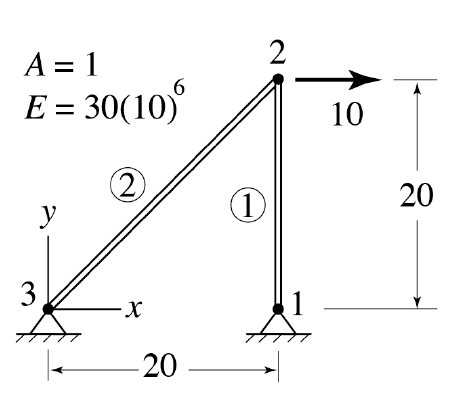
**Model 1**

Figure 1. Node and Element Numbers

Though not stated explicitly, I assume that it is **SI** units thus kg, m, s. So length is 20m; E = 30\*106 N/m2; A = 1m2; F=10N.

The import file is 2-D Truss Example.txt. Run the matlab script twoDTruss.m, we can get the following results:

============================================================

BAR TEST PROGRAM

============================================================

Displacements [m]

DOF 1 : +0.00000000

DOF 2 : +0.00000000

DOF 3 : +0.00002552

DOF 4 : -0.00000667

DOF 5 : +0.00000000

DOF 6 : +0.00000000

============================================================

Element Strains

#001 : -3.33333333e-07

#002 : +4.71404521e-07

============================================================

Element Stresses [Pa]

#001 : -1.00000000e+01

#002 : +1.41421356e+01

============================================================

Figure 2. Original and Deformed shape

From Fig. 2, we can see that the deformation is very small. Because A is 1 m2 while F is only 10 N.

In order to verify the result, we can look at the equilibrium function at node 2.

It is easy to get:

For element 1 (compression), its inner force is -10 N;

For element 2 (traction), its inner force is N;

So ,

They are the same with the results obtained by FEM program.

In the FEM program, the calculation sequence is displacement 🡪strain🡪stress. So it is a strong proof that the program works right.

**Model 2**

**Wind Load**

The steel structure is divided into 16 nodes and 36 elements. In order to simulate a rigid plate, I set a very large Young’s Modulus (E = 29X1012 psi) to that element. Also in order to keep the input file neat, I choose to change the units after importing Data to program.

The import file is Steel Structure.txt. Run the matlab script windLoad.m, we can get the following results:

============================================================

Steel Structure

============================================================

Displacements [m]

DOF 1 : +0.00000000

DOF 2 : +0.00000000

DOF 3 : +0.00000000

DOF 4 : +0.00000000

DOF 5 : +0.00228887

DOF 6 : +0.00084964

DOF 7 : +0.00228324

DOF 8 : -0.00084888

DOF 9 : +0.00749408

DOF 10 : +0.00149531

DOF 11 : +0.00748862

DOF 12 : -0.00149305

DOF 13 : +0.01485236

DOF 14 : +0.00196218

DOF 15 : +0.01484690

DOF 16 : -0.00195845

DOF 17 : +0.02369956

DOF 18 : +0.00227580

DOF 19 : +0.02369408

DOF 20 : -0.00227059

DOF 21 : +0.03347364

DOF 22 : +0.00246170

DOF 23 : +0.03346838

DOF 24 : -0.00245504

DOF 25 : +0.04371864

DOF 26 : +0.00254590

DOF 27 : +0.04370625

DOF 28 : -0.00253686

DOF 29 : +0.05400450

DOF 30 : +0.00256381

DOF 31 : +0.05399462

DOF 32 : -0.00253686

============================================================

Element Strains

#001 : +0.00000000e+00

#002 : +2.78751893e-04

#003 : -2.78502440e-04

#004 : -7.73757682e-05

#005 : +7.68356907e-05

#006 : -3.69820202e-06

#007 : +2.11834678e-04

#008 : -2.11343424e-04

#009 : -6.85662596e-05

#010 : +6.75026747e-05

#011 : -3.58473812e-06

#012 : +1.53172614e-04

#013 : -1.52688787e-04

#014 : -5.94869566e-05

#015 : +5.84394531e-05

#016 : -3.58808425e-06

#017 : +1.02894285e-04

#018 : -1.02409943e-04

#019 : -5.04162528e-05

#020 : +4.93676323e-05

#021 : -3.59238685e-06

#022 : +6.09907614e-05

#023 : -6.05158226e-05

#024 : -4.13348112e-05

#025 : +4.03065495e-05

#026 : -3.44867655e-06

#027 : +2.76248304e-05

#028 : -2.68436383e-05

#029 : -3.25950743e-05

#030 : +3.09037618e-05

#031 : -8.13265429e-06

#032 : +5.87453087e-06

#033 : -6.69511592e-12

#034 : -1.27185939e-05

#035 : +1.44951929e-05

#036 : -6.48244735e-06

============================================================

Element Stresses [Pa]

#001 : +0.00000000e+00

#002 : +5.57358728e+07

#003 : -5.56859950e+07

#004 : -1.54711271e+07

#005 : +1.53631397e+07

#006 : -7.39447955e+05

#007 : +4.23559121e+07

#008 : -4.22576868e+07

#009 : -1.37096838e+07

#010 : +1.34970222e+07

#011 : -7.16761079e+05

#012 : +3.06265518e+07

#013 : -3.05298117e+07

#014 : -1.18942957e+07

#015 : +1.16848495e+07

#016 : -7.17430131e+05

#017 : +2.05735025e+07

#018 : -2.04766593e+07

#019 : -1.00806270e+07

#020 : +9.87095745e+06

#021 : -7.18290427e+05

#022 : +1.21949784e+07

#023 : -1.21000154e+07

#024 : -8.26481123e+06

#025 : +8.05921240e+06

#026 : -6.89555846e+05

#027 : +5.52352853e+06

#028 : -5.36733076e+06

#029 : -6.51731865e+06

#030 : +6.17914417e+06

#031 : -1.62610765e+06

#032 : +1.17460047e+06

#033 : -1.33867478e+06

#034 : -2.54305693e+06

#035 : +2.89828428e+06

#036 : -1.29615213e+06

Figure 3. Original and Deformed Shape Due to Wind Load

Figure 3 shows the original and deformed shape of the steel structure. In order to make the deformation clear, I adjust the relative scaling of X and Y. The results accords with the physical sense.

*Point of impact change due to deformation:*

Run the function distance(displacements, eleNodes). The distance is 81.4873 m.

**Verification**

1. The displacements of node 1 and 2 (first four DOFs) are all zero. The strain and stress of element #1 is zero. These satisfy BCs. The strain of element #1 is -6.69511592e-12 which is very small. It accords with our expectation.
2. Use equilibrium of moment (at node 1), we can easily obtain the vertical reaction force of node 2 is

Add vertical force of element 3 and 4, we can run the following command

sigma(4)\*material(1,1)\*10/sqrt(125) + sigma(3)\*material(2,1)

The result is -3.3362e+04 .

So the equilibrium equation of vertical forces of node 2 is satisfied which is an evidence that the result is right.

1. Force equilibrium at node 5:

Vertical

sigma(12)\*material(2,1) + sigma(15)\*material(1,1)\*10/sqrt(125) - sigma(7)\*material(2,1) -… sigma(9)\*material(1,1)\*10/sqrt(125) = 2.5466e-11

Horizon

sigma(15)\*material(1,1)\*5/sqrt(125)+sigma(9)\*material(1,1)\*5/sqrt(125)+sigma(11)\*material(1,1)

= -444.8218 N -100lb

So the equilibrium equations are satisfied.

**Thermal Load**

In order to keep it simple, we still use Steel Structure.txt as the import file. The external force vector is updated in the program. Run the matlab script thermalLoad.m, we can get the following results:

============================================================

Steel Structure

============================================================

Displacements [m]

DOF 1 : +0.00000000

DOF 2 : +0.00000000

DOF 3 : +0.00000000

DOF 4 : +0.00000000

DOF 5 : +0.00055762

DOF 6 : +0.00065054

DOF 7 : +0.00072254

DOF 8 : +0.00001046

DOF 9 : +0.00248039

DOF 10 : +0.00129030

DOF 11 : +0.00264025

DOF 12 : +0.00001014

DOF 13 : +0.00568071

DOF 14 : +0.00193039

DOF 15 : +0.00584073

DOF 16 : +0.00001015

DOF 17 : +0.01016127

DOF 18 : +0.00257047

DOF 19 : +0.01032129

DOF 20 : +0.00001015

DOF 21 : +0.01592199

DOF 22 : +0.00321055

DOF 23 : +0.01608201

DOF 24 : +0.00001015

DOF 25 : +0.02296287

DOF 26 : +0.00385063

DOF 27 : +0.02312289

DOF 28 : +0.00001015

DOF 29 : +0.03128391

DOF 30 : +0.00449071

DOF 31 : +0.03144393

DOF 32 : +0.00001015

============================================================

Element Strains

#001 : +0.00000000e+00

#002 : +2.13432615e-04

#003 : +3.43261465e-06

#004 : +9.75682522e-05

#005 : +9.75682522e-05

#006 : +1.08221605e-04

#007 : +2.09894681e-04

#008 : -1.05319311e-07

#009 : +1.05228021e-04

#010 : +1.05228021e-04

#011 : +1.04901155e-04

#012 : +2.10003231e-04

#013 : +3.23140264e-09

#014 : +1.04993004e-04

#015 : +1.04993004e-04

#016 : +1.05003033e-04

#017 : +2.09999901e-04

#018 : -9.91449982e-11

#019 : +1.05000215e-04

#020 : +1.05000215e-04

#021 : +1.04999907e-04

#022 : +2.10000003e-04

#023 : +3.04267088e-12

#024 : +1.04999993e-04

#025 : +1.04999993e-04

#026 : +1.05000003e-04

#027 : +2.10000000e-04

#028 : -9.26847003e-14

#029 : +1.05000000e-04

#030 : +1.05000000e-04

#031 : +1.05000000e-04

#032 : +2.10000000e-04

#033 : +3.38813179e-21

#034 : +1.05000000e-04

#035 : +1.05000000e-04

#036 : +1.05000000e-04

============================================================

Element Stresses [Pa]

#001 : -2.09945359e+07

#002 : +6.86344302e+05

#003 : +6.86344302e+05

#004 : -1.48596282e+06

#005 : -1.48596282e+06

#006 : +6.44153308e+05

#007 : -2.10583818e+04

#008 : -2.10583816e+04

#009 : +4.55922375e+04

#010 : +4.55922375e+04

#011 : -1.97638798e+04

#012 : +6.46112184e+02

#013 : +6.46112371e+02

#014 : -1.39885872e+03

#015 : -1.39885873e+03

#016 : +6.06394343e+02

#017 : -1.98240145e+01

#018 : -1.98238403e+01

#019 : +4.29197594e+01

#020 : +4.29197464e+01

#021 : -1.86053957e+01

#022 : +6.08212720e-01

#023 : +6.08375841e-01

#024 : -1.31681750e+00

#025 : -1.31682846e+00

#026 : +5.70811571e-01

#027 : -1.86865436e-02

#028 : -1.85321169e-02

#029 : +4.04524282e-02

#030 : +4.04442771e-02

#031 : -1.74531834e-02

#032 : +6.54666044e-04

#033 : +6.77450044e-04

#034 : -1.41740976e-03

#035 : -1.42252586e-03

#036 : +6.34927860e-04

****============================================================

Figure 4. Original and Deformed Shape Due to Thermal Load

*Point of impact change due to deformation:*

Run the function distance(displacements, eleNodes). The distance is 65.9025 m.

Finally, I will briefly introduce the functions and scripts in my matlab program.

1. **importData.m**

This function imports data from txt file and initializes variables

Input:

fileName: file name, like 2-D Truss Example.txt

Output:

nodeNum: # of nodes

dofPerNode: # of DOFs per node

eleNum: # of elements

nodePerEle: # of nodes per element

spaceDim: space dimensions

materialSet: # of material sets

nodeCoordinate: global coordinate of nodes

ExForce: external force vector

eleNodes: the global node number of every element

prescribedDof: prescribed DOF

material: material property

title: question name (like steel structure or 2D truss)

1. **strucStiffMatrix.m**

This function calculates the structure stiffness matrix

Input:

nodeNum: # of nodes

dofPerNode: # of DOFs per node

eleNum: # of elements

nodePerEle: # of nodes per element

nodeCoordinate: global coordinate of nodes

eleNodes: the global node number of every element

material: material property

Output

stiffMatrix: structure stiffness matrix

1. **unitChange.m**

This script change units to SI units

Change lb to N;

Change psi to pascal;

Change in2 to m2;

Change ft to m.

1. **solveEqus.m**

This function adds BCs and solves the reduced equilibrium equations to get displacements

Input:

nodeNum: # of nodes

dofPerNode: # of DOFs per node

prescribedDof: prescribed DOFs

stiffMatrix: structure stiffness matrix

ExForce: external force vector

Output:

displacements: displacements vector

1. **draw.m**

This function plots the original and deformed shape

Input:

nodeCoordinate: coordinates of nodes

displacements: displacements

eleNodes: the global node number of every element

1. **temperature.m**

This function calculates the temperature

Input:

x: X coordinate

Output

T: temperature

1. **Output.m**

This function outputs results to command window

Input:

displacements: displacement of every DOF

sigma: stress of every element

title: question name

1. **calStresses.m**

This function calculates strains and stresses

Input:

sigmaZero: initial stress

epsilonZero: initial strain

eleNum: # of elements

nodeCoordinate: global coordinate of nodes

eleNodes: the global node number of every element

displacements: displacement of every DOF

material: material property

Output:

sigma: stress of every element

spsilon: strain of every element

1. **thermalForce.m**

This function calculates thermal load terms

Input:

nodeNum: # of nodes

dofPerNode: # of DOFs per node

eleNum: # of elements

nodeCoordinate: coordinates of every node

eleNodes: the global node number of every element

material: material property

alpha: coefficient of thermal expansion

Output:

thermaNodeLoad: node load caused by thermal

1. **thermalStrain.m**

This function calculates the initial strain caused by thermal

Input:

nodeCoordinate: coordinates of every node

eleNodes: the global node number of every element

eleNum: # of elements

material: material property matrix

alpha: coefficient of thermal expansion

Output:

epsilonZero: initial strain caused by thermal

1. **distance.m**

This function calculates point of impact change due to deformation

Input

displacments: displacements vector

eleNodes: the global node number of every element

Output

D: point of impact change

1. **twoDTruss.m**  
   This script calculates the 2-D truss example
2. **windLoad.m**This script calculates the steel structure under wind load
3. **thermalLoad.m**

This script calculates steel structure under thermal load