Powering Innovation That Drives Human Advancement

Automation challenge

Problem Answer - Preprocessing 1

First identify the two circular faces

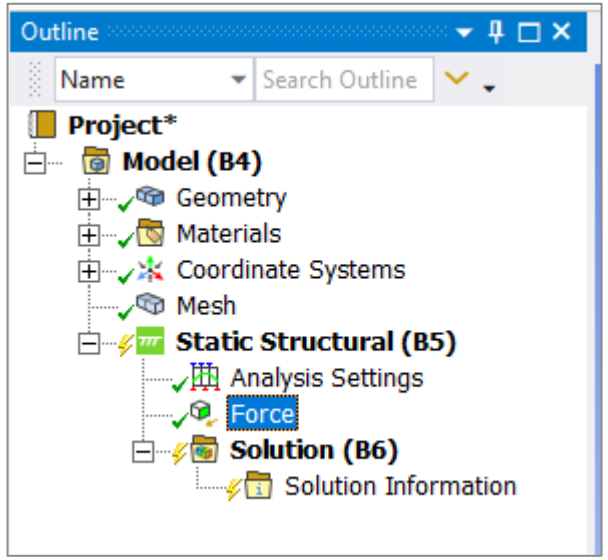
- How: a circle has only one edge! We know how to loop through all the faces. We can identify the two faces in a variety of way (Area, centroid, normal,….) we are going to do it counting the edges belonging to a surface. The two cylindrical surfaces will only have one edge

```
surfaces=[]
for assembly in DataModel.GeoData.Assemblies:
    for part in assembly.Parts:
        for body in part.Bodies:
            for surface in body.Faces:
                if len(surface.Edges)==1:
                    surfaces.append(surface)
```

We also need the length of the cylinder (L) and the cross-section area (A)

```
A=surfaces[0].Area
L=surfaces[0].Centroid[1]-surfaces[1].Centroid[1] # assuming cylinder axis=Y axis
```

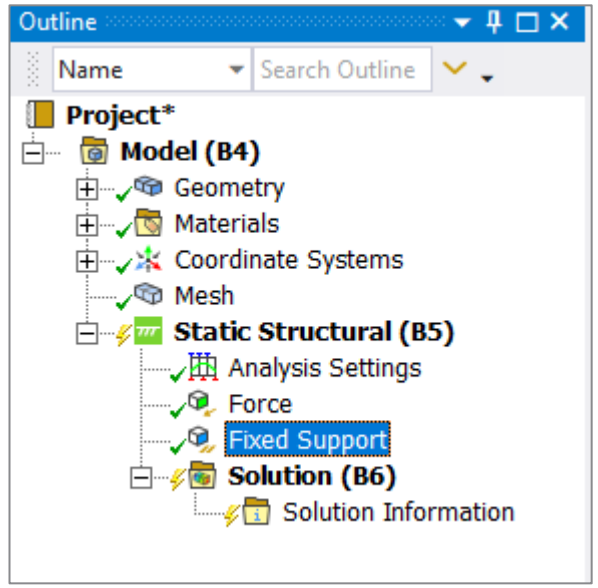
Problem Answer - Preprocessing 2



We can now create the force on the first surface with 1N magnitude

```
analysis= Model.Analyses[0]
Force=analysis.AddForce()
face1=DataModel.GeoData.GeoEntityById(surfaces[0].Id)
selection =
ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Entities = [face1]
Force.Location = selection
Force.Magnitude.Output.DiscreteValues = [Quantity("1 [N]")]
```

Problem Answer - Preprocessing 3



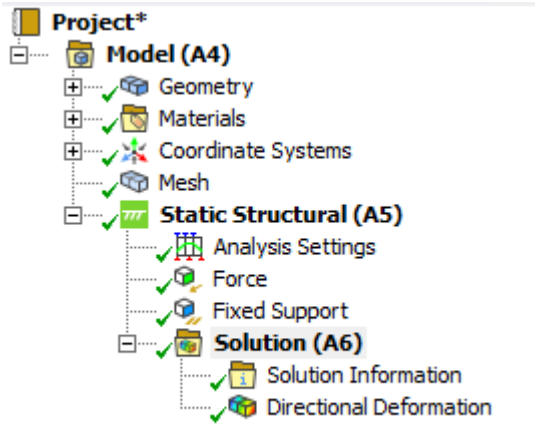
And a fixed support on the other surface:

```
support=analysis.AddFixedSupport()
face2=DataModel.GeoData.GeoEntityById(surfaces[1].Id)
selection =
ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Entities = [face2]
support.Location = selection
```

We are now ready to solve

```
analysis.Solve(True)
```

Problem Answer - Postprocessing 1

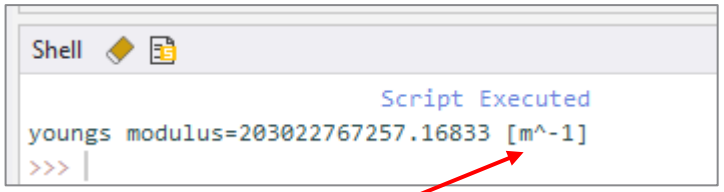


We now need to create the appropriate results and extract the maximum displacement in the axis direction:

```
res=analysis.Solution.AddDirectionalDeformation()
res.NormalOrientation=NormalOrientationType.YAxis
res.EvaluateAllResults()
DL=res.Maximum
```

Then we can calculate:

```
e=DL/L
S=1/A
E=S/e
print 'youngs modulus=' + str(E)
```



The value of the E has incorrect units.
Why?

Problem Answer - Postprocessing 1.1

Quantity failure

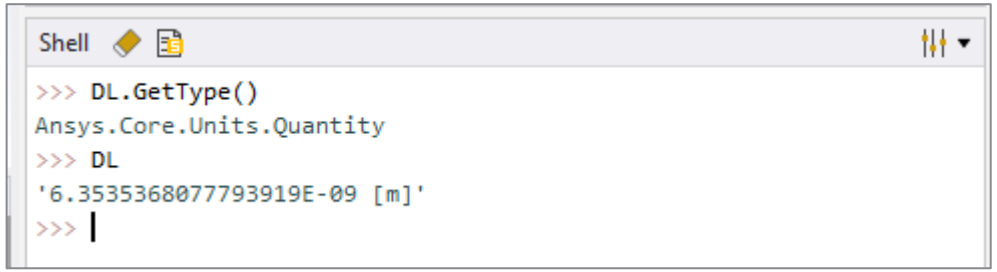
L is a System.Double without units
DL is a Quantity with units [m]
e should be unit-less
A should be m^2 but is also unit-less

```
S=1/A - is [] shoud be N/m^2
E=S/e  -[Pa/e]
```

Important, understand the parameters used!

Solutions: use unitless values everywhere (DL.Value), or preferably, use Quantities everywhere:

```
Length = Quantity(str(L)+ "[m]")
Area = Quantity(str(A)+ "[m^2]")
Force = Quantity(str(1)+ "[N]")
```



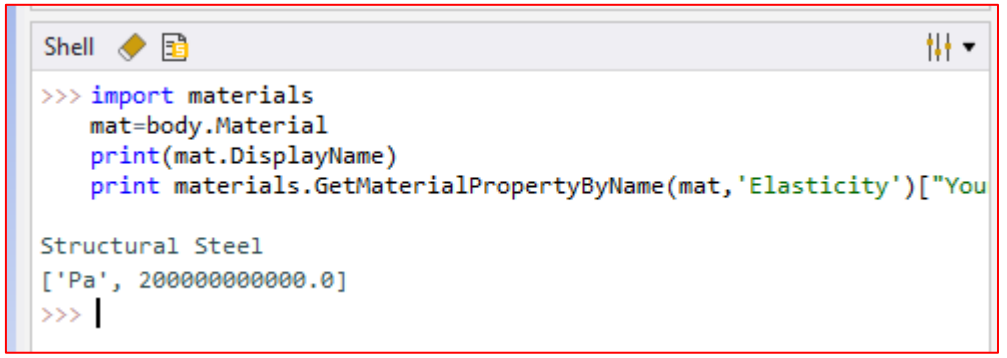
$$\textbf{Young's Modulus} = \frac{\sigma}{\varepsilon}\ ; \sigma = \frac{F}{A}; \varepsilon = \frac{\Delta L}{L}$$

```
>>> EModulus
'201419979353.70389 [J m^-3]'
```

Problem Answer - Postprocessing 2

We could compare this value with the material young's modulus
To view materials we can use the materials module

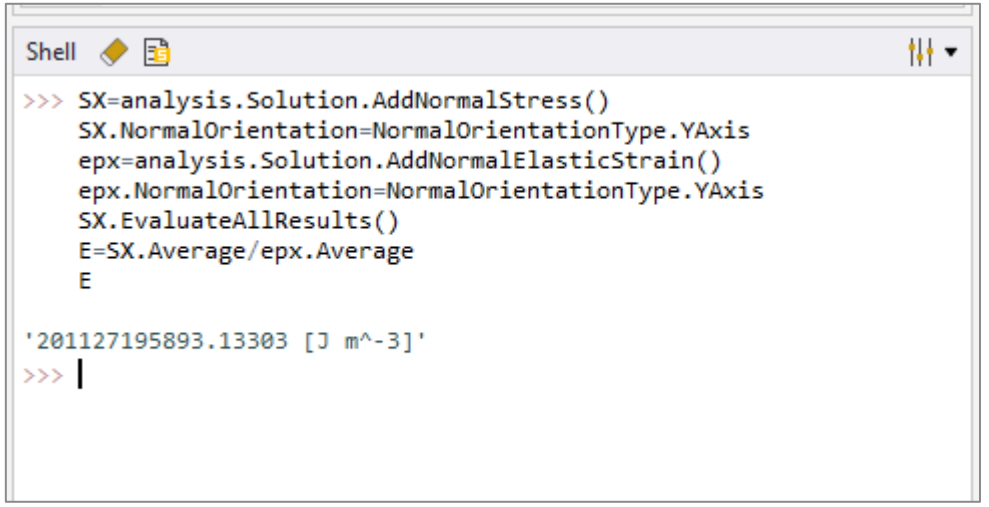
```
import materials
mat=body.Material
print(mat.DisplayName)
print materials.GetMaterialPropertyByName(mat,'Elasticity')["Young's Modulus"]
```



Problem Answer - Postprocessing 3

We could also have gotten the ["Young's Modulus"] by diving the average stress in the axis direction by the average strain in the same direction

```
SX=analysis.Solution.AddNormalStress()
SX.NormalOrientation=NormalOrientationType.YAxis
epx=analysis.Solution.AddNormalElasticStrain()
epx.NormalOrientation=NormalOrientationType.YAxis
SX.EvaluateAllResults()
E=SX.Average/epx.Average
E
```



Note the model is assumed to be aligned to the Y axis and the strain in calculated for that direction.


Ansys