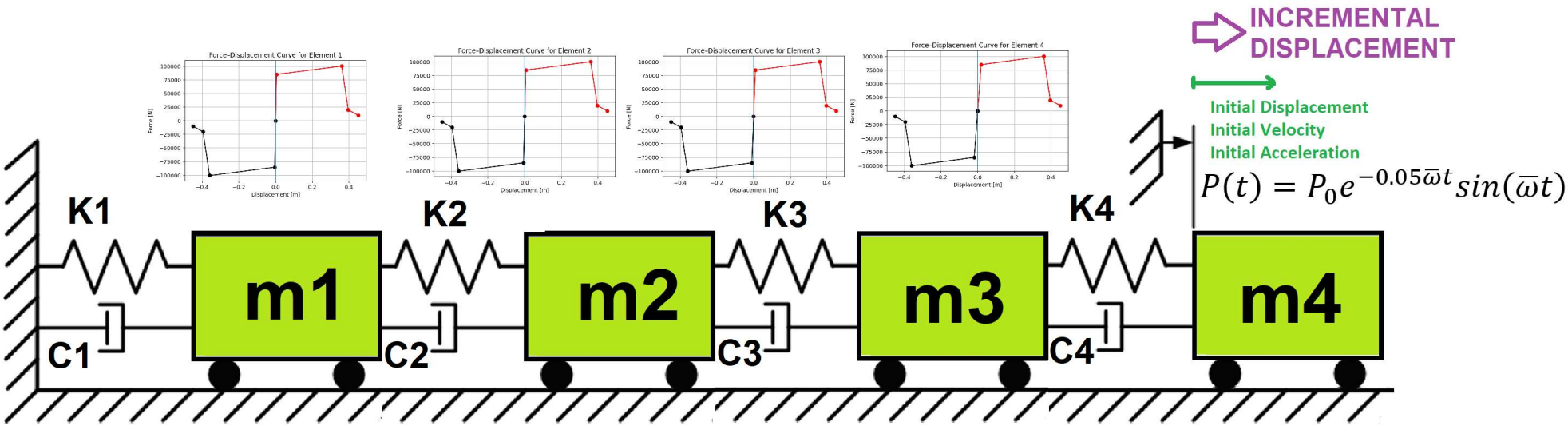
>> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<

COMPREHENSIVE NONLINEAR SEISMIC ASSESSMENT OF A MULTI-DEGREE-FREEDOM (MDOF) STRUCTURE: AN OPENSEES FRAMEWORK FOR STATIC PUSHOVER, CYCLIC DEGRADATION, STATIC TIME-HISTORY AND DYNAMIC TIME-HISTORY ANALYSIS

THIS PROGRAM IS WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)



$$Structural\ Ductility\ Damage\ Index = \frac{\Delta_d - \Delta_y}{\Delta_u - \Delta_y}$$

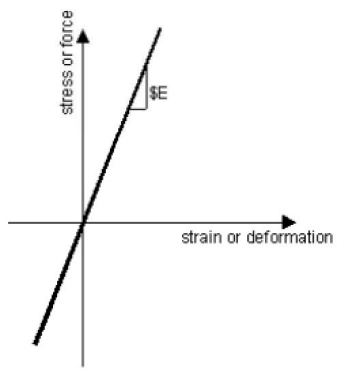
$\Delta_d = Lateral\ Displaement\ from\ Dynamic\ Analysis$

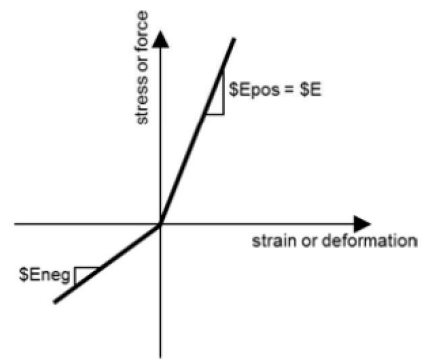
$\Delta_y = Lateral\ Yield\ Displaement\ from\ Pushover\ Analysis$

$\Delta_u = Lateral\ Ultimate\ Displaement\ from\ Pushover\ Analysis$

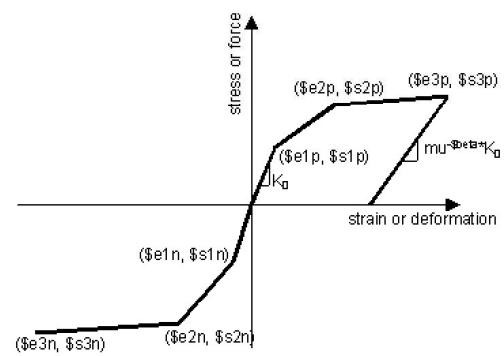
$$P(t) = P_0 e^{-0.05\bar{\omega}t} sin(\bar{\omega}t)$$

$$m\ddot{u} + c\dot{u} + ku = P(t)$$

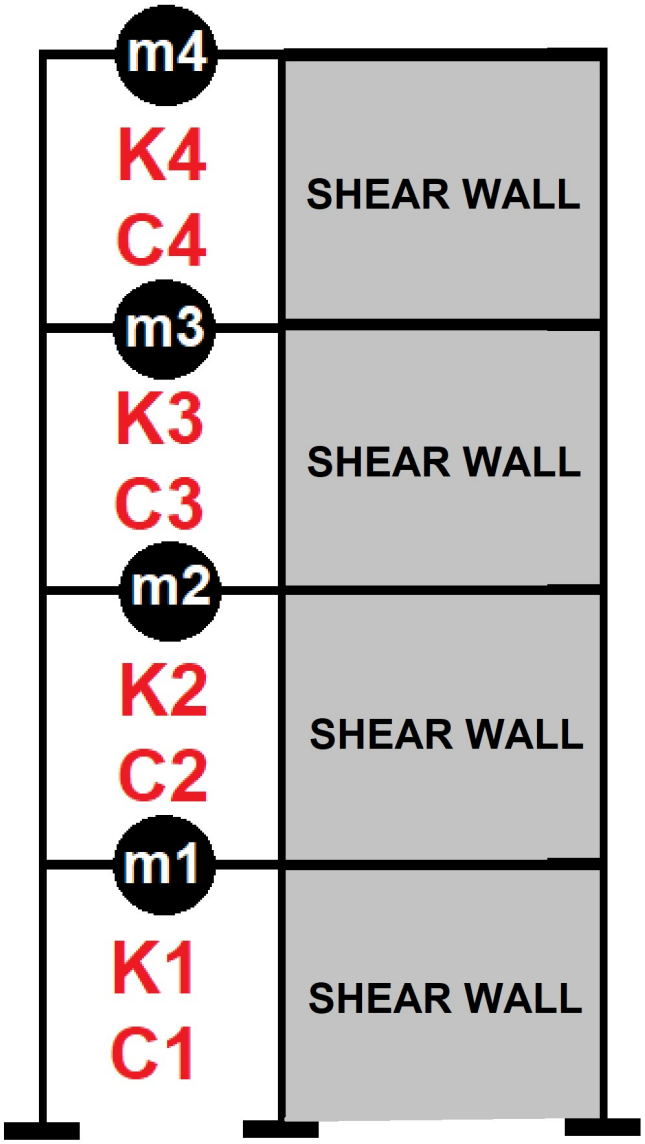




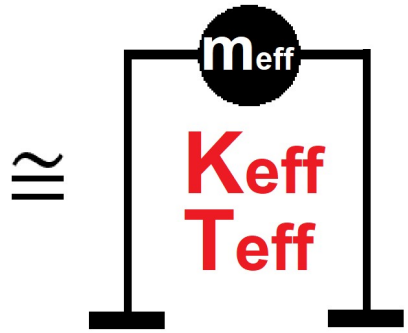
ELASTIC MATERIAL



INELASTIC MATERIAL

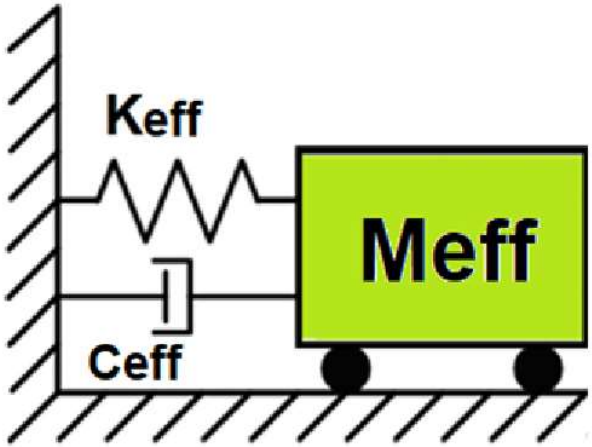


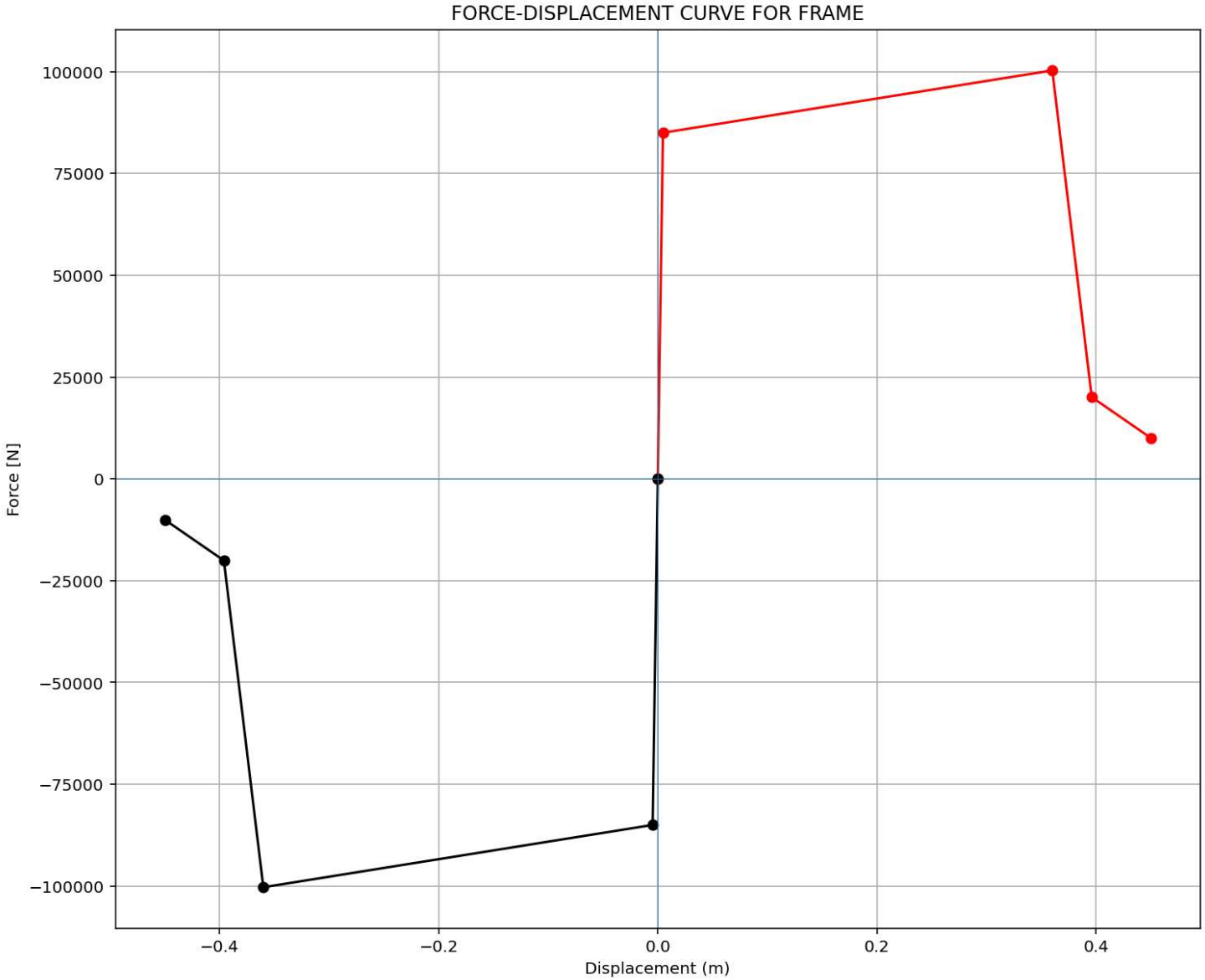
Displacement-based seismic analysis transformation, converting a multi-degree-of-freedom (MDOF) system into an equivalent single-degree-of-freedom (SDOF) system for seismic assessment

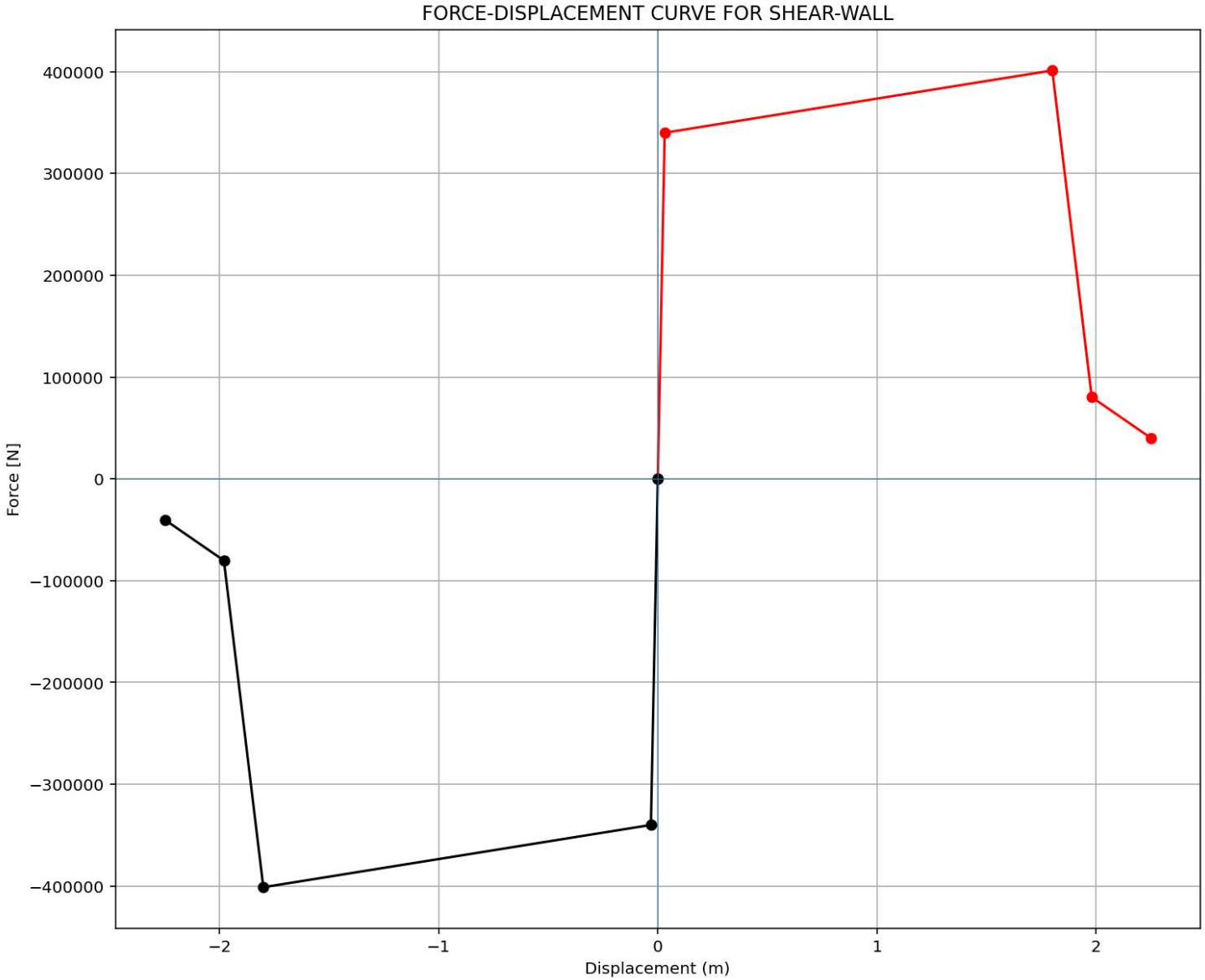


$$\Delta_d = \frac{\sum_{i=1}^{n} m_i \Delta_i^2}{\sum_{i=1}^{n} m_i \Delta_i} \qquad m_{eff} = \frac{\sum_{i=1}^{n} m_i \Delta_i}{\Delta_d}$$

$$k_{eff} = \frac{\sum_{i=1}^{n} k_i \Delta_i}{\Delta_d} \qquad T_{eff} = \frac{2\pi}{\sqrt{\frac{k_{eff}}{m_{eff}}}}$$




FORCE-DISPLACEMENT CURVE FOR FRAME
100000
75000
50000
25000
0
−25000
−50000
−75000
−100000
Force [N]
−0.4
−0.2
0.0
0.2
0.4
Displacement (m)


FORCE-DISPLACEMENT CURVE FOR SHEAR-WALL
Force [N]
400000
300000
200000
100000
0
−100000
−200000
−300000
−400000
−2
−1
0
1
2
Displacement (m)

Spyder (Python 3.12)

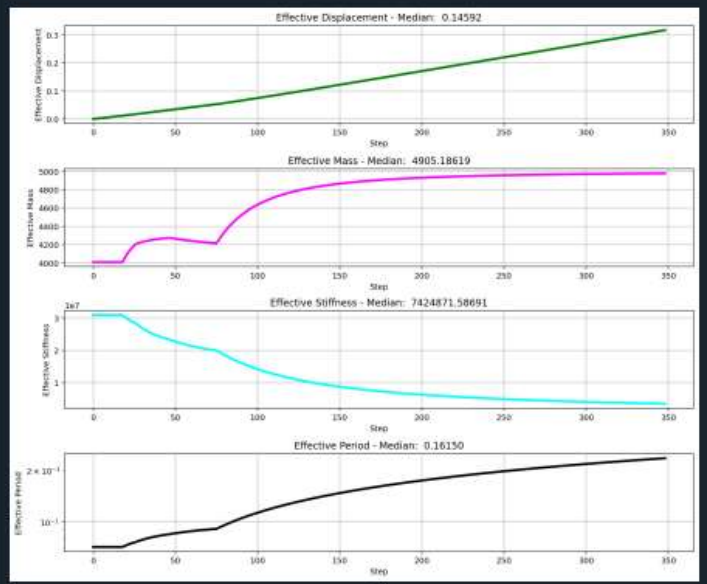
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Dell

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...TY_INDEX\33_MDOF_TWO_8_ANA\P(t)_MDOF_TWO_8_ANA.py

P(t)_MDOF_TWO_8_ANA.py | COMPUTE_EFFECTIVE_PROPERTIES_FUN.py

```
##########################################################################################
#                    >> IN THE NAME OF ALLAH, THE MOST GRACIOUS, THE MOST MERCIFUL <<
# COMPREHENSIVE NONLINEAR SEISMIC ASSESSMENT OF A MULTI-DEGREE-FREEDOM STRUCTURE : AN OPEN
#    FOR STATIC PUSHOVER, CYCLIC DEGRADATION, STATIC TIME-HISTORY AND DYNAMIC TIME-HISTORY
#------------------------------------------------------------------------------------------
#                       EACH STORY HAS FRAME AND SHEAR-WALL (OR BRACE) INTERACTION
#------------------------------------------------------------------------------------------
# EQUIVALENT SDOF SYSTEM DERIVATION VIA DISPLACEMENT-BASED SEISMIC DESIGN PROCEDURE WITH P
#------------------------------------------------------------------------------------------
#                                          P(t) = P0 sin(wt)
#                                     P(t) = P0 exp(-0.05wt) sin(wt)
#------------------------------------------------------------------------------------------
#         ASSESSMENT OF DUCTILITY DAMAGE INDICES FOR STRUCTURAL ELEMENTS AND SYSTEMS AND E
#                                     OF ENERGY DISSIPATION CAPACITY INDEX
#------------------------------------------------------------------------------------------
#                         THIS PROGRAM WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)
#                                   EMAIL: salar.d.ghashghaei@gmail.com
##########################################################################################
"""
Nonlinear Seismic Performance Assessment of a MDOF:
An OpenSeesPy Framework for Material and Geometric Nonlinearity Under Static, Cyclic, and

This OpenSeesPy script performs rigorous nonlinear static and dynamic analysis of a MDOF
 for performance-based earthquake engineering. The 2D model incorporates both material non
 (elastic-perfectly plastic or hysteretic steel with strain hardening)
 and geometric nonlinearity (corotational truss formulation) to capture P-delta effects
 and large displacements—essential for collapse assessment.

Eight analysis protocols are implemented:
(1) [PERIOD] : Structural Period
(2) [STATIC] : Gravity load analysis establishing dead load state
(3) [PUSHOVER] : Displacement-controlled pushover generating full capacity curves
 and plastic mechanism identification
(4) [CYCLIC DISPLACEMENT] : Symmetric cyclic displacement protocol capturing hysteresis,
```



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

Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 6, Col 108 | UTF-8 | CRLF | RW | Mem 43%

STATIC ANALYSIS

Spyder (Python 3.12)

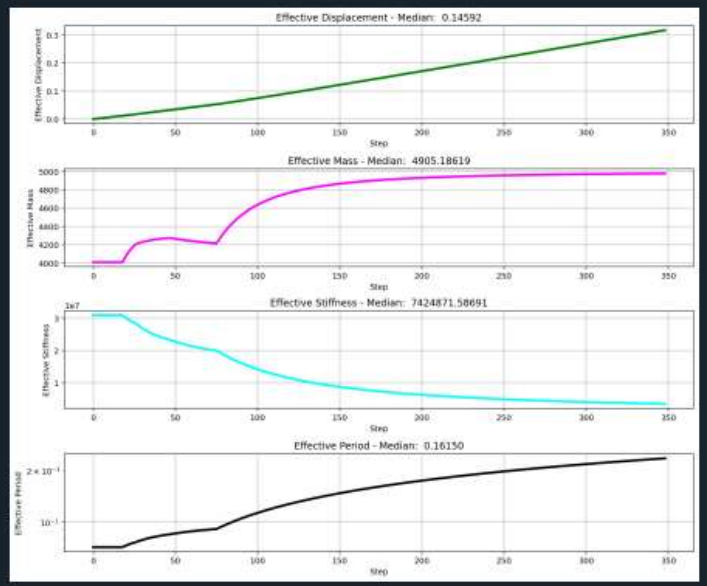
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Dell

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...TY_INDEX\33_MDOF_TWO_8_ANA\P(t)_MDOF_TWO_8_ANA.py

P(t)_MDOF_TWO_8_ANA.py | OPENSEEES_HYSTERETICSM_FORCE_DISP_FUN.py

```
        # Compute modal properties
        ops.modalProperties("-print", "-file", f"SALAR_ModalReport_{ANAL_TYPE}.txt", "-unorm
        return PERIODmin, PERIODmax

    if ANAL_TYPE == 'STATIC':
        ops.integrator('LoadControl', 1.0)
        ops.analysis('Static')
        OK = ops.analyze(1)
        S02.ANALYSIS(OK, 1, MAX_TOLERANCE, MAX_ITERATIONS)
        ops.reactions()
        reaction.append(ops.nodeReaction(1, 1))                    # BASE REACTION
        disp.append(ops.nodeDisp(center_node, 1))                  # DISPLACEMENT NOD
        DI.append(100*(np.abs(disp[-1])-DY)/(DSU-DY))              # DAMAGE INDEX
        if DI[-1] <= 0.0: # DAMAGE INDEX -> 0.0 <= DI <=100
            DI[-1] = 0.0
        if DI[-1] >= 100:
            DI[-1] = 100.0
        print('\n\nSTATIC ANALYSIS DONE.\n\n')

        DATA = (reaction, disp, DI)

        return  DATA

    if ANAL_TYPE == 'PUSHOVER': # STATIC TIME-HISTORY ANALYSIS
        IDctrlDOF = 1   # 1: Horizental Dispalcement - 2: Vertical Dispalcement
        DINCR = -0.001  # [m] Incremental Vertical Displacement
        DMAX = -0.35    # [m] Max. Displacement
        ops.integrator('DisplacementControl', center_node, IDctrlDOF, DINCR)
        ops.analysis('Static')
        Nsteps =  int(np.abs(DMAX/ DINCR))
        STEP = 0.0
        for step in range(Nsteps):
            OK = ops.analyze(1)
            S02.ANALYSIS(OK, 1, MAX_TOLERANCE, MAX_ITERATIONS)
```



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

Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 198, Col 94 | UTF-8 | CRLF | RW | Mem 47%

PUSHOVER ANALYSIS

Spyder (Python 3.12)

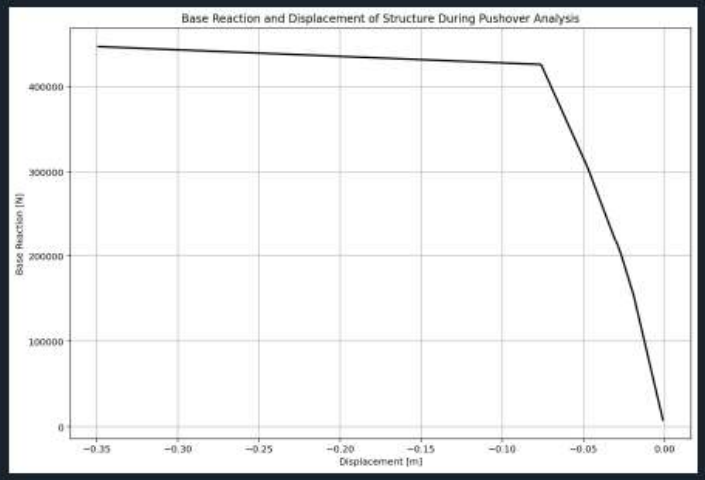
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Dell

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...TY_INDEX\33_MDOF_TWO_8_ANA\P(t)_MDOF_TWO_8_ANA.py

P(t)_MDOF_TWO_8_ANA.py | OPENSEEES_HYSTERETICSM_FORCE_DISP_FUN.py

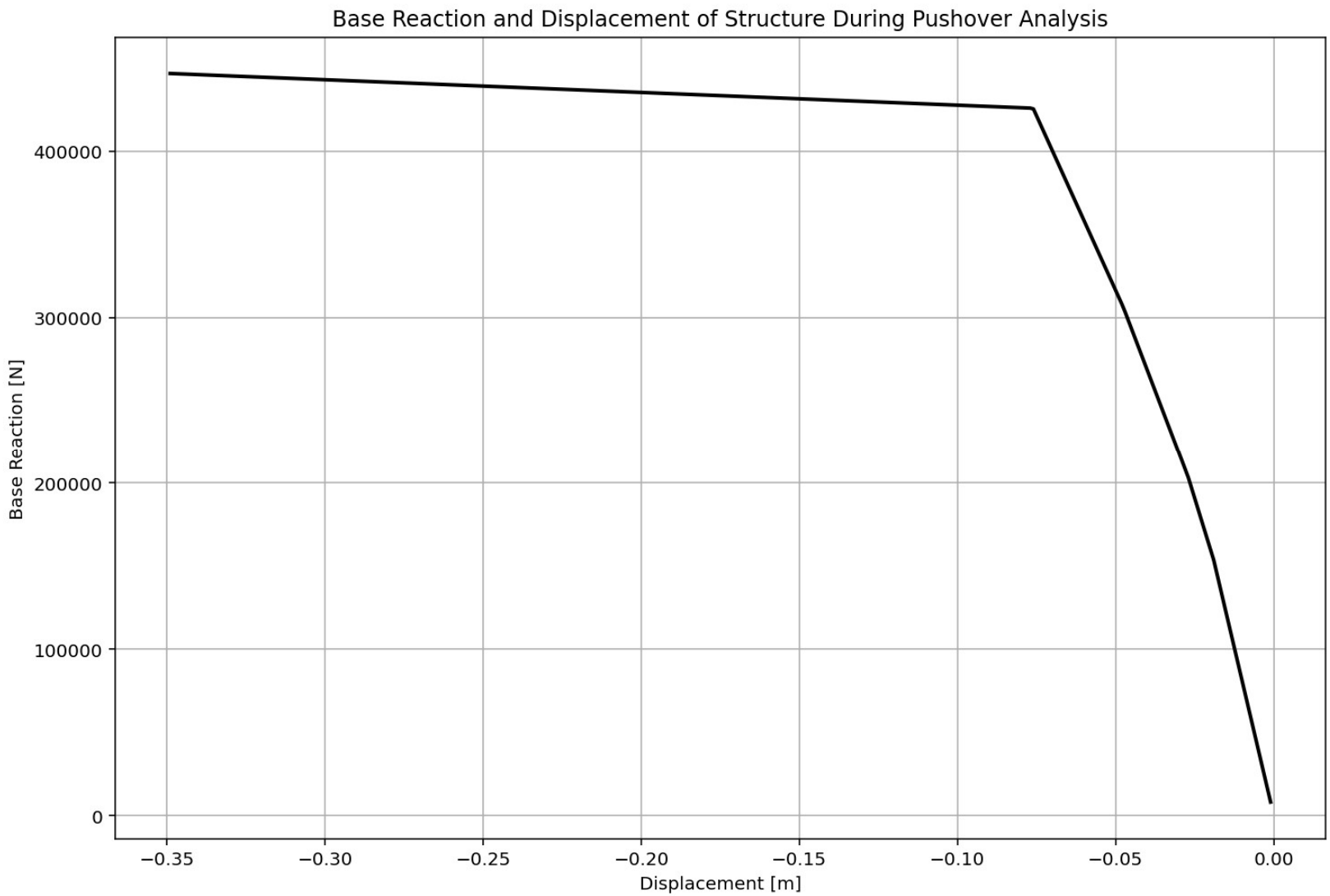
```python
    if ANAL_TYPE == 'PUSHOVER': # STATIC TIME-HISTORY ANALYSIS
        IDctrlDOF = 1   # 1: Horizental Dispalcement - 2: Vertical Dispalcement
        DINCR = -0.001  # [m] Incremental Vertical Displacement
        DMAX = -0.35    # [m] Max. Displacement
        ops.integrator('DisplacementControl', center_node, IDctrlDOF, DINCR)
        ops.analysis('Static')
        Nsteps =  int(np.abs(DMAX/ DINCR))
        STEP = 0.0
        for step in range(Nsteps):
            OK = ops.analyze(1)
            S02.ANALYSIS(OK, 1, MAX_TOLERANCE, MAX_ITERATIONS)
            ops.reactions()
            reaction.append(ops.nodeReaction(1, 1))                    # BASE REACTIO
            disp.append(ops.nodeDisp(center_node, 1))                  # DISPLACEMENT
            # IN EACH STEP, STRUCTURE PERIOD GOING TO BE CALCULATED
            #PERIODmin, PERIODmax = S06.RAYLEIGH_DAMPING(2, 0.5*DR, DR, 0, 1)
            PERIODmin, PERIODmax = S05.EIGENVALUE_ANALYSIS(2, PLOT=True)
            PERIOD_MIN.append(PERIODmin)
            PERIOD_MAX.append(PERIODmax)
            DI.append(100*(np.abs(disp[-1])-DY)/(DSU-DY))              # DAMAGE INDEX
            if DI[-1] <= 0.0: # DAMAGE INDEX -> 0.0 <= DI <=100
                DI[-1] = 0.0
            if DI[-1] >= 100:
                DI[-1] = 100.0
            # Store forces and displacements
            for ele_id in ele_force.keys():
                ele_force[ele_id].append(ops.eleResponse(ele_id, 'force')[0])        # [N]
            # Store displacements
            for node_id in node_displacements.keys():
                node_displacements[node_id].append(ops.nodeDisp(node_id, 1))
            STEP += 1
            #print(STEP, disp[-1], reaction[-1])
            print(f"Step: {STEP}, Displacement: {disp[-1]:.4f} mm, Reaction: {reaction[-1]
        else:
```

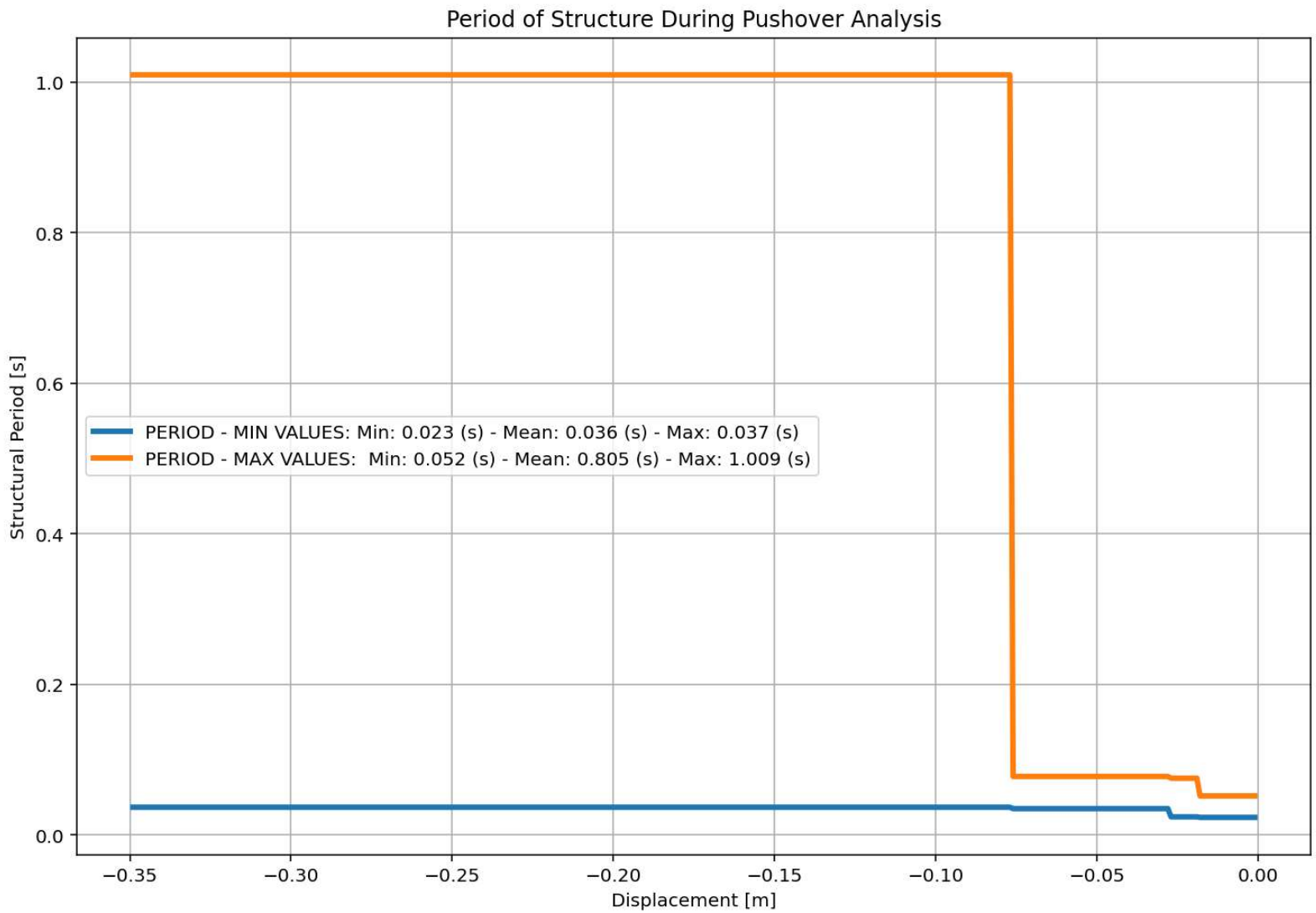


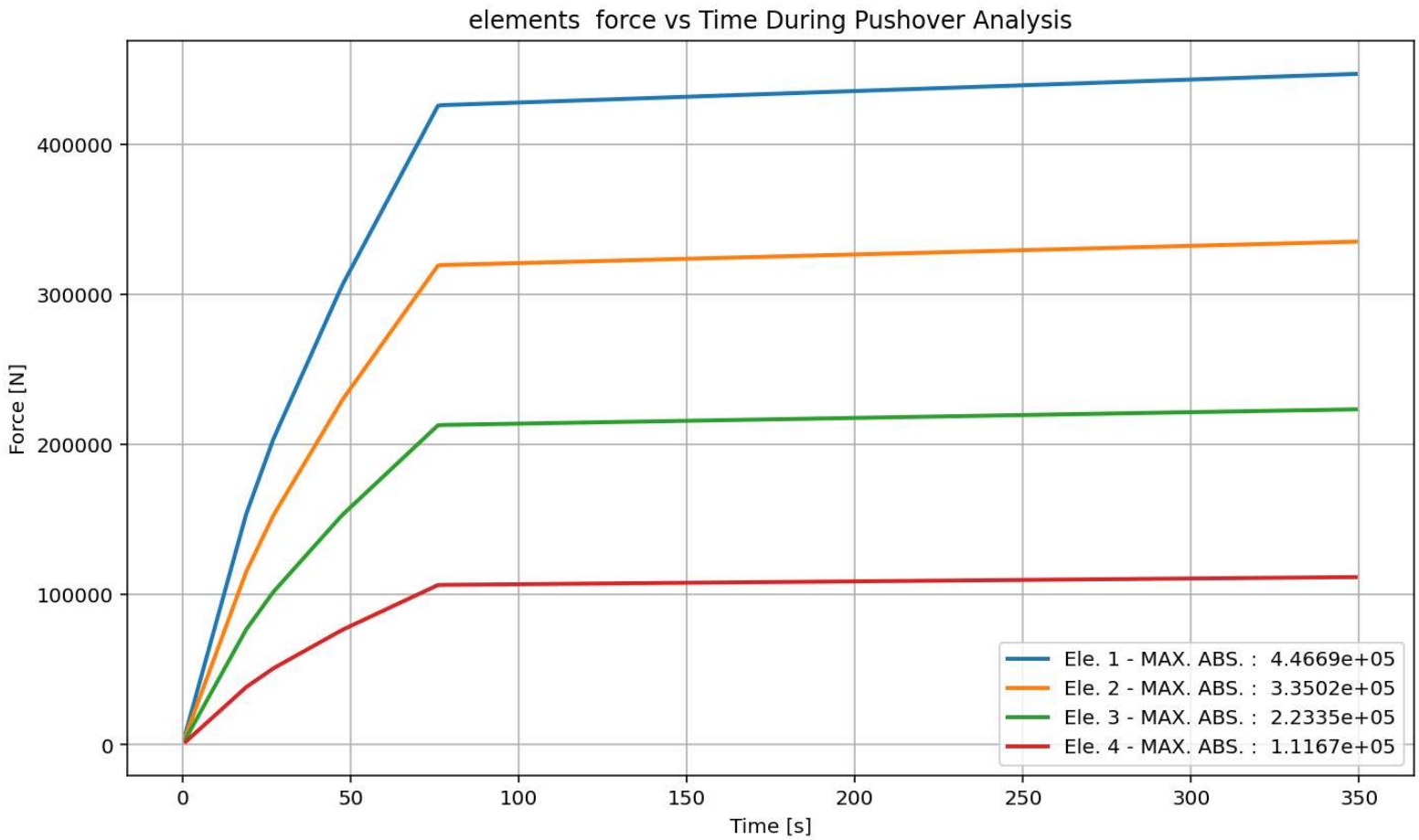
30 %

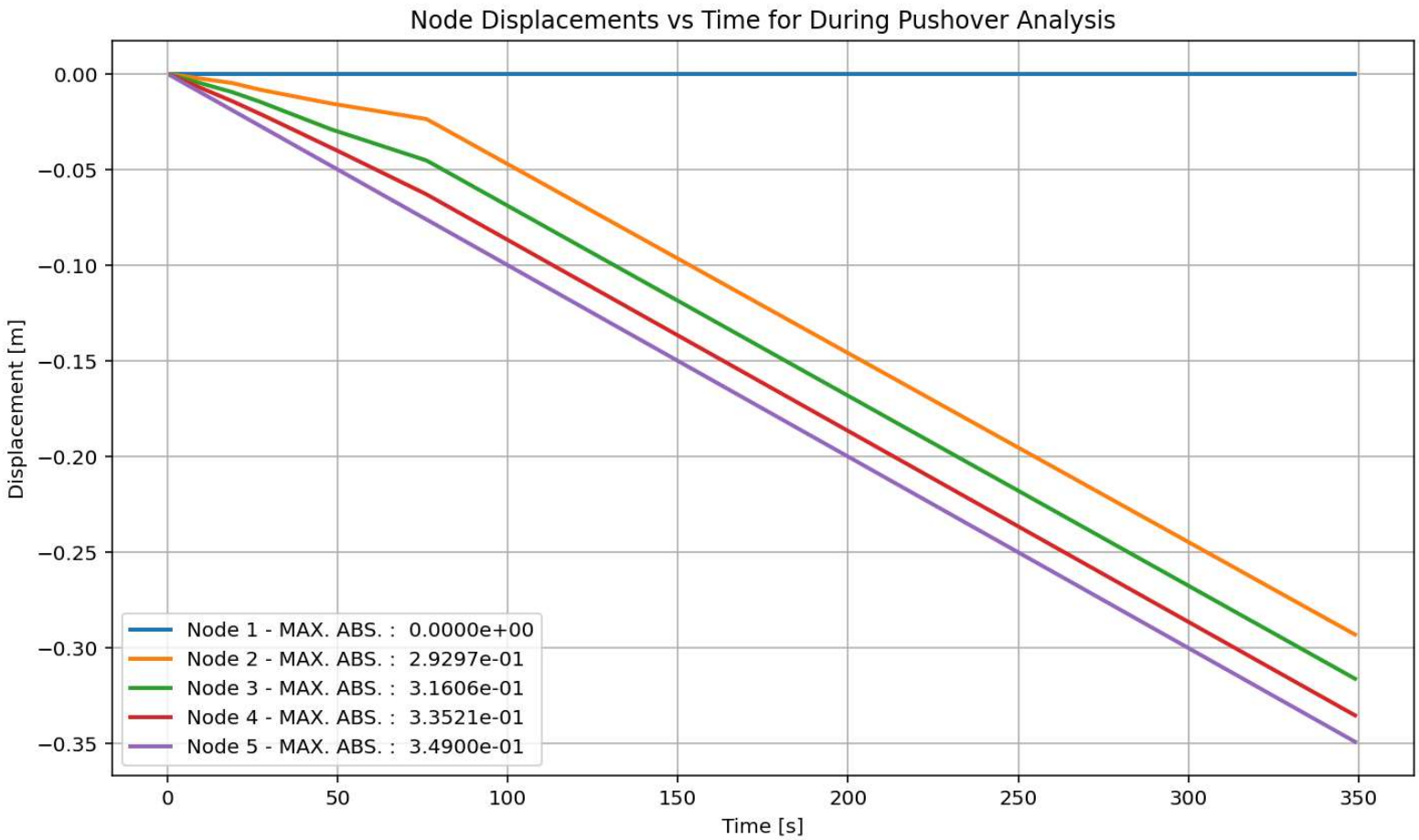
IPython Console | Files | Help | Variable Explorer | Debugger | Plots | History

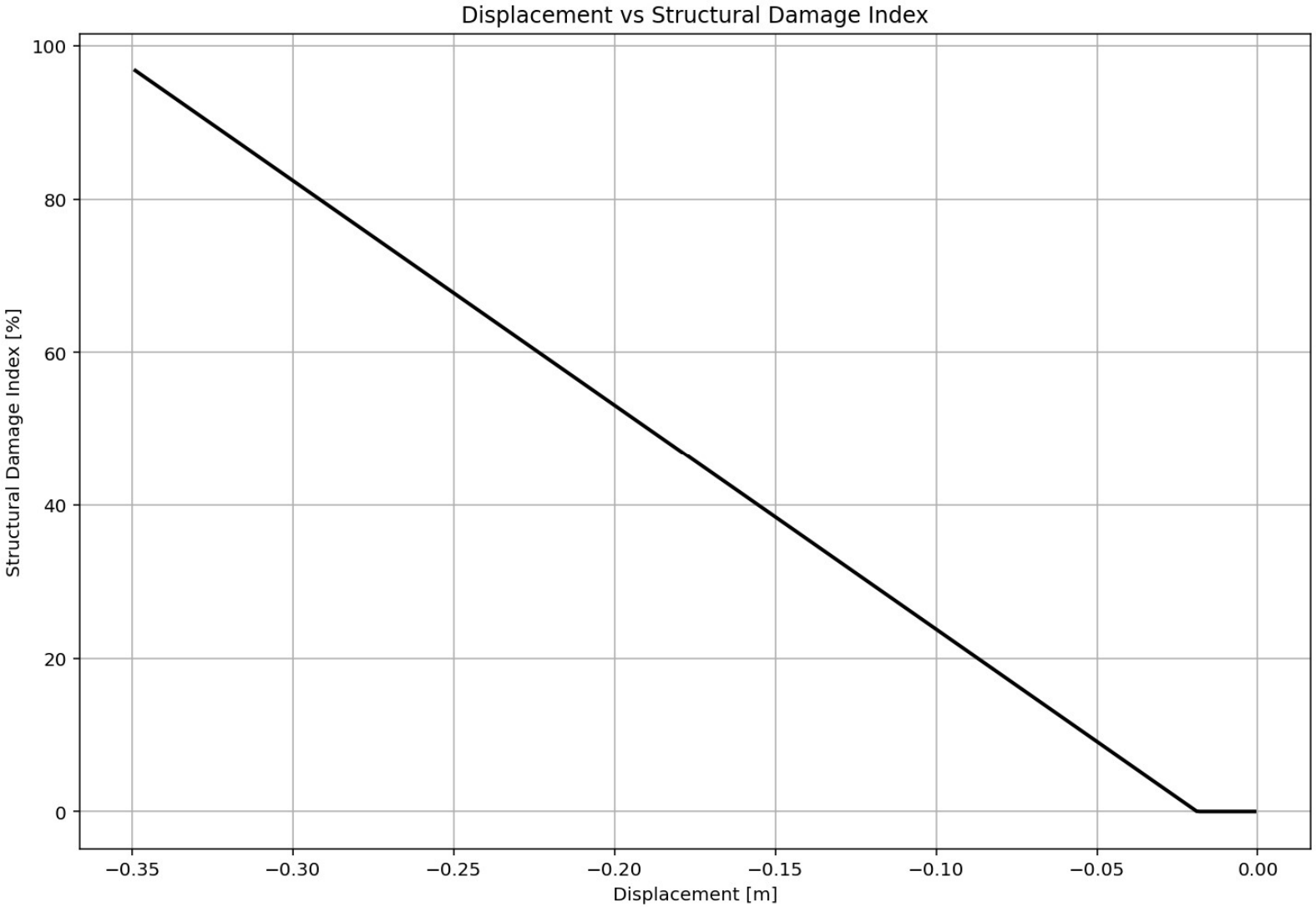
Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 198, Col 94 | UTF-8 | CRLF | RW | Mem 47%


Base Reaction and Displacement of Structure During Pushover Analysis
Base Reaction [N]
400000
300000
200000
100000
0
−0.35
−0.30
−0.25
−0.20
−0.15
−0.10
−0.05
0.00
Displacement [m]


Period of Structure During Pushover Analysis
PERIOD - MIN VALUES: Min: 0.023 (s) - Mean: 0.036 (s) - Max: 0.037 (s)
PERIOD - MAX VALUES: Min: 0.052 (s) - Mean: 0.805 (s) - Max: 1.009 (s)
Structural Period [s]
Displacement [m]
1.0
0.8
0.6
0.4
0.2
0.0
−0.35
−0.30
−0.25
−0.20
−0.15
−0.10
−0.05
0.00


elements force vs Time During Pushover Analysis
Force [N]
Time [s]
0
100000
200000
300000
400000
0
50
100
150
200
250
300
350
Ele. 1 - MAX. ABS. : 4.4669e+05
Ele. 2 - MAX. ABS. : 3.3502e+05
Ele. 3 - MAX. ABS. : 2.2335e+05
Ele. 4 - MAX. ABS. : 1.1167e+05


Node Displacements vs Time for During Pushover Analysis
Displacement [m]
Time [s]
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
−0.35
0
50
100
150
200
250
300
350
Node 1 - MAX. ABS. : 0.0000e+00
Node 2 - MAX. ABS. : 2.9297e-01
Node 3 - MAX. ABS. : 3.1606e-01
Node 4 - MAX. ABS. : 3.3521e-01
Node 5 - MAX. ABS. : 3.4900e-01


Displacement vs Structural Damage Index
Structural Damage Index [%]
100
80
60
40
20
0
−0.35
−0.30
−0.25
−0.20
−0.15
−0.10
−0.05
0.00
Displacement [m]

Spyder (Python 3.12)

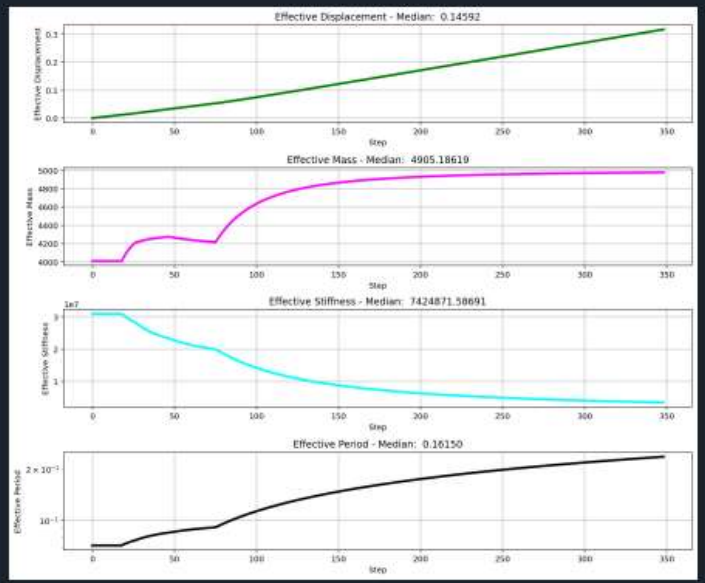
File Edit Search Source Run Debug Consoles Projects Tools View Help

...DEX_&_ENERGY_DISSIPATION_CAPACITY_INDEX\33_MDOF_TWO_8_ANA

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY..._ANA\COMPUTE_EFFECTIVE_PROPERTIES_FUN_PUSHOVER.py

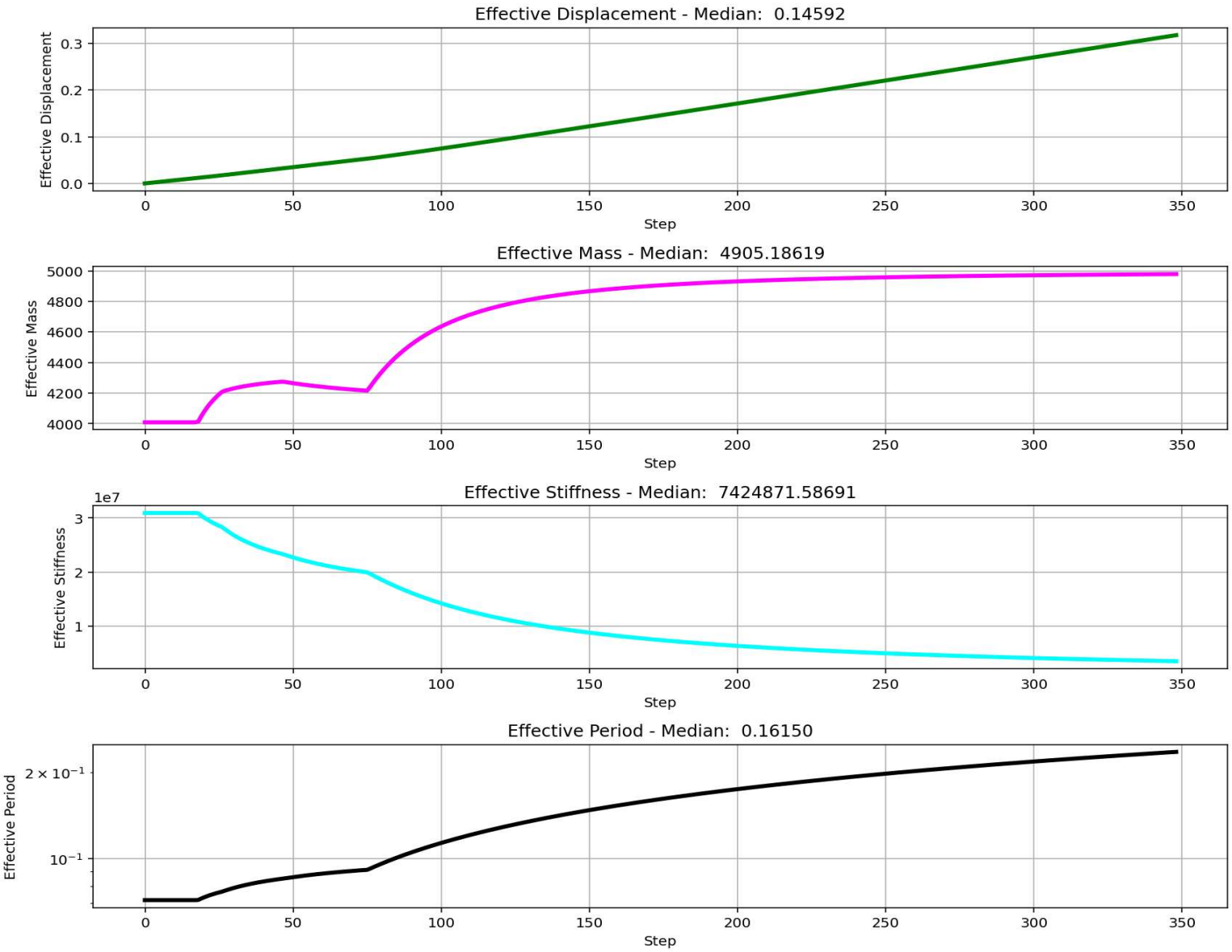
P(t)_MDOF_TWO_8_ANA.py | COMPUTE_EFFECTIVE_...ES_FUN_PUSHOVER.py

```python
#%% EQUIVALENT SDOF SYSTEM DERIVATION VIA DISPLACEMENT-BASED SEISMIC DESIGN PROCEDURE WITH P
# Change MDOF to SDOF System with Displacement Based Design Concept
# THIS PROGRAM WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)
"""
This script implements a displacement-based pushover transformation,
 converting a multi-degree-of-freedom (MDOF) system into an equivalent
 single-degree-of-freedom (SDOF) system for seismic assessment.
 It calculates effective modal properties—displacement, mass, and
 stiffness—by weighting element forces and nodal displacements according
 to a presumed deformed shape.
 The derived effective period provides a simplified dynamic characteristic
 for performance-based engineering. The visualizations effectively track the
 evolution of these equivalent parameters throughout the nonlinear analysis steps.
"""
#-----------------------------------------------------------------------
displacement_X_1 = np.array(list(node_displacements.values())[1])   # DOF 02
displacement_X_2 = np.array(list(node_displacements.values())[2])   # DOF 03
displacement_X_3 = np.array(list(node_displacements.values())[3])   # DOF 04
displacement_X_4 = np.array(list(node_displacements.values())[4])   # DOF 05

ele_force_01 = np.array(list(ele_force.values())[0])   # ELEMENT 01
ele_force_02 = np.array(list(ele_force.values())[1])   # ELEMENT 02
ele_force_03 = np.array(list(ele_force.values())[2])   # ELEMENT 03
ele_force_04 = np.array(list(ele_force.values())[3])   # ELEMENT 04

STIFF_X_1 = np.abs(ele_force_01 / displacement_X_1)
STIFF_X_2 = np.abs(ele_force_02 / displacement_X_2)
STIFF_X_3 = np.abs(ele_force_03 / displacement_X_3)
STIFF_X_4 = np.abs(ele_force_04 / displacement_X_4)

MX2 = (MASS[0] * np.square(displacement_X_1) +
       MASS[1] * np.square(displacement_X_2) +
       MASS[2] * np.square(displacement_X_3) +
       MASS[3] * np.square(displacement_X_4))
```



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

Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 1, Col 1 | UTF-8 | CRLF | RW | Mem 49%



$$\Delta_d = \frac{\sum_{i=1}^{n} m_i \Delta_i^2}{\sum_{i=1}^{n} m_i \Delta_i}$$

$$m_{eff} = \frac{\sum_{i=1}^{n} m_i \Delta_i}{\Delta_d}$$

$$k_{eff} = \frac{\sum_{i=1}^{n} k_i \Delta_i}{\Delta_d}$$

$$T_{eff} = \frac{2\pi}{\sqrt{\frac{k_{eff}}{m_{eff}}}}$$

CYCLIC DISPLACEMENT ANALYSIS

Spyder (Python 3.12)

File Edit Search Source Run Debug Consoles Projects Tools View Help

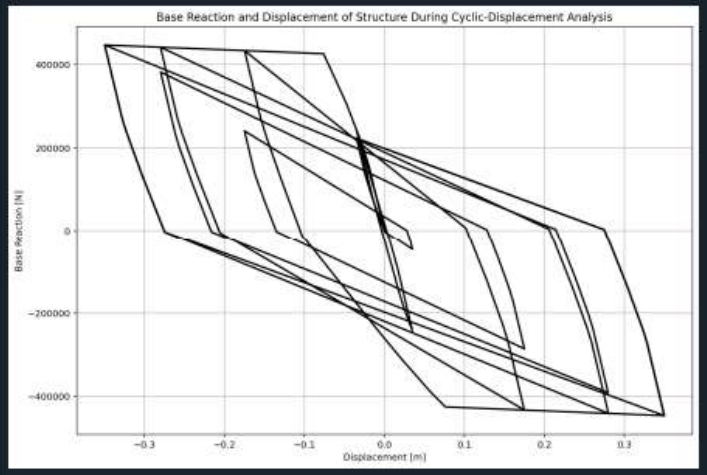
C:\Users\Dell

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...TY_INDEX\33_MDOF_TWO_8_ANA\P(t)_MDOF_TWO_8_ANA.py

P(t)_MDOF_TWO_8_ANA.py | COMPUTE_EFFECTIVE_PROPERTIES_FUN.py

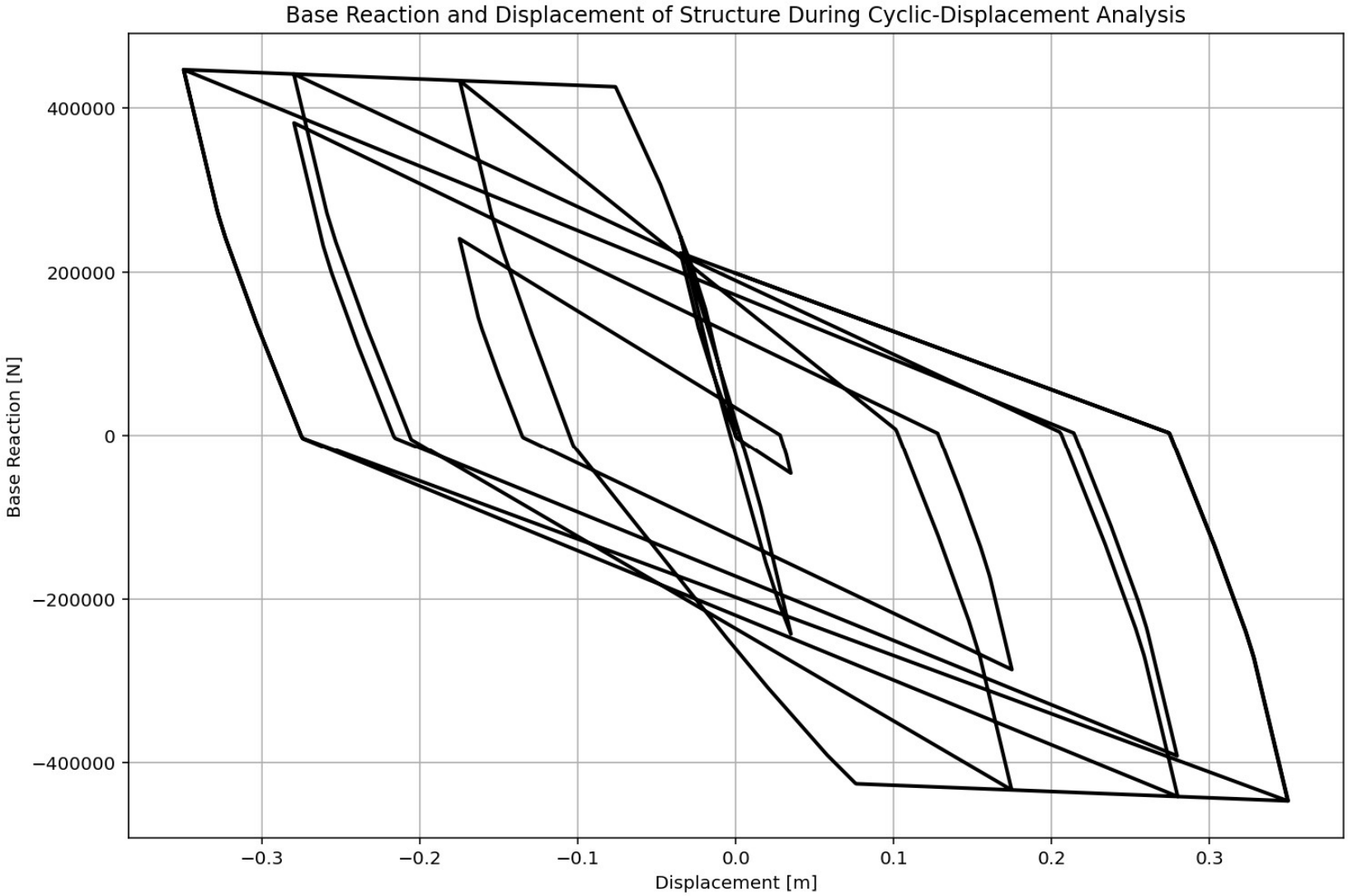
```python

    if ANAL_TYPE == 'CYCLIC_DISPLACEMENT': # STATIC TIME-HISTORY ANALYSIS
        IDctrlDOF = 1   # 1: Horizental Dispalcement - 2: Vertical Dispalcement
        DMAX = -0.35    # [m] Max. Displacement
        n_points = 10000
        # 4. CYCLIC DISPALCEMENT PROTOCOL
        # Key strain points (same logic as your protocol)
        key_disp = np.array([
             0.0,
             0.1*DMAX,   -0.1*DMAX,
             0.5*DMAX,   -0.5*DMAX,
             0.8*DMAX,   -0.8*DMAX,
             DMAX,       -DMAX,
             0.1*DMAX,     -0.1*DMAX,
             0.5*DMAX,     -0.5*DMAX,
             0.8*DMAX,     -0.8*DMAX,
             DMAX,       -DMAX,
             0.0
        ])

        # Generate 1000-point displacement protocol
        disp_protocol = np.interp(
            np.linspace(0, len(key_disp) - 1, n_points),
            np.arange(len(key_disp)),
            key_disp
        )

        ops.analysis('Static')
        STEP = 0.0
        for target_disp in disp_protocol:
            current_disp = ops.nodeDisp(center_node, 1) # DISPALCEMENT APPLIED IN MIDDLE N
            dU = target_disp - current_disp
            ops.integrator('DisplacementControl', center_node, IDctrlDOF, dU)
            OK = ops.analyze(1)
```

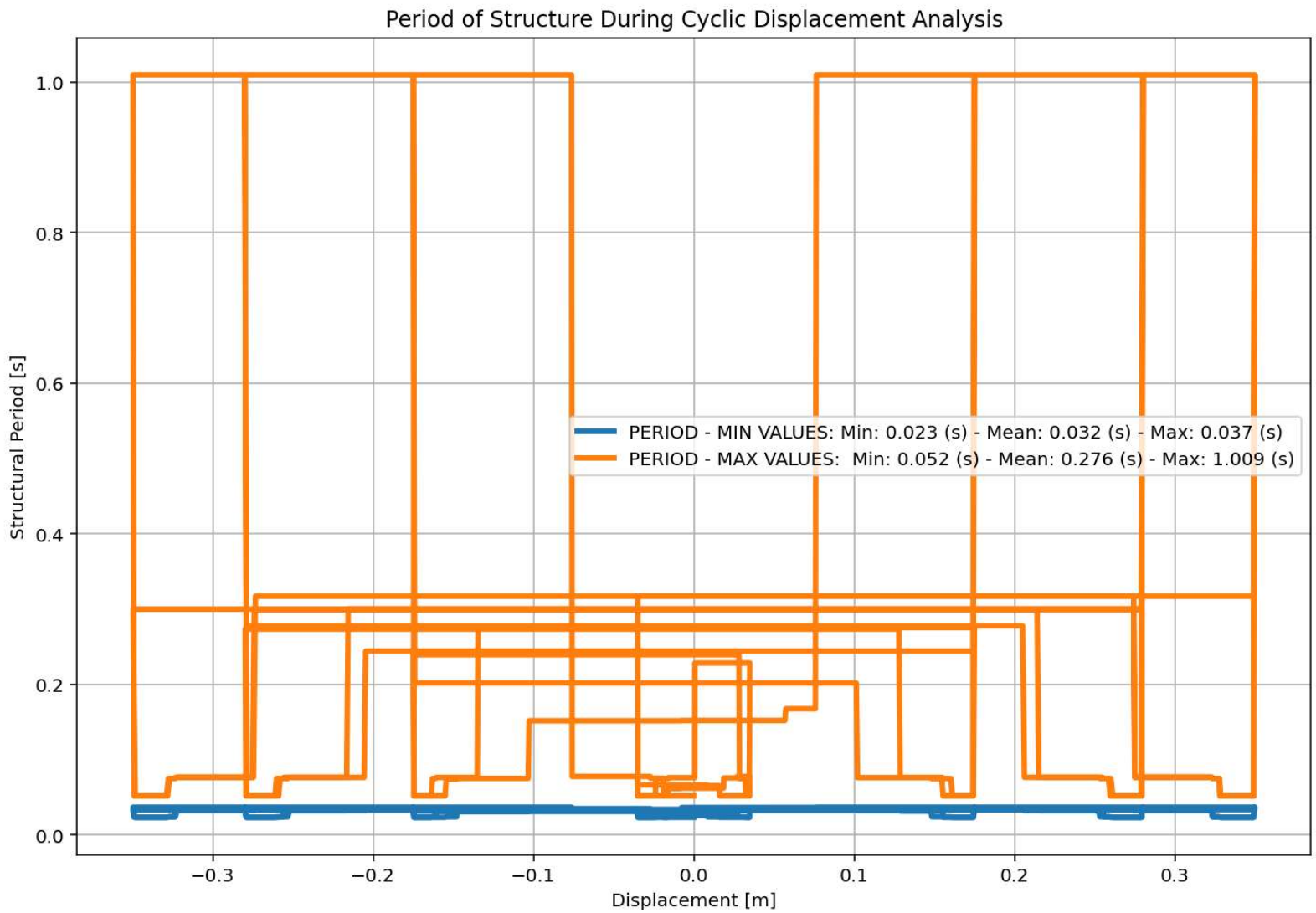
30 %

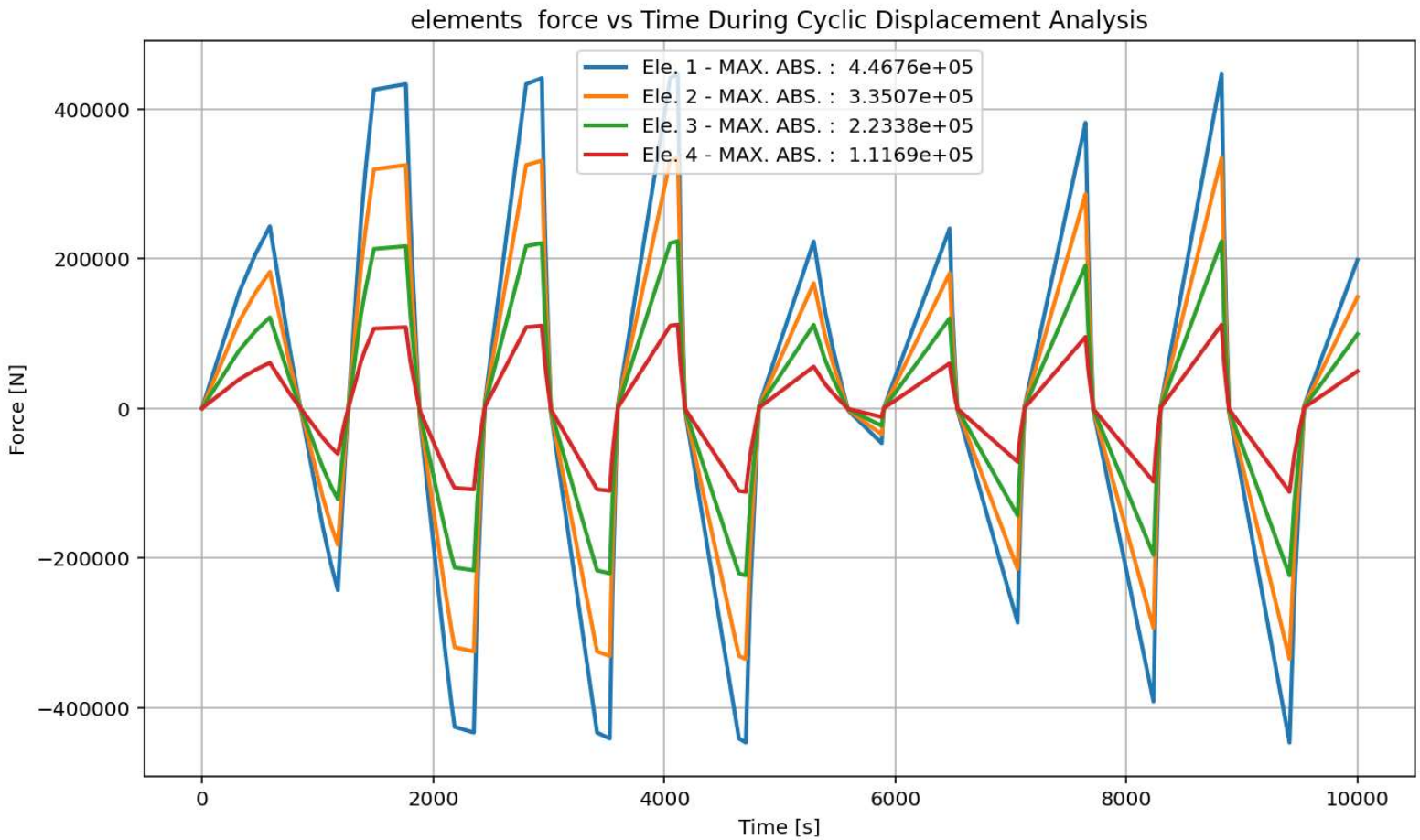


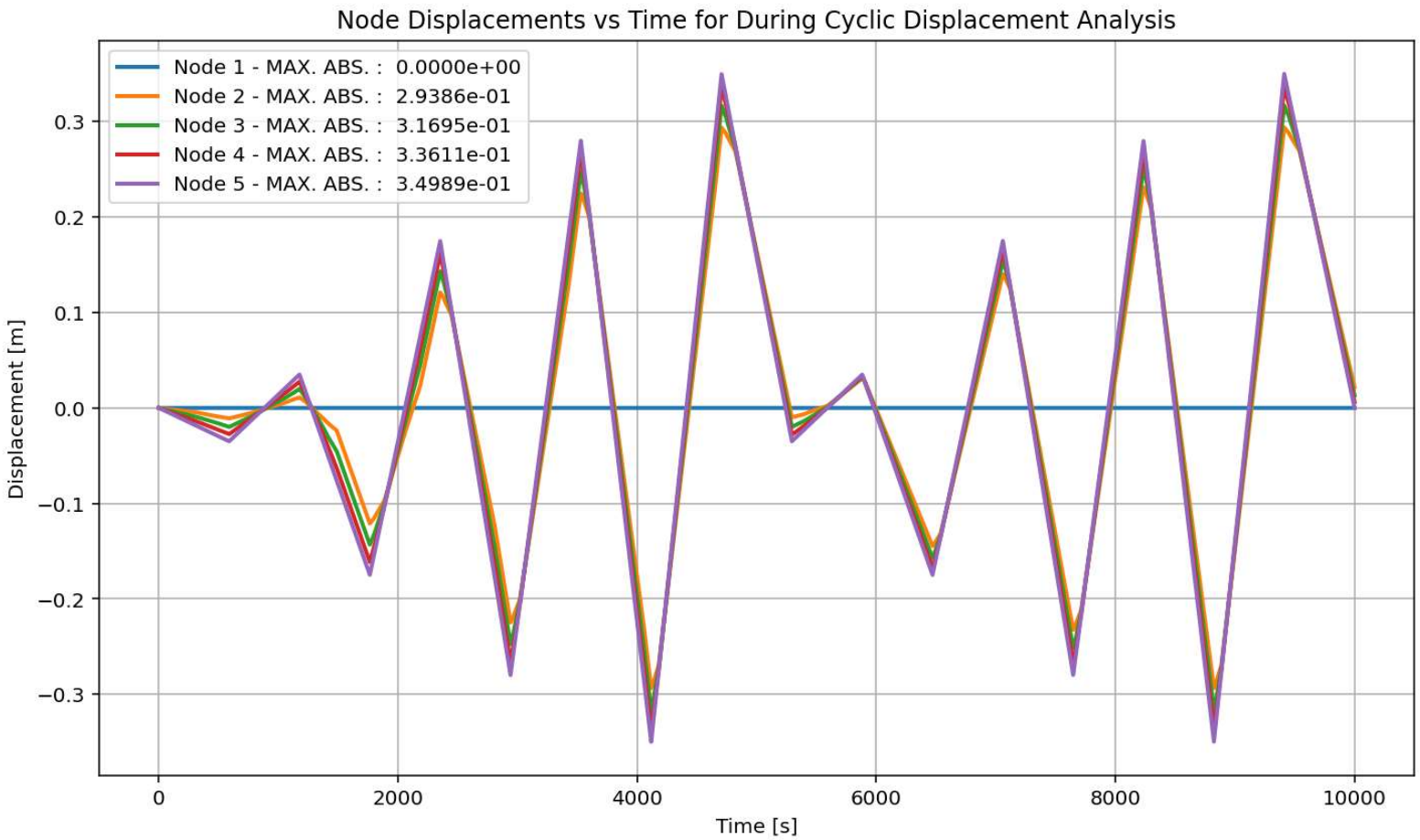
IPython Console | Files | Help | Variable Explorer | Debugger | Plots | History

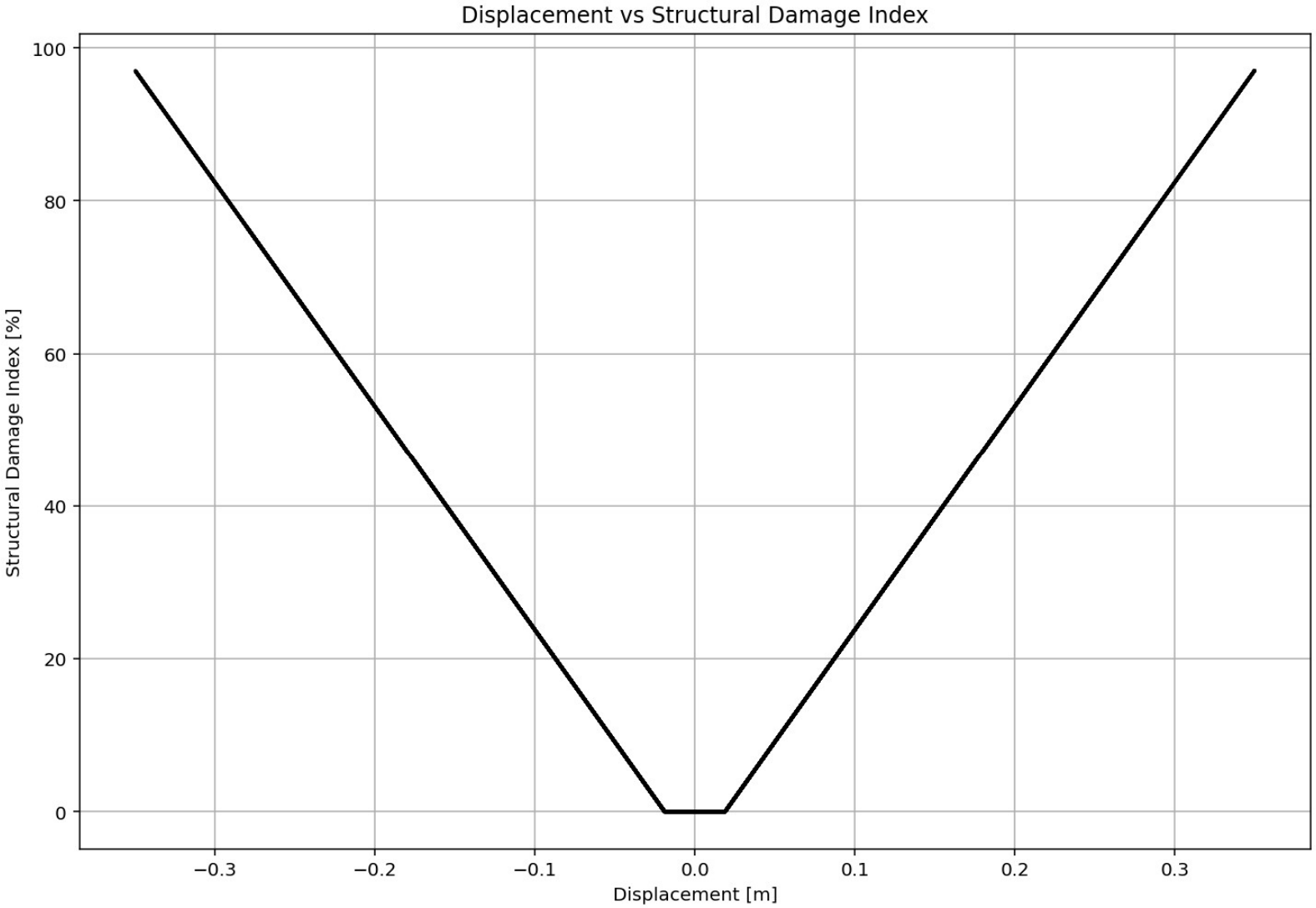
Inline Conda: anaconda3 (Python 3.12.7) LSP: Python Line 198, Col 94 UTF-8 CRLF RW Mem 49%

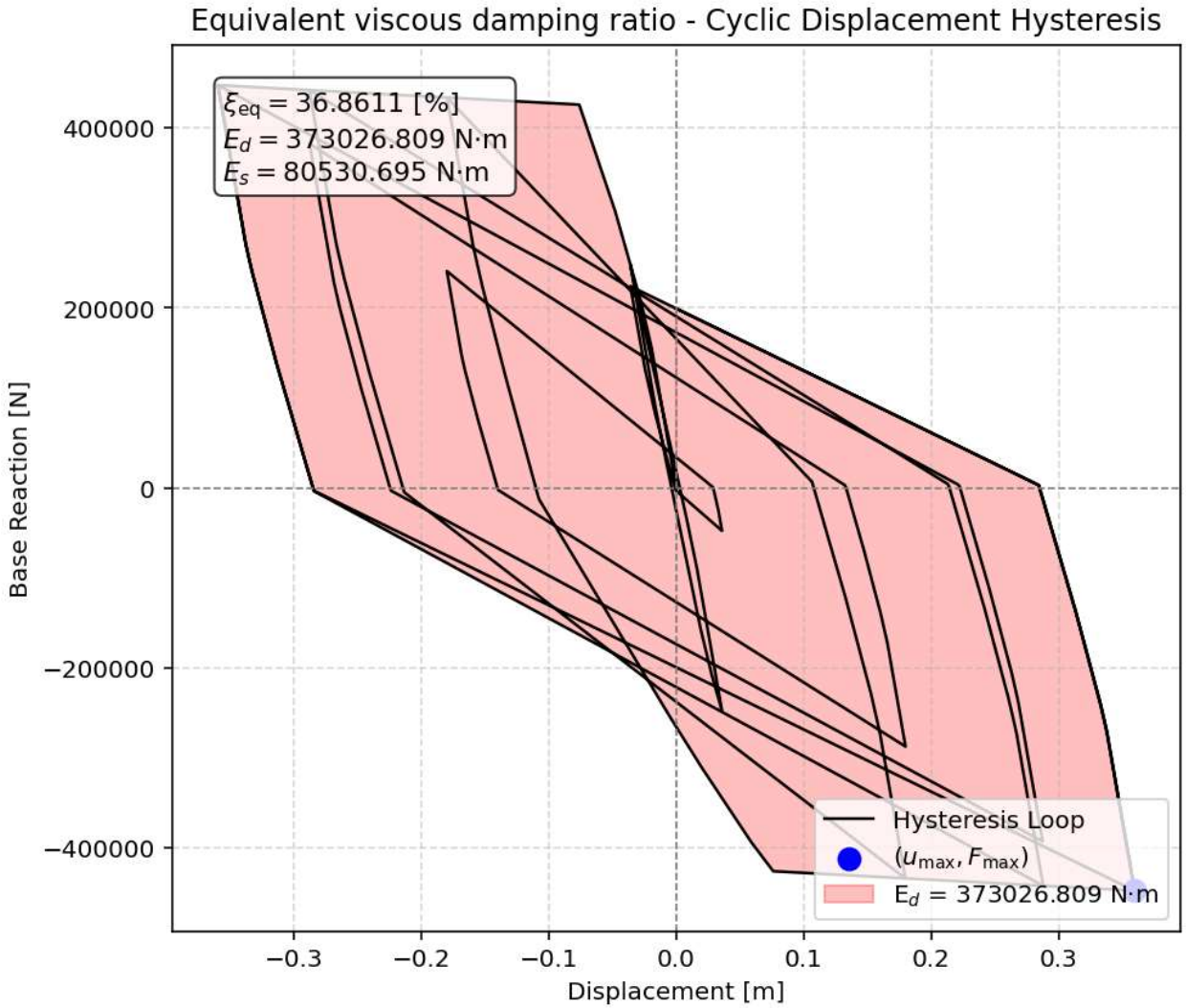

Base Reaction and Displacement of Structure During Cyclic-Displacement Analysis
400000
200000
0
−200000
−400000
Base Reaction [N]
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Displacement [m]


Period of Structure During Cyclic Displacement Analysis
PERIOD - MIN VALUES: Min: 0.023 (s) - Mean: 0.032 (s) - Max: 0.037 (s)
PERIOD - MAX VALUES: Min: 0.052 (s) - Mean: 0.276 (s) - Max: 1.009 (s)
Structural Period [s]
Displacement [m]
1.0
0.8
0.6
0.4
0.2
0.0
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3


elements force vs Time During Cyclic Displacement Analysis
Ele. 1 - MAX. ABS. : 4.4676e+05
Ele. 2 - MAX. ABS. : 3.3507e+05
Ele. 3 - MAX. ABS. : 2.2338e+05
Ele. 4 - MAX. ABS. : 1.1169e+05
400000
200000
0
−200000
−400000
Force [N]
0
2000
4000
6000
8000
10000
Time [s]


Node Displacements vs Time for During Cyclic Displacement Analysis
Node 1 - MAX. ABS. : 0.0000e+00
Node 2 - MAX. ABS. : 2.9386e-01
Node 3 - MAX. ABS. : 3.1695e-01
Node 4 - MAX. ABS. : 3.3611e-01
Node 5 - MAX. ABS. : 3.4989e-01
Displacement [m]
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
2000
4000
6000
8000
10000
Time [s]


Displacement vs Structural Damage Index
Structural Damage Index [%]
100
80
60
40
20
0
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Displacement [m]


Equivalent viscous damping ratio - Cyclic Displacement Hysteresis
ξ_{eq} = 36.8611 [%]
E_d = 373026.809 N·m
E_s = 80530.695 N·m
Hysteresis Loop
(u_{max}, F_{max})
E_d = 373026.809 N·m
Base Reaction [N]
Displacement [m]
400000
200000
0
−200000
−400000
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3

STATIC TIME-HISTORY WITH EXTERNAL TIME-DEPENDENT LOADING ANALYSIS

Spyder (Python 3.12)

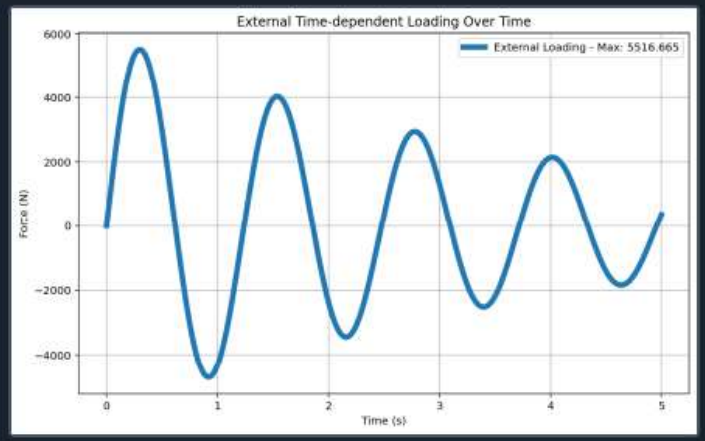
File Edit Search Source Run Debug Consoles Projects Tools View Help

...GE_INDEX_&_ENERGY_DISSIPATION_CAPACITY_INDEX\22_MDOF_8_ANA

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...N_CAPACITY_INDEX\22_MDOF_8_ANA\P(t)_MDOF_8_ANA.py

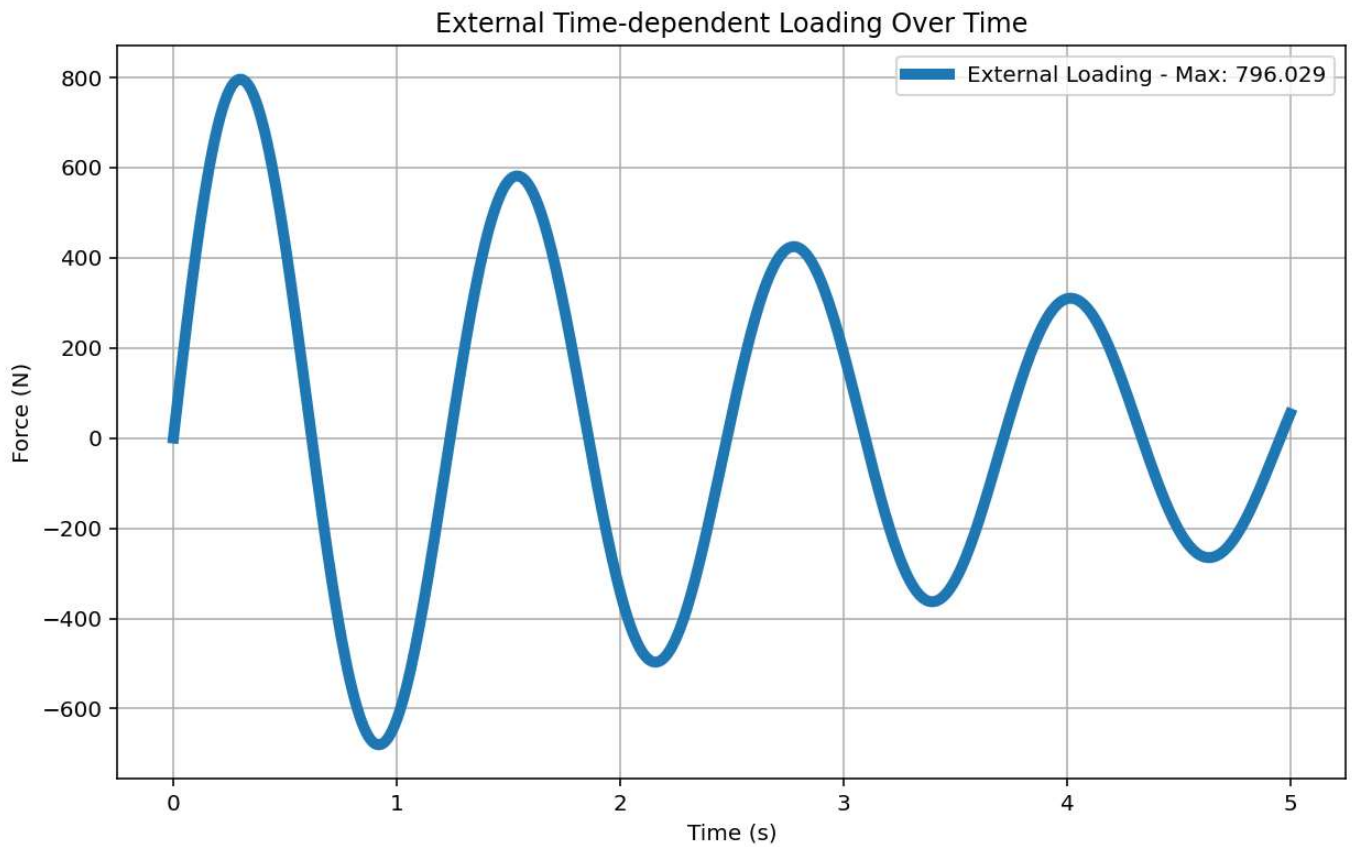
P(t)_MDOF_8_ANA.py

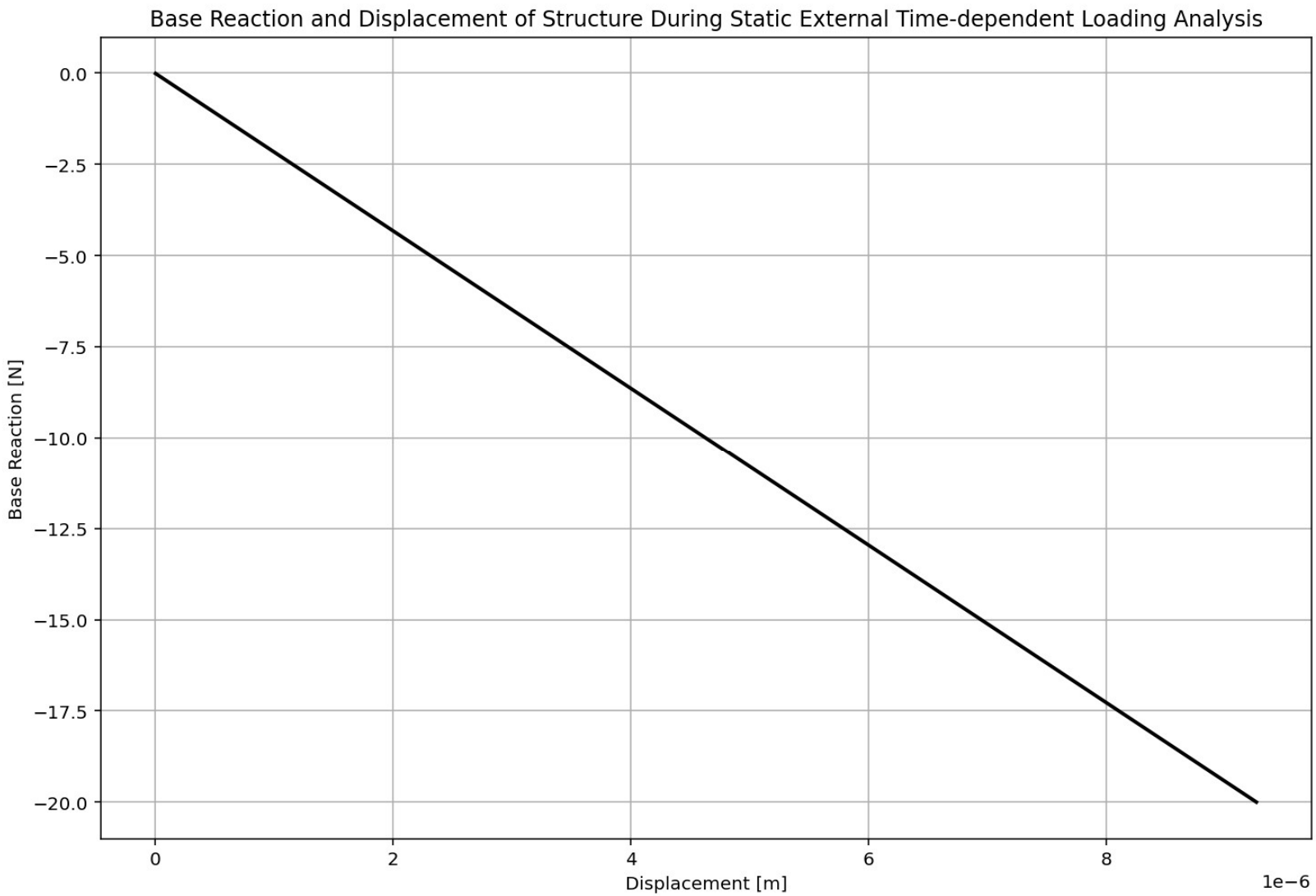
```python
    if ANAL_TYPE == 'STATIC_EXTERNAL_TIME-DEPENDENT_LOADING': # STATIC TIME-HISTORY ANALYS
        #%% DEFINE EXTERNAL TIME-DEPENDENT LOADING PROPERTIES
        # IN HERE ANALYSIS TIME AND DURATION ARE LOAD STEPS
        duration = 20.0             # [s] Analysis duration
        dt = 0.01                   # [s] Time step
        DT = dt                     # [s] Time step
        DT_time = 5.0               # [s] Total external Load Analysis Durations [*******]
        force_amplitude = 5960.0    # [N] Amplitude Force
        omega_DT = 5.0715           # [rad/s] Natural angular frequency
        time_steps = int(duration/dt)

        # Check function
        def CHECK_FUN(DT_time ,duration):
            if DT_time > duration:
                print('\n\nAnalysis Duration Must be greater than External Load Duration\n
                exit()
            return -1

        CHECK_FUN(DT_time ,duration)
        def EXTERNAL_TIME_DEPENDENT(force_amplitude, omega_DT, DT, DT_time): # P(t) = P0 s
            import numpy as np
            import matplotlib.pyplot as plt
            # External Load Durations
            num_steps = int(DT_time / DT)
            load_time = np.linspace(0, DT_time, num_steps)
            target_frequency = 1.0 * omega_DT  # Target excitation frequency
            DT_load = force_amplitude * np.sin(target_frequency * load_time)
            # Plot External Time-dependent Loading
            plt.figure(figsize=(10, 6))
            plt.plot(load_time, DT_load, label=f'External Loading - Max: {np.max(DT_load):
            plt.title('External Time-dependent Loading Over Time')
            plt.xlabel('Time (s)')
            plt.ylabel('Force (N)')
            plt.grid(True)
```

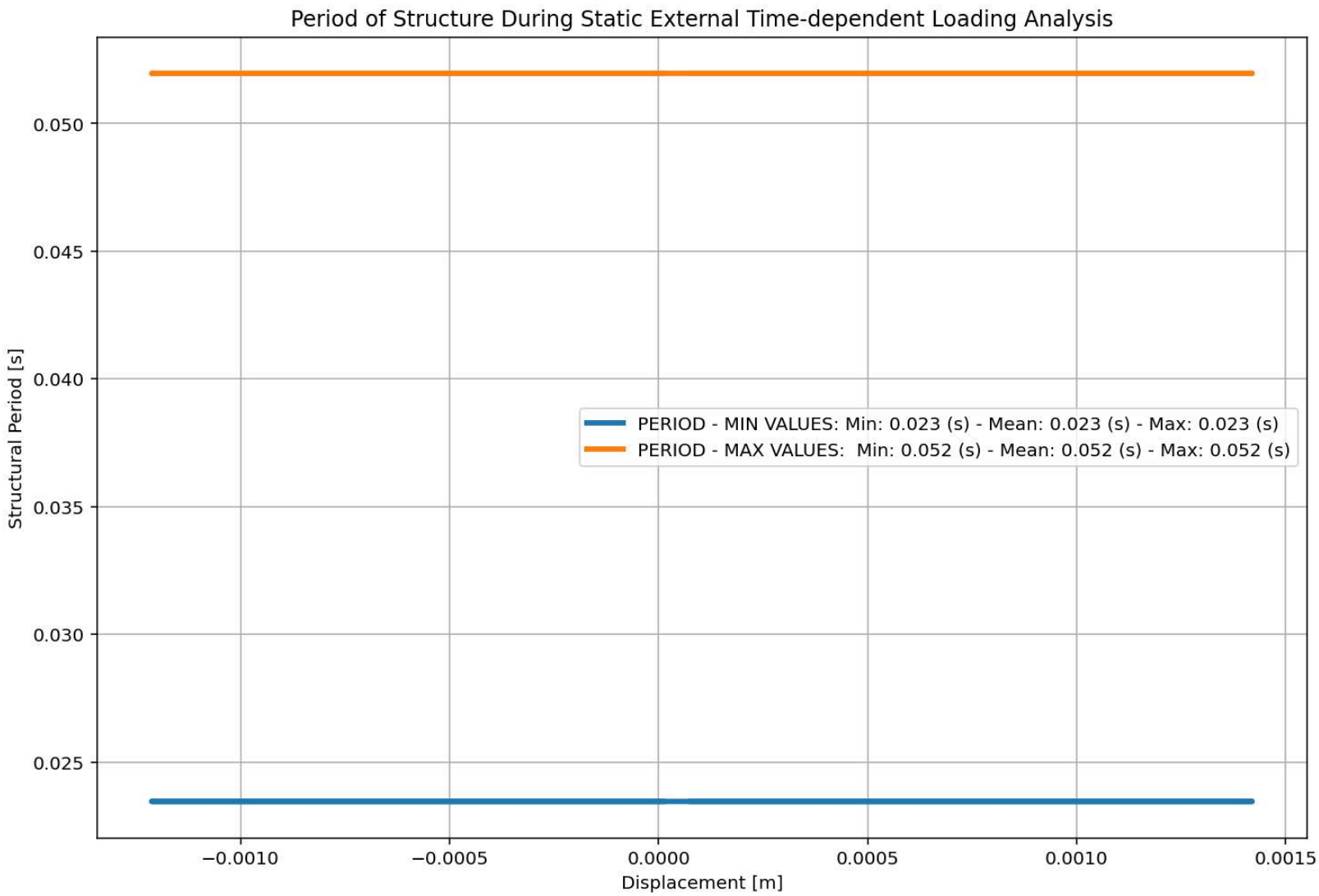


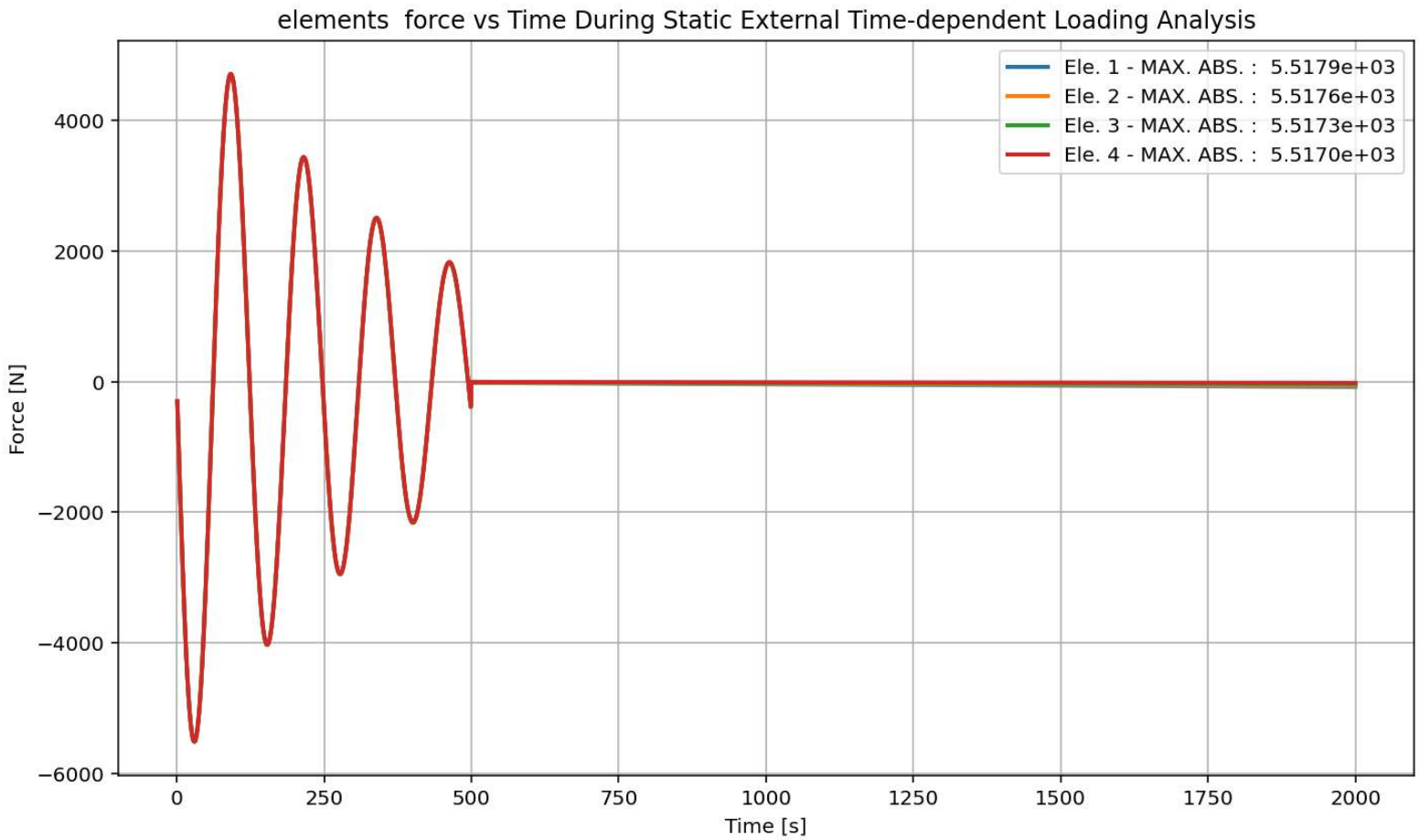
IPython Console | Files | Help | Variable Explorer | Debugger | Plots | History

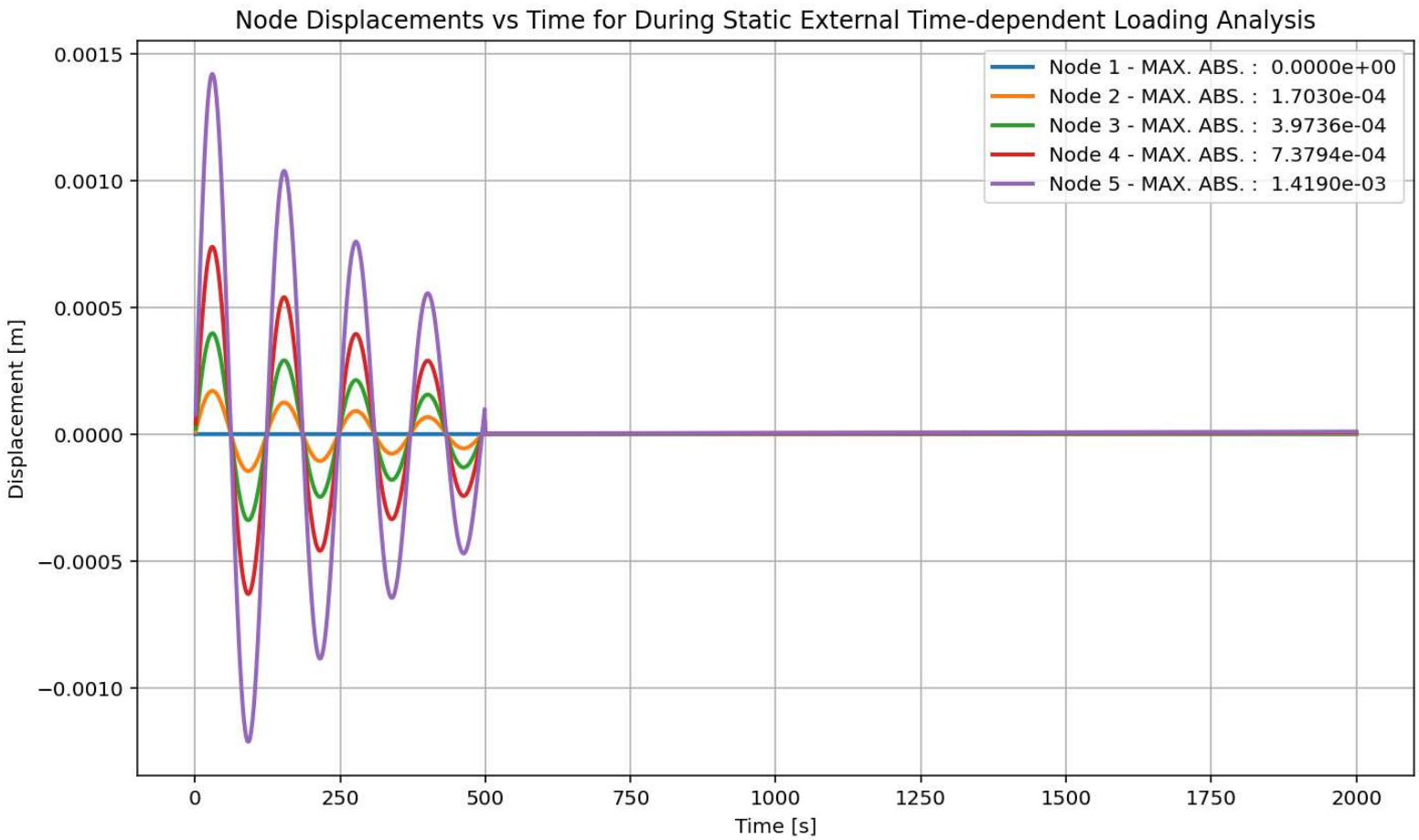
Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 1307, Col 23 | UTF-8 | CRLF | RW | Mem 45%

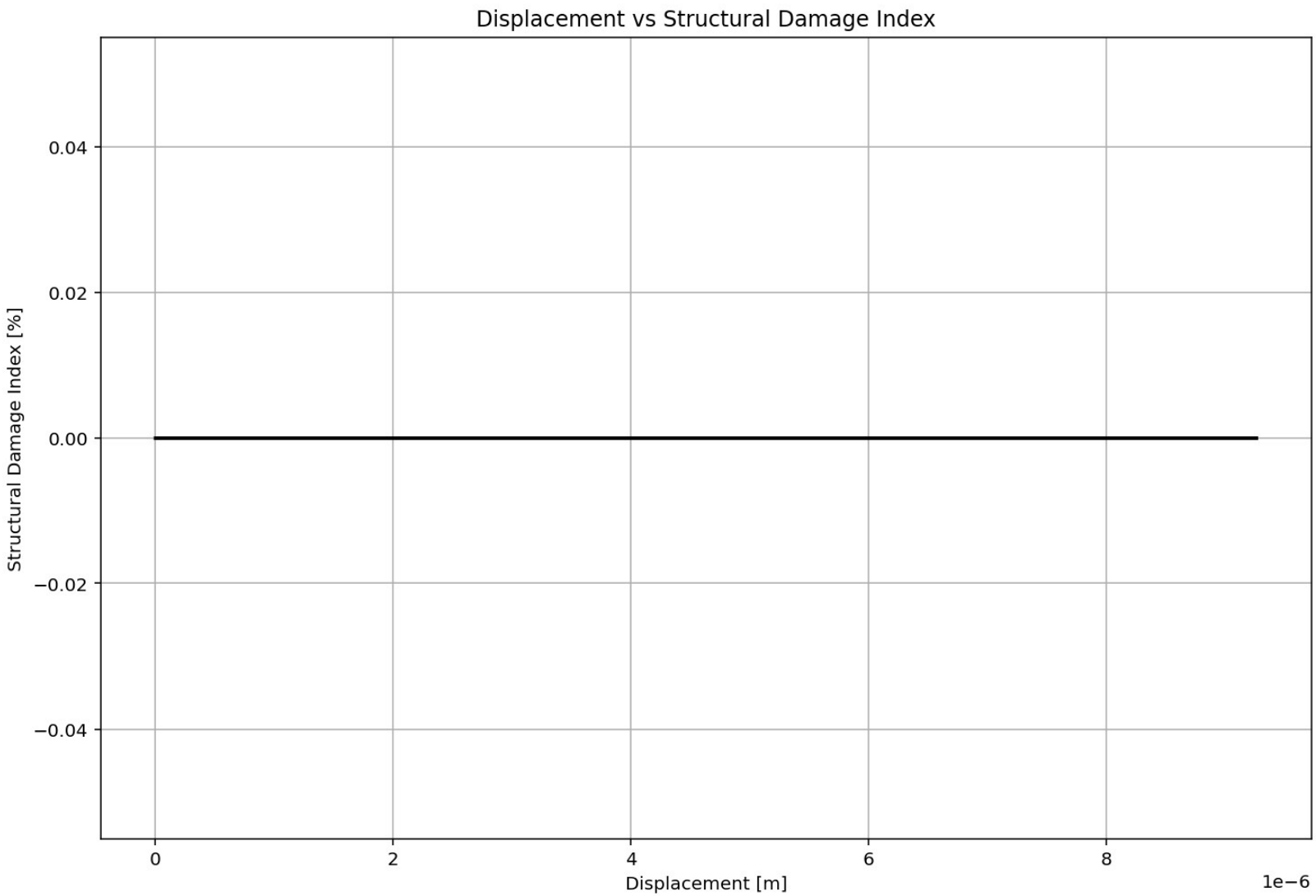

External Time-dependent Loading Over Time
External Loading - Max: 796.029
Force (N)
Time (s)
800
600
400
200
0
−200
−400
−600
0
1
2
3
4
5


Base Reaction and Displacement of Structure During Static External Time-dependent Loading Analysis
0.0
−2.5
−5.0
−7.5
−10.0
−12.5
−15.0
−17.5
−20.0
Base Reaction [N]
0
2
4
6
8
Displacement [m]
1e−6


Period of Structure During Static External Time-dependent Loading Analysis
Structural Period [s]
0.050
0.045
0.040
0.035
0.030
0.025
−0.0010
−0.0005
0.0000
0.0005
0.0010
0.0015
Displacement [m]
PERIOD - MIN VALUES: Min: 0.023 (s) - Mean: 0.023 (s) - Max: 0.023 (s)
PERIOD - MAX VALUES: Min: 0.052 (s) - Mean: 0.052 (s) - Max: 0.052 (s)


elements force vs Time During Static External Time-dependent Loading Analysis
Ele. 1 - MAX. ABS. : 5.5179e+03
Ele. 2 - MAX. ABS. : 5.5176e+03
Ele. 3 - MAX. ABS. : 5.5173e+03
Ele. 4 - MAX. ABS. : 5.5170e+03
Force [N]
4000
2000
0
−2000
−4000
−6000
0
250
500
750
1000
1250
1500
1750
2000
Time [s]


Node Displacements vs Time for During Static External Time-dependent Loading Analysis
Node 1 - MAX. ABS. : 0.0000e+00
Node 2 - MAX. ABS. : 1.7030e-04
Node 3 - MAX. ABS. : 3.9736e-04
Node 4 - MAX. ABS. : 7.3794e-04
Node 5 - MAX. ABS. : 1.4190e-03
Displacement [m]
0.0015
0.0010
0.0005
0.0000
−0.0005
−0.0010
0
250
500
750
1000
1250
1500
1750
2000
Time [s]


Displacement vs Structural Damage Index
Structural Damage Index [%]
0.04
0.02
0.00
−0.02
−0.04
0
2
4
6
8
Displacement [m]
1e−6

DYNAMIC TIME-HISTORY WITH EXTERNAL TIME-DEPENDENT LOADING ANALYSIS

Spyder (Python 3.12)

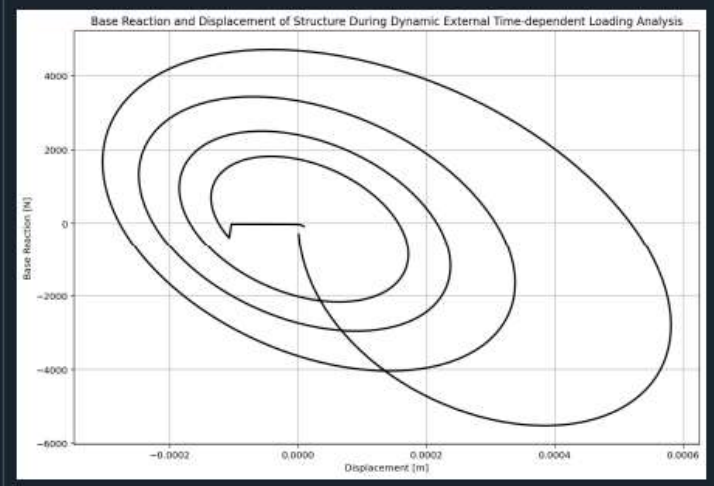
File Edit Search Source Run Debug Consoles Projects Tools View Help

...DEX_&_ENERGY_DISSIPATION_CAPACITY_INDEX\33_MDOF_TWO_8_ANA

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...TY_INDEX\33_MDOF_TWO_8_ANA\P(t)_MDOF_TWO_8_ANA.py

P(t)_MDOF_8_ANA.py | P(t)_MDOF_TWO_8_ANA.py

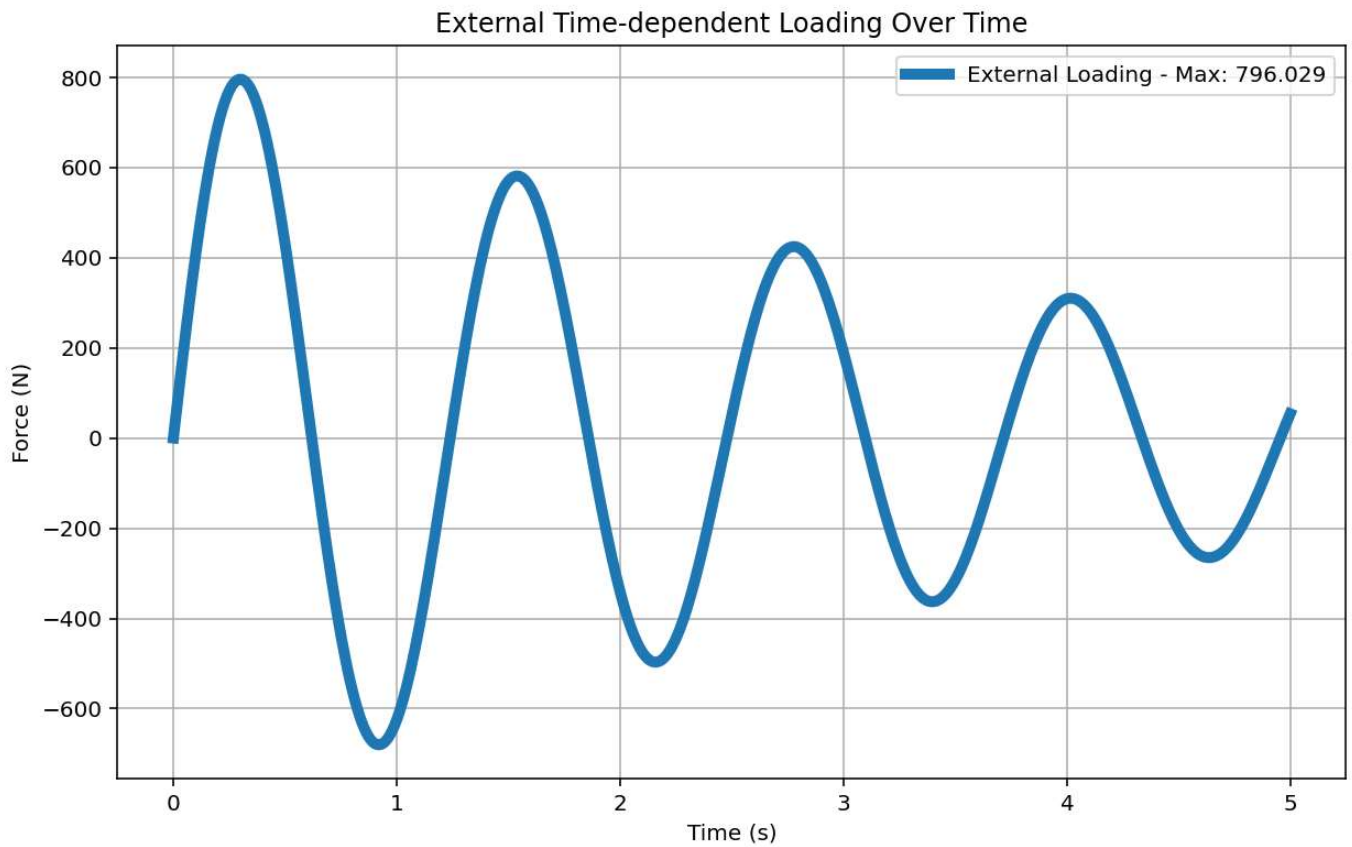
```
491
492     if ANAL_TYPE == 'DYNAMIC_EXTERNAL_TIME-DEPENDENT_LOADING': # DYNAMIC TIME-HISTORY ANAL
493         #%% DEFINE EXTERNAL TIME-DEPENDENT LOADING PROPERTIES
494         duration = 20.0             # [s] Analysis duration
495         dt = 0.01                   # [s] Time step
496         DT = dt                     # [s] Time step
497         DT_time = 5.0               # [s] Total external Load Analysis Durations [*******]
498         force_amplitude = 5960.0    # [N] Amplitude Force
499         omega_DT = 5.0715           # [rad/s] Natural angular frequency
500         DR = 0.03                   # Damping Ratio
501
502         # Check function
503         def CHECK_FUN(DT_time ,duration):
504             if DT_time > duration:
505                 print('\n\nAnalysis Duration Must be greater than External Load Duration\n
506                 exit()
507             return -1
508
509         CHECK_FUN(DT_time ,duration)
510         def EXTERNAL_TIME_DEPENDENT(force_amplitude, omega_DT, DT, DT_time): # P(t) = P0 s
511             import numpy as np
512             import matplotlib.pyplot as plt
513             # External Load Durations
514             num_steps = int(DT_time / DT)
515             load_time = np.linspace(0, DT_time, num_steps)
516             target_frequency = 1.0 * omega_DT  # Target excitation frequency
517             DT_load = force_amplitude * np.sin(target_frequency * load_time)
518             # Plot External Time-dependent Loading
519             plt.figure(figsize=(10, 6))
520             plt.plot(load_time, DT_load, label=f'External Loading - Max: {np.max(DT_load):
521             plt.title('External Time-dependent Loading Over Time')
522             plt.xlabel('Time (s)')
523             plt.ylabel('Force (N)')
524             plt.grid(True)
```

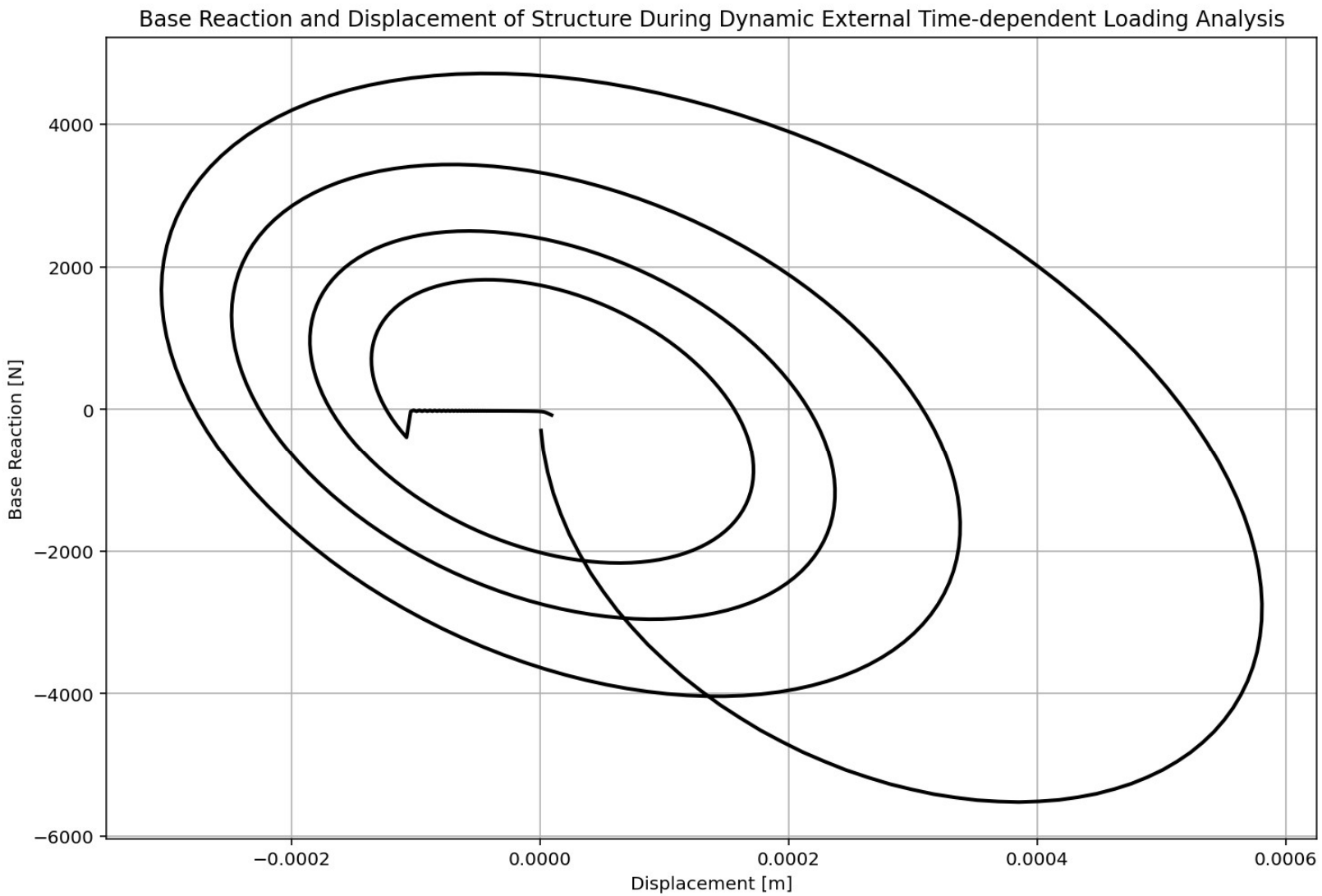


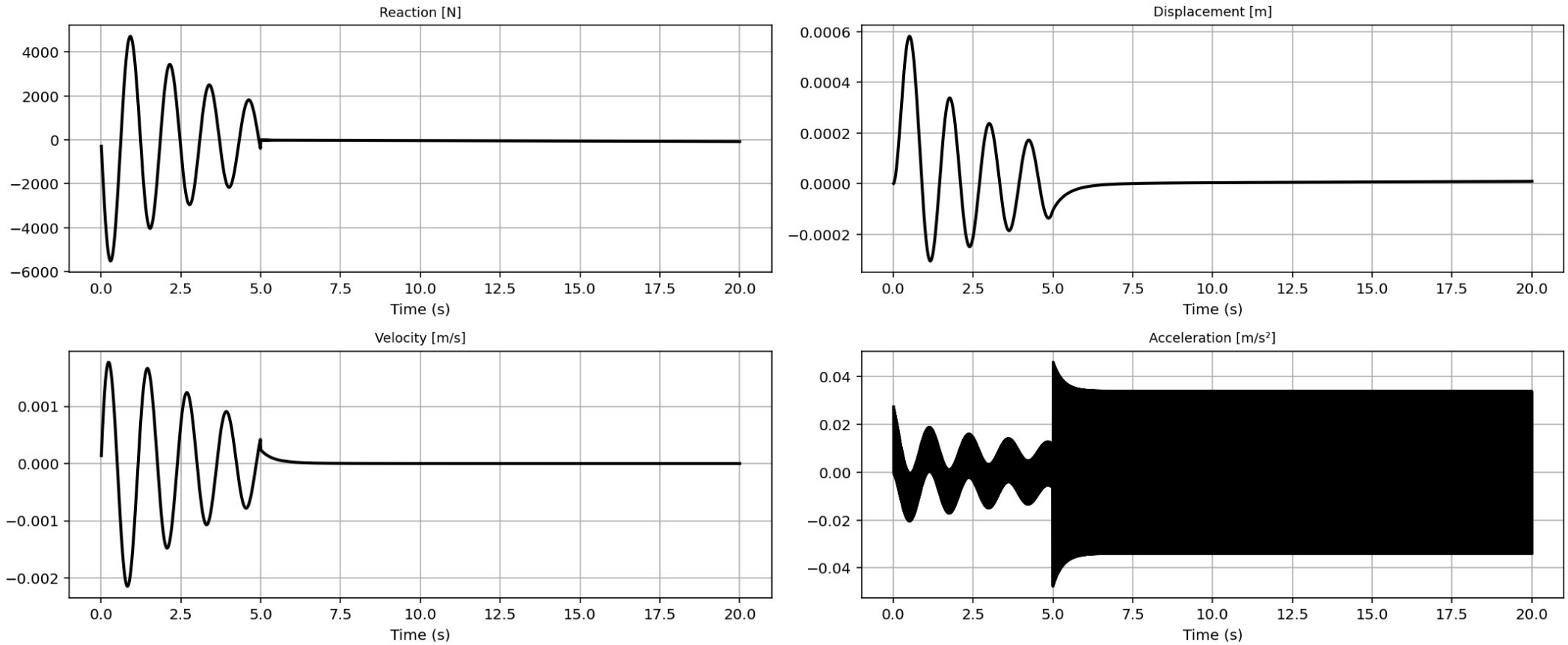
30 %

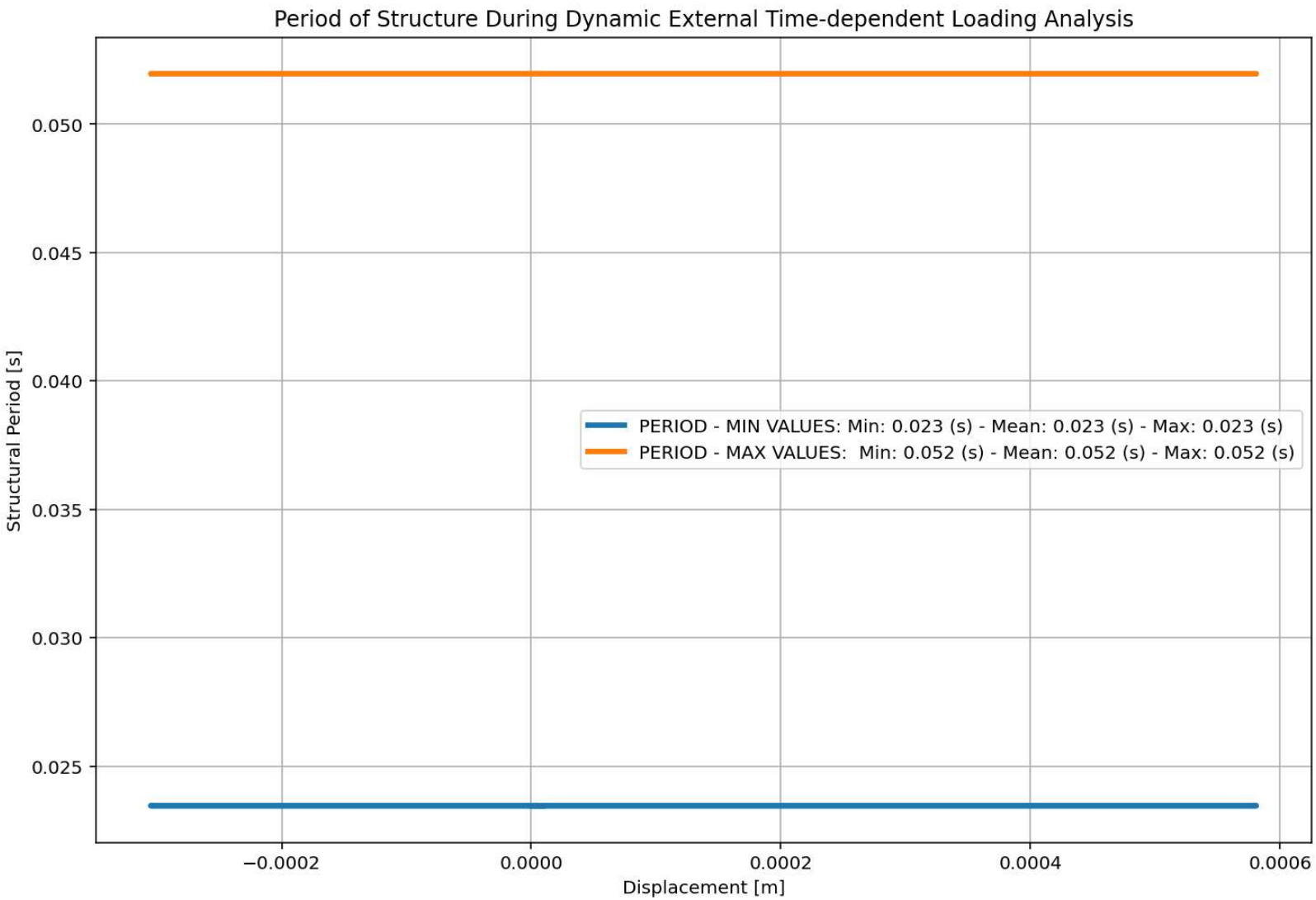
IPython Console | Files | Help | Variable Explorer | Debugger | Plots | History

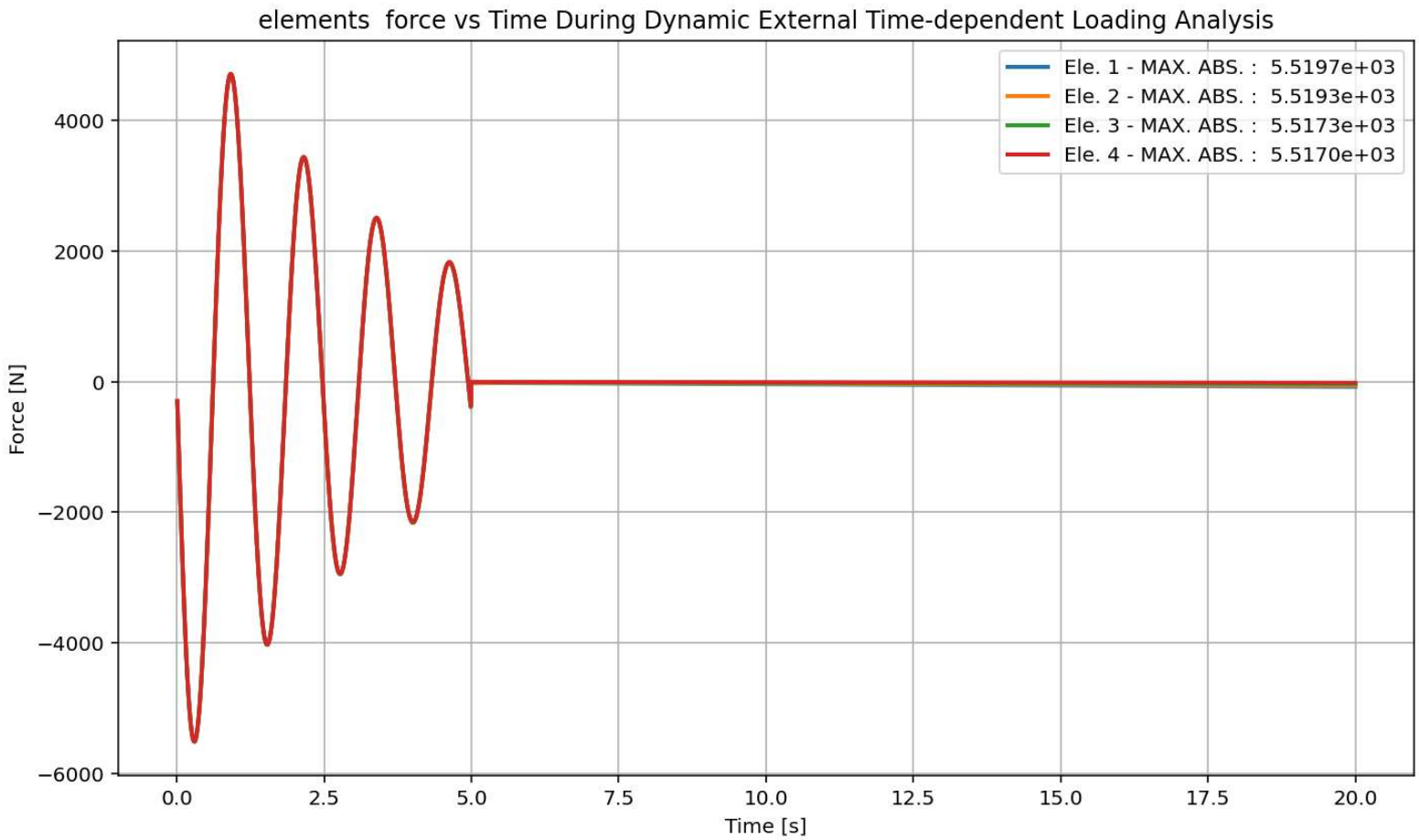
Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 169, Col 37 | UTF-8 | CRLF | RW | Mem 44%

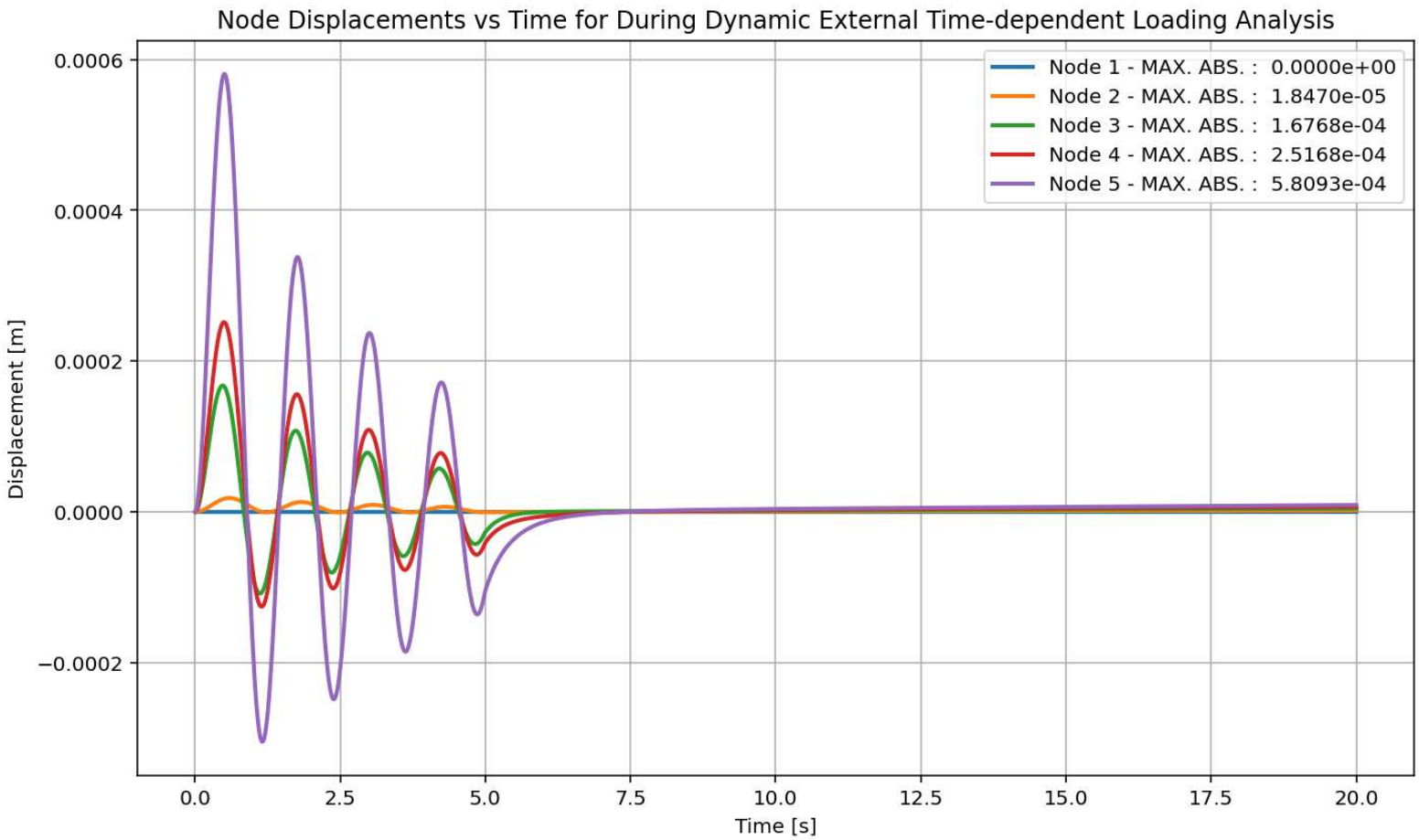

External Time-dependent Loading Over Time
External Loading - Max: 796.029
Force (N)
Time (s)
800
600
400
200
0
−200
−400
−600
0
1
2
3
4
5

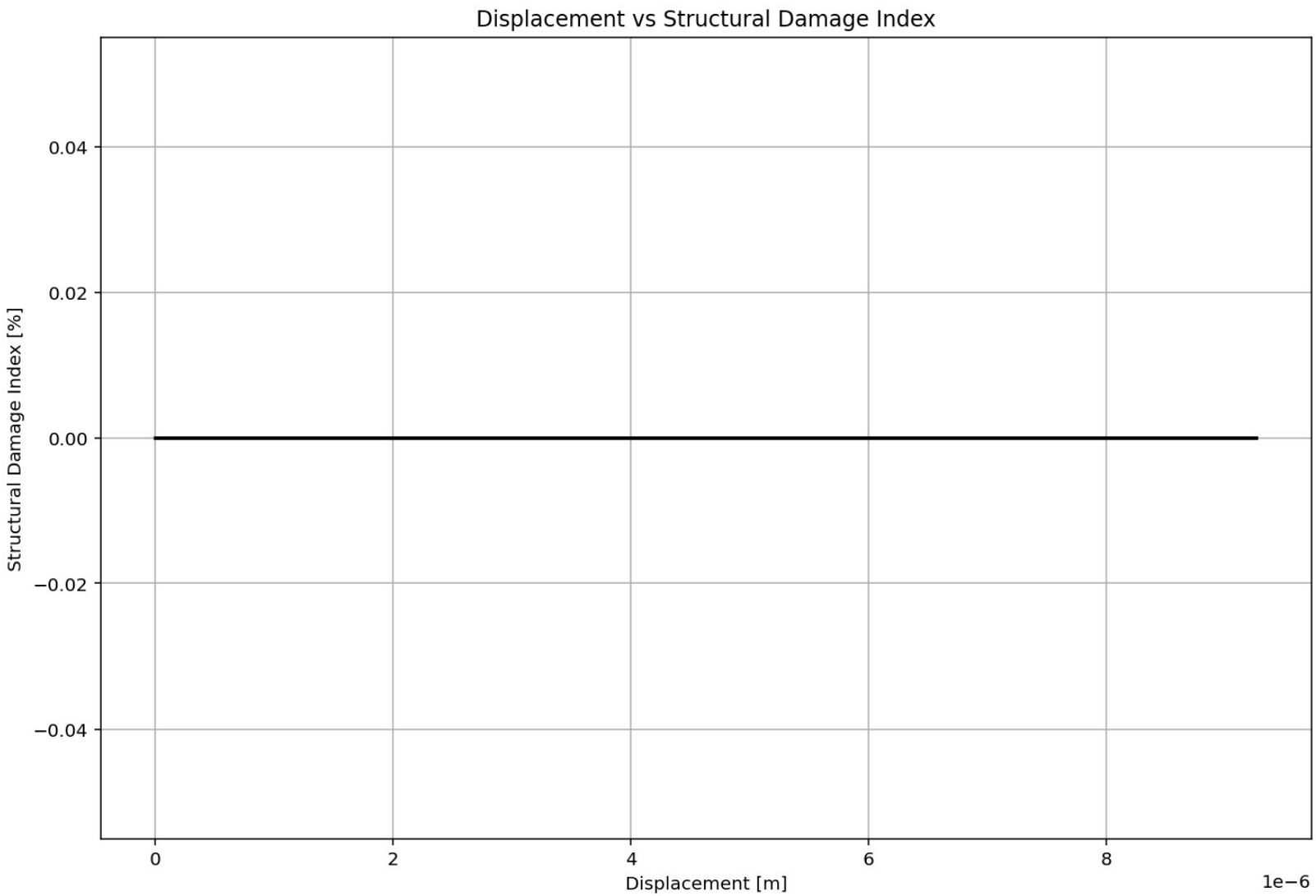

Base Reaction and Displacement of Structure During Dynamic External Time-dependent Loading Analysis
Base Reaction [N]
4000
2000
0
−2000
−4000
−6000
−0.0002
0.0000
0.0002
0.0004
0.0006
Displacement [m]


Reaction [N]
4000
2000
0
−2000
−4000
−6000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (s)
Displacement [m]
0.0006
0.0004
0.0002
0.0000
−0.0002
Time (s)
Velocity [m/s]
0.001
0.000
−0.001
−0.002
Time (s)
Acceleration [m/s²]
0.04
0.02
0.00
−0.02
−0.04
Time (s)


Period of Structure During Dynamic External Time-dependent Loading Analysis
Structural Period [s]
0.050
0.045
0.040
0.035
0.030
0.025
−0.0002
0.0000
0.0002
0.0004
0.0006
Displacement [m]
PERIOD - MIN VALUES: Min: 0.023 (s) - Mean: 0.023 (s) - Max: 0.023 (s)
PERIOD - MAX VALUES: Min: 0.052 (s) - Mean: 0.052 (s) - Max: 0.052 (s)


elements force vs Time During Dynamic External Time-dependent Loading Analysis
Ele. 1 - MAX. ABS. : 5.5197e+03
Ele. 2 - MAX. ABS. : 5.5193e+03
Ele. 3 - MAX. ABS. : 5.5173e+03
Ele. 4 - MAX. ABS. : 5.5170e+03
Force [N]
4000
2000
0
−2000
−4000
−6000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time [s]


Node Displacements vs Time for During Dynamic External Time-dependent Loading Analysis
Node 1 - MAX. ABS. : 0.0000e+00
Node 2 - MAX. ABS. : 1.8470e-05
Node 3 - MAX. ABS. : 1.6768e-04
Node 4 - MAX. ABS. : 2.5168e-04
Node 5 - MAX. ABS. : 5.8093e-04
Displacement [m]
0.0006
0.0004
0.0002
0.0000
−0.0002
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time [s]


Displacement vs Structural Damage Index
Structural Damage Index [%]
0.04
0.02
0.00
−0.02
−0.04
0
2
4
6
8
Displacement [m]
1e−6

FREE-VIBRATION ANALYSIS

Spyder (Python 3.12)

File Edit Search Source Run Debug Consoles Projects Tools View Help

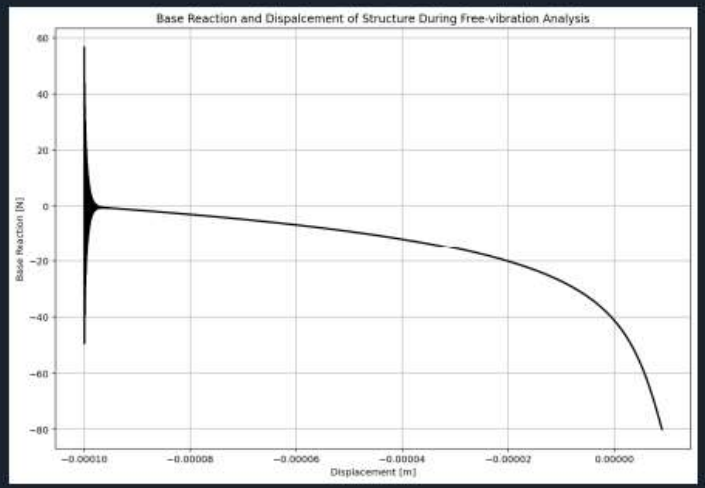
...DEX_&_ENERGY_DISSIPATION_CAPACITY_INDEX\33_MDOF_TWO_8_ANA

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...TY_INDEX\33_MDOF_TWO_8_ANA\P(t)_MDOF_TWO_8_ANA.py

P(t)_MDOF_TWO_8_ANA.py | COMPUTE_EFFECTIVE_..._FREE_VIBRATION.py

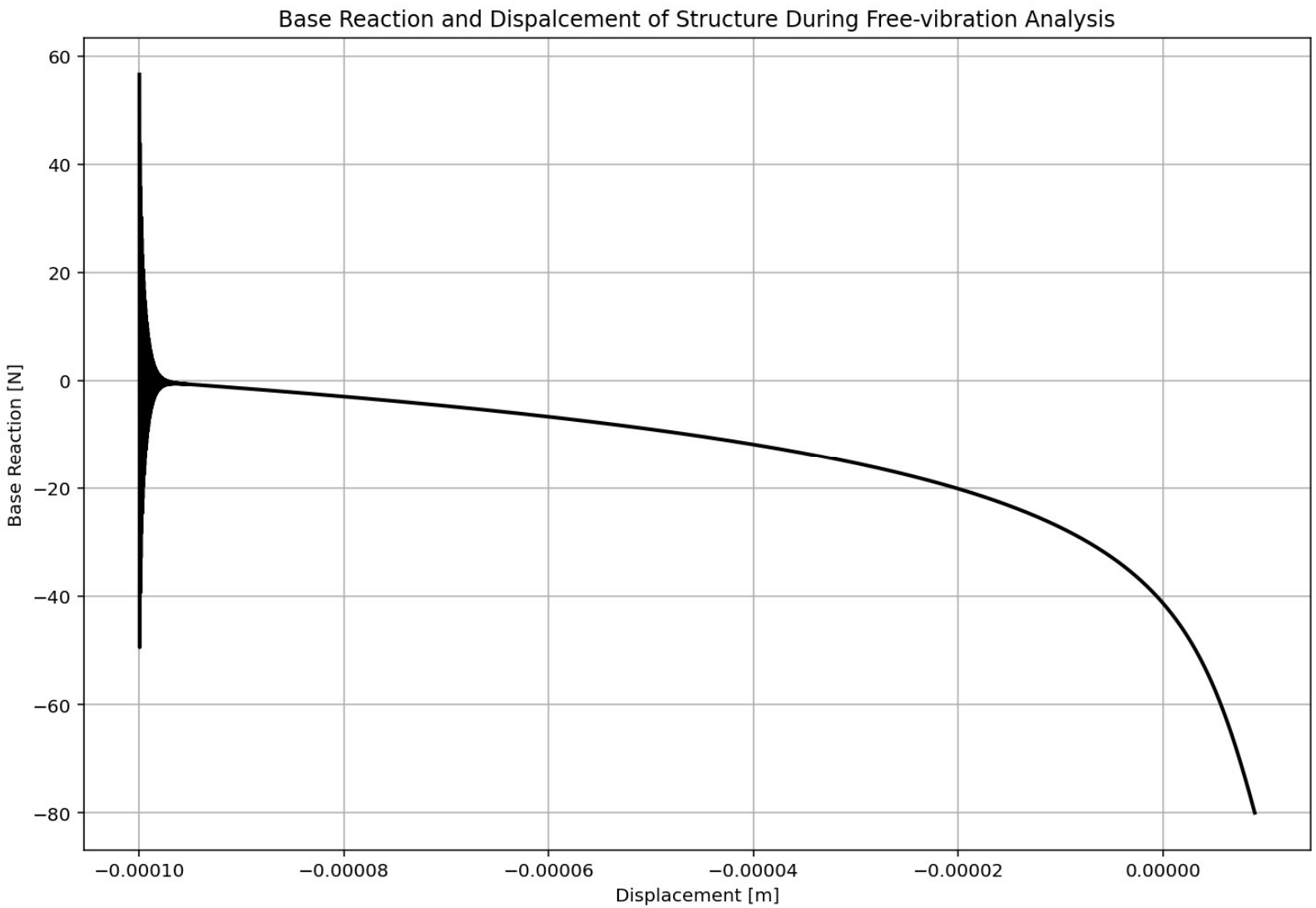
```python

    if ANAL_TYPE == 'FREE-VIBRATION': # DYNAMIC TIME-HISTORY ANALYSIS
        #%% DEFINE PARAMETERS FOR FREE-VIBRATION ANALYSIS
        u0 = -0.00010                       # [m] Initial displacement
        v0 = 0.000015                       # [m/s] Initial velocity
        a0 = 0.000065                       # [m/s^2] Initial acceleration
        IU = True                           # Free Vibration with Initial Displacement
        IV = True                           # Free Vibration with Initial Velocity
        IA = True                           # Free Vibration with Initial Acceleration
        duration = 20.0                     # [s] Analysis duration
        dt = 0.001                          # [s] Time step
        DR = 0.03                           # Damping Ratio

        disp_02, disp_03, disp_04, disp_05 = [], [], [], []
        velo_02, velo_03, velo_04, velo_05 = [], [], [], []
        accel_02, accel_03, accel_04, accel_05 = [], [], [], []

        if IU == True:
            # Define initial displacment
            ops.setNodeDisp(2, 1, u0, '-commit')
            ops.setNodeDisp(3, 1, u0, '-commit')
            ops.setNodeDisp(4, 1, u0, '-commit')
            ops.setNodeDisp(center_node, 1, u0, '-commit')
            # INFO LINK: https://openseespydoc.readthedocs.io/en/latest/src/setNodeDisp.ht
        if IV == True:
            # Define initial velocity
            ops.setNodeVel(2, 1, v0, '-commit')
            ops.setNodeVel(3, 1, v0, '-commit')
            ops.setNodeVel(4, 1, v0, '-commit')
            ops.setNodeVel(center_node, 1, v0, '-commit')
            # INFO LINK: https://openseespydoc.readthedocs.io/en/stable/src/setNodeVel.htm
        if IA == True:
            # Define initial  acceleration
            ops.setNodeAccel(2, 1, a0, '-commit')
```

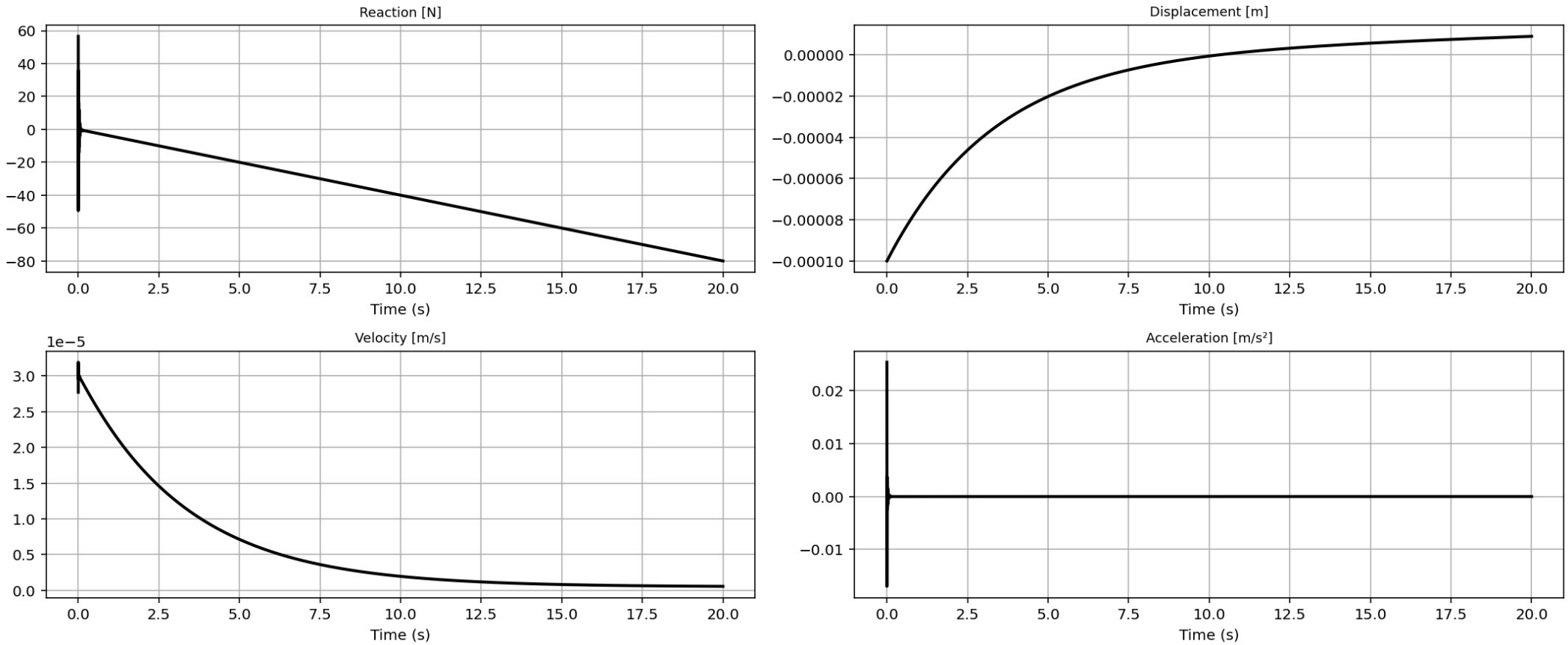
31 %

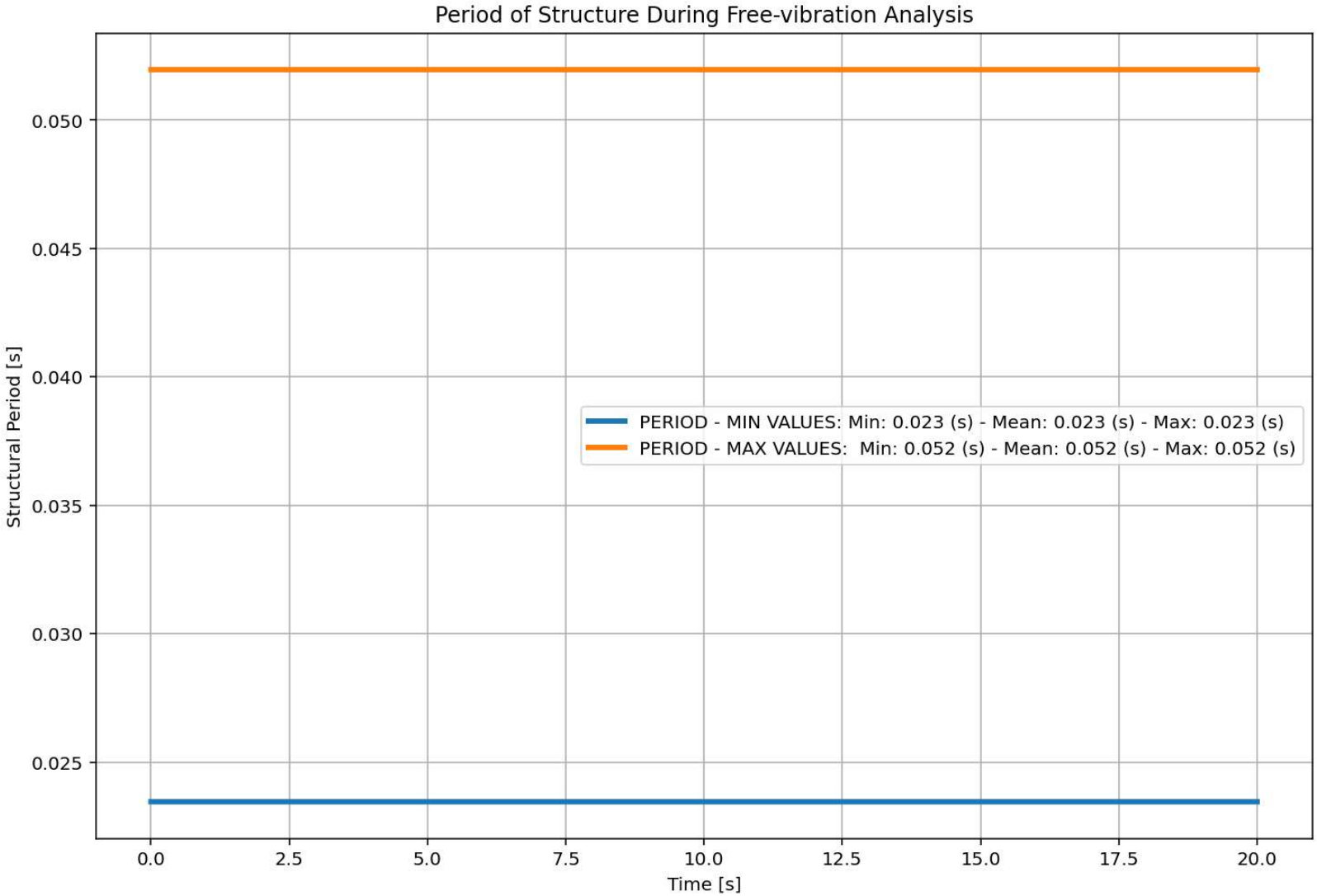


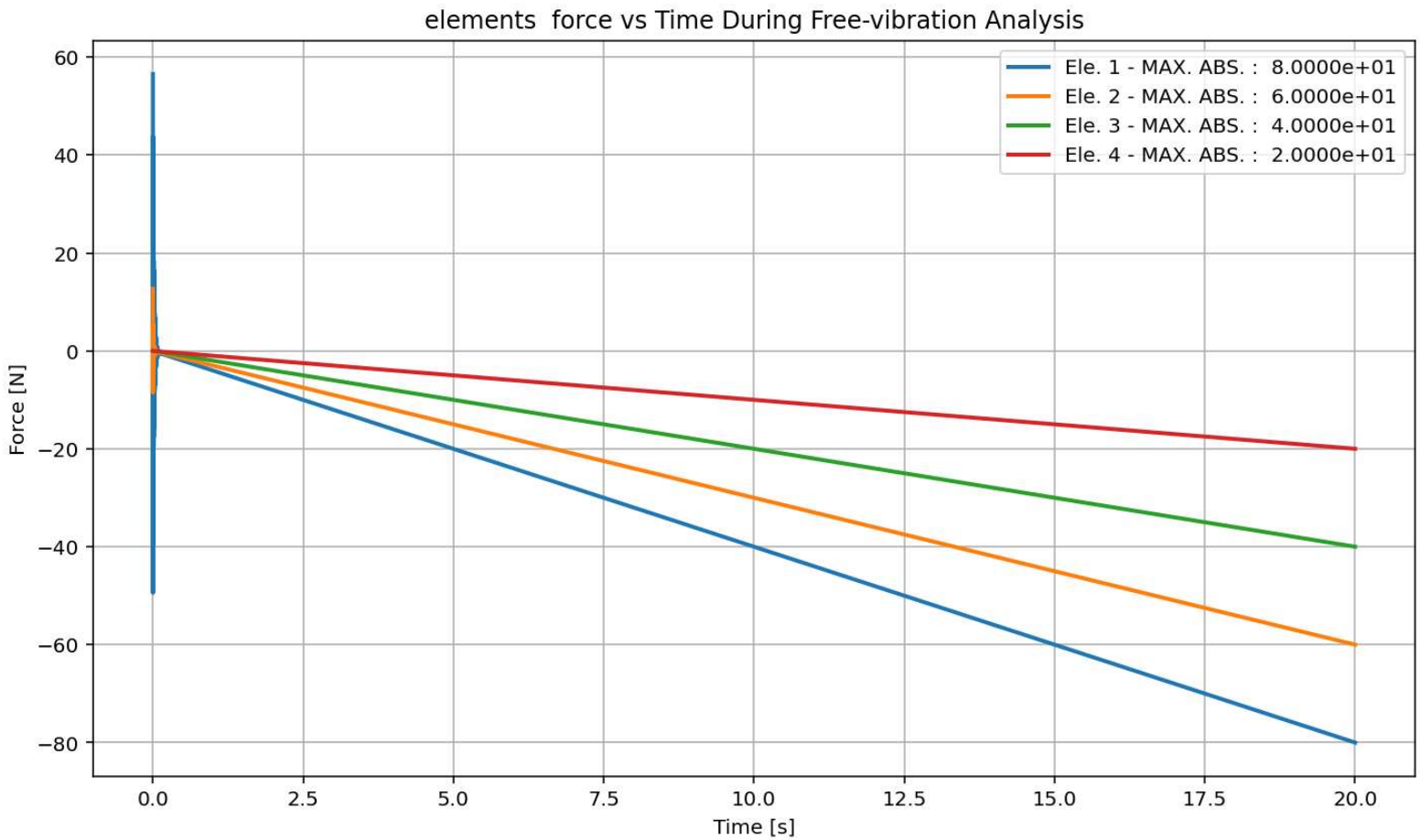
IPython Console | Files | Help | Variable Explorer | Debugger | Plots | History

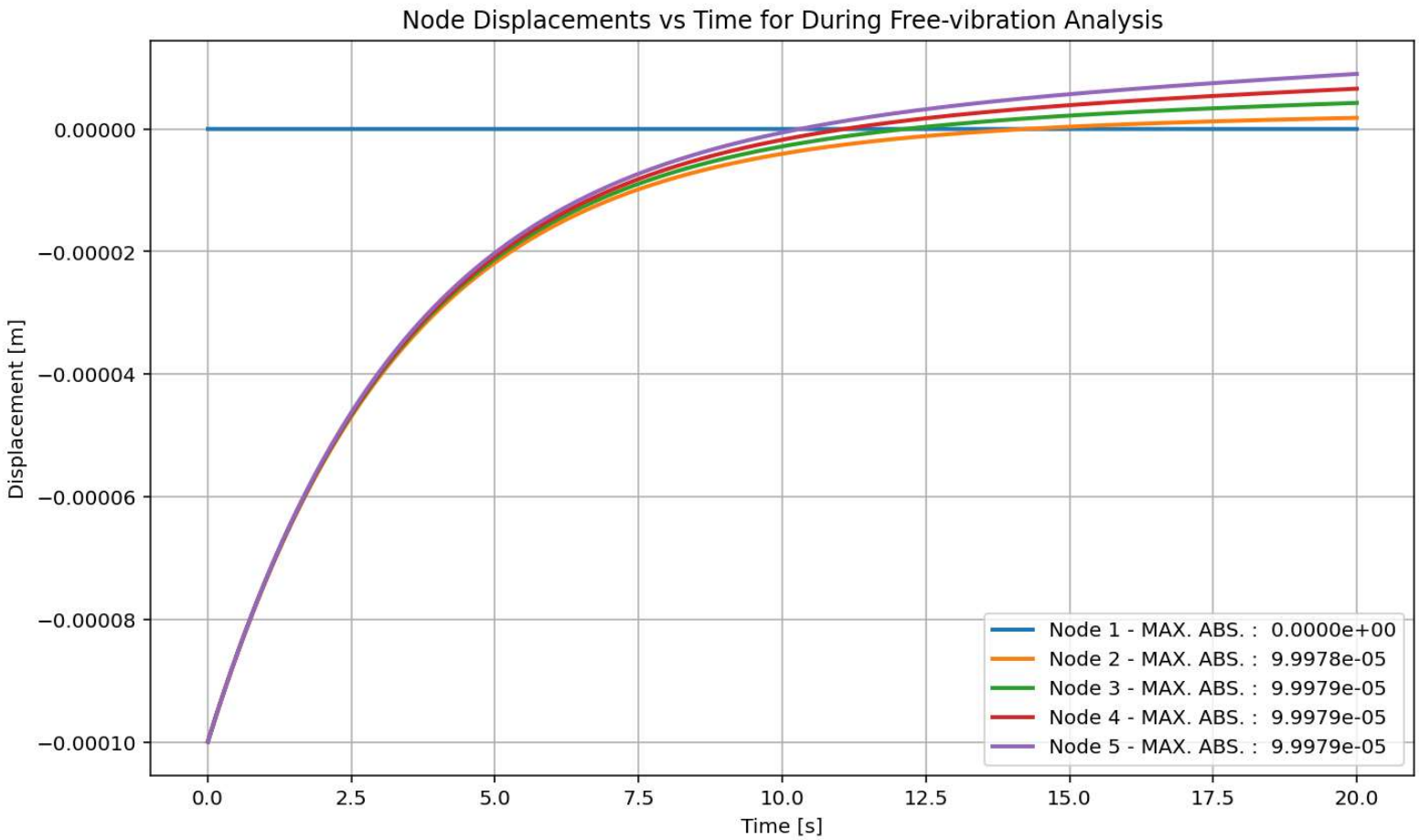
Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 1, Col 1 | UTF-8 | CRLF | RW | Mem 48%

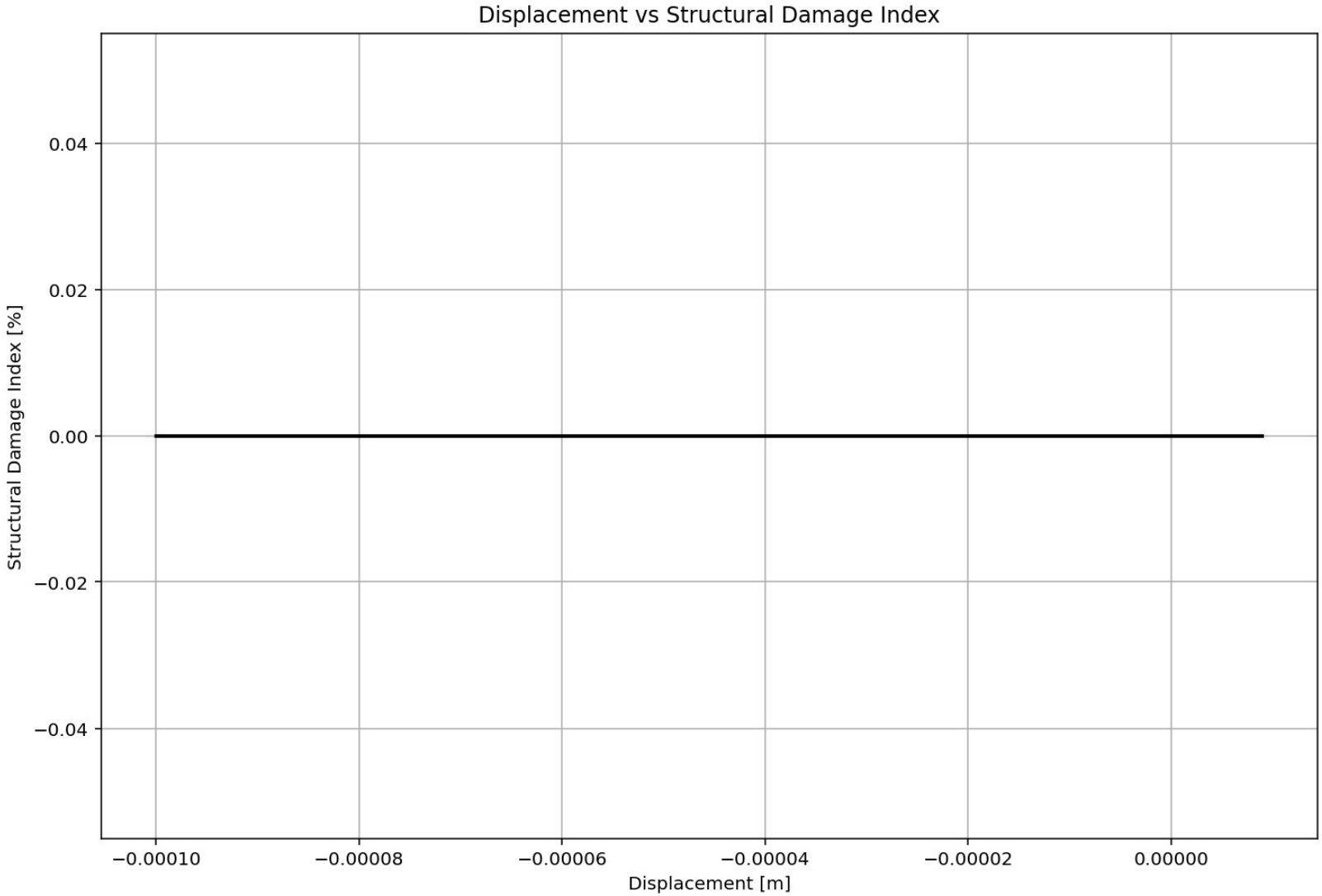

Base Reaction and Dispalcement of Structure During Free-vibration Analysis
Base Reaction [N]
60
40
20
0
−20
−40
−60
−80
−0.00010
−0.00008
−0.00006
−0.00004
−0.00002
0.00000
Displacement [m]

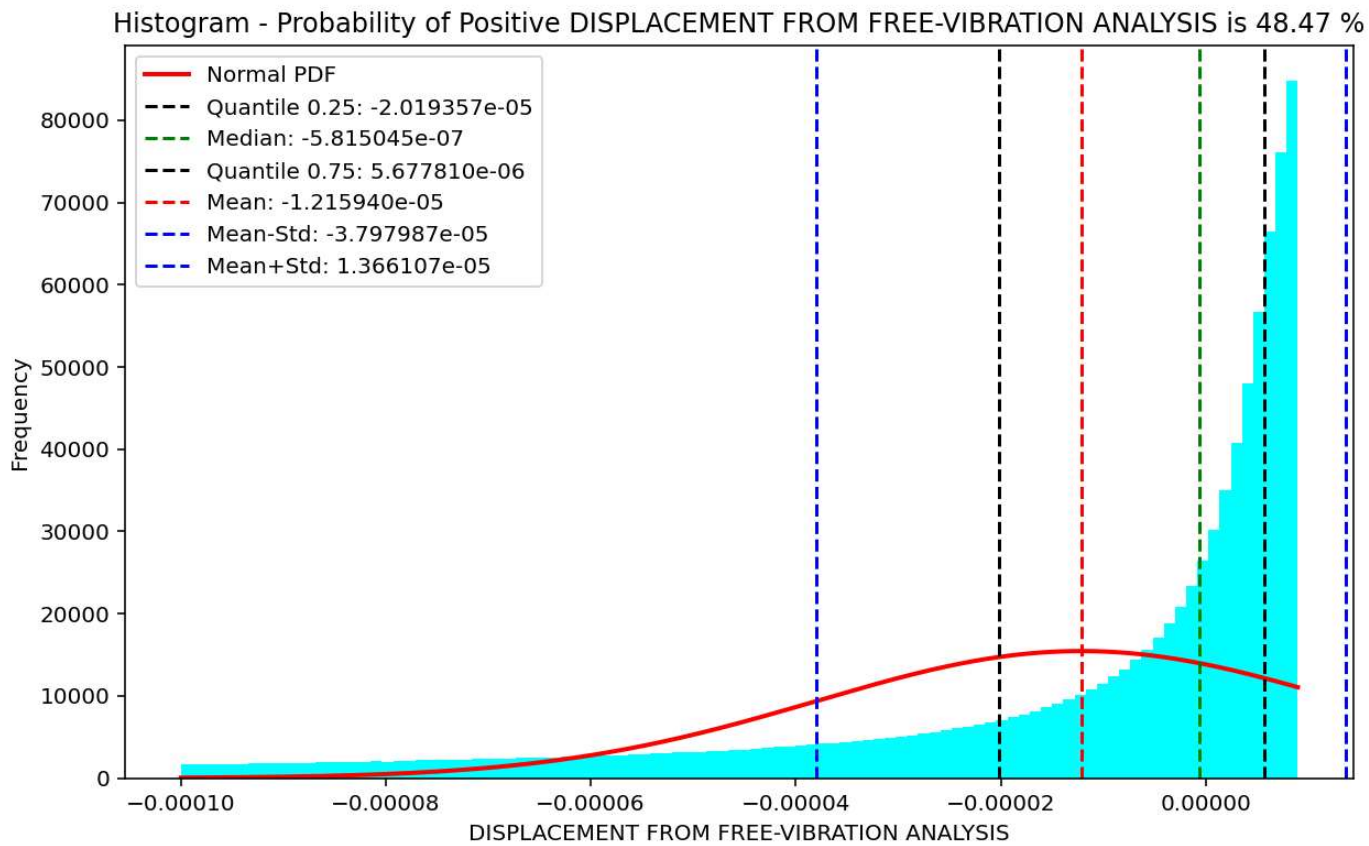

Reaction [N]
60
40
20
0
−20
−40
−60
−80
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (s)
Displacement [m]
0.00000
−0.00002
−0.00004
−0.00006
−0.00008
−0.00010
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (s)
Velocity [m/s]
1e−5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (s)
Acceleration [m/s²]
0.02
0.01
0.00
−0.01
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (s)

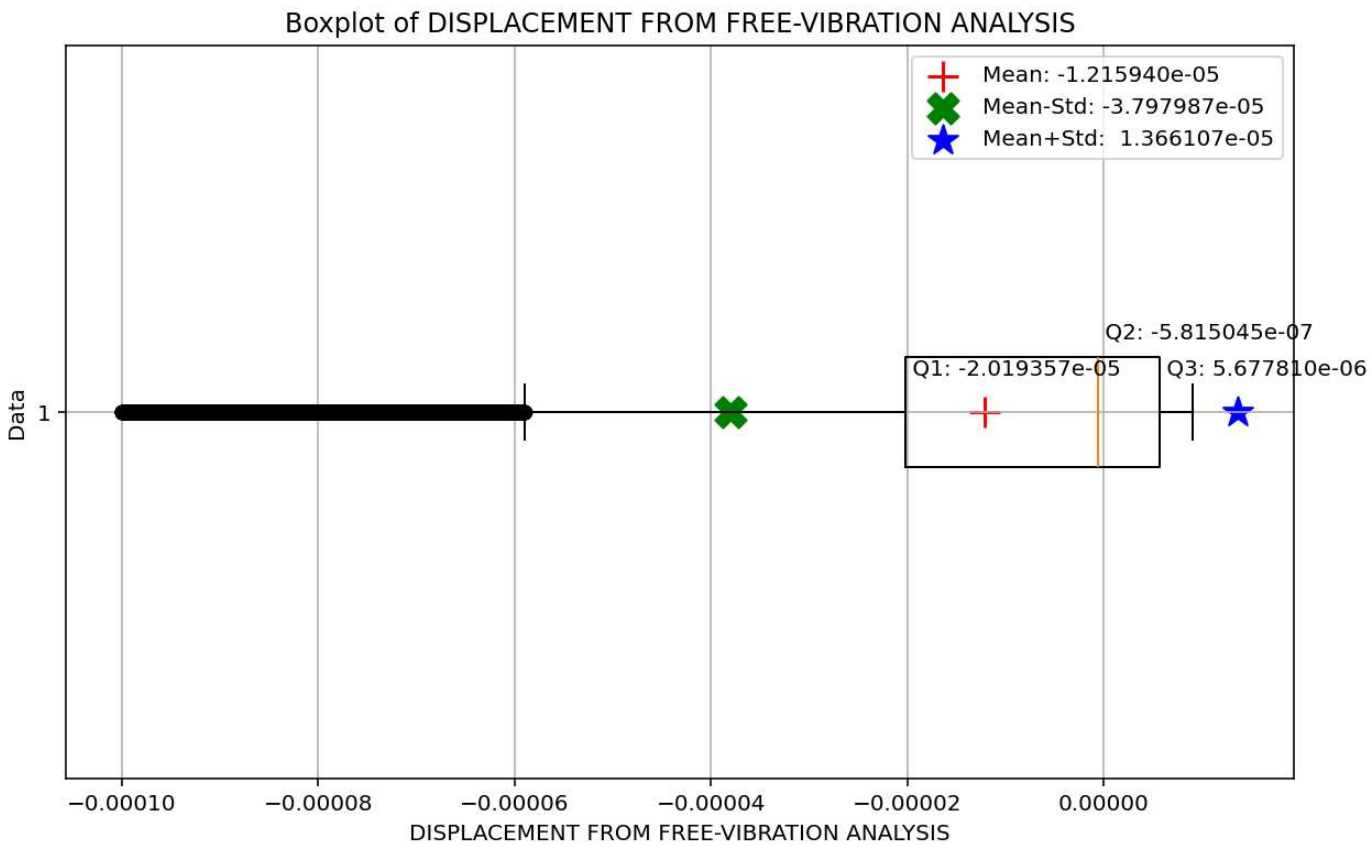

Period of Structure During Free-vibration Analysis
Structural Period [s]
0.050
0.045
0.040
0.035
0.030
0.025
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time [s]
PERIOD - MIN VALUES: Min: 0.023 (s) - Mean: 0.023 (s) - Max: 0.023 (s)
PERIOD - MAX VALUES: Min: 0.052 (s) - Mean: 0.052 (s) - Max: 0.052 (s)

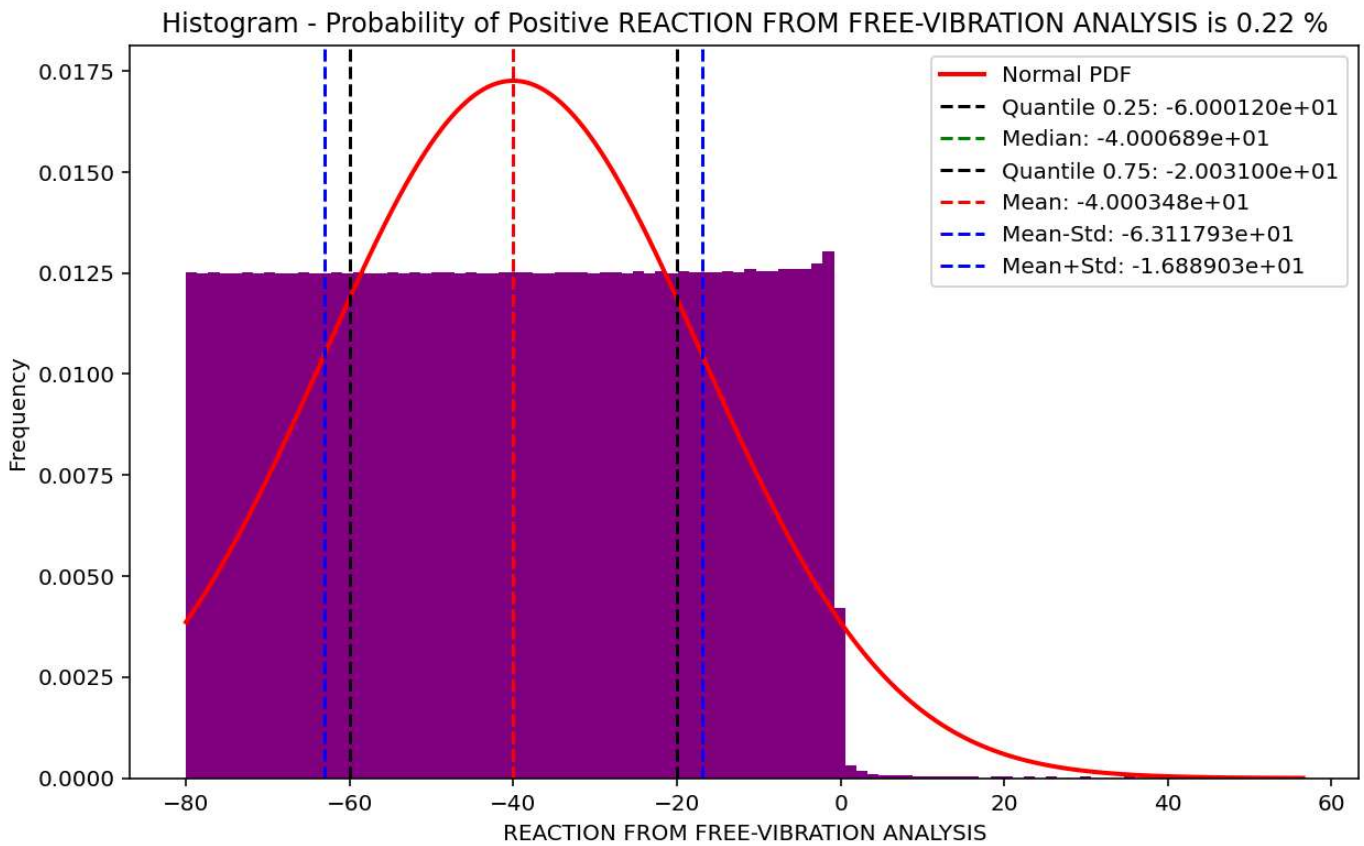

elements force vs Time During Free-vibration Analysis
Ele. 1 - MAX. ABS. : 8.0000e+01
Ele. 2 - MAX. ABS. : 6.0000e+01
Ele. 3 - MAX. ABS. : 4.0000e+01
Ele. 4 - MAX. ABS. : 2.0000e+01
Force [N]
60
40
20
0
−20
−40
−60
−80
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time [s]

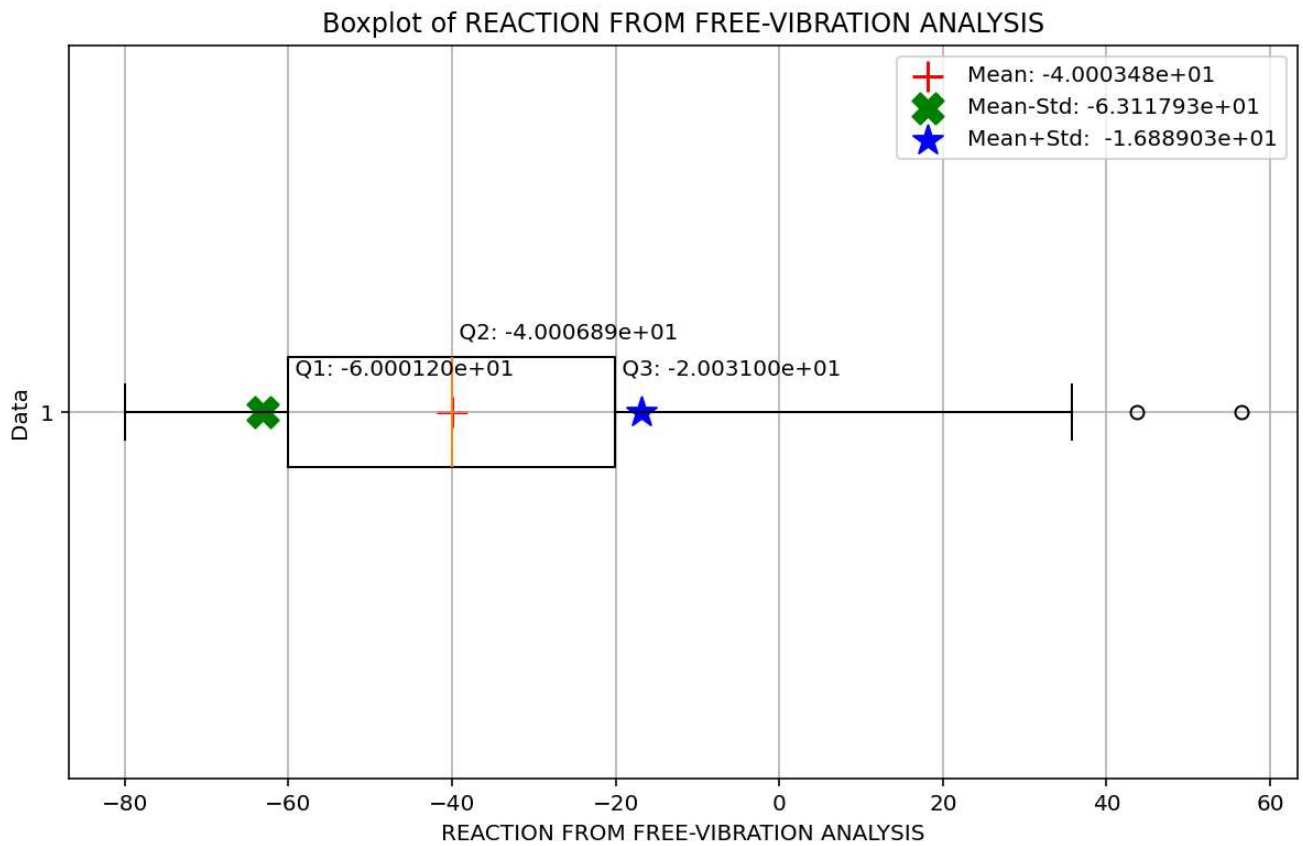

Node Displacements vs Time for During Free-vibration Analysis
Displacement [m]
Time [s]
0.00000
−0.00002
−0.00004
−0.00006
−0.00008
−0.00010
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Node 1 - MAX. ABS. : 0.0000e+00
Node 2 - MAX. ABS. : 9.9978e-05
Node 3 - MAX. ABS. : 9.9979e-05
Node 4 - MAX. ABS. : 9.9979e-05
Node 5 - MAX. ABS. : 9.9979e-05

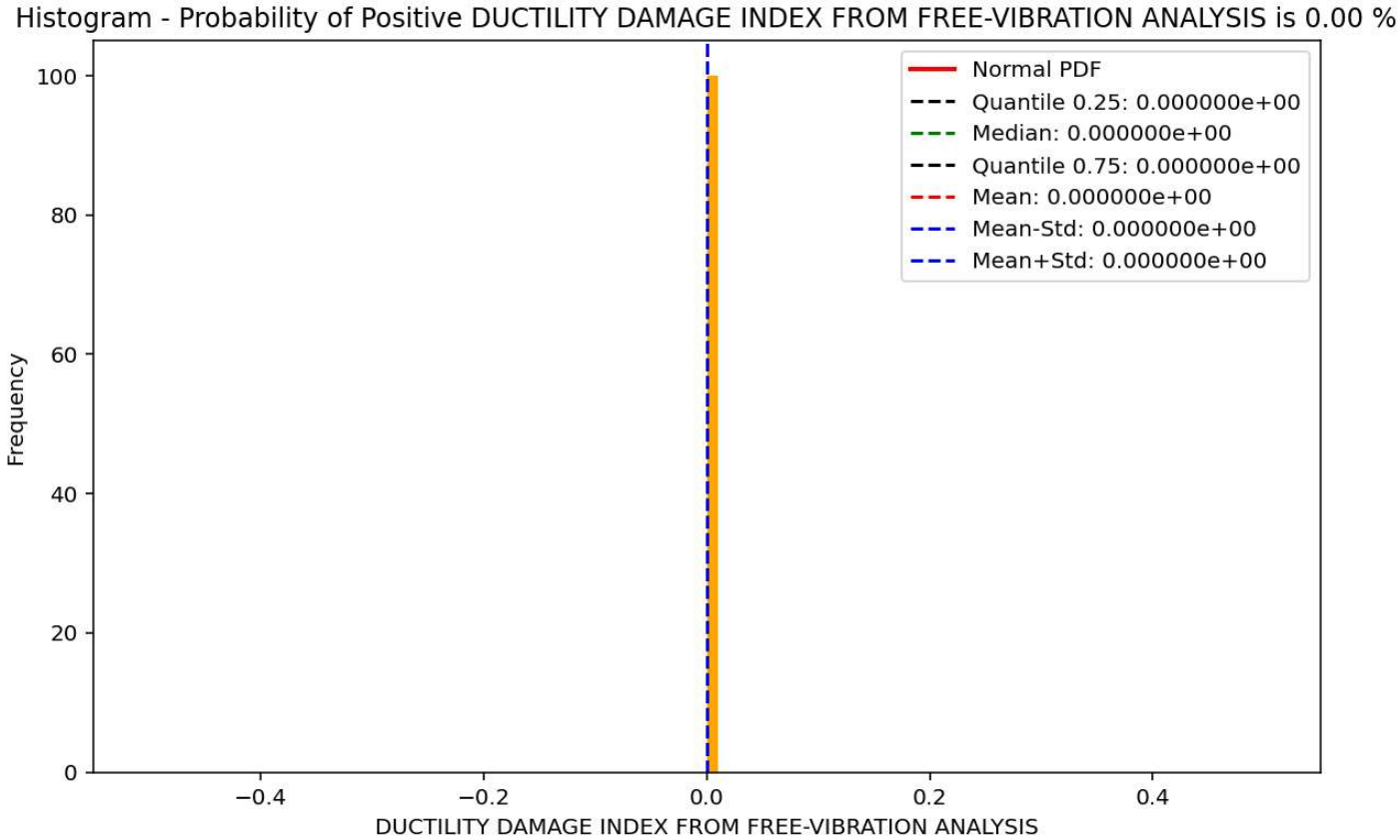

Displacement vs Structural Damage Index
Structural Damage Index [%]
0.04
0.02
0.00
−0.02
−0.04
−0.00010
−0.00008
−0.00006
−0.00004
−0.00002
0.00000
Displacement [m]

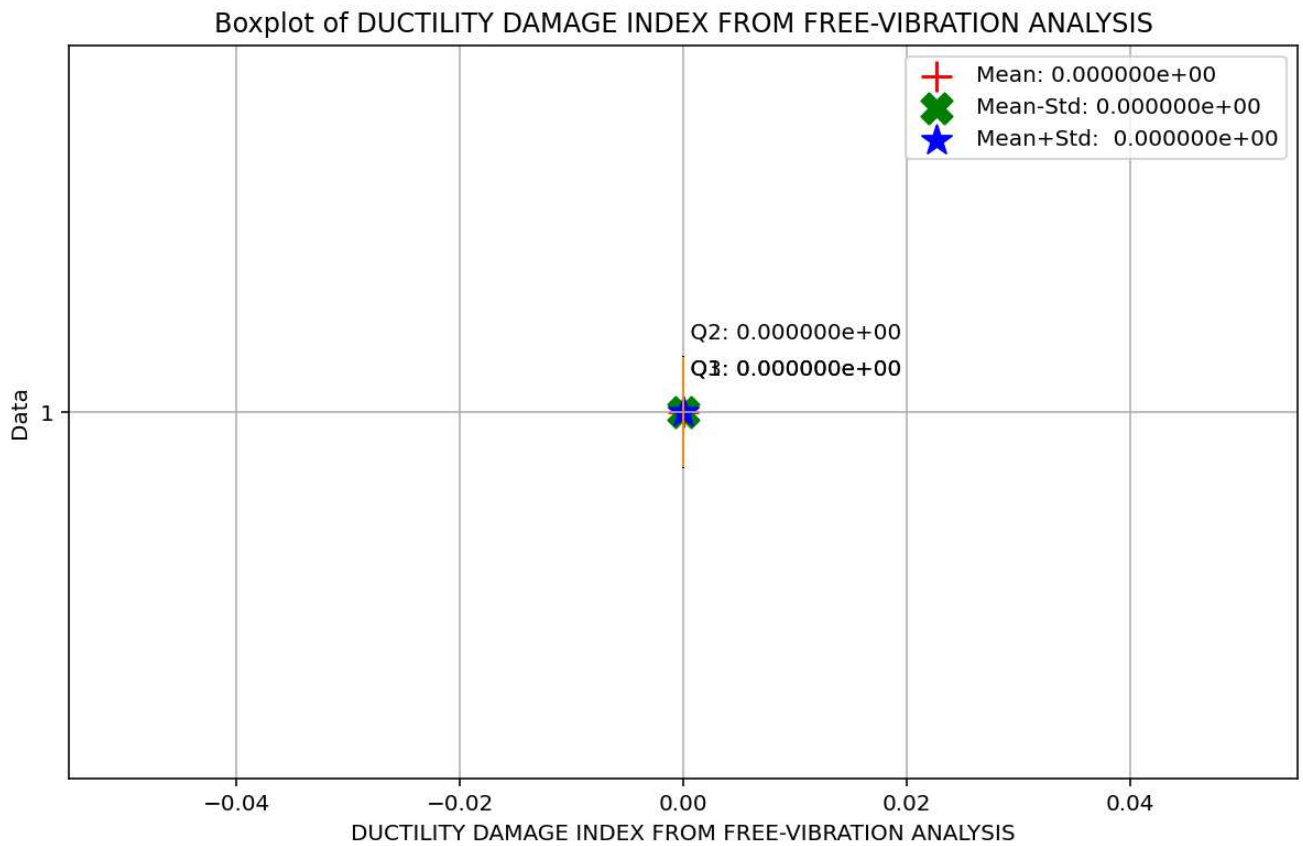

Histogram - Probability of Positive DISPLACEMENT FROM FREE-VIBRATION ANALYSIS is 48.47 %
Normal PDF
Quantile 0.25: -2.019357e-05
Median: -5.815045e-07
Quantile 0.75: 5.677810e-06
Mean: -1.215940e-05
Mean-Std: -3.797987e-05
Mean+Std: 1.366107e-05
Frequency
0
10000
20000
30000
40000
50000
60000
70000
80000
−0.00010
−0.00008
−0.00006
−0.00004
−0.00002
0.00000
DISPLACEMENT FROM FREE-VIBRATION ANALYSIS

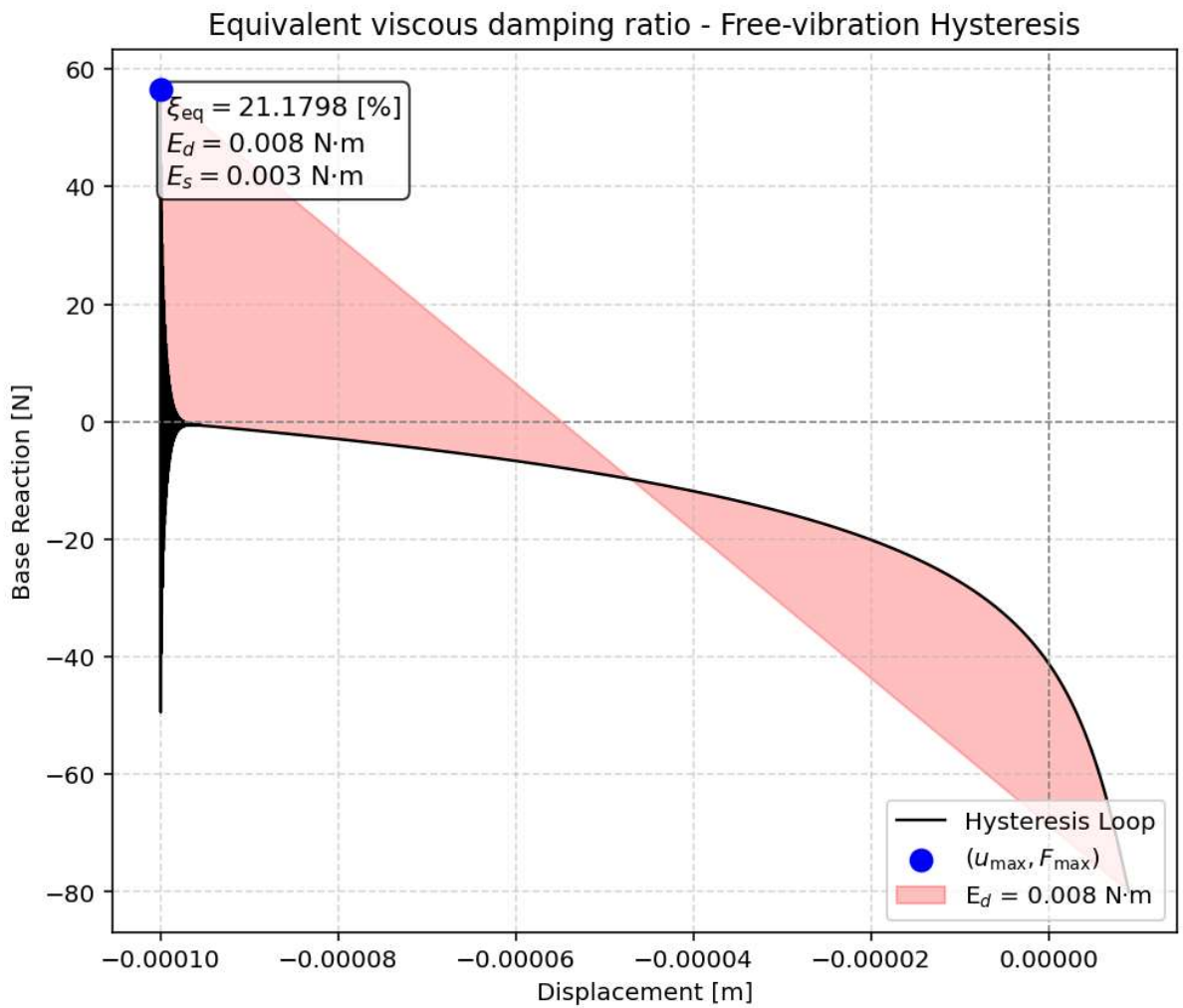

Boxplot of DISPLACEMENT FROM FREE-VIBRATION ANALYSIS
Mean: -1.215940e-05
Mean-Std: -3.797987e-05
Mean+Std: 1.366107e-05
Q2: -5.815045e-07
Q1: -2.019357e-05
Q3: 5.677810e-06
Data
1
−0.00010
−0.00008
−0.00006
−0.00004
−0.00002
0.00000
DISPLACEMENT FROM FREE-VIBRATION ANALYSIS


Histogram - Probability of Positive REACTION FROM FREE-VIBRATION ANALYSIS is 0.22 %
Normal PDF
Quantile 0.25: -6.000120e+01
Median: -4.000689e+01
Quantile 0.75: -2.003100e+01
Mean: -4.000348e+01
Mean-Std: -6.311793e+01
Mean+Std: -1.688903e+01
Frequency
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
−80
−60
−40
−20
0
20
40
60
REACTION FROM FREE-VIBRATION ANALYSIS


Boxplot of REACTION FROM FREE-VIBRATION ANALYSIS
Mean: -4.000348e+01
Mean-Std: -6.311793e+01
Mean+Std: -1.688903e+01
Q2: -4.000689e+01
Q1: -6.000120e+01
Q3: -2.003100e+01
Data
1
−80
−60
−40
−20
0
20
40
60
REACTION FROM FREE-VIBRATION ANALYSIS


Histogram - Probability of Positive DUCTILITY DAMAGE INDEX FROM FREE-VIBRATION ANALYSIS is 0.00 %
Normal PDF
Quantile 0.25: 0.000000e+00
Median: 0.000000e+00
Quantile 0.75: 0.000000e+00
Mean: 0.000000e+00
Mean-Std: 0.000000e+00
Mean+Std: 0.000000e+00
Frequency
0
20
40
60
80
100
−0.4
−0.2
0.0
0.2
0.4
DUCTILITY DAMAGE INDEX FROM FREE-VIBRATION ANALYSIS


Boxplot of DUCTILITY DAMAGE INDEX FROM FREE-VIBRATION ANALYSIS
Mean: 0.000000e+00
Mean-Std: 0.000000e+00
Mean+Std: 0.000000e+00
Q2: 0.000000e+00
Q3: 0.000000e+00
Data
1
−0.04
−0.02
0.00
0.02
0.04
DUCTILITY DAMAGE INDEX FROM FREE-VIBRATION ANALYSIS


Equivalent viscous damping ratio - Free-vibration Hysteresis
$\xi_{eq} = 21.1798$ [%]
$E_d = 0.008$ N·m
$E_s = 0.003$ N·m
Hysteresis Loop
(u_{max}, F_{max})
E_d = 0.008 N·m
Base Reaction [N]
Displacement [m]
60
40
20
0
−20
−40
−60
−80
−0.00010
−0.00008
−0.00006
−0.00004
−0.00002
0.00000

Spyder (Python 3.12)

File Edit Search Source Run Debug Consoles Projects Tools View Help

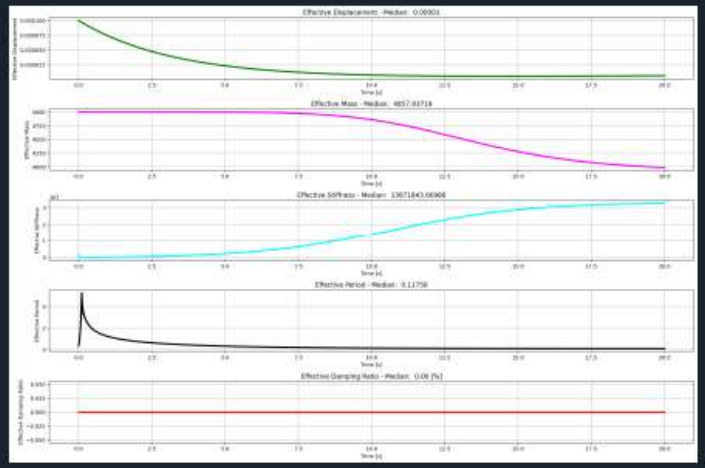
...DEX_&_ENERGY_DISSIPATION_CAPACITY_INDEX\33_MDOF_TWO_8_ANA

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...OMPUTE_EFFECTIVE_PROPERTIES_FUN_FREE_VIBRATION.py

P(t)_MDOF_TWO_8_ANA.py | COMPUTE_EFFECTIVE_..._FREE_VIBRATION.py

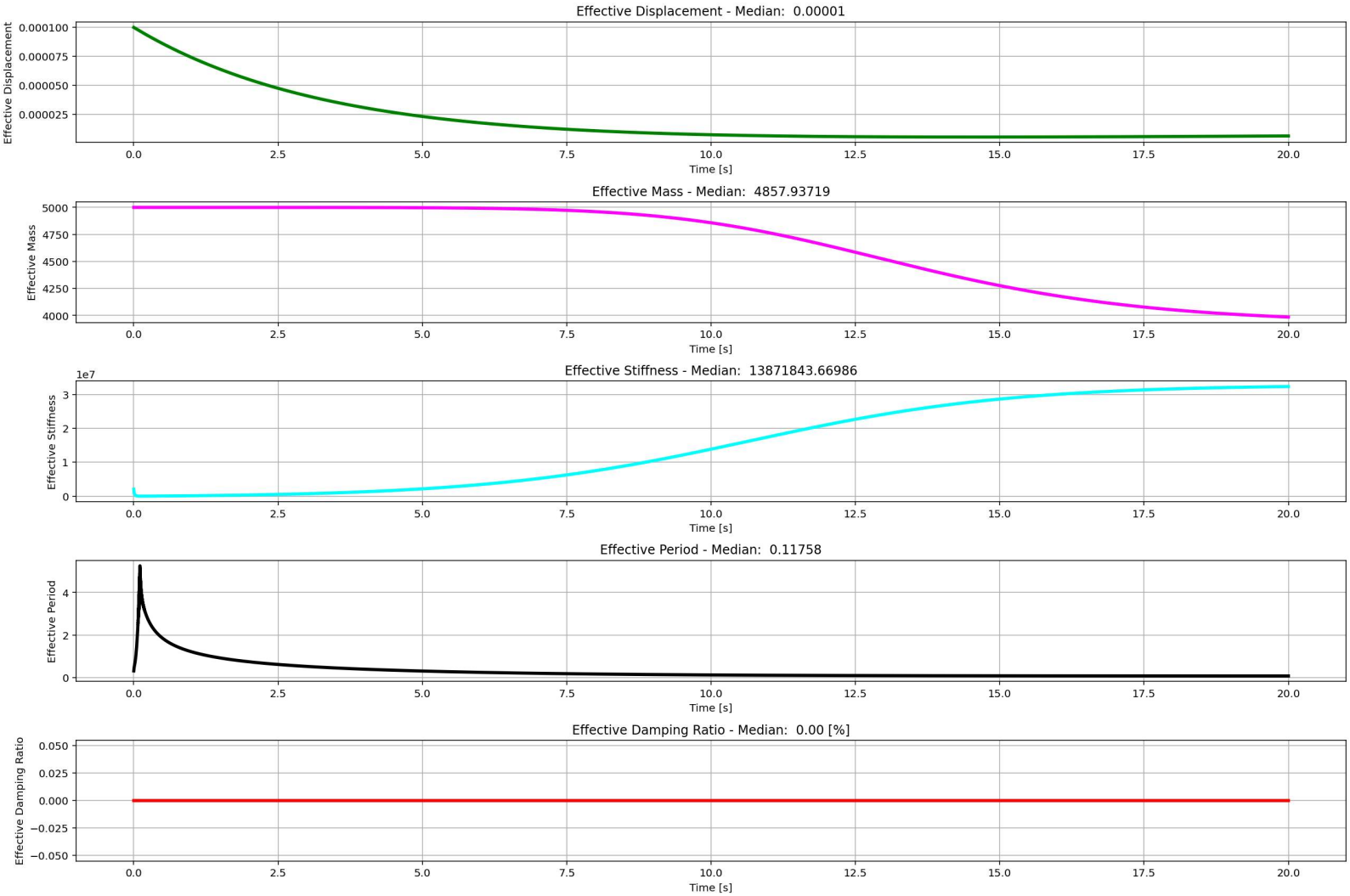
```python
#%% EQUIVALENT SDOF SYSTEM DERIVATION VIA DISPLACEMENT-BASED SEISMIC DESIGN PROCEDURE WITH
# Change MDOF to SDOF System with Displacement Based Design Concept
# THIS PROGRAM WRITTEN BY SALAR DELAVAR GHASHGHAEI (QASHQAI)
"""
This script implements a displacement-based pushover transformation,
 converting a multi-degree-of-freedom (MDOF) system into an equivalent
 single-degree-of-freedom (SDOF) system for seismic assessment.
 It calculates effective modal properties—displacement, mass, and
 stiffness—by weighting element forces and nodal displacements according
 to a presumed deformed shape.
 The derived effective period provides a simplified dynamic characteristic
 for performance-based engineering. The visualizations effectively track the
 evolution of these equivalent parameters throughout the nonlinear analysis steps.
"""
#-----------------------------------------------------------------------------
import numpy as np

displacement_X_1 = np.array(list(node_displacements.values())[1])   # DOF 02
displacement_X_2 = np.array(list(node_displacements.values())[2])   # DOF 03
displacement_X_3 = np.array(list(node_displacements.values())[3])   # DOF 04
displacement_X_4 = np.array(list(node_displacements.values())[4])   # DOF 05

ele_force_01 = np.array(list(ele_force.values())[0])   # ELEMENT 01
ele_force_02 = np.array(list(ele_force.values())[1])   # ELEMENT 02
ele_force_03 = np.array(list(ele_force.values())[2])   # ELEMENT 03
ele_force_04 = np.array(list(ele_force.values())[3])   # ELEMENT 04

STIFF_X_1 = np.abs(ele_force_01 / displacement_X_1)
STIFF_X_2 = np.abs(ele_force_02 / displacement_X_2)
STIFF_X_3 = np.abs(ele_force_03 / displacement_X_3)
STIFF_X_4 = np.abs(ele_force_04 / displacement_X_4)

MX2 = (MASS[0] * np.square(displacement_X_1) +
       MASS[1] * np.square(displacement_X_2) +
```

17 %



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

Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 1, Col 1 | UTF-8 | CRLF | RW | Mem 40%



$$\Delta_d = \frac{\sum_{i=1}^{n} m_i \Delta_i^2}{\sum_{i=1}^{n} m_i \Delta_i}$$

$$m_{eff} = \frac{\sum_{i=1}^{n} m_i \Delta_i}{\Delta_d}$$

$$k_{eff} = \frac{\sum_{i=1}^{n} k_i \Delta_i}{\Delta_d}$$

$$T_{eff} = \frac{2\pi}{\sqrt{\frac{k_{eff}}{m_{eff}}}}$$

$$\xi_{eff} = \frac{\sum_{i=1}^{n} \xi_i \Delta_i}{\Delta_d}$$

SEISMIC ANALYSIS

Spyder (Python 3.12)

File Edit Search Source Run Debug Consoles Projects Tools View Help

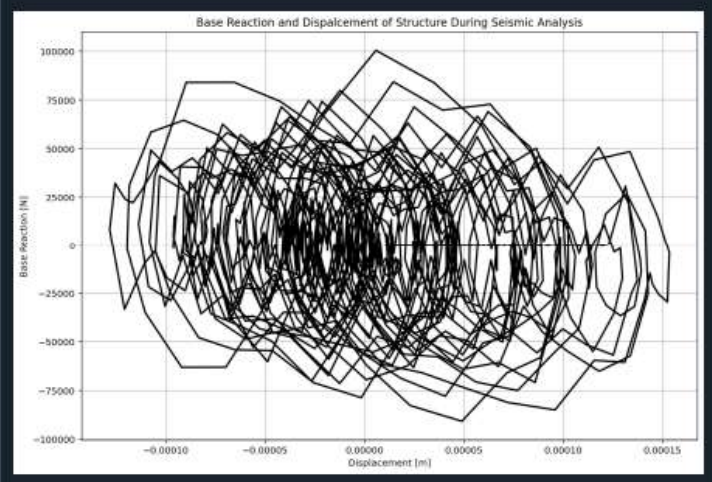
...DEX_&_ENERGY_DISSIPATION_CAPACITY_INDEX\33_MDOF_TWO_8_ANA

C:\Users\Dell\Desktop\OPENSEES_FILES\+EIGHT_ANALY...TY_INDEX\33_MDOF_TWO_8_ANA\P(t)_MDOF_TWO_8_ANA.py

P(t)_MDOF_8_ANA.py | P(t)_MDOF_TWO_8_ANA.py

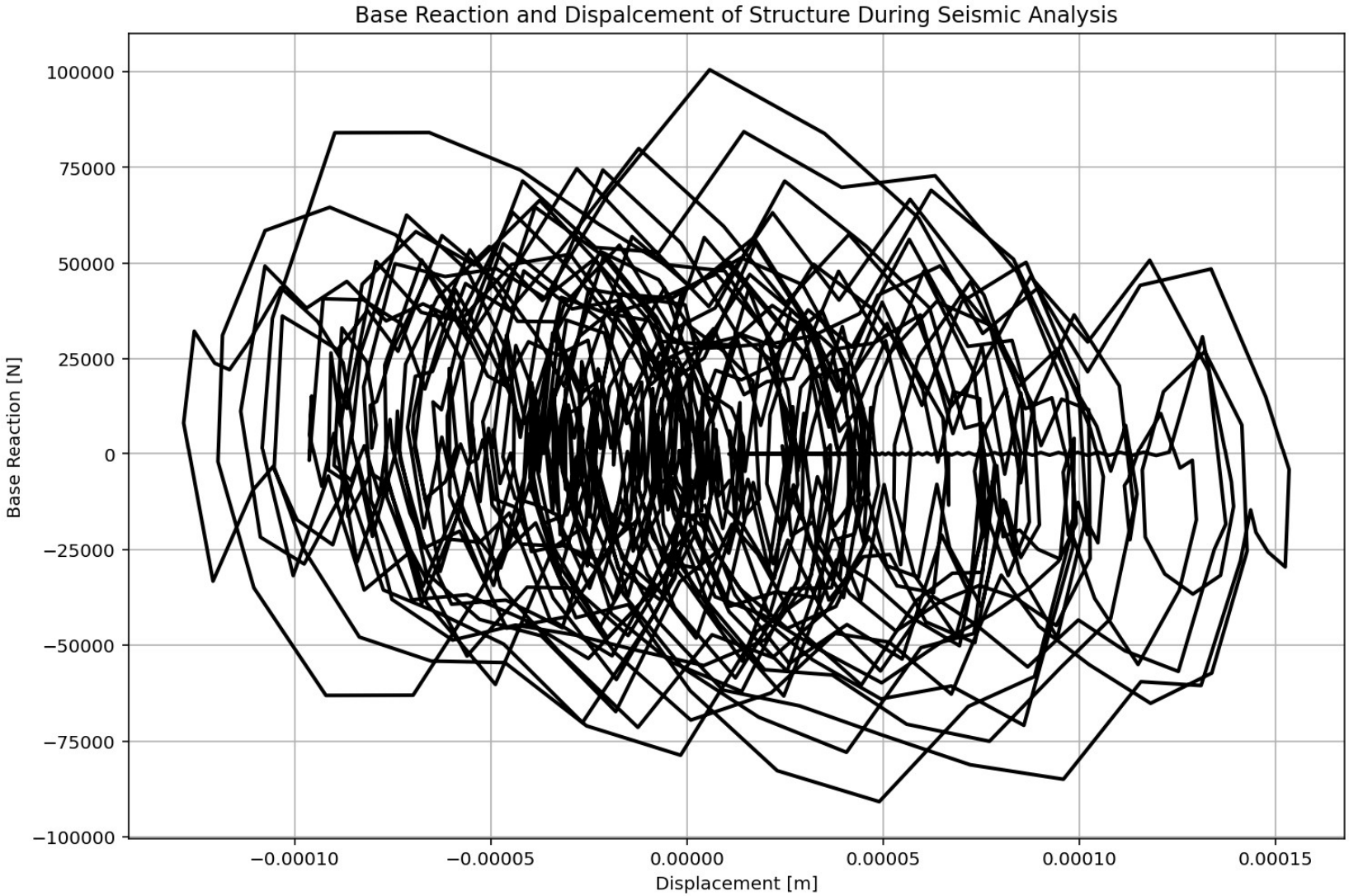
```
702
703     if ANAL_TYPE == 'SEISMIC': # DYNAMIC TIME-HISTORY ANALYSIS
704         #%% DEFINE PARAMETERS FOR SEISMIC ANALYSIS
705         duration = 20.0                 # [s] Analysis duration
706         dt = 0.01                       # [s] Time step
707         GMfact = 9810                   # [mm/s²]standard acceleration of gravity or st
708         SSF_X = 0.10                    # Seismic Acceleration Scale Factor in X Direct
709         SSF_Y = 0.10                    # Seismic Acceleration Scale Factor in Y Direct
710         iv0_X = 0.0005                  # [mm/s] Initial velocity applied to the node
711         iv0_Y = 0.0005                  # [mm/s] Initial velocity applied to the node
712         st_iv0 = 0.0                    # [s] Initial velocity applied starting time
713         SEI = 'X'                       # Seismic Direction
714         DR = 0.03                       # Damping ratio
715
716         # Define time series for input motion (Acceleration time history)
717         if SEI == 'X':
718             SEISMIC_TAG_01 = 100
719             ops.timeSeries('Path', SEISMIC_TAG_01, '-dt', dt, '-filePath', 'Ground_Acceler
720             # Define load patterns
721             # pattern UniformExcitation $patternTag $dof -accel $tsTag <-vel0 $vel0> <-fac
722             ops.pattern('UniformExcitation', SEISMIC_TAG_01, 1, '-accel', SEISMIC_TAG_01,
723         if SEI == 'Y':
724             SEISMIC_TAG_02 = 200
725             ops.timeSeries('Path', SEISMIC_TAG_02, '-dt', dt, '-filePath', 'Ground_Acceler
726             ops.pattern('UniformExcitation', SEISMIC_TAG_02, 2, '-accel', SEISMIC_TAG_02,
727         if SEI == 'XY':
728             SEISMIC_TAG_01 = 100
729             ops.timeSeries('Path', SEISMIC_TAG_01, '-dt', dt, '-filePath', 'Ground_Acceler
730             # Define load patterns
731             # pattern UniformExcitation $patternTag $dof -accel $tsTag <-vel0 $vel0> <-fac
732             ops.pattern('UniformExcitation', SEISMIC_TAG_01, 1, '-accel', SEISMIC_TAG_01,
733             SEISMIC_TAG_02 = 200
734             ops.timeSeries('Path', SEISMIC_TAG_02, '-dt', dt, '-filePath', 'Ground_Acceler
735             ops.pattern('UniformExcitation', SEISMIC_TAG_02, 2, '-accel', SEISMIC_TAG_02,
```

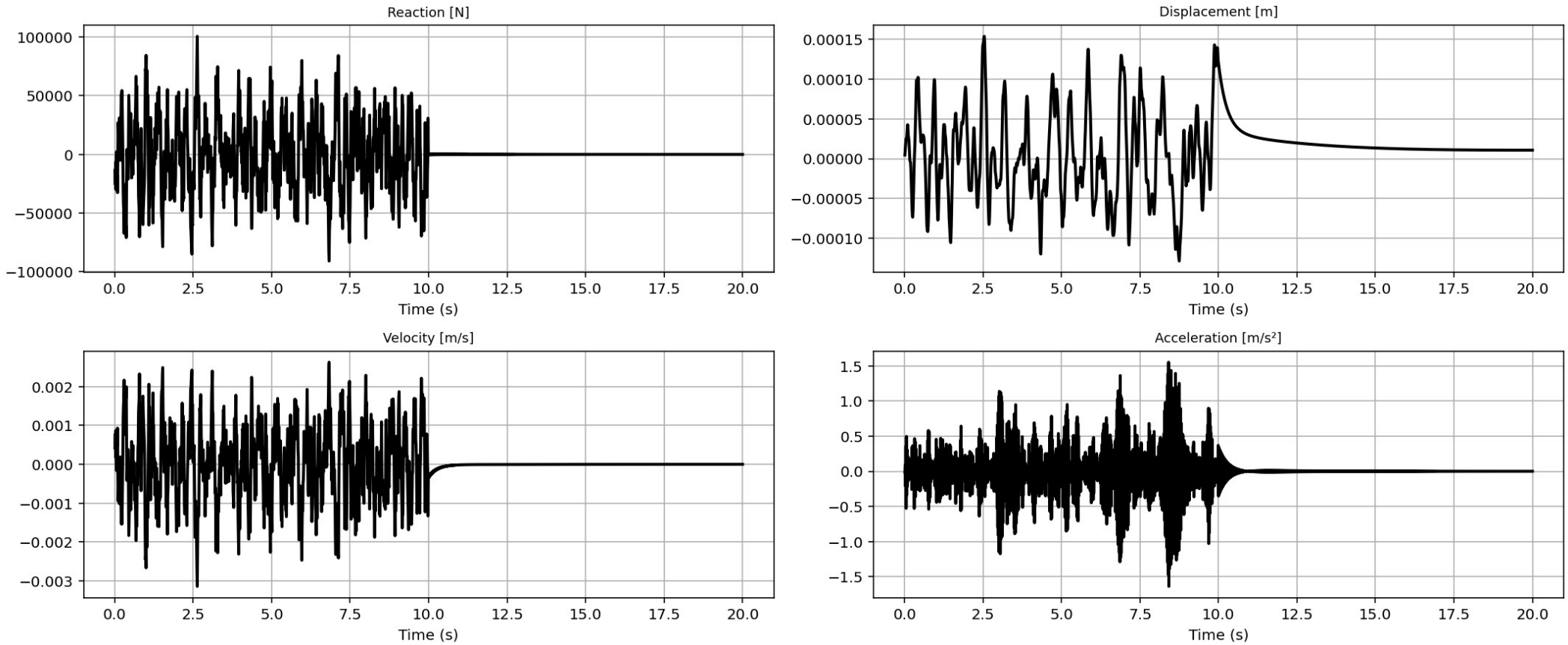
30 %

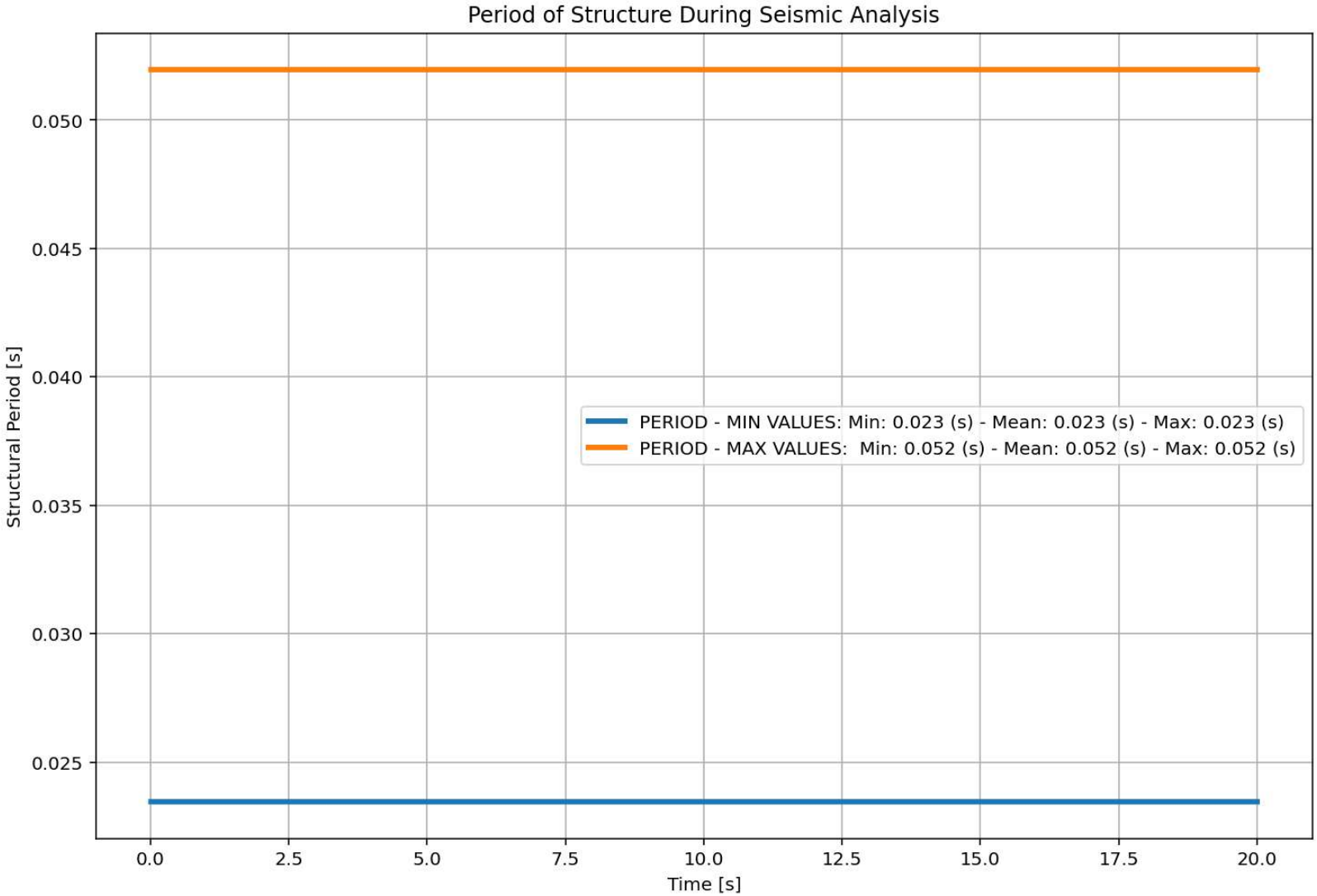


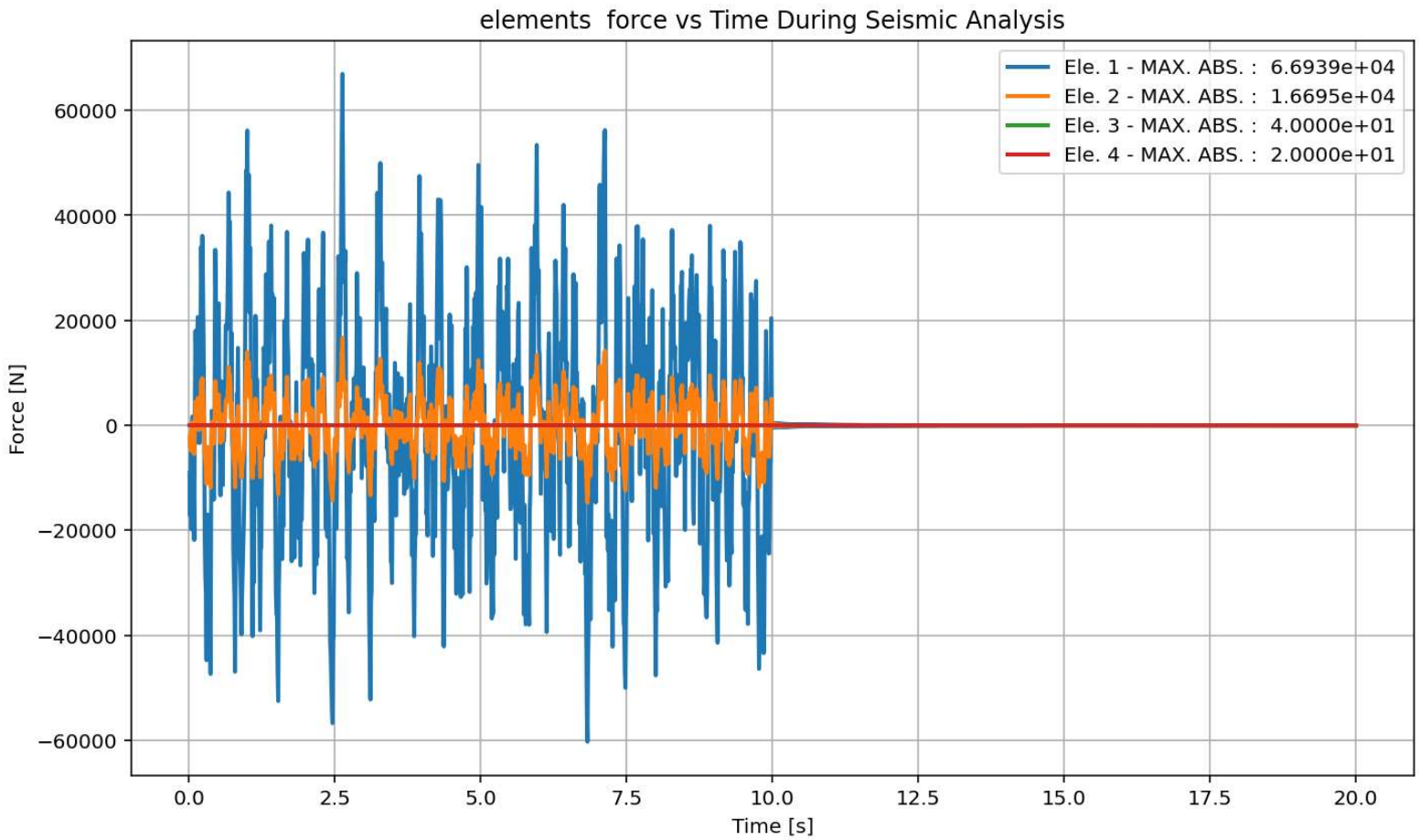
IPython Console | Files | Help | Variable Explorer | Debugger | Plots | History

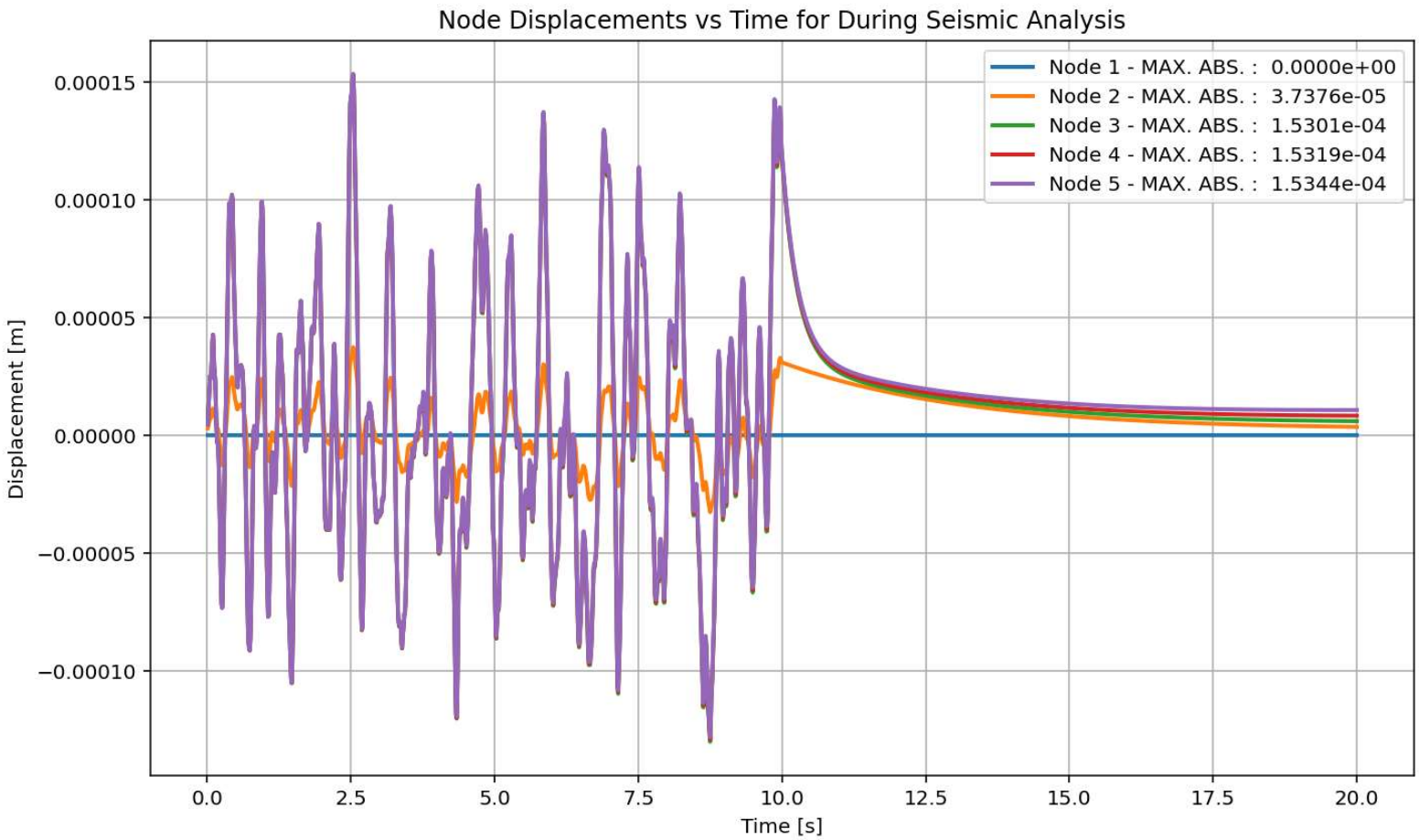
Inline | Conda: anaconda3 (Python 3.12.7) | LSP: Python | Line 702, Col 5 | UTF-8 | CRLF | RW | Mem 46%

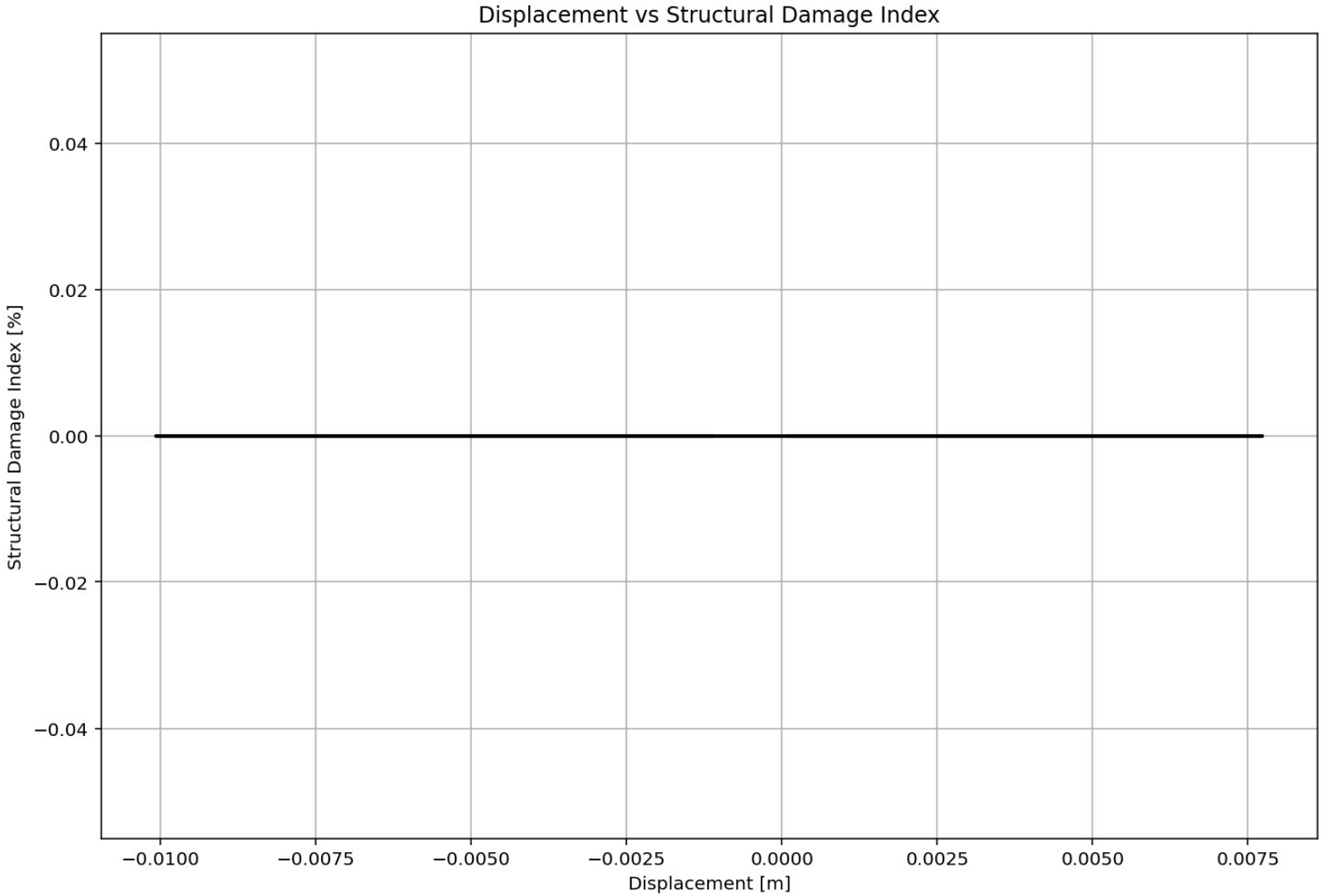

Base Reaction and Dispalcement of Structure During Seismic Analysis
100000
75000
50000
25000
0
−25000
−50000
−75000
−100000
Base Reaction [N]
−0.00010
−0.00005
0.00000
0.00005
0.00010
0.00015
Displacement [m]

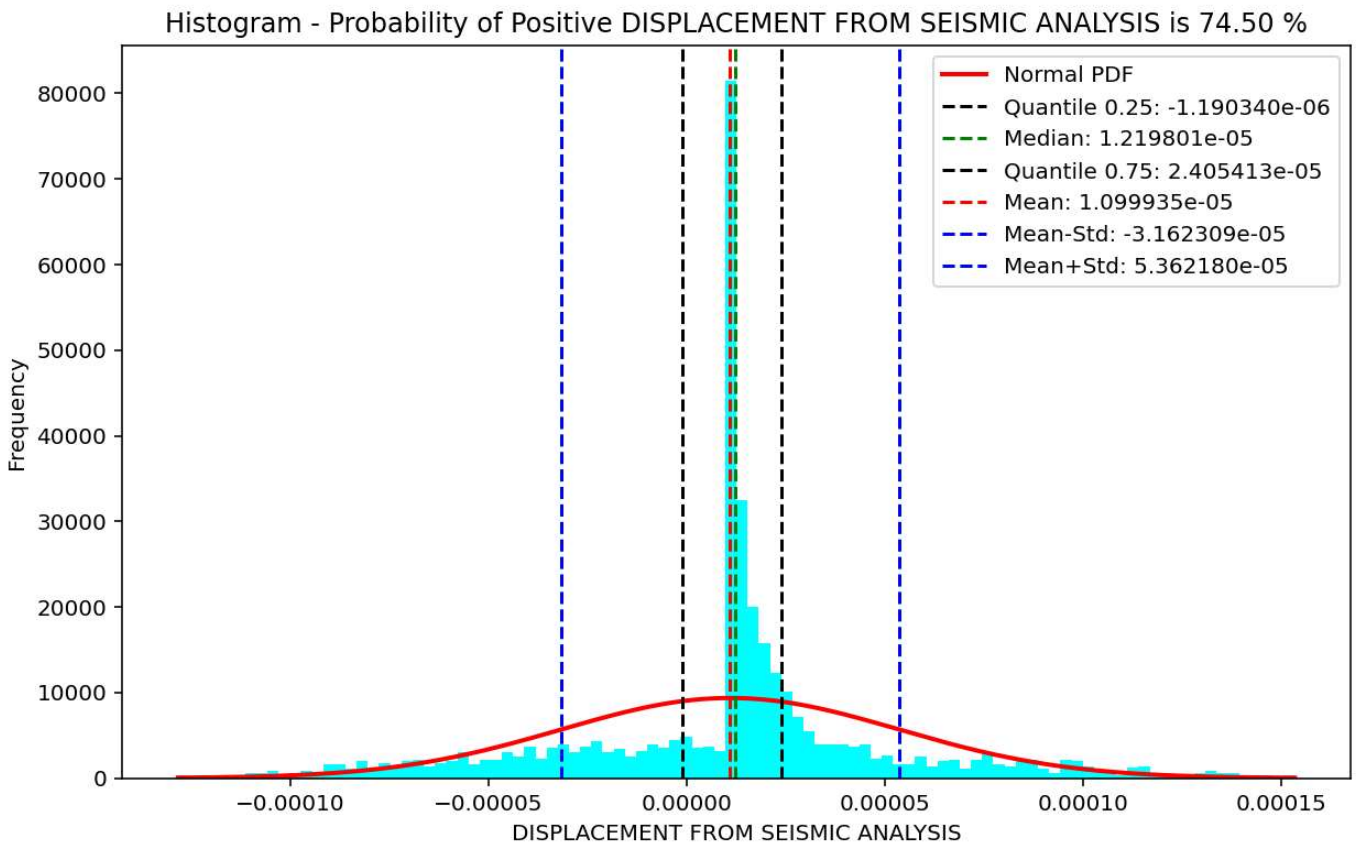

Reaction [N]
100000
50000
0
−50000
−100000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time (s)
Displacement [m]
0.00015
0.00010
0.00005
0.00000
−0.00005
−0.00010
Time (s)
Velocity [m/s]
0.002
0.001
0.000
−0.001
−0.002
−0.003
Time (s)
Acceleration [m/s²]
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Time (s)

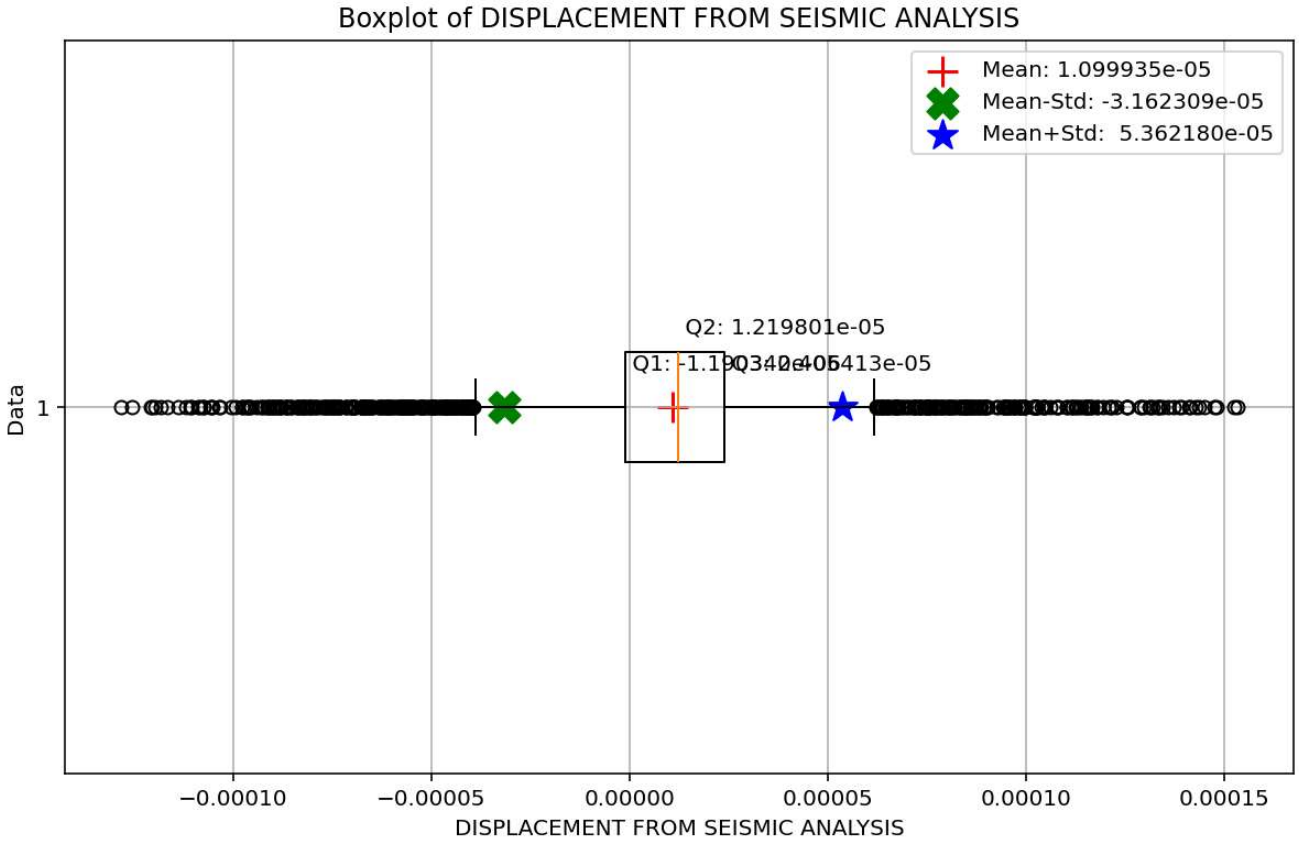

Period of Structure During Seismic Analysis
Structural Period [s]
0.050
0.045
0.040
0.035
0.030
0.025
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time [s]
PERIOD - MIN VALUES: Min: 0.023 (s) - Mean: 0.023 (s) - Max: 0.023 (s)
PERIOD - MAX VALUES: Min: 0.052 (s) - Mean: 0.052 (s) - Max: 0.052 (s)

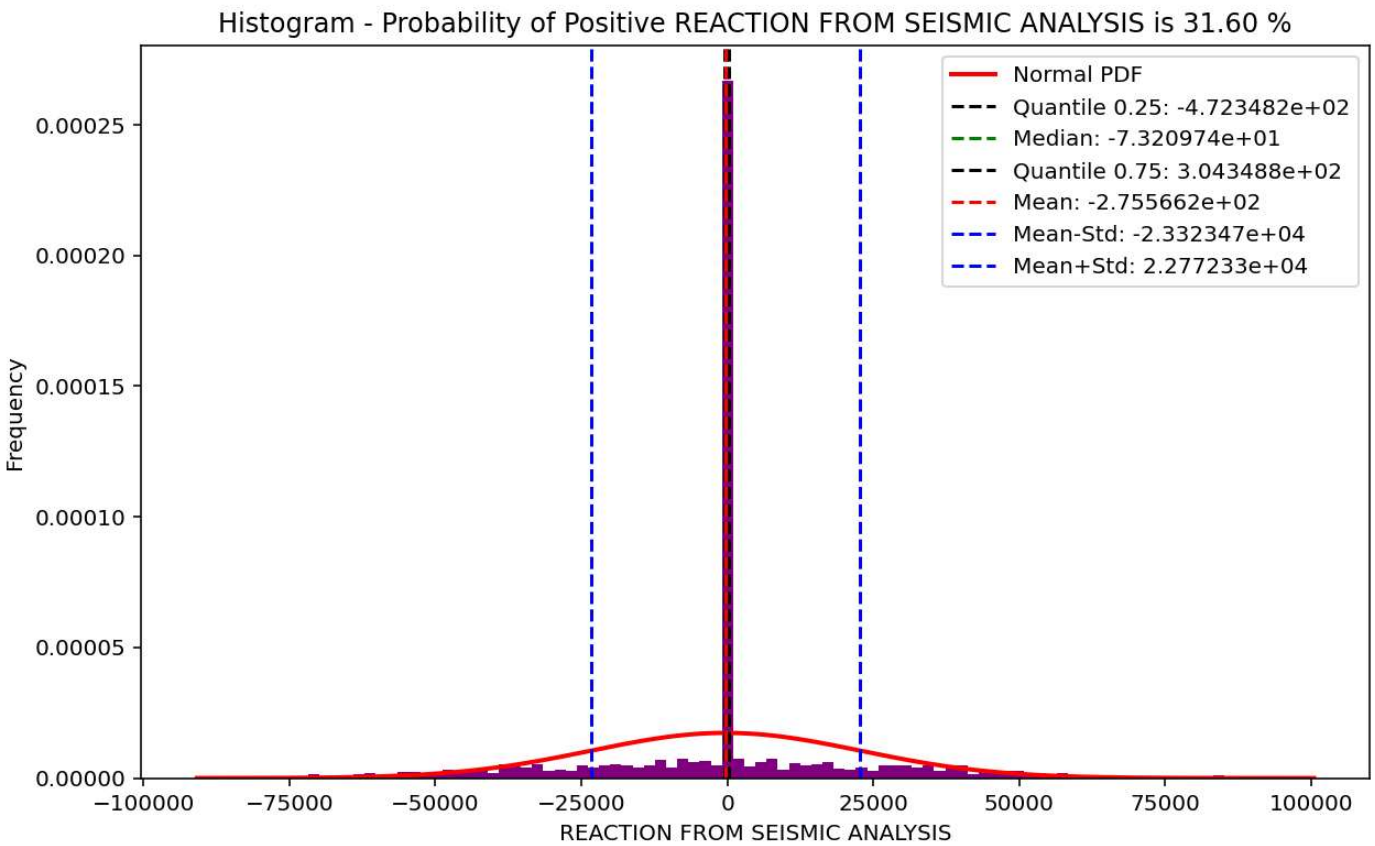

elements force vs Time During Seismic Analysis
Ele. 1 - MAX. ABS. : 6.6939e+04
Ele. 2 - MAX. ABS. : 1.6695e+04
Ele. 3 - MAX. ABS. : 4.0000e+01
Ele. 4 - MAX. ABS. : 2.0000e+01
Force [N]
60000
40000
20000
0
−20000
−40000
−60000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time [s]

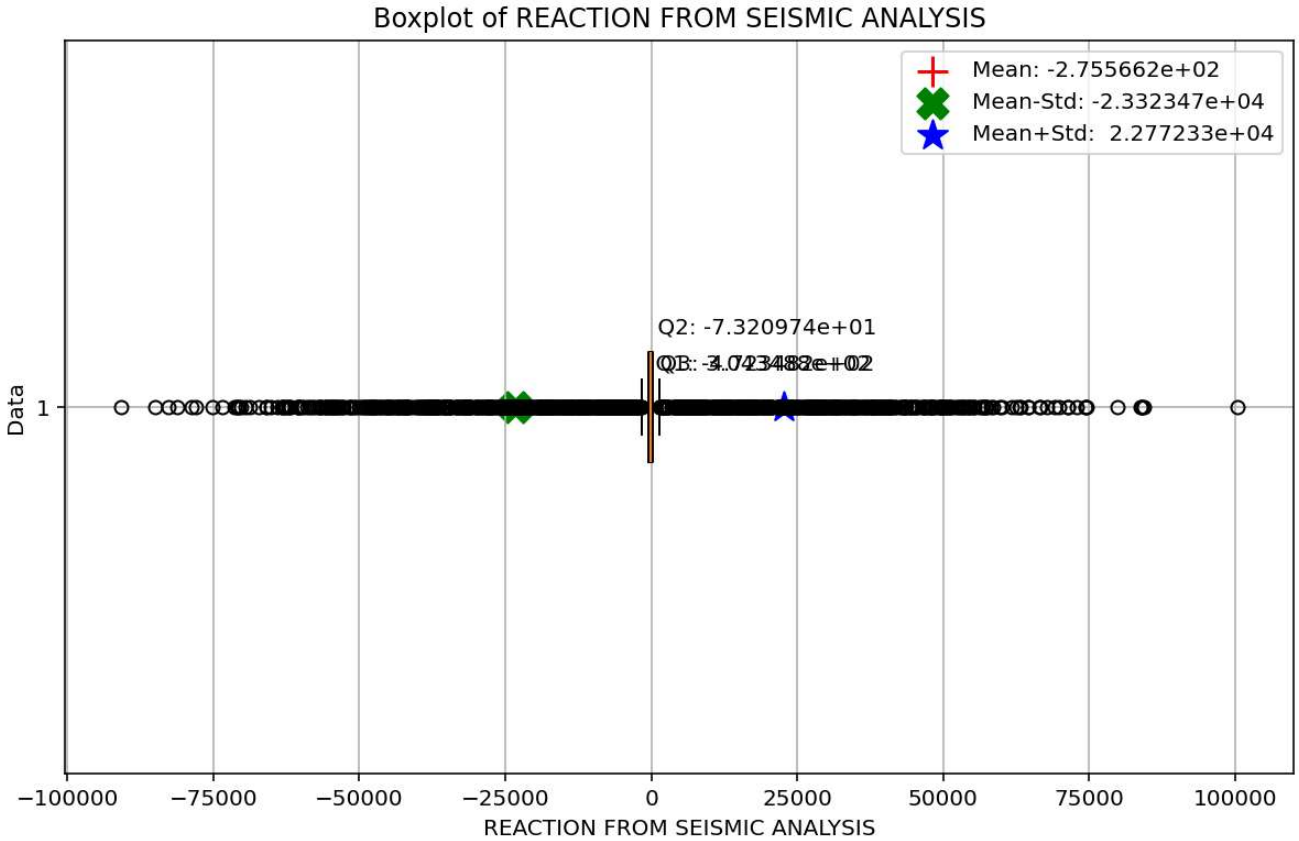

Node Displacements vs Time for During Seismic Analysis
Node 1 - MAX. ABS. : 0.0000e+00
Node 2 - MAX. ABS. : 3.7376e-05
Node 3 - MAX. ABS. : 1.5301e-04
Node 4 - MAX. ABS. : 1.5319e-04
Node 5 - MAX. ABS. : 1.5344e-04
Displacement [m]
0.00015
0.00010
0.00005
0.00000
−0.00005
−0.00010
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Time [s]

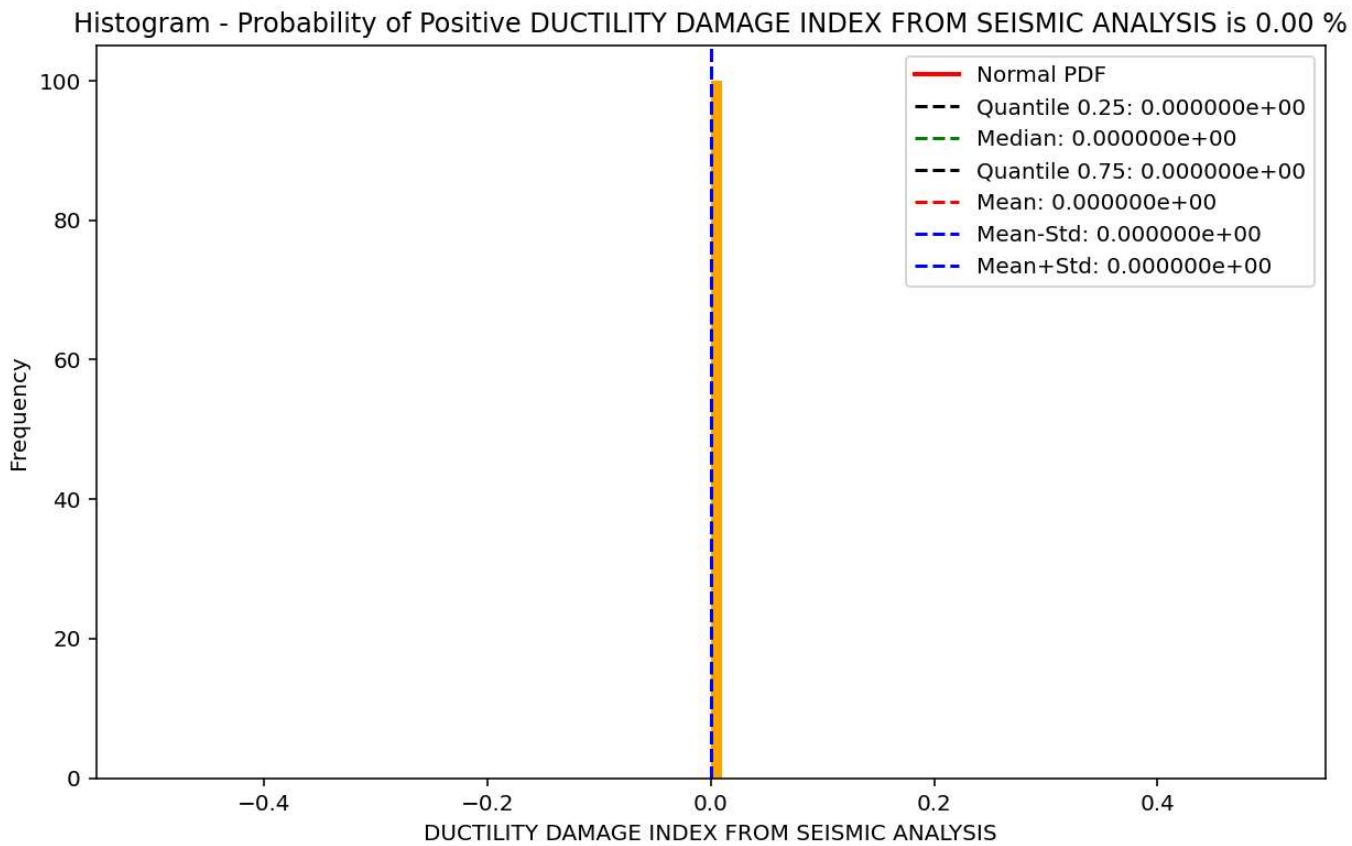

Displacement vs Structural Damage Index
Structural Damage Index [%]
0.04
0.02
0.00
−0.02
−0.04
−0.0100
−0.0075
−0.0050
−0.0025
0.0000
0.0025
0.0050
0.0075
Displacement [m]

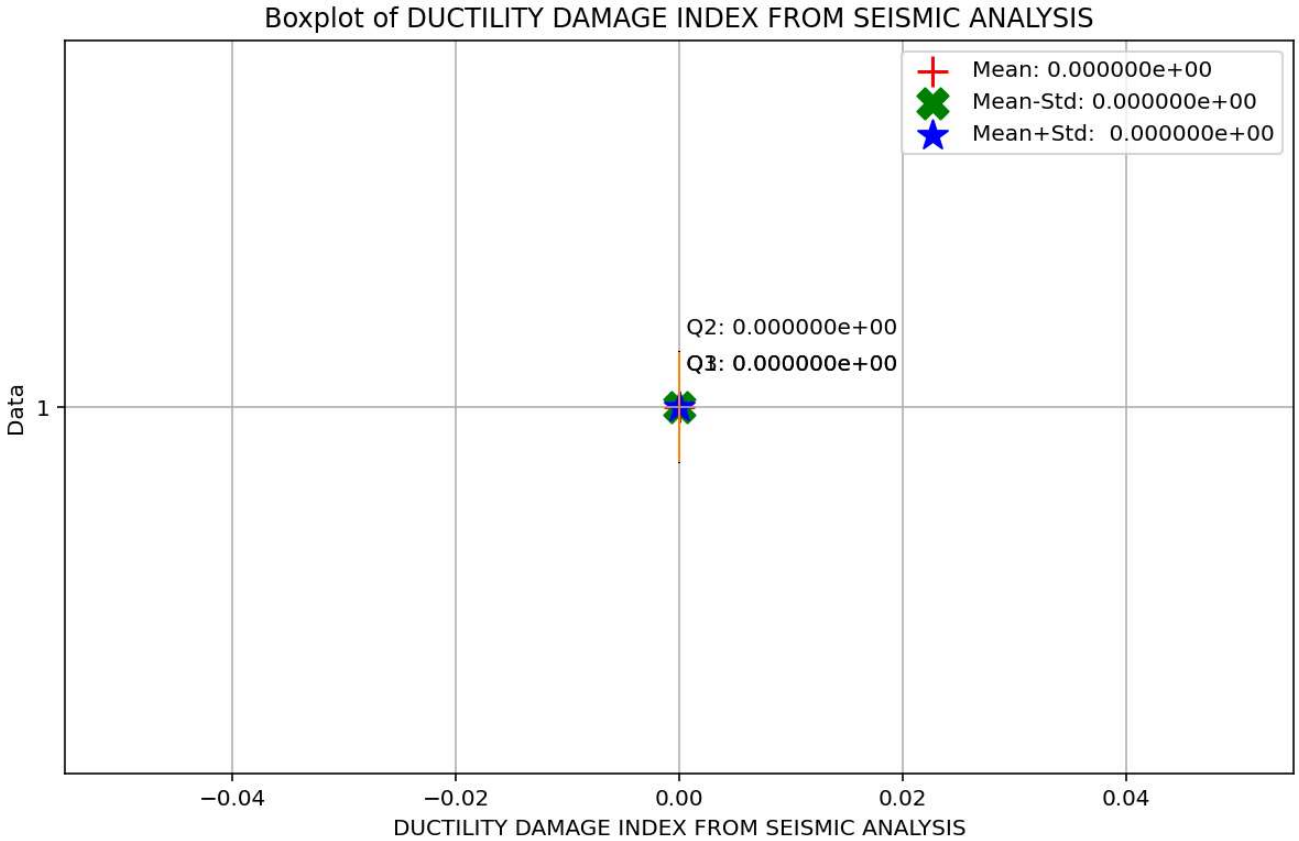

Histogram - Probability of Positive DISPLACEMENT FROM SEISMIC ANALYSIS is 74.50 %
Normal PDF
Quantile 0.25: -1.190340e-06
Median: 1.219801e-05
Quantile 0.75: 2.405413e-05
Mean: 1.099935e-05
Mean-Std: -3.162309e-05
Mean+Std: 5.362180e-05
Frequency
80000
70000
60000
50000
40000
30000
20000
10000
0
−0.00010
−0.00005
0.00000
0.00005
0.00010
0.00015
DISPLACEMENT FROM SEISMIC ANALYSIS

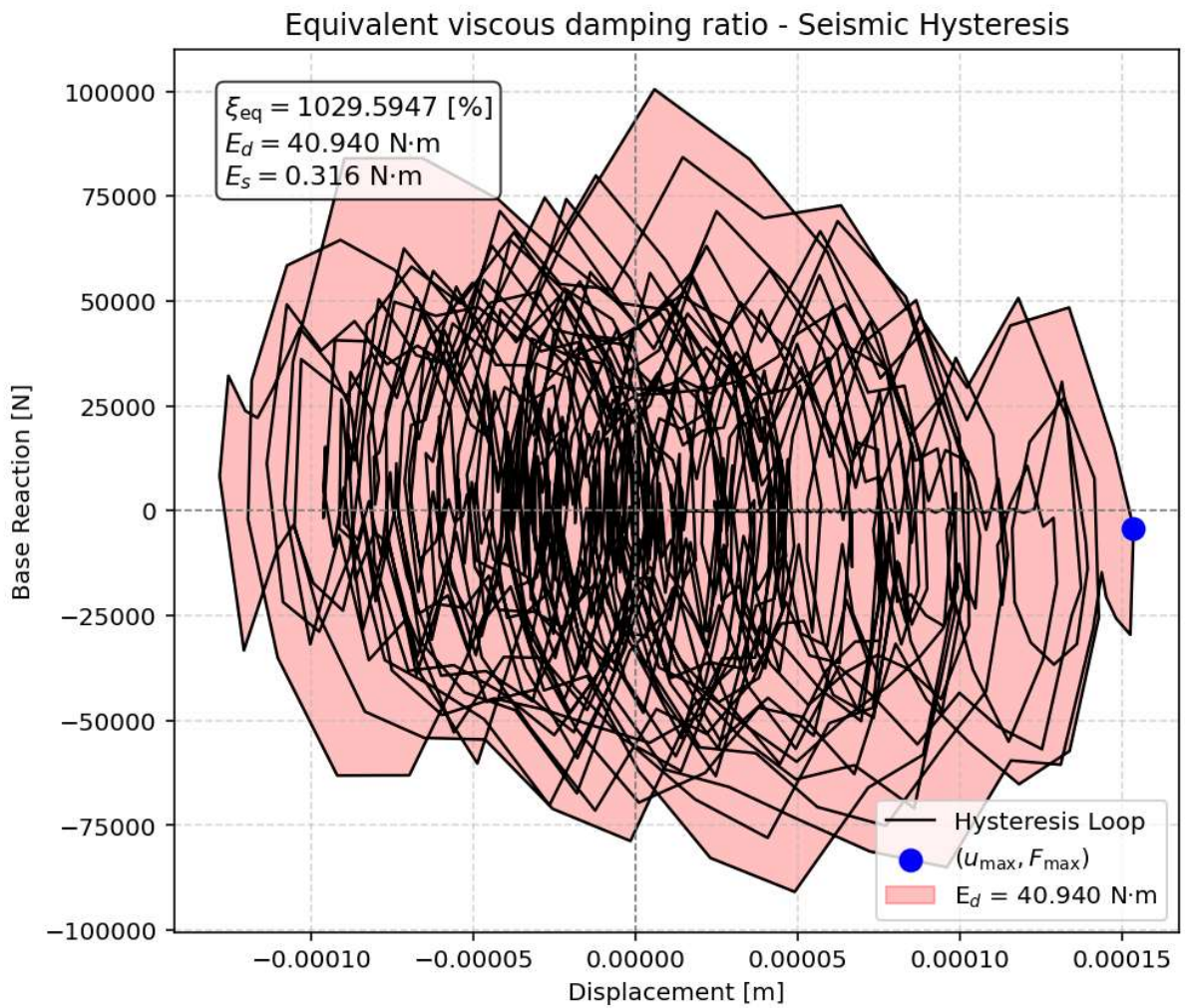

Boxplot of DISPLACEMENT FROM SEISMIC ANALYSIS
Mean: 1.099935e-05
Mean-Std: -3.162309e-05
Mean+Std: 5.362180e-05
Q2: 1.219801e-05
Q1: -1.190342e-05
Q3: 4.406413e-05
Data
1
−0.00010
−0.00005
0.00000
0.00005
0.00010
0.00015
DISPLACEMENT FROM SEISMIC ANALYSIS

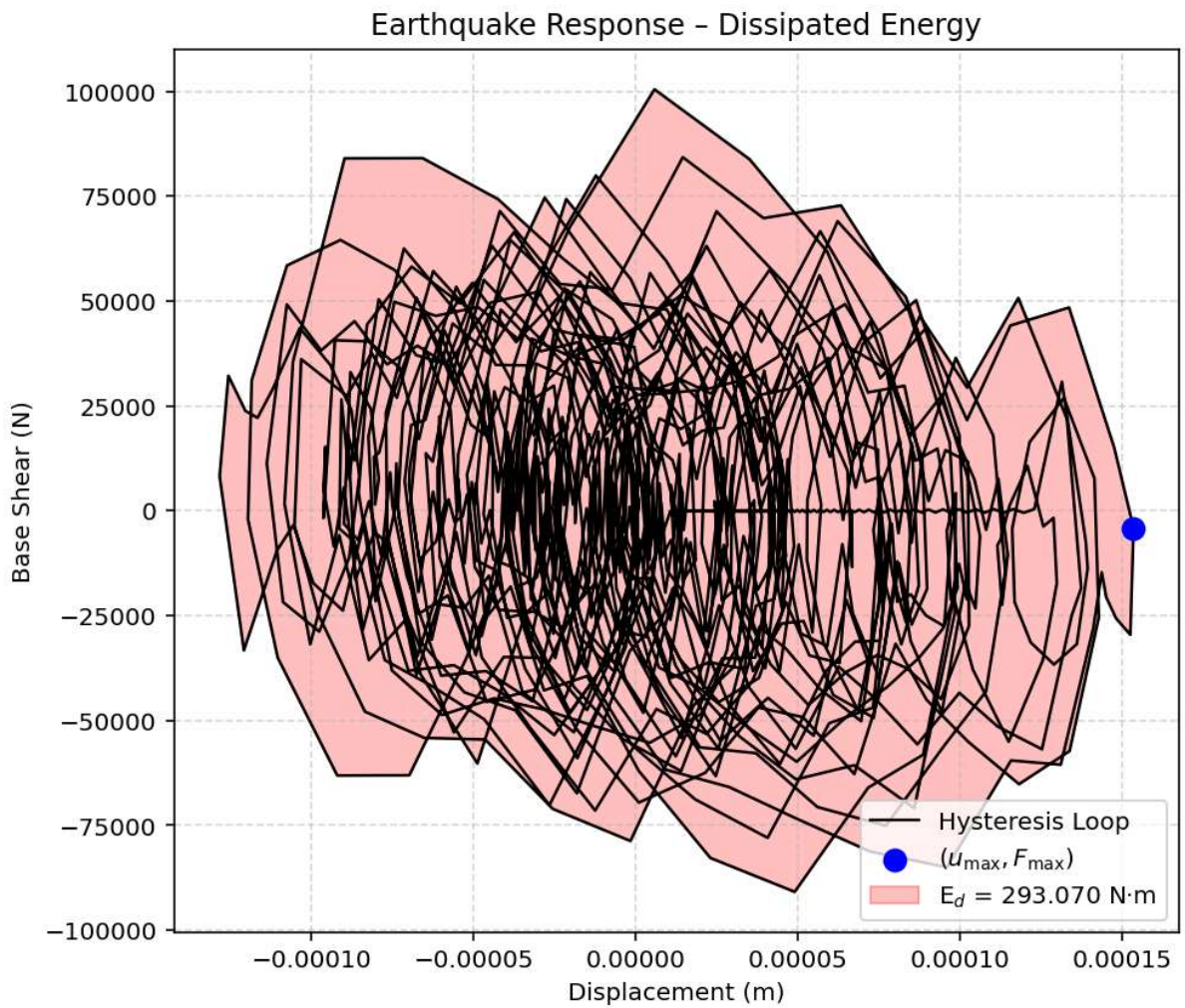

Histogram - Probability of Positive REACTION FROM SEISMIC ANALYSIS is 31.60 %
Normal PDF
Quantile 0.25: -4.723482e+02
Median: -7.320974e+01
Quantile 0.75: 3.043488e+02
Mean: -2.755662e+02
Mean-Std: -2.332347e+04
Mean+Std: 2.277233e+04
Frequency
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
−100000
−75000
−50000
−25000
0
25000
50000
75000
100000
REACTION FROM SEISMIC ANALYSIS

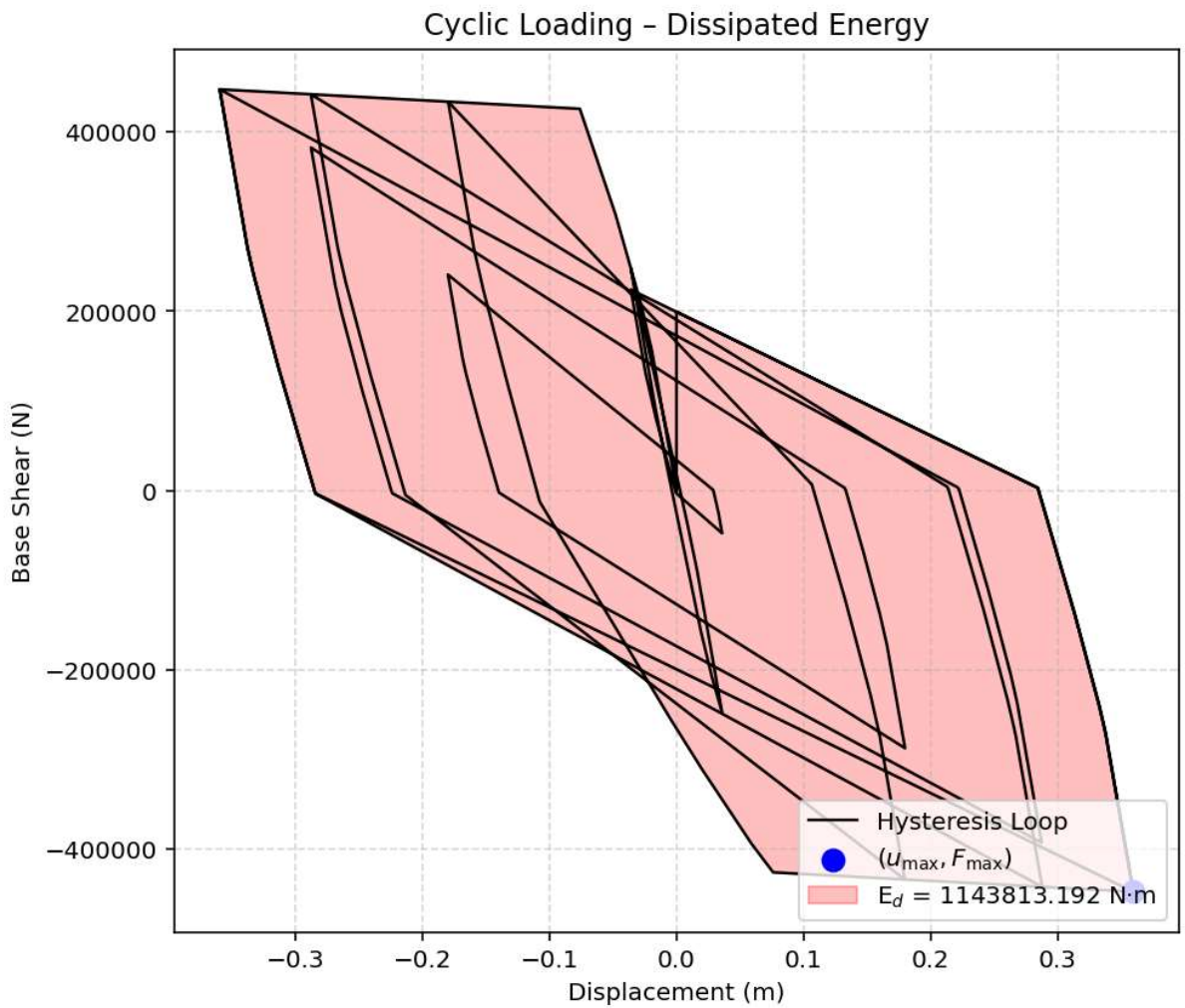

Boxplot of REACTION FROM SEISMIC ANALYSIS
Mean: -2.755662e+02
Mean-Std: -2.332347e+04
Mean+Std: 2.277233e+04
Q2: -7.320974e+01
Data
1
−100000
−75000
−50000
−25000
0
25000
50000
75000
100000
REACTION FROM SEISMIC ANALYSIS


Histogram - Probability of Positive DUCTILITY DAMAGE INDEX FROM SEISMIC ANALYSIS is 0.00 %
Normal PDF
Quantile 0.25: 0.000000e+00
Median: 0.000000e+00
Quantile 0.75: 0.000000e+00
Mean: 0.000000e+00
Mean-Std: 0.000000e+00
Mean+Std: 0.000000e+00
Frequency
0
20
40
60
80
100
−0.4
−0.2
0.0
0.2
0.4
DUCTILITY DAMAGE INDEX FROM SEISMIC ANALYSIS


Boxplot of DUCTILITY DAMAGE INDEX FROM SEISMIC ANALYSIS
Mean: 0.000000e+00
Mean-Std: 0.000000e+00
Mean+Std: 0.000000e+00
Q2: 0.000000e+00
Q3: 0.000000e+00
Data
1
−0.04
−0.02
0.00
0.02
0.04
DUCTILITY DAMAGE INDEX FROM SEISMIC ANALYSIS

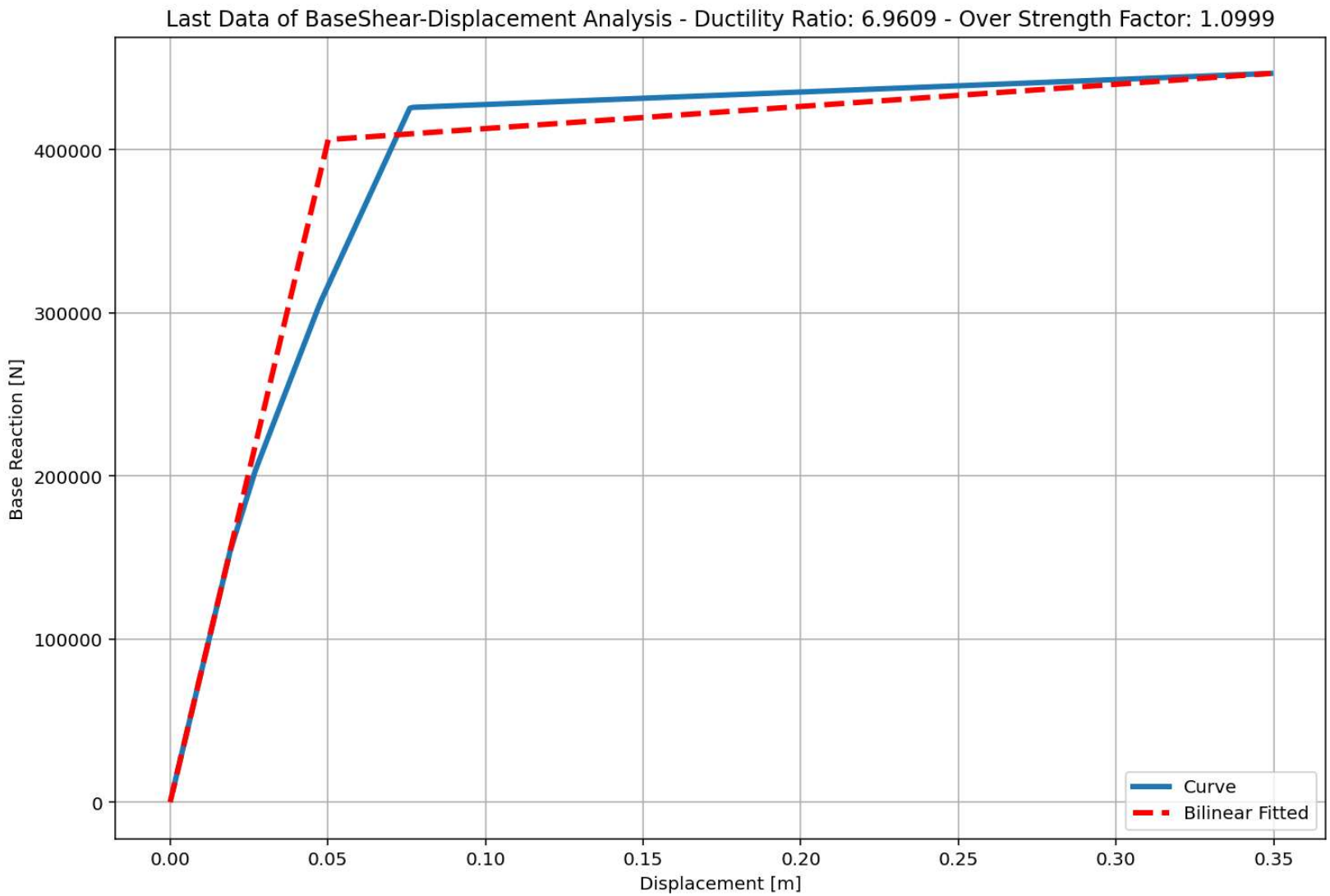

Equivalent viscous damping ratio - Seismic Hysteresis
$\xi_{eq} = 1029.5947$ [%]
$E_d = 40.940$ N·m
$E_s = 0.316$ N·m
Hysteresis Loop
(u_{max}, F_{max})
E_d = 40.940 N·m
Base Reaction [N]
Displacement [m]
100000
75000
50000
25000
0
−25000
−50000
−75000
−100000
−0.00010
−0.00005
0.00000
0.00005
0.00010
0.00015


Earthquake Response – Dissipated Energy
Base Shear (N)
Displacement (m)
100000
75000
50000
25000
0
−25000
−50000
−75000
−100000
−0.00010
−0.00005
0.00000
0.00005
0.00010
0.00015
Hysteresis Loop
(u_{max}, F_{max})
E_d = 293.070 N·m

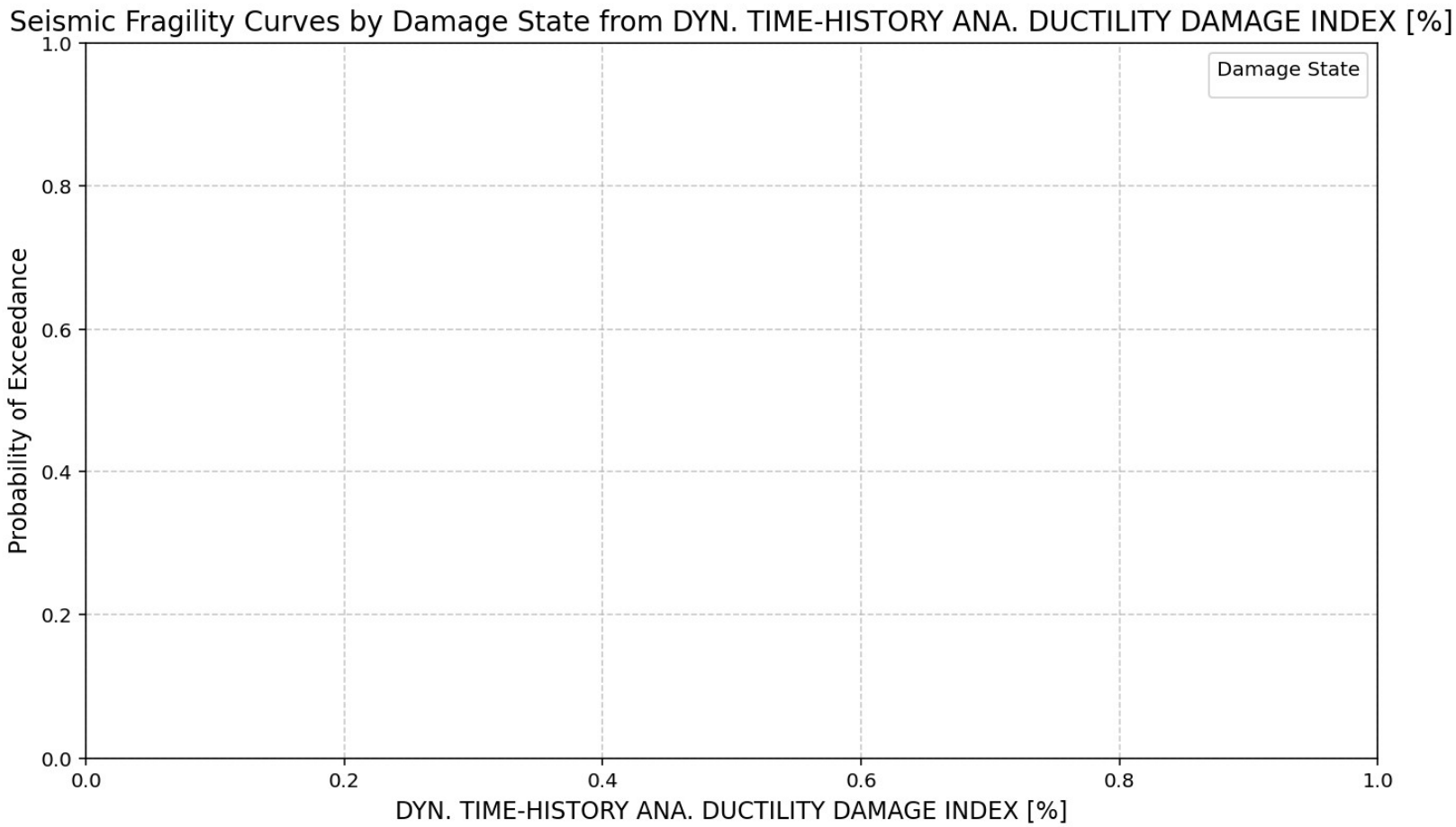

Cyclic Loading – Dissipated Energy
Base Shear (N)
Displacement (m)
400000
200000
0
−200000
−400000
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Hysteresis Loop
(u_{max}, F_{max})
E_d = 1143813.192 N·m

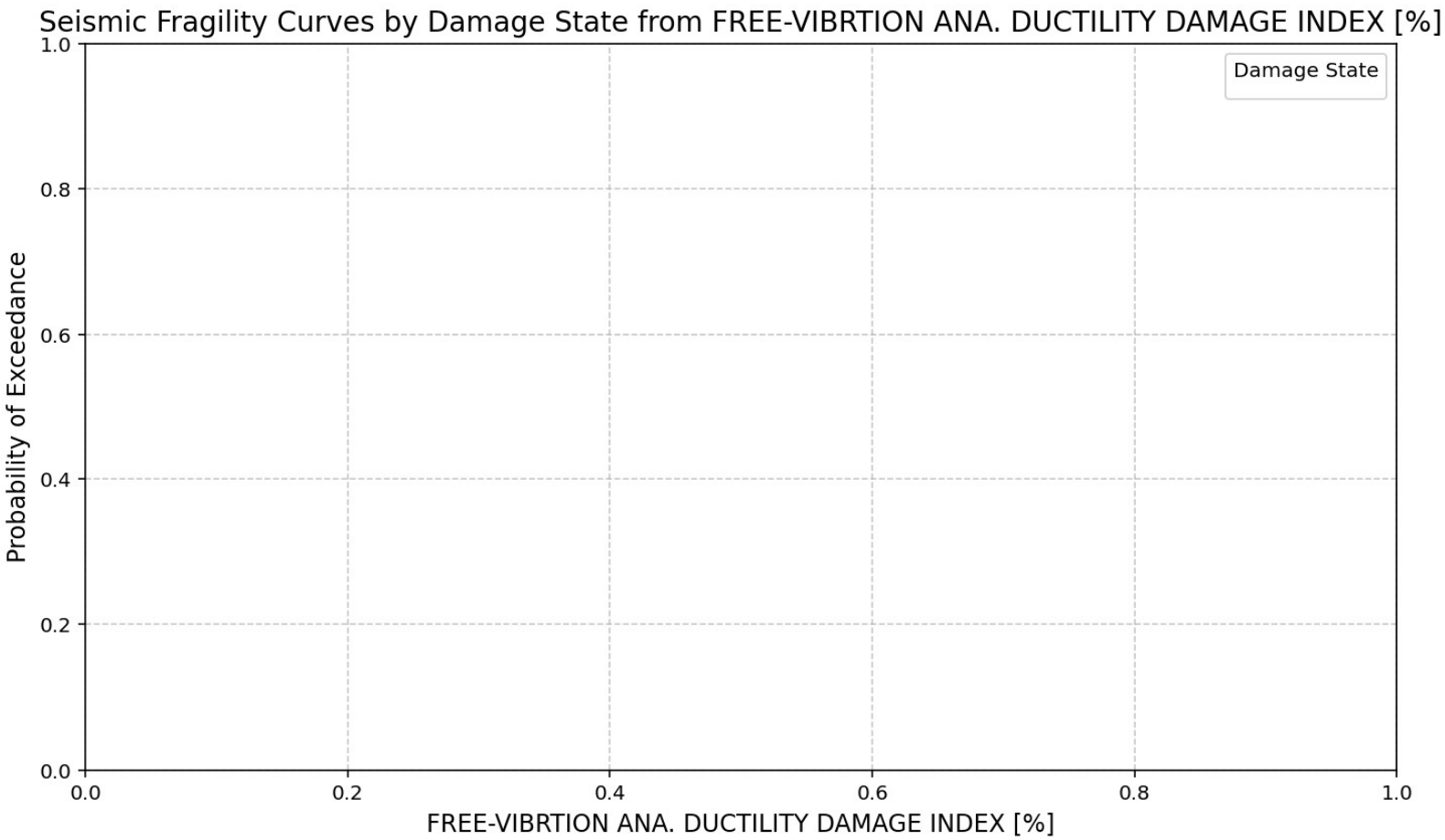
Dissipated Energy from Earthquake= 293.07 N·m
Dissipated Energy from Cyclic Displacement= 1143813.19 N·m

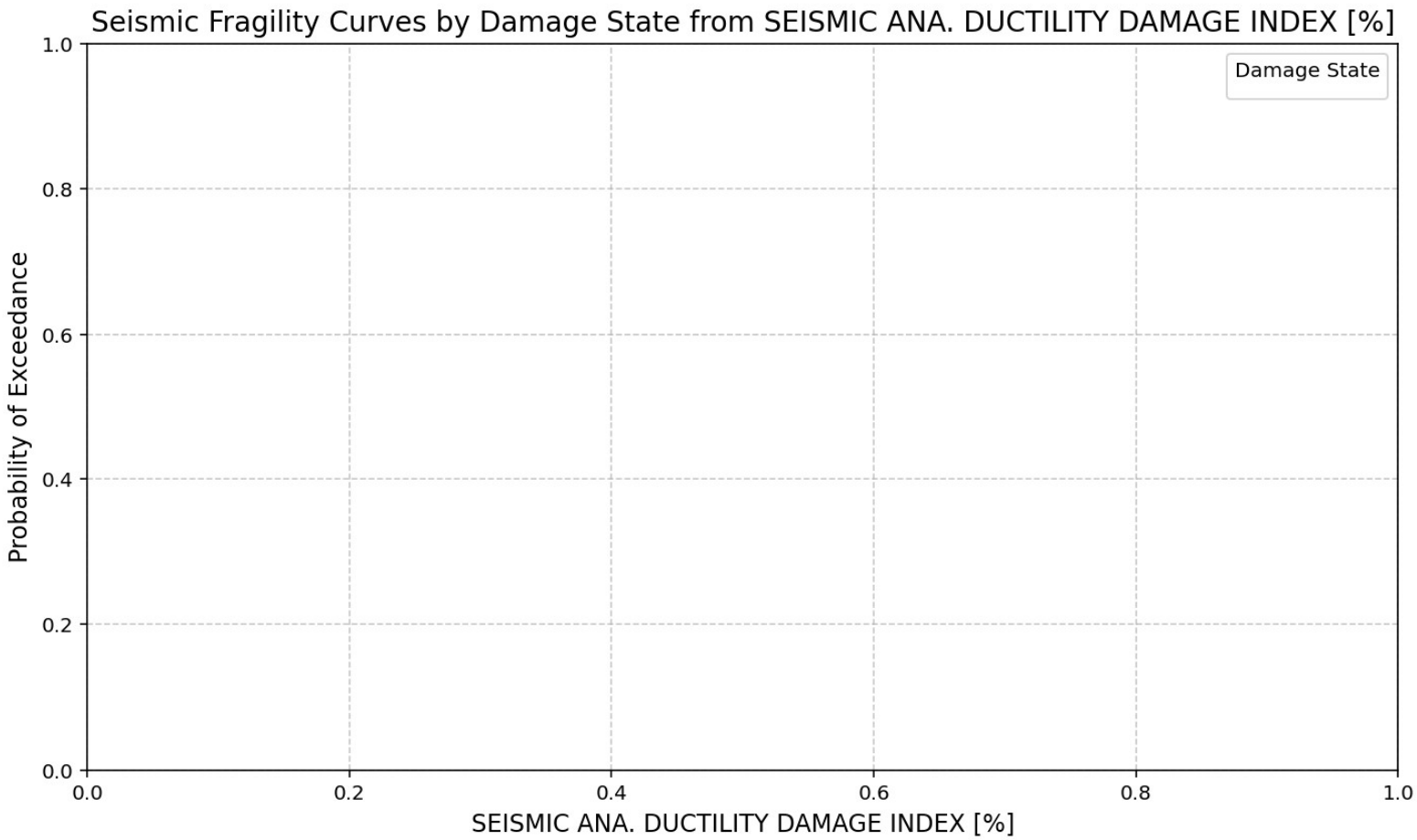
DISSIPATED ENERGY CAPACITY INDEX: 0.026 [%]
ZONE 1: VERY MINOR DAMAGE


Last Data of BaseShear-Displacement Analysis - Ductility Ratio: 6.9609 - Over Strength Factor: 1.0999
Base Reaction [N]
400000
300000
200000
100000
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Displacement [m]
Curve
Bilinear Fitted

```
+==========================+
=   Analysis curve fitted =
    Disp       Base Shear
----------------------------
[[0.00000000e+00 0.00000000e+00]
 [5.01374697e-02 4.06113504e+05]
 [3.49000000e-01 4.46693631e+05]]
+==========================+
+-----------------------------------------------------+
 Structure Elastic Stiffness :    8100000.00
 Structure Plastic Stiffness :    1279924.45
 Structure Tangent Stiffness :    135781.92
 Structure Ductility Ratio :      6.96
 Structure Over Strength Factor:   1.10
+-----------------------------------------------------+
Over Strength Coefficient (Ω0):     1.0999
Displacement Ductility Ratio (μ):   6.9609
Ductility Coefficient (Rμ):        6.9609
Structural Behavior Coefficient (R): 7.6564
Structural Ductility Damage Index in X Direction: -16.7248 (%)
```


Seismic Fragility Curves by Damage State from DYN. TIME-HISTORY ANA. DUCTILITY DAMAGE INDEX [%]
Damage State
Probability of Exceedance
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
DYN. TIME-HISTORY ANA. DUCTILITY DAMAGE INDEX [%]


Seismic Fragility Curves by Damage State from FREE-VIBRTION ANA. DUCTILITY DAMAGE INDEX [%]
Damage State
Probability of Exceedance
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
FREE-VIBRTION ANA. DUCTILITY DAMAGE INDEX [%]


Seismic Fragility Curves by Damage State from SEISMIC ANA. DUCTILITY DAMAGE INDEX [%]
Damage State
Probability of Exceedance
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
SEISMIC ANA. DUCTILITY DAMAGE INDEX [%]