Assignment 2 – Code Performance Grade

"Second Try"

Daniel Jones

Problem 1 Results (part 1 simple check):

Problem setup used:

```
def get_problem_setup():

    # Nodal positions
    nodes = np.array([
        [0,  0,   0],   # N0 (Fixed Support)
        [3,  28/3,  22/3],   # N1
        [6,  56/3,  44/3],   # N2
        [9,  84/3,  66/3],   # N3
        [12, 112/3, 88/3],   # N4
        [15, 140/3, 110/3],  # N5
        [18, 56, 44]         # N6 (Free End)
    ], dtype=float)

    # Material & section properties
    r = 1  # Radius of circular cross-section
    E = 10000  # Young's Modulus
    nu = 0.3  # Poisson's ratio

    # Computed section properties
    A = np.pi * r**2
    I_y = np.pi * r**4 / 4.0
    I_z = np.pi * r**4 / 4.0
    I_rho = np.pi * r**4 / 2.0
    J = np.pi * r**4 / 2.0

    # Element connectivity:
    connection = np.array([
        [
            i, i+1, E, nu, A, I_y, I_z, I_rho, J,
            [
                -(nodes[i+1][1] - nodes[i][1]) / np.sqrt((nodes[i+1][0] - nodes[i][0])**2 +
(nodes[i+1][1] - nodes[i][1])**2),
                (nodes[i+1][0] - nodes[i][0]) / np.sqrt((nodes[i+1][0] - nodes[i][0])**2 +
(nodes[i+1][1] - nodes[i][1])**2),
                0
            ]
        ]
        for i in range(len(nodes) - 1)
    ], dtype=object)

    # Supports:
```

```
    supports = np.array([
        [0, 1,1,1, 1,1,1],  # N0 (Fully fixed)
        [1, 0,0,0, 0,0,0],  # N1 (Free)
        [2, 0,0,0, 0,0,0],  # N2 (Free)
        [3, 0,0,0, 0,0,0],  # N3 (Free)
        [4, 0,0,0, 0,0,0],  # N4 (Free)
        [5, 0,0,0, 0,0,0],  # N5 (Free)
        [6, 0,0,0, 0,0,0]   # N6 (Free end)
    ])

    # External loads:
    loads = np.zeros((7,6))  # Initialize 7 nodes with zero force
    loads[6] = [0.05, -0.1, 0.23, 0.1, -0.025, -0.08]  # Apply force & moment at N6

    return nodes, connection, loads, supports
```

Code Output:

```
ython.exe c:/Users/danie/OneDrive/Desktop/ME700/ME700-Assignment-2/src/Critical_Load_Analy
sis/criticalloadanalysis_tutorial.py
Nodal Displacements & Rotations:
  Node 0: U=(0.000000e+00, 0.000000e+00, 0.000000e+00), R=(0.000000e+00, 0.000000e+00, 0.0
00000e+00)
  Node 1: U=(2.157731e-02, -1.041745e-01, 1.238067e-01), R=(2.484266e-02, -2.826207e-03, -
6.709610e-03)
  Node 2: U=(8.132309e-02, -3.924037e-01, 4.662503e-01), R=(4.519593e-02, -5.148397e-03, -
1.222413e-02)
  Node 3: U=(1.717793e-01, -8.281799e-01, 9.839178e-01), R=(6.105981e-02, -6.966569e-03, -
1.654355e-02)
  Node 4: U=(2.854878e-01, -1.374996e+00, 1.633396e+00), R=(7.243430e-02, -8.280725e-03, -
1.966789e-02)
  Node 5: U=(4.149906e-01, -1.996344e+00, 2.371272e+00), R=(7.931940e-02, -9.090863e-03, -
2.159713e-02)
  Node 6: U=(5.528296e-01, -2.655717e+00, 3.154134e+00), R=(8.171511e-02, -9.396984e-03, -
2.233128e-02)

Reaction Forces & Moments at Constrained Nodes:
  Node 0: F=(-5.000000e-02, 1.000000e-01, -2.300000e-01), M=(-1.738000e+01, 1.965000e+00,
4.680000e+00)
Critical Load Factor = 24.83051
```

Reaction force at Node 0: Node 0: Reaction Force (Fx, Fy, Fz) = (0.000000e+00, 0.000000e+00, 0.000000e+00)

Reaction moment at Node 0: Node 0: Moment (Mx, My, Mz) = (0.000000e+00, 0.000000e+00, 0.000000e+00)

Displacement at Node 3: Node 3: Displacement (x, y, z) = (1.717793e-01, -8.281799e-01, 9.839178e-01)

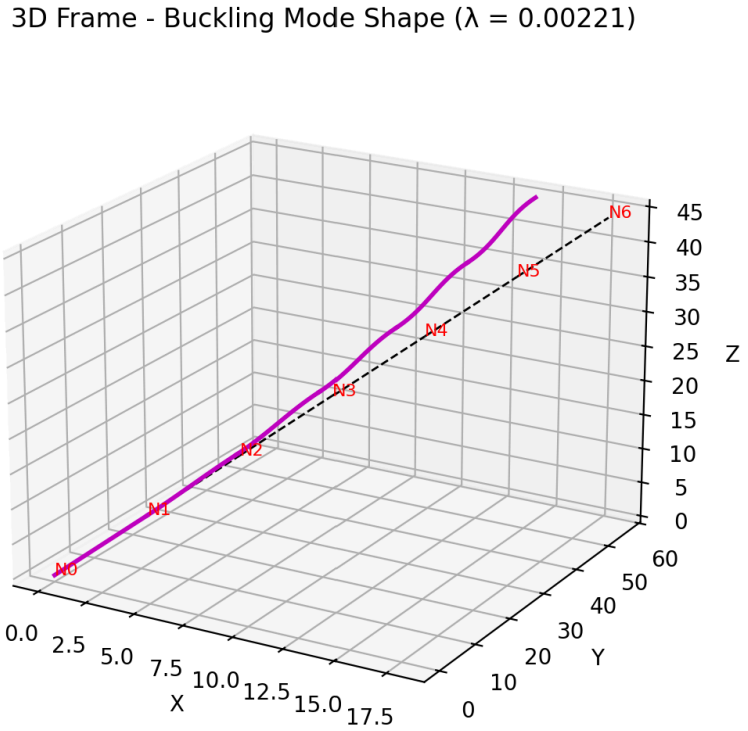
Rotation at Node 3: Node 3: Rotation (rx, ry, rz) = (6.105981e-02, -6.966569e-03, -1.654355e-02)

Displacement at Node 6: Node 6: Displacement (x, y, z) = (5.528296e-01, -2.655717e+00, 3.154134e+00)

Rotation at Node 6: Node 6: Rotation (rx, ry, rz) = (8.171511e-02, -9.396984e-03, -2.233128e-02)

Elastic Critical Load factor (λ): 24.83051

Plot of structure interpolated deformed shape:



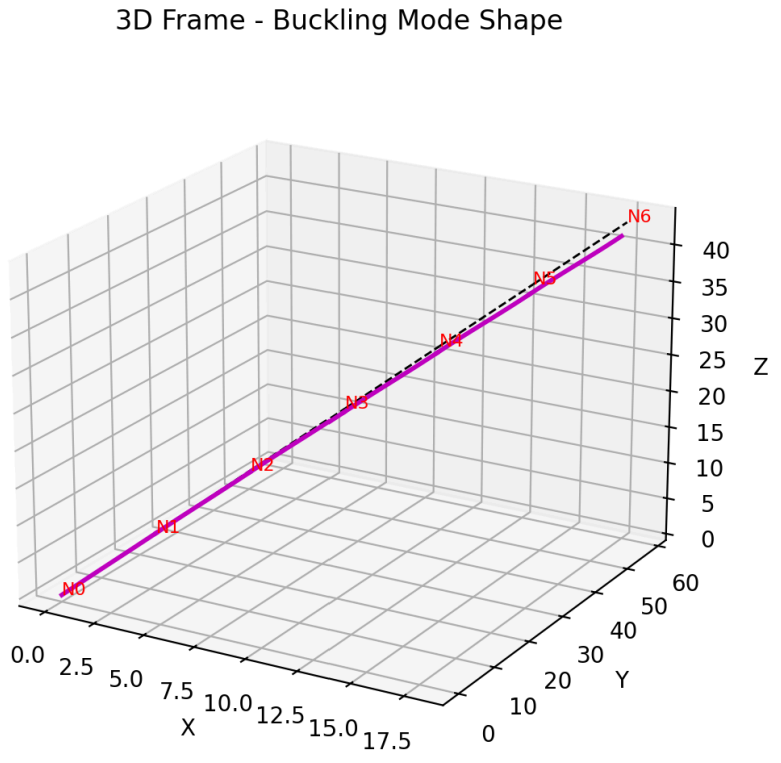
Problem 2 Results (part 2 simple check):

Problem setup used: (Same as part 1, except for loads)

```
# External loads:
# Load Constants
x1 = 18
x0 = 0
y1 = 56
y0 = 0
z1 = 44
z0 = 0
P = 1
L = np.sqrt((x1-x0) ** 2.0 + (y1 - y0) ** 2.0 + (z1 - z0) ** 2.0_)
Fx = -1.0 * P * (x1 - x0) / L
Fy = -1.0 * P * (y1 - y0) / L
Fz = -1.0 * P * (z1 - z0) / L
loads = np.zeros((7,6))  # Initialize 7 nodes with zero force
loads[6] = [Fx, Fy, Fz, 0.0, 0.0, 0.0]  # Apply force & moment at N6
```

Elastic Critical Load factor (λ): 3.59137

Plot of structure interpolated deformed shape:



Problem 3 Results (complete code check):

Problem setup used:

```python
def get_problem_setup():
    # Length Constants
    L1 = 15.0
    L2 = 30.0
    L3 = 14.0
    L4 = 16.0
    # Nodal positions
    nodes = np.array([
        [0, 0, 0],          # N0
        [L1, 0, 0],         # N1
        [L1, L2, 0],        # N2
        [0, L2, 0],         # N3
        [0, 0, L3],         # N4
        [L1, 0, L3],        # N5
        [L1, L2, L3],       # N6
        [0, L2, L3],        # N7
        [0, 0, L3+L4],      # N8
        [L1, 0, L3+L4],     # N9
        [L1, L2, L3+L4],    # N10
        [0, L2, L3+L4]      # N11
    ])

    # Material & section properties
    # Element Type A Properties
    E_a = 10000
    nu_a = 0.3
    r_a = 1
    A_a = np.pi * r_a**2
    I_y_a = np.pi * r_a**4 / 4.0
    I_z_a = np.pi * r_a**4 / 4.0
    I_rho_a = np.pi * r_a**4 / 2.0
    J_a = np.pi * r_a**4 / 2.0
    # Element Type B Properties
    E_b = 50000
    nu_b = 0.3
    b = 0.5
    h = 1
    A_b = b * h
    I_y_b = (b * h**3) / 12.0
    I_z_b = (h * b**3) / 12.0
    I_rho_b = (b * h) / 12.0 * (b**2 + h**2)
    J_b = 0.028610026401666667

    # Element connectivity:
    connection = np.array([
        # Vertical elements (Type A)
        [0, 4, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],
        [1, 5, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],
        [3, 7, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],
        [2, 6, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],
        [4, 8, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],
        [5, 9, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],
        [7, 11, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],
        [6, 10, E_a, nu_a, A_a, I_y_a, I_z_a, I_rho_a, J_a, [1,0,0]],

        # Horizontal elements (Type B)
        [4, 5, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
        [5, 6, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
        [6, 7, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
        [7, 4, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
        [8, 9, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
```

```
        [9, 10, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
        [10, 11, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
        [11, 8, E_b, nu_b, A_b, I_y_b, I_z_b, I_rho_b, J_b, [0,0,1]],
    ], dtype=object)

    # Supports:
    supports = np.array([
        [0, 1,1,1, 1,1,1],  # N0 fully fixed
        [1, 1,1,1, 1,1,1],  # N1 fully fixed
        [2, 1,1,1, 1,1,1],  # N2 fully fixed
        [3, 1,1,1, 1,1,1],  # N3 fully fixed
        [4, 0,0,0, 0,0,0],  # N4 is free
        [5, 0,0,0, 0,0,0],  # N5 is free
        [6, 0,0,0, 0,0,0],  # N6 is free
        [7, 0,0,0, 0,0,0],   # N7 is free
        [8, 0,0,0, 0,0,0],  # N8 is free
        [9, 0,0,0, 0,0,0],  # N9 is free
        [10, 0,0,0, 0,0,0],  # N10 is free
        [11, 0,0,0, 0,0,0]   # N11 is free
    ])

    # External loads:
    loads = np.zeros((12,6))
    loads[8] = [0, 0, -1, 0, 0, 0]  # N8
    loads[9] = [0, 0, -1, 0, 0, 0]  # N9
    loads[10] = [0, 0, -1, 0, 0, 0] # N10
    loads[11] = [0, 0, -1, 0, 0, 0] # N11

    return nodes, connection, loads, supports
```

Code Output:

```
(me700) PS C:\Users\danie\OneDrive\Desktop\ME700> & C:/Users/danie/miniconda3/envs/me700/python.exe c:/Users/danie/OneDrive/Desktop/ME700/ME700-Assignment-2/src/Critical_Load_Analysis/criticalloadanalysis_tutorial.py
Nodal Displacements & Rotations:
  Node 0: U=(0.000000e+00, 0.000000e+00, 0.000000e+00), R=(0.000000e+00, 0.000000e+00, 0.000000e+00)
  Node 1: U=(0.000000e+00, 0.000000e+00, 0.000000e+00), R=(0.000000e+00, 0.000000e+00, 0.000000e+00)
  Node 2: U=(0.000000e+00, 0.000000e+00, 0.000000e+00), R=(0.000000e+00, 0.000000e+00, 0.000000e+00)
  Node 3: U=(0.000000e+00, 0.000000e+00, 0.000000e+00), R=(0.000000e+00, 0.000000e+00, 0.000000e+00)
  Node 4: U=(6.565513e-19, 1.388410e-20, -4.456338e-04), R=(-1.443382e-21, 5.544115e-20, 1.477038e-21)
  Node 5: U=(6.601852e-19, 7.322498e-21, -4.456338e-04), R=(-2.003055e-21, 5.577283e-20, 5.887916e-22)
  Node 6: U=(4.337091e-19, 7.321890e-21, -4.456338e-04), R=(-1.654426e-21, 4.489518e-20, 1.642139e-21)
  Node 7: U=(4.355353e-19, 1.338474e-20, -4.456338e-04), R=(-1.024312e-21, 4.411357e-20, 6.591211e-22)
  Node 8: U=(1.514138e-18, 5.001107e-20, -9.549297e-04), R=(1.154499e-22, 3.432910e-20, 2.357401e-21)
  Node 9: U=(1.514197e-18, 5.067461e-20, -9.549297e-04), R=(-4.884569e-21, 3.419947e-20, 4.956246e-22)
  Node 10: U=(1.252682e-18, 5.023801e-20, -9.549297e-04), R=(-5.863027e-21, 5.201404e-20, 2.778587e-21)
  Node 11: U=(1.251669e-18, 5.046657e-20, -9.549297e-04), R=(-8.945510e-22, 4.835508e-20, -3.449668e-22)

Reaction Forces & Moments at Constrained Nodes:
  Node 0: F=(-9.220869e-18, -1.298458e-19, 1.000000e+00), M=(1.718656e-18, -9.564849e-17, -6.373972e-19)
  Node 1: F=(-9.265936e-18, 2.300858e-19, 1.000000e+00), M=(-4.868891e-19, -9.615004e-17, -2.540856e-19)
  Node 2: F=(-4.102488e-18, 1.462864e-19, 1.000000e+00), M=(-9.587403e-20, -5.390355e-17, -7.086445e-19)
  Node 3: F=(-4.353133e-18, -2.134504e-19, 1.000000e+00), M=(2.068790e-18, -5.521958e-17, -2.844354e-19)
Critical Load Factor = 56.54756
```

Elastic Critical Load factor (λ): 56.54756
Plot of structure interpolated deformed shape: