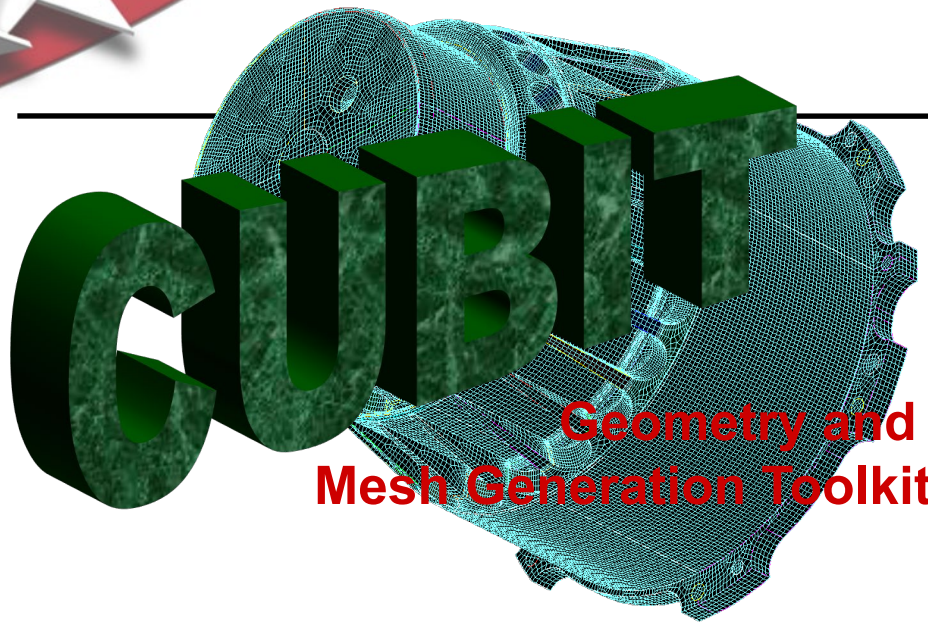


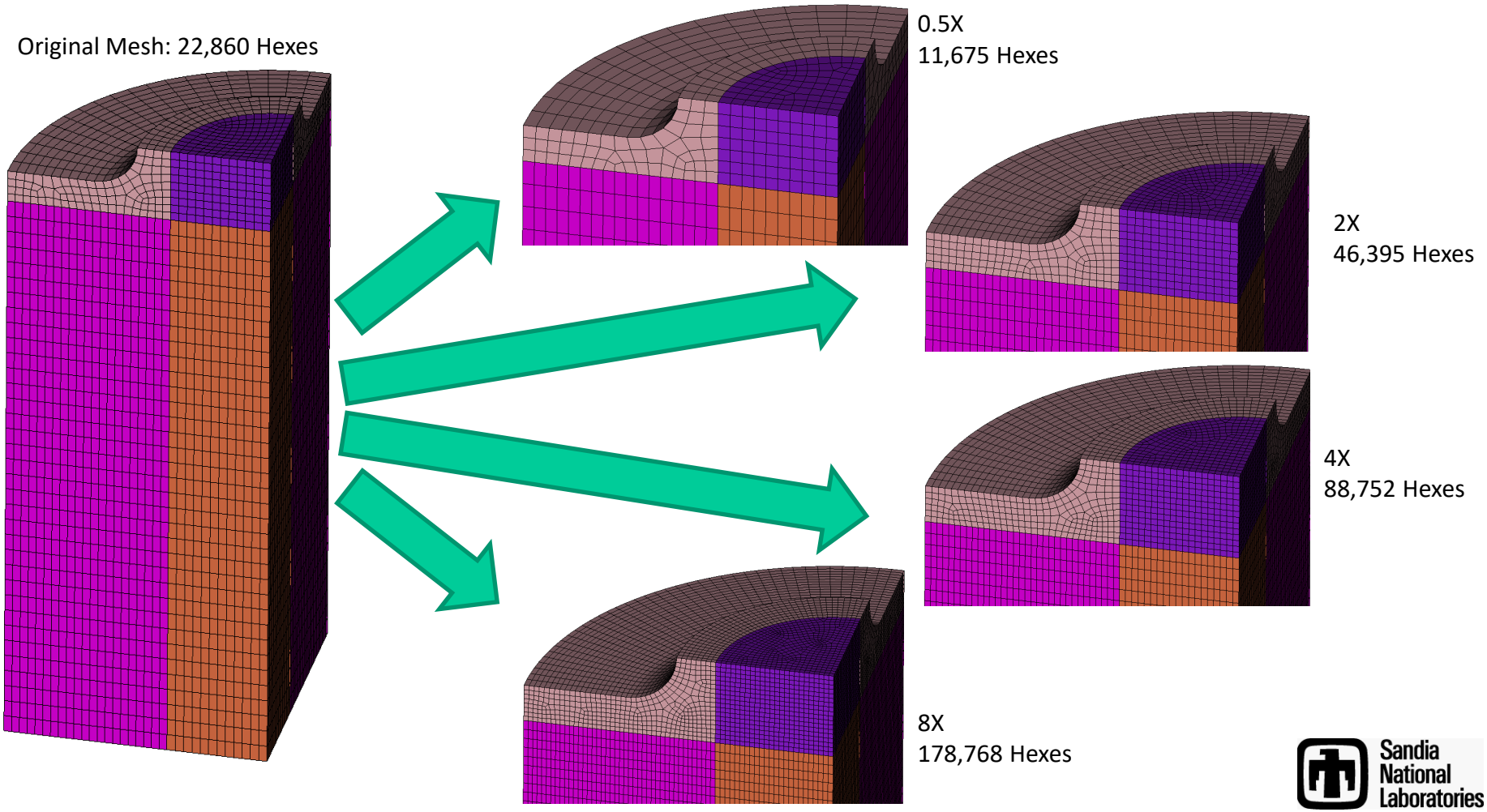
CUBIT Fast-Start Tutorial

24. Mesh Scaling

Mesh Scaling

Simulation Modeling Sciences

The goal of Mesh Scaling is, given an initial all-hexahedral mesh, produce a series of incrementally finer meshes with similar relative sizing attributes for use in solution verification and convergence studies.

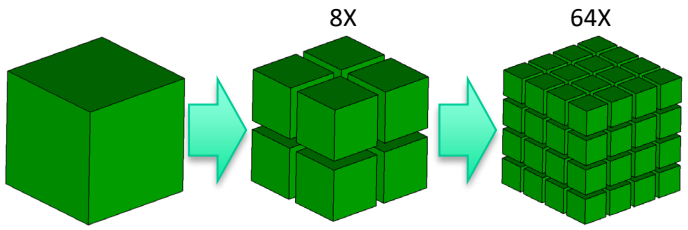


Sandia National Laboratories

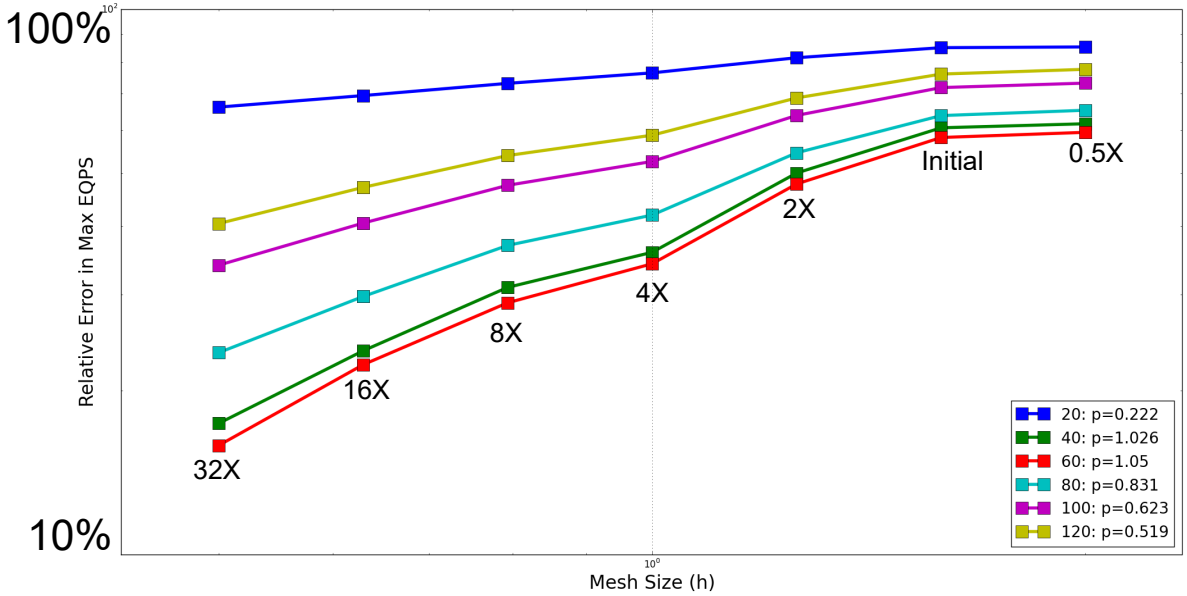
Mesh Scaling

Simulation Modeling Sciences

Traditional hexahedral refinement techniques require 8X multiplier for each mesh in the series and provide only refinement, and no coarsening. Computing resources are exhausted before many meshes in the series can be run.



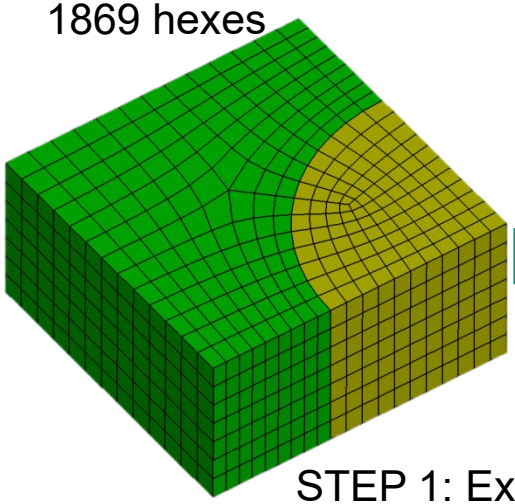
In contrast, Mesh Scaling allows the user to specify any element count multiplier. This allows more meshes to be built and computed with smaller element counts, providing more solution verification data. On the chart below, Mesh Scaling gave us a series of 7 meshes (0.5X, initial, 2X, 4X, 8X, 16X, 32X), while traditional refinement would have only provided 2 meshes (initial, & 8X).



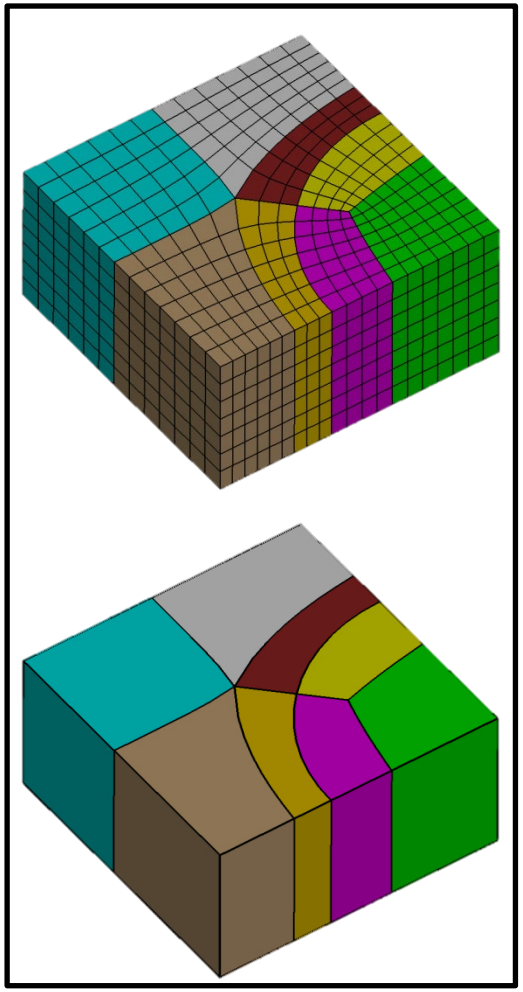
Sandia National Laboratories

Mesh Scaling Algorithm: maintain_structure

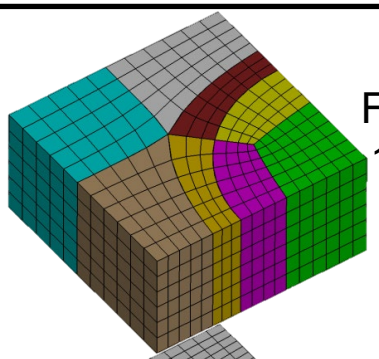
Initial Mesh
1869 hexes



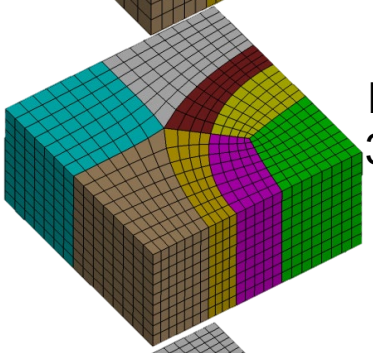
STEP 1: Extract the Block Decomposition



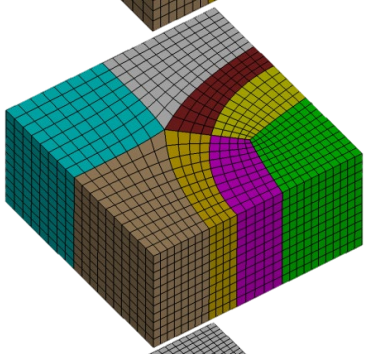
Step 2: Remesh each block at a different size



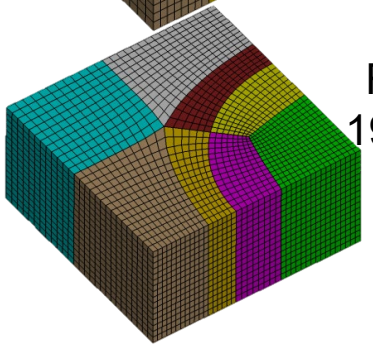
Factor 0.667
1230 hexes



Factor 1.75
3321 hexes



Factor 3.0
5841 hexes

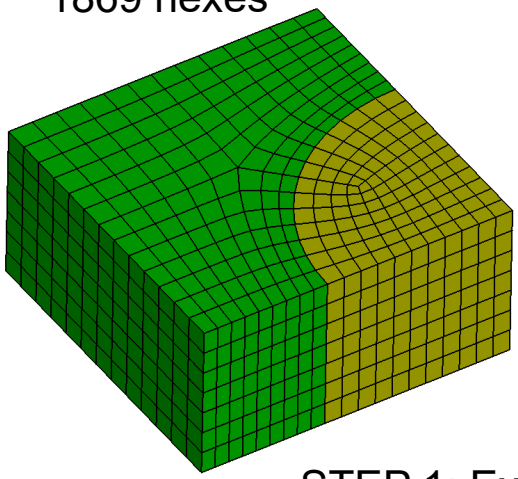


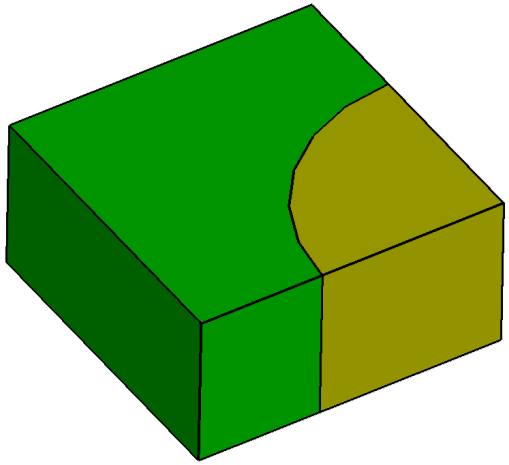
Factor 10.0
19,040 hexes

This works well if there is a lot of structure in the mesh. For more unstructured “pave-and-swept” meshes, we have extended mesh scaling to include Swept Blocks, as explained on the next slide.

Mesh Scaling Algorithm: swept_groups

Initial Mesh
1869 hexes

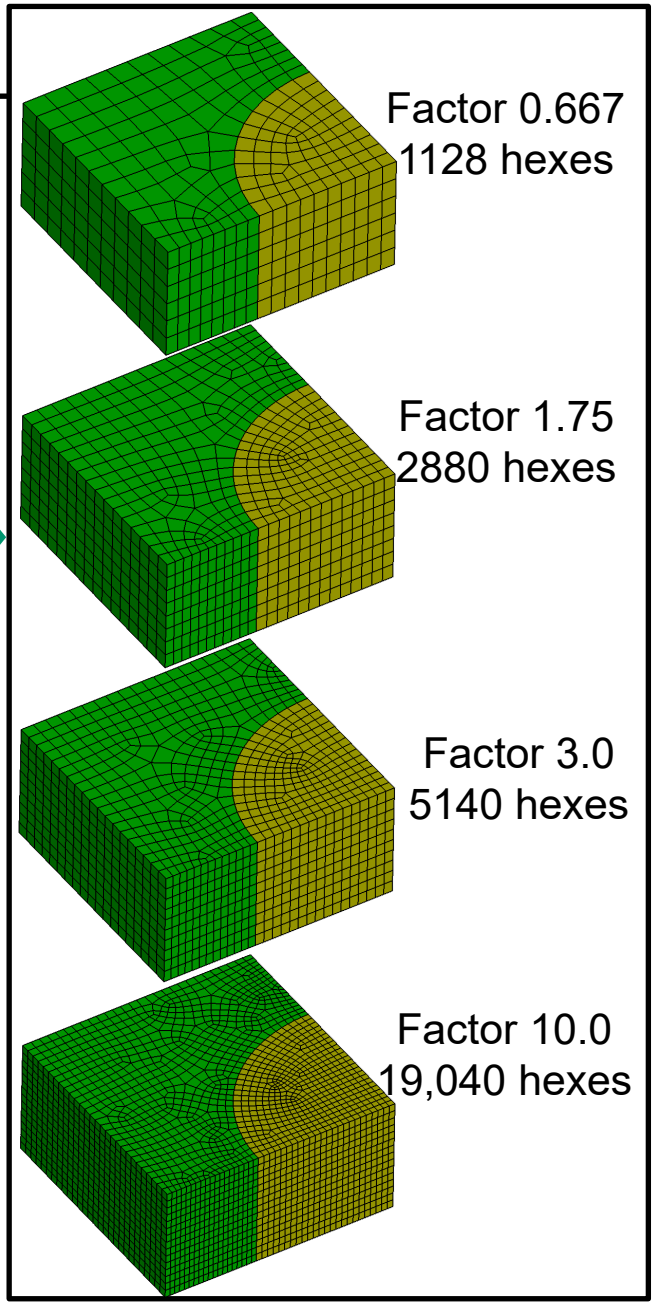




STEP 1: Extract the Block Decomposition

Produces a smooth mesh, but with a new set of singularities.

Step 2: Re-Pave-and-Sweep each block at a different size



Mesh Scaling
swept_blocks vs maintain_structure

maintain_structure

- Maintains number & type of mesh singularities
- Singularity positions may change slightly
- Element directionality maintained.
- Eliminates source of noise in solution verification
- Limited ability to coarsen
- Can produce skewed elements in thin regions (advanced options available for controlling this)

Swept_blocks (default)

- Produces high quality “pave-and-swept” meshes
- Change number & type of mesh singularities
- Directionality in mesh elements may change
- Modified singularities and element directions result in noise in solution verification
- Larger ability to coarsen

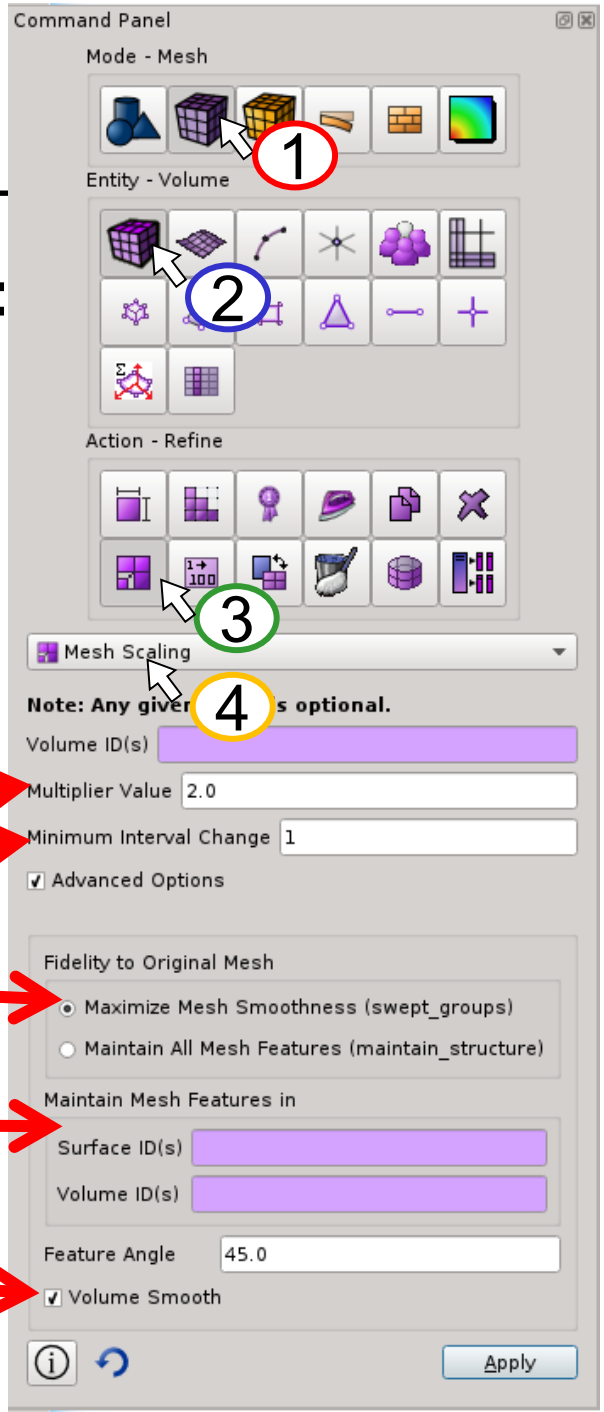
Mesh Scaling GUI

- **Mesh Scaling is under the GUI panel at:**
  - Mode: Mesh
  - Entity: Volume
  - Action: Refine
  - Mesh Scaling
- **Or you can issue the command:**

scale mesh [multiplier <double>]
[minimum <int>]
[{SWEPT_BLOCKS|maintain_structure}]

[force_structured in {[volume <ids>] [surface <ids>]}]

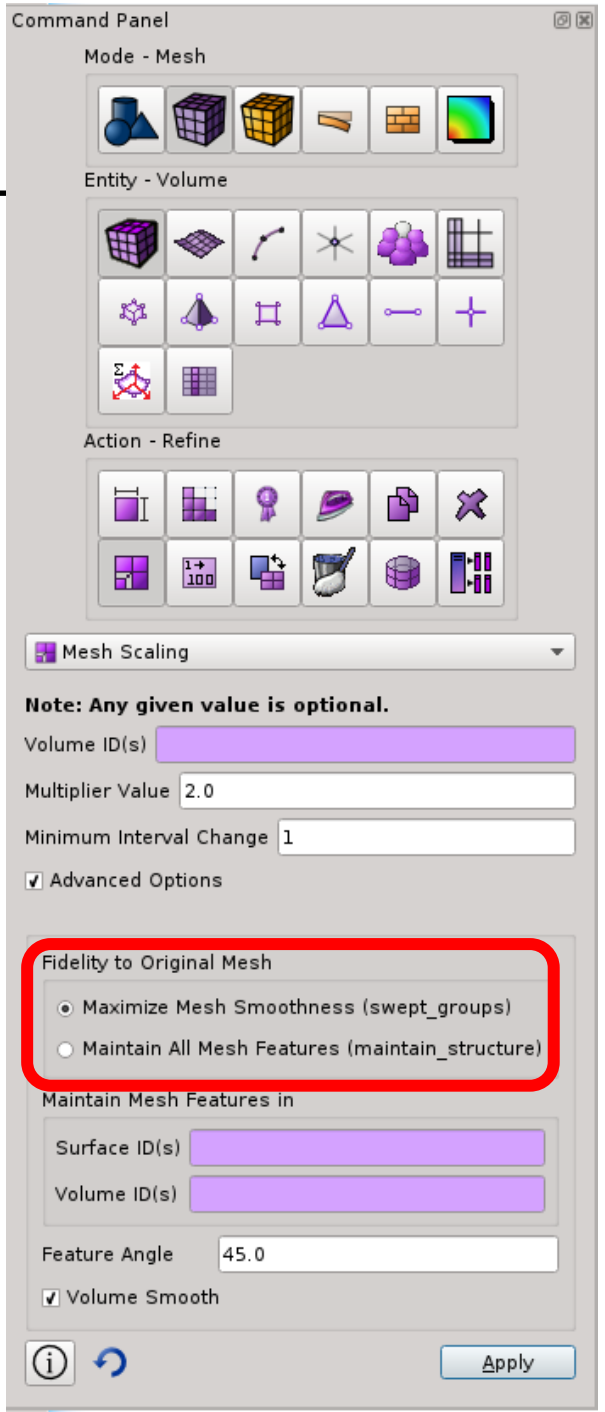
[smooth_volume {on|off}]



Mesh Scaling GUI

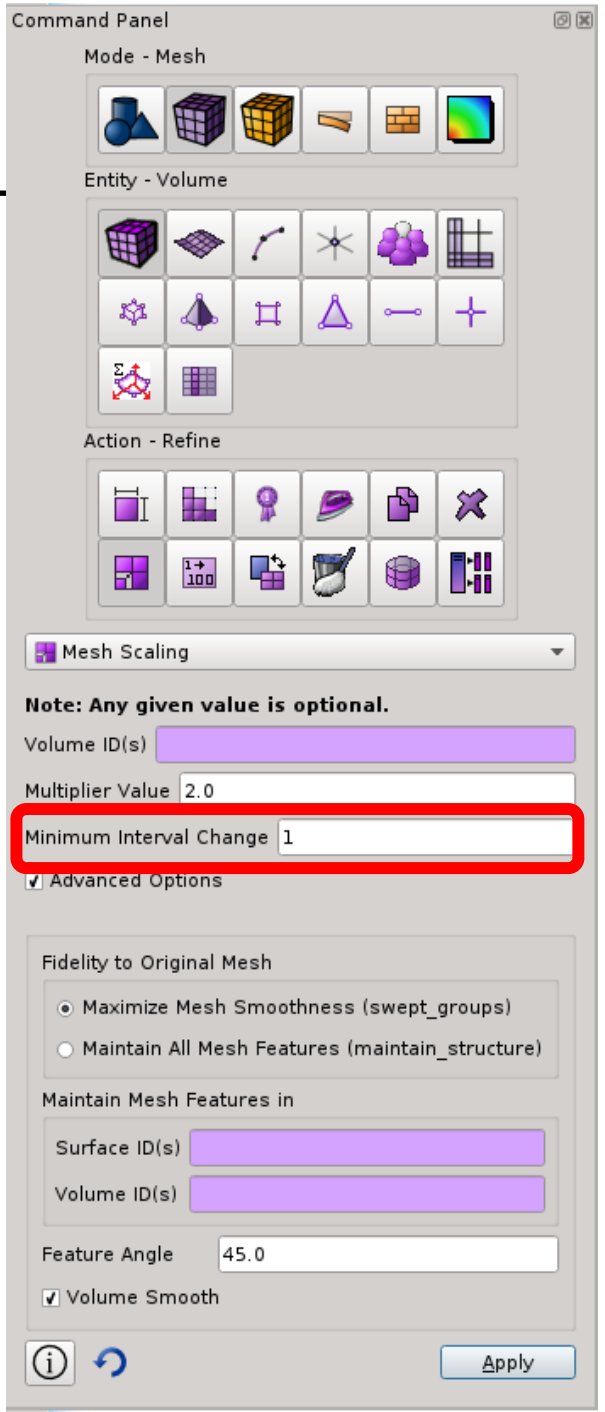
- The “Maintain All Mesh Features” scales the mesh with the maintain_structure method.

- The “Maximize Mesh Smoothness” option scales the mesh with swept blocks in swept regions, and structured blocks in structured regions.



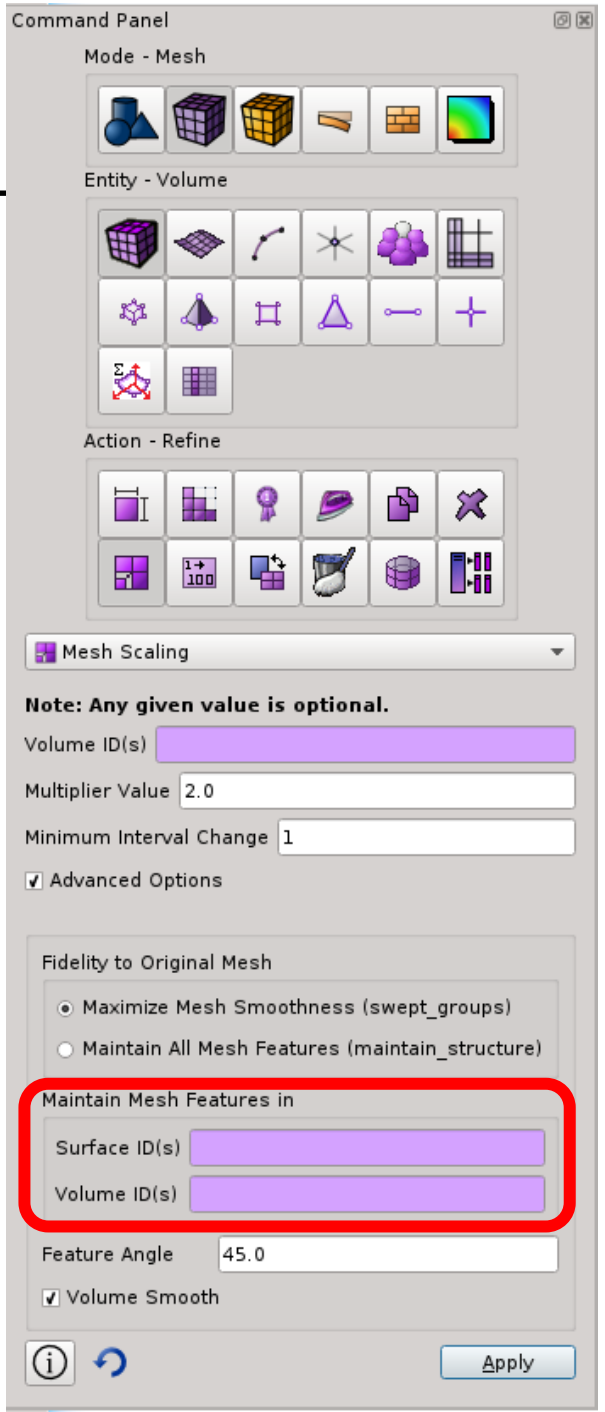
Mesh Scaling GUI

- **The "Minimum Interval Change" parameter specifies the minimum number of intervals that will be added to each curve of a block in the block decomposition.**
  - Specifying a value >0 attempts to ensure that the mesh changes a little bit, everywhere, although not guaranteed.
  - Specifying values >0 can cause mesh scaling to overshoot the specified multiplier, and produce more elements than requested.



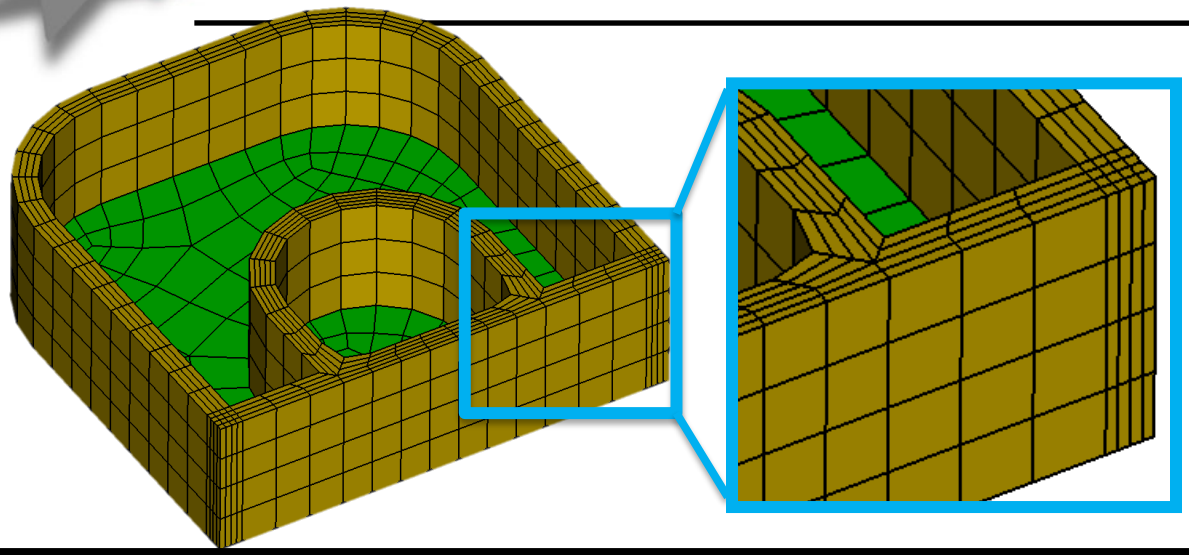
Mesh Scaling GUI

- The “Maintain Mesh Features in” option allows you to force structured blocks in some regions, allowing swept blocks everywhere else. This is a useful alternative to maintain_structure if the user has carefully crafted a structured zone of the mesh and wants it maintained through mesh scaling, but doesn’t care if mesh structure changes elsewhere.



Mesh Scaling with swept_blocks can sometimes destroy structure in mesh

Simulation Modeling Sciences

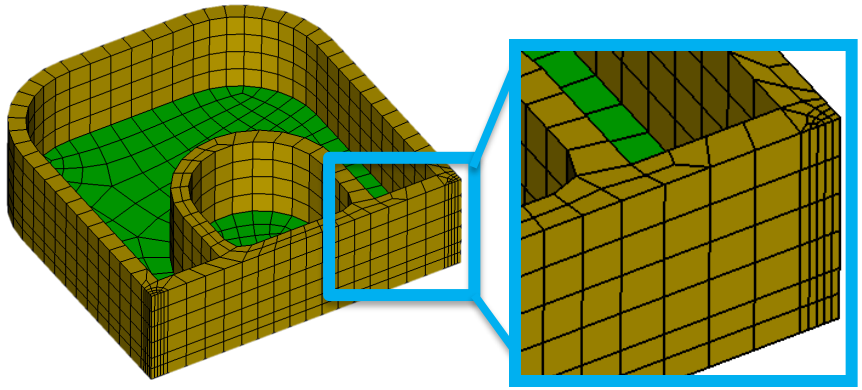


Initial Mesh: user has taken great care to build a structured mesh on rim.

The yellow elements are identified as a single swept block. There are also 2 green swept blocks

Cubit Command:

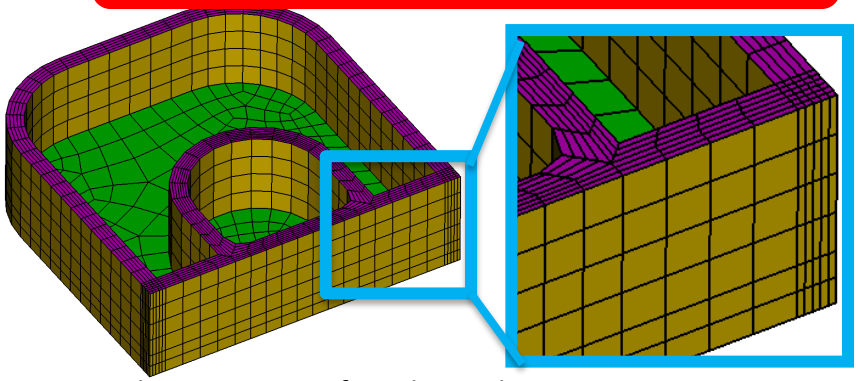
```
Scale mesh multi 2.0 swept_blocks
```



The source of each swept block is remeshed with the paver. The paver does not duplicate high aspect ratio structured meshes.

Cubit Command:

```
Scale mesh multi 2.0 swept_blocks
force_structured in surface 264
```



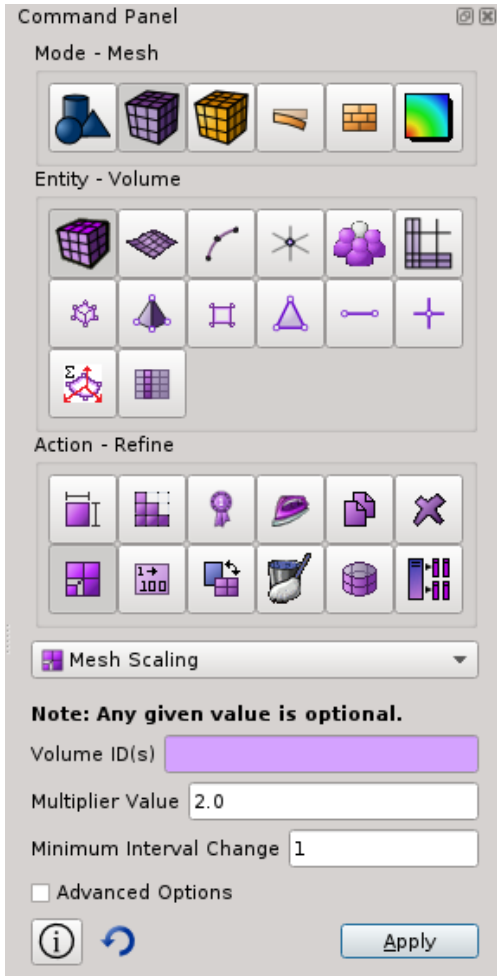
Surface 264 is the magenta surface that is the source of the sweep. By specifying to force it structured, that swept block is treated as a structured block, maintaining the structure.

Sandia National Laboratories

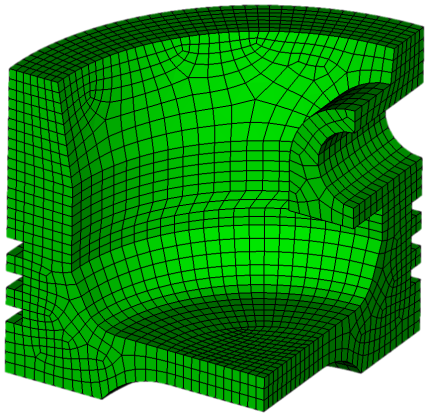
Mesh Scaling Exercise 1

Simulation Modeling Sciences

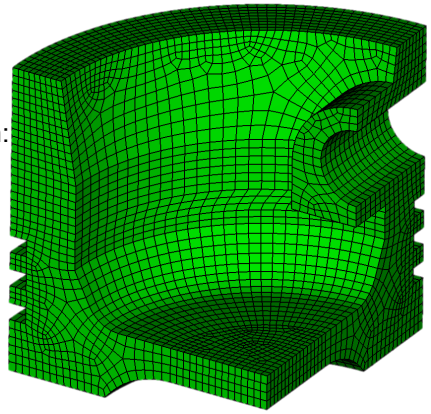
1. Open the cub file “MeshScaling.cub”
2. Open the Mesh Scaling command panel
3. Hit “Apply” to scale the mesh using the default multiplier of 2.0. Type “list totals” at the command line, and notice that the number of elements in the model roughly doubles.
4. Change the multiplier to 0.5 and hit apply again. Type “list totals” and notice that the number of elements drops back to roughly what it was originally.
5. Experiment using different multipliers to scale the mesh to different mesh resolutions.
6. Open the Advanced Options, and scale again with maintain_structure to observe the difference



Original Mesh:
6753 hex elements

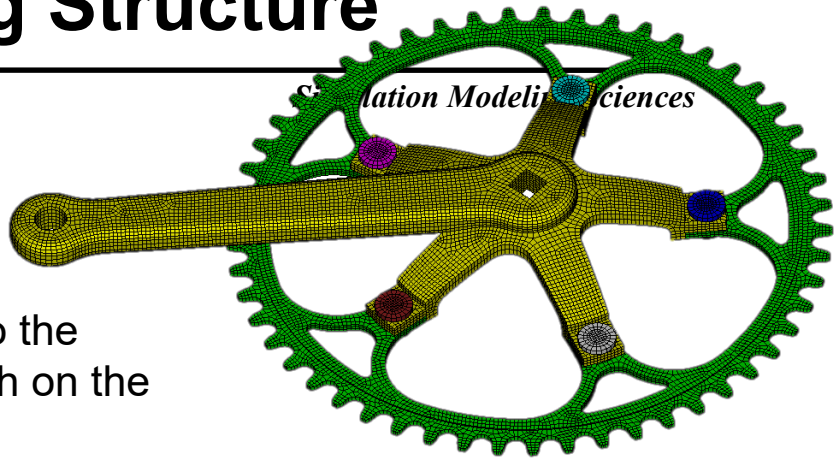


2X scaled mesh:
13,606 hex elements

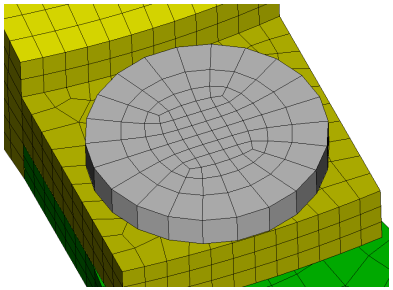


Sandia National Laboratories

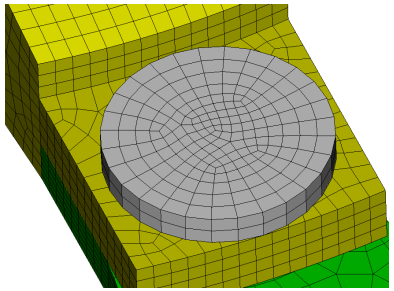
Mesh Scaling Exercise 2
Selective Maintaining Structure



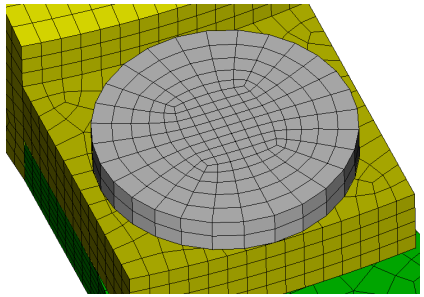
1. Open the cub file “MeshScalingAssembly.cub”
2. Open the Mesh Scaling command panel
3. Zoom into one of the bolts which holds the crank to the gear. Notice the nice circle pattern structured mesh on the bolt.



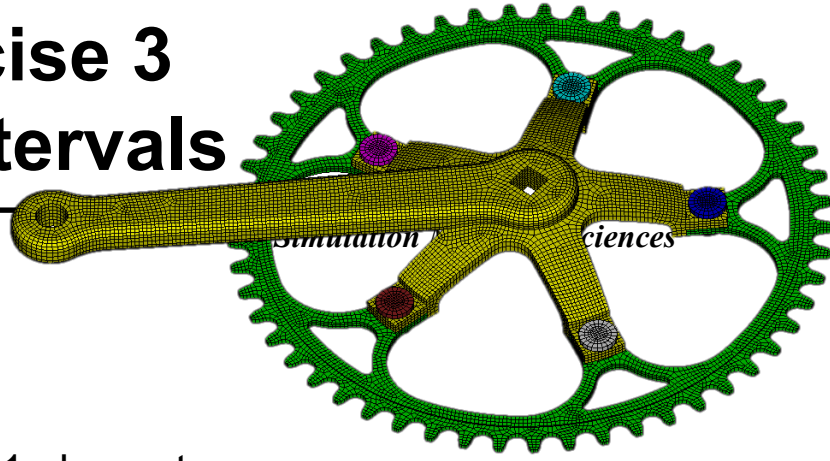
4. Scale the mesh with a multiplier of 2.0, using all the default settings. You will get a mesh that looks something like the picture on the right. Notice that the structured mesh is replaced with a paved mesh



5. Reset, then open cub file “MeshScalingAssembly.cub” a 2nd time.
6. Go back to the Mesh Scaling command panel, and scale the mesh again but specify the bolt volume ID in the Maintain Mesh Features in Volume ID(s) field. Notice that the structured circle pattern mesh is preserved.



Mesh Scaling Exercise 3
Minimum Through Intervals



1. Open the cub file “MeshScalingAssembly.cub”

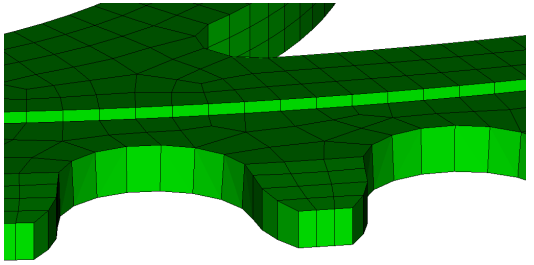
2. Open the Mesh Scaling command panel

3. Zoom into the side of the gear. Notice there is only 1 element through the thickness of the gear and the step above the teeth on the gear.

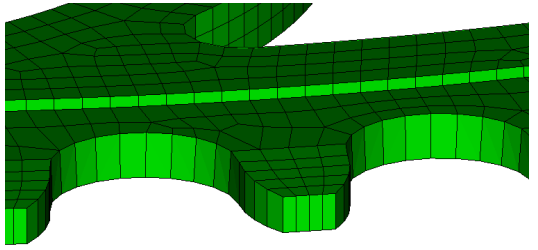
4. Scale the mesh with a multiplier of 2.0, using all the default settings except change the “Minimum Interval Change” to 0. Inspect the through intervals on the teeth, no new intervals have been added. At small multipliers (like 2X), sufficient elements get added before intervals are added through this thickness.

5. Reset Cubit, then open cub file “MeshScalingAssembly.cub” a 2nd time. Scale the model again but using a “Minimum Interval Change” of 1. Zoom back into the teeth and inspect the number of intervals in the scaled meshes. While the extra intervals are added, this causes the number of elements to exceed the desired multiplier.

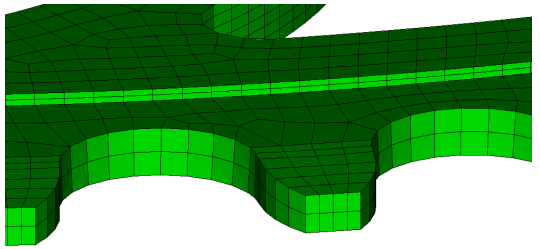
Initial Mesh: 32,073 hexes



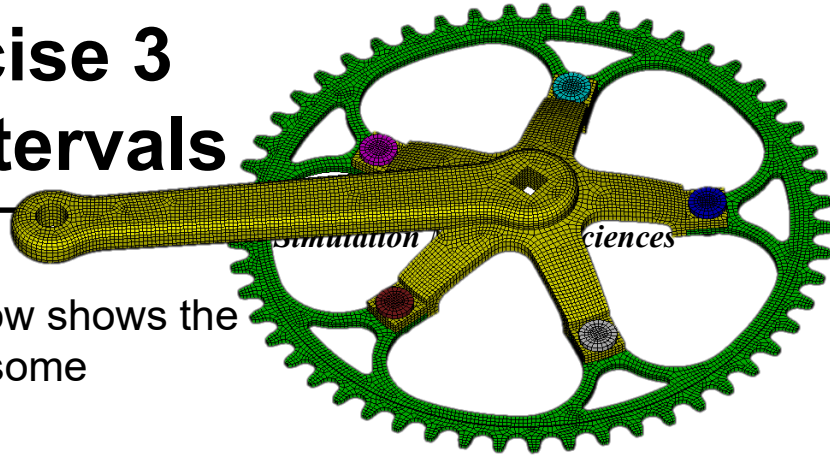
2X minimum 0 results in 63,280 hexes



2X minimum 1 results in 81,818 hexes



Mesh Scaling Exercise 3
Minimum Through Intervals



On the MeshScalingAssembly model, the chart below shows the number of elements produced by mesh scaling for some (multiplier, minimum interval change) combinations.

Initial Mesh has 32,073 hexes

Requested Multiplier	Minimum Interval Change 0		Minimum Interval Change 1		Minimum Interval Change 2	
	#elems	Effective Multiplier	#elems	Effective Multiplier	#elems	Effective Multiplier
2X	63,280	1.97	81,818	2.56	109,620	3.42
4X	140,777	4.39	140,777	4.39	179,212	5.59
8x	252,202	7.86	252,202	7.86	297904	9.29

Specifying a “Minimum Interval Change” can cause mesh scaling to overshoot the multiplier, sometimes significantly.

Sierra Mesh Scale

In addition to the “scale mesh” command in Cubit, Mesh Scaling is also available as a batch Sierra program, released as part of Sierra 4.38. The basic usage is:

```
mesh_scale --inmesh=mymesh.exo
           --outmesh=mymesh.out.exo
           --multiplier=3.5
           --intmin=1
           --sweptblocks
```

(sweptblocks not ON by default until Sierra 4.40)

Using this form of the mesh_scale command, the new nodes will be projected to the skin of the original mesh, since no CAD geometry for the model is provided.

The Sierra version of mesh scaling:

1. Uses the same mesh scaling library as the Cubit version.
2. Is more memory efficient than the Cubit version of mesh scaling, allowing for significantly larger meshes to be generated than the Cubit version.
3. Will become the parallel version of mesh scaling to produce meshes for high performance computing.

Sierra Mesh Scale With CAD Geometry

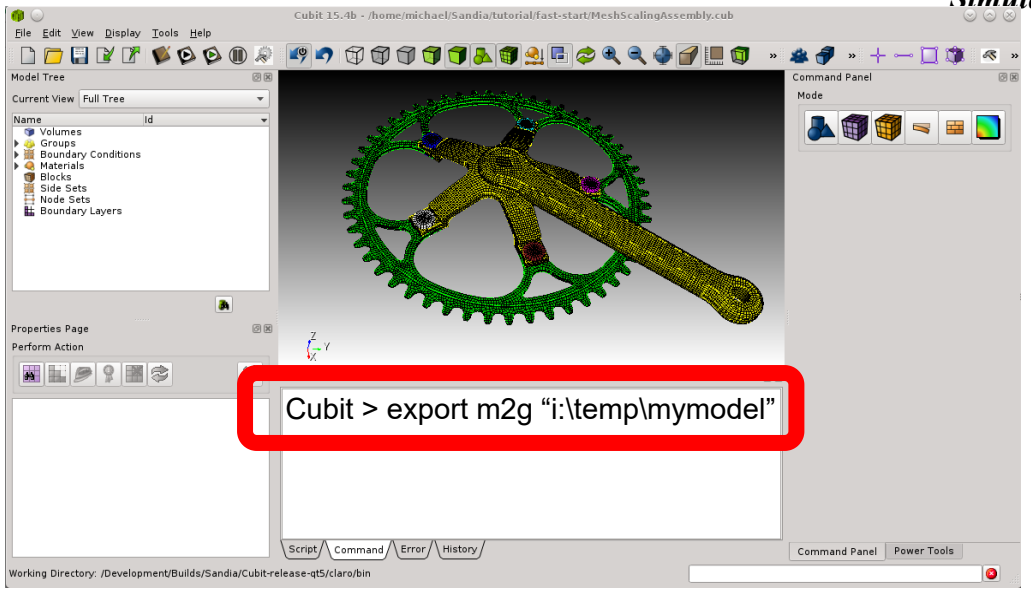
Optionally, the original ACIS CAD geometry from Cubit can be passed into Sierra mesh_scale so new nodes are projected to the original CAD.

```
mesh_scale --inmesh=mymesh.g
           --outmesh=mymesh.out.g
           --multiplier=3.5
           --intmin=1
           --sweptblocks
           --geom=mycad.sat
           --geomassoc=mycad.m2g
```

The sat file is an ACIS file exported from Cubit.
The m2g file contains associativity information between the mesh in “mymesh.exo” and “mycad.sat”

To use this feature of mesh_scale, you need to use the “export m2g” command in Cubit to prepare the exo, sat, and m2g files.

Generating the geometry input files for Sierra mesh_scale

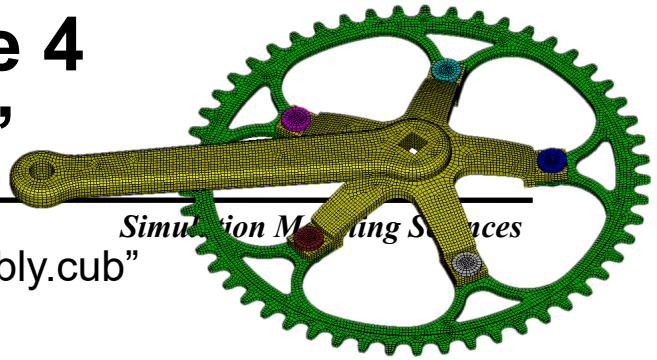


With the initial mesh in Cubit 15.1 or later, the command: `export m2g <fileroot>` will export 3 files for input to Sierra mesh_scale:

- `fileroot.g` : The exo file containing the mesh to scale
- `fileroot.sat` : The geometry to project new nodes to
- `fileroot.m2g` : Associativity information, so Sierra mesh scale knows what surface or curve to project to when scaling.

NOTE: Before issuing the `export m2g` command, you may need to issue "set geometry version 2400" so that the ACIS file exported is a version that can be imported into Sierra mesh_scale. Sierra mesh_scale uses ACIS 2400, but Cubit 15.1 uses ACIS 2520.

Mesh Scaling Exercise 4
Sierra “mesh_scale”



Simulation Modeling Sciences

1. In Cubit 15.1 or later, open the cub file “MeshScalingAssembly.cub”
2. From the Cubit command prompt type:

```
set geometry version 2400
export m2g “c:\some\path\mymodel”
```

3. Close Cubit.
4. Copy the files mymodel.sat, mymodel.exo, and mymodel.m2g to an environment which contains a Sierra 4.38 or later version of mesh_scale.

NOTE: If you have a Sandia CEE blade account, do the following:

1. Mount your CEE home directory to the classroom PC as “`\\cee\<username>`” and copy files over.
2. Connect to your blade from the classroom PC with RGS, NX, or Leostream
3. In a command prompt : `module load sierra`

5. Launch Sierra mesh_scale from command prompt as follows:

```
mesh_scale --inmesh=mymodel.g --outmesh=mymodel.out.g
           --multiplier=2.0 --intmin=1
           --geom=mymodel.sat --geomassoc=mymodel.m2g
           --sweptblocks
```

6. Import the output file “mymodel.out.g” into Cubit for inspection.

Sandia National Laboratories