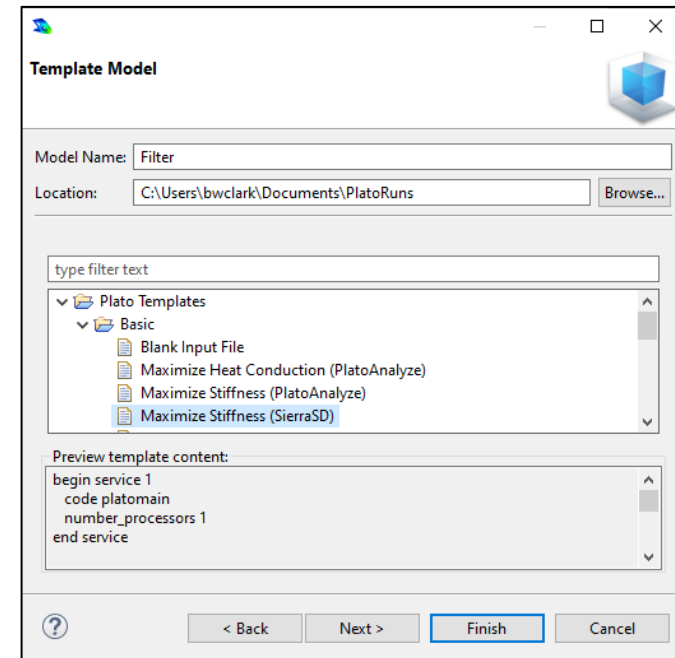
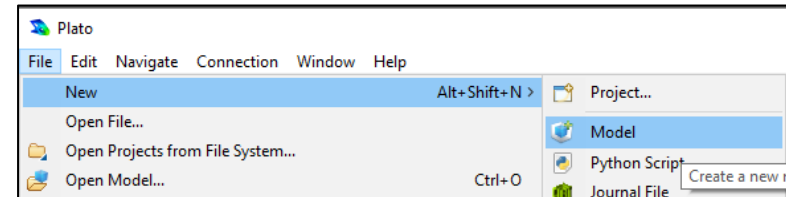


Filter Tutorial

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Create a New Model

- Choose **File->New->Model** in the menu
- Choose **New Model** then **Next**
- Choose **Create From Template** then **Next**
- Enter **Filter** as the **Model Name**
- Choose the **Plato Templates->Basic->Maximize Stiffness (SierraSD)** template and then **Finish**



Create the Design Domain

- At the **CUBIT** command prompt type the following commands (see Hint on next slide):

brick x 10 y 3 z 0.1

brick x 1 y 1 z 0.1

volume 2 move 5.5 0 0

imprint volume all

delete volume 2

- Note the surface created from the imprint (highlighted in orange).

```

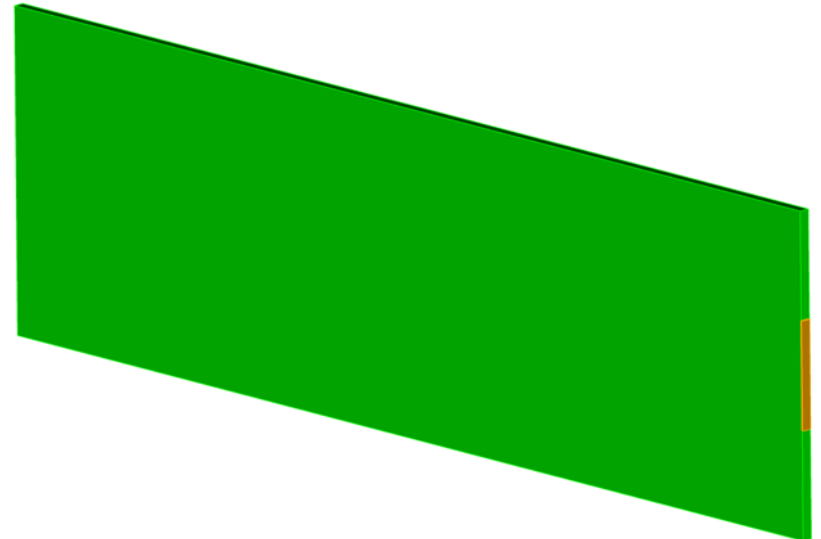
Console  Machines
CUBIT Console

CUBIT> brick x 10 y 3 z .1
Successfully created brick volume 1
Journalled Command: brick x 10 y 3 z 0.1

CUBIT> brick x 1 y 1 z .1
Successfully created brick volume 2
Journalled Command: brick x 1 y 1 z 0.1

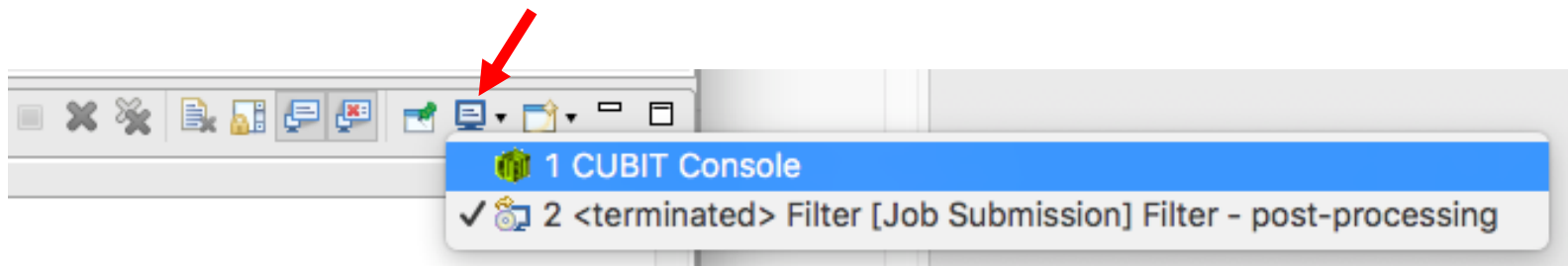
CUBIT> vol 2 move 5.5 0 0
Journalled Command: volume 2 move 5.5 0 0

CUBIT> imprint all
Preserving undo information done
  
```



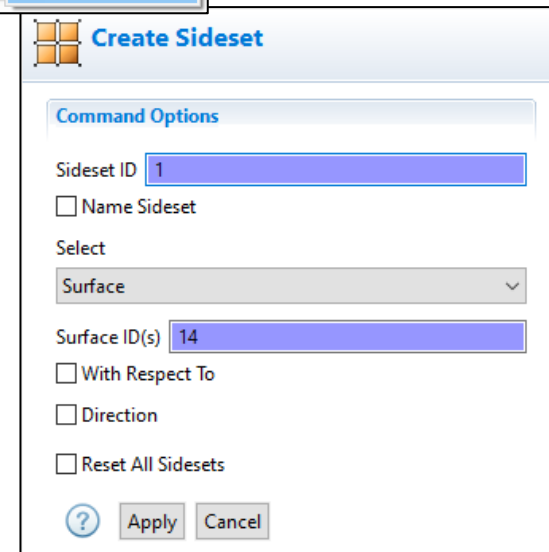
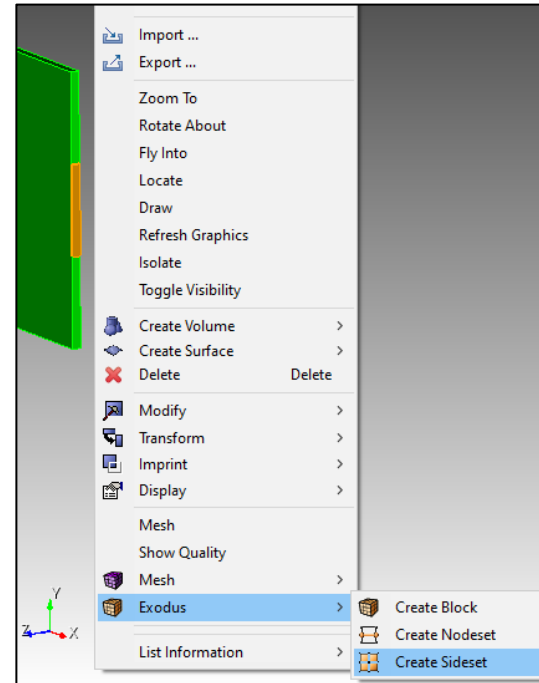
Hint: Cubit Console

- If you don't see the **CUBIT>** prompt in the console window you are just looking at the wrong console (there may be multiple to choose from loaded in Plato that use the single console window). Click on the icon at the top right of the console window that looks like a computer screen to toggle through the different consoles that are currently loaded. Or you can click on the down arrow next to the computer screen icon to see all of the currently loaded consoles and choose one from the list.



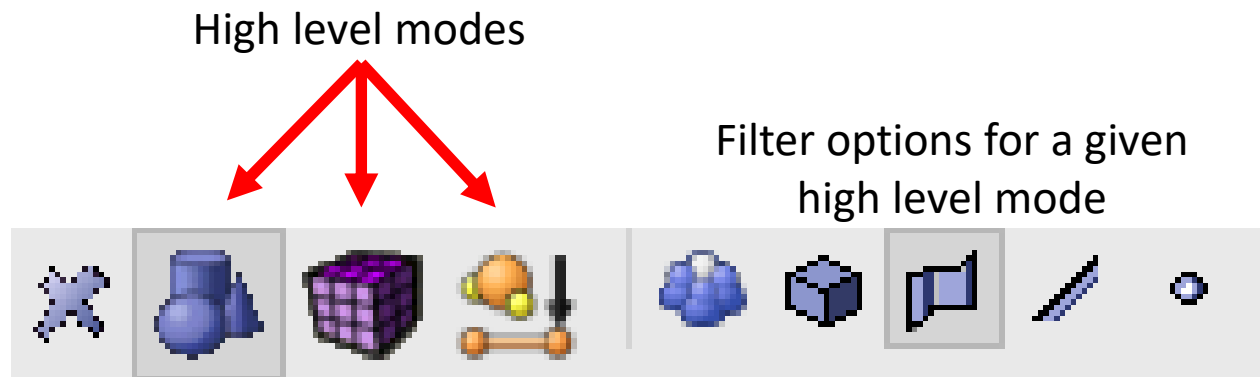
Define Load Application Location

- Select and right-click the middle right surface (see Hint on next slide) and choose **Exodus->Create Sideset**.
- In the Create Sideset panel click the “Apply” button to accept the default values and create the sideset. Note that the id of the new sidset is 1. We will use this later on when creating the actual load that is applied to this sideset.



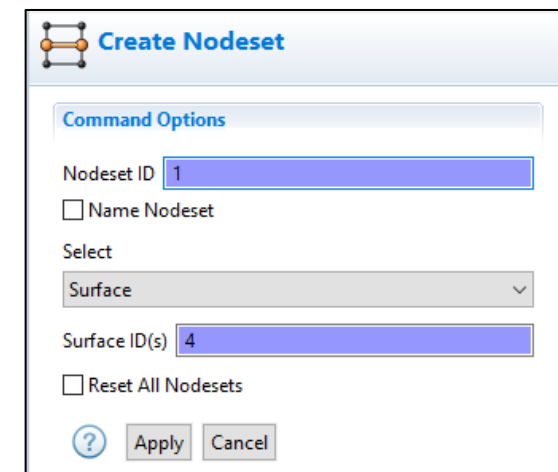
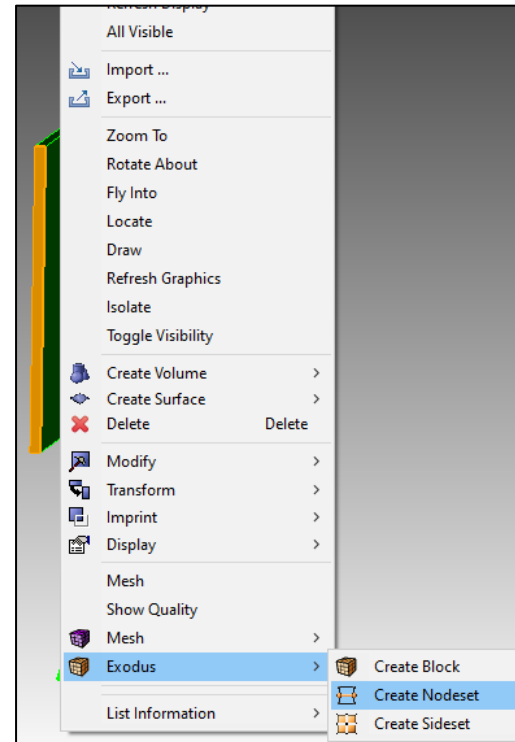
Hint: Entity Selection

- The selection filter toolbar is at the bottom left of the graphics window.
- There are 3 high level modes: Geometry, Mesh, and Exodus
- Within each high level mode there are various options for filtering what will get selected by clicks in the graphics window.



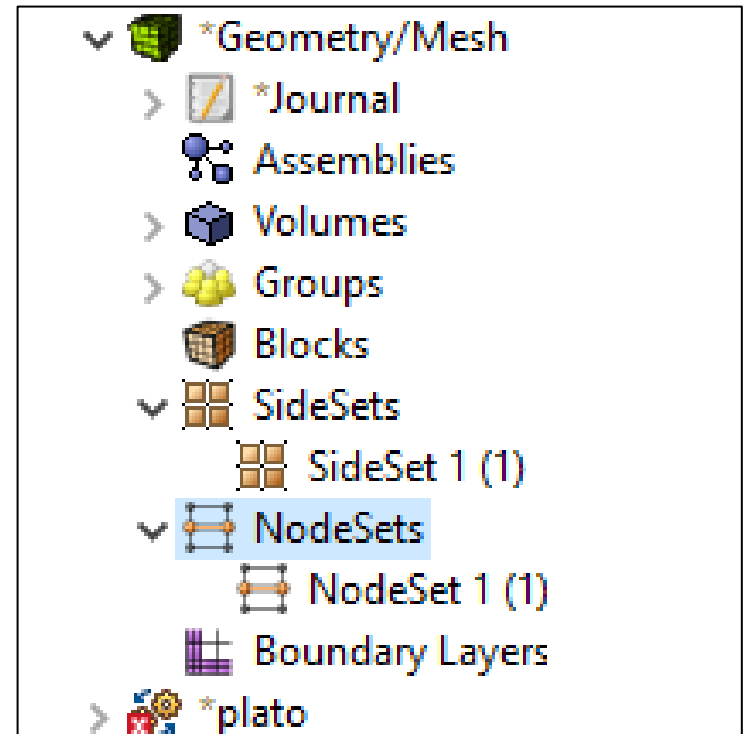
Define BC Application Location

- Select and right-click on the single left surface and choose **Exodus->Create Nodeset**.
- In the “Create Nodeset” panel click the “Apply” button to accept the defaults and create the nodeset. Note the nodeset id is 1. We will use this in creating the fixed boundary condition.

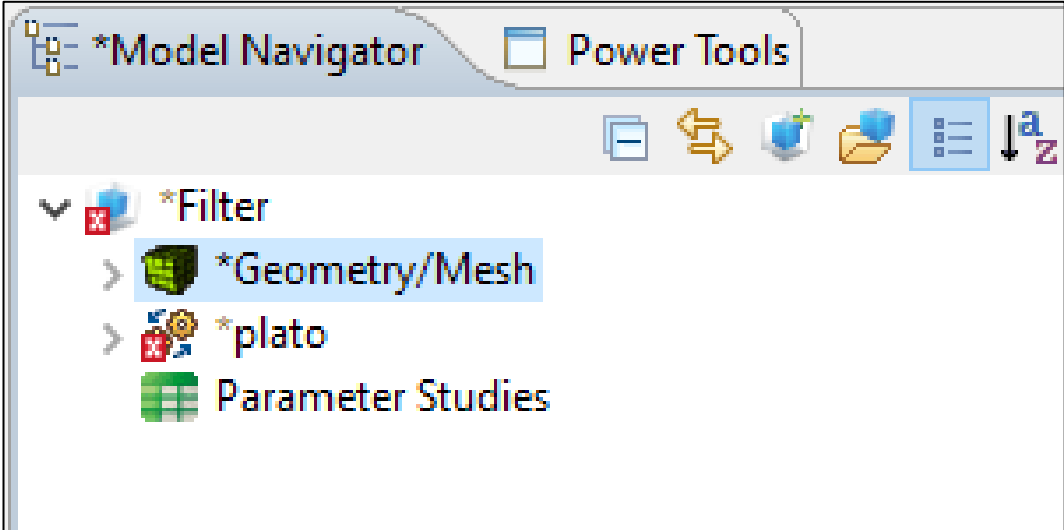


Sidesets and Nodeset in the Model Tree

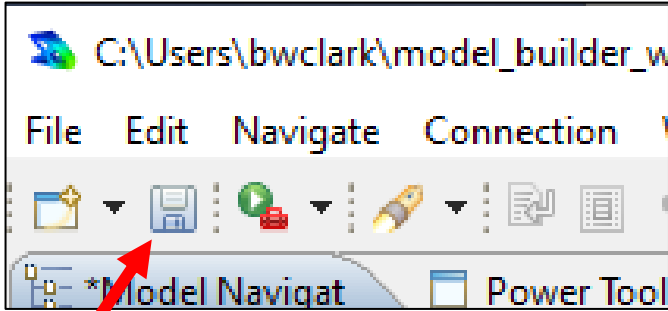
- The sideset and nodeset we just created can be seen in the Model tree under the “Geometry/Mesh” node.



Save the Geometry/Mesh

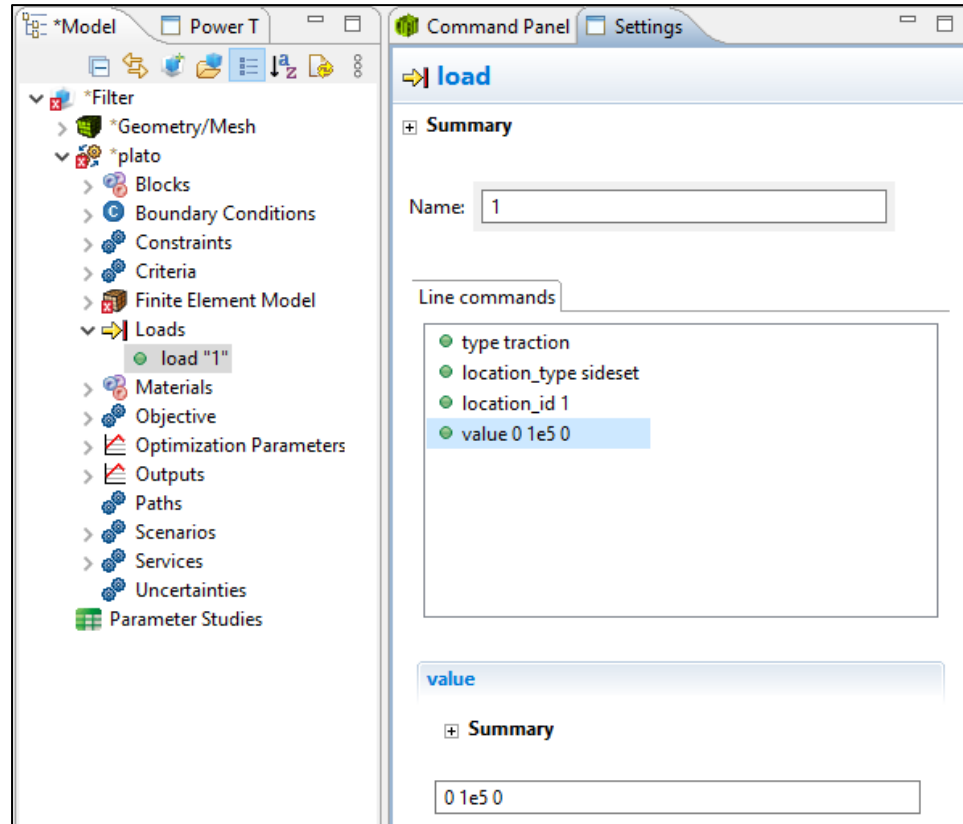


- Click on the “Geometry/Mesh” node in the tree and then click the Save icon in the toolbar.



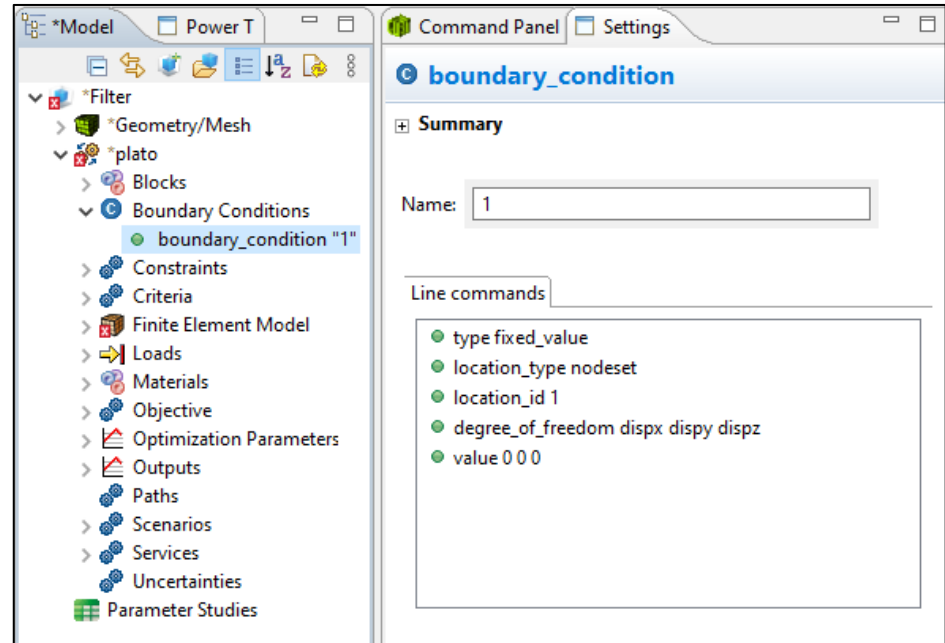
Create a Traction Load

- Expand the “plato” node in the Model tree and then expand the “Loads” node. In the template we used for this model there was already a load defined. Click on “load 1” in the tree to bring up its information in the Settings panel. The “location_id” is already set to “1” meaning it will be applied to sideset 1. Change the value to “0 1e5 0” indicating there will be a traction load with a positive y component of 1e5 applied to sideset 1.
- Note: If the template did NOT already have a load defined, we could have created one by clicking on the “Load” hyperlink that shows up in the Settings panel when you click on the “Loads” node in the tree.



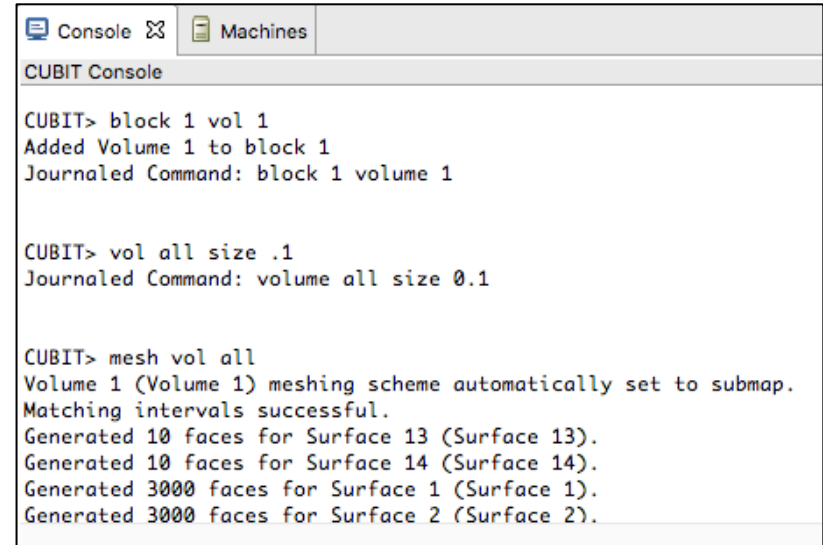
Create a Fixed BC

- Expand the “plato” node in the Model tree and then expand the “Boundary Conditions” node. In the template we used for this model there was already a boundary condition defined. Click on “boundary_condition 1” in the tree to bring up its information in the Settings panel. The “location_id” is already set to “1” meaning it will be applied to nodeset 1. The “degree_of_freedom” and “value” parameters specify that displacement in all 3 directions will be fixed so there is nothing to change.
- Note: If the template did NOT already have a boundary condition defined, we could have created one by clicking on the “Boundary condition” hyperlink that shows up in the Settings panel when you click on the “Boundary Conditions” node in the tree.



Define Block and Mesh Domain

- At the **CUBIT** prompt type the following commands:
 block 1 volume 1
 volume all size 0.1
 mesh volume all
- Click on **Geometry/Mesh** node in the tree and then click the **Save** icon in the toolbar to save the model

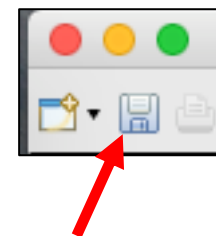
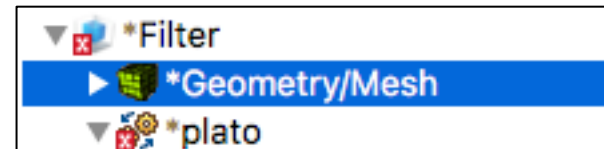


```

CUBIT Console
CUBIT> block 1 vol 1
Added Volume 1 to block 1
Journaled Command: block 1 volume 1

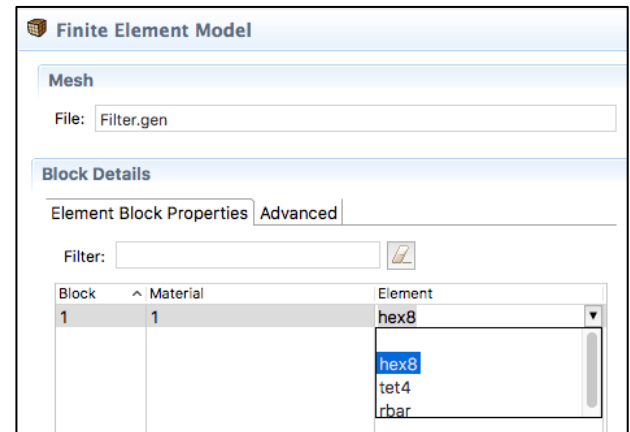
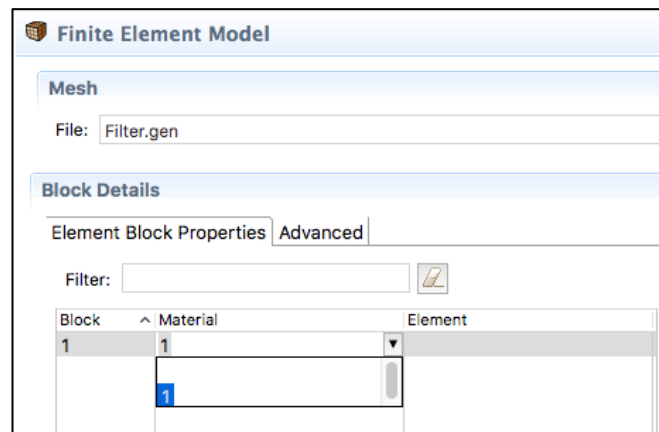
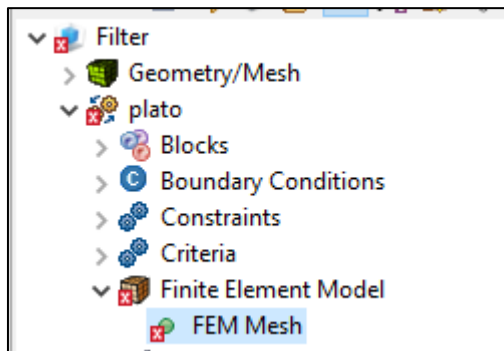
CUBIT> vol all size .1
Journaled Command: volume all size 0.1

CUBIT> mesh vol all
Volume 1 (Volume 1) meshing scheme automatically set to submap.
Matching intervals successful.
Generated 10 faces for Surface 13 (Surface 13).
Generated 10 faces for Surface 14 (Surface 14).
Generated 3000 faces for Surface 1 (Surface 1).
Generated 3000 faces for Surface 2 (Surface 2).
    
```



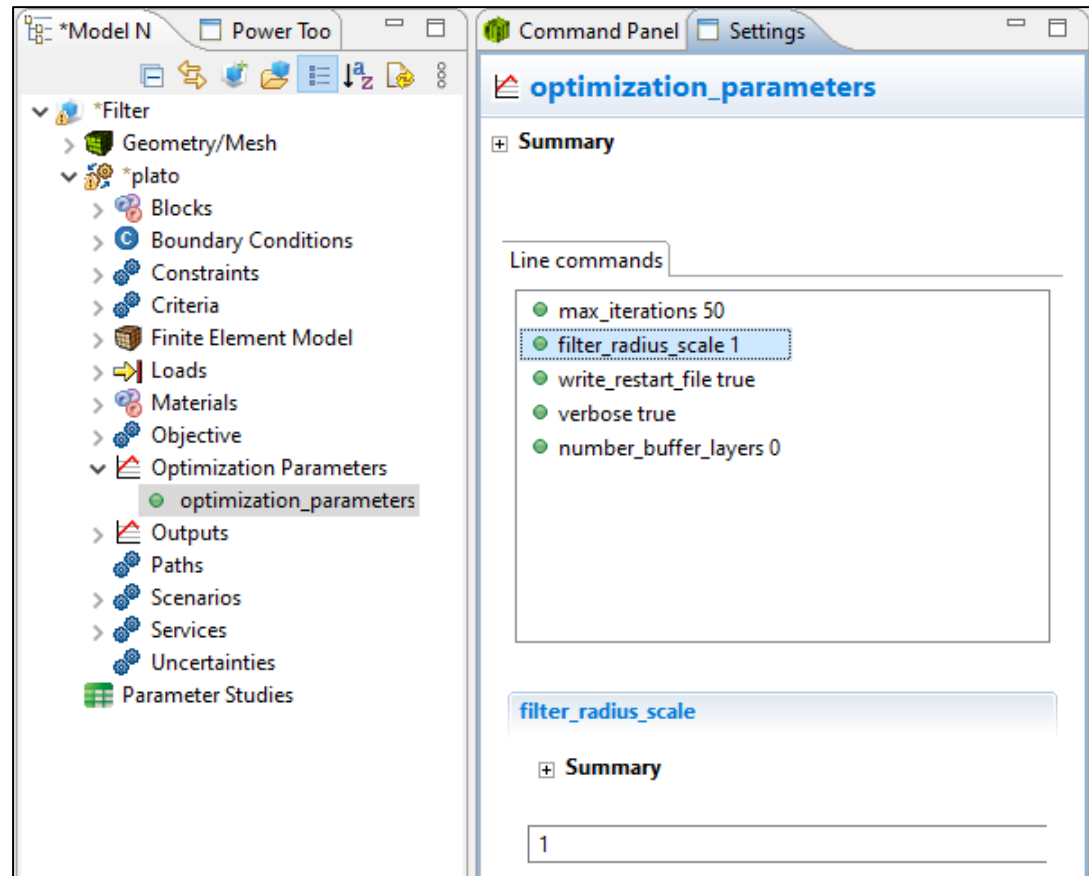
Assign Material and Element Type

- Click on the **FEM Mesh** node in the tree under **plato->Finite Element Model**
- In the **Settings** panel click in the **Material** area next to Block 1 and select **1** from the dropdown list
- Then click in the **Element** area next to Block 1 and select **hex8** from the dropdown list



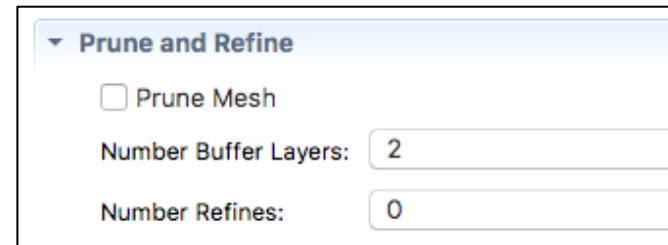
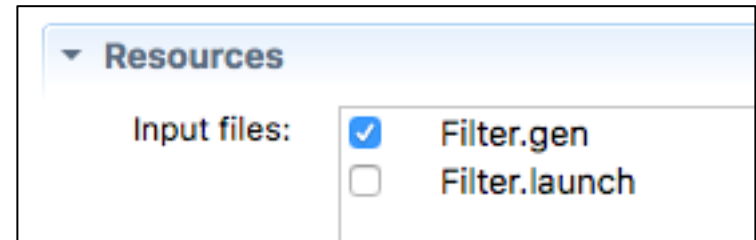
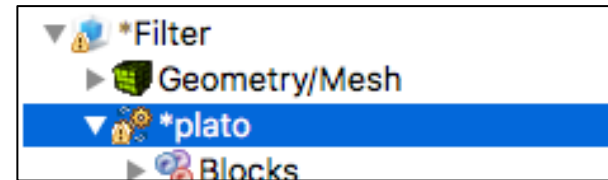
Set the Filter Scale

- Click on the **Optimization Parameters**-> **optimization parameters** node in the tree
- In the **Settings** panel click on **filter_radius_scale** and type in **1** for the value and hit the **return** key. **Filter_radius_scale** specifies the scale factor that will be multiplied by the average mesh edge length to determine the filter radius.



Run the Optimization

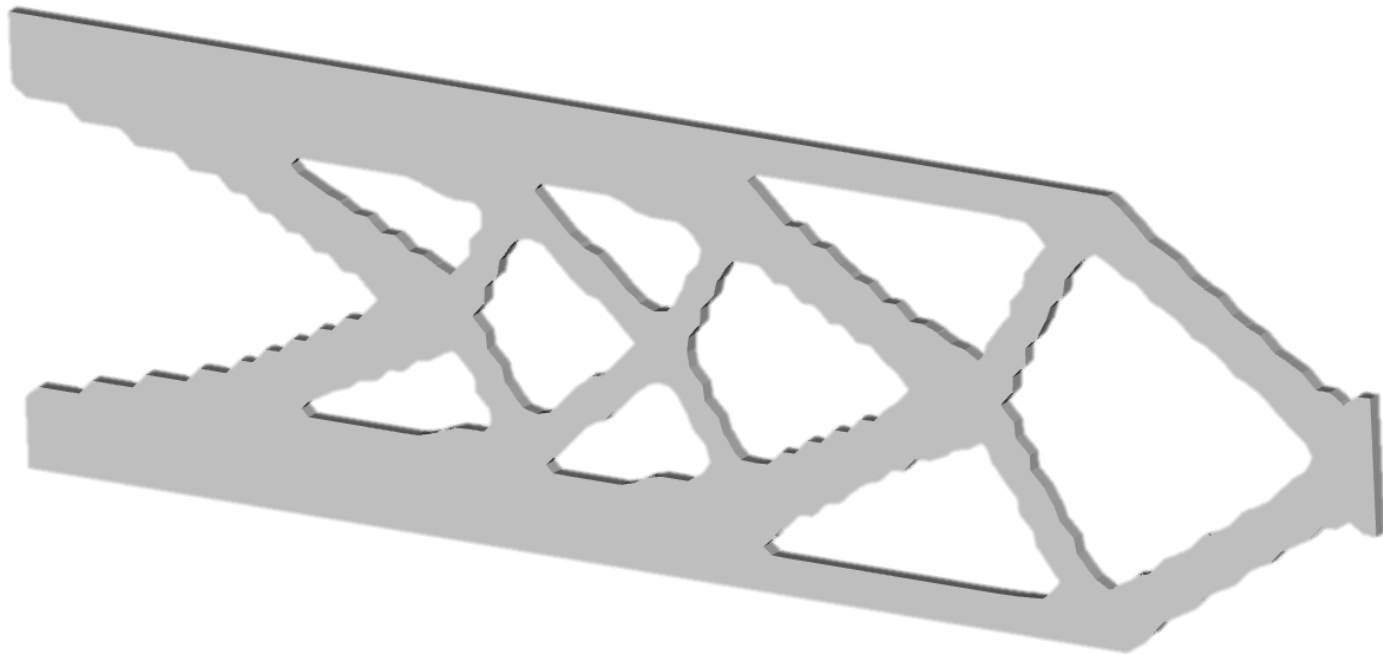
- Click on the **plato** node in the tree to bring up the job submission panel in the **Settings** view
- Choose **Plato** as the code and then choose the machine and execution template you want to use



- In the **Resources** area make sure **Filter.gen** is checked
- In the **Prune and Refine** area make sure **Prune Mesh** is unchecked and **Number Refines** is **0**. We will not use these features in this example.
- Choose any other preferences and launch the job by clicking on **Submit Simulation Job** toward the top of the panel

Results for filter_radius_scale=1.0

- With a **filter_radius_scale** value of 1.0 the filter smoothing at each node will not take into account any neighbors and will have little to no effect on the result. Notice the rough edges.

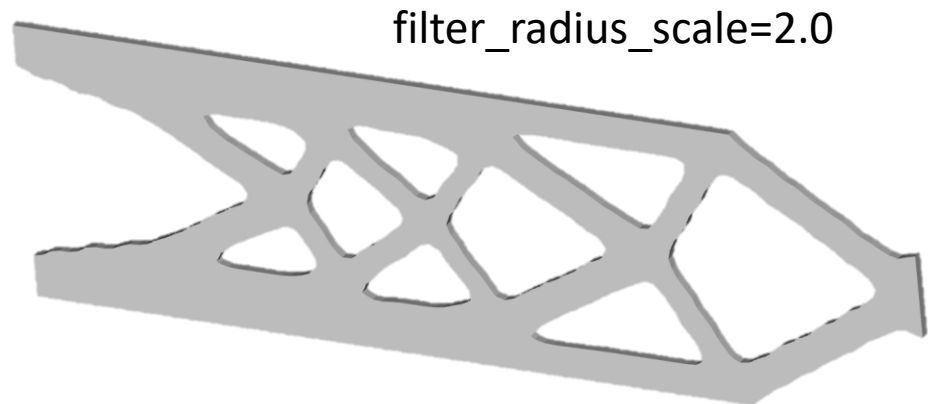
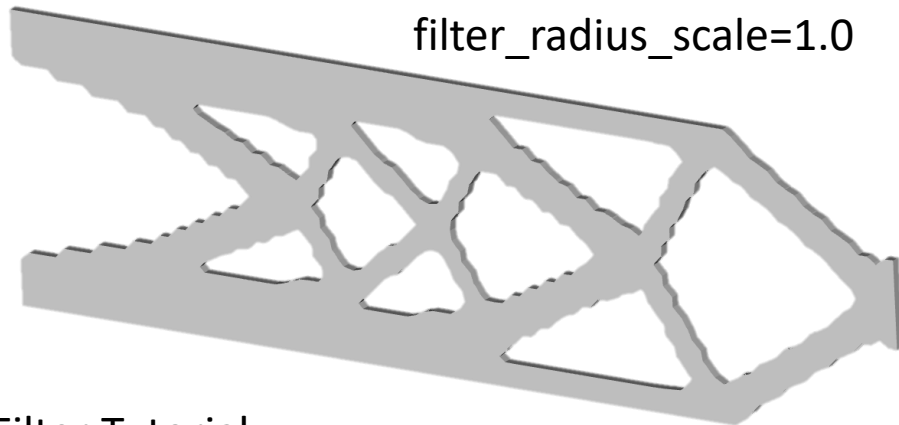


Rerun with `filter_radius_scale=2.0`

- Change the **filter_radius_scale** value to **2.0**
- Resubmit the job with all of the same parameters by clicking on the **plato** node in the tree and then click on the **Submit Simulation Job** in the job submission area in the **Settings** panel

Results for filter_radius_scale=2.0

- Notice how as we increase the **filter_radius_scale** value the results become smoother. In general, this is one of the effects of using a filter. A **filter_radius_scale** of **2.0** means that the filter radius of influence is 2.0 times the average mesh edge length. With the radius larger than one element width the filter actually starts having influence on neighbors and starts smoothing out local discontinuities or roughness.

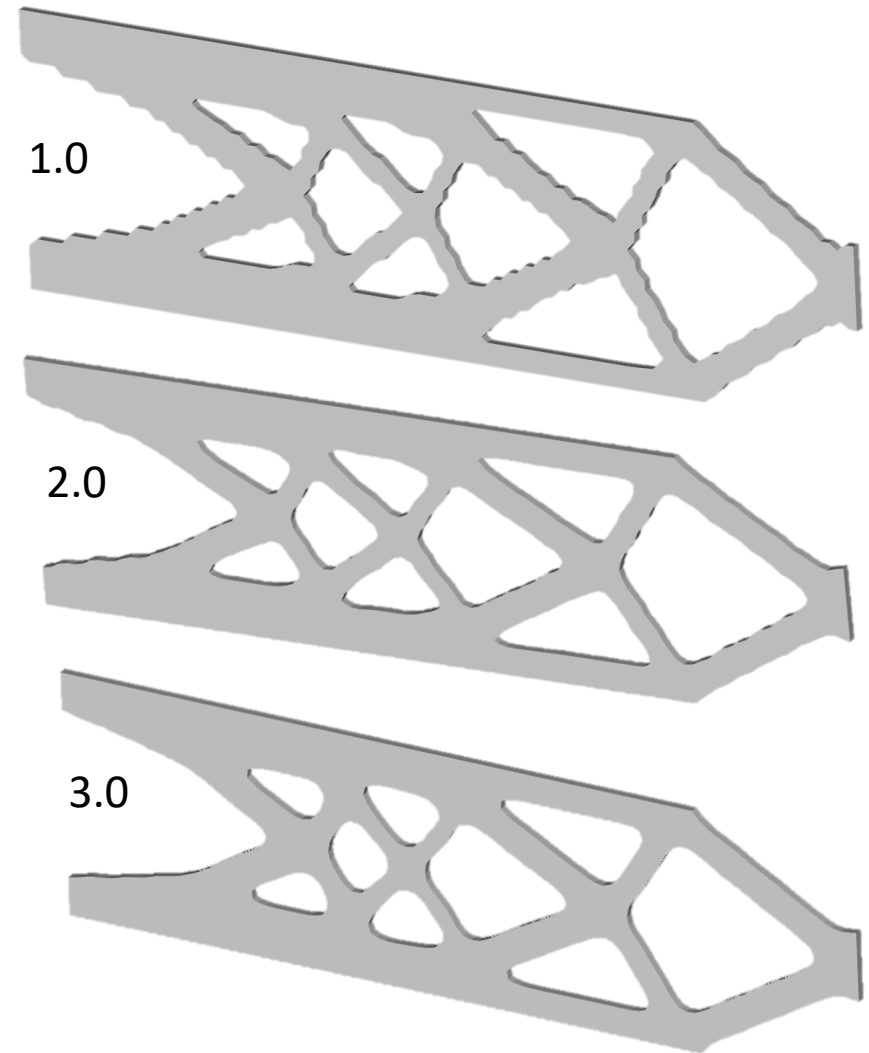


Rerun with `filter_radius_scale=3.0`

- Change the **filter_radius_scale** value to **3.0**
- Resubmit the job with all of the same parameters by clicking on the **plato** node in the tree and then click on the **Submit Simulation Job** in the job submission area in the **Settings** panel

Results for filter_radius_scale=3.0

- Again, increasing **filter_radius_scale** to **3.0** creates an even smoother result.
- With **filter_radius_scale** at 3.0 the filter radius is 3 elements wide and has a broader effect.

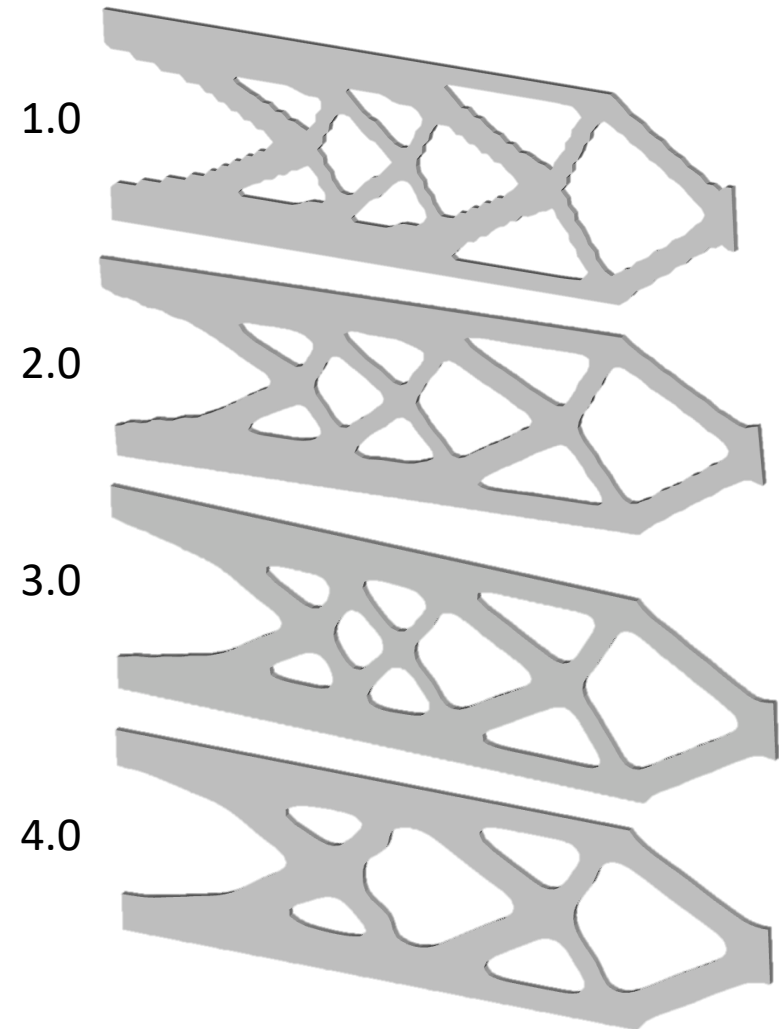


Rerun with `filter_radius_scale=4.0`

- Change the **`filter_radius_scale`** value to **4.0**
- Also, change the **`max_iterations`** to **70**.
- Resubmit the job with all of the same parameters by clicking on the **plato** node in the tree and then click on the **Submit Simulation Job** in the job submission area in the **Settings** panel

Results for filter_radius_scale=4.0

- Again, increasing **filter_radius_scale** to **4.0** creates an even smoother result. However, now the topology has changed. You can see that the result has bulkier members and some of the thinner members are gone all together. As the filter radius gets larger it has a more global smoothing effect and will increase the size of the smallest allowable feature.



Mesh Refinement

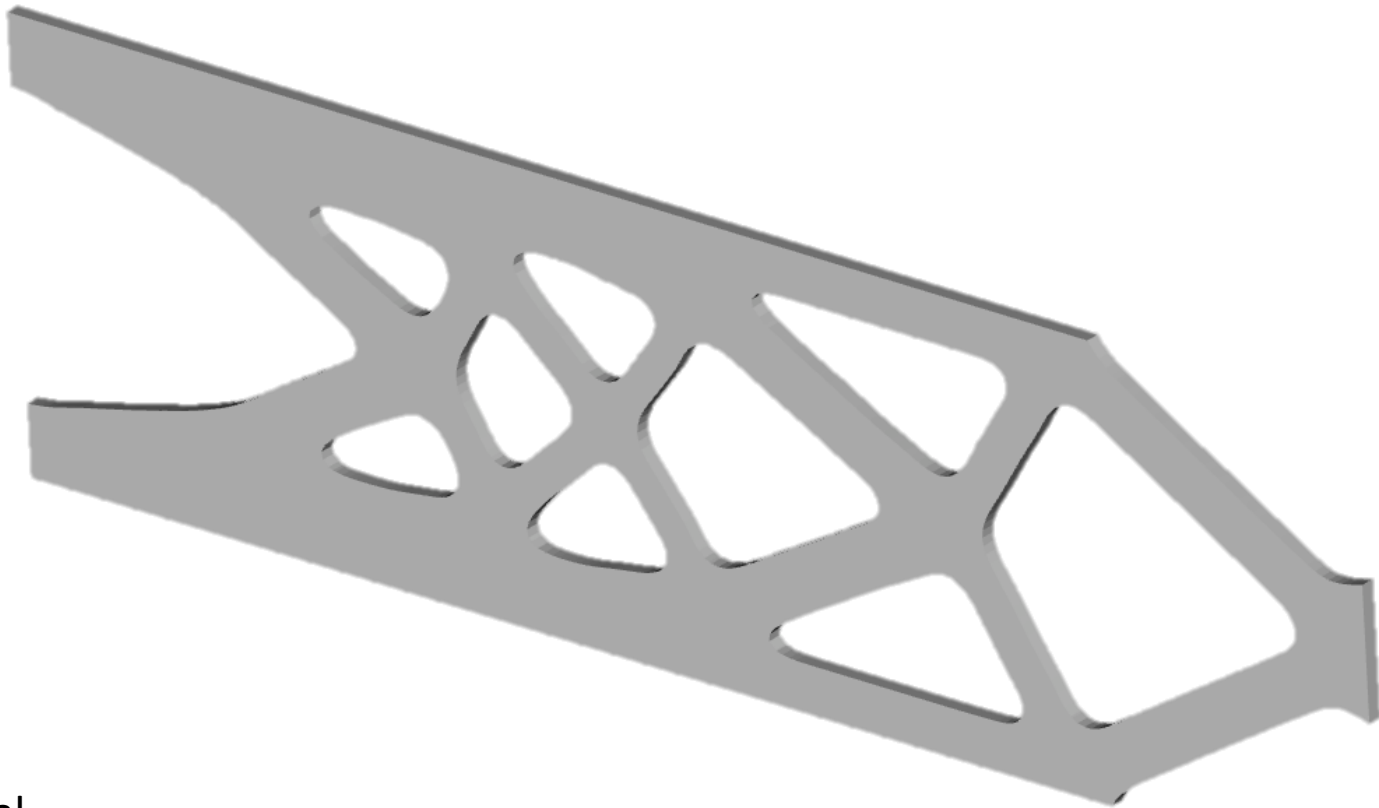
- So, what if you want to resolve fine features in your design but the filtering is limiting how small the features can be?
- The solution is to use a finer mesh. The finer the mesh, the finer the features that can be resolved in the design.
- Let's keep our **filter_radius_scale** at 4.0 but use a finer mesh.

“Refine” the Mesh

- Click on the **Filter** node in the tree under **Geometry/Mesh** to make sure your commands apply to our starting mesh and not one of the result meshes.
- In the **CUBIT** console window type “**edge vis on**” and hit return to turn the mesh edge visibility back on.
- Type the following commands at the **CUBIT** prompt:
 - delete mesh
 - curve 12 interval 1
 - volume all size 0.05
 - mesh volume all
- The “curve...interval” command forces CUBIT to keep one mesh interval through the thickness so that we are only refining the mesh in the XY plane. This will help keep the problem size down.
- Now, rerun the optimization (you may want to increase the number of processors being used for **service “2”**).

Results from Mesh Refinement

- The image below shows how by refining the mesh we were able to maintain the smaller features while still filtering at a radius of 4 elements wide.



filter_radius_scale vs. filter_radius_absolute

- If you were paying close attention to what we just did with mesh refinement you noticed that we kept the filter radius at 4 elements wide but we changed the size of our elements (mesh refinement) from 0.1 to 0.05. This means that the actual filter radius value went from 0.4 to 0.2. This still allowed us to get the smoothness and fine features we wanted. But it highlights the fact that **filter_radius_scale** provides a radius that is relative to mesh edge length. If you want to specify an absolute filter radius value you can use the **filter_radius_absolute** command instead. When using progressive mesh refinement it is often a good idea to use an absolute filter radius so that the feature size in the design stays roughly the same.

Specifying an Absolute Filter Radius

- To use **filter_radius_absolute** instead of **filter_radius_scale** first bring up the topology optimization parameters in the settings panel by clicking on **optimization_parameters** in the tree.
- Right-click on **filter_radius_scale** in the settings panel and choose **Delete**.
- Right-click somewhere in the blank part of the parameters window and choose **Add** and then **filter_radius_absolute**.
- Then enter the desired value for **filter_radius_absolute**.

