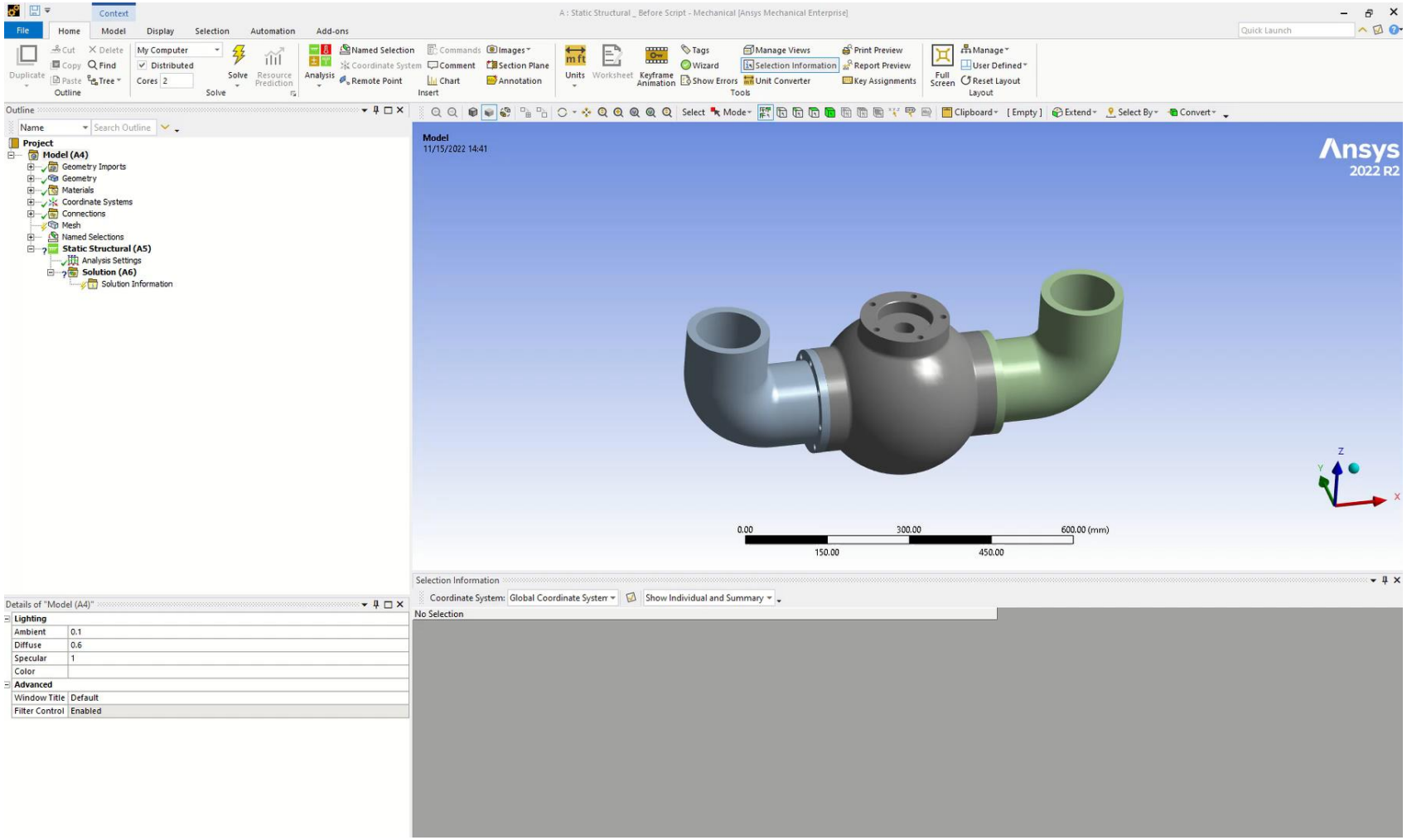
Powering Innovation That Drives Human Advancement

Getting started with Mechanical scripting

What we are going to learn today: complete model setup automation



Python 101

Python in one slide

- **Easy to learn/write/read**
  - No variable declaration
  - No variable initialization
  - Indentation defines where loops/functions end

- **Syntax**
  - Case sensitive
  - # 1-line comment
  - “”” Multi line comment “”“

- **Operators**
  - + , - , = , /
  - Logical operators: and, or, not
  - Relational operators: > , <, == , !=
  - Membership: in, not in
  - Identity: is, is not

- **Flow control**
  - if , if…else , elif
  - while loop
  - for loop

```
count = 0
while (count < 9):
   print 'The count is:', count
   count += 1  #same as: count = count + 1

print "Good bye!"
```

```
for letter in 'Python':     # First Example
   print 'Current Letter :', letter

fruits = ['banana', 'apple',  'mango']
for fruit in fruits:        # Second Example
   print 'Current fruit :', fruit

for index in range(len(fruits)):  # Third Example
   print 'Current fruit :', fruits[index]
```

- **Functions**
  - Begin with keyword: def

```
# Function definition is here
def printinfo( name, age ):
   "This prints a passed info into this function"
   print "Name: ", name
   print "Age ", age
   return

# Now you can call printinfo function
printinfo( age=50, name="miki" );
printinfo( "Raj", 30 );
```

- **Data types**
  - Numbers, strings, lists, dictionaries, tuples

```
list1 = ['physics', 'chemistry', 1997, 2000];
list2 = [1, 2, 3, 4, 5 ];
list3 = ["a", "b", "c", "d"];

print "list1[0]: ", list1[0]      # list1[0]:  physics
print "list2[1:5]: ", list2[1:5]  # list2[1:5]:  [2, 3, 4, 5]
```

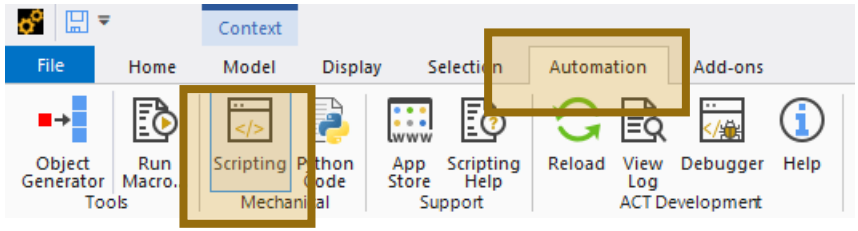
```
dict0 = {'Alice': '2341', 'Beth': '9102', 'Cecil': '3258'}
dict1 = { 'abc': 456 };
dict2 = { 'abc': 123, 98.6: 37 };

Ndict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'};
print "dict['Name']: ", dict['Name'];  # dict['Name']:  Zara
print "dict['Age']: ", dict['Age'];    # dict['Age']:  7
```

Learning Python

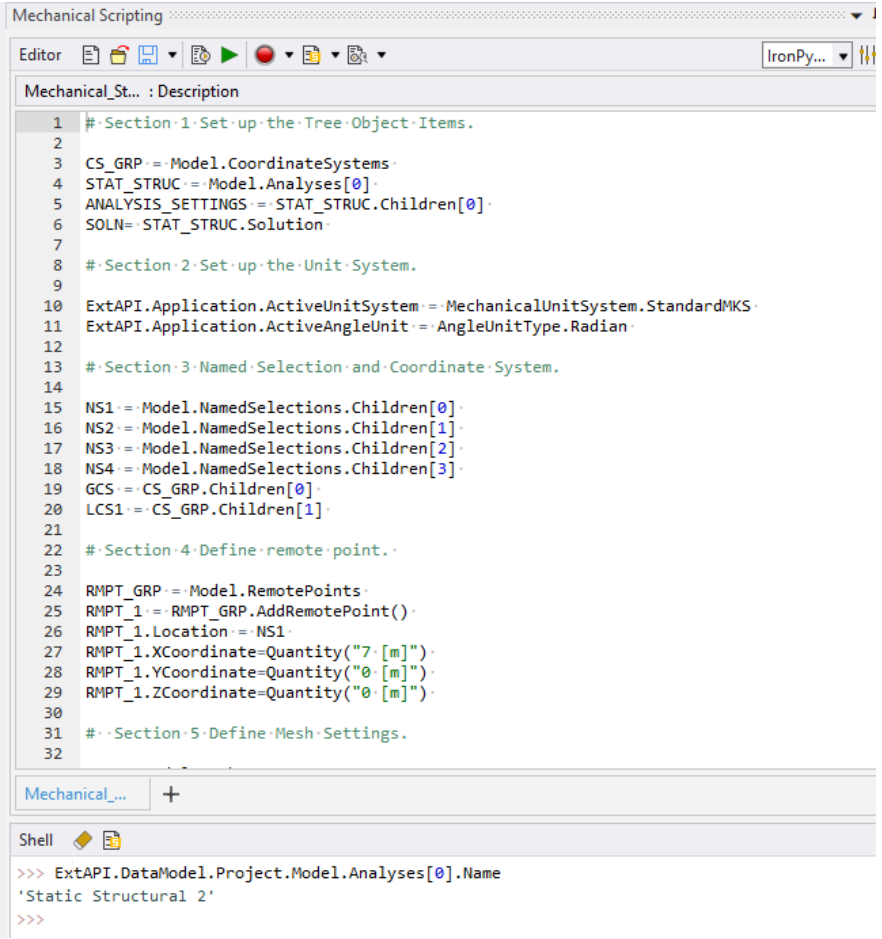
- Many resources available online (LinkedIn Learning, W3Schools, GeekforGeeks, …)
- One training offered by Ansys on the Ansys Innovation Space

Using the Mechanical scripting console

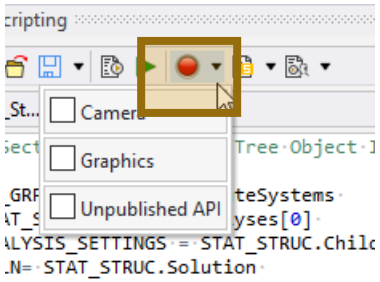
Mechanical scripting console



Access Mechanical scripting console

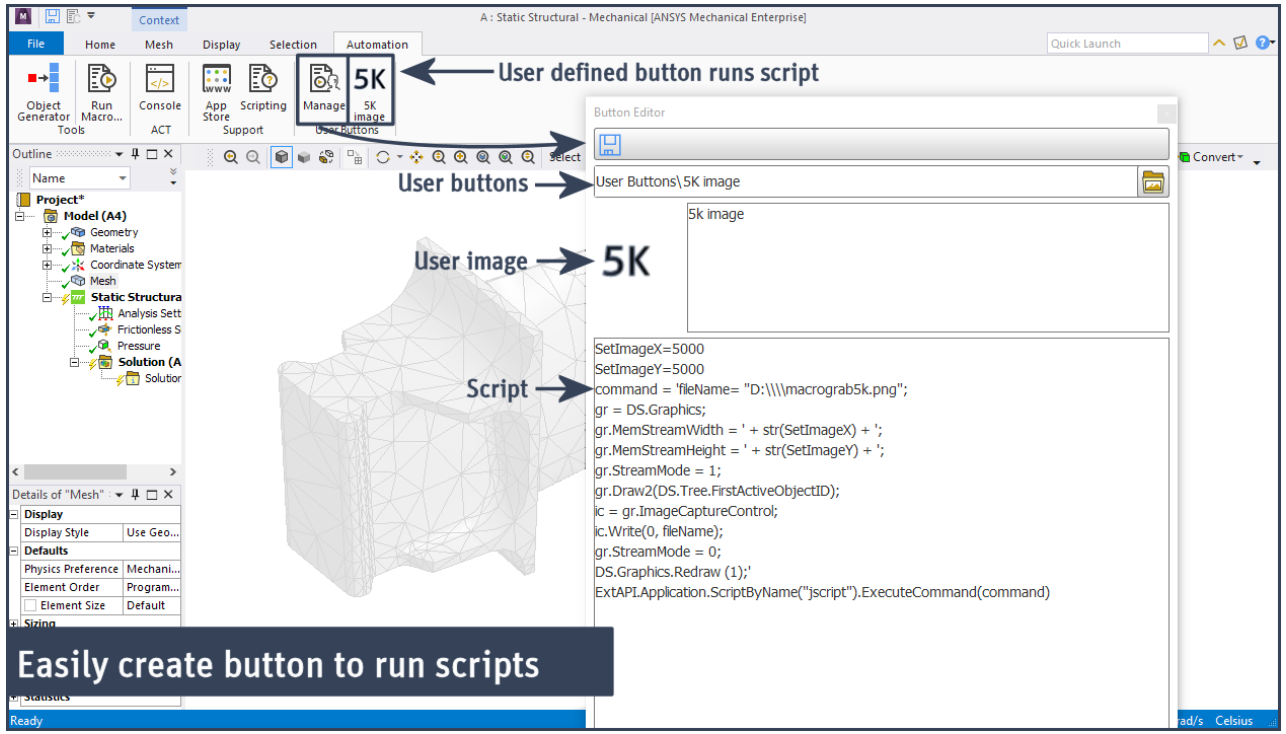


Script editor and Shell command prompt

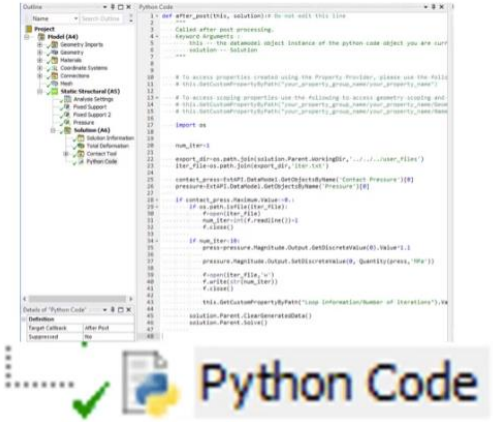


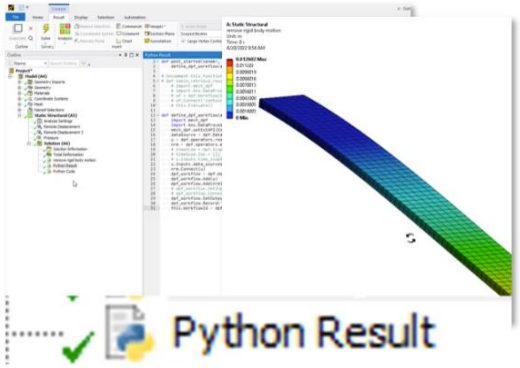
Recording capabilities

More advanced usage



Save script to button
(Script executed when clicked on)



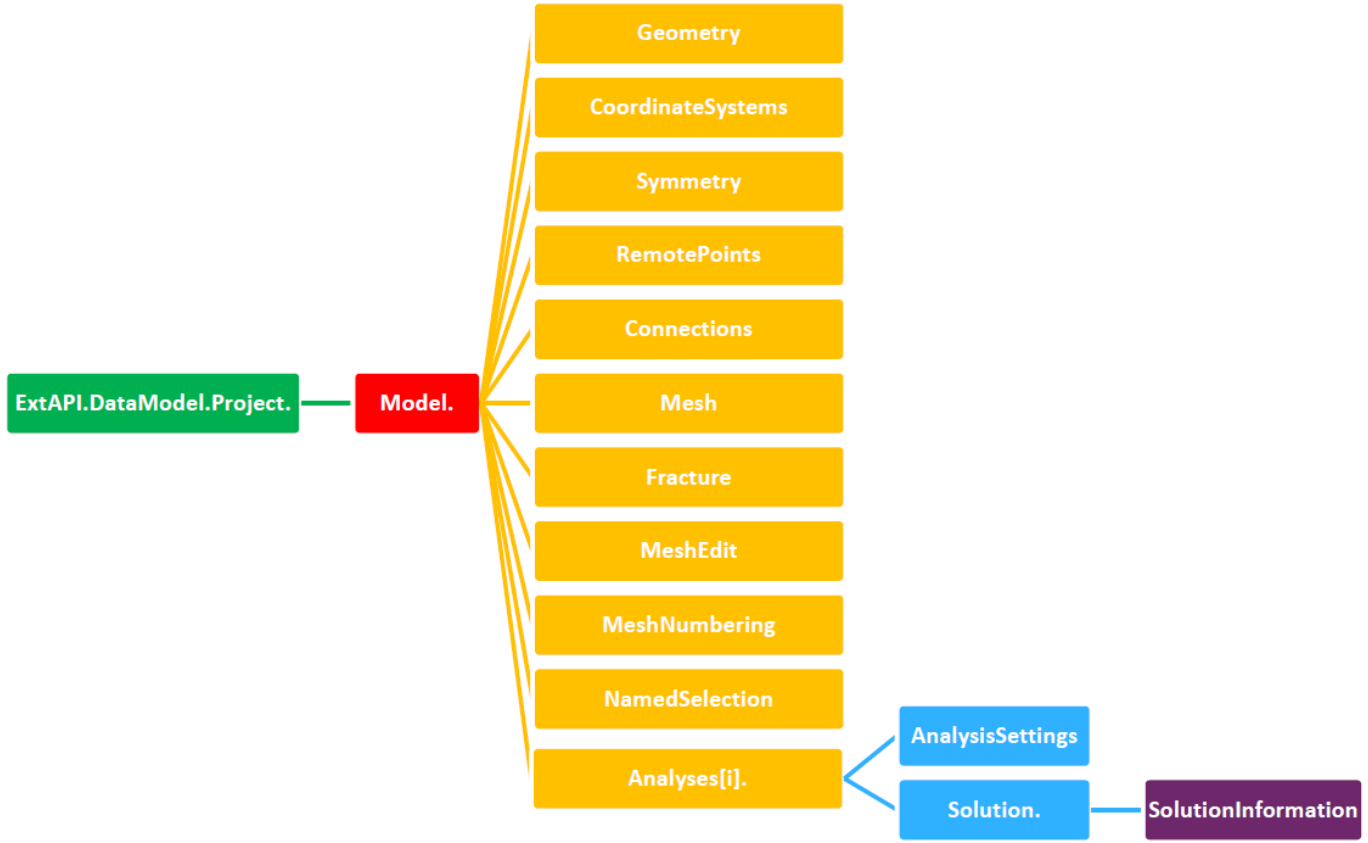


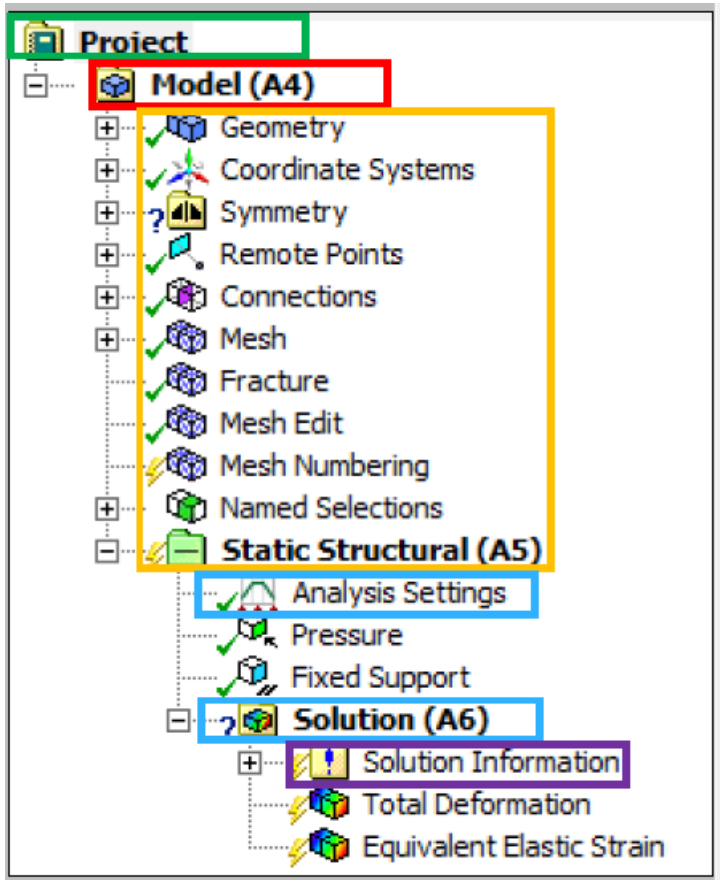
Execute code automatically during simulation

Understanding Mechanical automation API

Mechanical Automation API

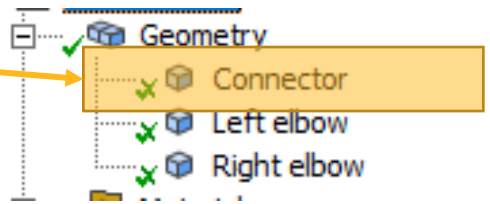
```
# Reference model
model = ExtAPI.DataModel.Project.Model
```

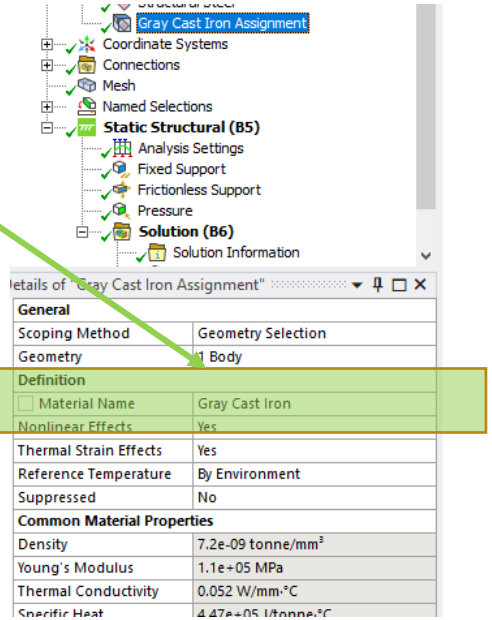


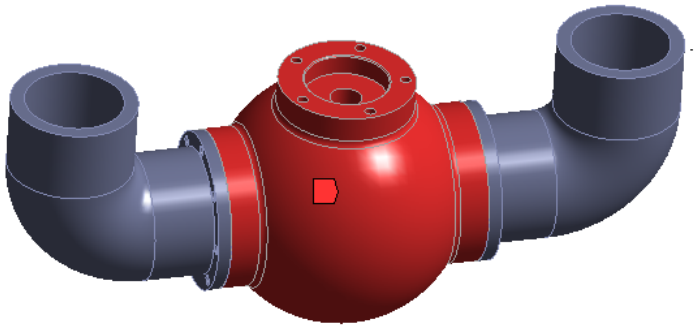


Let’s dive into the script: define material assignment

```
# Change material assignement for body named "Connector"
import materials
matAssignment = model.Materials.AddMaterialAssignment()
body =  ExtAPI.DataModel.GetObjectsByName("Connector")[0].GetGeoBody()
tempSel = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
tempSel.Ids = [body.Id]
matAssignment.Location = tempSel
matAssignment.Material = "Gray Cast Iron"
ExtAPI.DataModel.Tree.Refresh()
```

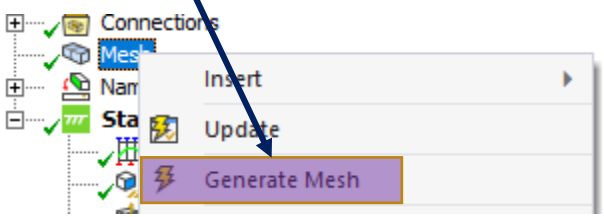


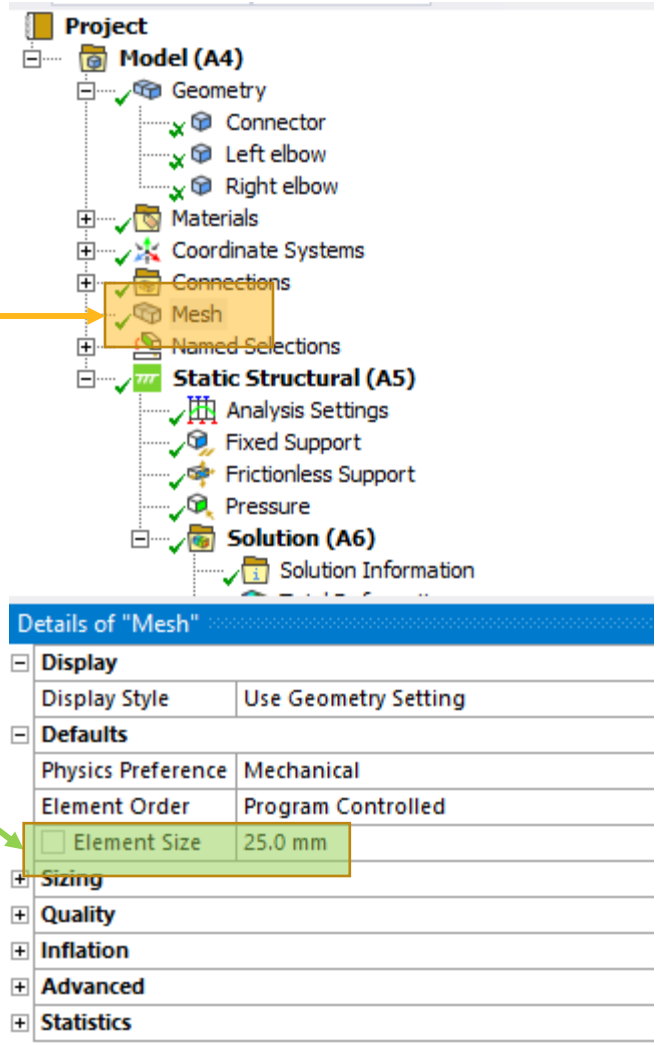




Let’s dive into the script: define mesh settings

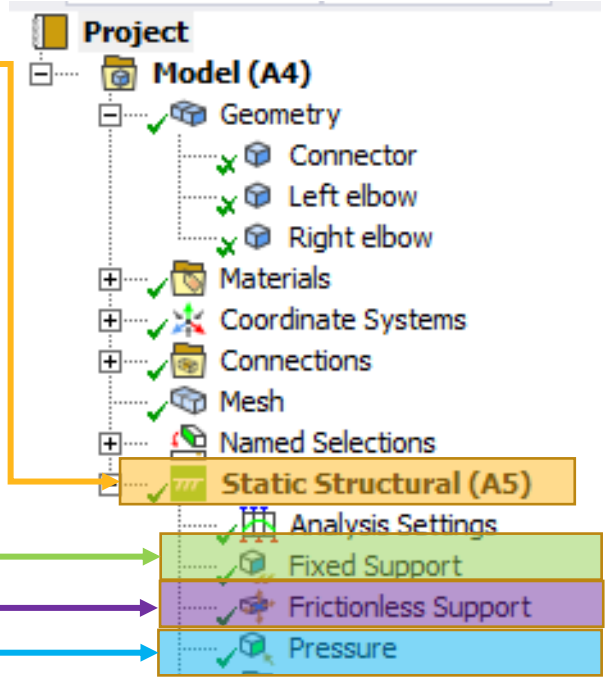
```
# Define mesh settings
mesh = model.Mesh
mesh.ElementSize = Quantity('25 [mm]')
mesh.GenerateMesh()
```

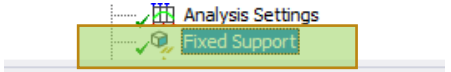




Let’s dive into the script: define boundary conditions

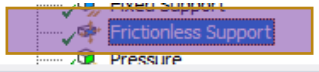
```
# Define boundary conditions:
analysis = model.Analyses[0]

fixedSupport = analysis.AddFixedSupport()
fixedSupport.Location = ExtAPI.DataModel.GetObjectsByName("NSFixedSupportFaces")[0]

frictionlessSupport = analysis.AddFrictionlessSupport()
frictionlessSupport.Location = ExtAPI.DataModel.GetObjectsByName("NSFrictionlessSupportFaces")[0]

pressure = analysis.AddPressure()
pressure.Location = ExtAPI.DataModel.GetObjectsByName("NSInsideFaces")[0]
pressure.Magnitude.Inputs[0].DiscreteValues = [Quantity("0 [s]"), Quantity("1 [s]")]
pressure.Magnitude.Output.DiscreteValues = [Quantity("0 [Pa]"), Quantity("15 [MPa]")]
```





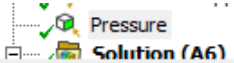
Details of "Fixed Support"

Scope	
Scoping Method	Named Selection
Named Selection	NSFixedSupportFaces
Definition	
ID (Beta)	92
Type	Fixed Support
Suppressed	No



Details of "Frictionless Support"

Scope	
Scoping Method	Named Selection
Named Selection	NSFrictionlessSupportFaces
Definition	
ID (Beta)	94
Type	Frictionless Support
Suppressed	No



Details of "Pressure"

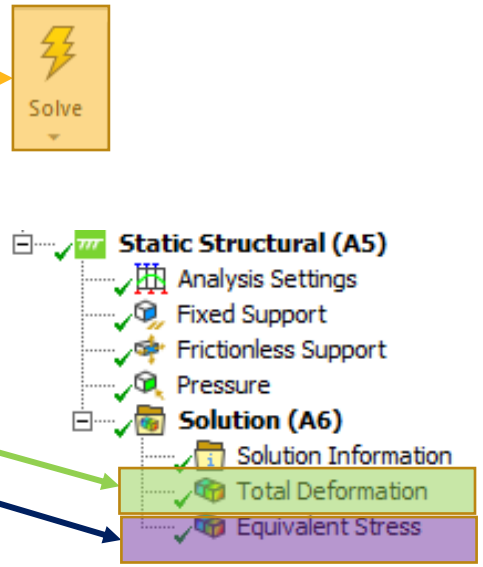
Scope	
Scoping Method	Named Selection
Named Selection	NSInsideFaces
Definition	
ID (Beta)	96
Type	Pressure
Define By	Normal To
Applied By	Surface Effect
Loaded Area	Deformed
Magnitude	Tabular Data
Suppressed	No
Tabular Data	
Independent Variable	Time

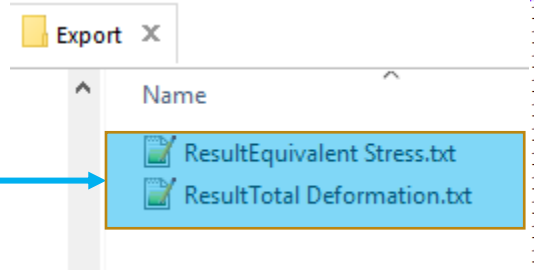
Tabular Data

	Steps	Time [s]	Pressure [MPa]
1	1	0.	0.
2	1	1.	15.
*			

Let’s dive into the script: solve and postprocess

```
# Solve model
model.Solve()

# Add results
solution = analysis.Solution
solution.AddTotalDeformation()
solution.AddEquivalentStress()
solution.EvaluateAllResults()

# Export result values to a text file
filePath=r"D:\Export\Result"
fileExtension=r".txt"
results =  solution.GetChildren(DataModelObjectCategory.Result,True)
for result in results:
    fileName = str(result.Name)
    result.ExportToTextFile(True,filePath+fileName+fileExtension)
```






```
Node Number Equivalent (von-Mises) Stress (MPa)
11136   4.9244
11137   5.2469
11138   4.3853
11139   5.9214
11140   5.5261
11141   4.1676
11142   2.8028
11143   3.7985
11144   20.445
11145   22.654
11146   27.851
11147   36.228
```

Navigating the tree and selecting objects

Navigating the Mechanical tree

- All objects that exist only once in the tree have a direct access point through ExtAPI.DataModel.Project.Model

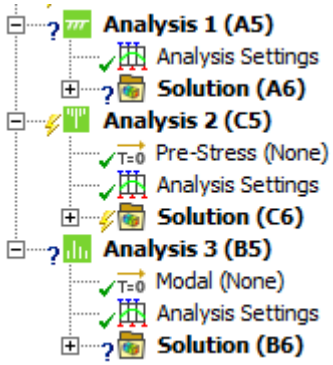
- For objects that can exist more than once, there can be two methods :

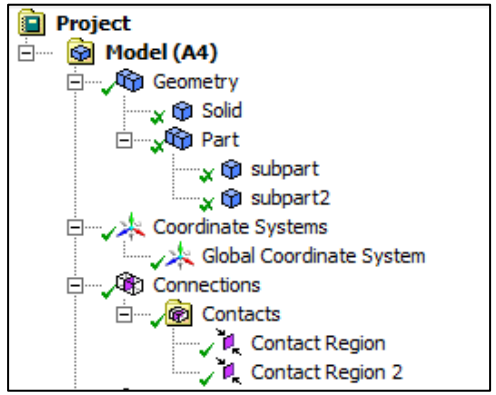
  - Analyses are stored in a list:

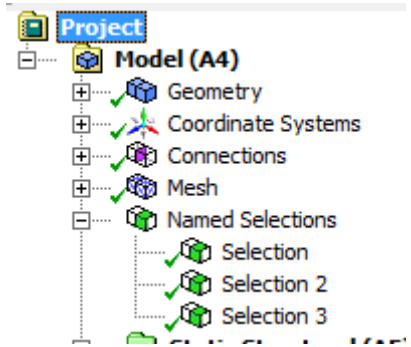
```
ExtAPI.DataModel.Project.Model.Analyses[0].Name = 'My new analysis name'
```

  - Other objects: Children objects

```
ExtAPI.DataModel.Project.Model.Geometry.Children[1].Children[0].Name -> subpart
ExtAPI.DataModel.Project.Model.Connections.Children[0]
ns = [i for i in ExtAPI.DataModel.Project.Model.NamedSelections.Children if i.Name=="Selection"][0]
Connections.GetChildren(DataModelObjectCategory.ContactRegion, True)[0]
```



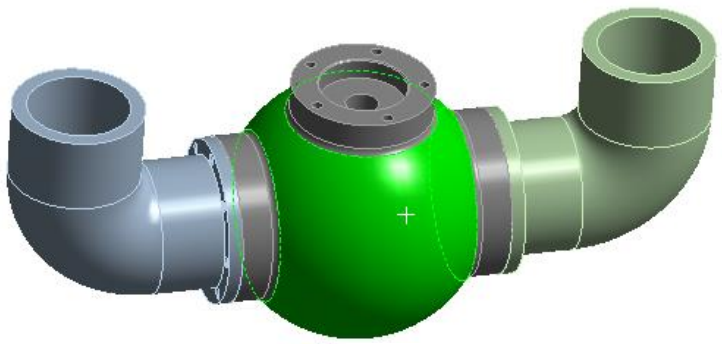




Selecting entities

- Get IDs of entities currently selected in the UI:

```
ExtAPI.SelectionManager.CurrentSelection
```



```
>>> ExtAPI.SelectionManager.CurrentSelection

[34]
```

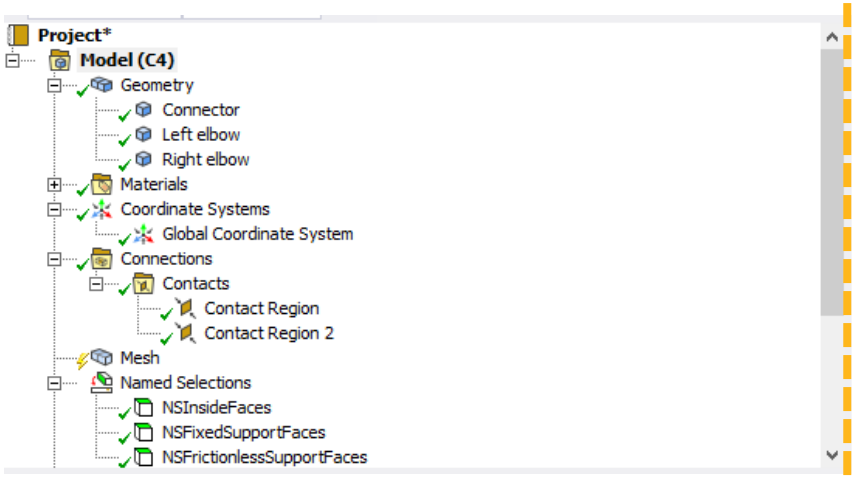
- Create a new selection and use it:

```
# create a new empty selection
tempSel = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
# provide list of Ids of entities to select
tempSel.Ids = [34]
# create pressure condition
pressure = ExtAPI.DataModel.Project.Model.Analyses[0].AddPressure()
# assign location
pressure.Location=tempSel
```

Automate Preprocessing

Geometry and geodata

ExtAPI.DataModel.Project.Model.Geometry



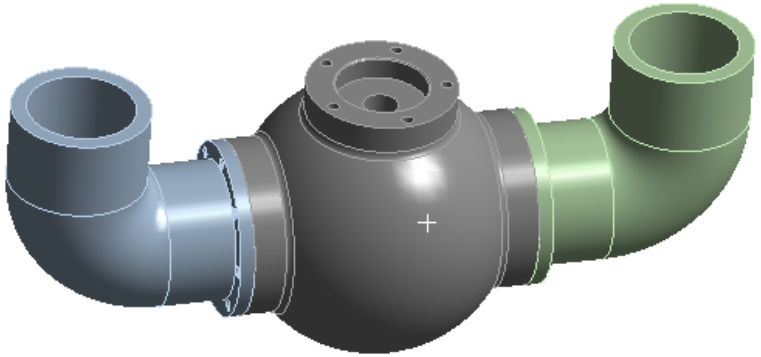
Details of "Connector"

Graphics Properties	
Definition	
Suppressed	No
ID (Beta)	19
Stiffness Behavior	Flexible
Coordinate System	Default Coordinate System
Reference Temperature	By Environment
Treatment	None
Reference Frame	Lagrangian
Material	
Assignment	Structural Steel
Nonlinear Effects	Yes
Thermal Strain Effects	Yes
Bounding Box	
Properties	
Statistics	
CAD Attributes	
PartTolerance:	0.00000001
Color:143.143.175	

```
>>> geom = ExtAPI.DataModel.Project.Model.Geometry
>>> body = geom.Children[0].Children[0]
>>> body.Name
'Connector'
>>> body.StiffnessBehavior
Flexible
>>>
```

ExtAPI.DataModel.Geodata

Model
13/04/2021 15:37

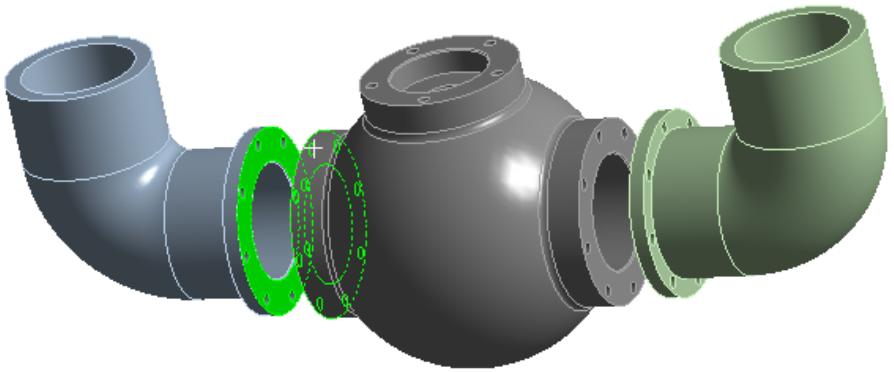


```
>>> geodata = ExtAPI.DataModel.GeoData
>>> geobody = geodata.Assemblies[0].Parts[0].Bodies[0]
>>> geobody.Name
'Connector'
>>> geobody.Volume
0.0127583884848715
>>> geobody.Faces.Count
61
```

Connections

- Create a new contact region and define its type:

```
connections = ExtAPI.DataModel.Project.Model.Connections
contact = connections.AddContactRegion()
contact.ContactType=ContactType.NoSeparation
```



- Specify contact and target faces (with manual selection of faces):

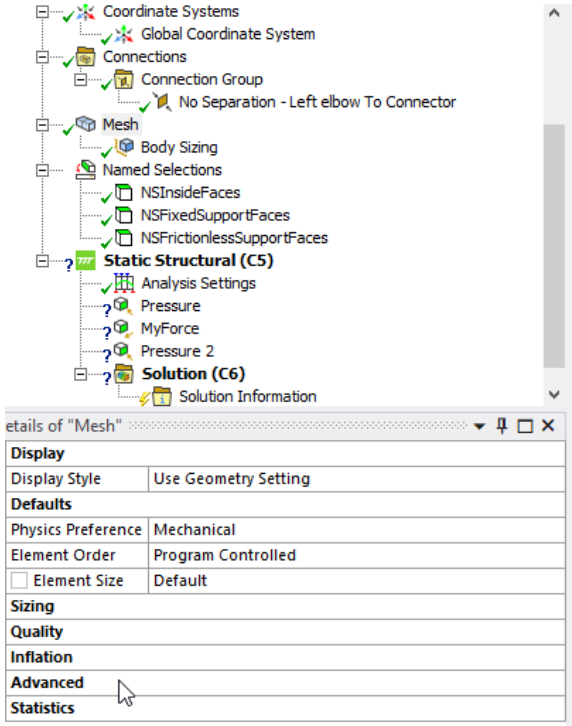
```
contact.SourceLocation = ExtAPI.SelectionManager.CurrentSelection
contact.TargetLocation = ExtAPI.SelectionManager.CurrentSelection
```

Details of "No Separation - Left elbow To Connector"

Scope	
Scoping Method	Geometry Selection
Contact	1 Face
Target	1 Face
Contact Bodies	Left elbow
Target Bodies	Connector
Protected	No
Definition	
Type	No Separation
Scope Mode	Manual
Behavior	Program Controlled
Trim Contact	Program Controlled
Suppressed	No
Advanced	

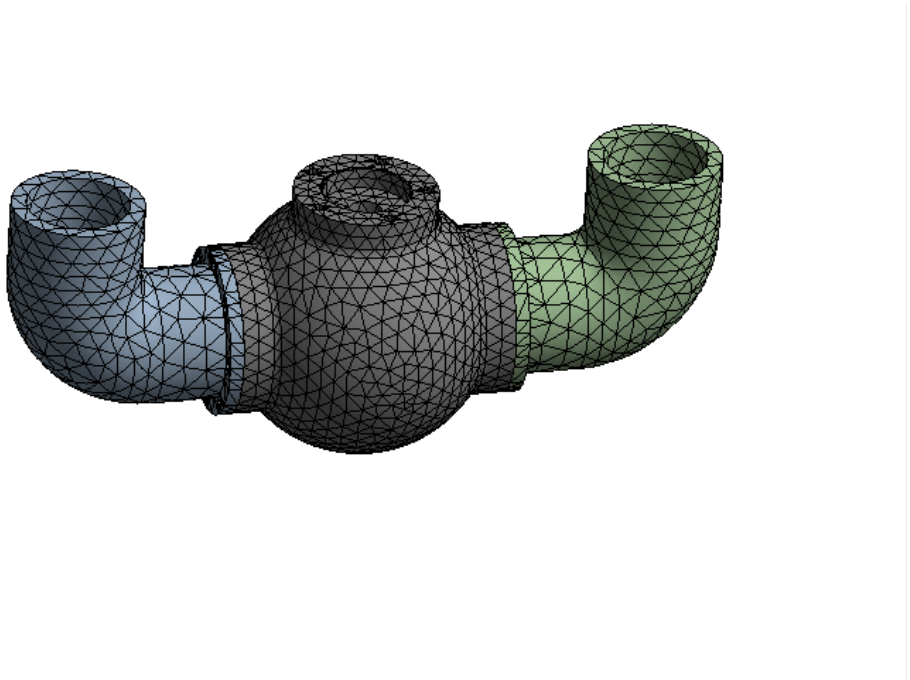
Mesh and mesh data

ExtAPI.DataModel.Project.Model.Mesh



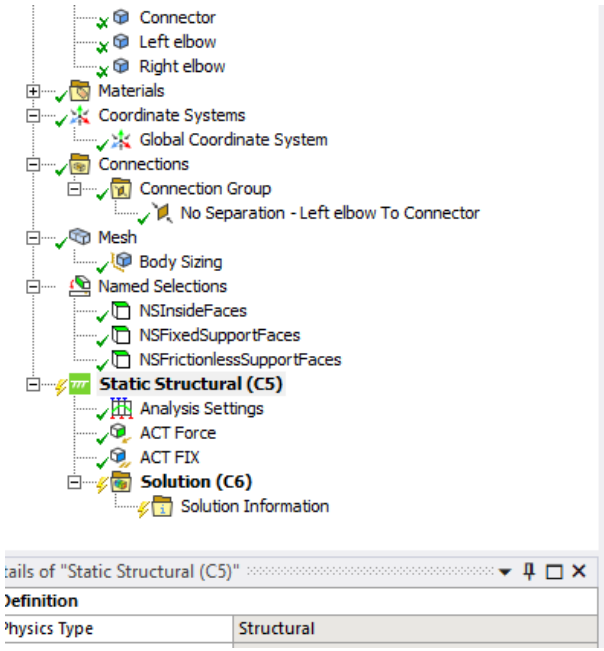

```
>>> mesh = ExtAPI.DataModel.Project.Model.Mesh
>>> sizing = mesh.AddSizing()
>>> sizing.ElementSize
'0 [mm]'
>>> sizing.ElementSize = Quantity('20 [mm]')

>>> body = ExtAPI.DataModel.GetObjectsByName("Connector")[0].GetGeoBody()
>>> tempSel = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
>>> tempSel.Ids = [body.Id]
>>> sizing.Location = tempSel
```

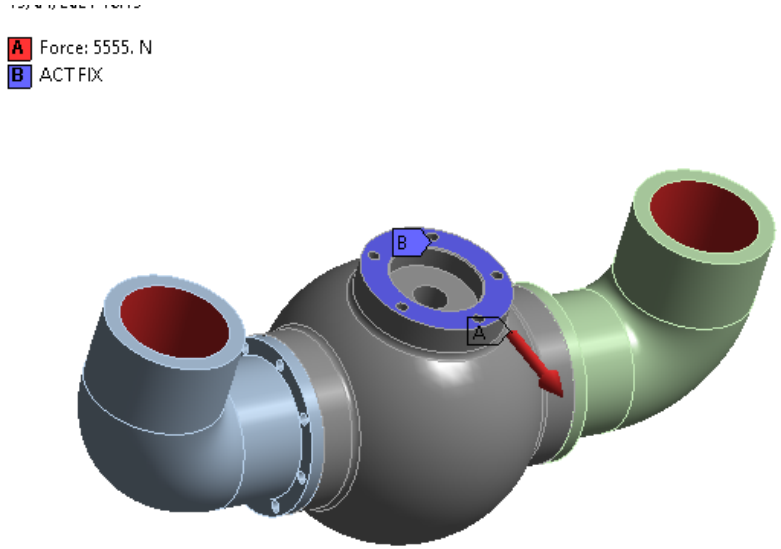
ExtAPI.DataModel.Project.Model.Analyses[0].MeshData



```
>>> meshData = ExtAPI.DataModel.Project.Model.Analyses[0].MeshData
>>> meshData.ElementCount
21833
>>> meshData.NodeCount
36615
>>> meshData.Elements[0].Centroid
[-0.16265137107062, -0.208395264572271, -0.342956119942903]
>>> meshData.Elements[0].CornerNodeIds
[51, 815, 933, 604]
```

Boundary conditions





Details of "ACT Force"

Scope	
Scoping Method	Named Selection
Named Selection	NSInsideFaces
Definition	
ID (Beta)	119
Type	Force
Define By	Components
Applied By	Surface Effect
Coordinate System	Global Coordinate System
X Component	0. N (ramped)
Y Component	-5555. N (ramped)
Z Component	0. N (ramped)
Suppressed	No

Details of "ACT FIX"

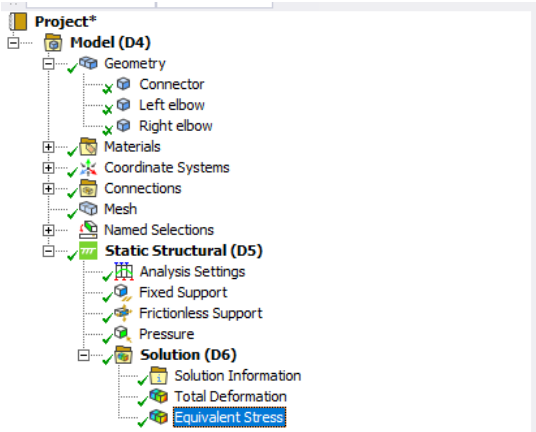
Scope	
Scoping Method	Named Selection
Named Selection	NSFixedSupportFaces
Definition	
ID (Beta)	121
Type	Fixed Support
Suppressed	No

```
FY = -5555
analysis = ExtAPI.DataModel.Project.Model.Analyses[0]
F = analysis.AddForce()
F.DefineBy = LoadDefineBy.Components
F.YComponent.Output.SetDiscreteValue(0, Quantity(FY, "N"))
F.Location = ExtAPI.DataModel.GetObjectsByName("NSInsideFaces")[0]
F.Name = "ACT Force"
G = analysis.AddFixedSupport()
G.Location = ExtAPI.DataModel.GetObjectsByName("NSFixedSupportFaces")[0]
G.Name = "ACT FIX"
```

Automate postprocessing

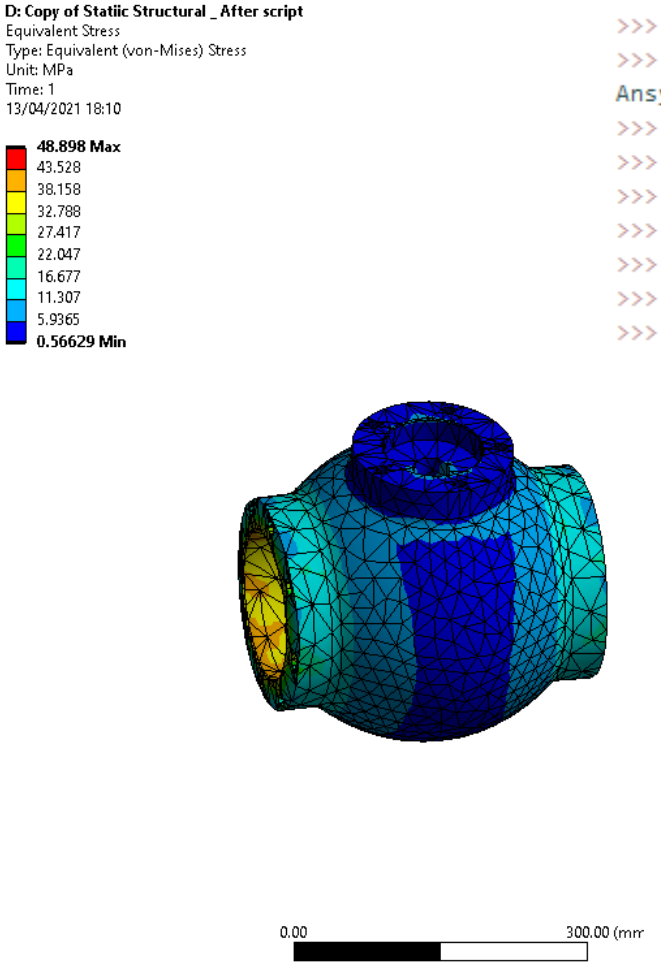
Inserting results

- Create standard results:



Details of "Equivalent Stress"

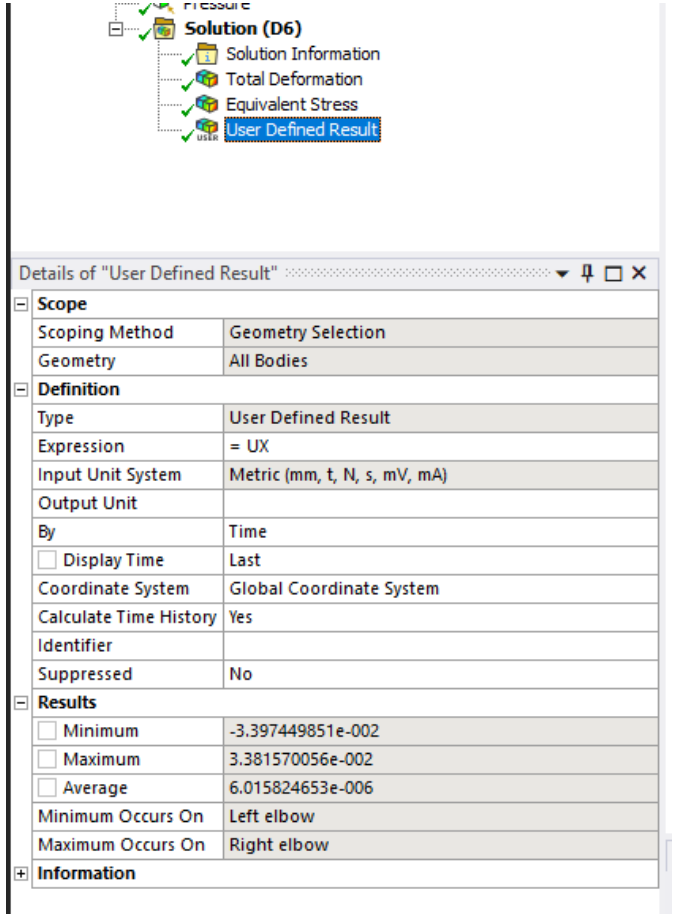
Scope	
Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Type	Equivalent (von-Mises) Stress
By	Time
Display Time	Last
Calculate Time History	Yes
Identifier	
Suppressed	No
Integration Point Results	
Display Option	Averaged
Average Across Bodies	No
Results	
Minimum	0.5662877948 MPa
Maximum	48.89832556 MPa
Average	15.41198206 MPa



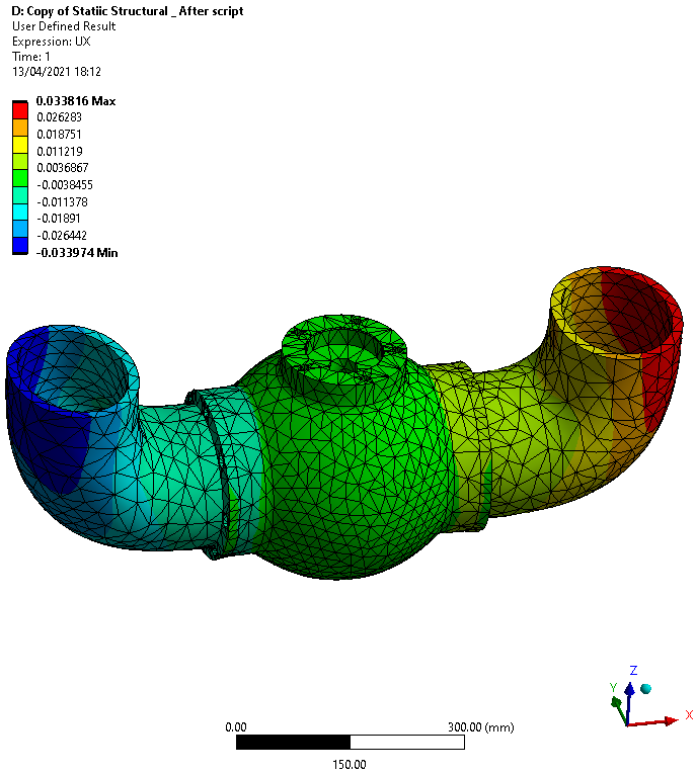

```
>>> solution = ExtAPI.DataModel.Project.Model.Analyses[0].Solution
>>> solution.AddTotalDeformation()
Ansys.ACT.Automation.Mechanical.Results.DeformationResults.TotalDeformation
>>> stressConnector = solution.AddEquivalentStress()
>>> body=ExtAPI.DataModel.GetObjectsByName("Connector")[0].GetGeoBody()
>>> tempSel = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
>>> tempSel.Ids = [body.Id]
>>> stressConnector.Location = tempSel
>>> solution.EvaluateAllResults()
>>>
```

Inserting results

- Create user-defined results:

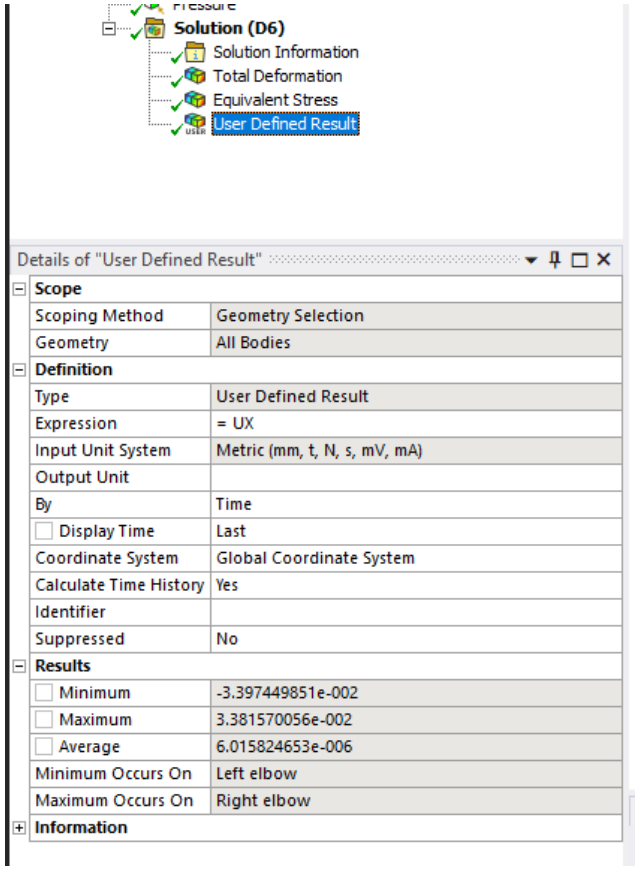


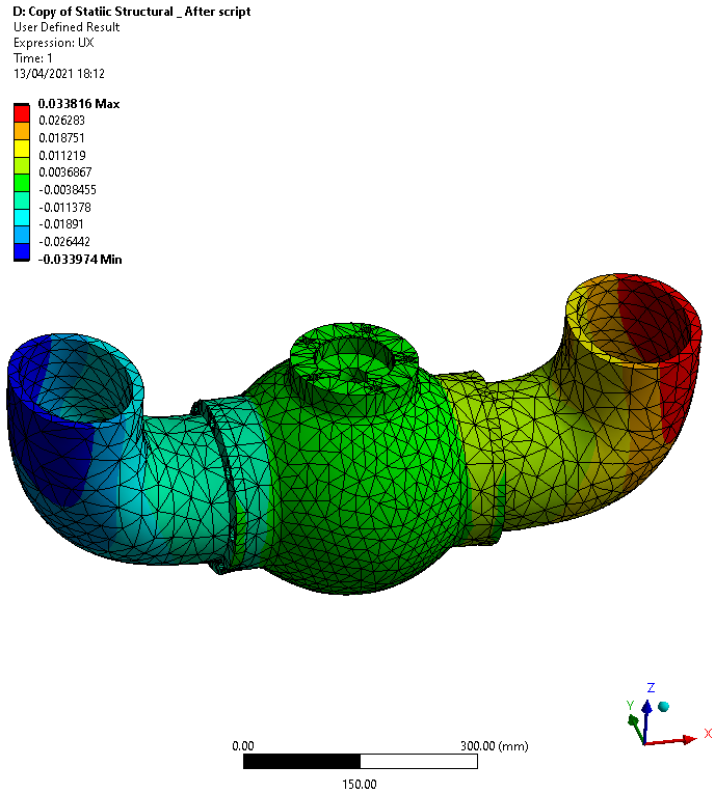
Scope	
Scoping Method	Geometry Selection
Geometry	All Bodies
Definition	
Type	User Defined Result
Expression	= UX
Input Unit System	Metric (mm, t, N, s, mV, mA)
Output Unit	
By	Time
Display Time	Last
Coordinate System	Global Coordinate System
Calculate Time History	Yes
Identifier	
Suppressed	No
Results	
Minimum	-3.397449851e-002
Maximum	3.381570056e-002
Average	6.015824653e-006
Minimum Occurs On	Left elbow
Maximum Occurs On	Right elbow
Information	




```
>>> UDR = solution.AddUserDefinedResult()
>>> UDR.Expression="UX"
>>> solution.EvaluateAllResults()
```

Accessing plot data






```
>>> UDR.PlotData
              Node        Values

0            11136   -2.3445E-05
1            11137    6.6036E-06
2            11138    4.6504E-05
...
26170        11133     0.0077966
26171        11134     0.0077826
26172        11135     0.0081679
>>>
```

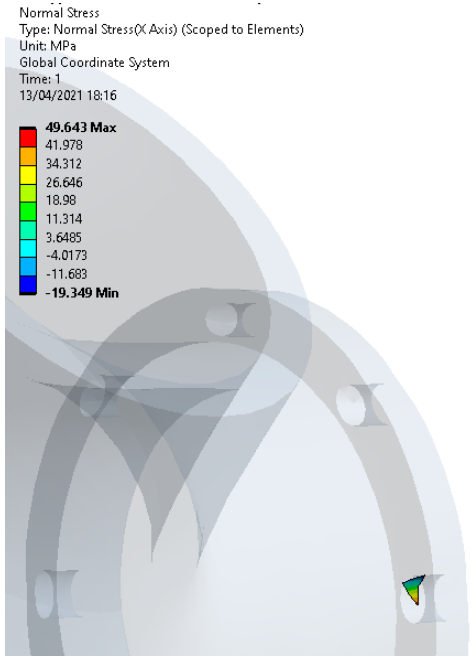
Access data from the result file

- Retrieve all components of stress tensor on one element:

```
model=ExtAPI.DataModel.Project.Model # refer to model
reader = model.Analyses[0].GetResultsData() # get results data of first analysis in the tree
StressResults = reader.GetResult('S') # obtain stress results
StressElement1 = StressResults.GetElementValues(1) # retrieve results on element n°1
```

- Retrieve stress values in one direction:

```
>>> model=ExtAPI.DataModel.Project.Model # refer to model
>>> reader = model.Analyses[0].GetResultsData() # get results data of first analysis in the tree
>>> StressResults = reader.GetResult('S') # obtain stress results
>>> StressResults.SelectComponents(["X"]) # select direction
>>> StressXElement1=StressResults.GetElementValues(1)
>>> StressXElement1
[-0.292112916707993, 27.8284664154053, -19.3488845825195, 49.6434173583984]
>>>
```



More more advanced and efficient postprocessing

Consider using DPF (Data Processing Framework)

.RST
.RTH
.d3plot

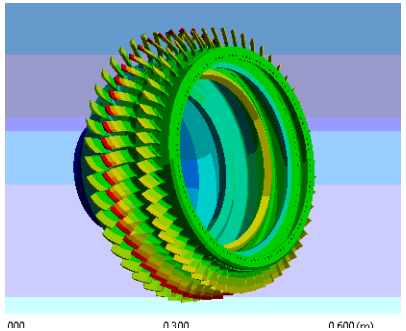
.cas
.dat
.vtk
.hdf5

Access Simulation Data

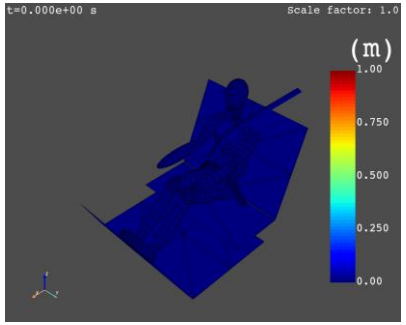
Manipulate/ Transform Simulation Data

Visualize/ Export Output **Field**

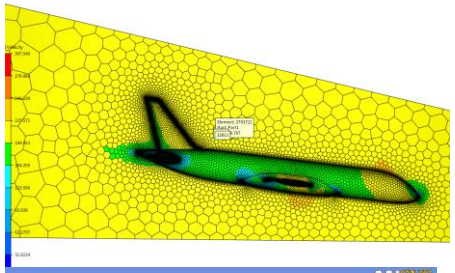
MAPDL



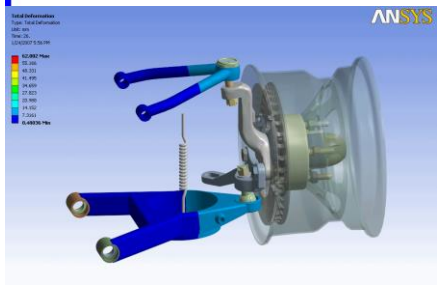
LSDyna



Fluids



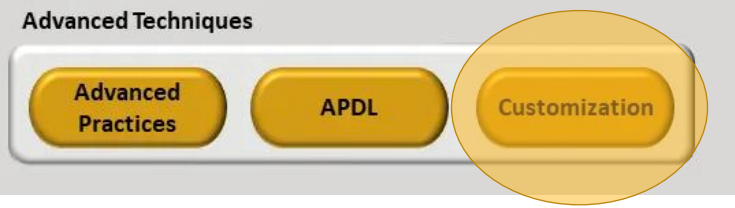
Motion

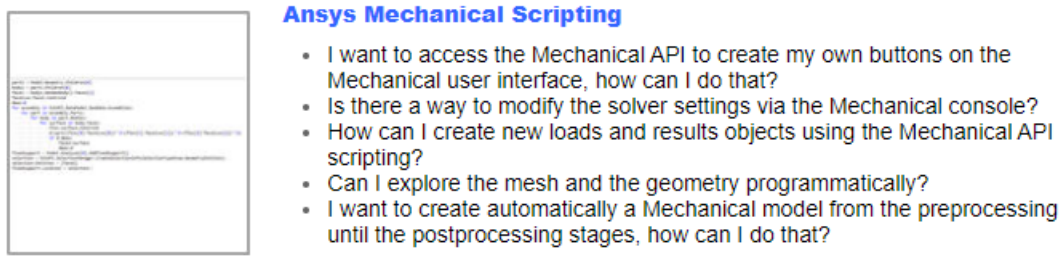


Useful resources

Training material

- On the Ansys Learning Hub

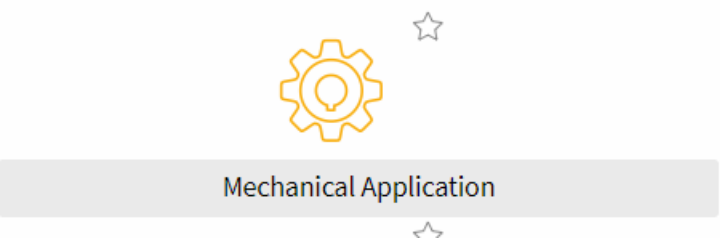


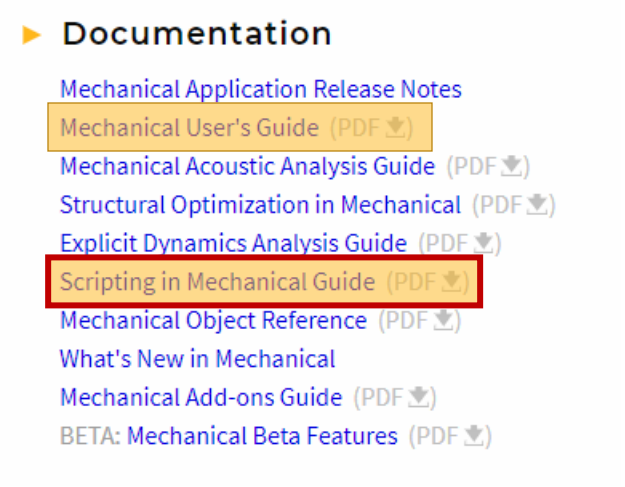


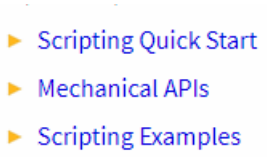
Download complete course, Watch previous recordings, Enroll in an upcoming class

Documentation

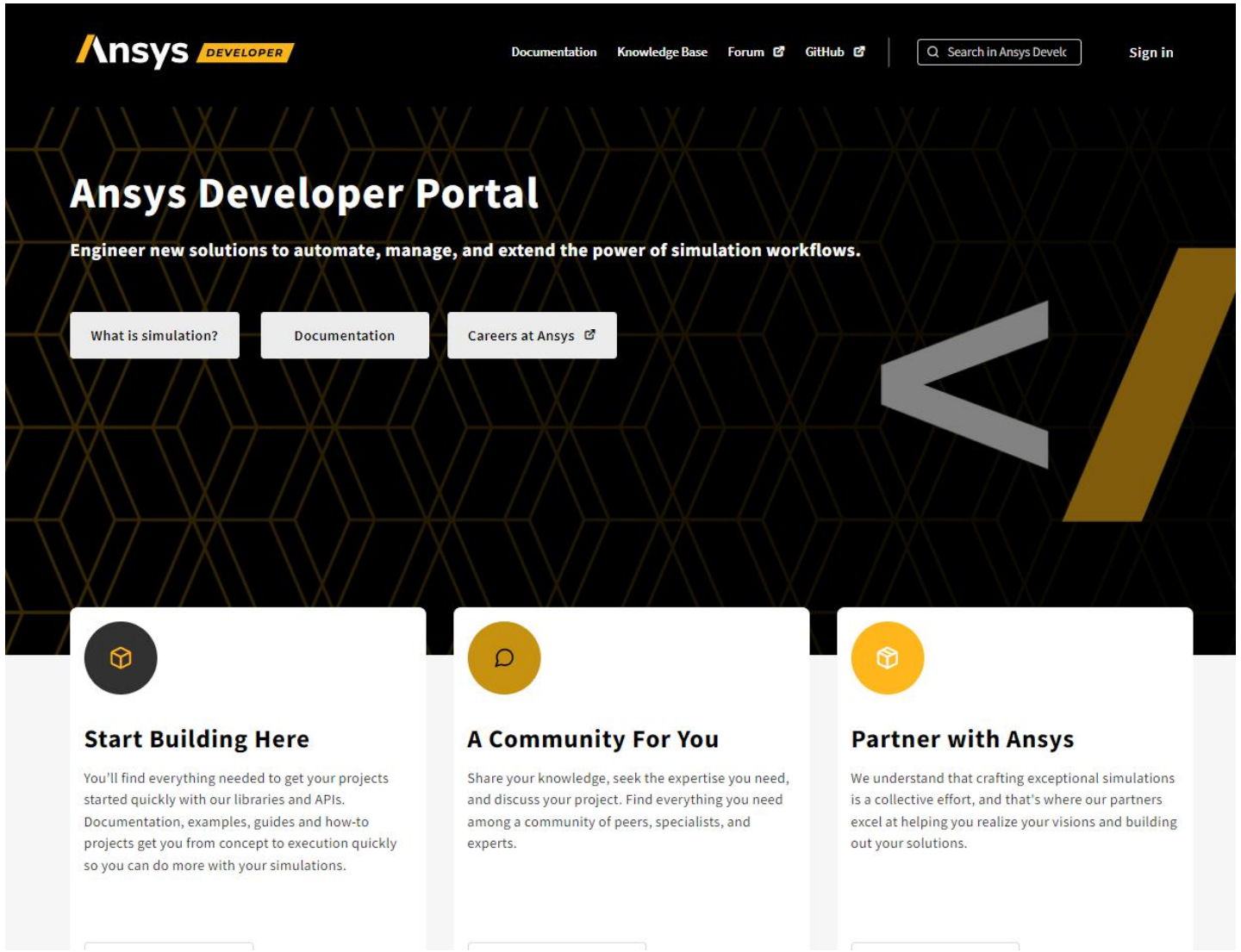
- In the Ansys Help





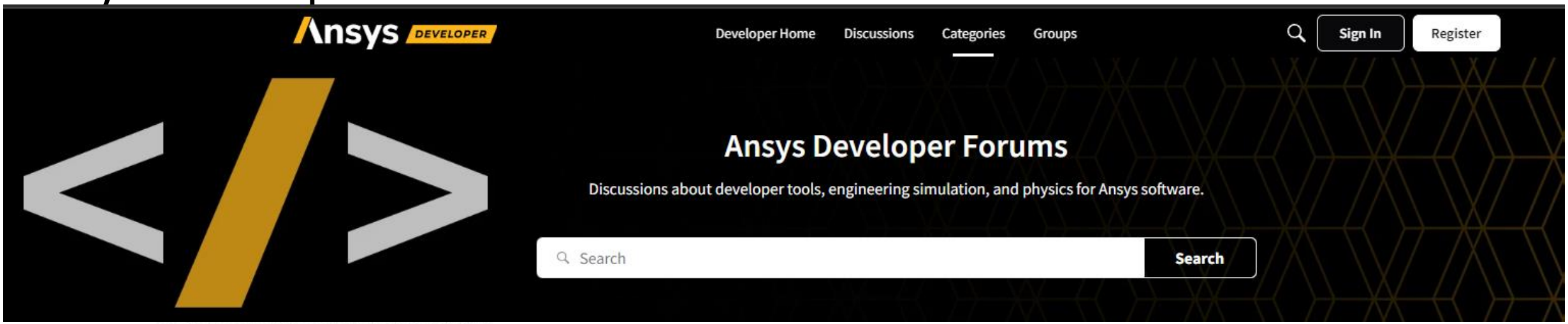


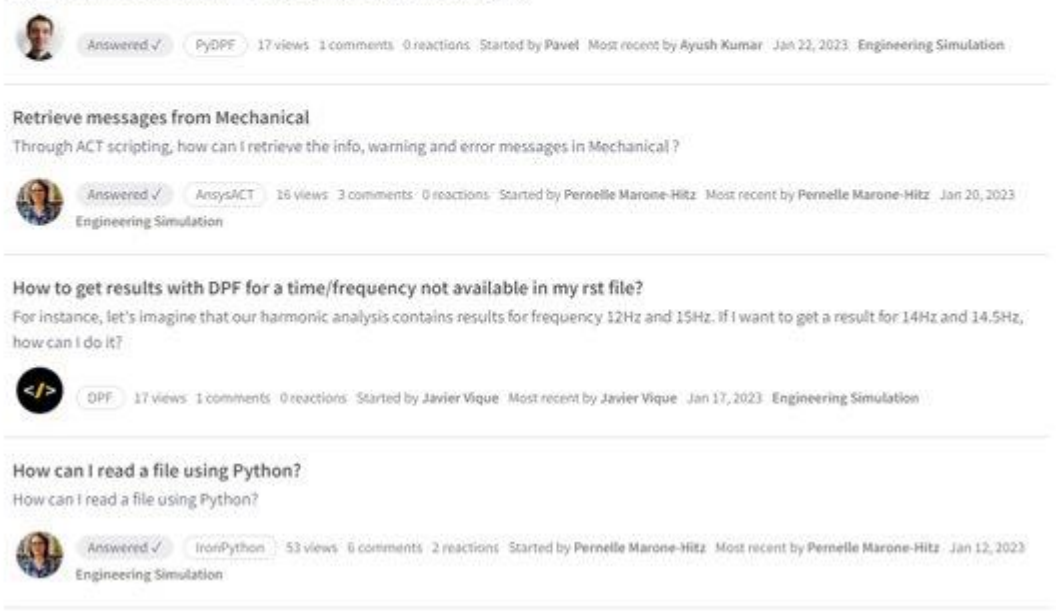
Ansys Developer Portal



https://developer.ansys.com

Ansys Developer Forum





https://discuss.ansys.com/

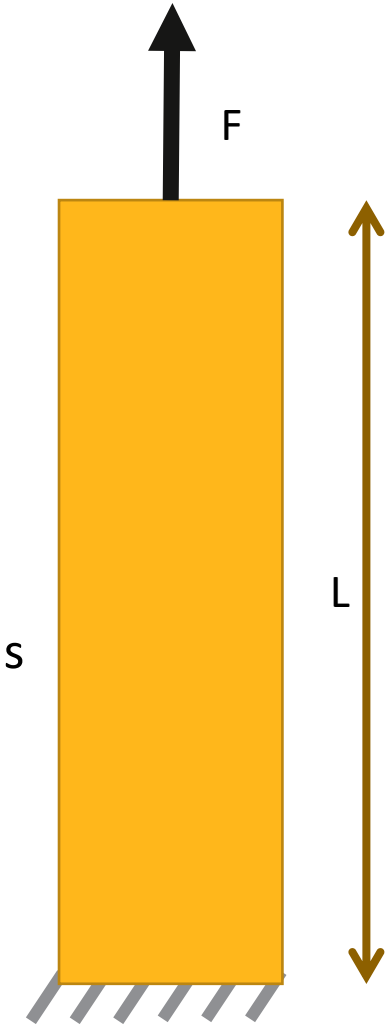
Automation cheat slides

Yes, we'll share this slide deck

Automation challenge

Automation challenge: Tensile pull test on cylinder

Challenge: The model must be entirely set up, run and post processed without using Mechanical's GUI. Only the scripting console may be used



To get started import a cylinder shape into Mechanical

Using the console:

- Apply a pull force of 1N on one of the cylinder face
- Apply a fixed support to the opposite face
- Solve the model
- Extract the maximum displacement in the pull test direction
- From this result calculate the young's modulus and compare it with the material's young's modulus

Reminder

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

Yes, we'll send a slide deck with the solution

Ansys