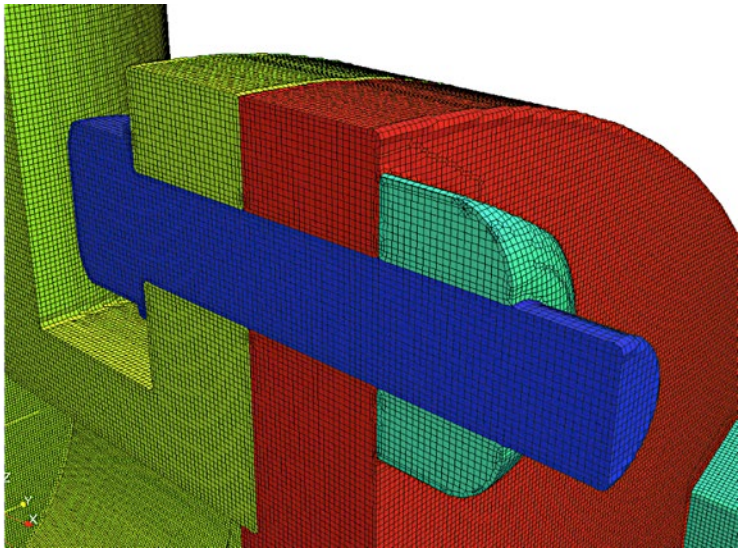




Example: Auto Pillow Insertion

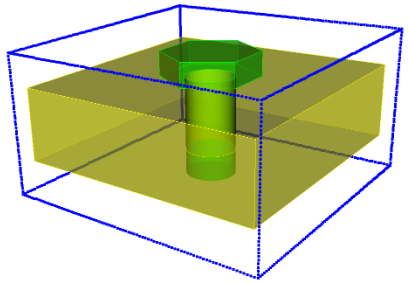






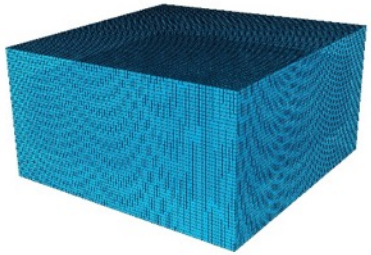
Example of mesh generated with Sculpt using the Auto-pillow option to capture multiple materials

Sculpt Sidesets

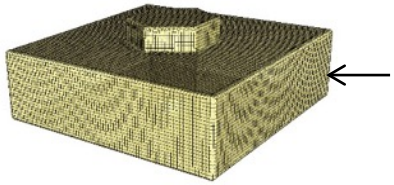


Sculpt provides four different methods for generating and maintaining Sidesets

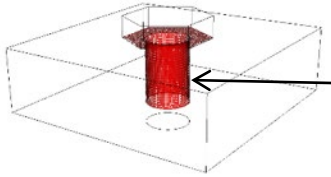
1. **Fixed**: Exactly three sidesets are generated



Sideset 1: all sides at the six sides of the original Cartesian grid. These are present only if the Void option is selected, or the geometry intersects the boundary



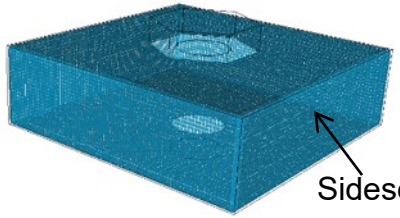
Sideset 2: all sides that are at a free surface of any block of elements

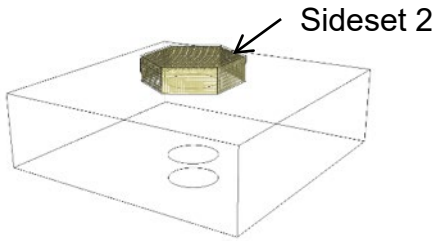


Sideset 3: all sides that define the interface between different blocks

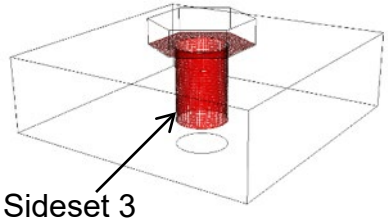
2. **Variable**: Sidesets generated at every unique surface, either at a free surface or between blocks

In this example there a 4 sidesets generated

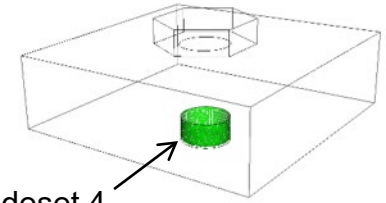




Sideset 1



Sideset 3
(between block 1 and 2)



Sideset 4

6 sidesets are also generated for each face of the Cartesian grid if using the Void option. Sidesets are also generated where the geometry is cut by the boundary.

Sandia National Laboratories

Sculpt Sidesets

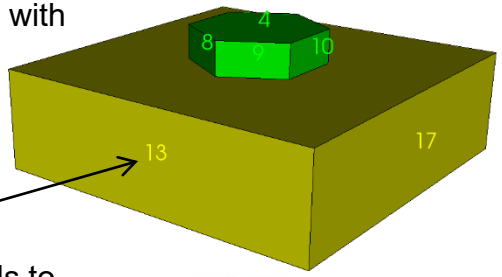
Sculpt provides four different methods for generating and maintaining Sidesets

Note that Nodesets are not yet supported in Sculpt

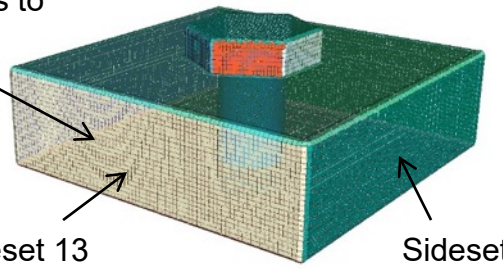
3. Geometric Surfaces: One sideset generated for each geometric surface defined in the CAD model

In this example there a 17 sidesets generated. One for each surface in the CAD model

CAD model with surface IDs labeled



Sideset 13 Corresponds to surface 13

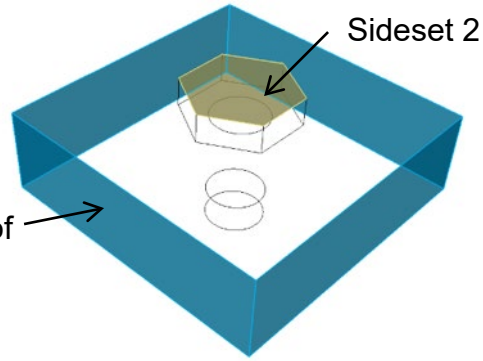


Sideset 13

Sideset 17

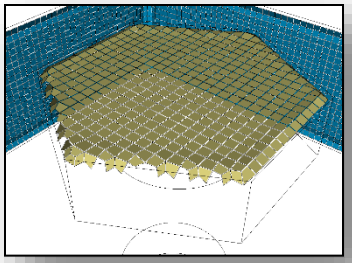
4. Geometric Sidesets: Sidesets generated based on sidesets defined on the CAD model

User defines 2 sidesets on the solid model

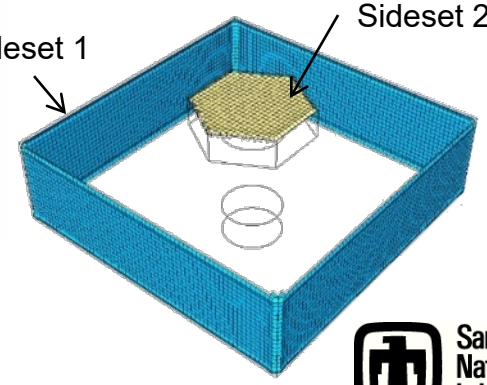


Sideset 2

Sideset 1 (composed of 4 surfaces)



Note that sides are only an approximation of the surface. They may not exactly align with bounding curves



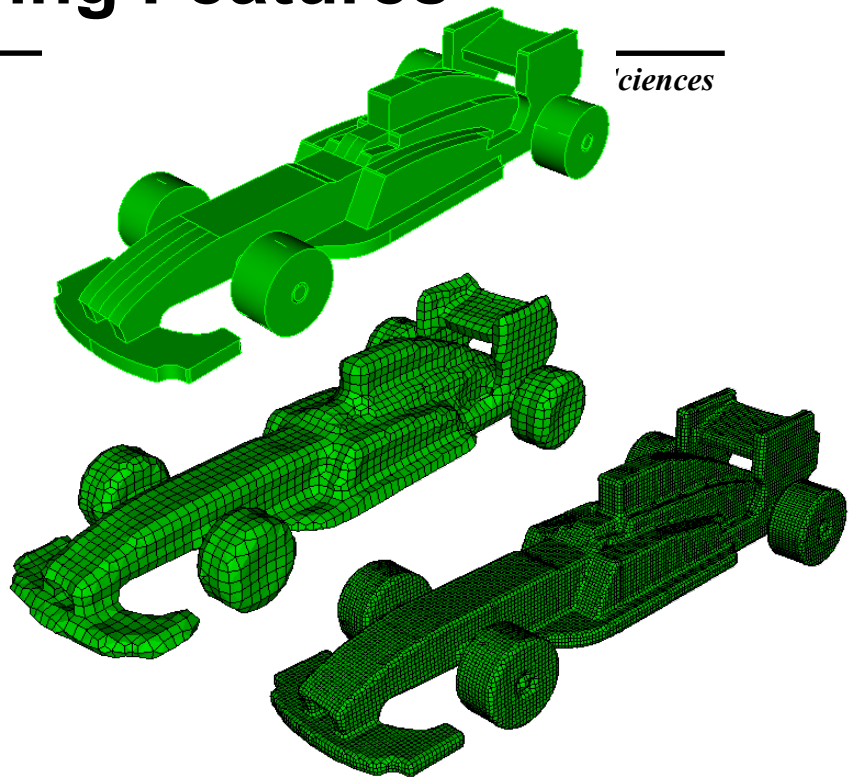
Sideset 1

Sideset 2

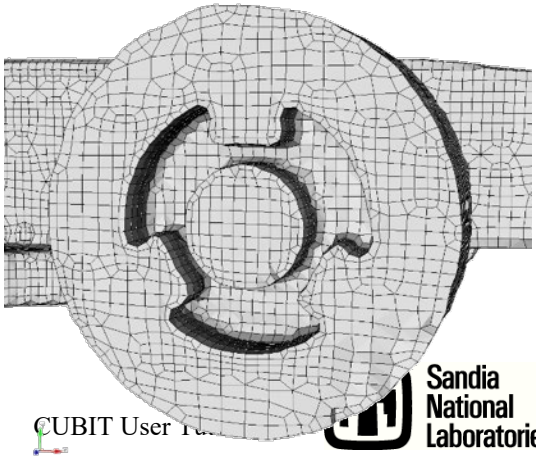
Sandia National Laboratories

Exercise 5. Capturing Features

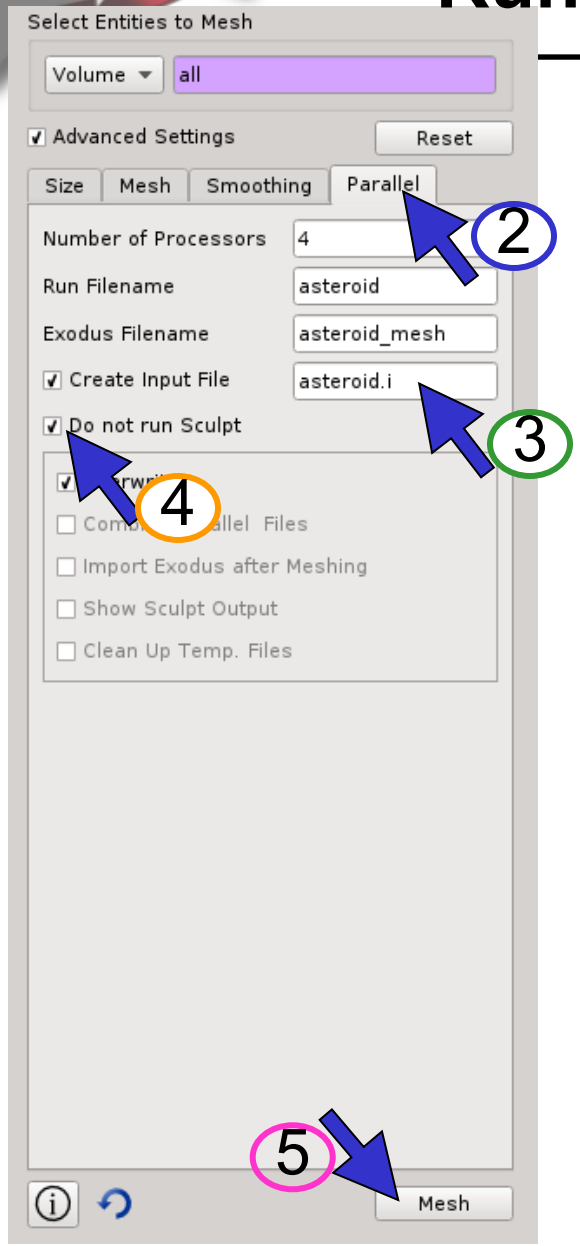
1. Import the model "**B3.sat**"
2. Generate a mesh using the default settings. Note the mesh quality and how well the mesh represents the geometry.
3. Try meshing with a smaller size. At what size do you need to accurately capture the features in this model.
4. Are there Simplifications you can make to the model that may make feature capture more reasonable?
5. Reset and Import the model "**B3-2.sat**". Note the difference in the features
6. Try meshing again. What cell size do you need to capture the geometry?
7. Mesh the "**B3.sat**" model using the adapt option and try to capture the small features in the hubs. Compare the number of elements without adapt.



Adapt option used with 2 levels of refinement and default threshold

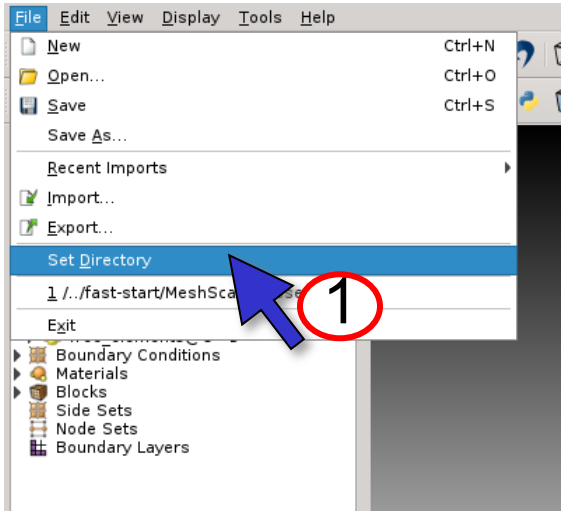


Running Sculpt in Batch



Sculpt can be run in batch on your local desktop or on an HPC machine

To generate batch input files



1. Set your working directory
2. Click the Parallel Tab
3. Fill in names for files or use default (sculpt_parallel)
4. Select Do Not Run Sculpt
5. Click Mesh

With the *Do Not Run Sculpt* option, no mesh will be generated. Instead the batch input files will be written to your working directory

Running Sculpt in Batch

In your working directory you should see the following files

Diatom File: Contains a list of all geometry used as input to Sculpt

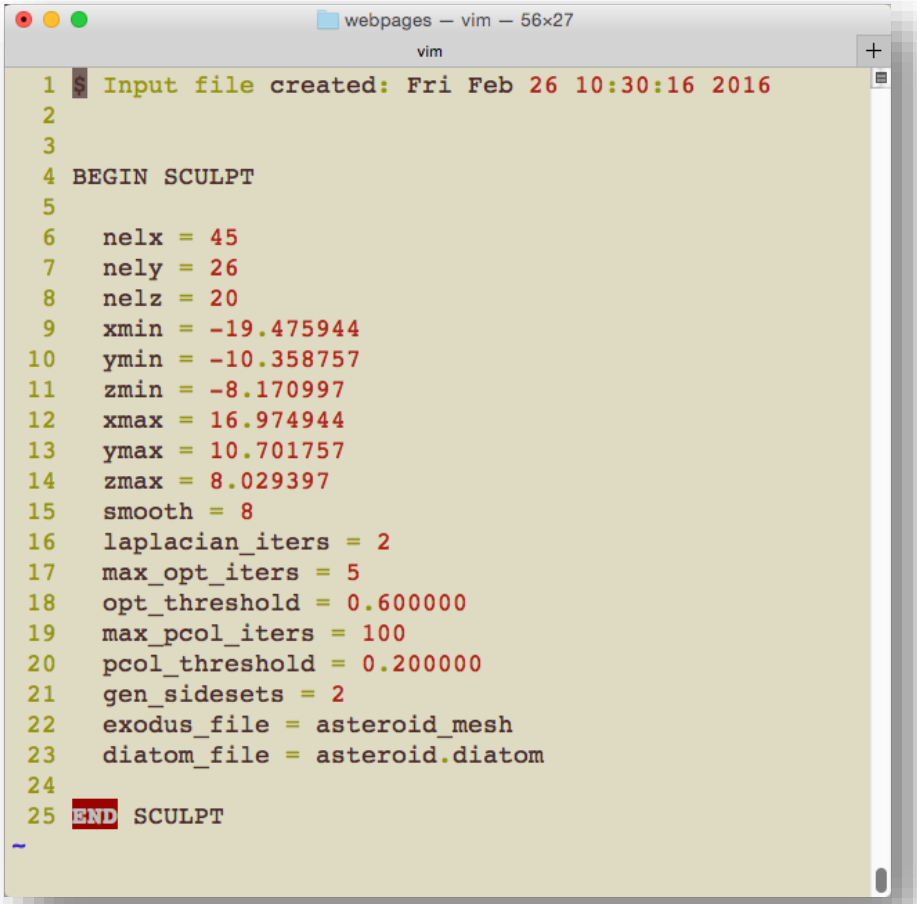
Input File: Contains a human readable form of all input parameters to sculpt

```
sjowen@S974435:~/cubit/scratch$ ls -l
total 3856
-rw-r--r--  1 sjowen  staff      160 Feb 26 10:30 asteroid.diatom
-rw-r--r--  1 sjowen  staff      440 Feb 26 10:30 asteroid.i
-rwxr-xr-x  1 sjowen  staff      283 Feb 26 10:30 asteroid.run
-rw-r--r--  1 sjowen  staff  1960613 Feb 26 10:30 asteroid_Volume_1.stl
sjowen@S974435:~/cubit/scratch$
```

Run File: Contains a unix command line script to run sculpt with all parameters

STL File: Triangle-based geometry description. One per volume

Running Sculpt in Batch



Edit the **.i** file to view the sculpt input parameters

For a description of parameters, use the –h option on the unix command line

Type:
`sculpt`

To see a summary of all sculpt options

Type:
`sculpt –h <option>`

To see help on the any sculpt option

Running Sculpt in Batch

To run sculpt in batch:

`sculpt –i <inuput file> -j <num_procs>`

For example:

`sculpt –i asteroid.i -j 4`

```
================ MESH SUMMARY ==================
Base Filename    asteroid_mesh
Num Procs        4
Num Nodes        8533
Num Elements     7211
Num Blocks       1
Num Nodesets     0
Num Sidesets     1
Num Bad Qual     0
Num Poor Qual    0
Min Quality      0.251209
Avg Quality      0.868188
Min Edge Len     0.225095
Min Qual Rank    3

Job Completed Fri Feb 26 10:50:16 2016
Elapsed Time              0.871374 sec. (0.014523 min.)
Total Time on 4 Procs     3.431796 sec. (0.057197 min.)
Slow Rank                 1
Done!
```

Observe the mesh summary when the mesh is completed

Number Hexes < 0.0 Scaled Jacobian

Number Hexes < 0.2 Scaled Jacobian

Minimum mesh quality

Running Sculpt in Batch


```
sjowen@S974435:~/cubit/scratch$ ls
asteroid.diatom
asteroid.i
asteroid.run
asteroid_Volume_1.stl
asteroid_mesh.e.4.0
asteroid_mesh.e.4.1
asteroid_mesh.e.4.2
asteroid_mesh.e.4.3
```

Exodus mesh files created (1 per processor)

To combine mesh files:

epu –p <num_procs> <base_file_name>

For example:

epu –p 4 asteroid_mesh

To import mesh in to Cubit to view mesh from command line

import mesh “asteroid_mesh.e” no_geom

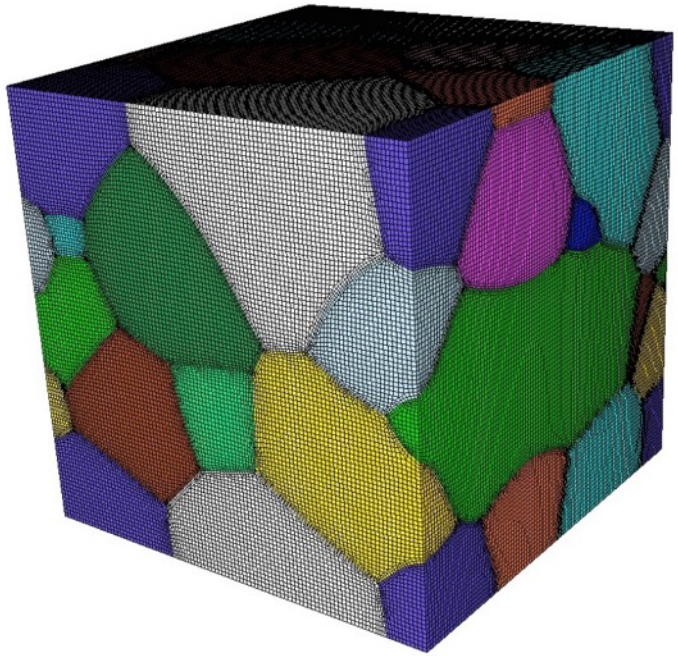
Example 6. Microstructures

input_micro option

Input is ASCII text file Cartesian Grid of Volume fraction material data

Required header info

```
TITLE = triple line system
VARIABLES = x y z, phi_1, phi_2, phi_3
ZONE i = 2 , j = 2 , k = 2
0.0000      0.0000      0.0000      0.5000      0.5000      0.0000
1.0000      0.0000      0.0000      0.3333      0.3333      0.3334
0.0000      1.0000      0.0000      1.0000      0.0000      0.0000
1.0000      1.0000      0.0000      0.0000      1.0000      0.0000
0.0000      0.0000      1.0000      0.2000      0.4000      0.4000
1.0000      0.0000      1.0000      0.6000      0.1000      0.3000
0.0000      1.0000      1.0000      0.0000      0.0000      1.0000
1.0000      1.0000      1.0000      0.9000      0.0000      0.1000
```



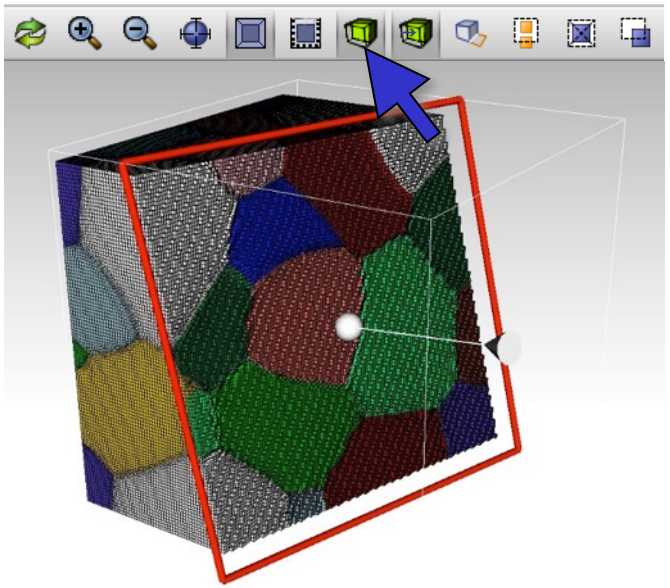
x, y, z cell centroid

Volume fraction for each material in variable list

Example 6. Microstructures

1. Create an input file (`micro.i`) with the following parameters

```
Begin sculpt
    input_micro = micro.tec
    exodus_file = micro
    micro_expand = 2
    pillow = 1
End Sculpt
```



2. Run sculpt with the following command

```
sculpt –i micro.i –j 8
```

3. Combine the parallel mesh files

```
epu –p 8 micro
```

4. Import the mesh into Cubit. From Cubit command line:

cubit >import mesh “micro.e” no_geom

5. Draw the material blocks

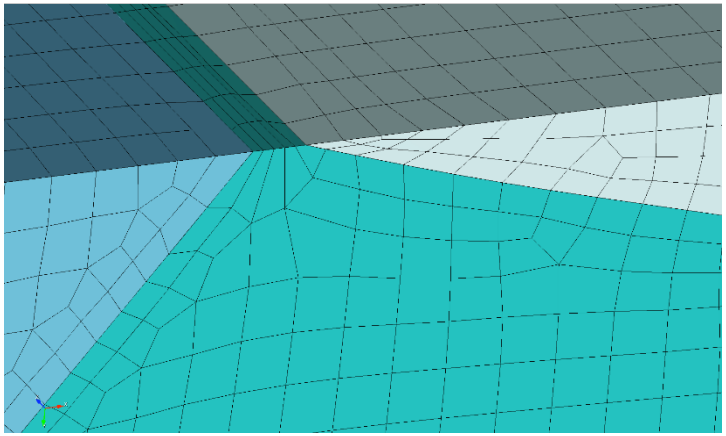
cubit >draw block all

Example 6. Microstructures

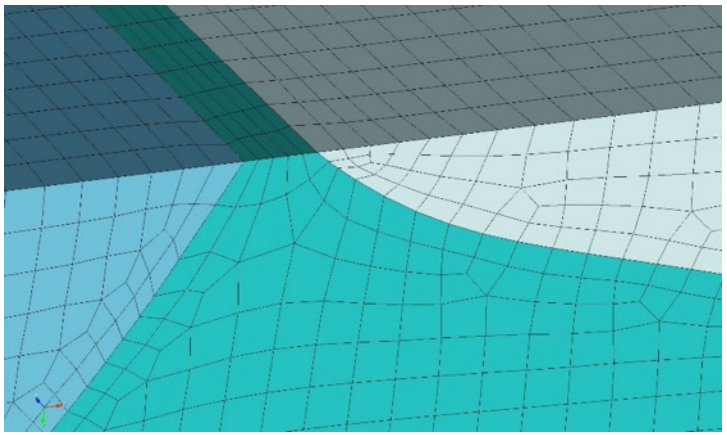
micro_expand option

Sculpt option to expand the block by N layers

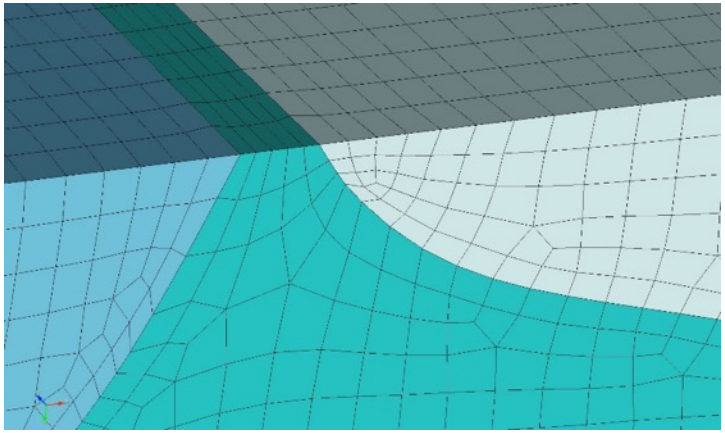
Affects the orthogonally of the surface interfaces at the boundary



No expansion



1 expansion layer



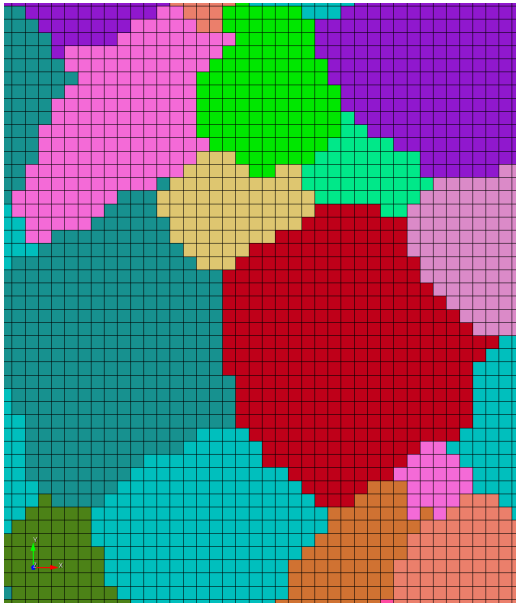
2 expansion layers

Example 7. Microstructures

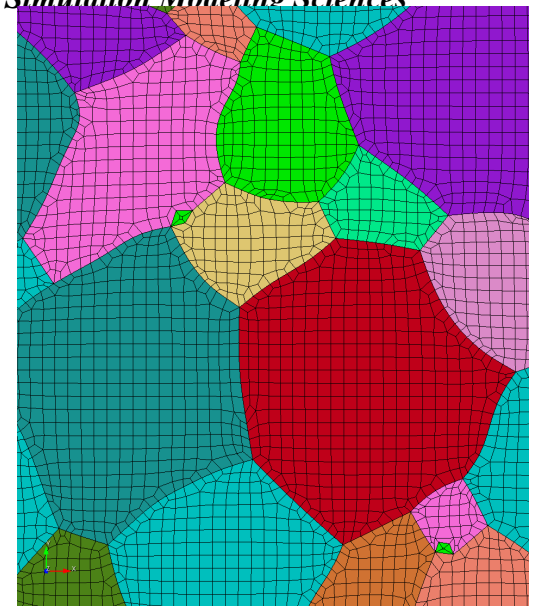
input_cart_exo

From an exodus file
1. Cartesian Grid
2. Each element assigned to a unique material block

Create a conformal mesh with smooth interfaces



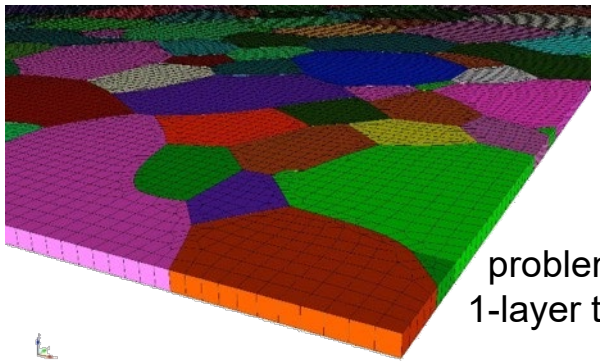
Cartesian Exodus File



Smooth material interfaces

1 Generate a conformal mesh with the following input file

```
Begin sculpt
    input_cart_exo = micro_2D.e
    exodus_file = micro_3D
End Sculpt
```



Note that this problem is actually a 1-layer thick 3D mesh

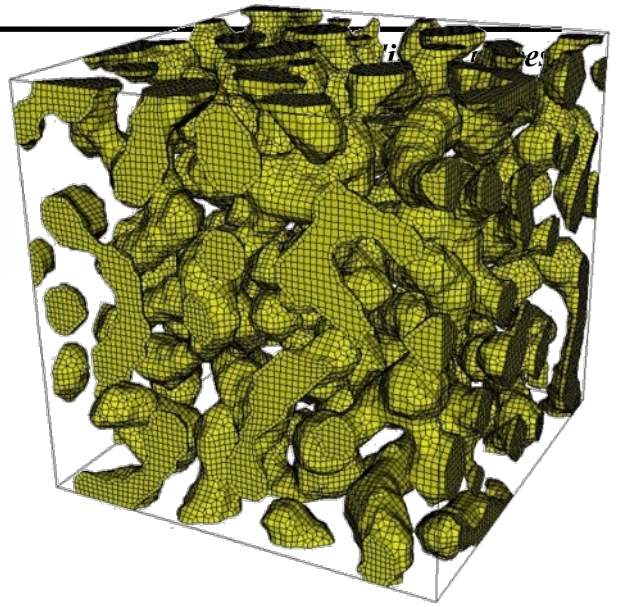
Example 8. Microstructures

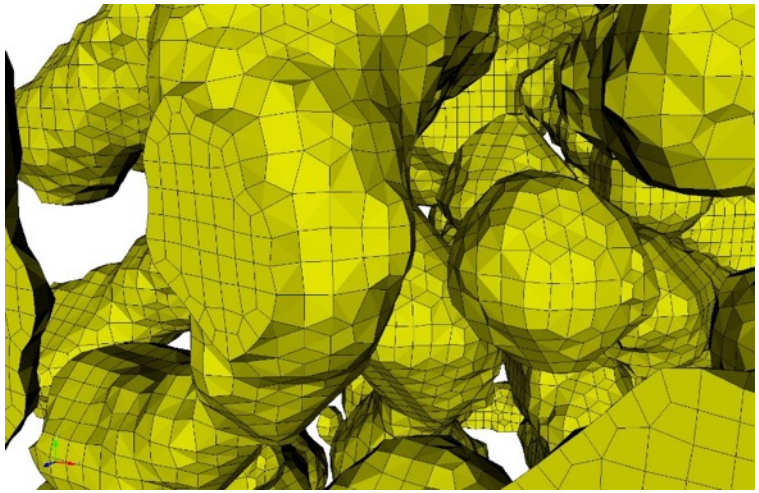
input_spn

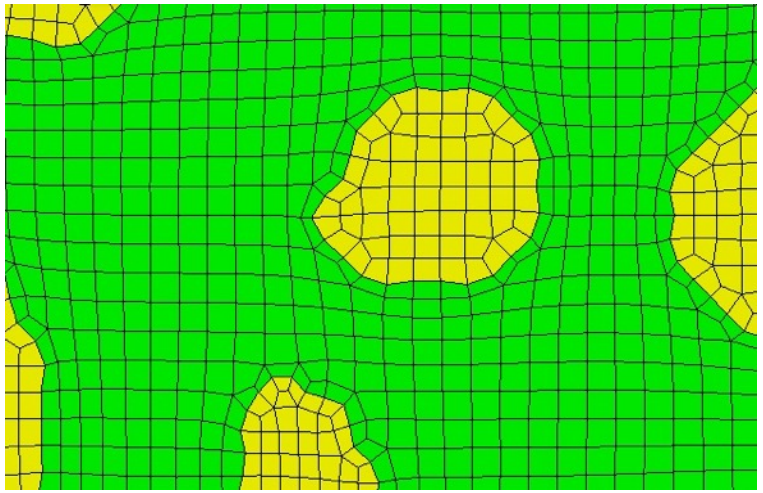
From an ascii file
1. One integer per cell
2. Integer represents dominant material in cell

```
1 1 2 1 2 1 1 1 1
1 2 2 1 2 2 1 1 2
2 1 1 1 2 1 1 2 2
```

Simple spn input file comprised of one integer per cell





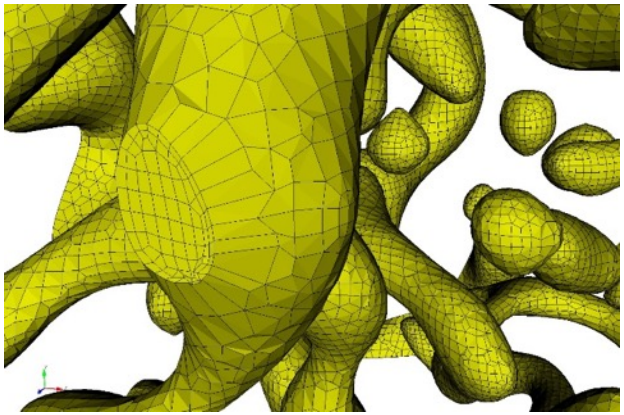


Example 8. Microstructures

input_spn

1. Generate a conformal mesh with the following input file
2. View the results in Cubit
3. Modify the smoothing options and view the results. Try the following options:

```
smooth = 9
csmooth = 2
```



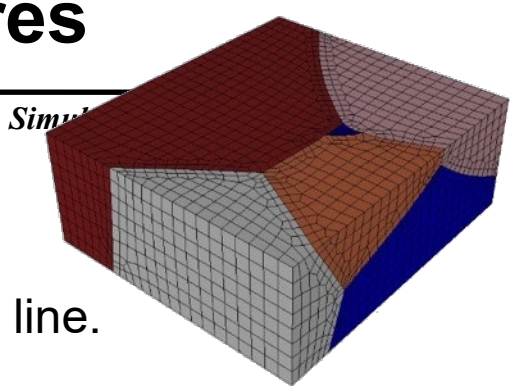
```
Begin sculpt
    input_spn = TwoPhase.spn
    exodus_file = TwoPhase

    # number of cells (required)
    nelx = 64
    nely = 64
    nelz = 64

    # smoothing options
    smooth = 8
    csmooth = 5
    laplacian_iters = 10
    max_opt_iters = 20

End Sculpt
```

Example 9. Microstructures



Sculpt can be used to generate mesh-based geometry that can be used in Cubit for tet meshing

1. Run the input file "**sculpt_micro9.i**" on the command line.

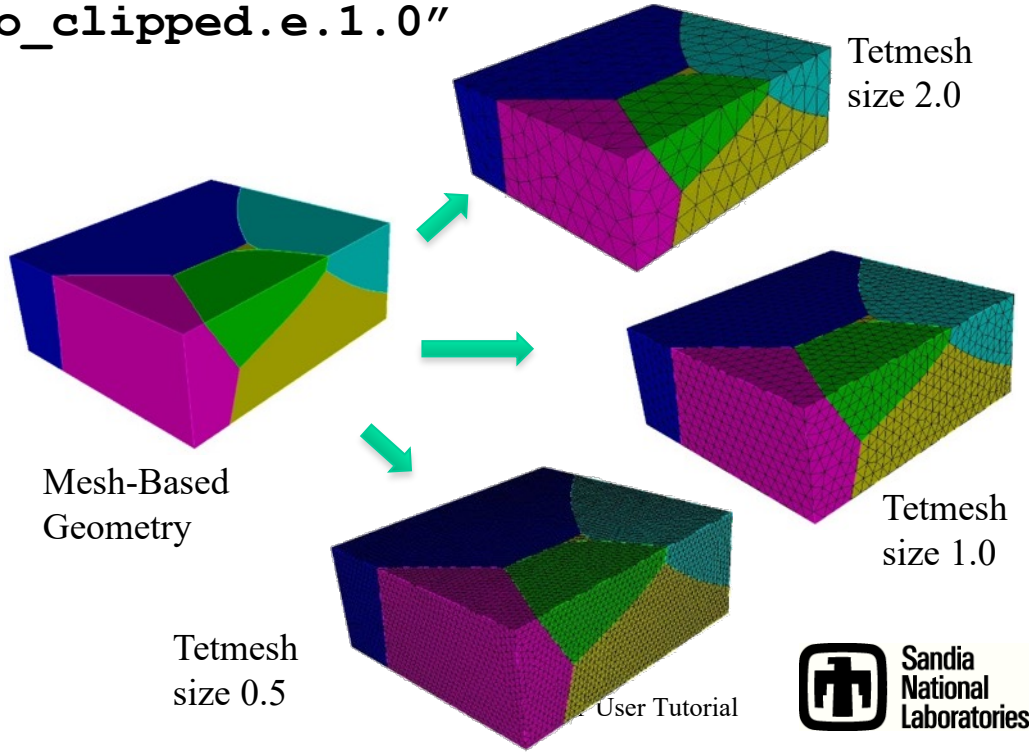
```
sculpt -i sculpt_micro9.i
```

2. Import the resulting exodus mesh into cubit as mesh based geometry

```
Import mesh geom "micro_clipped.e.1.0"
```

3. Delete the triangle mesh and generate a tet mesh

```
Delete mesh
Vol all size 0.5
Vol all scheme tetmesh
Mesh vol all
```



Sandia National Laboratories