SOLUTION

CE 325 Spring 2026 HW#6

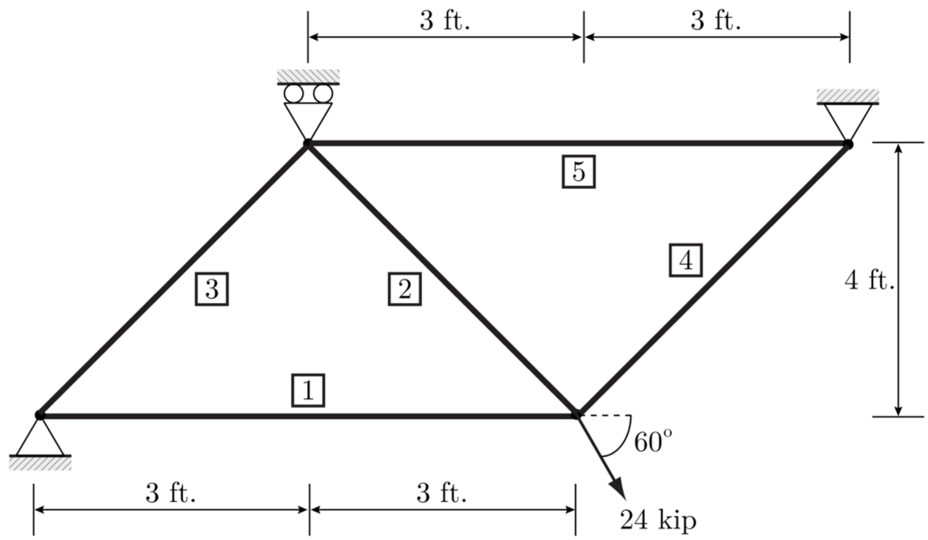
Due Thursday, March 26, at the beginning of class

1. For the 2D truss structure shown below determine joint displacements, axial bar forces and stresses, and support reactions using the Matrix Displacement Method:

   a. (10 pts) By hand

   b. (10 pts) Using Python

   c. (10 pts) Using SAP2000



E = 10,000 ksi ; A = 2.0 in.^{2} for all members

Problem 1

Executive Summary

Problem Statement

For the 2D truss structure, determine joint displacements, axial bar forces and stresses, and reactions using the Matrix Displacement Method:

a. Using Python
b. Using SAP2000

Results

$$\{d\} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 0.0289 \\ -0.0579 \\ 0.0246 \end{Bmatrix} in$$

$$Q1\ bar\ force = 8.02\ kips\ (T)$$
$$Q2\ bar\ force = 16.3\ kips\ (T)$$
$$Q3\ bar\ force = 4.92\ kips\ (T)$$
$$Q4\ bar\ force = 9.67\ kips\ (T)$$
$$Q5\ bar\ force = 6.83\ kips\ (C)$$

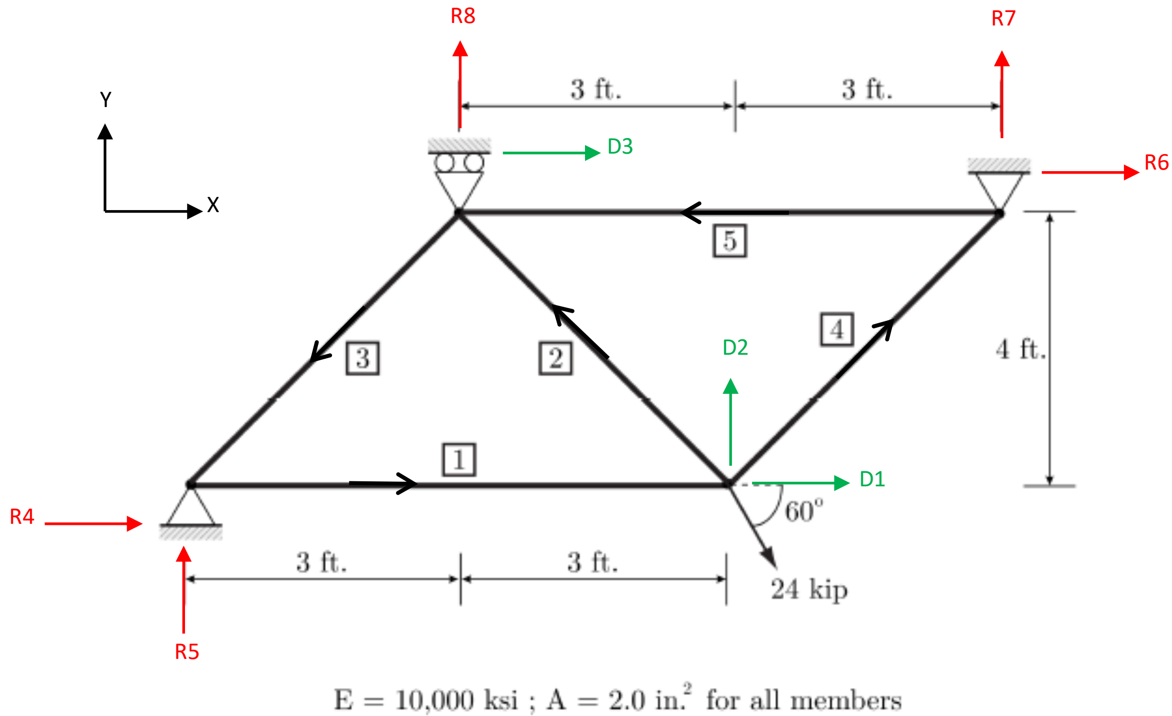
$$Q1\ bar\ stress = 4.01\ ksi\ (T)$$
$$Q2\ bar\ stress = 8.15\ ksi\ (T)$$
$$Q3\ bar\ stress = 2.46\ ksi\ (T)$$
$$Q4\ bar\ stress = 4.84\ ksi\ (T)$$
$$Q5\ bar\ stress = 3.42\ ksi\ (C)$$

$$R4 = -11.0\ kips$$
$$R5 = -3.94\ kips$$
$$R6 = -1.03\ kips$$
$$R7 = 7.74\ kips$$
$$R8 = 17.0\ kips$$

Problem 1

Technical Summary

1. Label dofs and reactions



2. Create force vector {P}

$$P = \begin{Bmatrix} P_x \\ P_y \\ 0 \end{Bmatrix} = \begin{Bmatrix} 24\cos(60) \\ -24\sin(60) \\ 0 \end{Bmatrix} = \begin{Bmatrix} 12 \\ -20.78 \\ 0 \end{Bmatrix} kips$$

3. Code Numbers

Member #	Code Number			
Member 1	4	5	1	2
Member 2	1	2	3	8
Member 3	3	8	4	5
Member 4	1	2	6	7
Member 5	6	7	3	8

4. Determine local [k] =

$$[k] = \frac{EA}{L}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Mem1: \left(\frac{EA}{L}\right)^1 = \frac{(10{,}000)(2)}{(72)} = 277.78\ \frac{kip}{in} \quad c = 1.0 \quad s = 0.0$$

$$Mem2: \left(\frac{EA}{L}\right)^2 = \frac{(10{,}000)(2)}{(60)} = 333.33\ \frac{kip}{in} \quad c = -0.6 \quad s = 0.8$$

$$Mem3: \left(\frac{EA}{L}\right)^3 = \frac{(10{,}000)(2)}{(60)} = 333.33\ \frac{kip}{in} \quad c = -0.6 \quad s = -0.8$$

$$Mem4: \left(\frac{EA}{L}\right)^4 = \frac{(10{,}000)(2)}{(60)} = 333.33\ \frac{kip}{in} \quad c = 0.6 \quad s = 0.8$$

$$Mem5: \left(\frac{EA}{L}\right)^5 = \frac{(10{,}000)(2)}{(72)} = 277.78\ \frac{kip}{in} \quad c = -1.0 \quad s = 0.0$$

5. Compute transformation matrix [T]

$$[T] = \begin{bmatrix} C & S & 0 & 0 \\ -S & C & 0 & 0 \\ 0 & 0 & C & S \\ 0 & 0 & -S & C \end{bmatrix}$$

$$[T]^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[T]^2 = \begin{bmatrix} -0.6 & 0.8 & 0 & 0 \\ -0.8 & -0.6 & 0 & 0 \\ 0 & 0 & -0.6 & 0.8 \\ 0 & 0 & -0.8 & -0.6 \end{bmatrix}$$

$$[T]^3 = \begin{bmatrix} -0.6 & -0.8 & 0 & 0 \\ 0.8 & -0.6 & 0 & 0 \\ 0 & 0 & -0.6 & -0.8 \\ 0 & 0 & 0.8 & -0.6 \end{bmatrix}$$

$$[T]^4 = \begin{bmatrix} 0.6 & 0.8 & 0 & 0 \\ -0.8 & 0.6 & 0 & 0 \\ 0 & 0 & 0.6 & 0.8 \\ 0 & 0 & -0.8 & 0.6 \end{bmatrix}$$

$$[T]^5 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

6. Compute global [K]

$$[K] = [T]^T[k][T]$$

$$[K]^1 = \begin{bmatrix} 277.78 & 0 & -277.78 & 0 \\ 0 & 0 & 0 & 0 \\ -277.78 & 0 & 277.78 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{kip}{in}$$

$$[K]^2 = \begin{bmatrix} 120 & -160 & -120 & 160 \\ -160 & 213.33 & 160 & -213.33 \\ -120 & 160 & 120 & -160 \\ 160 & -213.33 & -160 & 213.33 \end{bmatrix} \frac{kip}{in}$$

$$[K]^3 = \begin{bmatrix} 120 & 160 & -120 & -160 \\ 160 & 213.33 & -160 & -213.33 \\ -120 & -160 & 120 & 160 \\ -160 & -213.33 & 160 & 213.33 \end{bmatrix} \frac{kip}{in}$$

$$[K]^4 = \begin{bmatrix} 120 & 160 & -120 & -160 \\ 160 & 213.33 & -160 & -213.33 \\ -120 & -160 & 120 & 160 \\ -160 & -213.33 & 160 & 213.33 \end{bmatrix} \frac{kip}{in}$$

$$[K]^5 = \begin{bmatrix} 277.78 & 0 & -277.78 & 0 \\ 0 & 0 & 0 & 0 \\ -277.78 & 0 & 277.78 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{kip}{in}$$

7. Assemble [S] using Code # Method

$$S = \begin{bmatrix} K_{33}^1 + K_{11}^2 + K_{11}^4 & K_{34}^1 + K_{12}^2 + K_{12}^4 & K_{13}^2 \\ K_{43}^1 + K_{21}^2 + K_{21}^4 & K_{44}^1 + K_{22}^2 + K_{22}^4 & K_{23}^2 \\ K_{31}^2 & K_{32}^2 & K_{33}^2 + K_{11}^3 + K_{11}^5 \end{bmatrix}$$

$$S = \begin{bmatrix} 517.78 & 0 & -120 \\ 0 & 426.67 & 160 \\ -120 & 160 & 517.78 \end{bmatrix} \frac{kip}{in}$$

8. Solve {P}=[S]{d}

$$\{d\} = [S]^{-1}\{P\}$$

$$\{\boldsymbol{d}\} = \begin{Bmatrix} \boldsymbol{d_1} \\ \boldsymbol{d_2} \\ \boldsymbol{d_3} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0.0289} \\ \mathbf{-0.0579} \\ \mathbf{0.0246} \end{Bmatrix} \boldsymbol{in.}$$

9. Compatibility

$$\{v\}^1 = \begin{Bmatrix} 0 \\ 0 \\ d_1 \\ d_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0.0289 \\ -0.0579 \end{Bmatrix} in.$$

$$\{v\}^2 = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0.0289 \\ -0.0579 \\ 0.0246 \\ 0 \end{Bmatrix} in.$$

$$\{v\}^3 = \begin{Bmatrix} d_3 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0.0246 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \ in.$$

$$\{v\}^4 = \begin{Bmatrix} 0 \\ 0 \\ d_1 \\ d_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0.0289 \\ -0.0579 \end{Bmatrix} \ in.$$

$$\{v\}^5 = \begin{Bmatrix} d_3 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0.0246 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \ in.$$

10. Compute Global Member Forces

$$\{F\} = [K]\{v\}$$

$$\{F\}^1 = \begin{Bmatrix} -8.02 \\ 0 \\ 8.02 \\ 0 \end{Bmatrix} kips$$

$$\{F\}^2 = \begin{Bmatrix} 9.78 \\ -13.04 \\ -9.78 \\ 13.04 \end{Bmatrix} kips$$

$$\{F\}^3 = \begin{Bmatrix} 2.95 \\ 3.94 \\ -2.95 \\ -3.94 \end{Bmatrix} kips$$

$$\{F\}^4 = \begin{Bmatrix} -5.80 \\ -7.74 \\ 5.80 \\ 7.74 \end{Bmatrix} kips$$

$$\{F\}^5 = \begin{Bmatrix} -6.83 \\ 0 \\ 6.83 \\ 0 \end{Bmatrix} kips$$

11. Calculate Bar Forces and Stresses

$$\{Q\} = [T]\{F\}$$

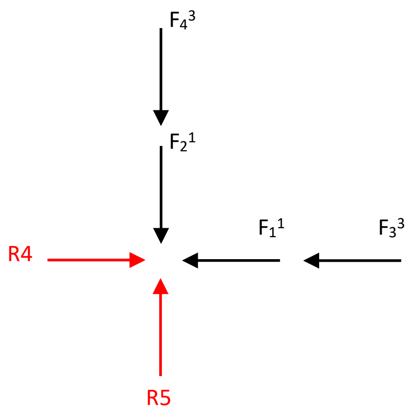
$$\boldsymbol{Q1\ bar\ force = 8.02\ kips\ (T)}$$
$$\boldsymbol{Q2\ bar\ force = 16.31\ kips\ (T)}$$
$$\boldsymbol{Q3\ bar\ force = 4.92\ kips\ (T)}$$
$$\boldsymbol{Q4\ bar\ force = 9.67\ kips\ (T)}$$
$$\boldsymbol{Q5\ bar\ force = 6.83\ kips\ (C)}$$

12. Calculate Bar Stresses

$$\sigma = F/A$$

$$\boldsymbol{Q1\ bar\ stress = 4.01\ ksi\ (T)}$$
$$\boldsymbol{Q2\ bar\ stress = 8.15\ ksi\ (T)}$$
$$\boldsymbol{Q3\ bar\ stress = 2.46\ ksi\ (T)}$$
$$\boldsymbol{Q4\ bar\ stress = 4.84\ ksi\ (T)}$$
$$\boldsymbol{Q5\ bar\ stress = 3.42\ ksi\ (C)}$$

13. Determine Support Reactions from Equilibrium



$$\Sigma F_x = 0, \quad \Sigma F_y = 0$$

$$\boldsymbol{R4 = F_1^1 + F_3^3 = -11.0\ kips}$$

$$\boldsymbol{R5 = F_2^1 + F_4^3 = -3.94\ kips}$$

$$\boldsymbol{R6 = F_3^4 + F_1^5 = -1.03\ kips}$$

$$\boldsymbol{R7 = F_4^4 + F_2^5 = 7.74\ kips}$$

$$\boldsymbol{R8 = F_4^2 + F_2^3 + F_4^5 = 17.0\ kips}$$

Appendix

Problem 1

Python INPUT

```
'''
HW6&7 Part b
units: kips, in
'''
######## IMPORT MODULES ########

import numpy as np # import matrix operators
solve = np.linalg.solve            # set up a shorthand for factorization

######## DEFINE FUNCTIONS ########
###
def Create_TK(LX,LY,E,A):
    L = np.sqrt(LX**2+LY**2)
    c = LX/L #cosine
    s = LY/L #sine
    k = (E*A/L)*np.array([[1,0,-1,0],
                    [0,0,0,0],
                    [-1,0,1,0],
                    [0,0,0,0]])

    T = np.array([[c,s,0,0],
              [-s,c,0,0],
              [0,0,c,s],
              [0,0,-s,c]])

    K = np.transpose(T).dot(k).dot(T)
    return T, K
###
def Assemble_S(S,k,MemNum,NEdof,NSdof,CodeNum):
    for KRow in range(0,NEdof,1):
        SRow=CodeNum[MemNum-1,KRow]
        if SRow <= NSdof:
            for KCol in range(0,NEdof,1):
                SCol=CodeNum[MemNum-1,KCol]
                if SCol <= NSdof:
                    S[SRow-1,SCol-1]=S[SRow-1,SCol-1]+k[KRow,KCol]

    return S
###

######## DEFINE SYSTEM FOR ANALYSIS ########

# Define the number of element dofs (always 4 for the 2D truss element)
NEdof=4

# Define the number of Structural dofs
NSdof=3

# Initialize S
S = np.zeros((NSdof,NSdof),dtype=float)

#Modulus of Elasticity
```

```
E = 10000 #ksi

# Define {P} vector
P = np.array([[24*np.cos(60*np.pi/180)], [-24*np.sin(60*np.pi/180)], [0]])
print 'P = '
print P.round(2),'kips'
print

# Define the code numbers
CodeNum=np.array([[4,5,1,2],
                 [1,2,3,8],
                 [3,8,4,5],
                 [1,2,6,7],
                 [6,7,3,8]]) # Here we write out each row explicitly
print 'CodeNum = '
print CodeNum
print

# Define geometry and loading for each member. Create the member stiffness
# and then assemble [S]. Note that no units are shown, you just need to be
# be consistent.

MemNum=1
A=2
LX=6*12 # sign (+/-) important
LY=0   # sign (+/-) important
T1, K1 = Create_TK(LX,LY,E,A)
S = Assemble_S(S,K1,MemNum,NEdof,NSdof,CodeNum)
print 'T1 ='
print T1.round(2)
print
print 'K1 ='
print K1.round(2),'kip/in'
print

MemNum=2
A=2
LX=-3*12 # sign (+/-) important
LY=4*12  # sign (+/-) important
T2, K2 = Create_TK(LX,LY,E,A)
S = Assemble_S(S,K2,MemNum,NEdof,NSdof,CodeNum)
print 'T2 ='
print T2.round(2)
print
print 'K2 ='
print K2.round(2),'kip/in'
print

MemNum=3
A=2
LX=-3*12 # sign (+/-) important
LY=-4*12 # sign (+/-) important
T3, K3 = Create_TK(LX,LY,E,A)
S = Assemble_S(S,K3,MemNum,NEdof,NSdof,CodeNum)
print 'T3 ='
print T3.round(2)
print
print 'K3 ='
```

```
print K3.round(2),'kip/in'
print

MemNum=4
A=2
LX=3*12 # sign (+/-) important
LY=4*12 # sign (+/-) important
T4, K4 = Create_TK(LX,LY,E,A)
S = Assemble_S(S,K4,MemNum,NEdof,NSdof,CodeNum)
print 'T4 ='
print T4.round(2)
print
print 'K4 ='
print K4.round(2),'kip/in'
print

MemNum=5
A=2
LX=-6*12 # sign (+/-) important
LY=0 # sign (+/-) important
T5, K5 = Create_TK(LX,LY,E,A)
S = Assemble_S(S,K5,MemNum,NEdof,NSdof,CodeNum)
print 'T5 ='
print T5.round(2)
print
print 'K5 ='
print K5.round(2),'kip/in'
print

print 'S ='
print S.round(2),'kip/in'
print


######## Solve ########

d = solve(S,P)

print 'd ='
print d.round(4),'inches'
print

######## POST-PROCESS ########

# Use compatibility to create the {v} vectors for each element
v1 = np.array([[0],[0],d[0],d[1]])
v2 = np.array([d[0],d[1],d[2],[0]])
v3 = np.array([d[2],[0],[0],[0]])
v4 = np.array([d[0],d[1],[0],[0]])
v5 = np.array([[0],[0],d[2],[0]])

# Calculate the member end forces
F1 = K1.dot(v1)      # GLOBAL {F} = [K]{v}
F2 = K2.dot(v2)
F3 = K3.dot(v3)
F4 = K4.dot(v4)
F5 = K5.dot(v5)
```

```
print 'F1 ='
print F1.round(2),'kips'
print
print 'F2 ='
print F2.round(2),'kips'
print
print 'F3 ='
print  F3.round(2),'kips'
print
print 'F4 ='
print  F4.round(2),'kips'
print
print 'F5 ='
print  F5.round(2),'kips'
print

Q1 = T1.dot(F1)       # LOCAL {Q} = [T]{F}
Q2 = T2.dot(F2)
Q3 = T3.dot(F3)
Q4 = T4.dot(F4)
Q5 = T5.dot(F5)

print 'Q1 ='
print Q1.round(2),'kips'
print
print 'Q2 ='
print Q2.round(2),'kips'
print
print 'Q3 ='
print  Q3.round(2),'kips'
print
print 'Q4 ='
print  Q4.round(2),'kips'
print
print 'Q5 ='
print  Q5.round(2),'kips'
print

#Calculate the bar (axial) forces and stresses
BarForces = np.array([Q1[2],Q2[2],Q3[2],Q4[2],Q5[2]])
print 'BarForces = '
print BarForces
print

BarStresses = BarForces/2
print 'BarStresses = '
print BarStresses
print

# Calculate the support reactions
R4 = F1[0]+F3[2]
R5 = F1[1]+F3[3]
R6 = F4[2]+F5[0]
R7 = F4[3]+F5[1]
R8 = F2[3]+F3[1]+F5[3]

print 'R4 =', R4[0].round(2),'kips'
print
```

```
print 'R5 =', R5[0].round(2),'kips'
print
print 'R6 =', R6[0].round(2),'kips'
print
print 'R7 =', R7[0].round(2),'kips'
print
print 'R8 =', R8[0].round(2),'kips'
print
```

Python OUTPUT

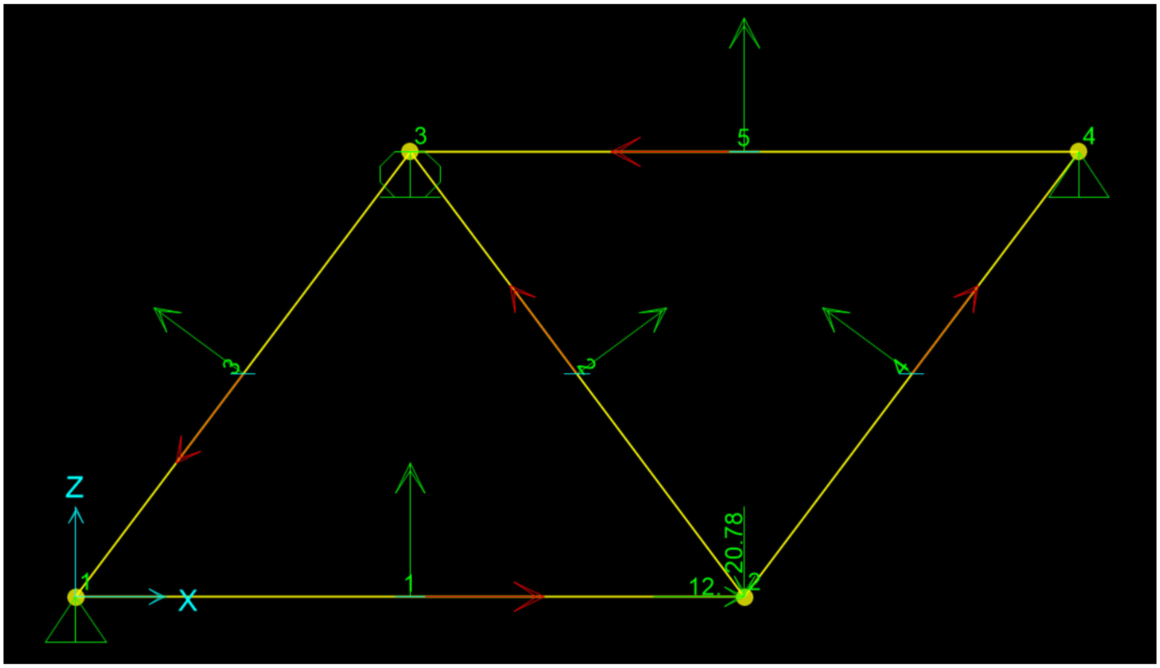
```
P =
[[ 12.  ]
 [-20.78]
 [  0.  ]] kips

CodeNum =
[[4 5 1 2]
 [1 2 3 8]
 [3 8 4 5]
 [1 2 6 7]
 [6 7 3 8]]

T1 =
[[ 1.  0.  0.  0.]
 [-0.  1.  0.  0.]
 [ 0.  0.  1.  0.]
 [ 0.  0. -0.  1.]]

K1 =
[[ 277.78    0.   -277.78    0.  ]
 [   0.      0.      0.      0.  ]
 [-277.78    0.    277.78    0.  ]
 [   0.      0.      0.      0.  ]] kip/in

T2 =
[[-0.6  0.8  0.   0. ]
 [-0.8 -0.6  0.   0. ]
 [ 0.   0.  -0.6  0.8]
 [ 0.   0.  -0.8 -0.6]]

K2 =
[[ 120.   -160.   -120.    160.  ]
 [-160.    213.33  160.   -213.33]
 [-120.    160.    120.   -160.  ]
 [ 160.   -213.33 -160.    213.33]] kip/in

T3 =
[[-0.6 -0.8  0.   0. ]
 [ 0.8 -0.6  0.   0. ]
 [ 0.   0.  -0.6 -0.8]
 [ 0.   0.   0.8 -0.6]]

K3 =
[[ 120.    160.   -120.   -160.  ]
 [ 160.    213.33 -160.   -213.33]
 [-120.   -160.    120.    160.  ]
 [-160.   -213.33  160.    213.33]] kip/in
```

```
T4 =
[[ 0.6  0.8  0.   0. ]
 [-0.8  0.6  0.   0. ]
 [ 0.   0.   0.6  0.8]
 [ 0.   0.  -0.8  0.6]]

K4 =
[[ 120.    160.   -120.   -160.  ]
 [ 160.    213.33 -160.   -213.33]
 [-120.   -160.    120.    160.  ]
 [-160.   -213.33  160.    213.33]] kip/in

T5 =
[[-1.  0.  0.  0.]
 [-0. -1.  0.  0.]
 [ 0.  0. -1.  0.]
 [ 0.  0. -0. -1.]]

K5 =
[[ 277.78    0.   -277.78    0.  ]
 [   0.      0.      0.      0.  ]
 [-277.78    0.    277.78    0.  ]
 [   0.      0.      0.      0.  ]] kip/in

S =
[[ 517.78    0.   -120.  ]
 [   0.    426.67  160.  ]
 [-120.    160.    517.78]] kip/in

d =
[[ 0.0289]
 [-0.0579]
 [ 0.0246]] inches

F1 =
[[-8.02]
 [ 0.  ]
 [ 8.02]
 [ 0.  ]] kips

F2 =
[[  9.78]
 [-13.04]
 [ -9.78]
 [ 13.04]] kips

F3 =
[[ 2.95]
 [ 3.94]
 [-2.95]
 [-3.94]] kips

F4 =
[[-5.8 ]
 [-7.74]
 [ 5.8 ]
 [ 7.74]] kips
```

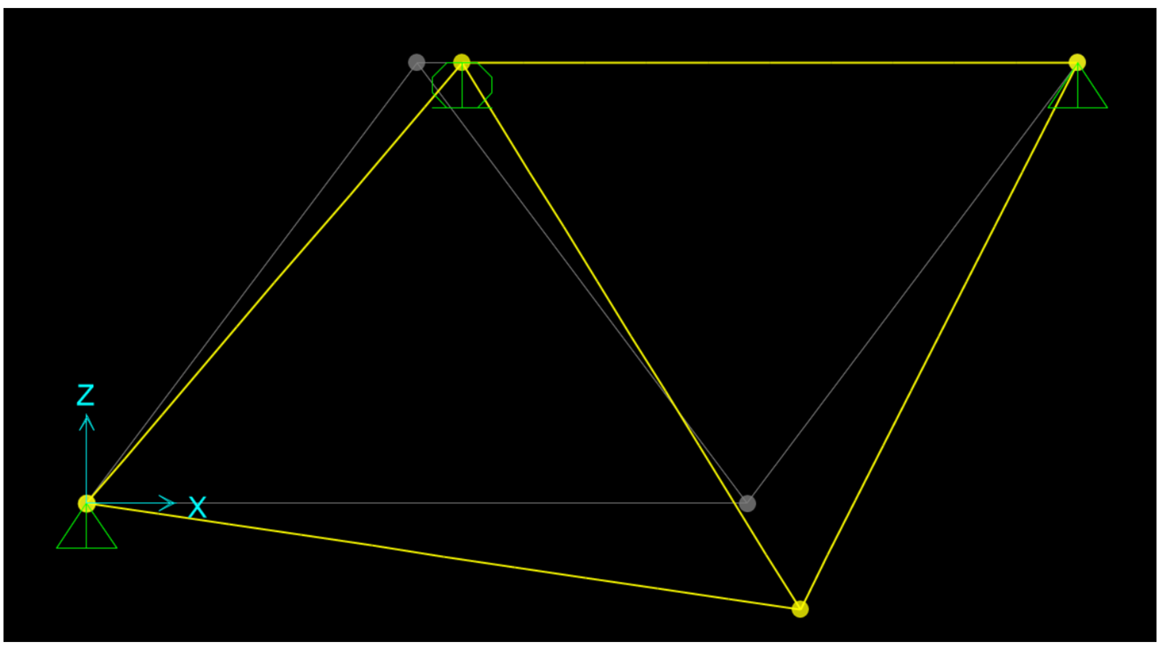
```
F5 =
[[-6.83]
 [ 0.  ]
 [ 6.83]
 [ 0.  ]] kips

Q1 =
[[-8.02]
 [ 0.  ]
 [ 8.02]
 [ 0.  ]] kips

Q2 =
[[-16.31]
 [  0.  ]
 [ 16.31]
 [  0.  ]] kips

Q3 =
[[-4.92]
 [ 0.  ]
 [ 4.92]
 [ 0.  ]] kips

Q4 =
[[-9.67]
 [ 0.  ]
 [ 9.67]
 [ 0.  ]] kips

Q5 =
[[ 6.83]
 [ 0.  ]
 [-6.83]
 [ 0.  ]] kips

BarForces =
[[  8.02118458]
 [ 16.30606057]
 [  4.91914676]
 [  9.67470154]
 [ -6.83214828]]

BarStresses =
[[ 4.01059229]
 [ 8.15303029]
 [ 2.45957338]
 [ 4.83735077]
 [-3.41607414]]

R4 = -10.97 kips

R5 = -3.94 kips

R6 = -1.03 kips

R7 = 7.74 kips
```

R8 = 16.98 kips

SAP OUTPUT



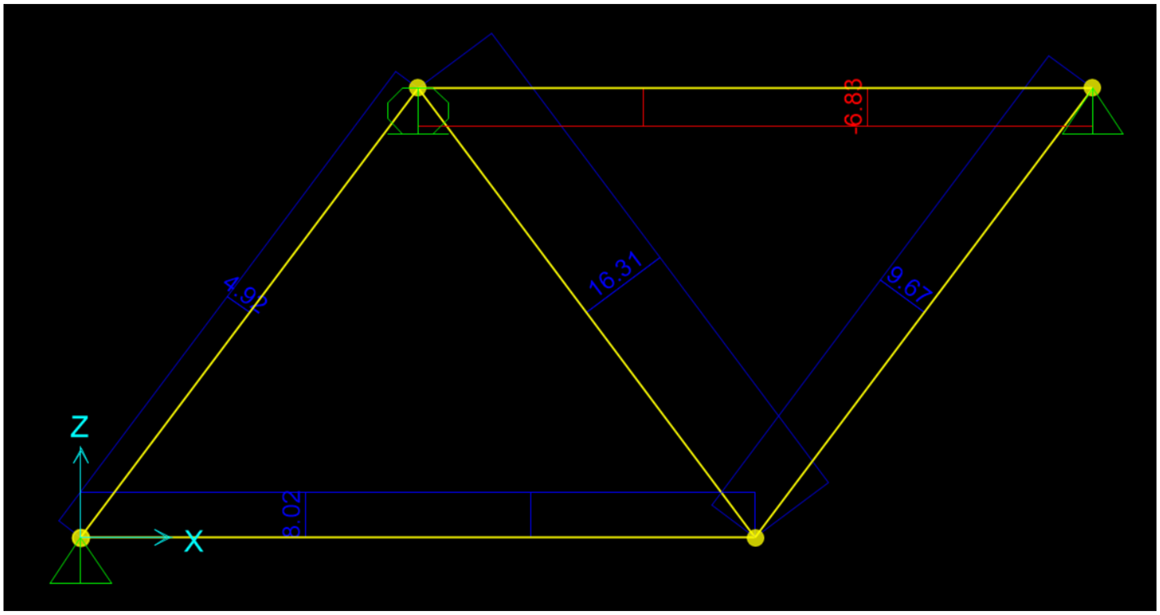
SAP Truss Model - Restraints, Loads, Labels, and Frame Element Local Axes



Deformed Shape

TABLE: Joint Displacements								
Joint	**OutputCase**	**CaseType**	**U1**	**U2**	**U3**	**R1**	**R2**	**R3**
Text	Text	Text	in	in	in	Radians	Radians	Radians
1	**DEAD**	**LinStatic**	**0**	**0**	**0**	**0**	**0**	**0**
2	**DEAD**	**LinStatic**	**0.028876**	**0**	**-0.057937**	**0**	**0**	**0**
3	**DEAD**	**LinStatic**	**0.024596**	**0**	**0**	**0**	**0**	**0**

4	**DEAD**	**LinStatic**	**0**	**0**	**0**	**0**	**0**	**0**



Frame Element Forces

TABLE: Element Forces - Frames				
Frame	**Station**	**OutputCase**	**CaseType**	**P**
Text	in	Text	Text	Kip
1	**0**	**DEAD**	**LinStatic**	**8.021**
2	**0**	**DEAD**	**LinStatic**	**16.306**
3	**0**	**DEAD**	**LinStatic**	**4.919**
4	**0**	**DEAD**	**LinStatic**	**9.675**
5	**0**	**DEAD**	**LinStatic**	**-6.832**

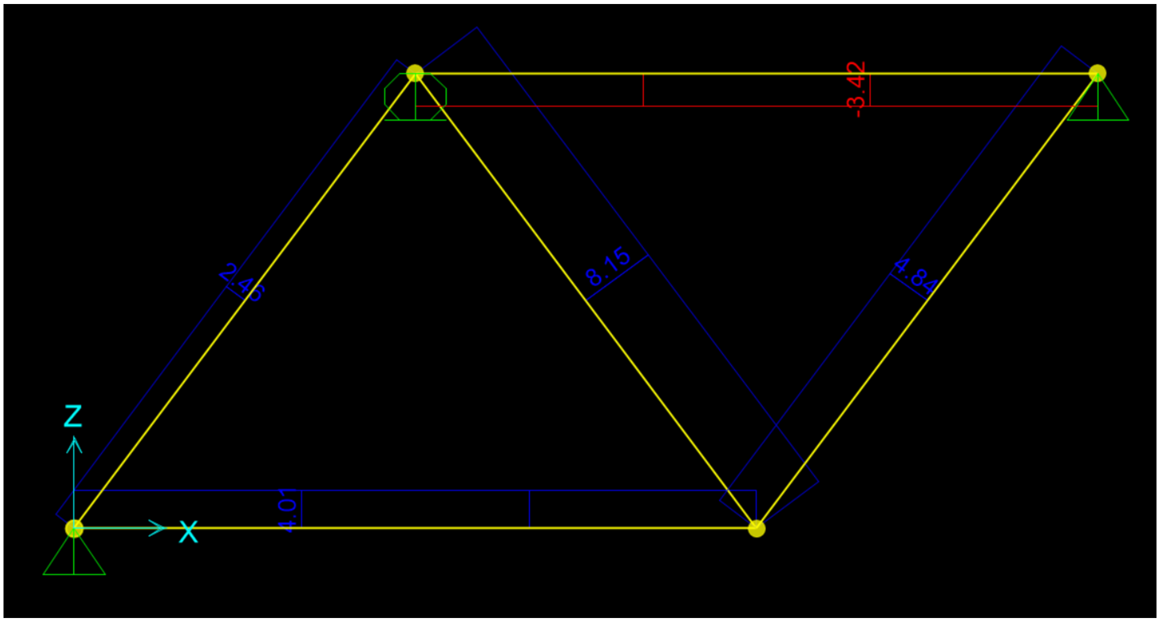
NOTE

Stations in SAP

SAP displays element output at stations. Stations are simply arbitrary points along a member. For trusses, the axial forces/stresses will not vary along a member, so every station will report the same, constant value. Therefore, for trusses, it is only necessary to report one station value for each element.

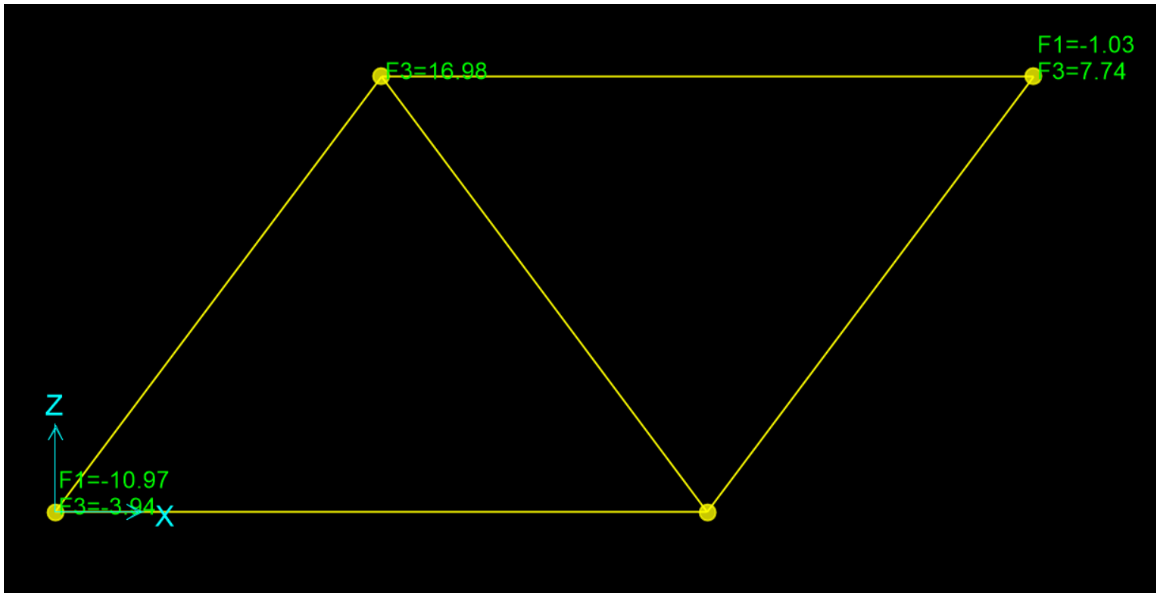
This will not be the case for Beams or 2D Frames, as member loads will impose varying forces/stresses along a member.

Stations will be explained further in the SAP Beam Tutorial and the SAP 2D Frame Tutorial.



Frame Element Stresses (S11)

TABLE: Element Stresses - Frames							
Frame	**Station**	**OutputCase**	**CaseType**	**Point**	**X2**	**X3**	**S11**
Text	in	Text	Text	Text	in	in	Kip/in2
1	**0**	**DEAD**	**LinStatic**	**0**	**0**	**0**	**4.0106**
2	**0**	**DEAD**	**LinStatic**	**0**	**0**	**0**	**8.153**
3	**0**	**DEAD**	**LinStatic**	**0**	**0**	**0**	**2.4596**
4	**0**	**DEAD**	**LinStatic**	**0**	**0**	**0**	**4.8374**
5	**0**	**DEAD**	**LinStatic**	**0**	**0**	**0**	**-3.4161**



Joint Reactions

TABLE: Joint Reactions								
Joint	**OutputCase**	**CaseType**	**F1**	**F2**	**F3**	**M1**	**M2**	**M3**
Text	Text	Text	Kip	Kip	Kip	Kip-in	Kip-in	Kip-in
1	**DEAD**	**LinStatic**	**-10.973**	**0**	**-3.935**	**0**	**0**	**0**
3	**DEAD**	**LinStatic**	**0**	**0**	**16.98**	**0**	**0**	**0**
4	**DEAD**	**LinStatic**	**-1.027**	**0**	**7.74**	**0**	**0**	**0**