

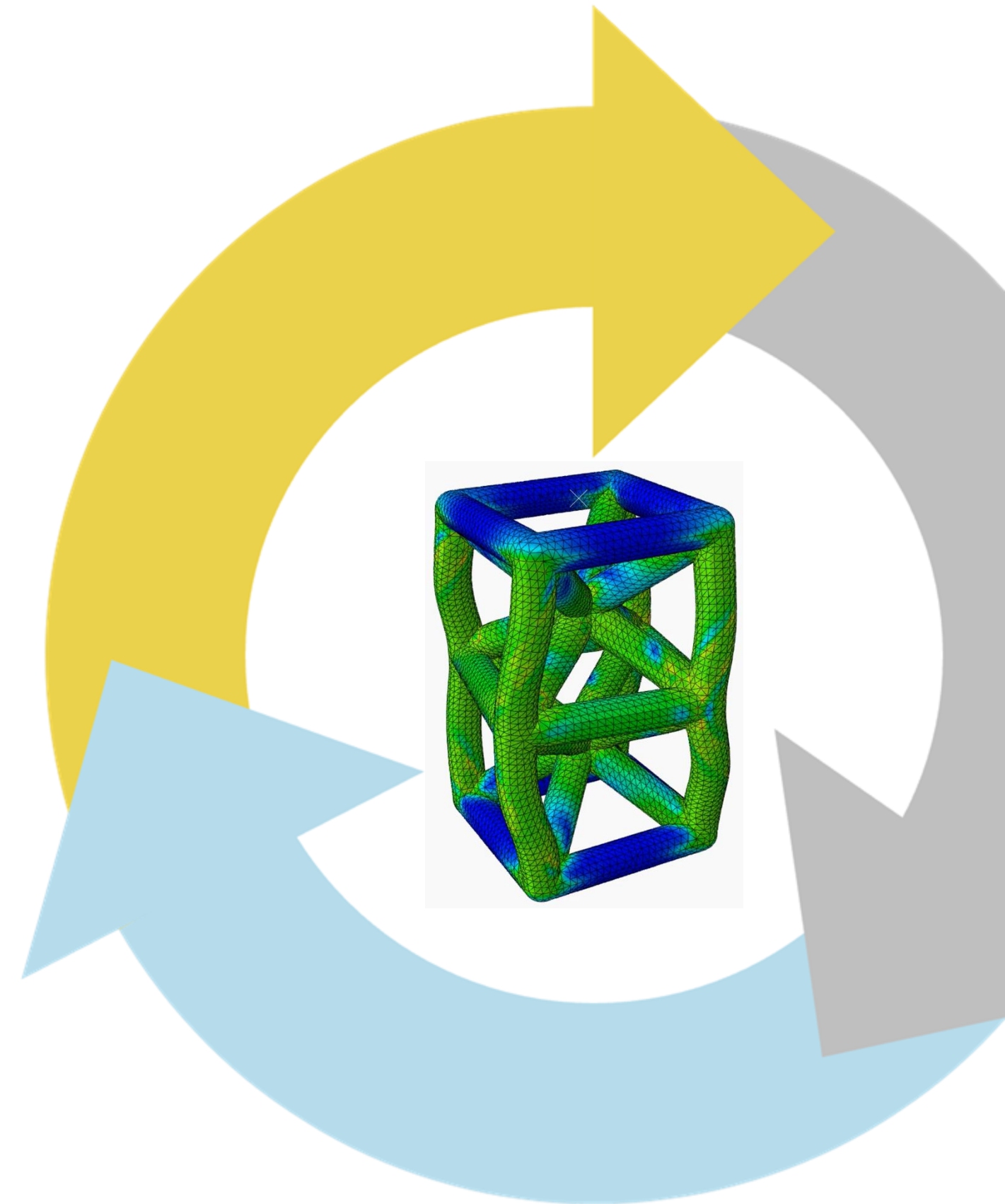
Force-Displacement Curve

For a 1x2x1 SC-BCC Column

With ABAQUS and Python Graphing

Presented By: Alec Perkins

working under Dr. Patrick Ziemke



Contents

01 Units and Material

02 Parts and Assembly

03 Boundary Conditions

04 Step and Tracked Data

05 Mesh and Job

06 Results

Units and Material



Intended Units

Units Used:
Mass: kg
Length: Millimeters Stress: MPa - Newton*m-2 Force: Newtons
Time: Seconds



Material - Titanium Ti-6Al-4V

Material Property:
Titanium Ti-6Al-4V (Grade 5), Annealed:
Data Used:
Density: 4.43e-6 kg*mm-3
Elastic: Youngs = 113.8e3 MPa, 0.342
Plastic: Yield = 880 MPa, Plastic Strain = 0, Ultimate = 950 MPa, Ultimate Strain = 0.14

Parts and Assembly

Part Descriptions

Simple Cubic Body Centered Cubic, Unit Cell:

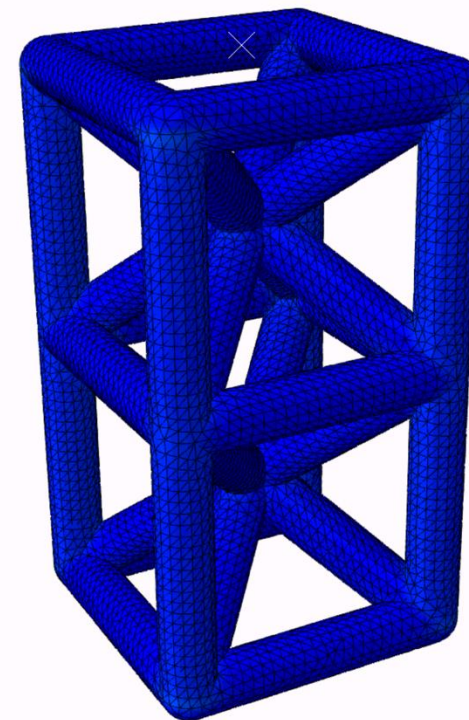
Strut Part Dimension Lengths:

- Horizontal Strut Lengths: 100mm
- Total Diagonal Strut Length: 173.21mm
- Strut Radius: 10mm

Rounded Corner Part:

- Sphere Diameter: 10mm

SCBCC 1 x 2 Column Assembly



Assembly Instructions:

- Linear Pattern horizontal and vertical struts
- Rotate Diagonal Strut about the vertical axis
- Translate Sphere Part to one corner coincident with a horizontal strut end center
- Linear Pattern Sphere Part to all corners of unit cell
- Merge all Instances into one SCBCC Unit Cell Part
- Linear Pattern and Merge the new SCBCC Unit Cell into a 1x2x1 Column Part

Boundary Conditions

BCs and Reference Points

Boundary Conditions:

- Displacement applied to top center RP with B.C. for top surface
- Top Center RP at (50, 200, 50)
- Fixed Base in all translational and rotational directions
- Fixed Top in directions U1, U2

Constraints

Displacement Constraint:

- Apply displacement to Top Center RP and Rigid Body Tie Constraint to the top surface set
- Ensure the BC applied allow the top surface of the column to move rigidly with the Top Center RP compression displacement

Step and Tracked Data



Time and Displacement Step

Step Parameters:

- Loading Step: Static, general
- Non-linear geometry
- 10,000 increments, 0.01s initial, 1e-8s smallest, 0.1s max



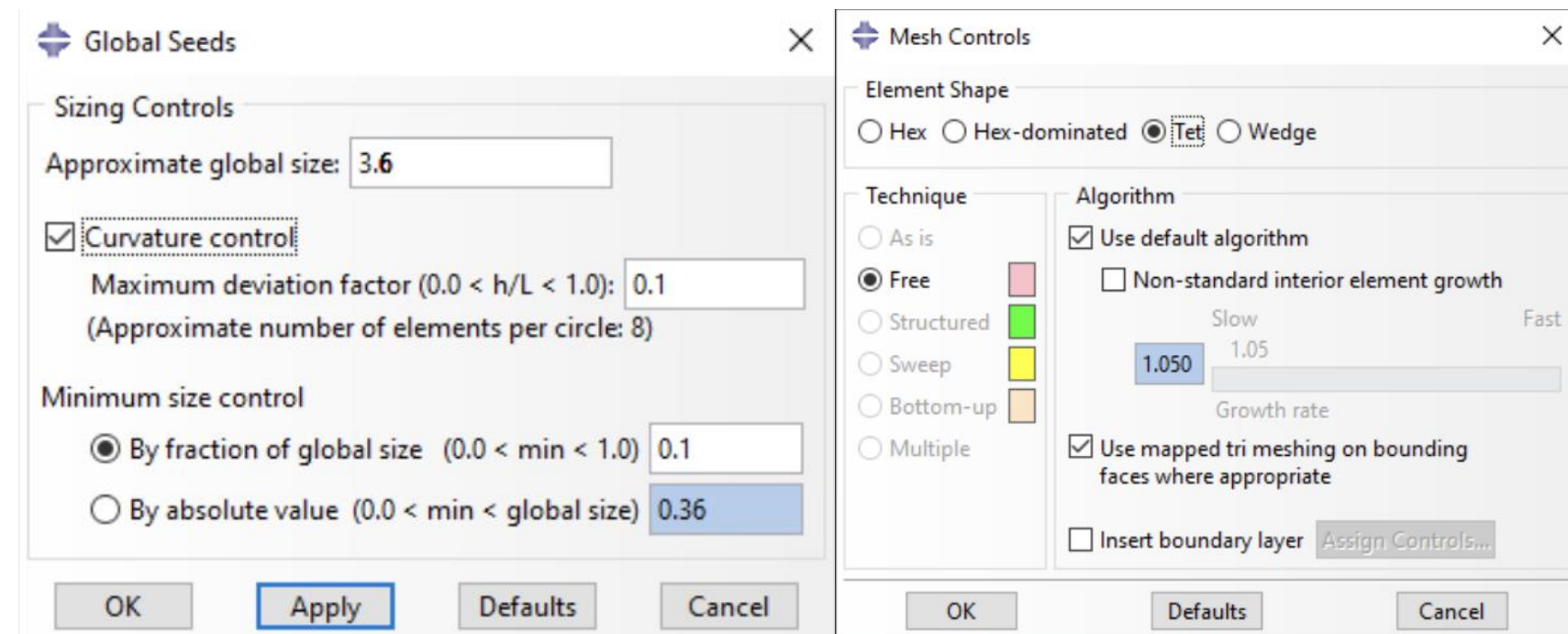
History

History Output:

- At load RP set: RF1, RF2, RF3, U1, U2, U3

Mesh and Job

Meshing and Seeding



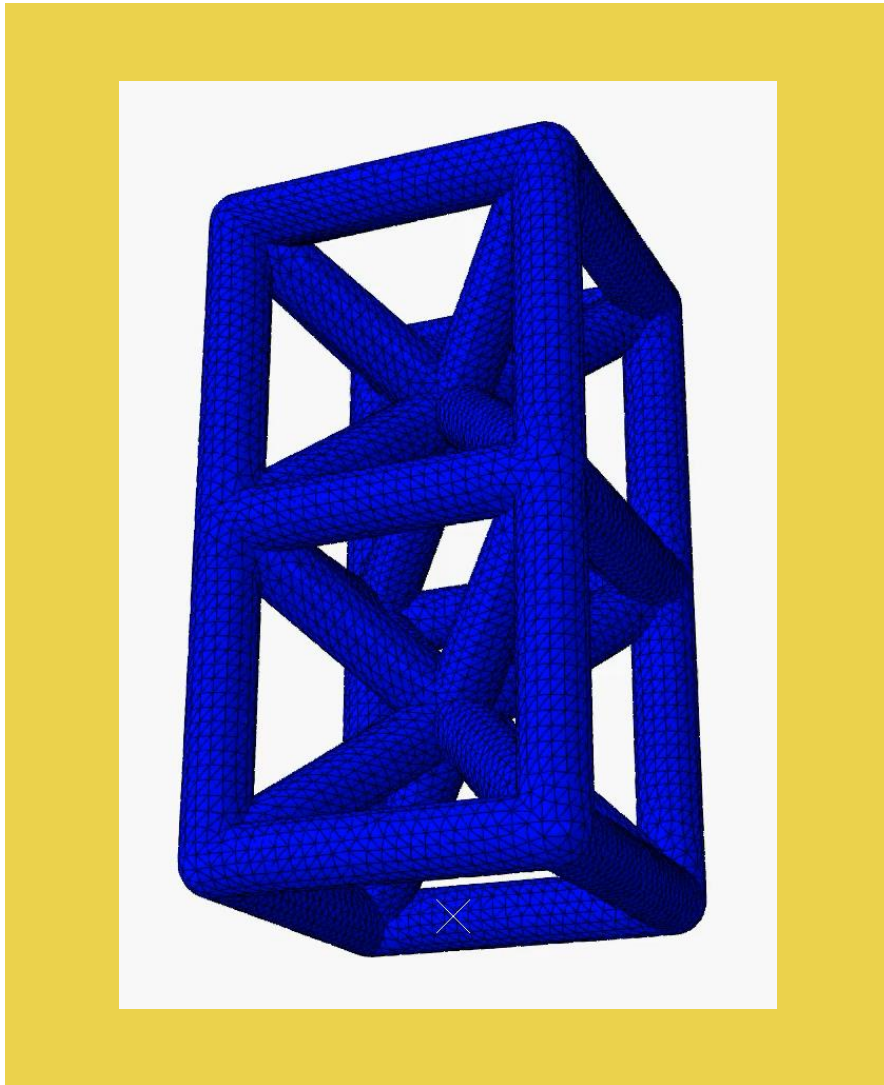
Creating the Job

Create a Job for a 30mm Vertical Displacement:

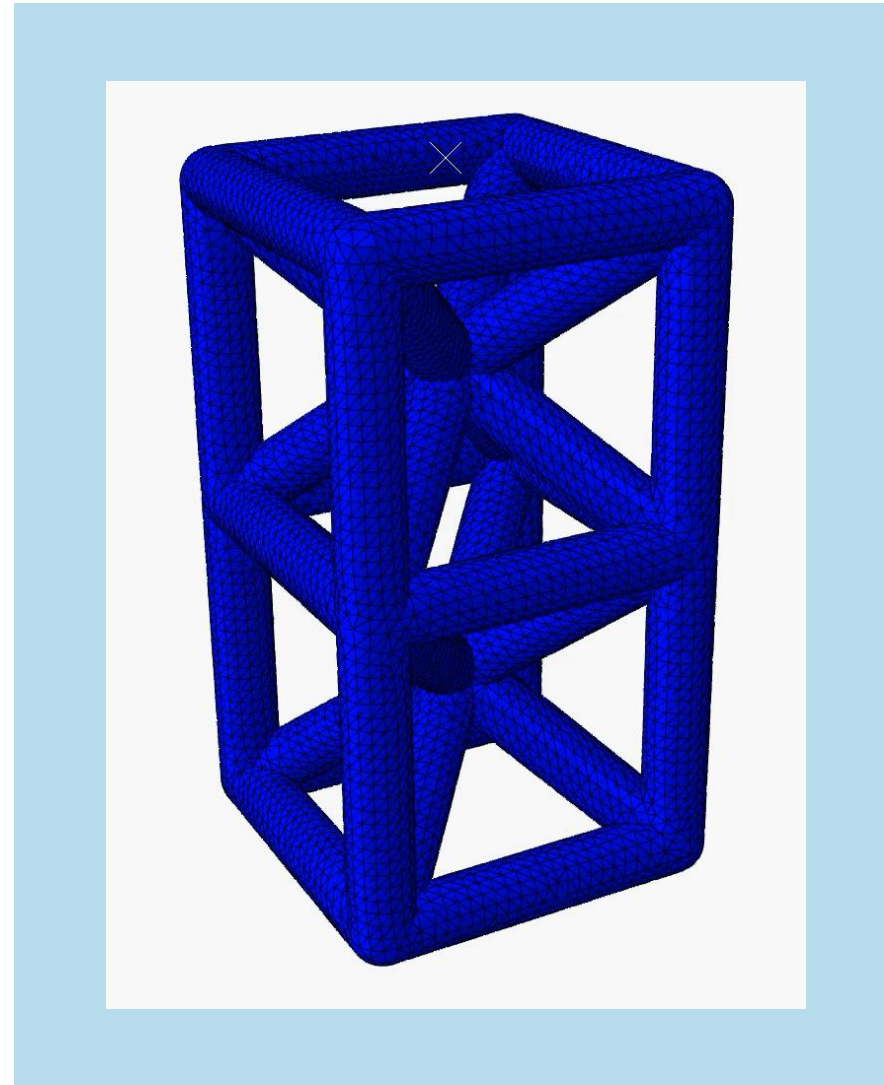
- Relocate as many processors that are available to run the Simulation (duration can vary depending on processor allocation)
- Monitor job for distortion errors or nonlinear warnings

Watch Videos of 30mm Displacement Results:

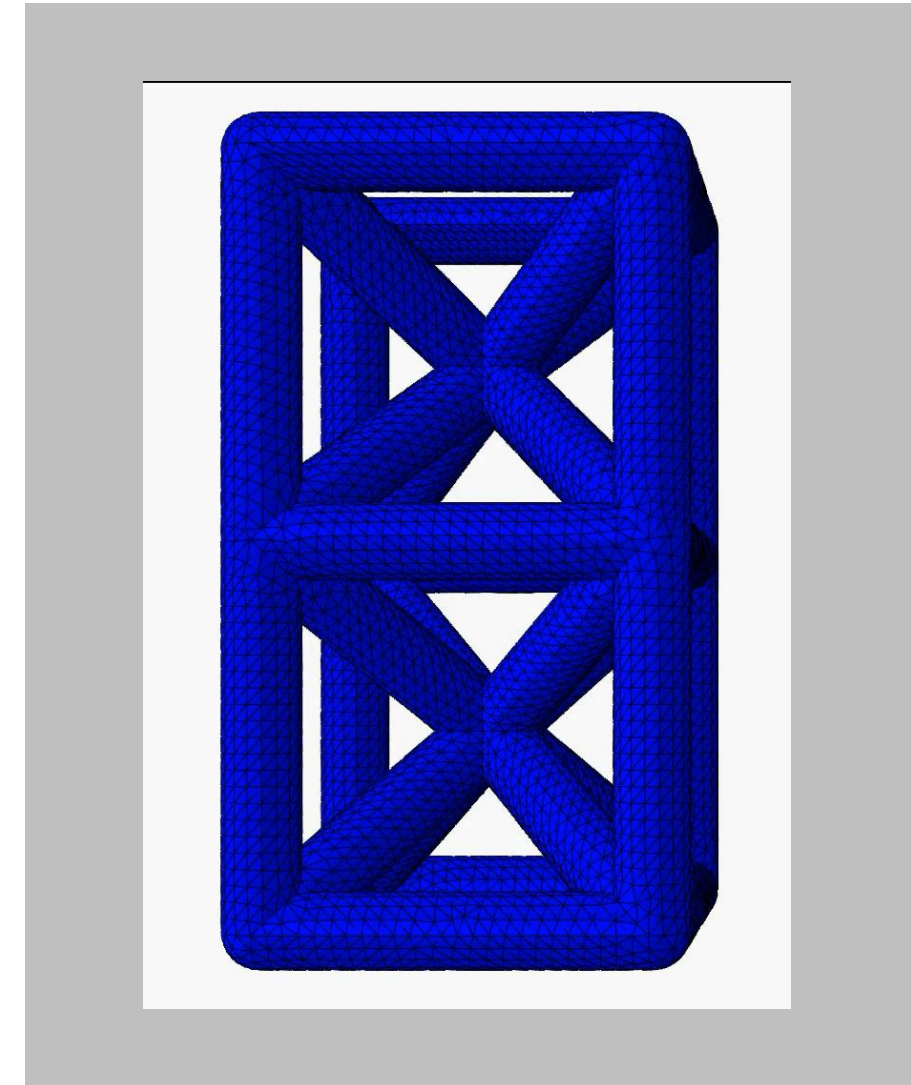
Results



Bottom ISO View



Top ISO View



XY- Plane Perspective

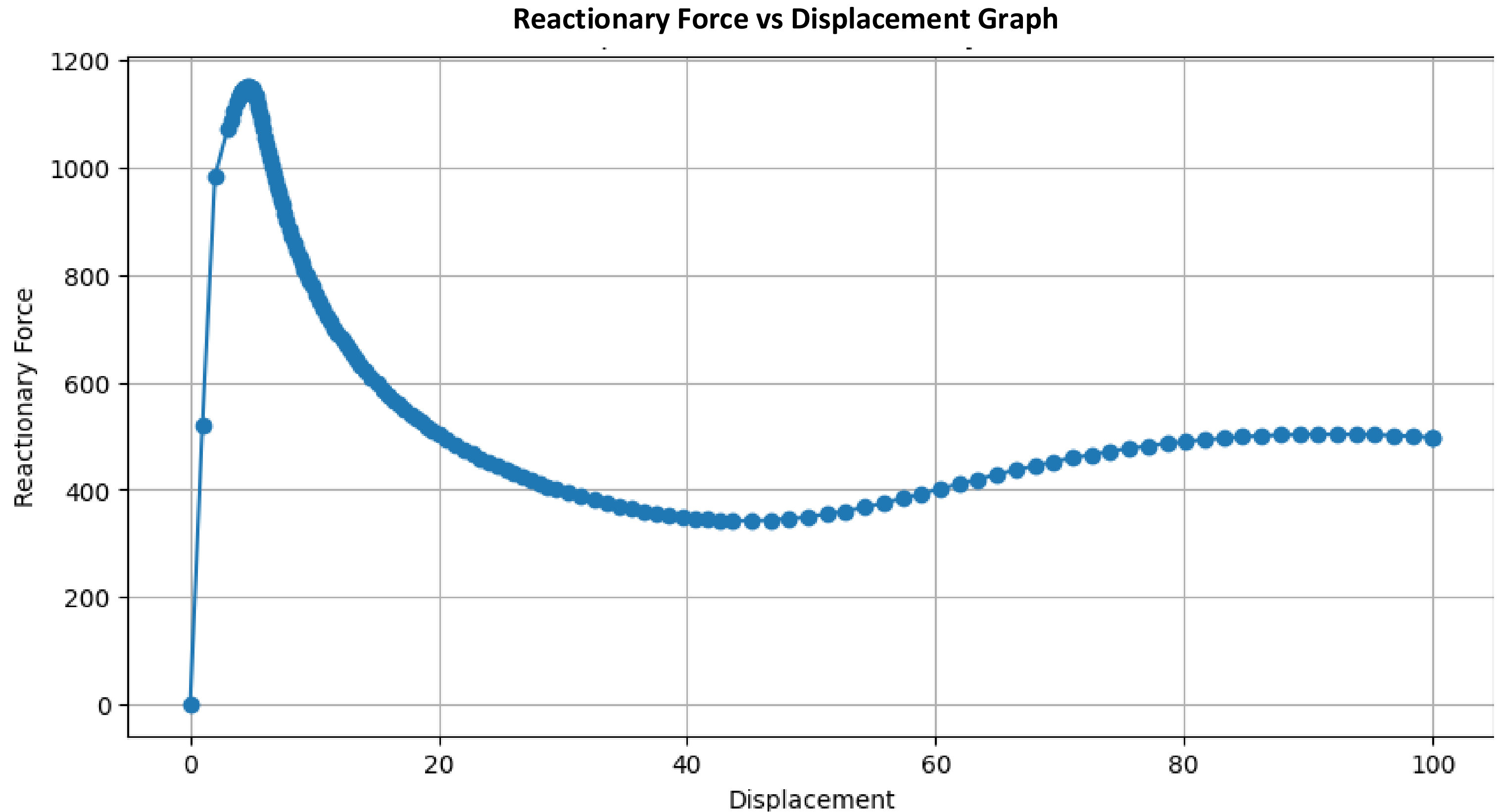
Example Reactionary Force vs. Displacement Data:

	main.py	data.csv	
			+ ↕ 🖼
1	Displacement,ReactionaryForce		
2	,0,0		
3	,1,521.149		
4	,2,981.736		
5	,3,1071.02		
6	,3.25,1088.98		
7	,3.5,1105.58		
8	,3.75,1120.26		
9	,4,1132.66		
10	,4.01562,1133.36		
11	,4.03125,1134.05		
12	,4.04688,1134.73		
13	,4.0625,1135.4		
14	,4.08594,1136.39		
15	,4.12109,1137.83		
16	,4.15625,1139.2		
17	,4.19141,1140.5		
18	,4.22656,1141.73		
19	,4.2793,1143.43		
20	,4.3584,1145.62		
21	,4.47705,1148.04		
22	,4.5957,1149.34		
23	,4.71436,1149.45		
24	,4.83301,1148.26		
25	,4.95166,1145.73		
26	,5.07031,1141.9		
27	,5.1593,1137.99		
28	,5.24829,1132.57		
29	,5.33728,1125.73		
30	,5.42627,1117.96		
31	,5.51526,1109.67		
32	,5.60425,1101.16		
33	,5.69324,1092.48		
34	,5.78223,1083.65		
35	,5.91571,1070		
36	,6.04919,1056.11		
37	,6.18268,1042.22		
38	,6.31616,1028.5		
39	,6.44965,1015		
40	,6.58313,1001.85		
41	,6.71661,989.013		
42	,6.8501,976.513		
43	,6.98358,964.423		
44	,7.11707,952.674		
45	,7.25055,941.282		

Plot Data for Reactionary Force vs. Displacement with Python:

```
< > main.py data.csv + ↗ 📄
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Replace 'path_to_your_file.csv' with the actual path to your CSV file
5 csv_file_path = 'data.csv'
6
7 # Read the CSV file
8 data = pd.read_csv(csv_file_path)
9
10 # Assuming the CSV file has two columns: 'Displacement' and 'ReactionaryForce'
11 displacement = data['Displacement']
12 reactionary_force = data['ReactionaryForce']
13
14 # Basic data validation
15 if len(displacement) != 150 or len(reactionary_force) != 150:
16     raise ValueError("The CSV file must contain exactly 150 values for both displacement and reactionary force.")
17
18 # Plotting the data
19 plt.figure(figsize=(10, 5))
20 plt.plot(displacement, reactionary_force, marker='o')
21 plt.xlabel('Displacement')
22 plt.ylabel('Reactionary Force')
23 plt.title('Displacement vs Reactionary Force')
24 plt.grid(True)
25 plt.show()
26
27 # Print the first few rows of data for verification
28 print(data.head())
29
30 # Save the processed data to a new CSV file if needed
31 output_csv_path = 'processed_data.csv'
32 data.to_csv(output_csv_path, index=False)
33 print(f"Processed data saved to {output_csv_path}")
34
```

Resulting Graph

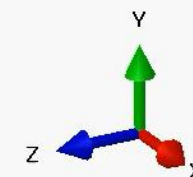
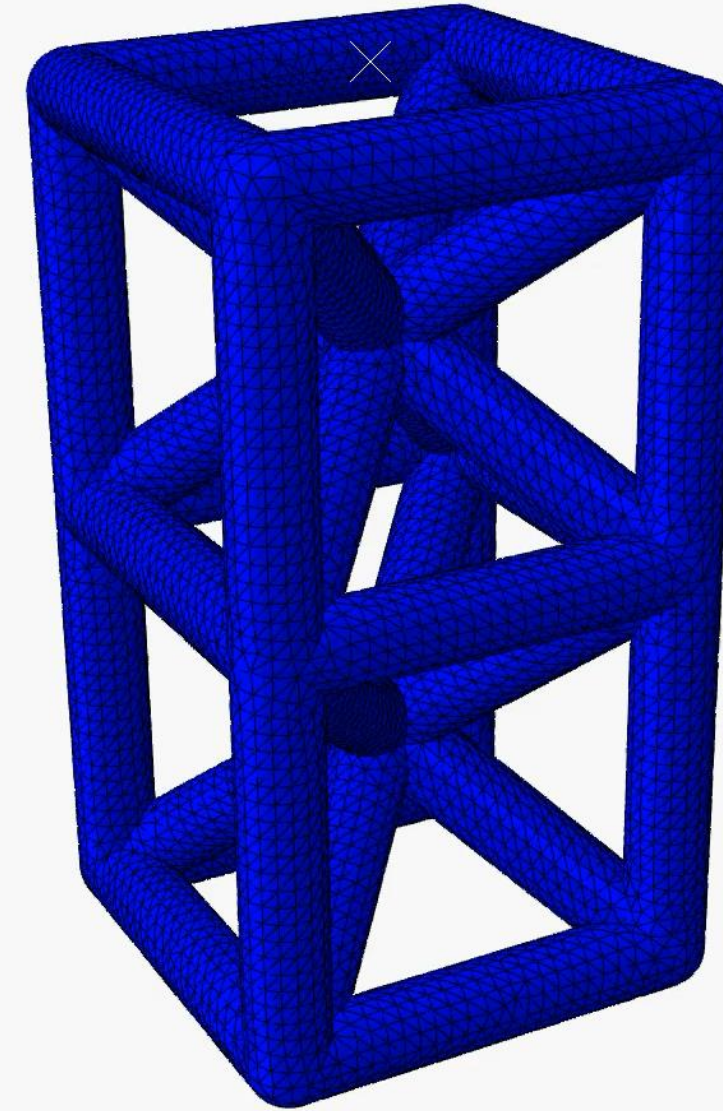
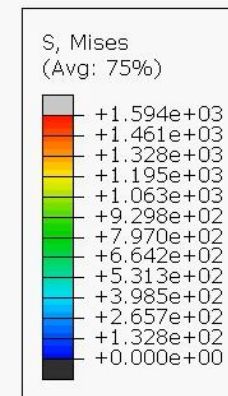


Credits

Thank you for your time!

Contact the following Email with any questions:

AlecPerkins@UCSB.edu



Displacing a 2x1 SCBCC Column -20mm in the y-axis and no rotation
ODB: NoRotate-20mm-yaxis-displacement-Copy1725052632.316.odb

Abaqus/Standard 2022 Mon May 27 07:42:30 Pacific Daylight Time 2024

Step: Displacement, -30mm displacement in the y-axis of the top surface of the 1x2 SC-BCC Column
Increment 0: Step Time = 0.000
Primary Var: S, Mises
Deformed Var: U Deformation Scale Factor: +1.000e+00