Parametric Study of a Loaded Beam

Parametric studies let you solve an engineering challenge with a bit of cleverness and a bit more brute force. Essentially, you can set up one or more loops over a range of values for key variables under consideration, such as material strength, geometric configuration, or grid resolution. You can then compare the results side-by-side to detect optimal configurations or quantitative differences.

In this project, you will use Python to create a simple beam, add a load to the end, and solve the model. The beam should be a 0.1 m×0.1 m square extruded to a length L. You should `ENCASTRE` one end and apply a face load (`ConcentratedForce`) of 1 kN in the $-y$ direction to a consistent set of nodes along the other end (*i.e.*, a row or two of nodes). Use a mesh of *at least* 0.02 m in resolution and any appropriate element, but justify your selection.

You will then vary both the length of the beam and the material it is composed of. (Note that this table is not of *pairs* to test, but of the range for each value; that is, there will be 4×4=16 cases to test.)

	Material			**Beam Length**
A36 steel	E = 207 GPa	ν = 0.30	ρ = 7850 kg·m^{-3}	L = 1 m
Gray cast iron	E = 166 GPa	ν = 0.26	ρ = 7300 kg·m^{-3}	L = 2 m
Al 6061	E = 169 GPa	ν = 0.33	ρ = 2700 kg·m^{-3}	L = 3 m
Titanium	E = 103 GPa	ν = 0.34	ρ = 4510 kg·m^{-3}	L = 4 m

Add two nested loops around the main code which allow you to test all of the cases:

```
materials = {}
materials['steel'] = (207e9, 0.30, 7850) # and so forth
lengths = range(1,5,1) # doesn't include right-hand value
for material in materials:
    for length in lengths:
        ...
        currentMaterial = myModel.Material(name=materials[mat])
        elasticProperties = (materials[mat][0], materials[mat][1])
        currentMaterial.Elastic(table=(elasticProperties, ) )
        densityProperties = (materials[mat][2])
        currentMaterial.Density(table=(densityProperties, ) )
        ...
        currentJobName = 'beam_%s_%dm'%(mat,length)
        ...
```

You will find it useful to either name all of the jobs differently (*e.g.*, `beam_steel_1m`) or to capture the field output in an `XYData` table (see Masoud Safdari's script `SafdariHeatOpt.py` for an example of this). You should report the maximum stress (and its location) and the maximum displacement for each simulation run in tabular format. You may acquire files to be used as the basis of this simulation from the course web site.

```
SafdariHeatOpt.py
beamExample.py
```

(If you require assistance with writing Python code due to unfamiliarity with the language, you may request help from either myself or the TA. You shouldn't need much more than what's in this document, however.)

As before, you will document your simulations in a 10–12 page report (with figures) containing the sections:

a) Problem description (shape, grid, etc.)
b) Details of the simulation and numerical parameters
c) Observations of numerical behavior (residual convergence rate, mesh behavior, etc.)
d) Discussion of the resulting physics as discussed above, including plots

The report should be formatted with 1.5 line spacing, 1 inch margins on all sides, and set in 11 point serif font. All figures and tables should be numbered and have labels and captions.