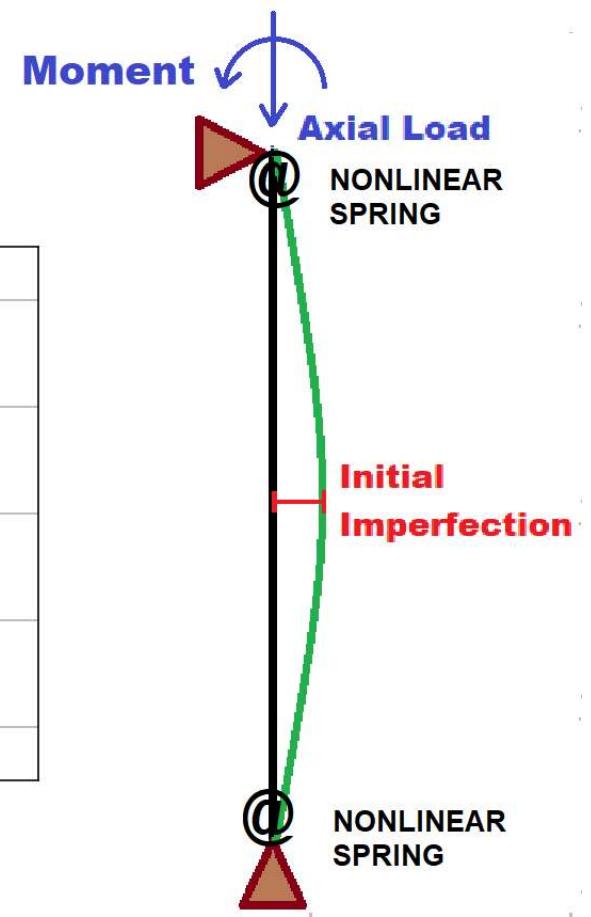
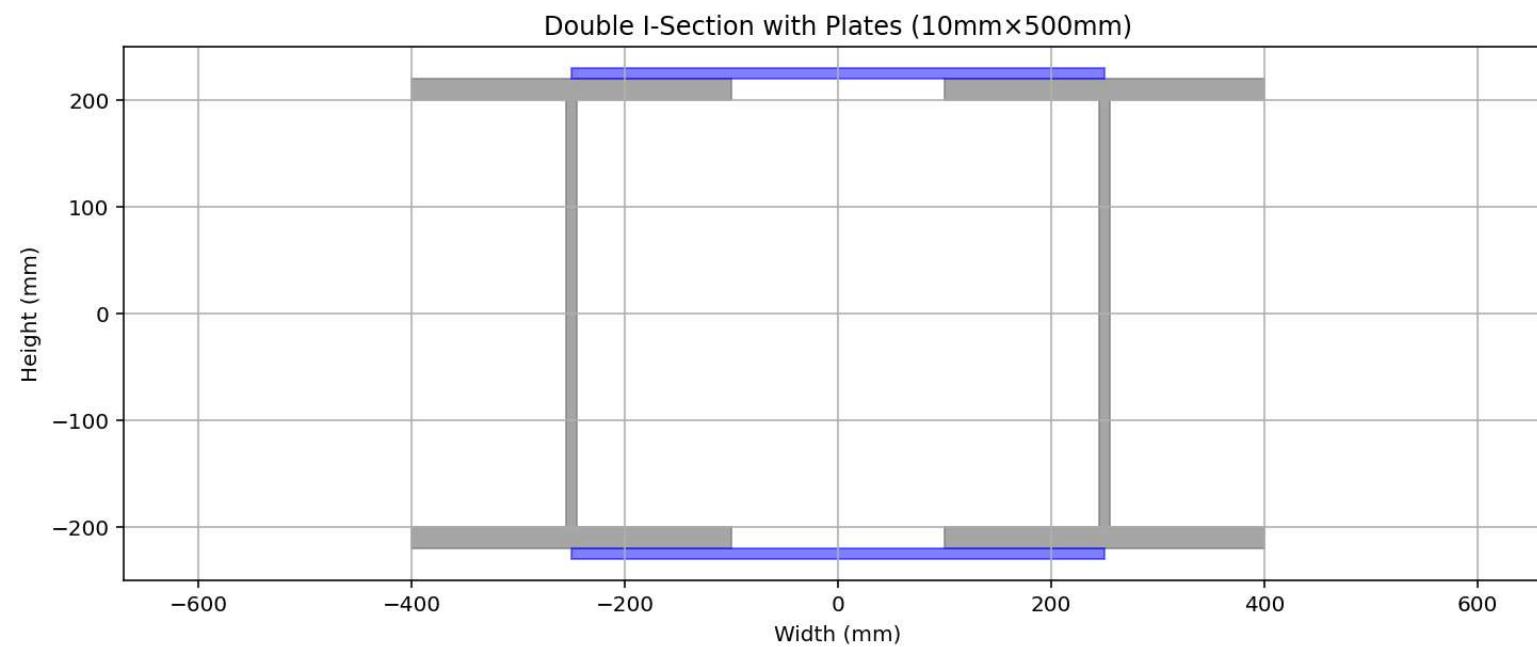


>> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<

LEVERAGING PARALLEL COMPUTING IN OPENSEES TO SIMULATE GEOMETRIC AND MATERIAL NONLINEARITY IN MULTI-MODE POST-BUCKLING OF SEMI-RIGID STEEL COLUMNS

WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)



WITHOUT PARALLEL COMPUTING

Spyder (Python 3.12)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Del\Desktop\OPENSEES_FILES\MULTI-MODE-POST_BUCKLING_STEEL_COLUMN_SEMI-RIGID_NONLINEAR.py

MULTI-MODE-POST_BU...RIGID_NONLINEAR.py X MULTI-MODE-POST_BU...ALLEL COMPUTING.py X

```
1 #####>>> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<
2 # INVESTIGATION OF MULTI-MODE POST-BUCKLING PHENOMENA IN SEMI-RIGID STEEL COLUMNS
3 # USING OPENSEES CONSIDERING THE GEOMETRIC AND MATERIAL PROPERTIES NONLINEARITY
4 #
5 #-----#
6 # IT MODELS A 2D INELASTIC BEAM-COLUMN WITH AN INITIAL IMPERFECTION (FOUR DIFFRENET SHAPES)
7 # AND APPLIES AN AXIAL COMPRESSIVE LOAD TO ANALYZE LARGE DISPLACEMENTS.
8 # 1. MODEL SETUP: A COLUMN OF LENGTH L IS DEFINED WITH N ELEMENTS, INCORPORATING A SMALL INITIAL
9 # IMPERFECTION (HALF-SINE WAVE).
10 # 2. NODES & BOUNDARY CONDITIONS: NODES ARE CREATED, WITH THE BOTTOM FIXED IN X, Y AND THE TOP FIXED IN
11 # X BUT FREE IN Y AND ROTATION.
12 # 3. ELEMENT DEFINITION: THE COLUMN IS MODELED USING ELASTIC BEAM-COLUMN ELEMENTS WITH COROTATIONAL
13 # TRANSFORMATION FOR GEOMETRIC NONLINEARITY.
14 # 4. LOAD APPLICATION: A STATIC AXIAL FORCE P IS APPLIED AT THE TOP NODE.
15 # 5. ANALYSIS SETUP: A DISPLACEMENTCONTROL INTEGRATOR IS USED TO INCREMENTALLY PUSH THE COLUMN DOWNWARDS.
16 # 6. NONLINEAR SOLVER: THE NEWTON METHOD IS USED WITH A NORMDISPINC TEST FOR CONVERGENCE.
17 # 7. ANALYSIS EXECUTION: THE LOOP PERFORMS INCREMENTAL LOADING STEPS, RECORDING AXIAL DISPLACEMENT,
18 # LATERAL DISPLACEMENT, AND AXIAL FORCE.
19 # 8. BUCKLING BEHAVIOR CAPTURE: LATERAL DISPLACEMENTS AT THE MID-HEIGHT NODE INDICATE POST-BUCKLING
20 # DEFORMATION.
21 # 9. RESULTS EXTRACTION: REACTION FORCES AT THE BASE NODE PROVIDE THE AXIAL COMPRESSIVE LOAD.
22 # 10. PLOTTING: THE SCRIPT VISUALIZES AXIAL FORCE VS. LATERAL DISPLACEMENT, SHOWING THE POST-BUCKLING
23 # RESPONSE OF THE COLUMN.
24 #
25 # THIS PROGRAM WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)
26 # EMAIL: salar.d.ghashghaei@gmail.com
27 #####
28 import openseespy.opensees as ops
29 import numpy as np
30 import matplotlib.pyplot as plt
31 import STEEL_FIBER_SECTION as S01
32 import ANALYSIS_FUNCTION as S02
33 import time as TI
34 #%%-
```

Start Time: 18:43:16
Finish Time: 20:15:04

No plots to show

Run plot-generating code in the Editor or IPython console to see your figures appear here. This pane only supports static images, so it can't display interactive plots like Bokeh, Plotly or Altair.

IPython Console Files Help Variable Explorer Debugger Plots History

WITH PARALLEL COMPUTING

Spyder (Python 3.12) - X

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Dell\Desktop\OPENSEES_FILES\MULTI-MODE-P...COLUMN_SEMI-RIGID_NONLINEAR_PARALLEL_COMPUTING.py

MULTI-MODE-POST_BU...RIGID_NONLINEAR.py MULTI-MODE-POST_BU...ALLEL_COMPUTING.py

```
1 ##### >>> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<
2 # LEVERAGING PARALLEL COMPUTING IN OPENSEES TO SIMULATE GEOMETRIC AND MATERIAL NONLINEARITY IN MULTI-MODE #
3 # POST-BUCKLING OF SEMI-RIGID STEEL COLUMNS
4 #
5 #
6 # IT MODELS A 2D INELASTIC BEAM-COLUMN WITH AN INITIAL IMPERFECTION (FOUR DIFFERENT SHAPES)
7 # AND APPLIES AN AXIAL COMPRESSIVE LOAD TO ANALYZE LARGE DISPLACEMENTS.
8 # 1. MODEL SETUP: A COLUMN OF LENGTH L IS DEFINED WITH N ELEMENTS, INCORPORATING A SMALL INITIAL
9 # IMPERFECTION (HALF-SINE WAVE).
10 # 2. NODES & BOUNDARY CONDITIONS: NODES ARE CREATED, WITH THE BOTTOM FIXED IN X, Y AND THE TOP FIXED IN
11 # X BUT FREE IN Y AND ROTATION.
12 # 3. ELEMENT DEFINITION: THE COLUMN IS MODELED USING ELASTIC BEAM-COLUMN ELEMENTS WITH COROTATIONAL
13 # TRANSFORMATION FOR GEOMETRIC NONLINEARITY.
14 # 4. LOAD APPLICATION: A STATIC AXIAL FORCE P IS APPLIED AT THE TOP NODE.
15 # 5. ANALYSIS SETUP: A DISPLACEMENTCONTROL INTEGRATOR IS USED TO INCREMENTALLY PUSH THE COLUMN DOWNWARDS.
16 # 6. NONLINEAR SOLVER: THE NEWTON METHOD IS USED WITH A NORMDISPINC TEST FOR CONVERGENCE.
17 # 7. ANALYSIS EXECUTION: THE LOOP PERFORMS INCREMENTAL LOADING STEPS, RECORDING AXIAL DISPLACEMENT,
18 # LATERAL DISPLACEMENT, AND AXIAL FORCE.
19 # 8. BUCKLING BEHAVIOR CAPTURE: LATERAL DISPLACEMENTS AT THE MID-HEIGHT NODE INDICATE POST-BUCKLING
20 # DEFORMATION.
21 # 9. RESULTS EXTRACTION: REACTION FORCES AT THE BASE NODE PROVIDE THE AXIAL COMPRESSIVE LOAD.
22 # 10. PLOTTING: THE SCRIPT VISUALIZES AXIAL FORCE VS. LATERAL DISPLACEMENT, SHOWING THE POST-BUCKLING
23 # RESPONSE OF THE COLUMN.
24 #
25 # PARALLEL PROCESSING MEANS RUNNING SEVERAL TASKS AT THE SAME TIME INSTEAD OF ONE AFTER ANOTHER.
26 # IN YOUR ORIGINAL CODE, EACH MODE OF POST-BUCKLING ANALYSIS WAS CALCULATED IN SEQUENCE,
27 # SO THE CPU WORKED ON ONLY ONE MODE AT ANY MOMENT. IN THE REWRITTEN VERSION, THE JOBLIB LIBRARY ALLOWS
28 # ALL FOUR MODES TO RUN SIMULTANEOUSLY ON DIFFERENT CPU CORES. EACH CORE PROCESSES ONE MODE INDEPENDENTLY,
29 # SO THE TOTAL COMPUTATION TIME BECOMES MUCH SHORTER.
30 #
31 # MODERN COMPUTERS USUALLY HAVE MULTIPLE CORES, FOR EXAMPLE 4, 8, OR EVEN MORE. WHEN WE USE PARALLEL
32 # PROCESSING, WE DIVIDE THE WORKLOAD ACROSS THESE CORES. BECAUSE EACH MODE IS A SEPARATE AND INDEPENDENT
33 # ANALYSIS, THEY ARE PERFECT FOR PARALLEL EXECUTION. INSTEAD OF WAITING FOR MODE 1 TO FINISH BEFORE
34 # STARTING MODE 2, ALL MODES START TOGETHER AND FINISH ALMOST TOGETHER.
```

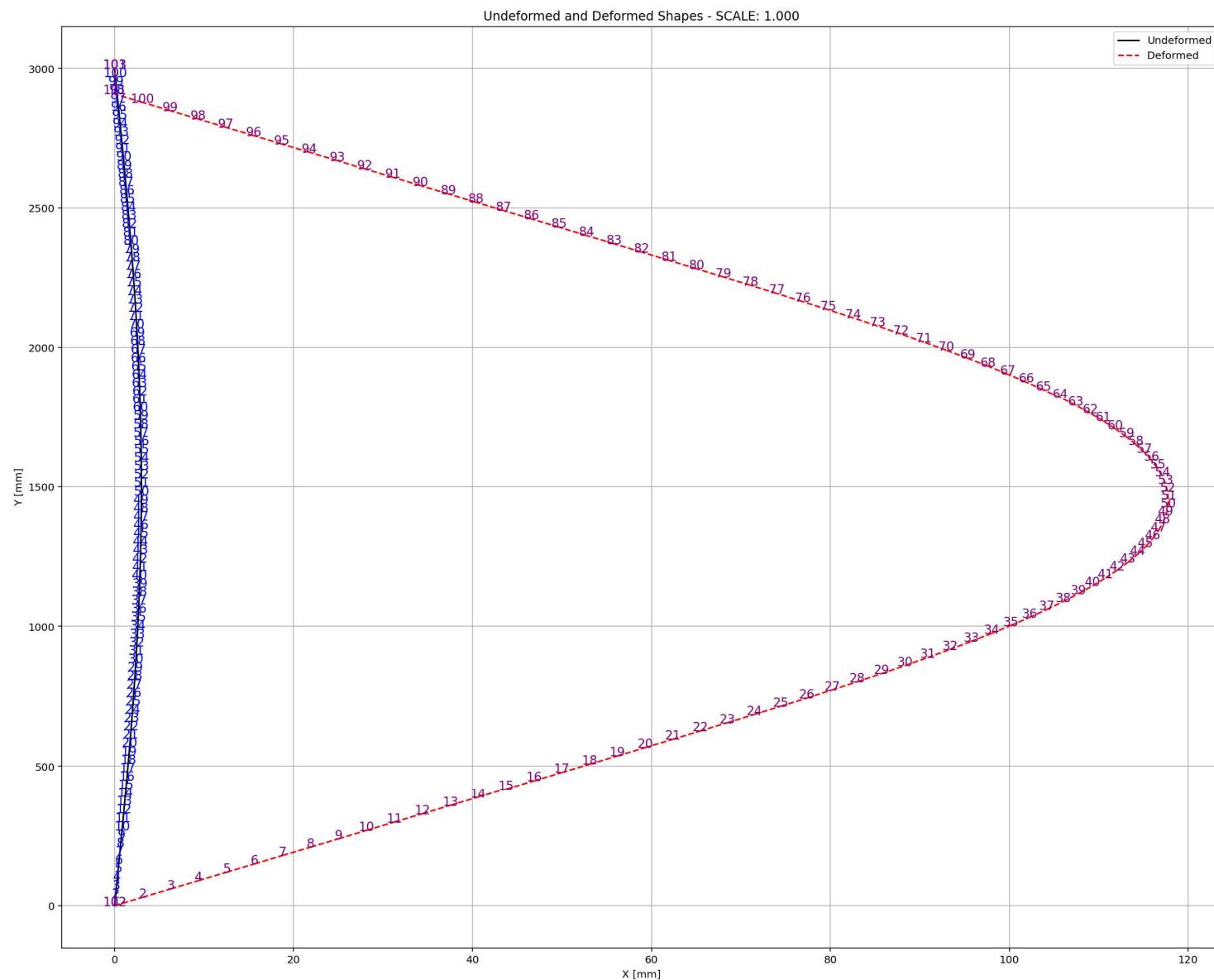
Start Time: 11:24:45
Finish Time: 11:58:02

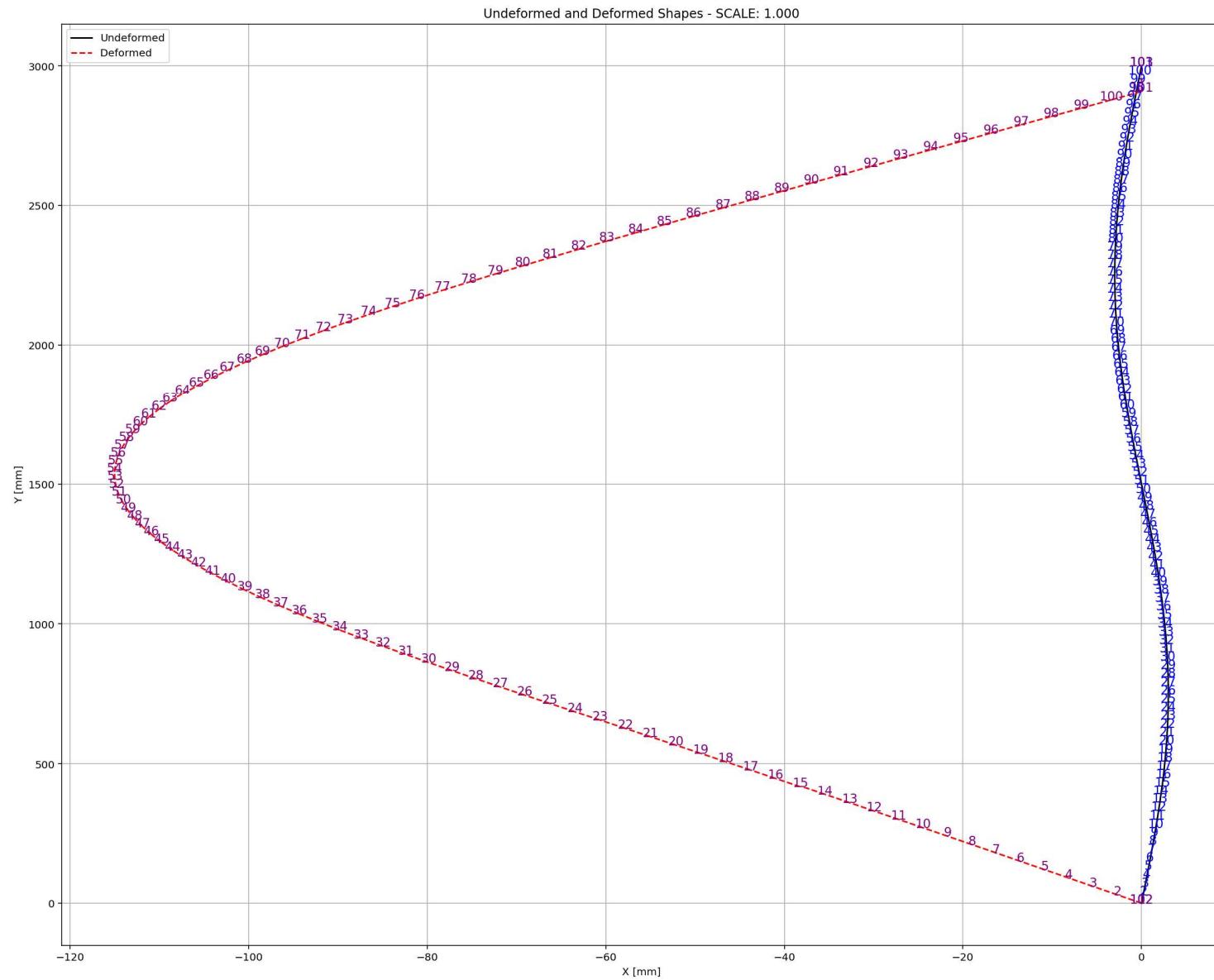
No plots to show

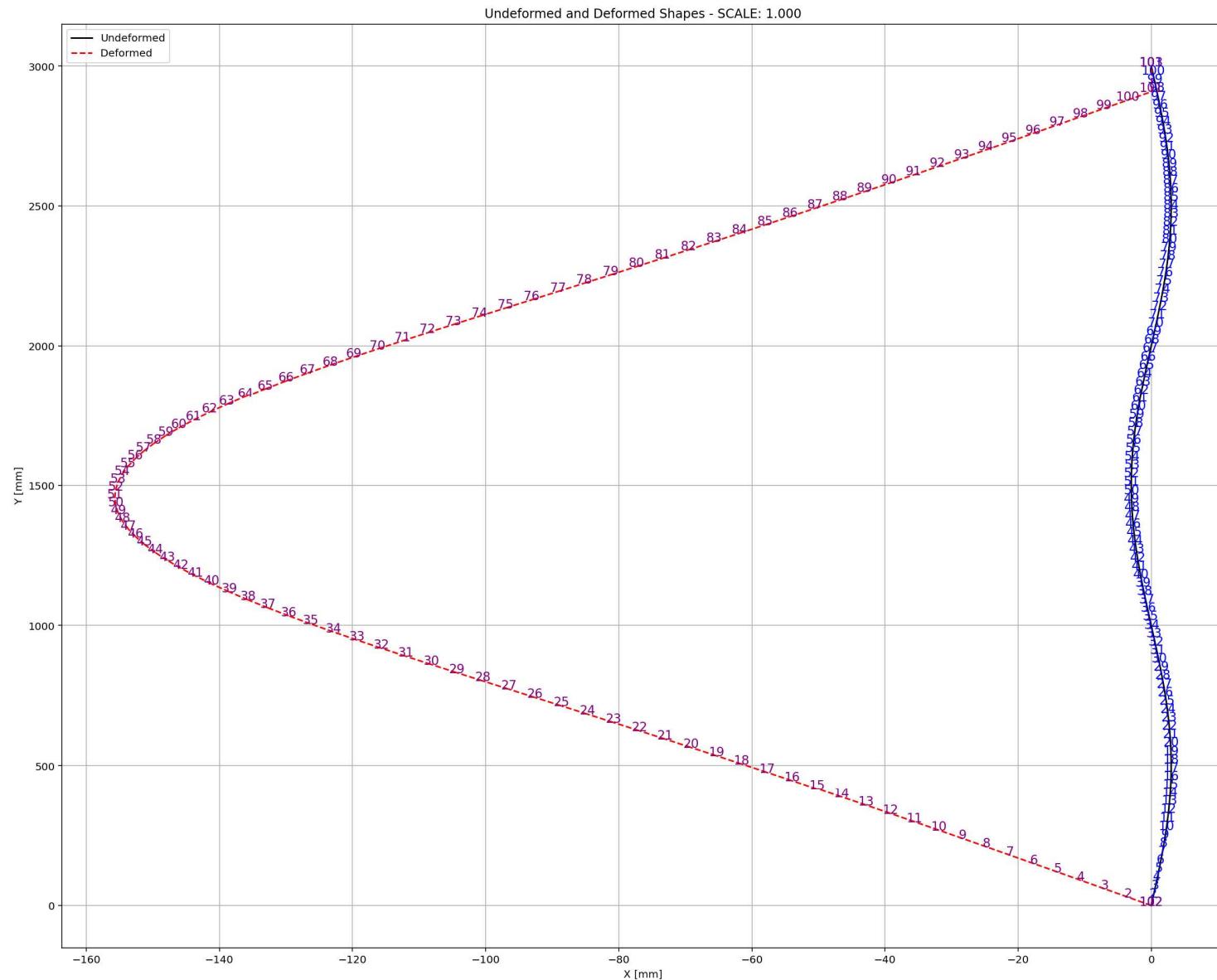
Run plot-generating code in the Editor or IPython console to see your figures appear here. This pane only supports static images, so it can't display interactive plots like Bokeh, Plotly or Altair.

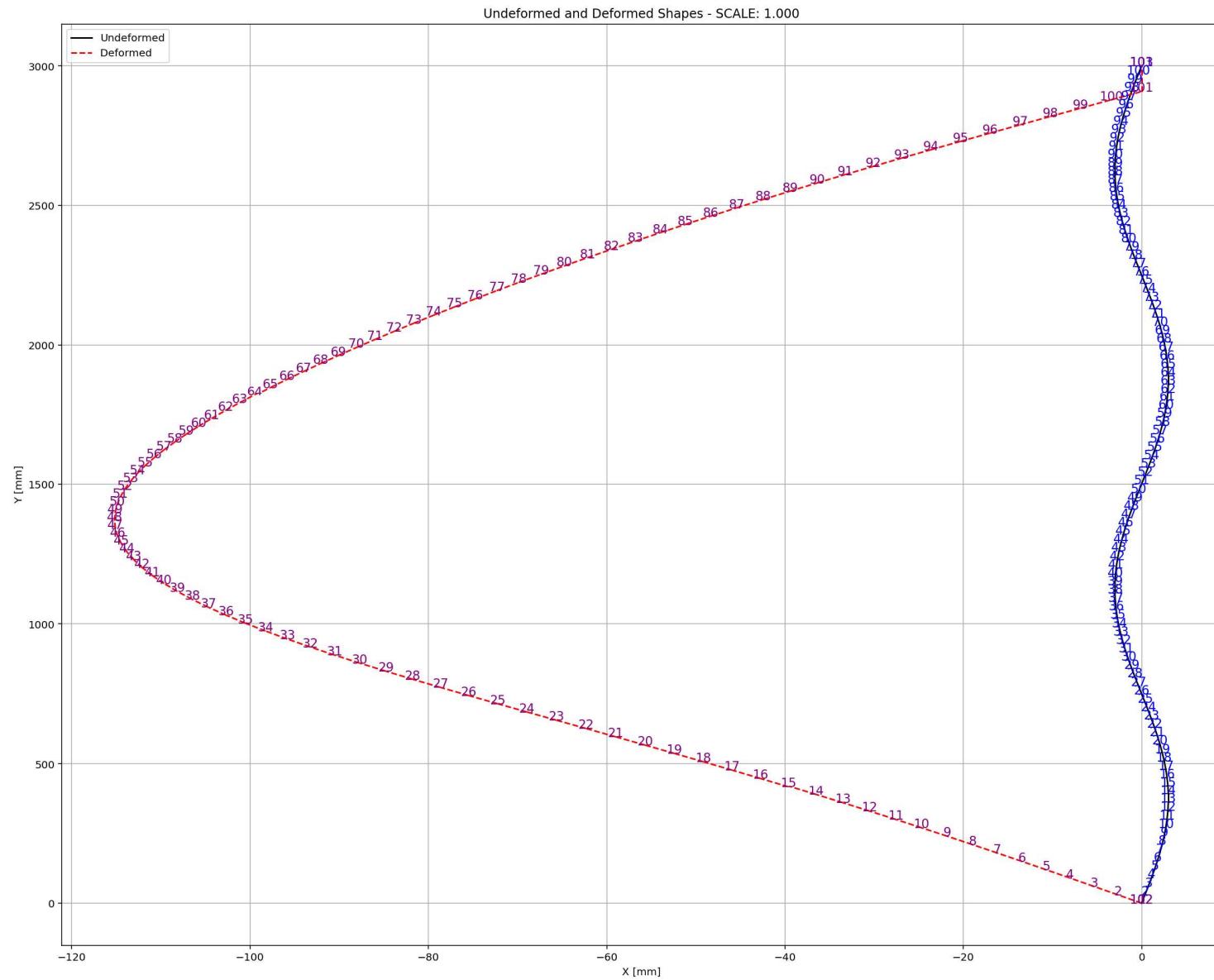
IPython Console Files Help Variable Explorer Debugger Plots History

**IN THIS ANALYSIS, USING
PARALLEL COMPUTING (JOBLIB),
WE CAN RUN THE CODE AT LEAST
8 TIMES FASTER**

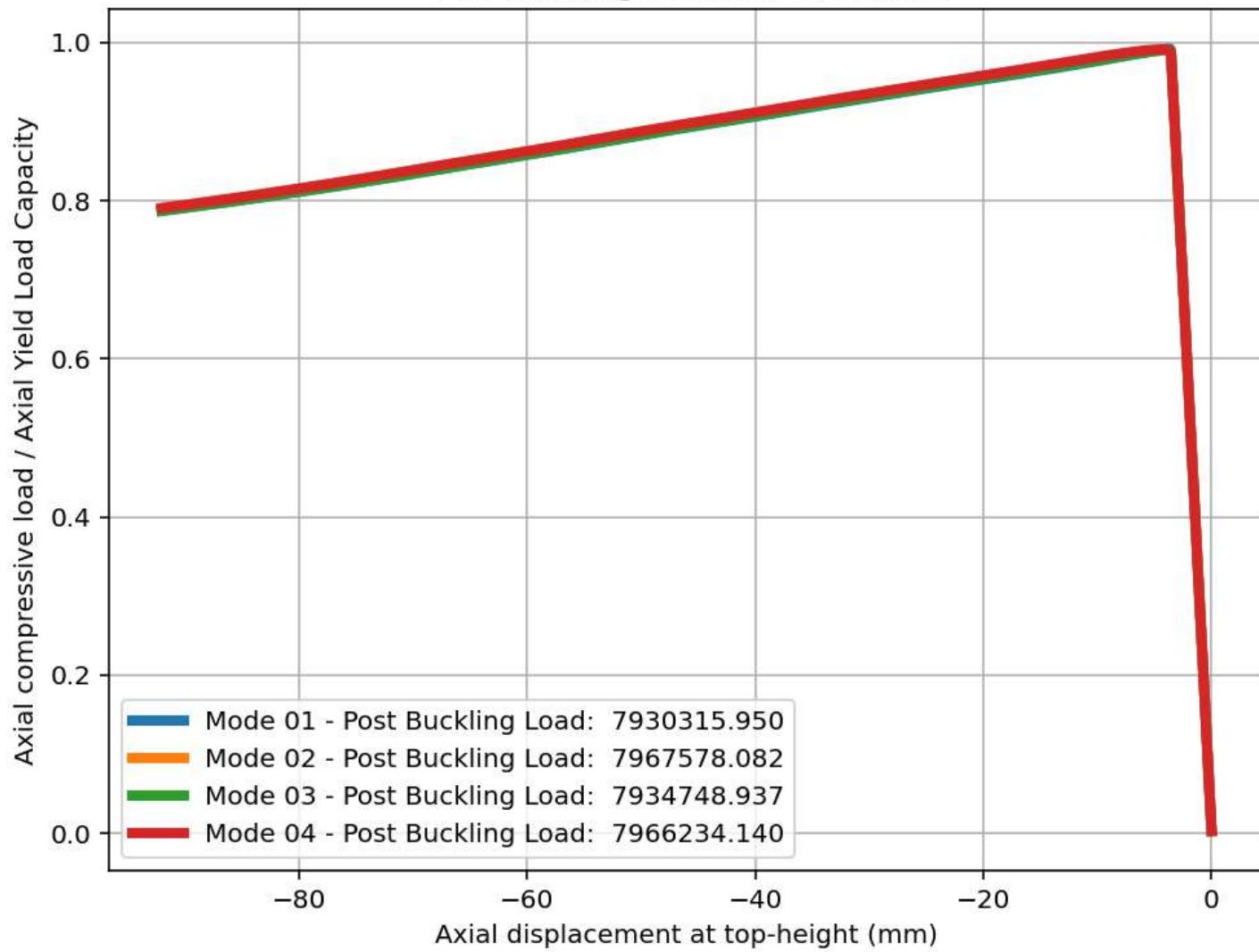




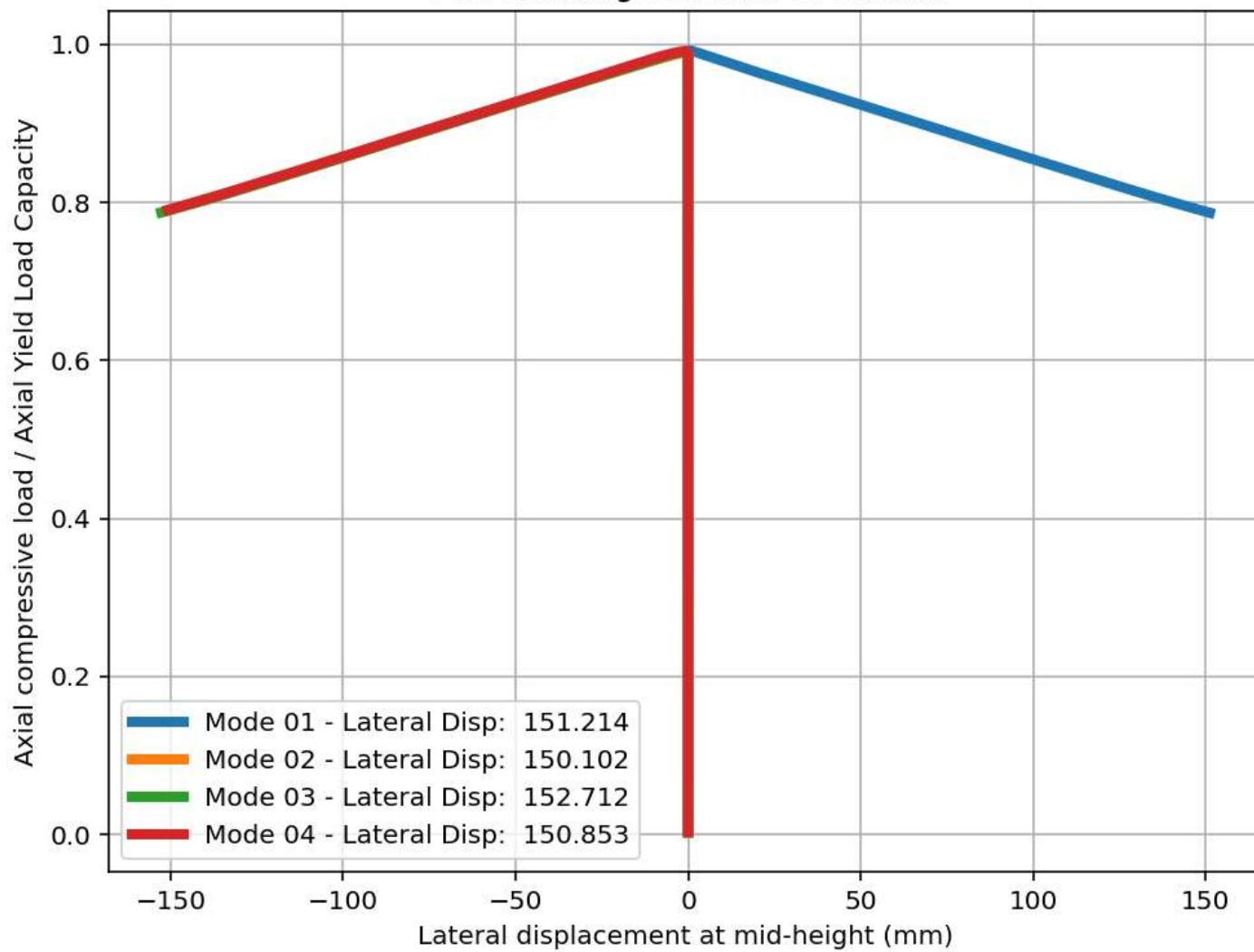




Post-buckling behavior of column



Post-buckling behavior of column



Post-buckling behavior of column

