Solutions to Homework 2

Question 1: 10 points.

The cantilevered frame structure carries vertical loads P kN at notes D, F, H and I. Use the **method of sections** to determine the forces in members a, b and c, as a function of the problem parameters L and P.

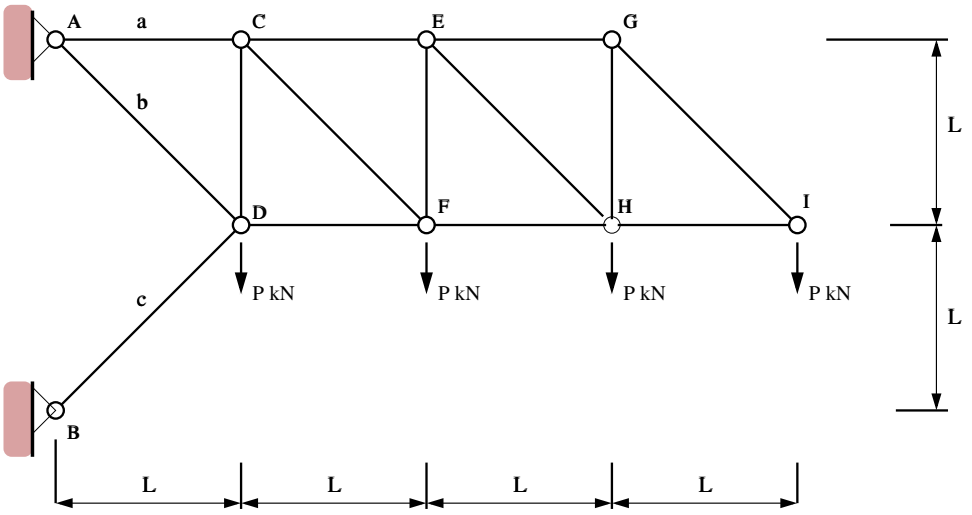

Figure 1: Cantilevered frame structure.

State if the members are in tension or compression.

Solution: We begin by cutting the structure as shown in Figure 1.

Taking moments about D for the right-hand substructure gives:

$$\sum M_D = 0 \qquad PL + 2PL + 3PL = LF_{ac} \longrightarrow F_{ac} = 6P(T). \tag{1}$$

Taking moments about A for the complete structure gives:

$$\sum M_A = 0 \qquad PL + 2PL + 3PL + 4PL = 2LH_b \longrightarrow H_b = 5P. \tag{2}$$

Next, observe that the orientation of element B-D is $\pi/4$ radians. Hence, the horizontal and vertical components of reaction force at B need to be equal, i.e., V_b = 5P and $F_{bd} = -5\sqrt{2}P(C)$.

Looking at the equilibrium of the structure at A, the horizontal and vertical reaction forces are: V_a = -P and H_a = -5P. Summing forces in the vertical direction at A, member force $F_{ad} = -\sqrt{2}P(C)$.

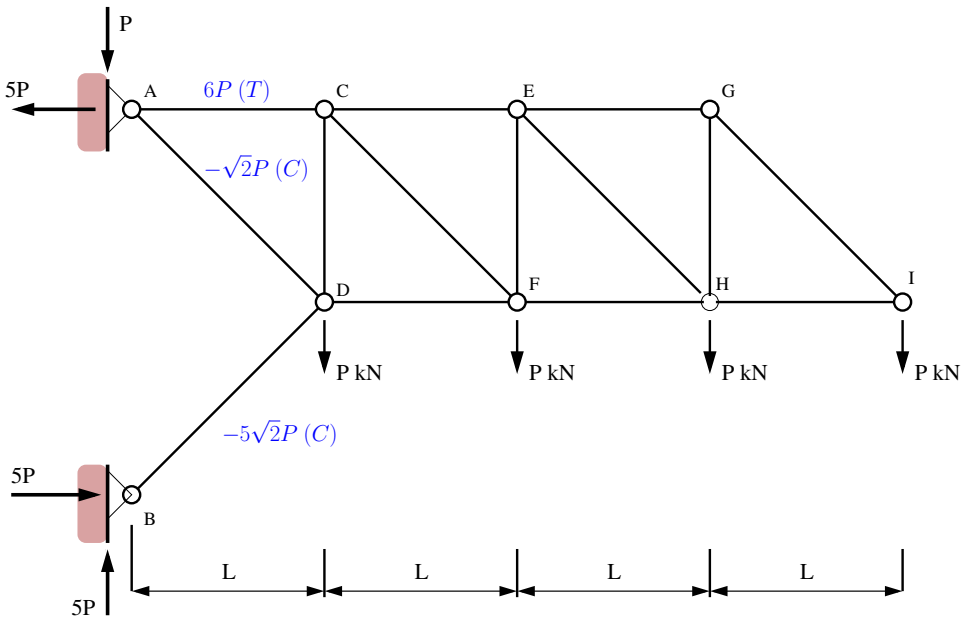

Figure 2: Critical forces in cantilevered frame structure.

Summary of Results: $F_{ac} = 6P(T)$, $F_{bd} = -5\sqrt{2}P(C)$, $F_{ad} = -\sqrt{2}P(C)$.

Question 2: 10 points.

If the maximum tensile force that any member in Figure 1 can carry is 15 kN, and the maximum compressive force is 10 kN (before buckling), what is the maximum value of loads P that can be carried.

Soln's. Maximum tensile and compressive forces are 6P (kN) and -10P (kN), respectively.

Tension constraint: P_{max} = 15/6 kN = 2.5 kN.

Compression constraint: P_{max} = -10/-10 kN = 1.0 kN.

Therefore, maximum allowable load is P_{max} = 1 kN.

Question 3: 10 points.

The cantilevered beam carries a vertical load P kN at D, and a uniform load W (N/m) along the segment A-B-C. The total loading on A-B-C will be 2WL. The two beam segments are connected by a hinge at point B. Determine the shear and bending moment throughout the beam as a function of P, W and L. Draw the shear and bending moments for the beam. Qualitatively sketch the deflected shape of the beam.

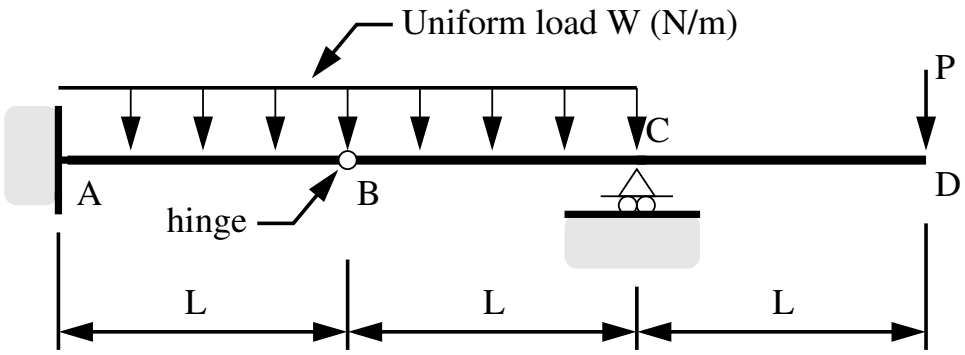

Figure 3: Cantilever beam with hinge connection to supported beam structure.

Soln's. Taking moments about B for the right-hand substructure:

$$\sum M_B = 0, \quad 2PL + (wL)(L/2) = V_C L \longrightarrow V_C = 2P + \frac{wL}{2}. \tag{3}$$

Next, consider vertical equilibrium of right-hand substructure:

$$\sum V = 0, \quad V_B + V_C = WL + P, \longrightarrow V_B = \frac{wL}{2} - P. \tag{4}$$

Taking moments about A:

$$\sum M_A = 0, \quad M_A = \frac{wL^2}{2} + \left[\frac{wL}{2} - P\right] \longrightarrow M_A = wL^2 - PL. \tag{5}$$

The shear force at B is wL/2 - P, which leads to:

$$\sum V = 0, \quad V_A = \frac{3wL}{2} - P. \tag{6}$$

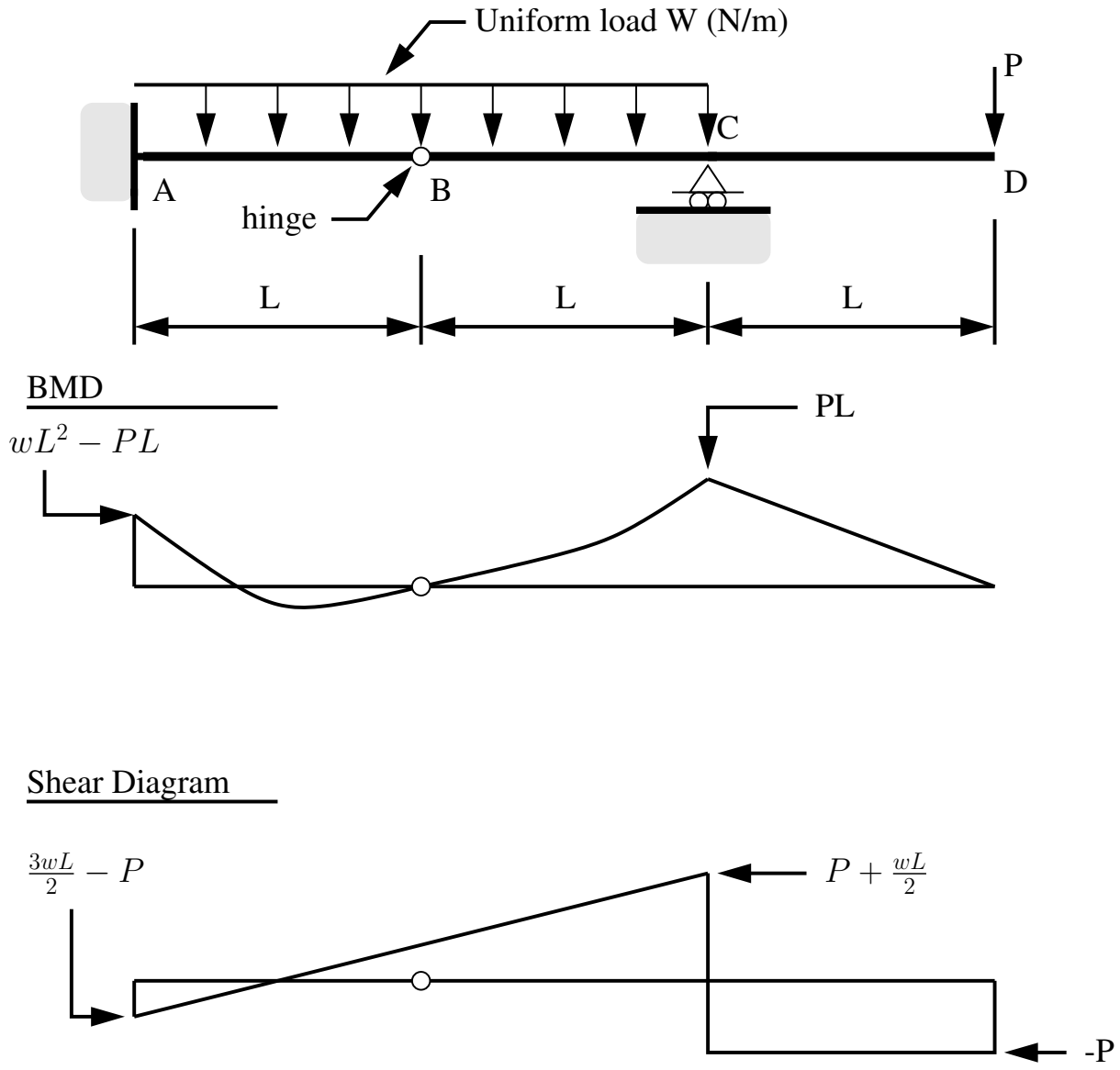

Figure 4: BMD and shear force diagrams.

Question 4: 20 points. In the design of crane structures, engineers are often required to compute the maximum and minimum member forces and support reactions due to a variety of loading conditions.

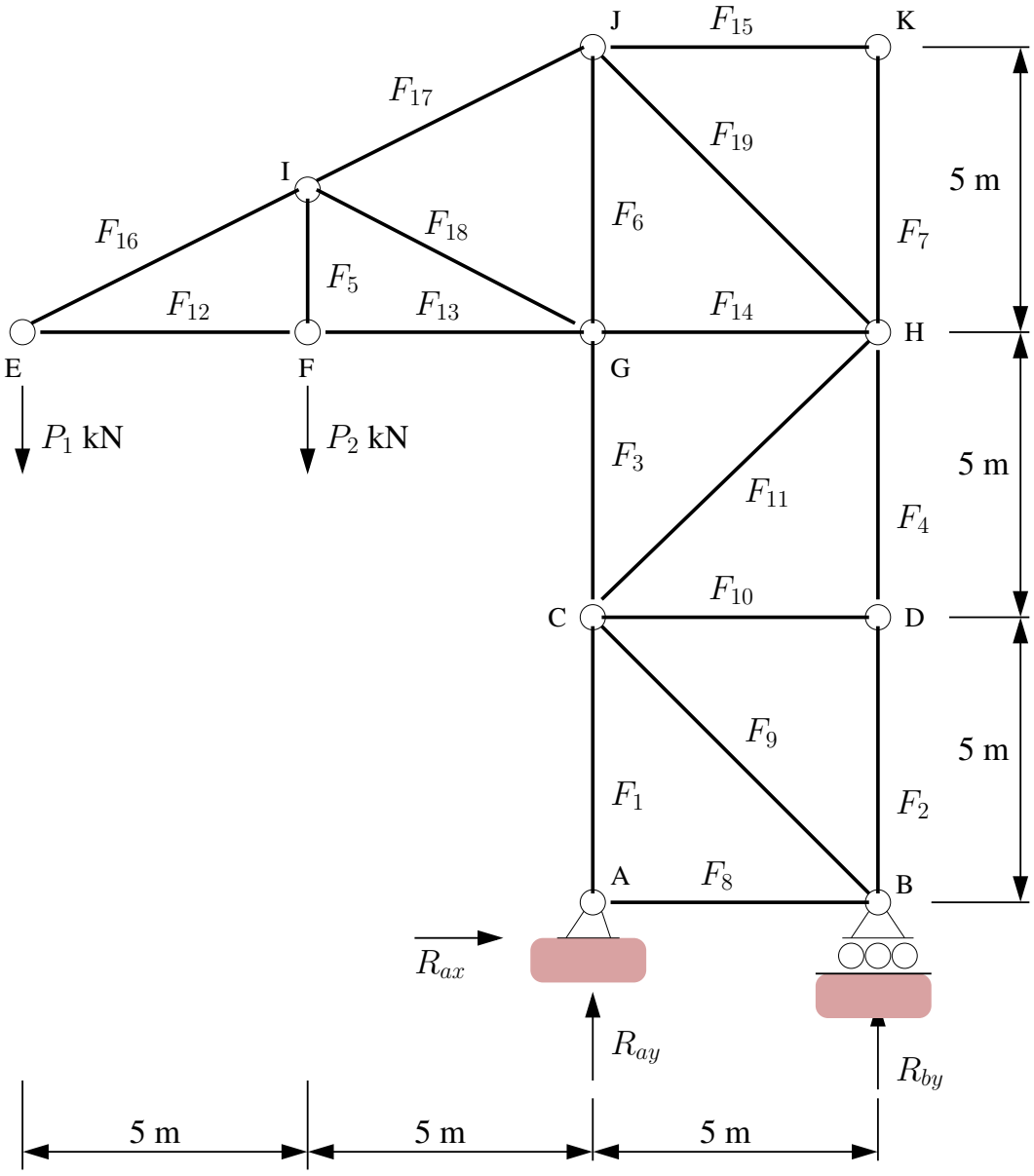

Figure 5: Front elevation of crane tower.

Figure 5 shows a nineteen bar pin-jointed truss carrying vertical loads P_1 kN and P_2 kN at joints `E` and `F`. The symbols F_1, F_2, $\cdots$ F_{19} represent the axial forces in truss members 1 through 19, and R_{ax},R_{ay}, and R_{dy} are the horizontal and vertical support reactions at joints `A` and `B`.

Write down the equations of equilibrium for joints `A` through `K` and put the equations in matrix

form. Now suppose that the crane supports a variety of payloads that, for engineering purposes, it can be represented by the sequence of external load vectors

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 250 \\ 0 \end{bmatrix}, \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 200 \\ 200 \end{bmatrix}, \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 300 \end{bmatrix} kN. \tag{7}$$

Develop a Python program that will solve the matrix equations for each of the external load conditions, and compute and print the minimum and maximum support reactions at nodes A and B, and axial forces in each of the truss members.

Equations of Equilibrium:

At Joint A:

$$\sum F_x = 0, \quad R_{ax} + F_8 = 0. \tag{8}$$

$$\sum F_y = 0, \quad R_{ay} + F_1 = 0. \tag{9}$$

At Joint B:

$$\sum F_x = 0, \quad F_8 + \frac{1}{\sqrt{2}} F_9 = 0. \tag{10}$$

$$\sum F_y = 0, \quad F_2 + \frac{1}{\sqrt{2}} F_9 + R_{by} = 0. \tag{11}$$

At Joint C:

$$\sum F_x = 0, \quad \frac{1}{\sqrt{2}} F_9 + F_{10} + \frac{1}{\sqrt{2}} F_{11} = 0. \tag{12}$$

$$\sum F_y = 0, \quad F_1 - F_3 + \frac{1}{\sqrt{2}} F_9 - \frac{1}{\sqrt{2}} F_{11} = 0. \tag{13}$$

At Joint D:

$$\sum F_x = 0, \quad F_{10} = 0. \tag{14}$$

$$\sum F_y = 0, \quad F_2 - F_4 = 0. \tag{15}$$

At Joint E:

$$\sum F_x = 0, \quad F_{12} + \frac{2}{\sqrt{5}} F_{16} = 0. \tag{16}$$

$$\sum F_y = 0, \quad \frac{1}{\sqrt{5}} F_{16} + P_1 = 0. \tag{17}$$

At Joint F:

$$\sum F_x = 0, \quad F_{12} - F_{13} = 0. \tag{18}$$

$$\sum F_y = 0, \quad F_5 - P_2 = 0. \tag{19}$$

At Joint G:

$$\sum F_x = 0, \quad F_{13} - F_{14} + \frac{2}{\sqrt{5}} F_{18} = 0. \tag{20}$$

$$\sum F_y = 0, \quad -F_3 + F_6 + \frac{1}{\sqrt{5}} F_{18} = 0. \tag{21}$$

At Joint H:

$$\sum F_x = 0, \quad \frac{1}{\sqrt{2}} F_{11} + F_{14} + \frac{1}{\sqrt{2}} F_{19} = 0. \tag{22}$$

$$\sum F_y = 0, \quad F_4 - F_7 + \frac{1}{\sqrt{2}} F_{11} - \frac{1}{\sqrt{2}} F_{19} = 0. \tag{23}$$

At Joint I:

$$\sum F_x = 0, \quad \frac{2}{\sqrt{5}} F_{16} - \frac{2}{\sqrt{5}} F_{17} - \frac{2}{\sqrt{5}} F_{18} = 0. \tag{24}$$

$$\sum F_y = 0, \quad F_5 + \frac{1}{\sqrt{5}}F_{16} - \frac{1}{\sqrt{5}}F_{17} + \frac{1}{\sqrt{5}}F_{18} = 0. \tag{25}$$

At Joint J:

$$\sum F_x = 0, \quad F_{15} - \frac{2}{\sqrt{5}}F_{17} + \frac{1}{\sqrt{2}}F_{19} = 0. \tag{26}$$

$$\sum F_y = 0, \quad F_6 + \frac{1}{\sqrt{5}}F_{17} + \frac{1}{\sqrt{2}}F_{19} = 0 \tag{27}$$

At Joint K:

$$\sum F_x = 0, \quad F_{15} = 0. \tag{28}$$

$$\sum F_y = 0, \quad F_7 = 0. \tag{29}$$

Python Source Code:

```
# ====================================================================
# TestTrussAnalysisCrane03.py: Compute distribution of element forces
# and support reactions in a 19-bar crane truss.
#
# Written by: Mark Austin                                March, 2026
# ====================================================================

import math
import numpy as np
from numpy.linalg import matrix_rank

# ============================================================
# Function to print one- and two-dimensional matrices ...
# ============================================================

def PrintMatrix(name, matrix):
    NoColumns = 6;

    # Compute no of blocks of rows to be printed .....

    if matrix.ndim == 1:
       noMatrixRows = matrix.shape[0]
       noMatrixCols = 1

    if matrix.ndim == 2:
       noMatrixRows = matrix.shape[0]
       noMatrixCols = matrix.shape[1]

    #  Compute number of blocks to be printed ...

    if noMatrixCols % NoColumns == 0:
       iNoBlocks = noMatrixCols/NoColumns;
    else:
       iNoBlocks = noMatrixCols/NoColumns + 1;

    #  Loop over the number of blocks ...

    for ib in range( int(iNoBlocks) ):
        iFirstColumn = ib*NoColumns + 1
        iLastColumn  = min ( (ib+1)*NoColumns, noMatrixCols )
```

```
        # Print title of matrix at top of each block ....

        print("Matrix: {:s} ".format(name) );

        # Label row and column nos */

        print("row/col     ", end="")
        colList = range(iFirstColumn, iLastColumn + 1)
        for col in [ *colList ]:
            print("       {:3d}    ".format(col),end="")
        print("")

        # Loop over rows and print matrix elements ....

        ii = 1
        for row in matrix:
            print(" {:3d}     ".format(ii),end="")
            colList = range( iFirstColumn, iLastColumn + 1)
            for col in [ *colList ]:
                if matrix.ndim == 1:
                   print(" {:12.5e} ".format( matrix[ii-1] ), end="")
                else:
                   print(" {:12.5e} ".format(matrix[ii-1][col-1]), end="")
            print("")
            ii = ii + 1
        print("")

# ================================================================
# Compute maximum and minumun of three numbers ...
# ================================================================

def maximum(a, b, c):
    list = [a, b, c]
    return max(list)

def minimum(a, b, c):
    list = [a, b, c]
    return min(list)

# ================================================================
# Print element forces ...
# ================================================================

def printElementForces(name, minF, maxF):
    if( minF < 0):
        print("---     Minimum {:s} = {:7.2f} (C) ... ".format( name, minF ) )
    else:
        print("---     Minimum {:s} = {:7.2f} (T) ... ".format( name, minF ) )

    if( maxF < 0):
        print("---     Maximum {:s} = {:7.2f} (C) ... ".format( name, maxF ) )
    else:
        print("---     Maximum {:s} = {:7.2f} (T) ... ".format( name, maxF ) )

# ================================================================
# main method ...
# ================================================================

def main():
    print("--- Enter TestTrussAnalysisCrane03.main()    ... ");
    print("--- ======================================== ... ");

    print("--- ");
    print("--- Part 1: Initialize coefficients for matrix equations ... ");

    nonodes  = 11;  # <--- no of nodes in crane structure ...
    maxterms =  4;  # <--- max no terms in an equation of equilibrium ...

    # Equilibrium and destination arrays ...

    equilibrium = np.zeros(( 2*nonodes, maxterms ))
    destination = np.zeros(( 2*nonodes, maxterms ))

    PrintMatrix("Equilibrium", equilibrium );
    PrintMatrix("Destination", destination );

    # Node A ...

    equilibrium[0][0] = 1   # < --- equilibrium in x direction ...
    equilibrium[0][1] = 1

    destination[0][0] =  8;
    destination[0][1] = 20;

    equilibrium[1][0] = 1   # < --- equilibrium in y direction ...
    equilibrium[1][1] = 1
```

```
destination[1][0] =  1;
destination[1][1] = 21;

# Node B ...

equilibrium[2][0] = 1    # < --- equilibrium in x direction ...
equilibrium[2][1] = 1/math.sqrt(2)

destination[2][0] = 8;
destination[2][1] = 9;

equilibrium[3][0] = 1    # < --- equilibrium in y direction ...
equilibrium[3][1] = 1/math.sqrt(2)
equilibrium[3][2] = 1

destination[3][0] =  2;
destination[3][1] =  9;
destination[3][2] = 22;

# Node C ...

equilibrium[4][0] = 1/math.sqrt(2) # <--- equilibrium in x direction ...
equilibrium[4][1] = 1
equilibrium[4][2] = 1/math.sqrt(2)

destination[4][0] =  9;
destination[4][1] = 10;
destination[4][2] = 11;

equilibrium[5][0] =  1              # <--- equilibrium in y direction ...
equilibrium[5][1] = -1
equilibrium[5][2] =  1/math.sqrt(2)
equilibrium[5][3] = -1/math.sqrt(2)

destination[5][0] =  1;
destination[5][1] =  3;
destination[5][2] =  9;
destination[5][3] = 11;

# Node D ...

equilibrium[6][0] =  1; # <--- equilibrium in x direction ...
destination[6][0] = 10;

equilibrium[7][0] =  1; # <--- equilibrium in y direction ...
equilibrium[7][1] = -1;

destination[7][0] = 2;
destination[7][1] = 4;

# Node E ...

equilibrium[8][0] = 1                 # <--- equilibrium in x direction ...
equilibrium[8][1] = 2/math.sqrt(5)

destination[8][0] = 12;
destination[8][1] = 16;

equilibrium[9][0] = -1/math.sqrt(5)   # <--- equilibrium in y direction ...
destination[9][0] = 16;

# Node F ...

equilibrium[10][0] =  1     # <--- equilibrium in x direction ...
equilibrium[10][1] = -1

destination[10][0] = 12;
destination[10][1] = 13;

equilibrium[11][0] = -1     # <--- equilibrium in y direction ...
destination[11][0] =  5;

# Node G ...

equilibrium[12][0] =  1     # <--- equilibrium in x direction ...
equilibrium[12][1] = -1
equilibrium[12][2] =  2/math.sqrt(5)

destination[12][0] = 13;
destination[12][1] = 14;
destination[12][2] = 18;

equilibrium[13][0] = -1;     # <--- equilibrium in y direction ...
equilibrium[13][1] =  1;
equilibrium[13][2] =  1/math.sqrt(5);
```

```
destination[13][0] =  3;
destination[13][1] =  6;
destination[13][2] = 18;

# Node H ...

equilibrium[14][0] =  1/math.sqrt(2);  # <--- equilibrium in x direction ...
equilibrium[14][1] =  1;
equilibrium[14][2] =  1/math.sqrt(2);

destination[14][0] = 11;
destination[14][1] = 14;
destination[14][2] = 19;

equilibrium[15][0] =   1               # <--- equilibrium in y direction ...
equilibrium[15][1] =   1/math.sqrt(2);
equilibrium[15][2] =  -1
equilibrium[15][3] =  -1/math.sqrt(2);

destination[15][0] =   4;
destination[15][1] =  11;
destination[15][2] =   7;
destination[15][3] =  19;

# Node I ...

equilibrium[16][0] =  2/math.sqrt(5);  # <--- equilibrium in x direction ...
equilibrium[16][1] = -2/math.sqrt(5);
equilibrium[16][2] = -2/math.sqrt(5);

destination[16][0] = 16;
destination[16][1] = 17;
destination[16][2] = 18;

equilibrium[17][0] =  1;               # <--- equilibrium in y direction ...
equilibrium[17][1] =  1/math.sqrt(5);
equilibrium[17][2] = -1/math.sqrt(5);
equilibrium[17][3] =  1/math.sqrt(5);

destination[17][0] =  5;
destination[17][1] = 16;
destination[17][2] = 17;
destination[17][3] = 18;

# Node J ...

equilibrium[18][0] =  1;               # <--- equilibrium in x direction ...
equilibrium[18][1] = -2/math.sqrt(5);
equilibrium[18][2] =  1/math.sqrt(2);

destination[18][0] = 15;
destination[18][1] = 17;
destination[18][2] = 19;

equilibrium[19][0] =  1;               # <--- equilibrium in y direction ...
equilibrium[19][1] =  1/math.sqrt(5);
equilibrium[19][2] =  1/math.sqrt(2);

destination[19][0] =  6;
destination[19][1] = 17;
destination[19][2] = 19;

# Node K ...

equilibrium[20][0] =  1;               # <--- equilibrium in x direction ...
destination[20][0] = 15;

equilibrium[21][0] =  1;               # <--- equilibrium in y direction ...
destination[21][0] =  7;

PrintMatrix("Equilibrium", equilibrium );
PrintMatrix("Destination", destination );

print("--- ");
print("--- Part 2: Systematically Assemble Matrix A ... ");
print("--- ");

# Allocate memory for A matrix ...

A = np.zeros(( 2*nonodes, 2*nonodes ))

# Systematically assemble A matrix from equilibrium and destination arrays ...

for row in range(2*nonodes):
    print("--- ");
    for col in range(maxterms):
        eitem = equilibrium[row][col];
```

```
        ditem = int( destination[row][col] );
        if eitem != 0 or ditem != 0:
            print("--- (row,col) --> {:2d}, {:d}: eitem --> {:9.5f}, ditem --> {:3d} ...".format( row, col, eitem, ditem ) );
            A[row][ditem-1] = eitem;

PrintMatrix("A", A);

print("--- ");
print("--- Part 2: Initialize 22x1 load vectors ... ");
print("--- ");

print("--- Load Case 1: Node e_y = -250 ...");

B1 = np.zeros(( 2*nonodes, 1 ))
B1[ 9][0] = -250.0;
B1[11][0] =    0.0;

PrintMatrix("Load Case 1:", B1);

print("--- Load Case 2: Node e_y = -200, node f_y = -200 ...");

B2 = np.zeros(( 2*nonodes, 1 ))
B2[ 9][0] = -200.0;
B2[11][0] = -200.0;

PrintMatrix("Load Case 2:", B2);

print("--- Load Case 3: Node f_y = -300 ...");

B3 = np.zeros(( 2*nonodes, 1 ))
B3[ 9][0] =    0.0;
B3[11][0] = -300.0;

PrintMatrix("Load Case 3:", B3);

print("--- ");
print("--- Part 4: Check properties of matrix A ... ");
print("--- ");

rank = matrix_rank(A)
det  = np.linalg.det(A)

print("--- Matrix A: rank = {:f}, det = {:f}  ...".format(rank, det) );

print("--- ");
print("--- Part 5: Solve A.F = B for three load cases ... ");
print("--- ");

F1 = np.linalg.solve(A, B1)
PrintMatrix("Load Case 1: Forces ...", F1);

F2 = np.linalg.solve(A, B2)
PrintMatrix("Load Case 2: Forces ...", F2);

F3 = np.linalg.solve(A, B3)
PrintMatrix("Load Case 3: Forces ...", F3);

print("--- ");
print("--- Part 6: Print support reactions and element-level forces ... ");

print("--- ");
print("--- Support Reactions and Element-Level Forces: Load Case 1: ...");
print("--- ---------------------------------------------------------- ...");
print("--- ");
print("---   Reaction A: R_ax = {:7.2f} ... ".format( F1[19][0] ) );
print("---              : R_ay = {:7.2f} ... ".format( F1[20][0] ) );
print("---   Reaction B: R_by = {:7.2f} ... ".format( F1[21][0] ) );
print("--- ");
print("--- Element Level Forces:");
print("--- ");
print("---   Element A-C: F1  = {:7.2f} ... ".format( F1[0][0] ) );
print("---   Element B-D: F2  = {:7.2f} ... ".format( F1[1][0] ) );
print("---   Element C-G: F3  = {:7.2f} ... ".format( F1[2][0] ) );
print("---   Element D-H: F4  = {:7.2f} ... ".format( F1[3][0] ) );
print("---   Element F-I: F5  = {:7.2f} ... ".format( F1[4][0] ) );
print("---   Element G-J: F6  = {:7.2f} ... ".format( F1[5][0] ) );
print("---   Element H-K: F7  = {:7.2f} ... ".format( F1[6][0] ) );
print("---   Element A-B: F8  = {:7.2f} ... ".format( F1[7][0] ) );
print("---   Element B-C: F09 = {:7.2f} ... ".format( F1[8][0] ) );
print("---   Element C-D: F10 = {:7.2f} ... ".format( F1[9][0] ) );
print("---   Element C-H: F11 = {:7.2f} ... ".format( F1[10][0] ) );
print("---   Element E-F: F12 = {:7.2f} ... ".format( F1[11][0] ) );
print("---   Element F-G: F13 = {:7.2f} ... ".format( F1[12][0] ) );
print("---   Element G-H: F14 = {:7.2f} ... ".format( F1[13][0] ) );
print("---   Element J-K: F15 = {:7.2f} ... ".format( F1[14][0] ) );
print("---   Element E-I: F16 = {:7.2f} ... ".format( F1[15][0] ) );
print("---   Element I-J: F17 = {:7.2f} ... ".format( F1[16][0] ) );
```

```
print("---   Element G-I: F18 = {:7.2f} ... ".format( F1[17][0] ) );
print("---   Element H-J: F19 = {:7.2f} ... ".format( F1[18][0] ) );

print("--- ");
print("--- Support Reactions and Element-Level Forces: Load Case 2: ...");
print("--- ------------------------------------------------------- ...");
print("--- ");
print("---   Reaction A: R_ax = {:7.2f} ... ".format( F2[19][0] ) );
print("---             : R_ay = {:7.2f} ... ".format( F2[20][0] ) );
print("---   Reaction B: R_by = {:7.2f} ... ".format( F2[21][0] ) );
print("--- ");
print("--- Element Level Forces:");
print("--- ");
print("---   Element A-C: F1  = {:7.2f} ... ".format( F2[0][0] ) );
print("---   Element B-D: F2  = {:7.2f} ... ".format( F2[1][0] ) );
print("---   Element C-G: F3  = {:7.2f} ... ".format( F2[2][0] ) );
print("---   Element D-H: F4  = {:7.2f} ... ".format( F2[3][0] ) );
print("---   Element F-I: F5  = {:7.2f} ... ".format( F2[4][0] ) );
print("---   Element G-J: F6  = {:7.2f} ... ".format( F2[5][0] ) );
print("---   Element H-K: F7  = {:7.2f} ... ".format( F2[6][0] ) );
print("---   Element A-B: F8  = {:7.2f} ... ".format( F2[7][0] ) );
print("---   Element B-C: F09 = {:7.2f} ... ".format( F2[8][0] ) );
print("---   Element C-D: F10 = {:7.2f} ... ".format( F2[9][0] ) );
print("---   Element C-H: F11 = {:7.2f} ... ".format( F2[10][0] ) );
print("---   Element E-F: F12 = {:7.2f} ... ".format( F2[11][0] ) );
print("---   Element F-G: F13 = {:7.2f} ... ".format( F2[12][0] ) );
print("---   Element G-H: F14 = {:7.2f} ... ".format( F2[13][0] ) );
print("---   Element J-K: F15 = {:7.2f} ... ".format( F2[14][0] ) );
print("---   Element E-I: F16 = {:7.2f} ... ".format( F2[15][0] ) );
print("---   Element I-J: F17 = {:7.2f} ... ".format( F2[16][0] ) );
print("---   Element G-I: F18 = {:7.2f} ... ".format( F2[17][0] ) );
print("---   Element H-J: F19 = {:7.2f} ... ".format( F2[18][0] ) );

print("--- ");
print("--- Support Reactions and Element-Level Forces: Load Case 3: ...");
print("--- ------------------------------------------------------- ...");
print("--- ");
print("---   Reaction A: R_ax = {:7.2f} ... ".format( F3[19][0] ) );
print("---             : R_ay = {:7.2f} ... ".format( F3[20][0] ) );
print("---   Reaction B: R_by = {:7.2f} ... ".format( F3[21][0] ) );
print("--- ");
print("--- Element Level Forces:");
print("--- ");
print("---   Element A-C: F1  = {:7.2f} ... ".format( F3[0][0] ) );
print("---   Element B-D: F2  = {:7.2f} ... ".format( F3[1][0] ) );
print("---   Element C-G: F3  = {:7.2f} ... ".format( F3[2][0] ) );
print("---   Element D-H: F4  = {:7.2f} ... ".format( F3[3][0] ) );
print("---   Element F-I: F5  = {:7.2f} ... ".format( F3[4][0] ) );
print("---   Element G-J: F6  = {:7.2f} ... ".format( F3[5][0] ) );
print("---   Element H-K: F7  = {:7.2f} ... ".format( F3[6][0] ) );
print("---   Element A-B: F8  = {:7.2f} ... ".format( F3[7][0] ) );
print("---   Element B-C: F09 = {:7.2f} ... ".format( F3[8][0] ) );
print("---   Element C-D: F10 = {:7.2f} ... ".format( F3[9][0] ) );
print("---   Element C-H: F11 = {:7.2f} ... ".format( F3[10][0] ) );
print("---   Element E-F: F12 = {:7.2f} ... ".format( F3[11][0] ) );
print("---   Element F-G: F13 = {:7.2f} ... ".format( F3[12][0] ) );
print("---   Element G-H: F14 = {:7.2f} ... ".format( F3[13][0] ) );
print("---   Element J-K: F15 = {:7.2f} ... ".format( F3[14][0] ) );
print("---   Element E-I: F16 = {:7.2f} ... ".format( F3[15][0] ) );
print("---   Element I-J: F17 = {:7.2f} ... ".format( F3[16][0] ) );
print("---   Element G-I: F18 = {:7.2f} ... ".format( F3[17][0] ) );
print("---   Element H-J: F19 = {:7.2f} ... ".format( F3[18][0] ) );

print("--- ");
print("--- Summary of Max/Min Reactions and Element-Level Forces ...");
print("--- ------------------------------------------------------ ...");

MinRax = minimum( F1[19][0], F2[19][0], F3[19][0] )
MaxRax = maximum( F1[19][0], F2[19][0], F3[19][0] )

MinRay = minimum( F1[20][0], F2[20][0], F3[20][0] )
MaxRay = maximum( F1[20][0], F2[20][0], F3[20][0] )

MinRby = minimum( F1[21][0], F2[21][0], F3[21][0] )
MaxRby = maximum( F1[21][0], F2[21][0], F3[21][0] )

print("--- ");
print("---   Reaction A: Minimum R_ax = {:7.2f}, Maximum R_ax = {:7.2f} ... ".format( MinRax, MaxRax) )
print("---             : Minimum R_ay = {:7.2f}, Maximum R_ay = {:7.2f} ... ".format( MinRay, MaxRay) )
print("--- ");
print("---   Reaction B: Minimum R_by = {:7.2f}, Maximum R_by = {:7.2f} ... ".format( MinRby, MaxRby) )

MinF1 = minimum( F1[0][0], F2[0][0], F3[0][0] )
MaxF1 = maximum( F1[0][0], F2[0][0], F3[0][0] )

MinF2 = minimum( F1[1][0], F2[1][0], F3[1][0] )
MaxF2 = maximum( F1[1][0], F2[1][0], F3[1][0] )
```

```
MinF3 = minimum( F1[2][0], F2[2][0], F3[2][0] )
MaxF3 = maximum( F1[2][0], F2[2][0], F3[2][0] )

MinF4 = minimum( F1[3][0], F2[3][0], F3[3][0] )
MaxF4 = maximum( F1[3][0], F2[3][0], F3[3][0] )

MinF5 = minimum( F1[4][0], F2[4][0], F3[4][0] )
MaxF5 = maximum( F1[4][0], F2[4][0], F3[4][0] )

MinF6 = minimum( F1[5][0], F2[5][0], F3[5][0] )
MaxF6 = maximum( F1[5][0], F2[5][0], F3[5][0] )

MinF7 = minimum( F1[6][0], F2[6][0], F3[6][0] )
MaxF7 = maximum( F1[6][0], F2[6][0], F3[6][0] )

MinF8 = minimum( F1[7][0], F2[7][0], F3[7][0] )
MaxF8 = maximum( F1[7][0], F2[7][0], F3[7][0] )

MinF9 = minimum( F1[8][0], F2[8][0], F3[8][0] )
MaxF9 = maximum( F1[8][0], F2[8][0], F3[8][0] )

MinF10 = minimum( F1[9][0], F2[9][0], F3[9][0] )
MaxF10 = maximum( F1[9][0], F2[9][0], F3[9][0] )

MinF11 = minimum( F1[10][0], F2[10][0], F3[10][0] )
MaxF11 = maximum( F1[10][0], F2[10][0], F3[10][0] )

MinF12 = minimum( F1[11][0], F2[11][0], F3[11][0] )
MaxF12 = maximum( F1[11][0], F2[11][0], F3[11][0] )

MinF13 = minimum( F1[12][0], F2[12][0], F3[12][0] )
MaxF13 = maximum( F1[12][0], F2[12][0], F3[12][0] )

MinF14 = minimum( F1[13][0], F2[13][0], F3[13][0] )
MaxF14 = maximum( F1[13][0], F2[13][0], F3[13][0] )

MinF15 = minimum( F1[14][0], F2[14][0], F3[14][0] )
MaxF15 = maximum( F1[14][0], F2[14][0], F3[14][0] )

MinF16 = minimum( F1[15][0], F2[15][0], F3[15][0] )
MaxF16 = maximum( F1[15][0], F2[15][0], F3[15][0] )

MinF17 = minimum( F1[16][0], F2[16][0], F3[16][0] )
MaxF17 = maximum( F1[16][0], F2[16][0], F3[16][0] )

MinF18 = minimum( F1[17][0], F2[17][0], F3[17][0] )
MaxF18 = maximum( F1[17][0], F2[17][0], F3[17][0] )

MinF19 = minimum( F1[18][0], F2[18][0], F3[18][0] )
MaxF19 = maximum( F1[18][0], F2[18][0], F3[18][0] )

print("--- ");
print("---   Element A-C: ")
printElementForces( "F1", MinF1, MaxF1)

print("---   Element B-D: ")
printElementForces( "F2", MinF2, MaxF2)

print("---   Element C-G: ")
printElementForces( "F3", MinF3, MaxF3)

print("---   Element D-H: ")
printElementForces( "F4", MinF4, MaxF4)

print("---   Element F-I: ")
printElementForces( "F5", MinF5, MaxF5)

print("---   Element G-J: ")
printElementForces( "F6", MinF6, MaxF6)

print("---   Element H-K: ")
printElementForces( "F7", MinF7, MaxF7)

print("---   Element A-B: ")
printElementForces( "F8", MinF8, MaxF8)

print("---   Element B-C: ")
printElementForces( "F9", MinF9, MaxF9)

print("---   Element C-D: ")
printElementForces( "F10", MinF10, MaxF10)

print("---   Element C-H: ")
printElementForces( "F11", MinF11, MaxF11)

print("---   Element E-F: ")
```

```
    printElementForces( "F12", MinF12, MaxF12)

    print("---   Element F-G: ")
    printElementForces( "F13", MinF13, MaxF13)

    print("---   Element G-H: ")
    printElementForces( "F14", MinF14, MaxF14)

    print("---   Element J-K: ")
    printElementForces( "F15", MinF15, MaxF15)

    print("---   Element E-I: ")
    printElementForces( "F16", MinF16, MaxF16)

    print("---   Element I-J: ")
    printElementForces( "F17", MinF17, MaxF17)

    print("---   Element G-I: ")
    printElementForces( "F18", MinF18, MaxF18)

    print("---   Element H-J: ")
    printElementForces( "F19", MinF19, MaxF19)

    print("--- ====================================== ... ");
    print("--- Leave TestTrussAnalysisCrane03.main()    ... ");

# call the main method ...

main()
```

Abbreviated Output:

```
--- Enter TestTrussAnalysisCrane03.main()    ...
--- ====================================== ...
---
--- Part 1: Initialize coefficients for matrix equations ...
Matrix: Equilibrium
row/col              1             2             3             4
   1       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   2       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   3       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   4       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   5       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   6       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   7       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   8       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   9       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  10       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  11       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  12       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  13       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  14       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  15       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  16       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  17       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  18       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  19       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  20       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  21       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  22       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00

Matrix: Destination
row/col              1             2             3             4
   1       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   2       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   3       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   4       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   5       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   6       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   7       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   8       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   9       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  10       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  11       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  12       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  13       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  14       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  15       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  16       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  17       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
```

```
  18       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  19       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  20       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  21       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  22       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00

Matrix: Equilibrium
row/col              1             2             3             4
   1       1.00000e+00   1.00000e+00   0.00000e+00   0.00000e+00
   2       1.00000e+00   1.00000e+00   0.00000e+00   0.00000e+00
   3       1.00000e+00   7.07107e-01   0.00000e+00   0.00000e+00
   4       1.00000e+00   7.07107e-01   1.00000e+00   0.00000e+00
   5       7.07107e-01   1.00000e+00   7.07107e-01   0.00000e+00
   6       1.00000e+00  -1.00000e+00   7.07107e-01  -7.07107e-01
   7       1.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   8       1.00000e+00  -1.00000e+00   0.00000e+00   0.00000e+00
   9       1.00000e+00   8.94427e-01   0.00000e+00   0.00000e+00
  10      -4.47214e-01   0.00000e+00   0.00000e+00   0.00000e+00
  11       1.00000e+00  -1.00000e+00   0.00000e+00   0.00000e+00
  12      -1.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  13       1.00000e+00  -1.00000e+00   8.94427e-01   0.00000e+00
  14      -1.00000e+00   1.00000e+00   4.47214e-01   0.00000e+00
  15       7.07107e-01   1.00000e+00   7.07107e-01   0.00000e+00
  16       1.00000e+00   7.07107e-01  -1.00000e+00  -7.07107e-01
  17       8.94427e-01  -8.94427e-01  -8.94427e-01   0.00000e+00
  18       1.00000e+00   4.47214e-01  -4.47214e-01   4.47214e-01
  19       1.00000e+00  -8.94427e-01   7.07107e-01   0.00000e+00
  20       1.00000e+00   4.47214e-01   7.07107e-01   0.00000e+00
  21       1.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  22       1.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00

Matrix: Destination
row/col              1             2             3             4
   1       8.00000e+00   2.00000e+01   0.00000e+00   0.00000e+00
   2       1.00000e+00   2.10000e+01   0.00000e+00   0.00000e+00
   3       8.00000e+00   9.00000e+00   0.00000e+00   0.00000e+00
   4       2.00000e+00   9.00000e+00   2.20000e+01   0.00000e+00
   5       9.00000e+00   1.00000e+01   1.10000e+01   0.00000e+00
   6       1.00000e+00   3.00000e+00   9.00000e+00   1.10000e+01
   7       1.00000e+01   0.00000e+00   0.00000e+00   0.00000e+00
   8       2.00000e+00   4.00000e+00   0.00000e+00   0.00000e+00
   9       1.20000e+01   1.60000e+01   0.00000e+00   0.00000e+00
  10       1.60000e+01   0.00000e+00   0.00000e+00   0.00000e+00
  11       1.20000e+01   1.30000e+01   0.00000e+00   0.00000e+00
  12       5.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  13       1.30000e+01   1.40000e+01   1.80000e+01   0.00000e+00
  14       3.00000e+00   6.00000e+00   1.80000e+01   0.00000e+00
  15       1.10000e+01   1.40000e+01   1.90000e+01   0.00000e+00
  16       4.00000e+00   1.10000e+01   7.00000e+00   1.90000e+01
  17       1.60000e+01   1.70000e+01   1.80000e+01   0.00000e+00
  18       5.00000e+00   1.60000e+01   1.70000e+01   1.80000e+01
  19       1.50000e+01   1.70000e+01   1.90000e+01   0.00000e+00
  20       6.00000e+00   1.70000e+01   1.90000e+01   0.00000e+00
  21       1.50000e+01   0.00000e+00   0.00000e+00   0.00000e+00
  22       7.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00

---
--- Part 2: Systematically Assemble Matrix A ...
---
---
--- (row,col) -->  0, 0: eitem -->   1.00000, ditem -->   8 ...
--- (row,col) -->  0, 1: eitem -->   1.00000, ditem -->  20 ...

... lines of output removed ...

--- (row,col) --> 19, 0: eitem -->   1.00000, ditem -->   6 ...
--- (row,col) --> 19, 1: eitem -->   0.44721, ditem -->  17 ...
--- (row,col) --> 19, 2: eitem -->   0.70711, ditem -->  19 ...
---
--- (row,col) --> 20, 0: eitem -->   1.00000, ditem -->  15 ...
---
--- (row,col) --> 21, 0: eitem -->   1.00000, ditem -->   7 ...

Matrix: A
row/col              1             2             3             4             5             6
   1       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   2       1.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   3       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   4       0.00000e+00   1.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   5       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   6       1.00000e+00   0.00000e+00  -1.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   7       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
   8       0.00000e+00   1.00000e+00   0.00000e+00  -1.00000e+00   0.00000e+00   0.00000e+00
   9       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  10       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  11       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00
  12       0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00  -1.00000e+00   0.00000e+00
```

13	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
14	0.00000e+00	0.00000e+00	-1.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00
15	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
16	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00	0.00000e+00	0.00000e+00
17	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
18	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00	0.00000e+00
19	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
20	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00
21	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
22	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00

Matrix: A

row/col	7	8	9	10	11	12
1	0.00000e+00	1.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
2	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
3	0.00000e+00	1.00000e+00	7.07107e-01	0.00000e+00	0.00000e+00	0.00000e+00
4	0.00000e+00	0.00000e+00	7.07107e-01	0.00000e+00	0.00000e+00	0.00000e+00
5	0.00000e+00	0.00000e+00	7.07107e-01	1.00000e+00	7.07107e-01	0.00000e+00
6	0.00000e+00	0.00000e+00	7.07107e-01	0.00000e+00	-7.07107e-01	0.00000e+00
7	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00	0.00000e+00	0.00000e+00
8	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
9	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00
10	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
11	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00
12	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
13	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
14	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
15	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	7.07107e-01	0.00000e+00
16	-1.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	7.07107e-01	0.00000e+00
17	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
18	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
19	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
20	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
21	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
22	1.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00

Matrix: A

row/col	13	14	15	16	17	18
1	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
2	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
3	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
4	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
5	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
6	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
7	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
8	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
9	0.00000e+00	0.00000e+00	0.00000e+00	8.94427e-01	0.00000e+00	0.00000e+00
10	0.00000e+00	0.00000e+00	0.00000e+00	-4.47214e-01	0.00000e+00	0.00000e+00
11	-1.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
12	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
13	1.00000e+00	-1.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	8.94427e-01
14	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	4.47214e-01
15	0.00000e+00	1.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
16	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
17	0.00000e+00	0.00000e+00	0.00000e+00	8.94427e-01	-8.94427e-01	-8.94427e-01
18	0.00000e+00	0.00000e+00	0.00000e+00	4.47214e-01	-4.47214e-01	4.47214e-01
19	0.00000e+00	0.00000e+00	1.00000e+00	0.00000e+00	-8.94427e-01	0.00000e+00
20	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	4.47214e-01	0.00000e+00
21	0.00000e+00	0.00000e+00	1.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
22	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00

Matrix: A

row/col	19	20	21	22
1	0.00000e+00	1.00000e+00	0.00000e+00	0.00000e+00
2	0.00000e+00	0.00000e+00	1.00000e+00	0.00000e+00
3	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
4	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00
5	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
6	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
7	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
8	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
9	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
10	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
11	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
12	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
13	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
14	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
15	7.07107e-01	0.00000e+00	0.00000e+00	0.00000e+00
16	-7.07107e-01	0.00000e+00	0.00000e+00	0.00000e+00
17	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
18	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
19	7.07107e-01	0.00000e+00	0.00000e+00	0.00000e+00
20	7.07107e-01	0.00000e+00	0.00000e+00	0.00000e+00
21	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
22	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00

```
--- Part 2: Initialize 22x1 load vectors ...
---
--- Load Case 1: Node e_y = -250 ...

Matrix: Load Case 1:
row/col               1
   1       0.00000e+00
   2       0.00000e+00
   3       0.00000e+00
   4       0.00000e+00
   5       0.00000e+00
   6       0.00000e+00
   7       0.00000e+00
   8       0.00000e+00
   9       0.00000e+00
  10      -2.50000e+02
  11       0.00000e+00
  12       0.00000e+00
  13       0.00000e+00
  14       0.00000e+00
  15       0.00000e+00
  16       0.00000e+00
  17       0.00000e+00
  18       0.00000e+00
  19       0.00000e+00
  20       0.00000e+00
  21       0.00000e+00
  22       0.00000e+00

--- Load Case 2: Node e_y = -200, node f_y = -200 ...

Matrix: Load Case 2:
row/col               1
   1       0.00000e+00
   2       0.00000e+00
   3       0.00000e+00
   4       0.00000e+00
   5       0.00000e+00
   6       0.00000e+00
   7       0.00000e+00
   8       0.00000e+00
   9       0.00000e+00
  10      -2.00000e+02
  11       0.00000e+00
  12      -2.00000e+02
  13       0.00000e+00
  14       0.00000e+00
  15       0.00000e+00
  16       0.00000e+00
  17       0.00000e+00
  18       0.00000e+00
  19       0.00000e+00
  20       0.00000e+00
  21       0.00000e+00
  22       0.00000e+00

--- Load Case 3: Node f_y = -300 ...

Matrix: Load Case 3:
row/col               1
   1       0.00000e+00
   2       0.00000e+00
   3       0.00000e+00
   4       0.00000e+00
   5       0.00000e+00
   6       0.00000e+00
   7       0.00000e+00
   8       0.00000e+00
   9       0.00000e+00
  10       0.00000e+00
  11       0.00000e+00
  12      -3.00000e+02
  13       0.00000e+00
  14       0.00000e+00
  15       0.00000e+00
  16       0.00000e+00
  17       0.00000e+00
  18       0.00000e+00
  19       0.00000e+00
  20       0.00000e+00
  21       0.00000e+00
  22       0.00000e+00

---
--- Part 4: Check properties of matrix A ...
---
--- Matrix A: rank = 22.000000, det = -0.126491  ...
```

```
---
--- Part 5: Solve A.F = B for three load cases ...
---

Matrix: Load Case 1: Forces ...
row/col              1
   1      -7.50000e+02
   2       5.00000e+02
   3      -7.50000e+02
   4       5.00000e+02
   5      -0.00000e+00
   6      -7.50000e+02
   7       3.97252e-14
   8       1.38778e-14
   9      -1.96262e-14
  10       1.38778e-14
  11       0.00000e+00
  12      -5.00000e+02
  13      -5.00000e+02
  14      -5.00000e+02
  15       4.99302e-15
  16       5.59017e+02
  17       5.59017e+02
  18       0.00000e+00
  19       7.07107e+02
  20      -1.38778e-14
  21       7.50000e+02
  22      -5.00000e+02

Matrix: Load Case 2: Forces ...
row/col              1
   1      -1.00000e+03
   2       6.00000e+02
   3      -1.00000e+03
   4       6.00000e+02
   5       2.00000e+02
   6      -9.00000e+02
   7       3.05729e-14
   8      -4.44089e-16
   9       6.28037e-16
  10      -4.44089e-16
  11       0.00000e+00
  12      -4.00000e+02
  13      -4.00000e+02
  14      -6.00000e+02
  15       2.63285e-14
  16       4.47214e+02
  17       6.70820e+02
  18      -2.23607e+02
  19       8.48528e+02
  20       4.44089e-16
  21       1.00000e+03
  22      -6.00000e+02

Matrix: Load Case 3: Forces ...
row/col              1
   1      -6.00000e+02
   2       3.00000e+02
   3      -6.00000e+02
   4       3.00000e+02
   5       3.00000e+02
   6      -4.50000e+02
   7       2.66107e-14
   8       1.11022e-14
   9      -1.57009e-14
  10       1.11022e-14
  11       0.00000e+00
  12       1.31643e-14
  13       0.00000e+00
  14      -3.00000e+02
  15       1.31643e-14
  16      -1.47181e-14
  17       3.35410e+02
  18      -3.35410e+02
  19       4.24264e+02
  20      -1.11022e-14
  21       6.00000e+02
  22      -3.00000e+02

---
--- Part 6: Print support reactions and element-level forces ...
---
--- Support Reactions and Element-Level Forces: Load Case 1: ...
--- ------------------------------------------------------- ...
---
---   Reaction A: R_ax =   -0.00 ...
---             : R_ay =  750.00 ...
```

```
---   Reaction B: R_by = -500.00 ...
---
--- Element Level Forces:
---
---   Element A-C: F1  = -750.00 ...
---   Element B-D: F2  =  500.00 ...
---   Element C-G: F3  = -750.00 ...
---   Element D-H: F4  =  500.00 ...
---   Element F-I: F5  =   -0.00 ...
---   Element G-J: F6  = -750.00 ...
---   Element H-K: F7  =    0.00 ...
---   Element A-B: F8  =    0.00 ...
---   Element B-C: F09 =   -0.00 ...
---   Element C-D: F10 =    0.00 ...
---   Element C-H: F11 =    0.00 ...
---   Element E-F: F12 = -500.00 ...
---   Element F-G: F13 = -500.00 ...
---   Element G-H: F14 = -500.00 ...
---   Element J-K: F15 =    0.00 ...
---   Element E-I: F16 =  559.02 ...
---   Element I-J: F17 =  559.02 ...
---   Element G-I: F18 =    0.00 ...
---   Element H-J: F19 =  707.11 ...
---
--- Support Reactions and Element-Level Forces: Load Case 2: ...
--- ------------------------------------------------------ ...
---
---   Reaction A: R_ax =    0.00 ...
---             : R_ay = 1000.00 ...
---   Reaction B: R_by = -600.00 ...
---
--- Element Level Forces:
---
---   Element A-C: F1  = -1000.00 ...
---   Element B-D: F2  =  600.00 ...
---   Element C-G: F3  = -1000.00 ...
---   Element D-H: F4  =  600.00 ...
---   Element F-I: F5  =  200.00 ...
---   Element G-J: F6  = -900.00 ...
---   Element H-K: F7  =    0.00 ...
---   Element A-B: F8  =   -0.00 ...
---   Element B-C: F09 =    0.00 ...
---   Element C-D: F10 =   -0.00 ...
---   Element C-H: F11 =    0.00 ...
---   Element E-F: F12 = -400.00 ...
---   Element F-G: F13 = -400.00 ...
---   Element G-H: F14 = -600.00 ...
---   Element J-K: F15 =    0.00 ...
---   Element E-I: F16 =  447.21 ...
---   Element I-J: F17 =  670.82 ...
---   Element G-I: F18 = -223.61 ...
---   Element H-J: F19 =  848.53 ...
---
--- Support Reactions and Element-Level Forces: Load Case 3: ...
--- ------------------------------------------------------ ...
---
---   Reaction A: R_ax =   -0.00 ...
---             : R_ay =  600.00 ...
---   Reaction B: R_by = -300.00 ...
---
--- Element Level Forces:
---
---   Element A-C: F1  = -600.00 ...
---   Element B-D: F2  =  300.00 ...
---   Element C-G: F3  = -600.00 ...
---   Element D-H: F4  =  300.00 ...
---   Element F-I: F5  =  300.00 ...
---   Element G-J: F6  = -450.00 ...
---   Element H-K: F7  =    0.00 ...
---   Element A-B: F8  =    0.00 ...
---   Element B-C: F09 =   -0.00 ...
---   Element C-D: F10 =    0.00 ...
---   Element C-H: F11 =    0.00 ...
---   Element E-F: F12 =    0.00 ...
---   Element F-G: F13 =    0.00 ...
---   Element G-H: F14 = -300.00 ...
---   Element J-K: F15 =    0.00 ...
---   Element E-I: F16 =   -0.00 ...
---   Element I-J: F17 =  335.41 ...
---   Element G-I: F18 = -335.41 ...
---   Element H-J: F19 =  424.26 ...
---
--- Summary of Max/Min Reactions and Element-Level Forces ...
--- ------------------------------------------------------ ...
---
---   Reaction A: Minimum R_ax =   -0.00, Maximum R_ax =    0.00 ...
---             : Minimum R_ay =  600.00, Maximum R_ay = 1000.00 ...
```

```
---
---   Reaction B: Minimum R_by = -600.00, Maximum R_by = -300.00 ...
---
---   Element A-C:
---     Minimum F1 = -1000.00 (C) ...
---     Maximum F1 = -600.00 (C) ...
---   Element B-D:
---     Minimum F2 =  300.00 (T) ...
---     Maximum F2 =  600.00 (T) ...
---   Element C-G:
---     Minimum F3 = -1000.00 (C) ...
---     Maximum F3 = -600.00 (C) ...
---   Element D-H:
---     Minimum F4 =  300.00 (T) ...
---     Maximum F4 =  600.00 (T) ...
---   Element F-I:
---     Minimum F5 =   -0.00 (T) ...
---     Maximum F5 =  300.00 (T) ...
---   Element G-J:
---     Minimum F6 = -900.00 (C) ...
---     Maximum F6 = -450.00 (C) ...
---   Element H-K:
---     Minimum F7 =    0.00 (T) ...
---     Maximum F7 =    0.00 (T) ...
---   Element A-B:
---     Minimum F8 =   -0.00 (C) ...
---     Maximum F8 =    0.00 (T) ...
---   Element B-C:
---     Minimum F9 =   -0.00 (C) ...
---     Maximum F9 =    0.00 (T) ...
---   Element C-D:
---     Minimum F10 =   -0.00 (C) ...
---     Maximum F10 =    0.00 (T) ...
---   Element C-H:
---     Minimum F11 =    0.00 (T) ...
---     Maximum F11 =    0.00 (T) ...
---   Element E-F:
---     Minimum F12 = -500.00 (C) ...
---     Maximum F12 =    0.00 (T) ...
---   Element F-G:
---     Minimum F13 = -500.00 (C) ...
---     Maximum F13 =    0.00 (T) ...
---   Element G-H:
---     Minimum F14 = -600.00 (C) ...
---     Maximum F14 = -300.00 (C) ...
---   Element J-K:
---     Minimum F15 =    0.00 (T) ...
---     Maximum F15 =    0.00 (T) ...
---   Element E-I:
---     Minimum F16 =   -0.00 (C) ...
---     Maximum F16 =  559.02 (T) ...
---   Element I-J:
---     Minimum F17 =  335.41 (T) ...
---     Maximum F17 =  670.82 (T) ...
---   Element G-I:
---     Minimum F18 = -335.41 (C) ...
---     Maximum F18 =    0.00 (T) ...
---   Element H-J:
---     Minimum F19 =  424.26 (T) ...
---     Maximum F19 =  848.53 (T) ...

--- ====================================== ...
--- Leave TestTrussAnalysisCrane03.main()    ...
```