Finite Element Method analysis of 3D space truss with application to power transmission tower

Group H

Course Instructor -

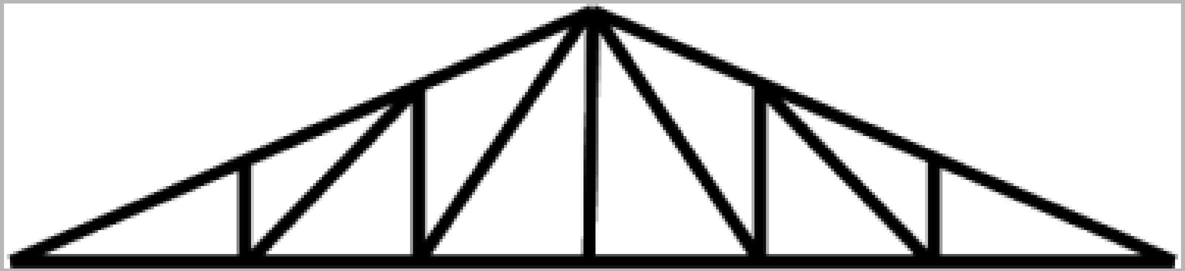
Dr. Ashok Kumar Pandey

NO	MEMBER NAME:	CONTRIBUTION:
1	**Aditya Agrawal (ME20MTECH14020)**	Matlab code, Literature survey, Presentation
2	**Sagar Claire (ME20MTECH11024)**	Matlab code, Analytical solution, Report making
3	**Pudtha Sadhana (ME18BTECH11028)**	Report making
4	**Regula Anusree (ME18BTECH11031)**	Report making
5	**Hari Haran Dandugula (ME20MTECH11012)**	Presentation
6	**Vinay Mishra (ME20MTECH11002)**	Presentation
7	**Shantanu Das (ME20MTECH14012)**	Literature survey, report
8	**Imran Riyaz (ME20MTECH14017)**	Literature survey, Diagrams
9	**Ajith Ramakrishnan (ME20MTECH14008)**	Literature survey, report

INTRODUCTION

- A truss is a structure composed of slender members joined together at their end points. The members are organized into connected triangles so that overall assembly behaves as a single object.

- The Triangulated system of members are connected in a way such that they only incur axial force, resulting in either a compression or tension force.



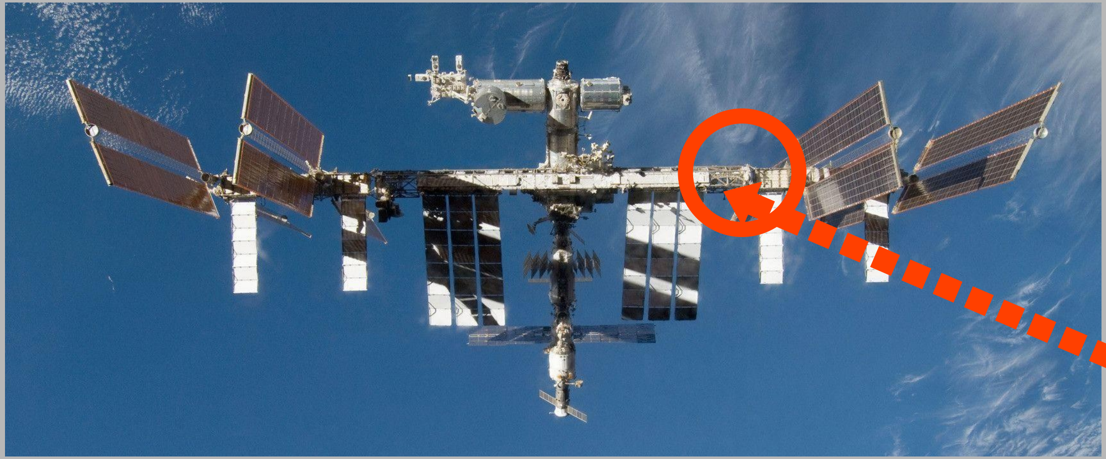
- Truss members are connected through pinned connections therefore the forces get transferred axially and there are very little bending moment and shear forces in the members.

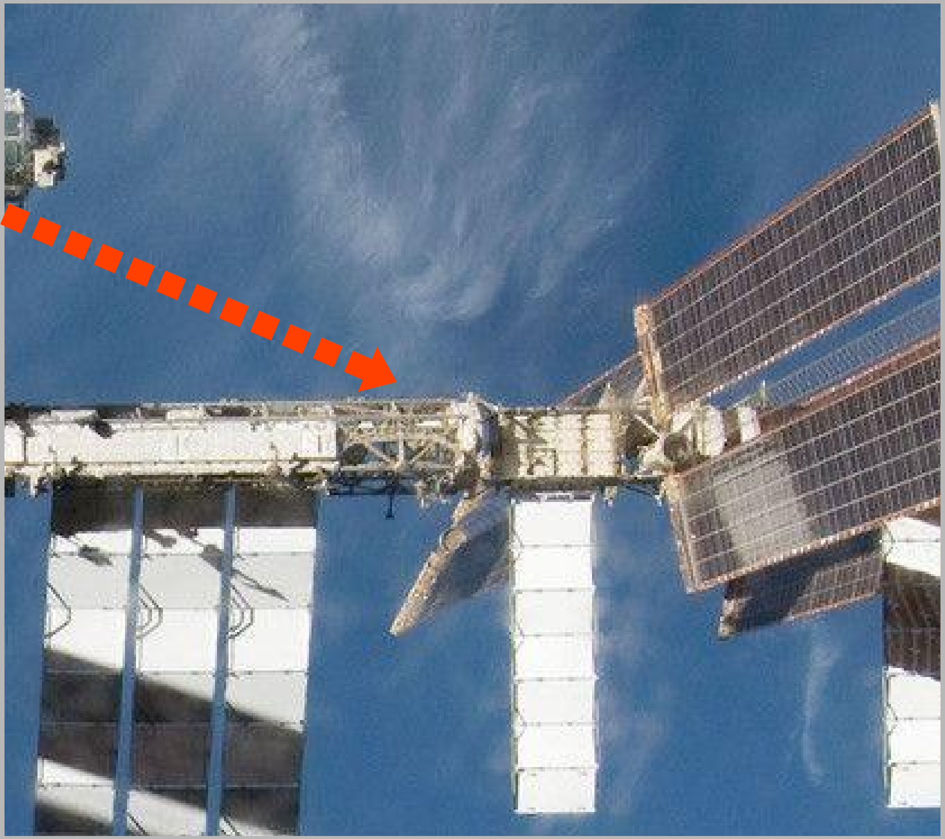
- Members are more efficient in carrying axial loads since the stresses are evenly distributed, rather than transverse loads which are concentrated. This combination makes truss systems more efficient than single-member alternatives

APPLICATIONS

- Roof of Factory Shade.
- Ware House
- Railway platform Roof
- Garage shed
- Transmission towers
- Crane truss
- Bridge Truss
- Sport Stadium Truss

APPLICATIONS





THEORETICAL FINITE ELEMENT FORMULATION OF TRUSS ELEMENT

Hooke's Law and Deformation Equations

$$\sigma = E\varepsilon \qquad \varepsilon = \frac{d\hat{u}}{d\hat{x}}$$

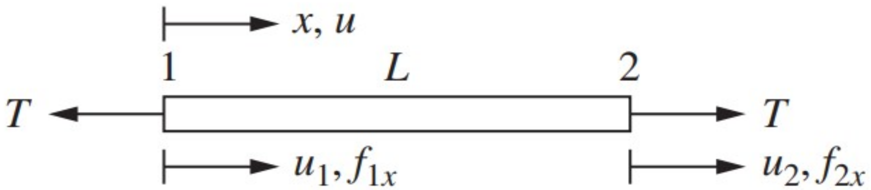
Equilibriums

$$A\sigma_x = T = \text{Constant} \qquad \frac{d}{d\hat{x}}\left(AE\frac{d\hat{u}}{d\hat{x}}\right) = 0$$

The following assumptions are used in deriving the bar element stiffness matrix:

1. The bar cannot sustain shear force or bending moment, that is, $f_{1y} = 0$, $f_{2y} = 0$, $m_1 = 0$ and $m_2 = 0$.
2. Any effect of transverse displacement is ignored.
3. Hooke's law applies; that is, axial stress σ_x is related to axial strain ε_x by $\sigma_x = E\varepsilon_x$.
4. No intermediate applied loads.

Step 1: Select element type



Step 2: Determination of displacement function

Assume a linear deformation function as

$$\hat{u} = a_1 + a_2\hat{x} = \left(\frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}\right)\hat{x} + \hat{d}_{1x}$$

In vector form

$$\hat{u} = [N_1 \ \ N_2]\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

$$N_1 = 1 - \frac{\hat{x}}{L}, \qquad N_2 = \frac{\hat{x}}{L}$$

N_1, N_2: shape function

Step 3: Strain and stress calculation

Deformation rate - strain

$$\varepsilon_x = \frac{d\hat{u}}{d\hat{x}} = \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}$$

Stress - strain

$$\sigma_x = E\varepsilon_x$$

Step 4: Derivation of element stiffness matrix

$$T = A\sigma_x = AE\left(\frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}\right)$$

$$\hat{f}_{1x} = -T = \frac{AE}{L}(\hat{d}_{1x} - \hat{d}_{2x})$$

$$\hat{f}_{2x} = T = \frac{AE}{L}(\hat{d}_{2x} - \hat{d}_{1x})$$

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

$$\underline{\hat{k}} = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

NOTE: In a truss element, stiffness (spring constant, *k*) is equivalent to *AE/L*

Step 5: Constitution of global stiffness matrix

Construct a global stiffness matrix in a global coordinate system

$$\underline{K} = [K] = \sum_{e=1}^{N} \underline{k}^{(e)} \qquad \underline{F} = \{F\} = \sum_{e=1}^{N} \underline{f}^{(e)}$$

Step 6: Calculation of nodal displacement

- Use boundary conditions
- Solve the system of linear algebraic equations $\underline{F} = \underline{K}\,\underline{d}$ to calculate the nodal deformation in a truss structure

Step 7: Calculation of stress and strain in an element

Compute a strain and stress at any point within an element

POTENTIAL ENERGY FORMULATION

$$\pi_p = \frac{AL}{2}\{d\}^T[B]^T[D]^T[B]\{d\} - \{d\}^T\{f\}$$

$$\{d\} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$[B] = \left[-\frac{1}{L}\ \frac{1}{L}\right]$$

$$\{f\} = \{P\}$$

$$\frac{\partial \pi_p}{\partial u_1} = 0 \quad \text{and} \quad \frac{\partial \pi_p}{\partial u_2} = 0$$

$$\{U^*\} = \{d\}^T[B]^T[D]^T[B]\{d\}$$

$$U^* = \frac{E}{L^2}(u_1^2 - 2u_1u_2 + u_2^2)$$

$$\{d\}^T\{f\} = u_1 f_{1x} + u_2 f_{2x}$$

POTENTIAL ENERGY FORMULATION

$$\frac{\partial \pi_p}{\partial u_1} = \frac{AL}{2}\left[\frac{E}{L^2}(2u_1 - 2u_2)\right] - f_{1x} = 0$$

$$\frac{\partial \pi_p}{\partial u_2} = \frac{AL}{2}\left[\frac{E}{L^2}(-2u_1 + 2u_2)\right] - f_{2x} = 0$$

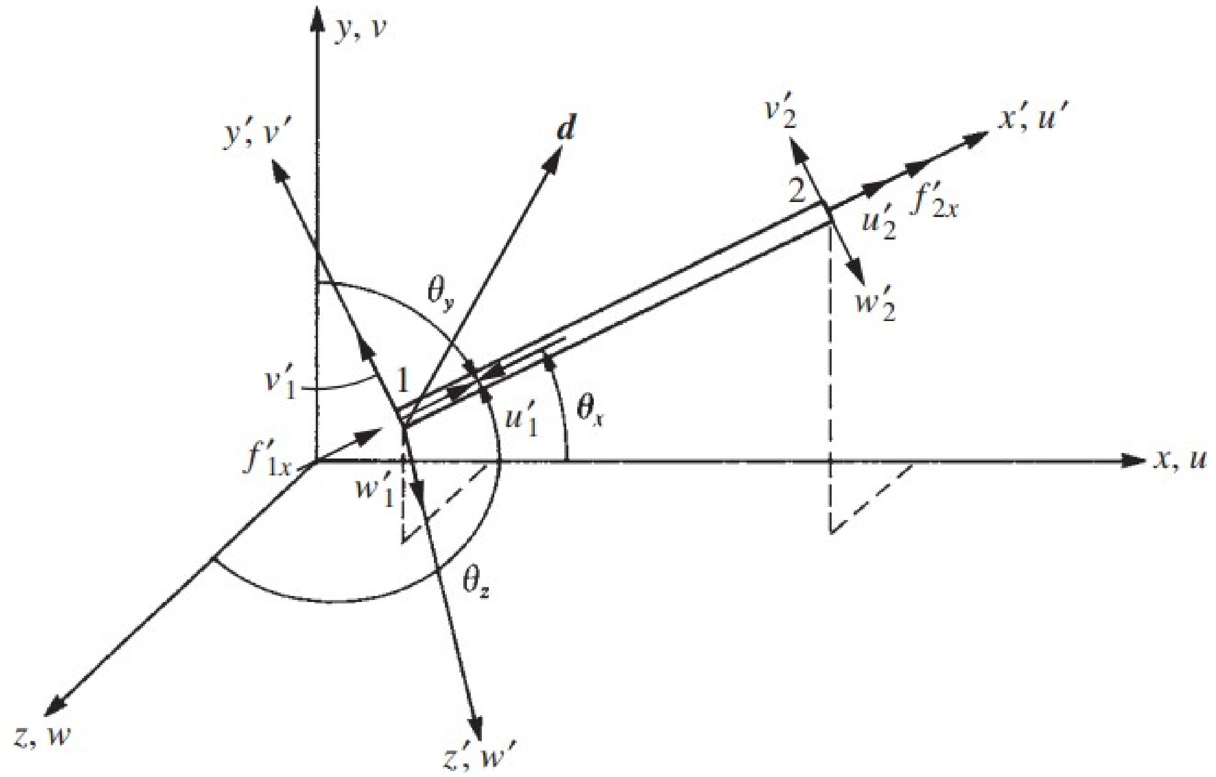
Combining above 2 equations-

$$\frac{\partial \pi_p}{\partial \{d\}} = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} - \begin{Bmatrix} f_{1x} \\ f_{2x} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

THEORETICAL FINITE ELEMENT FORMULATION OF 3D SPACE TRUSS ELEMENT

X’,y’,z’ = local coordinate system

X,y,z = global coordinate system



We can write local axial displacement at node 1 and 2 in matrix form as follows-

$$\begin{Bmatrix} u_1' \\ u_2' \end{Bmatrix} = \begin{bmatrix} C_x & C_y & C_z & 0 & 0 & 0 \\ 0 & 0 & 0 & C_x & C_y & C_z \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \end{Bmatrix}$$

$$\mathbf{i}' \cdot \mathbf{i} = \frac{x_2 - x_1}{L} = C_x$$

$$\mathbf{i}' \cdot \mathbf{j} = \frac{y_2 - y_1}{L} = C_y$$

$$\mathbf{i}' \cdot \mathbf{k} = \frac{z_2 - z_1}{L} = C_z$$

$$\{d'\} = \begin{Bmatrix} u_1' \\ u_2' \end{Bmatrix}, \{d\} = \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \end{Bmatrix}, \text{ and define } [T^*] = \begin{bmatrix} C_x & C_y & C_z & 0 & 0 & 0 \\ 0 & 0 & 0 & C_x & C_y & C_z \end{bmatrix}.$$

$$\{d'\} = [T^*]\{d\}$$

T* = transformation matrix
In local coordinates local force is related to local displacement as-

$$\{f'\} = [k']\{d'\}$$

Multiply both sides of equation by inverse of T*

$$[T^*]^T\{f'\} = [T^*]^T[k'][T^*]\{d\}$$

Because of orthogonal matrix property it can be written-

$$\{f\} = [T^*]^T[k'][T^*]\{d\}$$

$$\{f\} = [k]\{d\}$$

$$[k] = [T^*]^T [k'][T^*]$$

$$[k] = \begin{bmatrix} C_x & 0 \\ C_y & 0 \\ C_z & 0 \\ 0 & C_x \\ 0 & C_y \\ 0 & C_z \end{bmatrix} \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} C_x & C_y & C_z & 0 & 0 & 0 \\ 0 & 0 & 0 & C_x & C_y & C_z \end{bmatrix}$$

$$[k] = \frac{AE}{L} \begin{bmatrix} C_x^2 & C_xC_y & C_xC_z & -C_x^2 & -C_xC_y & -C_xC_z \\ & C_y^2 & C_yC_z & -C_xC_y & -C_y^2 & -C_yC_z \\ & & C_z^2 & -C_xC_z & -C_yC_z & -C_z^2 \\ & & & C_x^2 & C_xC_y & C_xC_z \\ & & & & C_y^2 & C_yC_z \\ \text{Symmetry} & & & & & C_z^2 \end{bmatrix}$$

$$\mathbf{i}' \cdot \mathbf{i} = \frac{x_2 - x_1}{L} = C_x$$

$$\mathbf{i}' \cdot \mathbf{j} = \frac{y_2 - y_1}{L} = C_y$$

$$\mathbf{i}' \cdot \mathbf{k} = \frac{z_2 - z_1}{L} = C_z$$

Determination of stress in an arbitrarily oriented element-

$$\begin{Bmatrix} f'_{1x} \\ f'_{2x} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u'_1 \\ u'_2 \end{Bmatrix}$$

$$\{\sigma\} = \frac{E}{L}[-1 \quad 1]\{d'\}$$

$$\{\sigma\} = \frac{E}{L}[-1 \quad 1][T^*]\{d\}$$

$$\{\sigma\} = [C']\{d\}$$

$$\{\sigma\} = \frac{E}{L}[-C_x \quad -C_y \quad -C_z \quad C_x \quad C_y \quad C_z] \begin{Bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{Bmatrix}$$

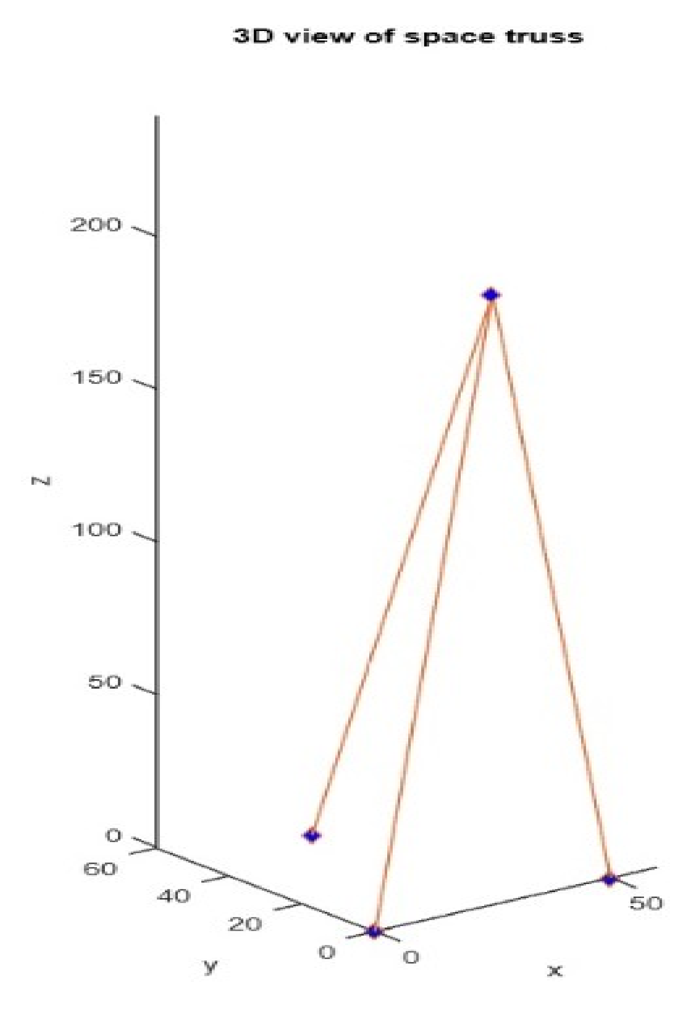
$$\mathbf{i}' \cdot \mathbf{i} = \frac{x_2 - x_1}{L} = C_x$$

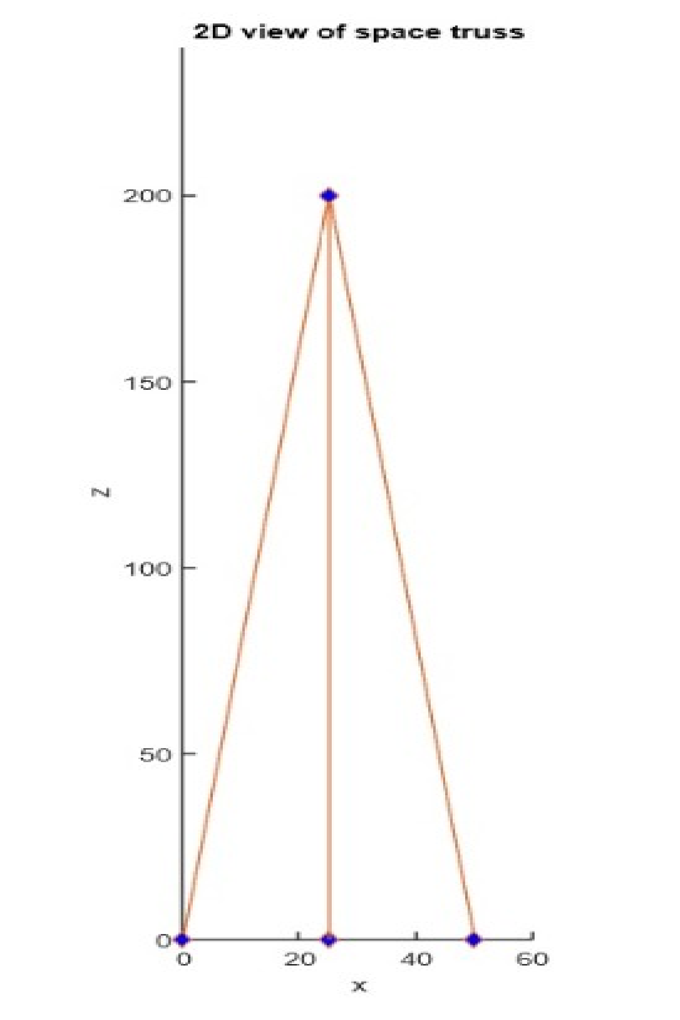
$$\mathbf{i}' \cdot \mathbf{j} = \frac{y_2 - y_1}{L} = C_y$$

$$\mathbf{i}' \cdot \mathbf{k} = \frac{z_2 - z_1}{L} = C_z$$

FINITE ELEMENT METHOD ANALYSIS OF 3D SPACE TRUSS

STEP1 – IDENTIFY SHAPE, ELEMENTS AND NODES





STEP1 – IDENTIFY SHAPE, ELEMENTS AND NODES

- Number of Elements = 3
- Number of Nodes = 4
- Element Connectivity Matrix – [1 2; 1 3; 1 4]
- Node Coordinates – [25 0 200; 25 50 0; 50 0 0; 0 0 0]
- Total Degree of Freedom = DOF per Node * No of Nodes = 3*4 = 12

STEP2 – DESCRIBE BOUNDARY AND FORCE CONDITIONS

- Prescribe Boundary Conditions , i.e. nodes which are fixed. In our case nodes (2,3,4) are fixed.

```
prescribedDof=[4:12]'
```

- Forces Applied at Nodes –

```
force(2)= 100;
force(3)=-100;
```

STEP3 – DETERMINE GLOBAL STIFFNESS MATRIX

MATLAB CODE

- stiffness=zeros(GDof);
- for e=1:numberElements
- indice=elementNodes(e,:) ;
- elementDof=[3*indice(1)-2 3*indice(1)-1 3*indice(1) 3*indice(2)-2 3*indice(2)-1 3*indice(2)] xa=abs(xx(indice(2))-xx(indice(1))); ya=abs(yy(indice(2))-yy(indice(1))); za=abs(zz(indice(2))-zz(indice(1)));
- length_element=sqrt(xa*xa+ya*ya+za*za);
- CX=abs(xa/length_element);
- CY=abs(ya/length_element);
- CZ=abs(za/length_element);
- k1=EA/length_element*[CX*CX CX*CY CX*CZ -CX*CX -CX*CY -CX*CZ; CY*CX CY*CY CY*CZ -CX*CY -CY*CY -CY*CZ; CZ*CX CZ*CY CZ*CZ -CX*CZ -CY*CZ -CZ*CZ; -CX*CX -CX*CY -CX*CZ CX*CX CX*CY CX*CZ; -CX*CY -CY*CY -CY*CZ CX*CY CY*CY CY*CZ; -CX*CZ -CY*CZ -CZ*CZ CX*CZ CY*CZ CZ*CZ];
- stiffness(elementDof,elementDof)= stiffness(elementDof,elementDof)+k1;
- end

STEP4 – DETERMINE NODE DISPLACEMENTS

- After finding Global Stiffness Matrix, put it in equation -

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

- Strike off the rows and columns for the fixed nodes

- Then determine displacements –

$$\{u\} = [K]^{-1}\{F\}$$

MATLAB CODE –

```
-activeDof = setdiff ([1:GDof]',[prescribedDof]) ;      %  get the unique values from both array
-U=pinv(stiffness(activeDof,activeDof))*force(activeDof);
-displacements=zeros(GDof,1);
-displacements(activeDof)=U;
```

STEP5 – DETERMINE ELEMENT STRESSES

- Calculating stresses is pretty much same as calculating stiffness matrix

- Start a loop for each element and determine the indices.

- After that identify the position of displacements in displacement vector using nodes indices.

- Calculate length and cosines directions for the elements using formula –

```
length_element=sqrt(xa*xa+ya*ya+za*za);
CX=abs(xa/length_element);
CY=abs(ya/length_element);
CZ=abs(za/length_element);
```

- Then stresses can be found by using following formula –

$$\{\sigma\} = \frac{E}{L}\begin{bmatrix} -C_x & -C_y & -C_z & C_x & C_y & C_z \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{Bmatrix}$$

MATLAB CODE-

```
indice=elementNodes(e,:) ;
elementDof=[3*indice(1)-2 3*indice(1)-1 3*indice(1) 3*indice(2)-2 3*indice(2)-1
3*indice(2)] ;
xa=abs(xx(indice(2))-xx(indice(1)));
ya=abs(yy(indice(2))-yy(indice(1)));
za=abs(zz(indice(2))-zz(indice(1)));

length_element=sqrt(xa*xa+ya*ya+za*za);
CX=abs(xa/length_element);
CY=abs(ya/length_element);
CZ=abs(za/length_element);

sigma(e)=E/length_element*[-CX -CY -CZ CX CY CZ]*displacements(elementDof);

disp('stresses')
sigma'
end
```

STEP6 – DETERMINE THE REACTION FORCES

- The Nodes which are fixed, will experience Reaction Forces.

- The Forces can be determined by,

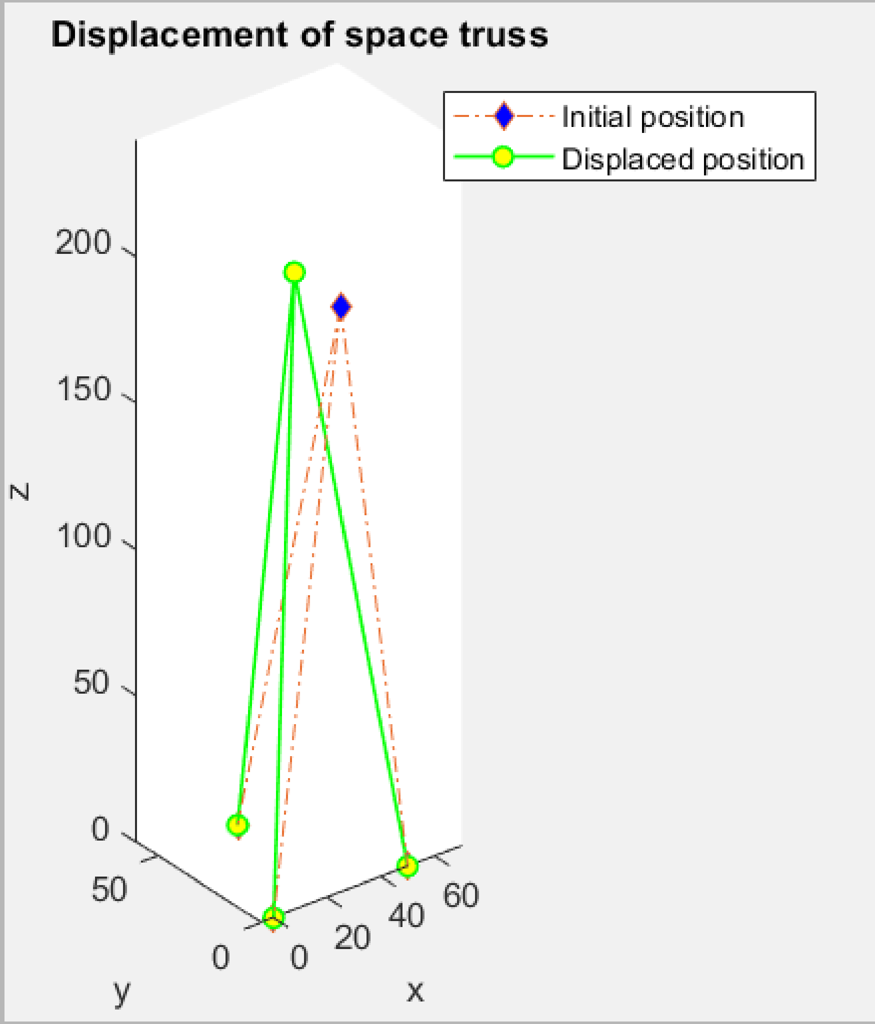
$$\{F\} = [K]\{U\}$$
$$\{R\} = \{F\} \text{ (Prescribed DOF)}$$

- In force vector, components whose position are corresponding to prescribed boundary conditions, are the Reaction Forces.

- **MATLAB CODE-**

```
F=stiffness*displacements;
reactions=F(prescribedDof);
disp('reactions')
[prescribedDof reactions]
```

RESULTS



Displacement of space truss
Initial position
Displaced position
200
150
100
50
0
z
50
0
y
0
20
40
60
x

Stress Plot of space truss
-4123105.6256 Pa
1511673.3278 Pa
1511673.3278 Pa
200
150
100
50
0
Z
50
0
Y
0
20
40
60
X

RESULTS – DISPLACEMENTS OF NODES

Node	Displacement in X Direction	Displacement in Y Direction	Displacement in Z Direction
1	0	0.047443	0.0030860
2	0	0	0
3	0	0	0
4	0	0	0

STRESS IN ELEMENTS

Element	Stress (*10^6)
1	-4.1286
2	1.5191
3	1.5191

REACTIONS AT FIXED NODES

Degree of Freedom	Reaction force (N)
4	0
5	–100.5718
6	402.7312
7	18.5257
8	0
9	–148.2175
10	–18.5257
11	0
12	-148.2175

ANALYTICAL SOLUTION

b + r = 3j……...statically determinate structure
b= no. of members
r= no. of reactions
j= no. of joints

In our case, b=3, r=9, j=4
3 + 9 = 3 × 4
12 = 12……. (statically determinate)

$\varepsilon fz = 0$
$R2\ cos 7.12° + R1\ cos 14.03° + R3 \cos 7.11° = -100 \ldots \ldots \ldots \ldots (1)$

$\varepsilon fx = 0$
$R2\ sin 7.12° = R3\ sin 7.11°$
$R2 = R3$

$from\ equation\ (1)$
$2R2 + R1\ cos 14.03° = -100 \ldots \ldots \ldots \ldots (2)$

ANALYTICAL VS FEM RESULTS

$\varepsilon f y = 0$
$R1\ sin 14.03° = -100$
$R1 = -412.49N$

$from\ equation\ (2)$
$R2 = R3 = 150.09 \cong 151N$

$\sigma 1 =$
-412.49
$0.0001 = -412.49 \times 104Pa$

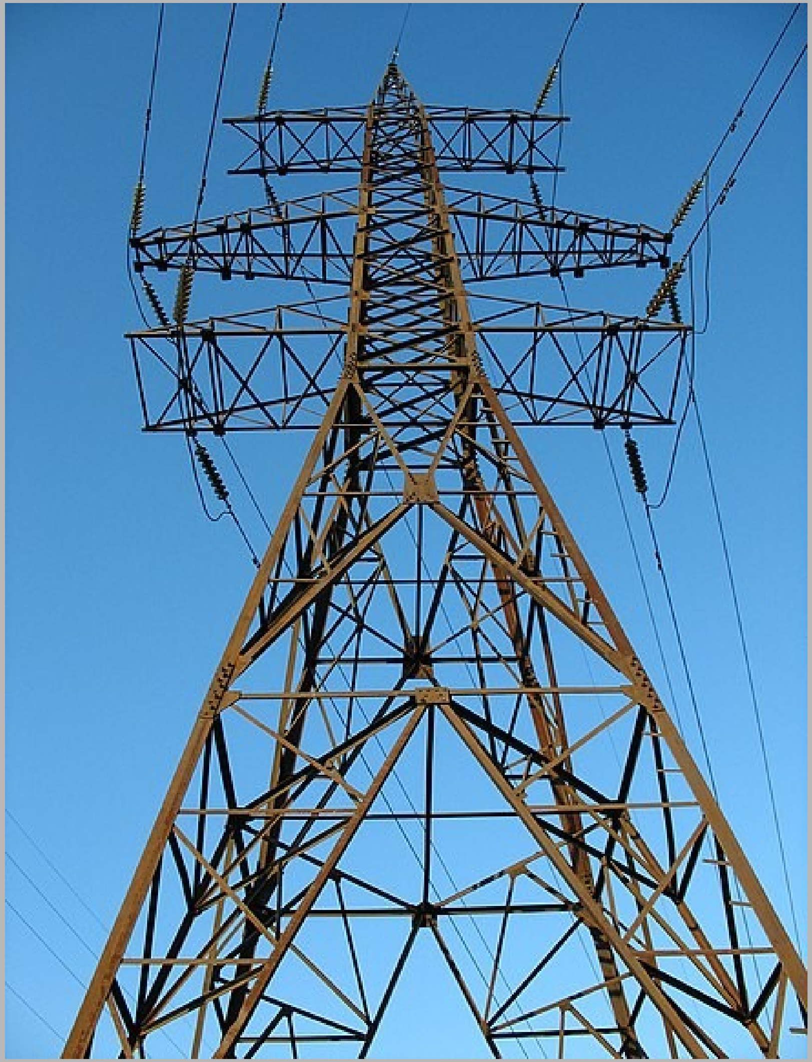
$\sigma 2 =$
151
$0.0001 = 151 \times 104Pa$

$\sigma 3 =$
151
$0.0001 = 151 \times 104Pa$

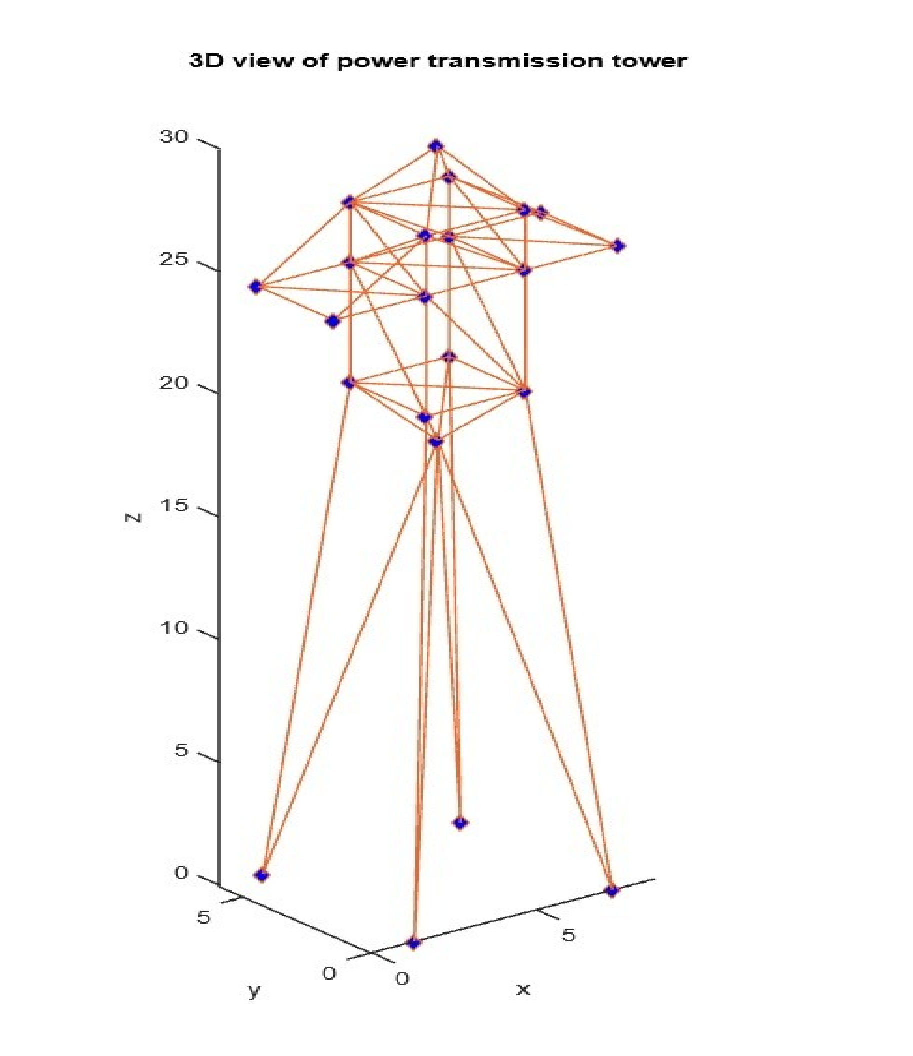
Elements	FEM Solution (Stresses) × [illegible]	Analytical Solution (Stresses) × [illegible]	Percentage Error
1	-412.86	-412.49	0.0896
2	151.91	151	0.602
3	151.91	151	0.602

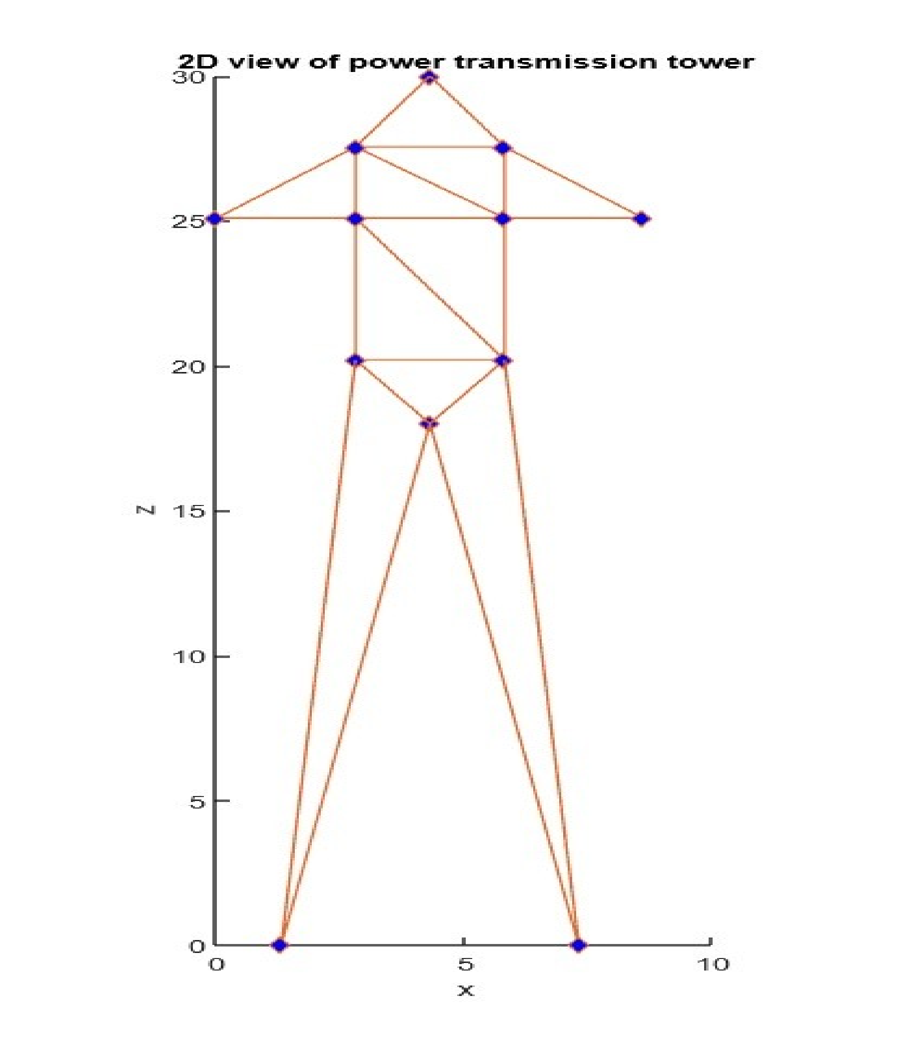
FINITE ELEMENT METHOD ANALYSIS OF POWER TRANSMISSION TOWER

- As you can see from the image, the transmission towers structure consists of steel lattice trusses.
- In general, the tower structure consists of three parts: the crossarms and/or bridges, the peaks, and the tower body.
- The use of truss structures is found advantageous because they are:
  - Easily adaptable to any shape
  - Easily divisible in suitable sections
  - Easy to repair, strengthen and extend.
  - Durable when properly protected against corrosion.
- In most cases a tower with four legs is preferred, as it offers structural advantages and occupies a relatively small ground area.



STEP1 – IDENTIFY SHAPE, ELEMENTS AND NODES





STEP1 – IDENTIFY SHAPE, ELEMENTS AND NODES

- Number of Elements = 59, Number of Nodes = 22

- Element Connectivity Matrix –

Interaction between nodes	Elements	Interaction between nodes	Elements	Interaction between nodes	Elements	Interaction between nodes	Elements
(1,5)	1	(8,12)	16	(19,16)	31	(11,16)	46
(2,6)	2	(13,14)	17	(19,12)	32	(10,13)	47
(3,7)	3	(14,15)	18	(10,20)	33	(8,9)	48
(4,8)	4	(15,16)	19	(14,20)	34	(10,7)	49
(5,6)	5	(13,16)	20	(15,21)	35	(13,12)	50
(6,7)	6	(9,13)	21	(11,21)	36	(14,11)	51
(7,8)	7	(10,14)	22	(18,19)	37	(5,22)	52
(5,8)	8	(11,15)	23	(20,21)	38	(6,22)	53
(9,10)	9	(12,16)	24	(5,7)	39	(7,22)	54
(10,11)	10	(13,17)	25	(9,11)	40	(8,22)	55
(11,12)	11	(14,17)	26	(13,15)	41	(1,22)	56
(9,12)	12	(16,17)	27	(12,18)	42	(2,22)	57
(5,9)	13	(15,17)	28	(10,21)	43	(3,22)	58
(6,10)	14	(9,18)	29	(7,12)	44	(4,22)	59
(7,11)	15	(18,13)	30	(9,16)	45		

STEP1 – IDENTIFY SHAPE, ELEMENTS AND NODES

- Node Coordinates –

Node	x-axis(m)	y-axis(m)	z-axis(m)
1	1.3	6	0
2	7.3	6	0
3	7.3	0	0
4	1.3	0	0
5	2.8	4.5	20.2
6	5.8	4.5	20.2
7	5.8	1.5	20.2
8	2.8	1.5	20.2
9	2.8	4.5	25.1
10	5.8	4.5	25.1
11	5.8	1.5	25.1

Node	x-axis(m)	y-axis(m)	z-axis(m)
12	2.8	1.5	25.1
13	2.8	4.5	27.55
14	5.8	4.5	27.55
15	5.8	1.5	27.55
16	2.8	1.5	27.55
17	4.3	3	30
18	0	4.5	25.1
19	0	1.5	25.1
20	8.6	4.5	25.1
21	8.6	1.5	25.1
22	4.3	3	18

- Total Degree of Freedom = DOF per Node * No of Nodes = 3*22 = 66

- Cross Section Area = $5*10^{-4}$ mm2, Young's Modulus = $2.1*10^{11}$

STEP2 – DESCRIBE BOUNDARY AND FORCE CONDITIONS

Design Wind Speed Vz:

$Vz = Vb \cdot k1 \cdot k2 \cdot k3$

Where,

Vb – Basic wind speed

k1 – Probability factor or Risk Coefficient

k2 – Terrain, height and structure size factor

k3 – Topography factor

Design Wind Pressure:

Pz = 0.6 Vz2 (N/m2)

Design Wind Force:

$F = CfAsPz$

Where,

Cf – Drag coefficient

A

s – Surface area

STEP2 – DESCRIBE BOUNDARY AND FORCE CONDITIONS

k1 = 1.07

Table showing value of k2

Height	k2
18	1.105
20.21	1.12
25.12	1.135
27.56	1.135
30	1.15

$k3 \approx 1$

Vb = 44m/s (Mumbai)

Height	$Vz = Vb \cdot k1 \cdot k2 \cdot k3$ **(m/s)**
18	52.023
20.21	52.73
25.12	53.43
27.56	53.43
30	54.142

Height	**Pz = 0.6 Vz 2 N/m2**
18	1623.83
20.21	1668.27
25.12	1712.85
27.56	1712.85
30	51758.8

STEP2 – DESCRIBE BOUNDARY AND FORCE CONDITIONS

$F = CfAsPz$; $Cf = 3.3$, $As = \pi DL$

At 18m,
$F = 3.3 \times \pi \times 0.025 \times 3.45 \times (1623.83 +2\ 1668.27)$
Therefore, Force at node 22 in y direction, F(65) = 1471.85N

Node	x load (N)	y load (N)	z load (N)
7	0	1744	0
8	0	1744	0
11	0	2382.826	0
12	0	3240	0
15	0	3077.42	0
16	0	2220.8	0
17	0	1457.65	0
19	0	1447.314	0
21	0	1447.314	0
22	0	1471.85	0

STEP2 – DESCRIBE BOUNDARY AND FORCE CONDITIONS

The various forces acting due to self-weight and due to weights of wires and cables is as follows:

Node	x load (N)	y load (N)	z load (N)
22	0	0	-20000
18	0	0	-7000
19	0	0	-7000
20	0	0	-7000
21	0	0	-7000
17	0	0	-7000

STEP2 – DESCRIBE BOUNDARY AND FORCE CONDITIONS

- Prescribe Boundary Conditions , i.e. nodes which are fixed. In our case nodes at the bottom of tower are fixed (1,2,3,4).

  prescribedDof=[1:12]’

- Forces Applied and Wind Forces at Nodes –

Node	x load (N)	y load (N)	z load (N)
7	0	1744	0
8	0	1744	0
11	0	2382.826	0
12	0	3240	0
15	0	3077.42	0
16	0	2220.8	0
17	0	1457.65	-7000
18	0	0	-7000
19	0	1447.314	-7000
20	0	0	-7000
21	0	1447.314	-7000
22	0	1471.85	-20000

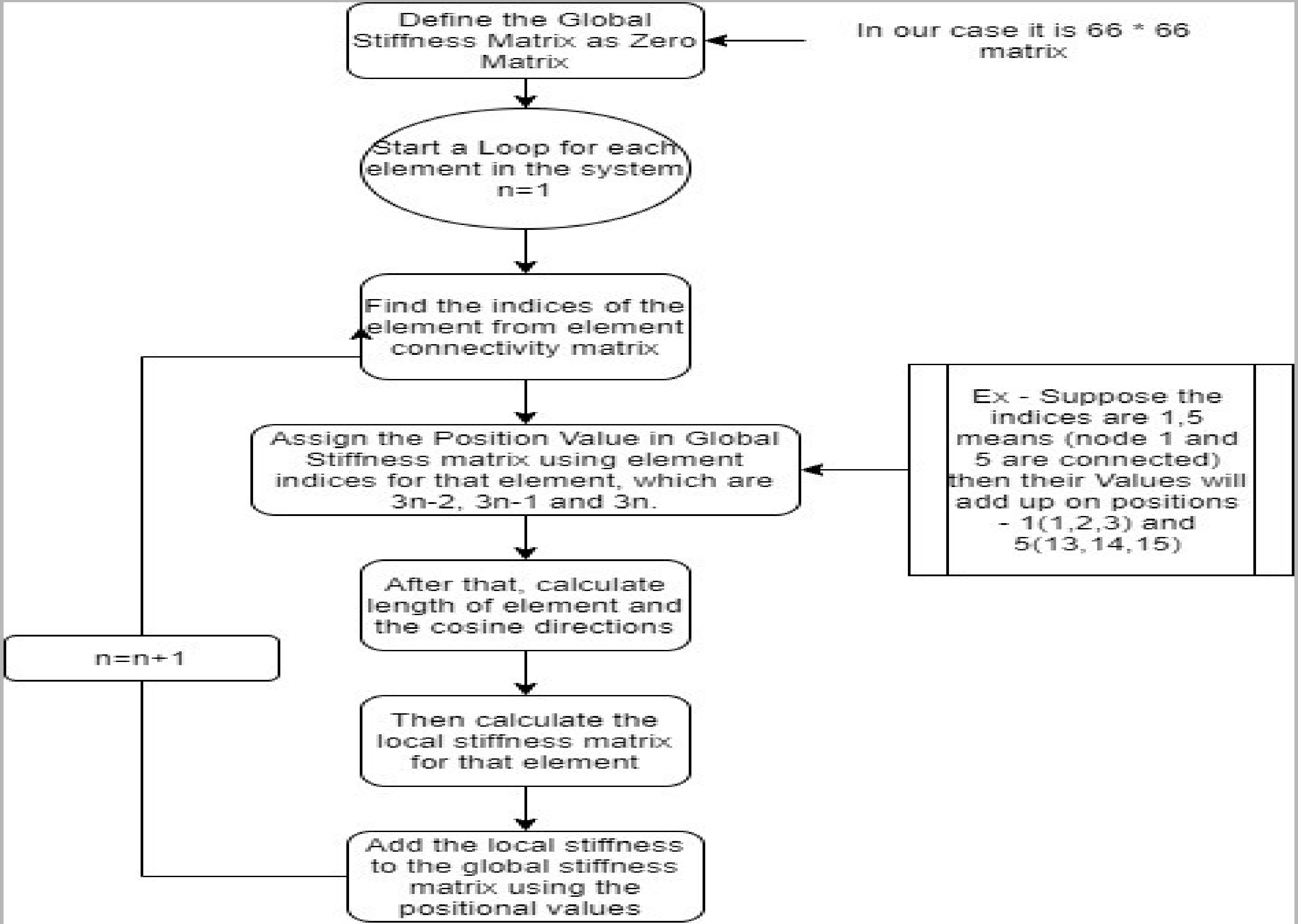
STEP3 – DETERMINE GLOBAL STIFFNESS MATRIX

- The next step is to call the function stiffness. In defining the stiffness function, first we need to initialize the stiffness matrix a zeros matrix with the dimensions of degrees of freedom (GDOF X GDOF). We will use the for loop to get the relation between nodes for the computation of stiffness matrix.

- For a particular node n, the degrees of freedom are given by 3n-2, 3n-1, 3n for x, y and z axes respectively. Then we calculated the length of the element by using coordinates of the nodes. And then from that calculating the cosine values. Using these values we calculate the element stiffness matrix. Then performing the assembly process to obtain global stiffness matrix.

- A Basic Flow Chart for determining the global stiffness matrix is -

STEP3 – DETERMINE GLOBAL STIFFNESS MATRIX



STEP3 – DETERMINE GLOBAL STIFFNESS MATRIX

MATLAB CODE-

- stiffness=zeros(GDof);
- for e=1:numberElements
- indice=elementNodes(e,:) ;
- elementDof=[3*indice(1)-2 3*indice(1)-1 3*indice(1) 3*indice(2)-2 3*indice(2)-1 3*indice(2)] xa=abs(xx(indice(2))-xx(indice(1))); ya=abs(yy(indice(2))-yy(indice(1))); za=abs(zz(indice(2))-zz(indice(1)));
- length_element=sqrt(xa*xa+ya*ya+za*za);
- CX=abs(xa/length_element);
- CY=abs(ya/length_element);
- CZ=abs(za/length_element);
- k1=EA/length_element*[CX*CX CX*CY CX*CZ -CX*CX -CX*CY -CX*CZ; CY*CX CY*CY CY*CZ -CX*CY -CY*CY -CY*CZ; CZ*CX CZ*CY CZ*CZ -CX*CZ -CY*CZ -CZ*CZ; -CX*CX -CX*CY -CX*CZ CX*CX CX*CY CX*CZ; -CX*CY -CY*CY -CY*CZ CX*CY CY*CY CY*CZ; -CX*CZ -CY*CZ -CZ*CZ CX*CZ CY*CZ CZ*CZ];
- stiffness(elementDof,elementDof)= stiffness(elementDof,elementDof)+k1;
- end

STEP4 – DETERMINE NODE DISPLACEMENTS

- After finding Global Stiffness Matrix, put it in equation -

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

- Strike off the rows and columns for the fixed nodes

- Then determine displacements –

$$\{u\} = [K]^{-1}\{F\}$$

MATLAB CODE –

```
-
-activeDof = setdiff ([1:GDof]',[prescribedDof]) ;      %  get the unique values from both array
-U=pinv(stiffness(activeDof,activeDof))*force(activeDof);
-displacements=zeros(GDof,1);
-displacements(activeDof)=U;
```

STEP5 – DETERMINE ELEMENT STRESSES

- Calculating stresses is pretty much same as calculating stiffness matrix
- Start a loop for each element and determine the indices.
- After that identify the position of displacements in displacement vector using nodes indices.
- Calculate length and cosines directions for the elements using formula –

```
length_element=sqrt(xa*xa+ya*ya+za*za);
CX=abs(xa/length_element);
CY=abs(ya/length_element);
CZ=abs(za/length_element);
```

- Then stresses can be found by using following formula –

$$\{\sigma\} = \frac{E}{L}\begin{bmatrix} -C_x & -C_y & -C_z & C_x & C_y & C_z \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{Bmatrix}$$

STEP5 – DETERMINE ELEMENT STRESSES

MATLAB CODE-

```
indice=elementNodes(e,:) ;
elementDof=[3*indice(1)-2 3*indice(1)-1 3*indice(1) 3*indice(2)-2 3*indice(2)-1 3*indice(2)] ;
xa=abs(xx(indice(2))-xx(indice(1)));
ya=abs(yy(indice(2))-yy(indice(1)));
za=abs(zz(indice(2))-zz(indice(1)));

length_element=sqrt(xa*xa+ya*ya+za*za);
CX=abs(xa/length_element);
CY=abs(ya/length_element);
CZ=abs(za/length_element);

sigma(e)=E/length_element*[-CX -CY -CZ CX CY CZ]*displacements(elementDof);

disp('stresses')
sigma'
end
```

STEP6 – DETERMINE THE REACTION FORCES

- The Nodes which are fixed, will experience Reaction Forces.

- The Forces can be determined by,

  {F} = [K]{U}
  {R} = {F} (Prescribed DOF)

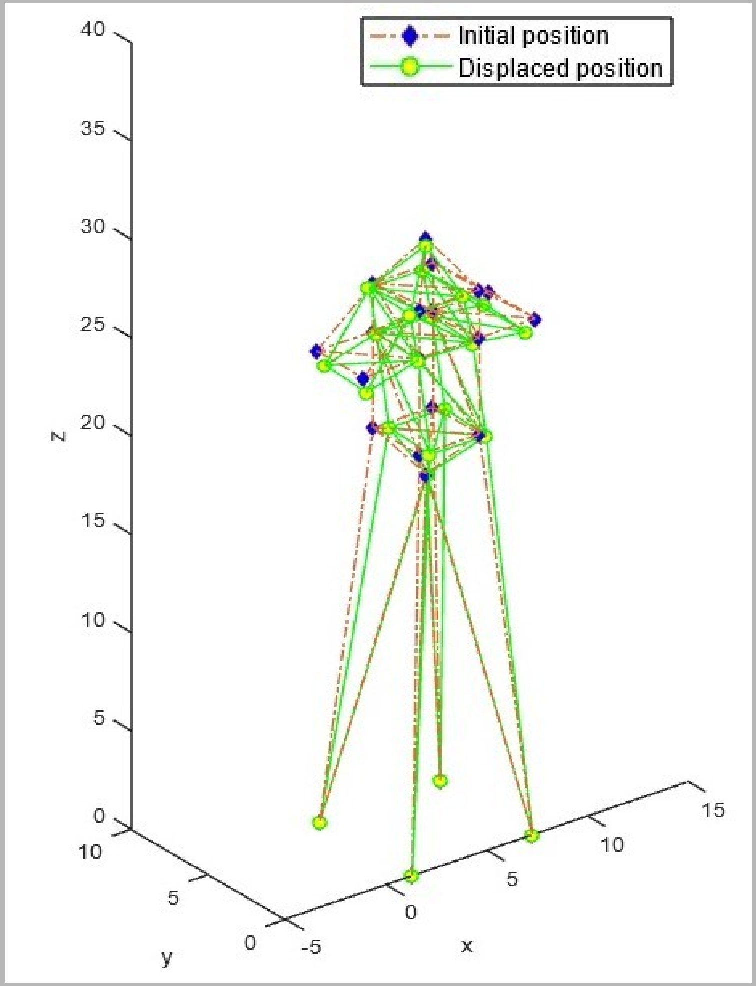
- In force vector, components whose position are corresponding to prescribed boundary conditions, are the Reaction Forces.

- **MATLAB CODE-**

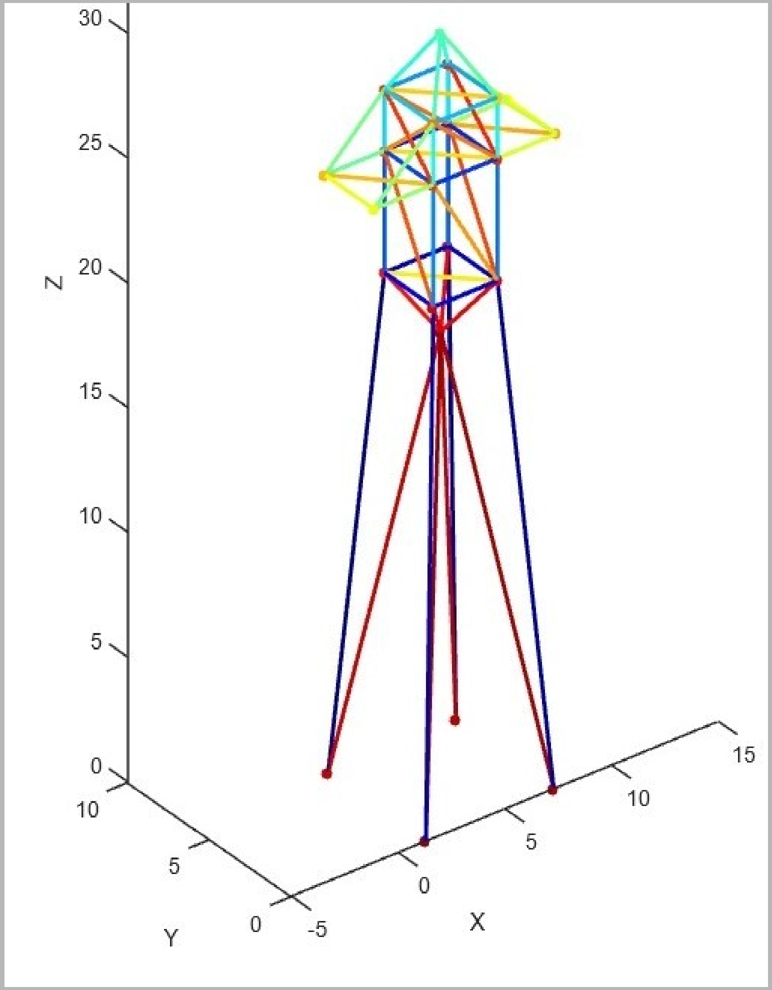
```
F=stiffness*displacements;
reactions=F(prescribedDof);
disp('reactions')
[prescribedDof reactions]
```

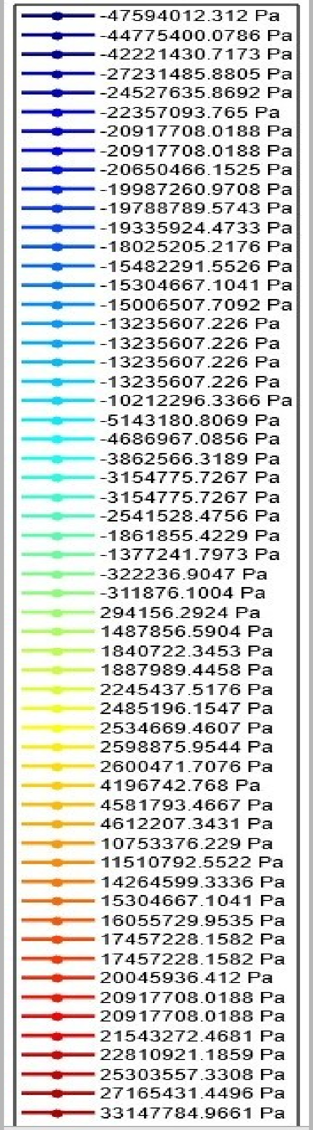
RESULTS

DISPLACEMENT



STRESS





RESULTS – DISPLACEMENTS OF NODES

Node	Displacement in x direction	Displacement in y direction	Displacement in z direction
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0.0001	0.0271	-0.0034
6	0	0.0269	-0.0033
7	0	0.027	0.0006
8	-0.0001	0.0272	0.0007
9	-0.0006	0.0371	-0.0045
10	-0.0008	0.037	-0.0045
11	-0.0006	0.0372	0.0004
12	-0.0004	0.0372	0.0004
13	-0.001	0.0413	-0.0047
14	-0.0007	0.0416	-0.0048
15	-0.0005	0.0417	0.0002
16	-0.0007	0.0413	0.0003
17	-0.0007	0.0456	-0.0024
18	-0.0003	0.0374	-0.006
19	-0.0002	0.0374	-0.0009
20	-0.001	0.0372	-0.0057
21	-0.0007	0.0372	-0.0007
22	0	0.0237	-0.0013

STRESS IN ELEMENTS

Element	Stress (*10^7)	Element	Stress (*10^7)	Element	Stress (*10^7)	Element	Stress (*10^7)
1	-5.5121	17	1.8023	33	-1.6	49	2.9252
2	-5.4967	18	-0.6254	34	2.126	50	-0.301
3	2.7242	19	1.5901	35	2.126	51	1.0687
4	2.7395	20	0.3691	36	-1.3298	52	0.8417
5	-0.7544	21	-1.7632	37	-0.2895	53	0.7101
6	-0.7544	22	-2.4065	38	0	54	-1.7417
7	1.1218	23	-1.7695	39	-0.093	55	-1.8733
8	-0.7545	24	-0.9284	40	-0.0939	56	-5.7707
9	-1.8038	25	-0.8039	41	0.334	57	-5.7707
10	-1.238	26	-0.4371	42	0.3959	58	2.9529
11	-1.3298	27	-0.122	43	-0.396	59	2.9529
12	-0.7043	28	-0.4888	44	0		
13	-4.876	29	-1.8702	45	-0.977		
14	-4.9555	30	2.126	46	0.0857		
15	-1.0393	31	2.126	47	-0.0857		
16	-1.1188	32	-1.6	48	2.9253		

REACTIONS AT FIXED NODES

Degree of Freedom	Reaction force (N)
1	0.6716
2	-0.6716
3	5.5494
4	-0.671
5	-0.671
6	5.5417
7	0.3401
8	-0.3401
9	-2.7917
10	-0.3407
11	-0.3407
12	-2.7994

Thank You