```
clear
clc

% Constants
L_A = sqrt(1.5^2 + 1.5^2);
L_B = 1.5;
L_C = 1.5;
L_D = 1.5;
L_E = sqrt(1.5^2 + 1.5^2);
theta_A = atand(1.5/1.5);
theta_B = atand(0/1.5);
theta_C = atand(1.5/0);
theta_D = atand(0/1.5);
theta_E = atand(-1.5/1.5);

diam = 0.050;
CsA = pi * diam^2 / 4; % Cross sectional area
E = 210e9;

F_2x = 2000;
F_2y = -3000;
F_3x = 0;
F_3y = 0;
F_4x = 0;

K_local = [ 1 0 -1 0; 0 0 0 0; -1 0 1 0; 0 0 0 0];

% Calc k in global cords for each memeber
[T_A, K_A] = kglobal(theta_A, L_A, CsA, E);
[T_B, K_B] = kglobal(theta_B, L_B, CsA, E);
[T_C, K_C] = kglobal(theta_C, L_C, CsA, E);
[T_D, K_D] = kglobal(theta_D, L_D, CsA, E);
[T_E, K_E] = kglobal(theta_E, L_E, CsA, E);

%Assemble the Global K matrix
K_global = [K_A(1,1)+K_B(1,1), K_A(1,2)+K_B(1,2), K_A(1,3), K_A(1,4),
K_B(1,3), K_B(1,4), 0, 0;
            K_A(2,1)+K_B(2,1), K_A(2,2)+K_B(2,2), K_A(2,3), K_A(2,4),
K_B(2,3), K_B(2,4), 0, 0;
            K_A(3,1), K_A(3,2), K_A(3,3)+K_C(1,1)+K_E(1,1),
K_A(3,4)+K_C(1,2)+K_E(1,2), K_C(1,3), K_C(1,4), K_E(1,3), K_E(1,4);
            K_A(4,1), K_A(4,2), K_A(4,3)+K_C(2,1)+K_E(2,1),
K_A(4,4)+K_C(2,2)+K_E(2,2), K_C(2,3), K_C(2,4), K_E(2,3), K_E(2,4);
            K_B(3,1), K_B(3,2), K_C(3,1), K_C(3,2),
K_B(3,3)+K_C(3,3)+K_D(1,1), K_B(3,4)+K_C(3,4)+K_D(1,2), K_D(1,3), K_D(1,4);
            K_B(4,1), K_B(4,2), K_C(4,1), K_C(4,2),
K_B(4,3)+K_C(4,3)+K_D(2,1), K_B(4,4)+K_C(4,4)+K_D(2,2), K_D(2,3), K_D(2,4);
            0, 0, K_E(3,1), K_E(3,2), K_D(3,1), K_D(3,2), K_D(3,3)+K_E(3,3),
K_D(3,4)+K_E(3,4);
            0, 0, K_E(4,1), K_E(4,2), K_D(4,1), K_D(4,2), K_D(4,3)+K_E(4,3),
K_D(4,4)+K_E(4,4);];
K_check = sum(K_global)
```

```
%Recuded system of equations based on boundary conditions
F_bndry = [F_2x;F_2y;F_3x;F_3y;F_4x];
K_bndry = K_global(3:7,3:7);

%Solve for unknown displacments
xySolve_1 = K_bndry\F_bndry;

%Construct full displacment vector in global cords
xySolve_2 =
[0;0;xySolve_1(1);xySolve_1(2);xySolve_1(3);xySolve_1(4);xySolve_1(5);0;]

%Calculate reaction forces
F_react = K_global * xySolve_2

%Find local displacments for each element
%local x_local=transfor*X_gloabal
X_local_A = T_A*[xySolve_2(1:4)];
X_local_B = T_B*[xySolve_2(1:2);xySolve_2(5:6)];
X_local_C = T_C*[xySolve_2(3:6)];
X_local_D = T_D*[xySolve_2(5:8)];
X_local_E = T_E*[xySolve_2(3:4);xySolve_2(7:8)];

%Calculate axial force
F_axial_A = (E*CsA/L_A)*K_local*X_local_A;
F_axial_B = (E*CsA/L_B)*K_local*X_local_B;
F_axial_C = (E*CsA/L_C)*K_local*X_local_C;
F_axial_D = (E*CsA/L_D)*K_local*X_local_D;
F_axial_E = (E*CsA/L_E)*K_local*X_local_E;

%Calculate stress
stress_A = F_axial_A/CsA
stress_B = F_axial_B/CsA
stress_C = F_axial_C/CsA
stress_D = F_axial_D/CsA
stress_E = F_axial_E/CsA


K_check =

   1.0e-07 *

  Columns 1 through 7

         0         0         0         0         0         0    0.1490

  Column 8

         0


xySolve_2 =

   1.0e-04 *
```

```
         0
         0
    0.1938
   -0.2453
    0.0909
   -0.2453
    0.1819
         0


F_react =

   1.0e+03 *

   -2.0000
    0.5000
    2.0000
   -3.0000
         0
    0.0000
    0.0000
    2.5000


stress_A =

   1.0e+05 *

    3.6013
         0
   -3.6013
         0


stress_B =

   1.0e+06 *

   -1.2732
         0
    1.2732
         0


stress_C =

   1.0e-09 *

   -0.4632
         0
    0.4632
         0
```

```
stress_D =

   1.0e+06 *

   -1.2732
         0
    1.2732
         0


stress_E =

   1.0e+06 *

    1.8006
         0
   -1.8006
         0
```

Published with MATLAB® R2023b