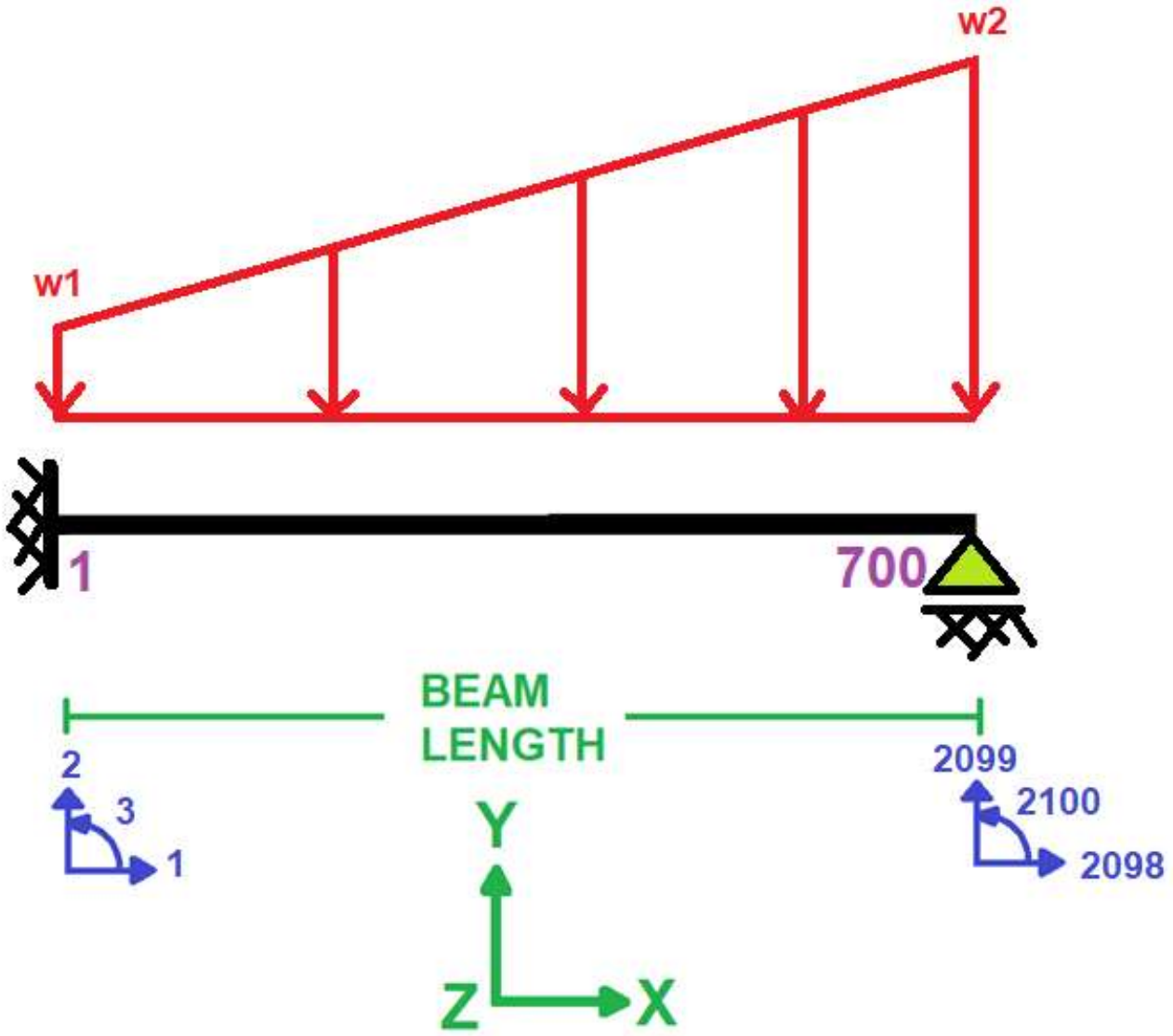
>> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<

LINEAR ELASTIC BEAM ANALYSIS USING OPENSEES

WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)


w2
w1
1
700
BEAM
LENGTH
2
3
1
2099
2100
2098
Y
Z
X

Spyder (Python 3.12)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Dell\Desktop\OPENSEES_FILES\SIMPLY_SUPPORTED_BEAM

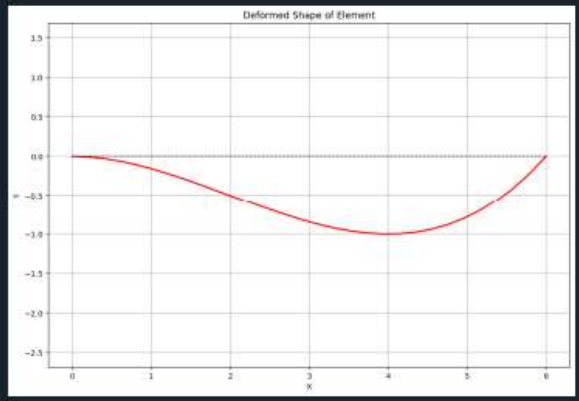
C:\Users\Dell\Desktop\OPENSEES_FILES\SIMPLY_SUPPORTED_BEAM\SIMPLY_SUPPORTED_BEAM.py

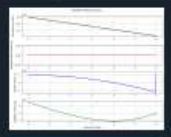
SIMPLY_SUPPORTED_BEAM.py

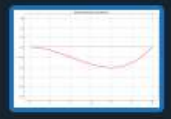
```
#############################################################################################
#                          >> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<
#                                  LINEAR ELASTIC BEAM ANALYSIS USING OPENSEES
#-------------------------------------------------------------------------------------------
#                          THIS PROGRAM WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)
#                                  EMAIL: salar.d.ghashghaei@gmail.com
#############################################################################################
"""
1. Model setup: Creates a 2D linear elastic beam of length 6 m divided into 699 elements, with specifi
2. Nodes and supports: Nodes are generated along the beam, with one end hinged and the other on a roll
3. Elements and loading: Each segment is defined as an `elasticBeamColumn` element, and the uniform lo
4. Static analysis: Performs a linear static analysis using `LoadControl` to calculate nodal displacem
5. Post-processing: The function `PLOT_ELEMENT_FORCES_AND_DEFORMED` extracts element end forces (axial
"""
import openseespy.opensees as ops
#import numpy as np
#import matplotlib.pyplot as plt

L  = 6.0          # [m] Beam length
E  = 2.1e11       # [Pa] Young's modulus
A  = 0.02         # [m^2] Cross-sectional area
Iz = 8e-4         # [m^4] Moment of inertia about the z-axis
w  = -10e3        # [N/m] Uniformly distributed load
NIDE_I, NIDE_J = 1, 700 # START , END NODE
nEle = NIDE_J-NIDE_I    # Number of elements
npts = 30               # Number of element division

#%%-------------------------------------------------------------------------------------------
ops.wipe()
ops.model('basic', '-ndm', 2, '-ndf', 3)
#%%-------------------------------------------------------------------------------------------
# Nodes
dx = L / nEle
for i in range(nEle + 1):
```

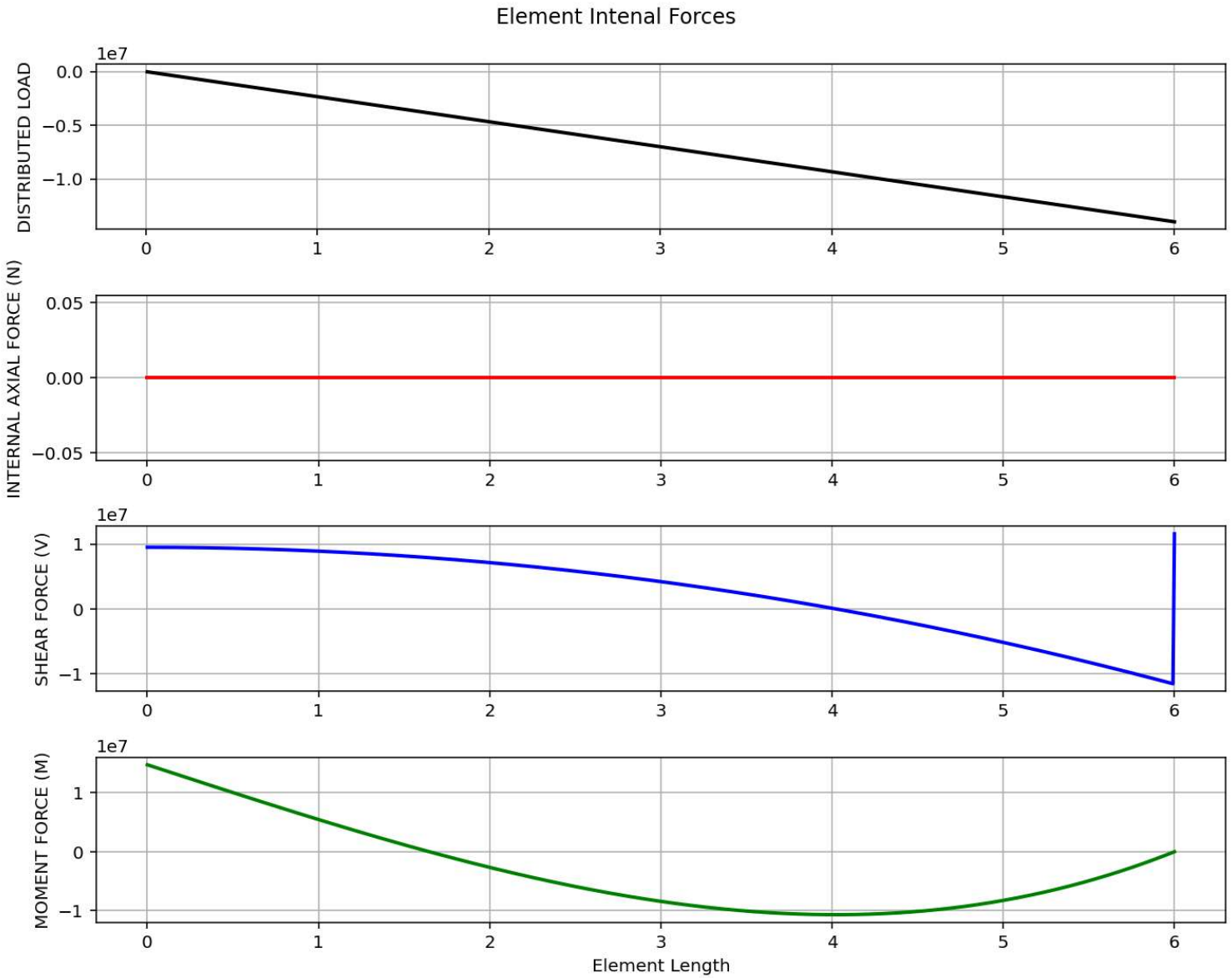


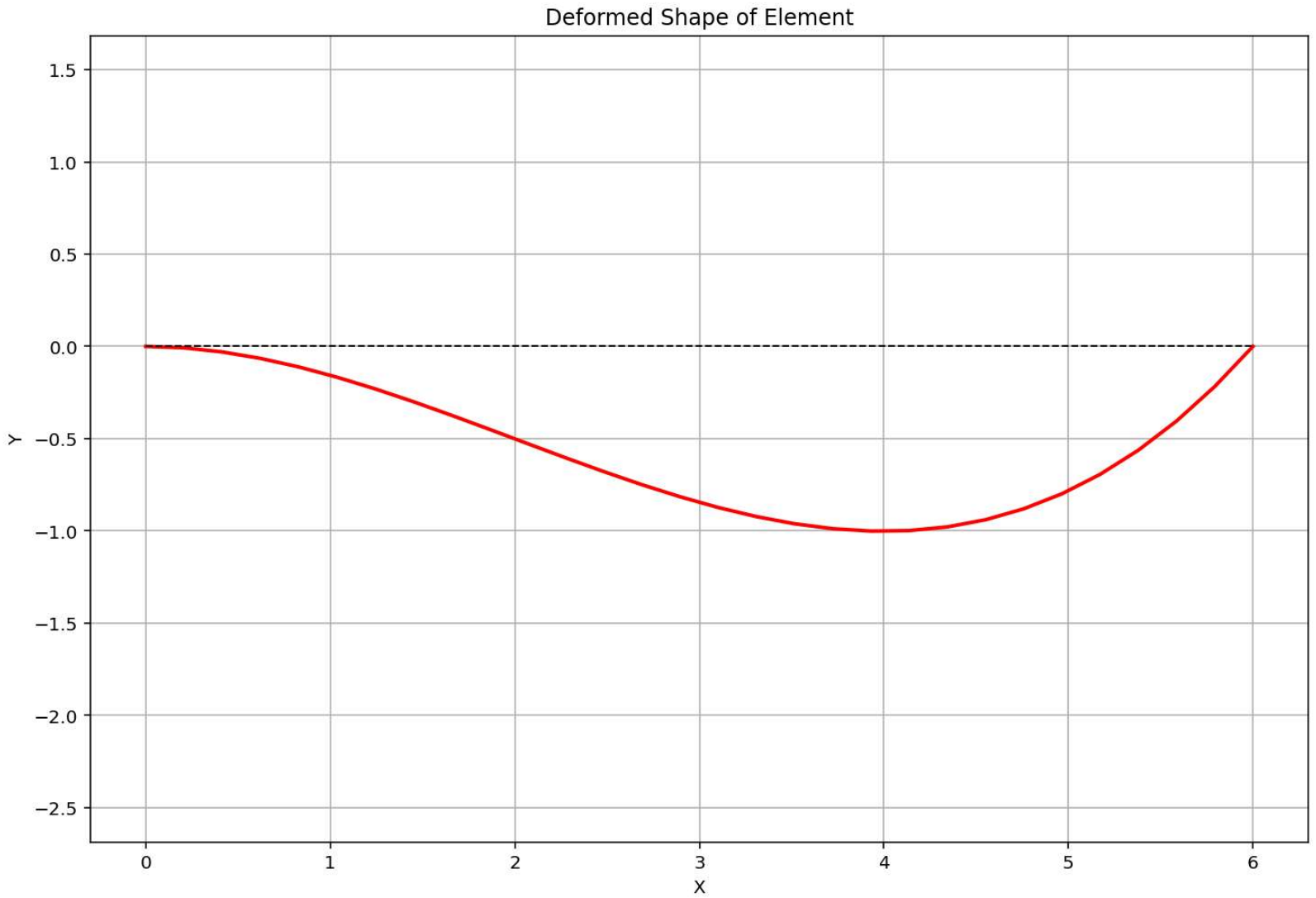




IPython Console | Files | Help | Variable Explorer | Debugger | Plots | History

Inline Conda: anaconda3 (Python 3.12.7) LSP: Python Line 3, Col 37 UTF-8 CRLF RW Mem 48%


Element Intenal Forces
1e7
DISTRIBUTED LOAD
0.0
−0.5
−1.0
0
1
2
3
4
5
6
INTERNAL AXIAL FORCE (N)
0.05
0.00
−0.05
1e7
SHEAR FORCE (V)
1
0
−1
1e7
MOMENT FORCE (M)
Element Length


Deformed Shape of Element
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
Y
0
1
2
3
4
5
6
X