



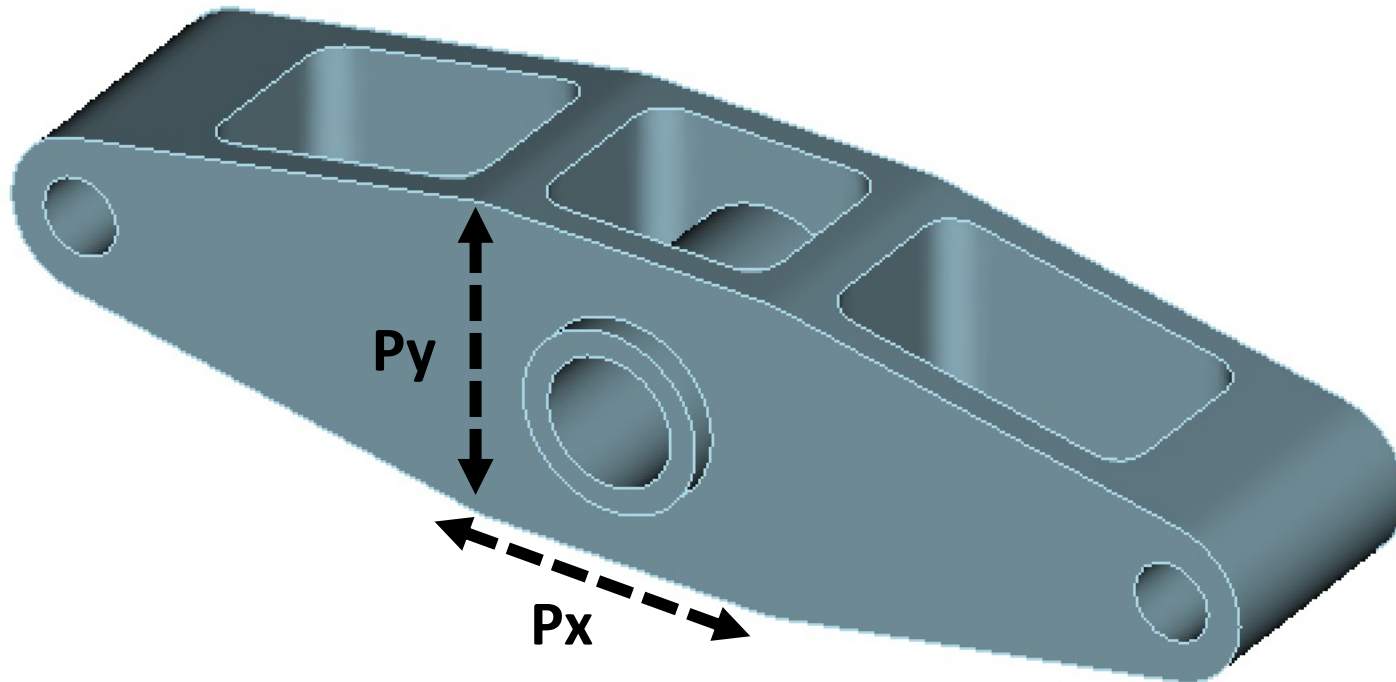
# Surrogate-Based Global Optimization Tutorial

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.

# Problem Description

In this tutorial we will demonstrate the use of a surrogate-based global optimization (SBGO) workflow for enabling global design studies for shape optimization problems with multiple objectives.

There are two shape design parameters:  $P_y$  and  $P_x$ , shown below.

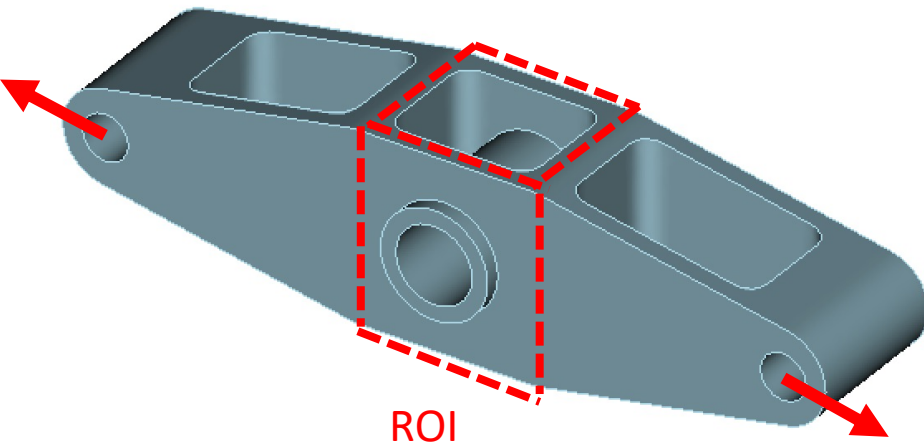


# Problem Description

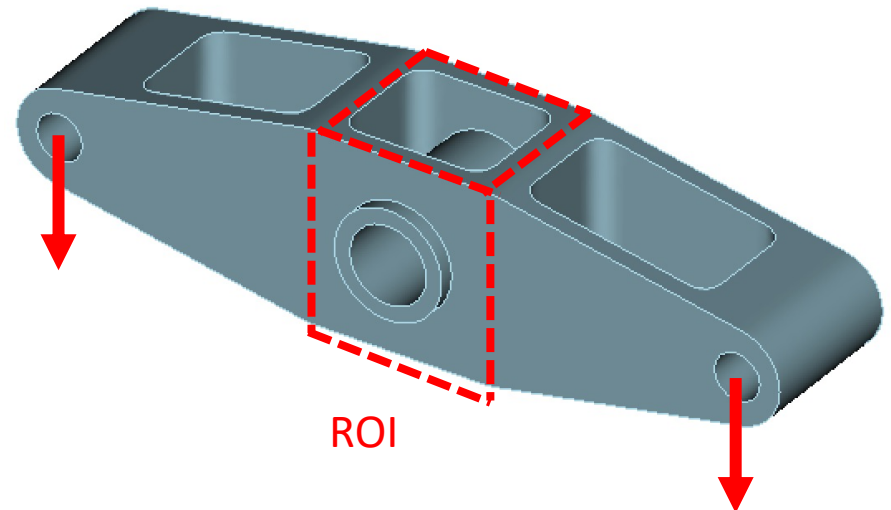
We will consider three objectives: (1) a volume averaged measure of the von mises stress under axial loading (scenario 1), (2) a volume averaged measure of the von mises stress under bending loading (scenario 2), and (3) the design volume. The volume averaged von mises stress will be computed within the region of interest (ROI) indicated below.

This tutorial will require running the Sierra SD physics performer.

Scenario 1

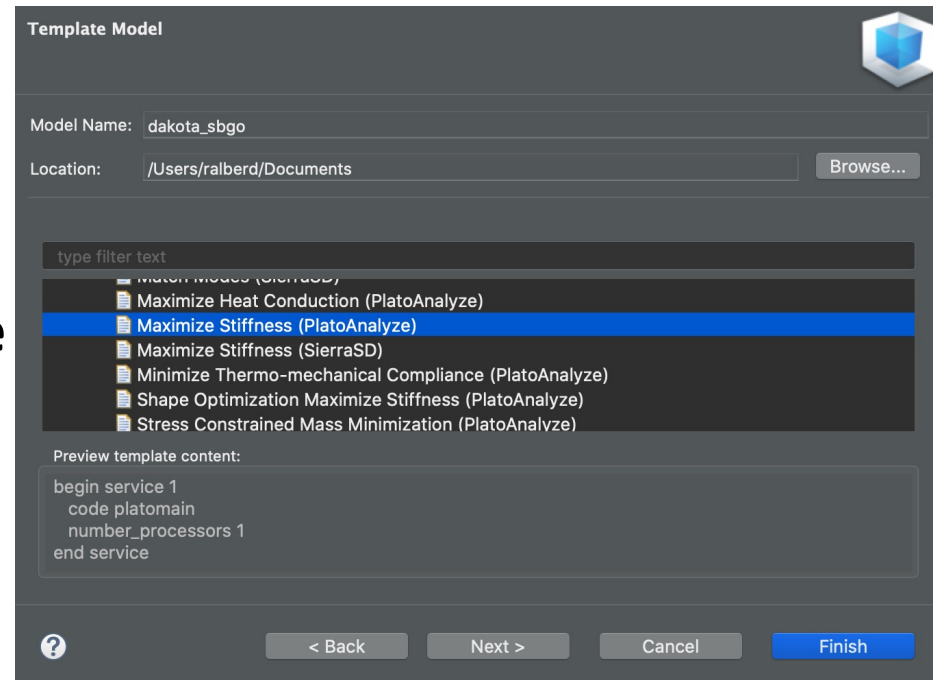
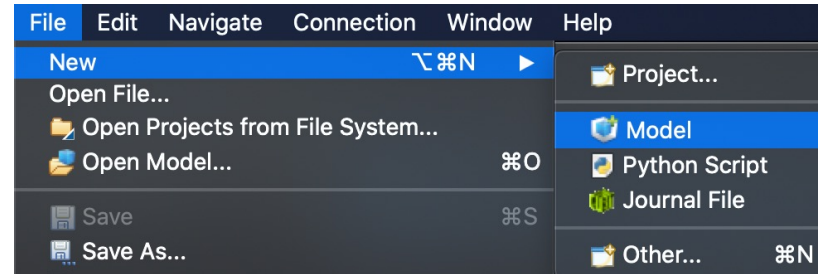


Scenario 2



# Create a New Model

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

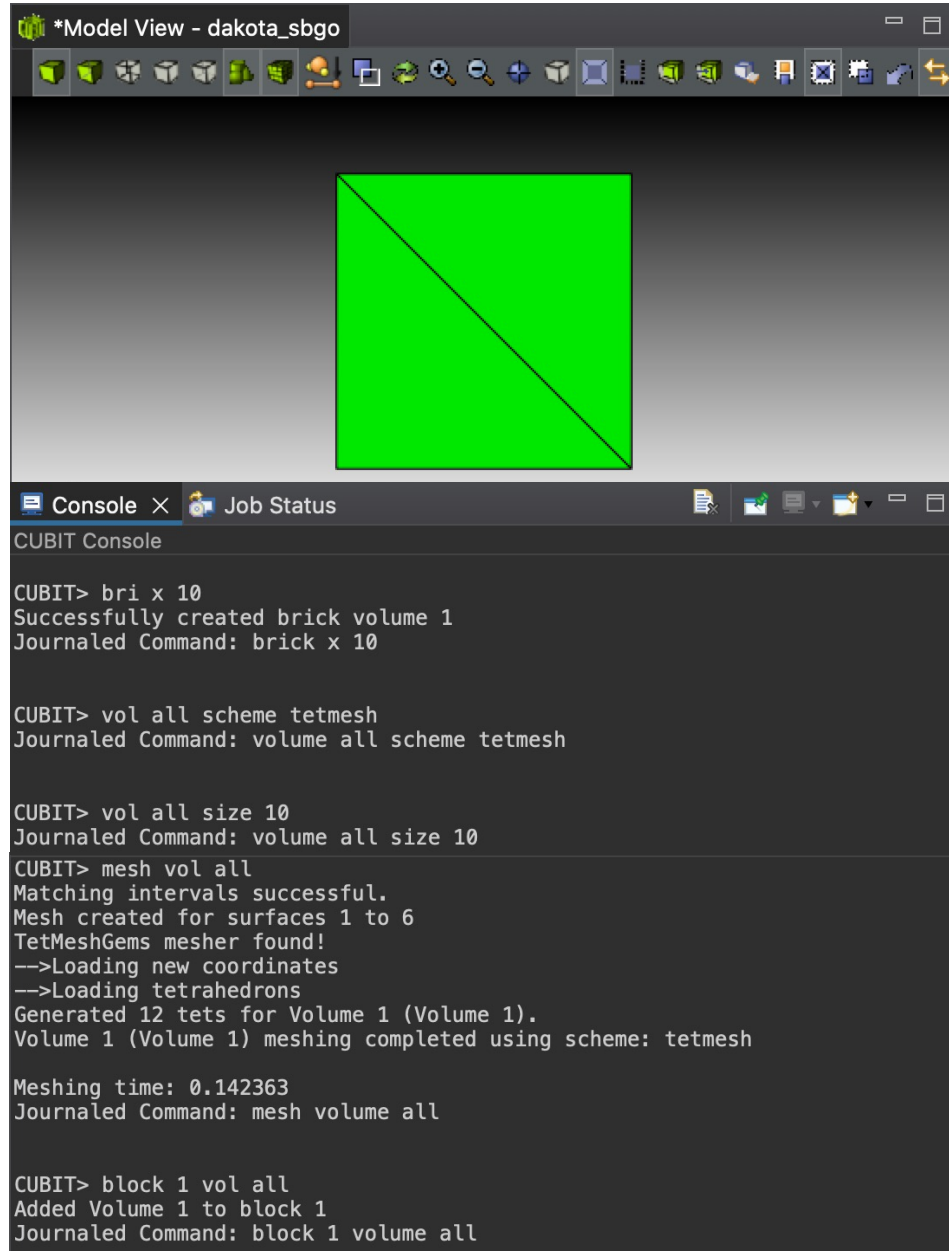


# Provide CSM FILE

- As this is a shape optimization problem, **Engineering Sketch Pad (ESP)** is used to create and mesh a geometry based on the shape design parameter values at each iteration.
- The file defining the geometry for ESP is “**dakota\_sbgo.csm**”.
- To ensure that this file is available when running the problem, place “**dakota\_sbgo.csm**” in the **dakota\_sbgo\_files/simulation** directory that was created with the new model.

# Create a Dummy Mesh

- Although ESP will create a new mesh at each iteration, a dummy mesh is needed for initialization.
- We will create a simple cube mesh for this purpose.
- In the **CUBIT console**, create a dummy mesh following the commands to the right.



The screenshot shows the CUBIT software interface. The top window, titled '\*Model View - dakota\_sbgo', displays a 3D model of a brick (a rectangular prism) in a bright green color. Below the model view is a console window titled 'Console X' and 'Job Status'. The console window contains the following text:

```

CUBIT Console

CUBIT> bri x 10
Successfully created brick volume 1
Journaled Command: brick x 10

CUBIT> vol all scheme tetmesh
Journaled Command: volume all scheme tetmesh

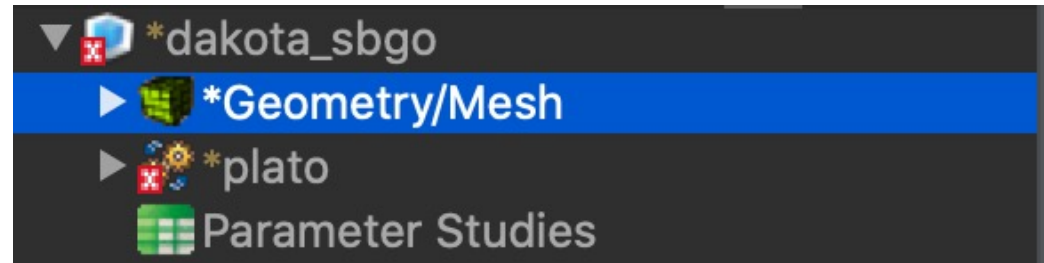
CUBIT> vol all size 10
Journaled Command: volume all size 10

CUBIT> mesh vol all
Matching intervals successful.
Mesh created for surfaces 1 to 6
TetMeshGems mesher found!
-->Loading new coordinates
-->Loading tetrahedrons
Generated 12 tets for Volume 1 (Volume 1).
Volume 1 (Volume 1) meshing completed using scheme: tetmesh

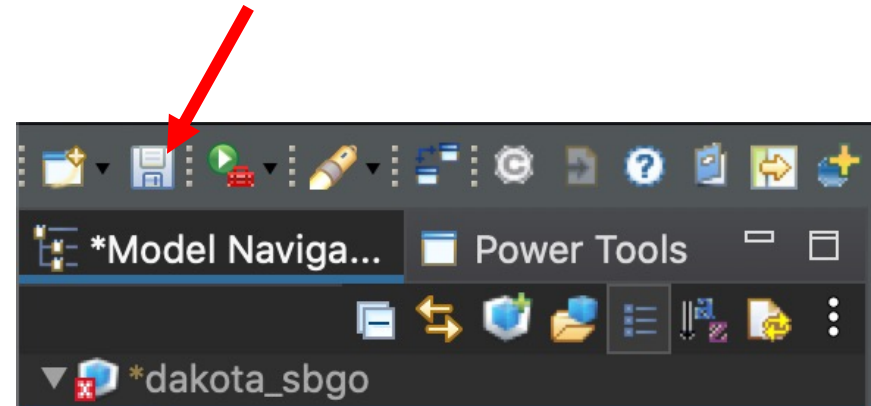
Meshing time: 0.142363
Journaled Command: mesh volume all

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

# Save the Geometry/Mesh

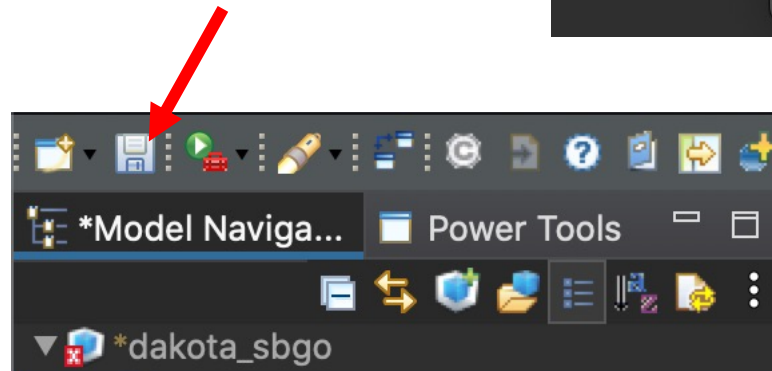
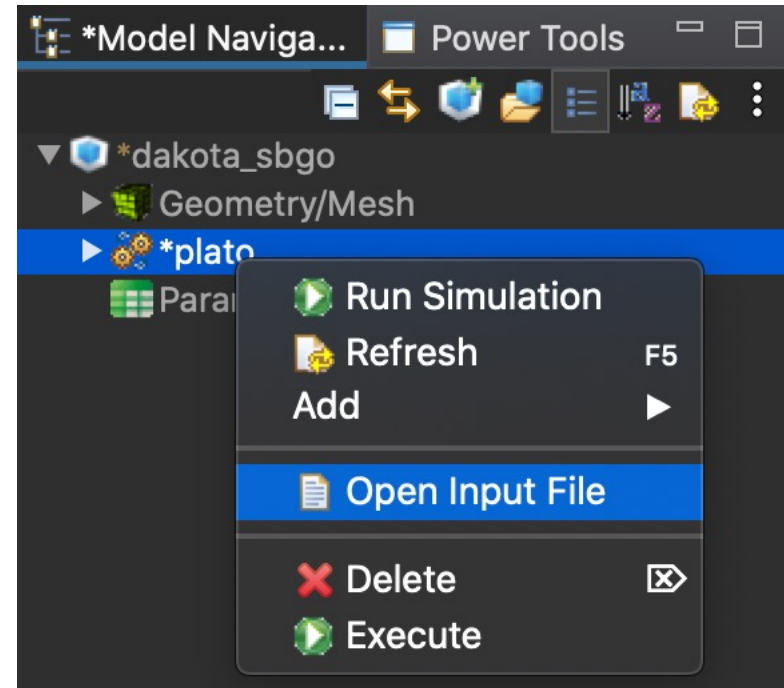


- Click on the “Geometry/Mesh” node in the tree and then click the Save icon in the toolbar.



# Load the Pre-defined Input Deck

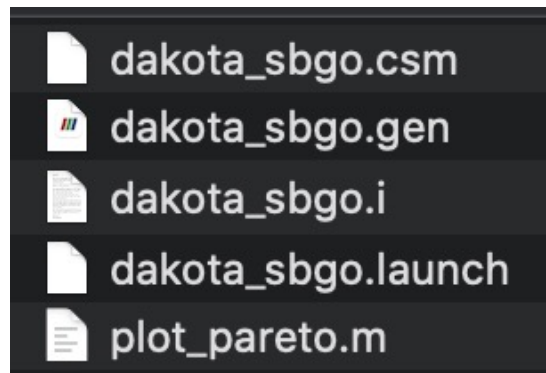
- Right-click on the “**plato**” node in the tree and choose “**Open Input File**”. This will open a text editor showing the current input file for this model. We will be replacing all of the text in the default input deck with that in the file called “**provided\_dakota\_sbgo.i**”.
- Open “**provided\_dakota\_sbgo.i**” in a text editor of your choice and copy and paste its contents into the input deck editor in Plato replacing the text that was there.
- Then click on the “**Save**” icon in the toolbar to save the model.





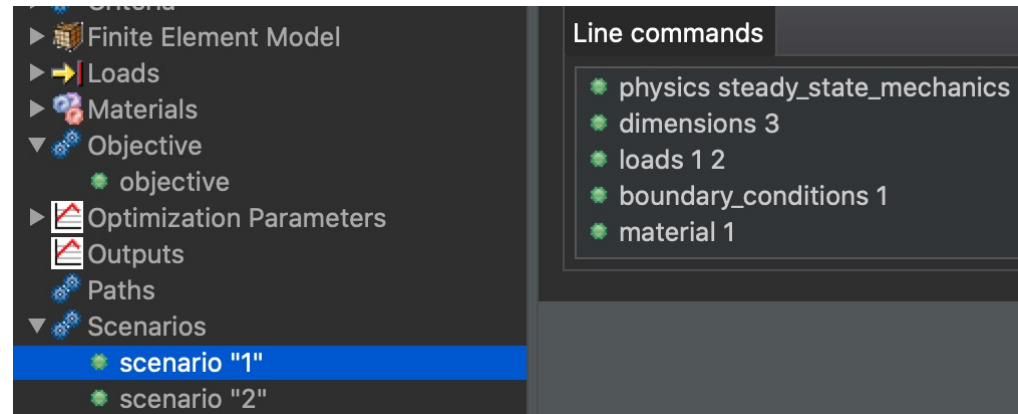
# Provide Postprocessing Script

- In order to postprocess the output provided by dakota for this surrogate-based global optimization workflow, a Matlab script “**plot\_pareto.m**” is provided with the tutorial.
- To ensure that this file is available in the working directory for postprocessing when the problem has run, place “**plot\_pareto.m**” in the **dakota\_sbgo\_files/simulation** directory that was created with the new model.
- At this point, you should have five files in your **dakota\_sbgo\_files/simulation** directory as shown below.

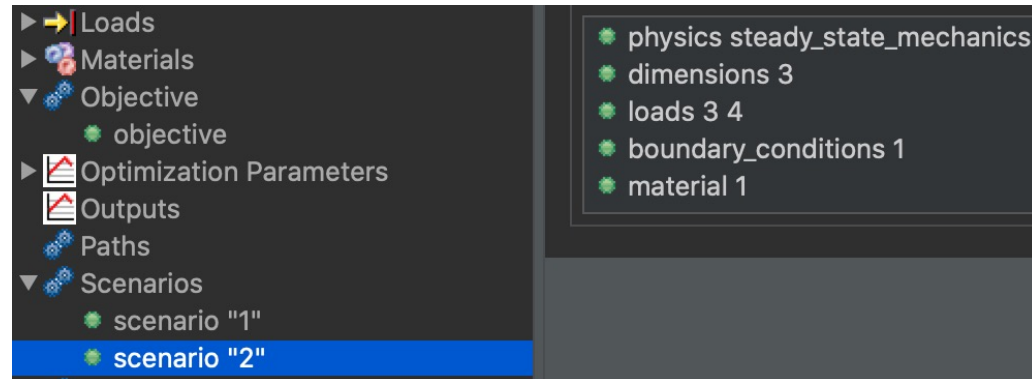


# Examine the Scenarios

- Click on “**scenario “1”**” in the tree and note that it uses loads 1 and 2. These loads will be used to define the axial case.

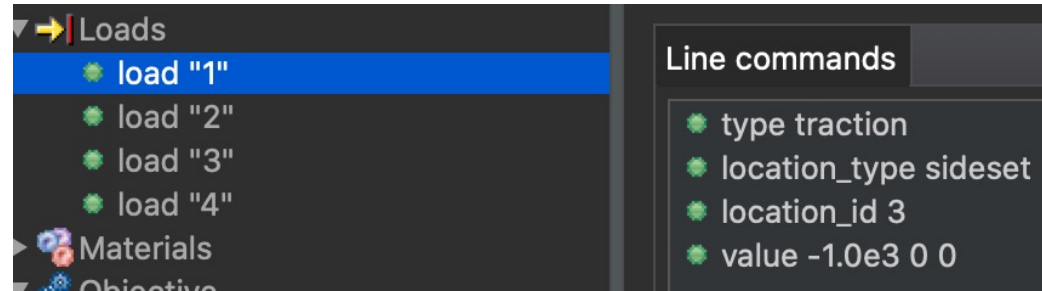


- Click on “**scenario “2”**” in the tree and note that it uses loads 3 and 4. These loads will be used to define the bending case.

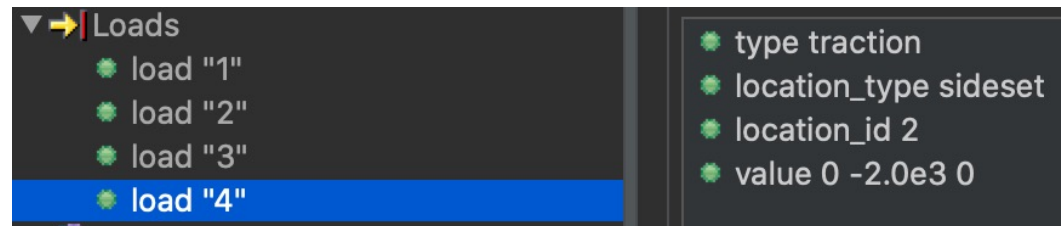
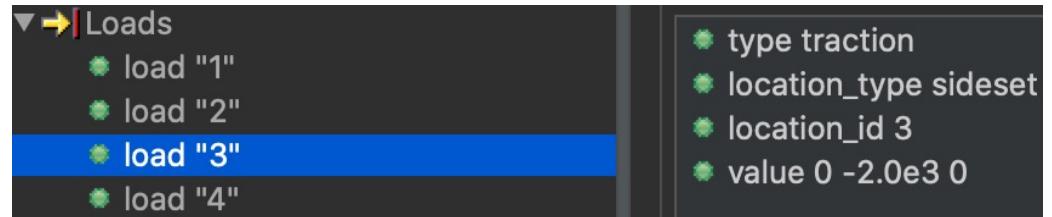


# Examine the Loads

- Click on “**load “1”**” in the tree and note the values. Do the same with “**load “2”**”. These loads define the axial case.

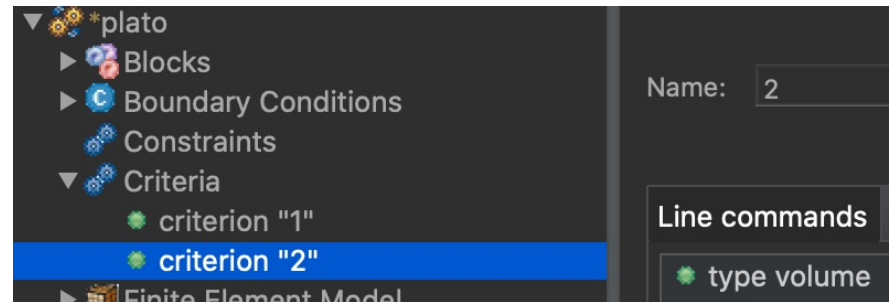
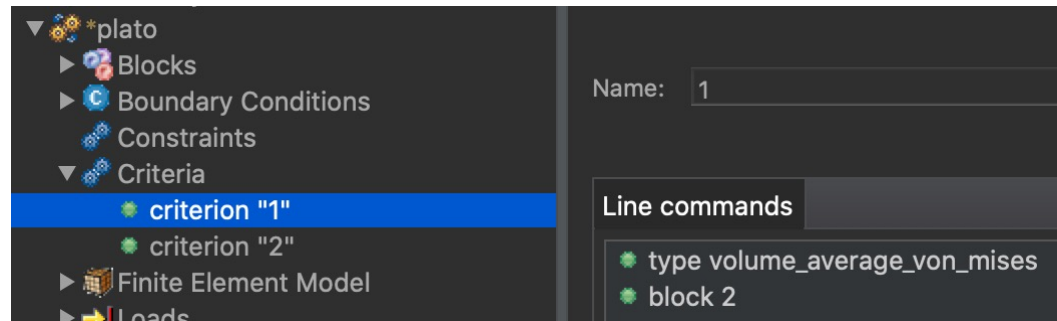


- Click on “**load “3”**” in the tree and note the values. Do the same with “**load “4”**”. These loads define the bending case.



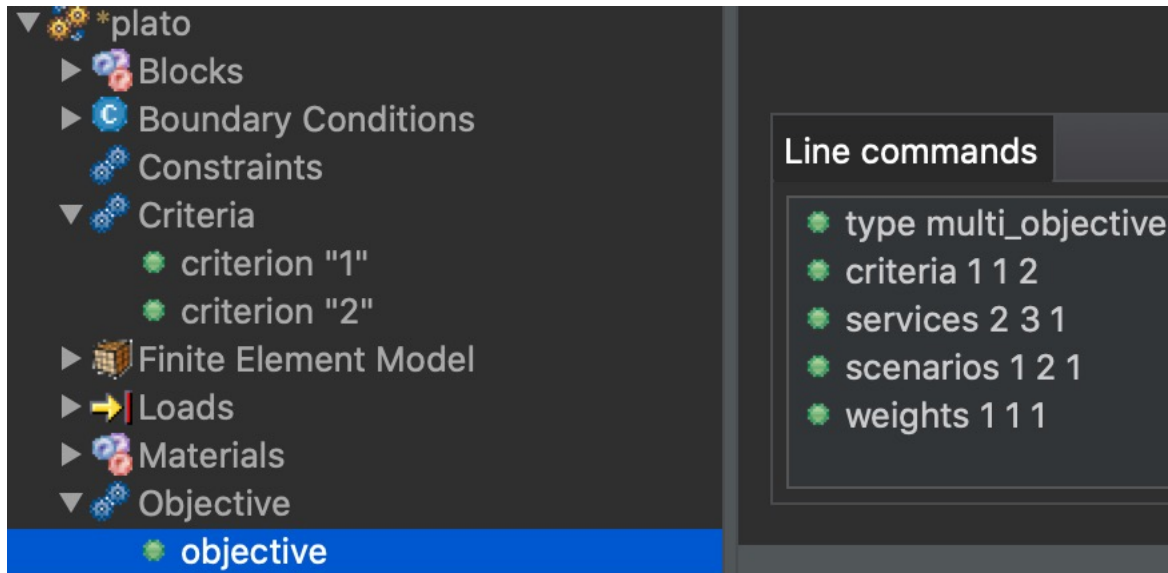
# Examine the Criteria

- Click on “**criteria “1”**” in the tree and note that it is of type “**volume\_average\_von\_mises**”. This criterion will be used as an objective such that Plato will try to reduce the average stress in the region of interest. The region is specified as block 2 in this criterion.
- Click on “**criteria “2”**” in the tree and note that it is of type “**volume**”. This criterion will be used as one of the objectives such that Plato will try to reduce the amount of volume used in the design.



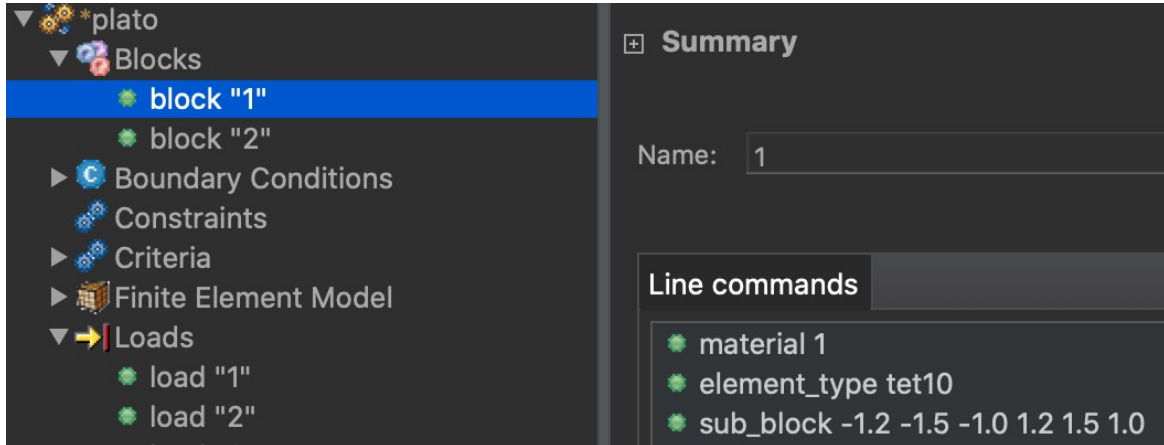
# Examine the Objective

- Click on “**Objective**” in the tree. Note that the type is “**multi\_objective**”, meaning that multiple, possibly competing, objectives will be considered simultaneously during optimization.
- The objectives to be considered are defined by their criterion-service-scenario combinations. In this case, criterion 1 (volume\_average\_von\_mises) is being evaluated for both scenario 1 (axial loading) and scenario 2 (bending loading). In addition, criterion 2 (volume) is being evaluated.

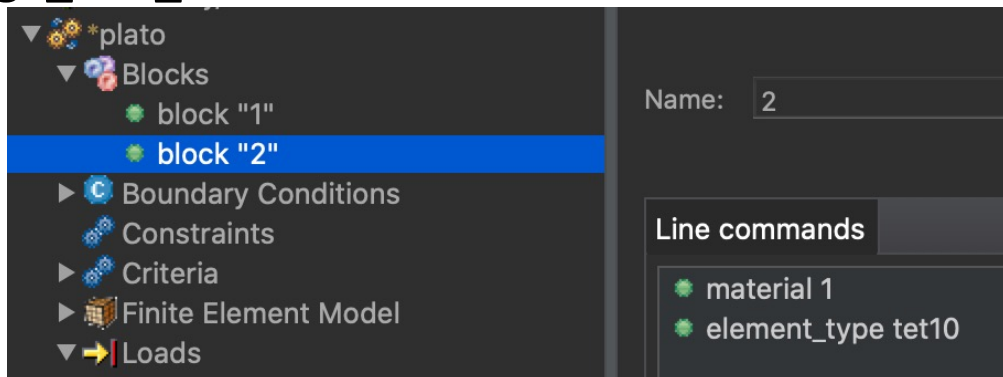


# Examine the Blocks

- Click on “**block “1”**” in the tree. Note the **sub\_block** keyword. This keyword is used to define a separate block from block 1 in terms of a bounding box. The bounding box coordinates follow the **sub\_block** keyword.

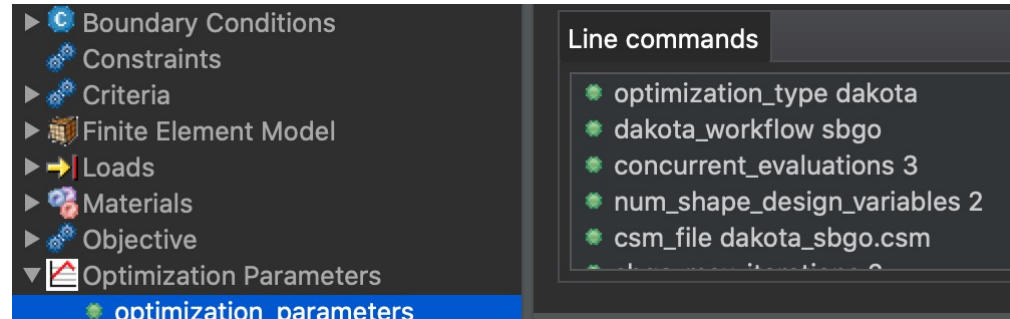


- Click on “**block “2”**” in the tree. This is the block which will be created as a **sub\_block** of block 1 and defines the region of interest where the **volume\_average\_von\_mises** criteria will be evaluated.



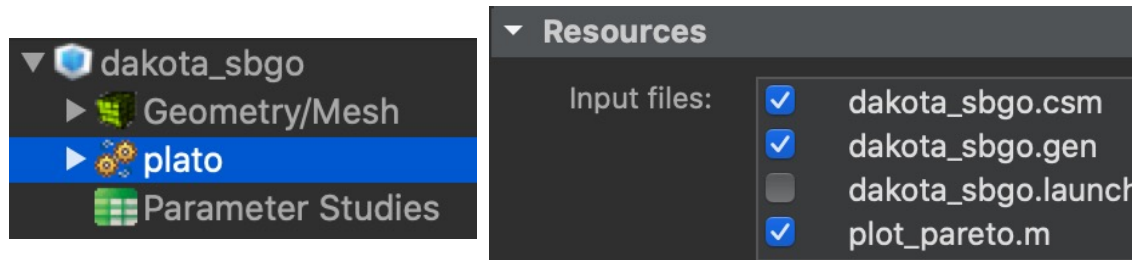
# Examine the Optimization Parameters

- Click on “**optimization\_parameters**” in the tree.
- Note first the **optimization\_type** is set to dakota, meaning that dakota will drive the optimization.
- Next, note that **dakota\_workflow** is set to sbgo. This indicates the supported surrogate-based global optimization workflow.
- Finally, note the parameters defined in the lower screenshot. These parameters define the sbgo workflow. The meanings of these parameters are given in the input deck reference manual.

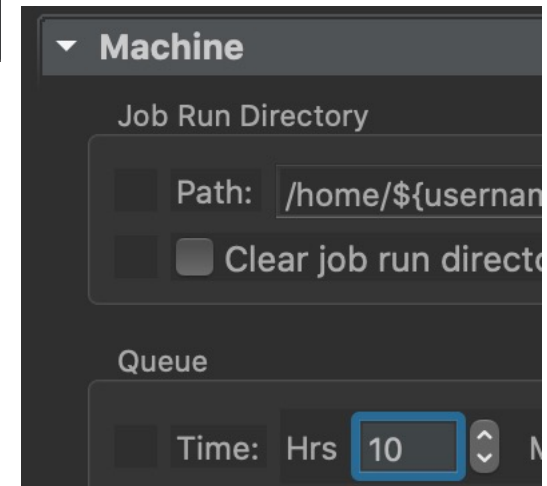


# Run the Optimization

- 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 you want to use. A configured execution template **PlatoEngine\_cee-compute016\_dakota.properties** is provided as part of this tutorial\*.



- In the **Resources** area make sure **dakota\_sbgo.gen**, **dakota\_sbgo.csm**, and **plot\_pareto.m** are checked so that these files gets pushed to the working directory.
- In the **Machine** area, allot a long time for running since many function evaluations will be needed (it should take around 2 hours).
- Choose any other preferences and launch the job by clicking on **Submit Simulation Job** toward the top of the panel.

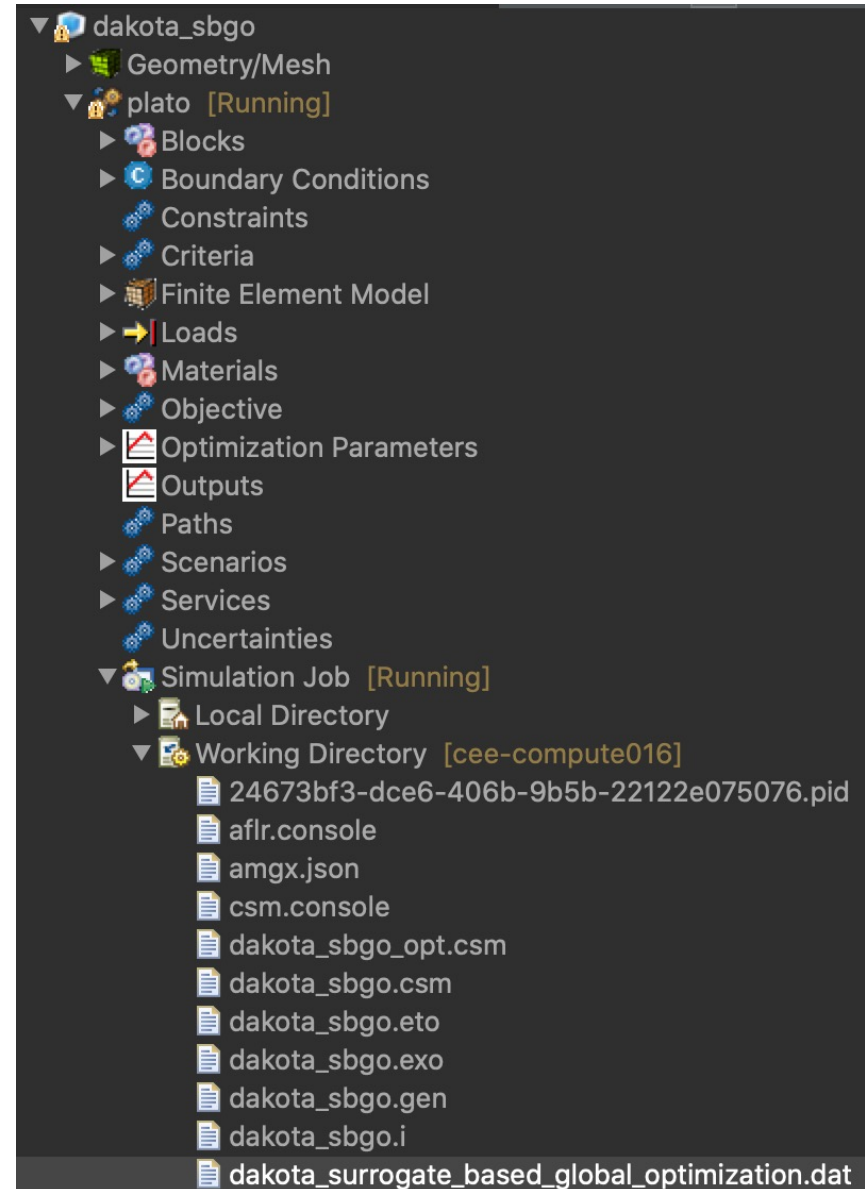


\*If running on a different machine, copy the execution instructions from **PlatoEngine\_cee-compute016\_dakota.properties** to the execution template on the desired machine. This is needed to properly run the example and perform postprocessing.



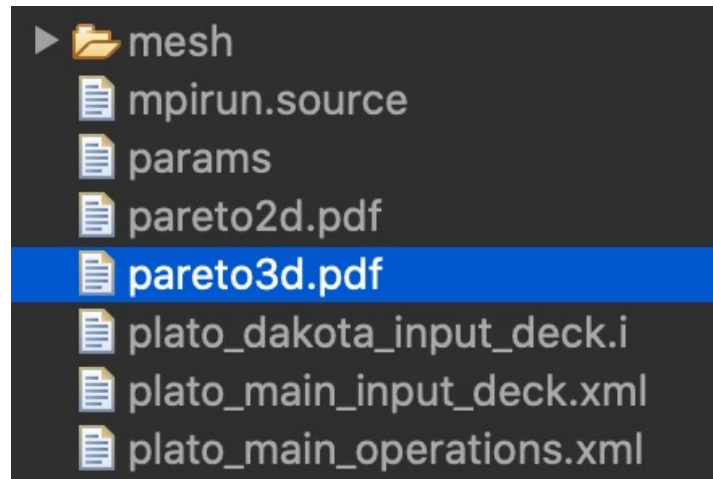
# Monitor the Optimization

- Click on “**Simulation Job**” in the tree and “**Working Directory**” to see the files produced by this run.
- The “**dakota\_surrogate\_based\_global\_optimisation.dat**” file contains the results of each evaluation performed.
- This file will populate after running an initial number of evaluations (do not expect it to contain results right away)



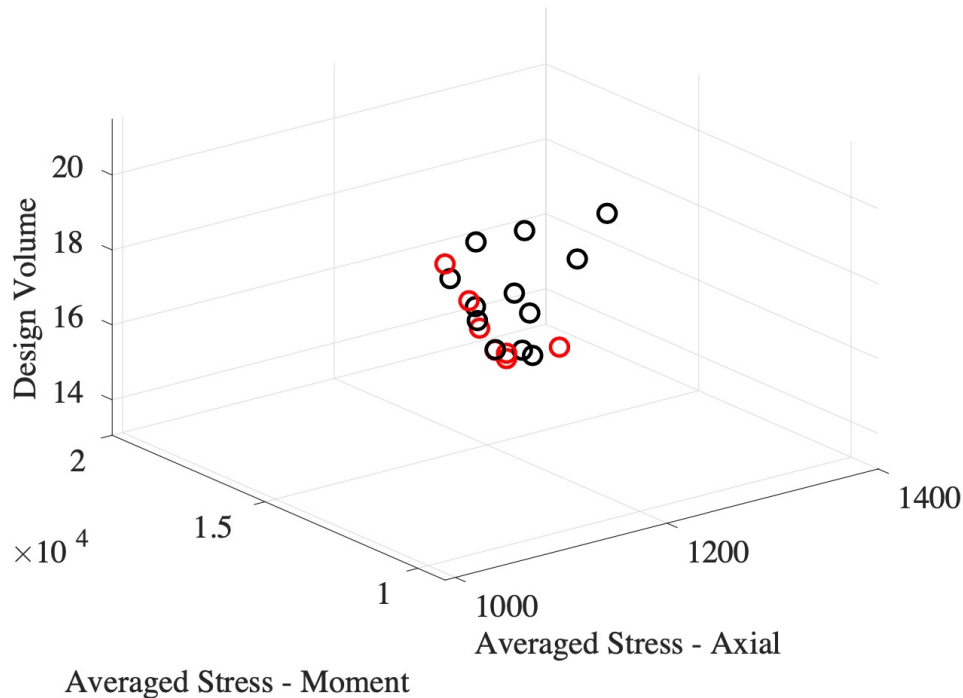
# Inspect the Results

- After the optimization run is completed, the postprocessing script will be run. This will produce two plots of the multi-objective pareto frontier for visualization: **pareto2d.pdf** and **pareto3d.pdf**.
- You can open these from the “**Working Directory**” node in the tree by double clicking on them. This should bring them up in your local pdf viewer.



# Inspect the Results

- Opening **pareto3d.pdf** should bring up a plot like the one below.
- The axes of this plot are the three different objectives.
- The **black** circles are points evaluated during an initial sampling of the parameter space, while the **red** circles correspond to points along the pareto frontier obtained during optimization.



# Inspect the Results

- Opening **pareto2d.pdf** should bring up a plot like the one below.
- The axes of this plot are the two volume average von mises objectives, while the size of the circle corresponds to the volume.
- Again, the **black** circles are points evaluated during an initial sampling of the parameter space, while the **red** circles correspond to points along the pareto frontier obtained during optimization.

