Name:	Ravi Chaudhary
SID:	**10947986**
Title:	**Finite Element Analysis of Truss**
Submission Date:	**29/11/2021**

Abstract:

In this analysis, theoretical, mathematical, and numerical methods were carried-out to analyse the truss structure acting under external force applied in node A i.e., 15KN. The theoretical method was carried-out to calculate the forces reacting at the supports and the internal forces in each element of the truss applying equation of static equilibrium. Similarly, the mathematical method was proceeded to calculate the nodal force and displacements using MATLAB calculating stiffness matrix of each element and combining as global stiffness matrix. Eventually, the numerical method was generated using Abacus with the purpose to find-out the value of nodal forces, displacements, support reactions and stress in each element and were compared to the values obtained theoretically and mathematically. Comparing the results obtained, accuracy and easy approach to solve complex Finite Element Analysis (FEA) problem, numerical method was found to be more appropriate.

Introduction:

The main objective of this analysis was to understand the concept of the truss structure along with the aim to be familiar with FEA related software's like Abacus and Ansys.

Finite Element Analysis (FEA) is broadly used to identify the truss structural issues, it's assembly, weak points, tension and compression. Truss consists of different members organized in a triangular manner where the members in truss are either in tension or compression. It is light, cost-effective and can be quickly installed and has great significance and uses in real world scenario. In this report, truss shown in **Fig 1** was completely studied and its behaviour, forces, displacements and stresses were analysed.

As according to the given load and mechanical properties of the truss it could be hypothesized that the nodal displacement as well as overall displacement of the truss would be very small. It was believed that the values obtained theoretically and mathematically would be close to the numerical values obtained using Abacus software.

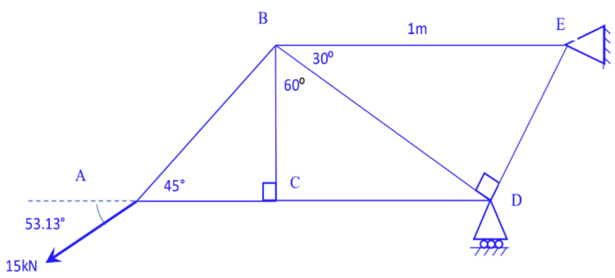


Fig 1: Truss Structure **(Coventry 2021)**

Table 1: Notations for nodes and elements

Nodes	Elements	Numbered elements
A (1)	AB	1
B (2)	BC	2
C (3)	CD	3
D (4)	DE	4
E (5)	AC	5
	BD	6
	BE	7

Theoretical calculations:

The theoretical calculation of the given truss was based on the concept of static equilibrium equation, applying the concept, the internal forces acting on each element as well as reaction forces at the supports can be calculated. It includes method of joints where each members of the structure are separated from joints and equilibrium equation is applied to determine the axial forces of each members. It allows to understand the truss acting under different load conditions. The truss frame shown in **figure 1**, consists roller support at **node 'D'** and pin support at **node 'E'**. Here, the roller support (i.e., node D) allows the movement in only X direction and has the force in only Y direction whereas the pin support (i.e., node E) constrains the structure from movement in both X and Y direction as it has both X and Y direction forces. The calculation was proceeded isolating each singular node and applying equilibrium condition in both X and Y direction where the sum of forces in both said directions were zero. There are no any moment forces in trusses as they are connected with pins and can be only subjected to be either in tension or compression forces, excluding shear force **(Alderliesten, 2021)**.

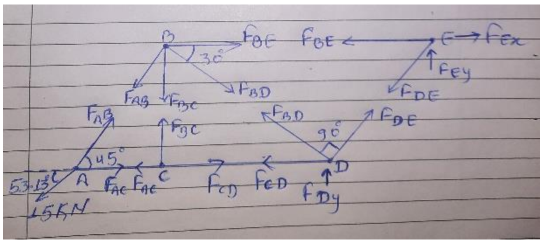


Fig 2: Truss structure with assumed force direction

Equilibrium Equation: $\Sigma F_x = 0 \; and \; \Sigma F_y = 0$

At node A

- $F_{AB} * \sin(45^{\circ}) - 15000 * \sin(53.13^{\circ}) = 0$
  ⇨ $\boldsymbol{F_{AB} = 16901.39\ N}$
- $F_{AB} * \cos(45^{\circ}) + F_{AC} - 15000 * \sin(53.13^{\circ}) = 0$
  ⇨ $\boldsymbol{F_{AC} = -2951.07\ N}$

At node C

- $F_{BC} = 0\ N$
- $(-F_{AC}) + F_{CD} = 0$
  ⇨ $\boldsymbol{F_{CD} = -2951.07\ N}$

Table 2: Support reaction forces at x and y direction

Supports	Reaction force (x-direction) N	Reaction force (Y-direction) N
Roller at node 'D'	0	53465.98
Pin at node 'E'	8953.53	-41466.0009

At node B

- $(-F_{AB}) * \sin(45^{\circ}) - F_{BC} - F_{BD} * \sin(30^{\circ}) = 0$
  ⇨ $\boldsymbol{F_{BD} = -23999.96\ N}$
- $(-F_{AB}) * \cos(45^{\circ}) + F_{BE} + F_{BD} * \cos(30^{\circ}) = 0$
  ⇨ $\boldsymbol{F_{BE} = 32784.56\ N}$

At node D

- $F_{DE} * \cos(60^{\circ}) - F_{BD} * \cos(30^{\circ}) - F_{CD} = 0$
  ⇨ $\boldsymbol{F_{DE} = -47662.07\ N}$
- $F_{DE} * \sin(60^{\circ}) + F_{BD} * \sin(30^{\circ}) + F_D y = 0$
  ⇨ $\boldsymbol{F_D y = 53465.98\ N}$

At node E

- $(-F_{BE}) - (F_{DE}) * \cos(60^{\circ}) + F_E x = 0$
  ⇨ $\boldsymbol{F_E x = 8953.53\ N}$
- $(-F_{DE}) * \sin(60^{\circ}) + F_E y = 0$
  ⇨ $\boldsymbol{F_E y = -41466.0009\ N}$

Table 3: Internal forces on each element and types of forces

Elements	Internal Forces (N)	Force type
AB	16901.39	Tension
BC	0	No force
CD	-2951.07	Compression
DE	-47662.07	Compression
AC	-2951.07	Compression
BD	-23999.96	Compression
BE	32784.56	Tension

Mathematical calculation:

Finite element method approaches the problem of stress calculation in the body with one or more than one dimension by splitting the body into finite elements that are connected together at nodes and the process is called discretization and the collection of nodes and elements is called mesh. In stress analysis problem, displacement is the fundamental variable needed to calculate at each node of a truss, stress can be simply calculated knowing how body displaces after force applied using mathematical method. Although the majority of problems can be solved analytically using partial differentiation but solving the complex problem using PDE could increase difficulties resulting in time consumption and unavoidable accuracy. Therefore, Finite Element Method (FEM) comes in practice and typical finite element mesh with more than hundred thousand degrees of freedom can be easily solved using software's like MATLAB, Excel, Ansys, Abacus etc. In case of truss structure analysis shown in **fig. 1**, mathematical model was developed using formula **F=K*u,** F is nodal forces; 'k' is element stiffness matrix and 'u' is nodal displacement. Here, the element stiffness matrix (k) defines the displacement of node in an element for a set of forces and moments applied to a node. To calculate displacement in each node, external loads need to be defined and boundary condition should be specified. Thus, the behaviour of singular element as well as the overall system can be understood with above mentioned analysis **(Öchsner & Makvandi, 2019).**

- The illustrated truss structure in fig. 1 was in 2-dimensional space i.e., the vectors, were in 2-D Cartesian plane. Each of the elements consist of 2-nodes i.e., x and y component at each node and angle of displacement along x-axis. The stiffness matrix for each element can be calculated using given values of area of cross-section (1.26*10^-3 m), young's modulus (260GPa) and the length of element in the formula $\frac{\boldsymbol{A*E}}{\boldsymbol{L}}$.

- Each individual local stiffness matrices need to be assembled into a Global Stiffness Matrix (GSM). Global stiffness matrix is a square matrix arranged in a row and columns which is equal to total number of degrees of freedom in a model **(calculation referred to appendix)**. The 10╳10 global stiffness matrix can be obtained as each node of truss had 2 degrees of freedom.
- Applying the boundary condition, GSM can be reduced eliminating the corresponding rows and columns of the nodal support forces as there is no moment. Displacement can be calculated either by inverting the achieved matrix or by conjugate gradient method. After resolving nodal displacement, stress and strain can be calculated throughout the mesh.
- Finally, the nodal forces in x and y direction can be calculated replacing the values of obtained displacements in the equation obtained from reduced global stiffness matrix **(De, 2021)**.

The results obtained using GSM was presented in the table below:

k=															
1.0e+08 *			1	2	3	4	5	6	7	8	9	10			
	-9000	1	10.2764	2.6781	-2.6781	-2.6781	-7.5983	0	0	0	0	0		u1x	u1x
	-12000	2	2.6781	2.6781	-2.6781	-2.6781	0	0	0	0	0	0		u1y	u1y
	F2x	3	-2.6781	-2.6781	8.762	1.0519	0	0	-2.8166	1.6262	-3.2673	0		u2x	u2x
	F2y	4	-2.6781	-2.6781	1.0519	11.2147	0	-7.5983	1.6262	-0.9389	0	0		u2y	u2y
	F3x	5	-7.5983	0	0	0	11.9546	0	-4.3563	0	0	0		u3x	u3x
	F3y	6	0	0	0	-7.5983	0	7.5983	0	0	0	0		u3y	u3y
	0	7	0	0	-2.8166	1.6262	-4.3563	0	8.8065	1.2033	-1.6336	-2.8295		u4x	u4x
	F4y	8	0	0	1.6262	-0.9389	0	0	1.2033	5.8398	-2.8295	-4.9009		u4y	0
	F5x	9	0	0	-3.2673	0	0	0	-1.6336	-2.8295	4.9009	2.8295		u5x	0
	F5y	10	0	0	0	0	0	0	-2.8295	-4.9009	2.8295	4.9009		u5y	0

Fig 3: 10╳10 Global stiffness matrix with representation of reduced matrix using pale grey colour **(Nm)**.

Table 4: Nodal displacements and forces in x and y direction.

Nodes	Nodal Displacement (X-direction) [1*10e-4]{m}	Nodal Displacement (Y-direction) [1*10e-4]{m}	Nodal Force (X-direction) [N]	Nodal Force (Y-direction) [N]
1.	1.5679	-8.5717	-8996.6	-11998
2.	-1.0052	-5.5506	-1.8946	-2.4404
3.	1.5284	-5.5506	0.1622	0
4.	1.4595	0	-2.2157	53330 (RF)
5.	0	0	9000.5 (RF)	-41297 (RF)

Numerical calculations:

FEA is the study and framework of the outcomes in the virtual environment which can predict the possible failure of the structures and can be used to solve the existing structures understanding their complexity. The numerical calculation of the finite element can be done using the FEA software's such as Abacus and Ansys through which immensely precise value can be obtained. It is widely used in the engineering world as it is capable of performing different types of analysis related to structural (static and dynamic), thermal, fluids, coupled-field etc. It automatically generates the required results according to given input and also represents it visually. The numerical analysis for the given truss (Fig. 1) was performed using Abacus with the following procedure explained below.

- The module needs to be defined before beginning the part design of the truss and selecting the part allows to choose modelling space **(2D selected)**, type and shape of the structure.
- After designing the structure using construction lines and angles, the mechanical and geometrical properties was defined in the property section. Where, cross-sectional area **(1.27*10^-3)** and Poisson ratio **(0.3)** were filled with the mentioned values.
- The design was constrained as associated system by switching it to assembly and proceeded to step section.

- In the step section, general static step was selected as a type of analysis to be performed. In case of numerous interacting parts, section-interaction needs to be defined.
- Switching to the load section allows to set the boundary conditions and define the loads acting on the truss. The **15KN** load at an angle **15.13**$^{\circ}$ was solved in x and y direction and was applied to **Node A (1)**. The boundary conditions were defined in roller and pin at **nodes D (4**) and **E (5)** respectively and their movements were restricted accordingly.
- The mesh reaction was selected afterwards and estimate global size was defined in order to discretise the elements of the structure.
- Ultimately, the job-section was proceeded and by creating a new job, it was submitted and the produced result could be seen in the visualization section. The required results i.e., nodal displacement and forces, support reaction, forces and stresses in all elements can be obtained **(Lemanski, 2021).**

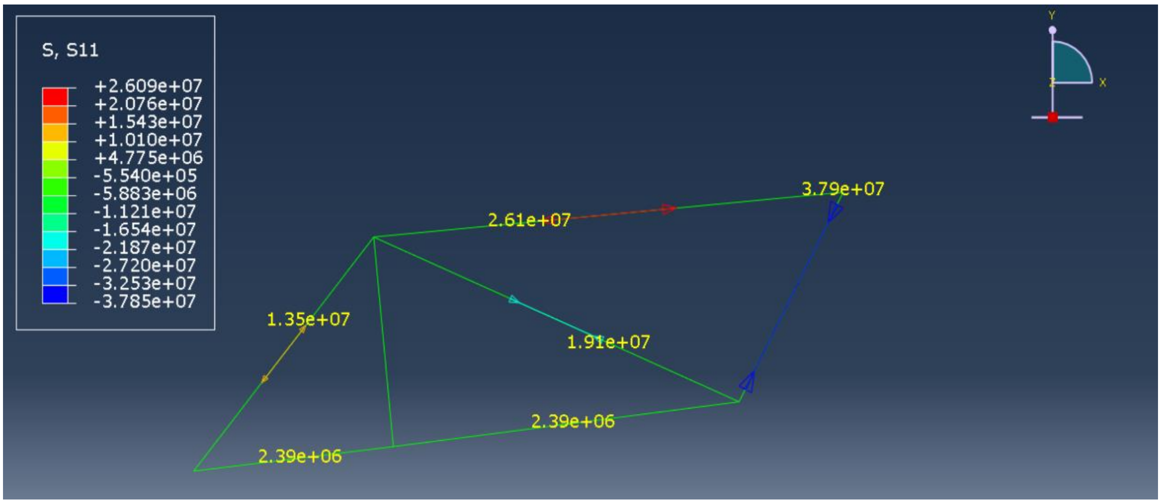


Fig 4: Abacus visualization showing stresses on each element deformation of truss

Table 5: Nodal displacements and forces obtained from Abacus

Nodes	Nodal Displacement (X-direction) [m]	Nodal Displacement (Y-direction) [m]	Nodal reaction Force (X-direction) [N]	Nodal Reaction Force (Y-direction) [N]
1.	1.56456e-04	-8.54984e-04	-9000.02	-11999.98
2.	-1.00343e-04	-5.53203e-04	0	0
3.	1.5248e-04	-5.53203e-04	0	0
4.	1.45593e-04	-5.31962e-32	0	53196.2 (RF)
5.	-9e-33	4.11962e-32	9000 (RF)	-41196.2 (RF)

Table 6: Stress and internal forces on elements

Elements	Stresses (1*10^6) $[N/m^2]$	Internal Forces (N)
AB (1)	13.5	16964.6
BC (2)	0	0
CD (3)	-2.39	-3003.4
DE (4)	-37.9	-47626.6
AC (5)	-2.39	-3003.4
BD (6)	-19.1	-24001.8
BE (7)	26.1	32798.2

Analysis and Discussion:

The forces, stresses and displacements of each element and node were calculated using different approaches i.e., theoretical, mathematical and numerical calculation. The reaction forces at supports calculated theoretically **(ref Table 2)** were close to the reaction forces calculated using mathematical method **(ref Table 4)** and also support reaction forces obtained mathematically were close to support reaction values obtained numerically **(ref Table 5)** but slightly higher difference can be seen in reaction forces value at node comparing theoretical calculation **(ref Table 2)** with numerical method **(ref Table 5)**. This might cause due to the process of rounding off the significant figure, minute error caused during hand calculation, though the result obtained was close and was acceptable. Considering the nodal displacements found using Abacus **(Ref table 5)** was approximately similar with small decimal figure difference compared to mathematical model as the software's has ability to automatically output the results unlike hand calculation. The similar case could be seen in nodal forces **(Ref table 4 and 5)** calculated using MATLAB and Abacus. Exact precision cannot be obtained as every software's perform calculation differently. In comparison, the abacus software surpassed to find the predicted values calculated using static equilibrium equation. Eventually, the internal forces obtained theoretically and using Abacus **(Ref table 3 and 6)** were slightly different that might cause due to rounding off inaccuracy due to hand calculation. However, the values obtained theoretically, mathematically and numerically were very close and were reasonable to the assumption as said in the introduction. Both the results obtained using MATLAB and Abacus were equally acceptable. The only difference is that MATLAB produces results according to the inputs provided in each step whereas the Abacus automatically generates the computerized results and is very efficient. Also, the theoretical approach is equivalently important to previously understand how the structure would behave and to predict the possible outcomes using different software's.

Both the theoretical and numerical method clearly showed that the stress **at element (BC)** was **0** which concludes that the element **BC** neither experienced tension nor compression force. Although it had no important function in the structure, it could be used as a support due to buckling. The element **BC** can be ignored if the structure has to be re-designed or can be set to different angle or position to improve overall stress distribution of the structure.

Similarly, safety of the overall structure is also significant. The factor of safety (FoS) of the analysed truss could be calculated taking the most stressed element of the structure i.e., **element 7 (BE)**, with the maximum stress value 26.09 MPa **(ref Fig. 3)**. Dividing max. stress with young's modulus of the material (250 MPa), FoS can be calculated and it was found to be 9.58 for the given truss structure. Whether the FoS would be 1 or less than 1, the given truss structure would plastically deform before failure.

<u>Conclusion</u>:

In conclusion, the purpose of the following case study was to determine the results theoretically, mathematically and to compare the values with numerically obtained results. It can be concluded that the Abacus is a reasonable software for the finite element analysis as it is easy to use, outputs precise results and has no mathematical procedure. The mathematical method is also a sensible approach to carry-out finite element analysis but due to the structure complexity, numerous inputs and due to rounding off nature of MATLAB could consequence in small imprecisions. The theoretical method is important to initially understand the structure and behaviour which can be later on compared to mathematical or numerical method.

To achieve more accurate results in any analysis of FEA, the mathematical calculation should be done using as less inputs as possible to obtain the certain results which can reduce the rounding off imprecision. Similarly, while carrying out theoretical procedure, including as more significant figures in the calculations can result in more closure values. The following analysis of the truss structure develops the theoretical and basic knowledge on the use of FEA software which can be very helpful during placement in industries and even in practical world scenario.

<u>References</u>

Alderliesten, R. (2021, April 15). *Methods of truss analysis*. Retrieved from Engineering LIbreTexts: https://eng.libretexts.org/Bookshelves/Mechanical_Engineering/Introduction_to_Aerospace_Structures_and_Materials_(Alderliesten)/02%3A_Analysis_of_Statically_Determinate_Structures/05%3A_Internal_Forces_in_Plane_Trusses/5.06%3A_Methods_of_Truss_Analysis

Öchsner, A., & Makvandi, R. (2019). *Finite Element for Truss and Frame Structures.* Gewerbestrasse, switzerland: Springer International Publishing.

De, P. S. (2021). *Development of Truss Equations.* Coventry University.

Lemanski, D. S. (2021). *Practical guide to FEA.* Coventry University.

MATLAB R2020a. (2020). Retrieved from Mathworks.

Abacus/CAE. (2017). Retrieved from Abacus FEM.

Appendix:

MATLAB script used as mathematical model

Script to find local matrix for element 1 (AB)

```
% solving for element 1 (AB)
A = 1.256637061*10^-3;   % area of the structure
E = 260*10^9;            % young's modulus
L = 0.61;                % length of element 1
a_1 = 45;                % angle of element 1
a_1 = deg2rad(a_1);
a = sin(a_1);
b = cos(a_1);
k = [b^2 a*b -b^2 -a*b; a*b a^2 -a*b -a^2; -b^2 -a*b b^2 a*b; -a*b -a^2 a*b a^2];
y_1 = E*A/L;
stiffness_1 = y_1.*k
```

Local stiffness matrix for element 1 (AB) in (Nm)

```
stiffness_1 =

   1.0e+08 *

    2.6781    2.6781   -2.6781   -2.6781
    2.6781    2.6781   -2.6781   -2.6781
   -2.6781   -2.6781    2.6781    2.6781
   -2.6781   -2.6781    2.6781    2.6781
```

Script to find local matrix for element 2 (BC)

```
% solving for element 2 (BC)
A = 1.256637061*10^-3;   % area of the structure
E = 260*10^9;            % young's modulus
L = 0.43;                % length of element 2
a_2 = 90;                % angle of element 2
a_2 = deg2rad(a_2);
a = sin(a_2);
b = cos(a_2);
k = [b^2 a*b -b^2 -a*b; a*b a^2 -a*b -a^2; -b^2 -a*b b^2 a*b; -a*b -a^2 a*b a^2];
y_2 = E*A/L;
stiffness_2 = y_2.*k
```

Local stiffness matrix for element 2 (BC) in (Nm)

```
stiffness_2 =

   1.0e+08 *

    0.0000    0.0000   -0.0000   -0.0000
    0.0000    7.5983   -0.0000   -7.5983
   -0.0000   -0.0000    0.0000    0.0000
   -0.0000   -7.5983    0.0000    7.5983
```

Script to find local matrix for element 3 (CD)

```
% solving for element 3 (CD)
A = 1.256637061*10^-3;  % area of the structure
E = 260*10^9;           % young's modulus
L = 0.75;               % length of element 3
a_3 = 0;                % angle of element 3
a_3 = deg2rad(a_3);
a = sin(a_3);
b = cos(a_3);
k = [b^2 a*b -b^2 -a*b; a*b a^2 -a*b -a^2; -b^2 -a*b b^2 a*b; -a*b -a^2 a*b a^2];
y_3 = E*A/L;
stiffness_3 = y_3.*k
```

Local stiffness matrix for element 3 (CD) in (Nm)

```
stiffness_3 =

   1.0e+08 *

    4.3563         0   -4.3563         0
         0         0         0         0
   -4.3563         0    4.3563         0
         0         0         0         0
```

Script to find local matrix for element 4 (DE)

```
% solving for element 4 (DE)
A = 1.256637061*10^-3;  % area of the structure
E = 260*10^9;           % young's modulus
L = 0.5;                % length of element 4
a_4 = 60;               % angle of element 4
a_4 = deg2rad(a_4);
a = sin(a_4);
b = cos(a_4);
k = [b^2 a*b -b^2 -a*b; a*b a^2 -a*b -a^2; -b^2 -a*b b^2 a*b; -a*b -a^2 a*b a^2];
y_4 = E*A/L;
stiffness_4 = y_4.*k
```

Local stiffness matrix for element 4 (DE) in (Nm)

```
stiffness_4 =

   1.0e+08 *

    1.6336    2.8295   -1.6336   -2.8295
    2.8295    4.9009   -2.8295   -4.9009
   -1.6336   -2.8295    1.6336    2.8295
   -2.8295   -4.9009    2.8295    4.9009
```

Script to find local matrix for element 5 (AC)

```
% solving for element 5 (AC)
A = 1.256637061*10^-3;  % area of the structure
E = 260*10^9;           % young's modulus
L = 0.43;               % length of element 5
a_5 = 0;                % angle of element 5
a_5 = deg2rad(a_5);
```

```
a = sin(a_5);
b = cos(a_5);
k = [b^2 a*b -b^2 -a*b; a*b a^2 -a*b -a^2; -b^2 -a*b b^2 a*b; -a*b -a^2 a*b a^2];
y_5 = E*A/L;
stiffness_5 = y_5.*k
```

Local stiffness matrix for element 5 (AC) in (Nm)

```
stiffness_5 =

1.0e+08 *

7.5983         0   -7.5983         0

         0         0         0         0

   -7.5983         0    7.5983         0

         0         0         0         0
```

Script to find local matrix for element 6 (BD)

```
% solving for element 6 (BD)
A = 1.256637061*10^-3;   % area of the structure
E = 260*10^9;            % young's modulus
L = 0.87;                % length of element 6
a_6 = 150;               % angle of element 6
a_6 = deg2rad(a_6);
a = sin(a_6);
b = cos(a_6);
k = [b^2 a*b -b^2 -a*b; a*b a^2 -a*b -a^2; -b^2 -a*b b^2 a*b; -a*b -a^2 a*b a^2];
y_6 = E*A/L;
stiffness_6 = y_6.*k
```

Local stiffness matrix for element 6 (BD) in (Nm)

```
stiffness_6 =

1.0e+08 *

    2.8166   -1.6262   -2.8166    1.6262

   -1.6262    0.9389    1.6262   -0.9389

   -2.8166    1.6262    2.8166   -1.6262

    1.6262   -0.9389   -1.6262    0.9389
```

Script to find local matrix for element 7 (BE)

```
% solving for element 7 (BE)
A = 1.256637061*10^-3;   % area of the structure
E = 260*10^9;            % young's modulus
L = 1;                   % length of element 6
a_7 = 0;                 % angle of element 5
a_7 = deg2rad(a_7);
```

```
a = sin(a_7);
b = cos(a_7);
k = [b^2 a*b -b^2 -a*b; a*b a^2 -a*b -a^2; -b^2 -a*b b^2 a*b; -a*b -a^2 a*b a^2];
y_7 = E*A/L;
stiffness_7 = y_7.*k
```

<u>Local stiffness matrix for element 7 (BE) in (Nm)</u>

```
stiffness_7 =

1.0e+08 *

  3.2673        0   -3.2673        0

       0        0         0        0

 -3.2673        0    3.2673        0

       0        0         0        0
```

<u>Script of Global Stiffness Matrix</u>:

```
K =

    [x1x1,y1x1,x2x1,y2x1,x3x1,y3x1,x4x1,y4x1,x5y1,y5x1;
    x1y1,y1y1,x2y1,y2y1,x3y1,y3y1,x4y1,y4y1,x5y1,y5y1;
    x1x2,y1x2,x2x2,y2x2,x3x2,y3x2,x4x2,y4x2,x5x2,y5x2;
    x1y2,y1y2,x2y2,y2y2,x3y2,y3y2,x4y2,y4y2,x5y2,y5y2;
    x1x3,y1x3,x2x3,y2x3,x3x3,y3x3,x4x3,y4x3,x5x3,y5x3;
    x1y3,y1y3,x2y3,y2y3,x3y3,y3y3,x4y3,y4y3,x5y3,y5y3;
    x1x4,y1x4,x2x4,y2x4,x3x4,y3x4,x4x4,y4x4,x5x4,y5x4;
    x1y4,y1y4,x2y4,y2y4,x3y4,y3y4,x4y4,y4y4,x5y4,y5y4;
    x1x5,y1x5,x2x5,y2x5,x3x5,y3x5,x4x5,y4x5,x5x5,y5x5;
    x1y5,y1y5,x2y5,y2y5,x3y5,y3y5,x4y5,y4y5,x5y5,y5y5]
```

	x1	y1	x2	y2	x3	y3	x4	y4	x5	y5
x1	x1x1+x1x1	y1x1+y1x1	x2x1	y2x1	x3x1	0	0	0	0	0
y1	x1y1+x1y1	y1y1+y1y1	x2y1	y2y1	0	0	0	0	0	0
x2	x1x2	y1x2	x2x2+x2x2+x2x2+x2x2	y2x2+y2x2+y2x2+y2x2	0	0	x4x2	y4x2	x5x2	0
y2	x1y2	y1y2	x2y2+x2y2+x2y2+x2y2	y2y2+y2y2+y2y2+y2y2	0	y3y2	x4y2	y4y2	0	0
x3	x1x3	0	0	0	x3x3+x3x3+x3x3	y3x3+y3x3+y3x3	x4x3	0	0	0
y3	0	0	0	y2y3	x3y3+x3y3+x3y3	y3y3+y3y3+y3y3	0	0	0	0
x4	0	0	x2x4	y2x4	x3x4	0	x4x4+x4x4+x4x4	y4x4+y4x4+y4x4	x5x4	y5x4
y4	0	0	x2y4	y2y4	0	0	x4y4+x4y4+x4y4	y4y4+y4y4+y4y4	x5y4	y5y4
x5	0	0	x2x5	0	0	0	x4x5	y4x5	x5x5	y5x5
y5	0	0	0	0	0	0	x4y5	y4y5	x5y5	y5y5

Fig 5: <u>Process of generated 10$\times$10 Global stiffness matrix</u>

Script of reduced global stiffness matrix

K =

```
[x1x1,y1x1,x2x1,y2x1,x3x1,y3x1,x4x1,y4x1,x5y1,y5x1;
x1y1,y1y1,x2y1,y2y1,x3y1,y3y1,x4y1,y4y1,x5y1,y5y1;
x1x2,y1x2,x2x2,y2x2,x3x2,y3x2,x4x2,y4x2,x5x2,y5x2;
x1y2,y1y2,x2y2,y2y2,x3y2,y3y2,x4y2,y4y2,x5y2,y5y2;
x1x3,y1x3,x2x3,y2x3,x3x3,y3x3,x4x3,y4x3,x5x3,y5x3;
x1y3,y1y3,x2y3,y2y3,x3y3,y3y3,x4y3,y4y3,x5y3,y5y3;
x1y4,y1y4,x2y4,y2y4,x3y4,y3y4,x4y4,y4y4,x5y4,y5y4]
```

Script for displacement calculation with the equations obtained from reduced GSM:

```
syms u1x u1y u2x u2y u3x u3y u4x
eqn1 = 10.2764*10^8*u1x + 2.6781*10^8*u1y - 2.6781*10^8*u2x - 2.6781*10^8*u2y - 7.5983*10^8*u3x == -9000;
eqn2 = 2.6781*10^8*u1x + 2.6781*10^8*u1y - 2.6781*10^8*u2x - 2.6781*10^8*u2y == -12000;
eqn3 = -2.6781*10^8*u1x - 2.6781*10^8*u1y + 8.762*10^8*u2x + 1.0519*10^8*u2y - 2.8166*10^8*u4x == 0;
eqn4 = -2.6781*10^8*u1x - 2.6781*10^8*u1y + 1.0519*10^8*u2x + 11.2147*10^8*u2y - 7.5983*10^8*u3y +
1.6262*10^8*u4x == 0;
eqn5 = -7.5983*10^8*u1x + 11.9546*10^8*u3x - 4.3563*10^8*u4x == 0;
eqn6 = -7.5983*10^8*u2y + 7.5983*10^8*u3y == 0;
eqn7 = -2.8166*10^8*u2x + 1.6262*10^8*u2y - 4.3563*10^8*u3x + 8.8065*10^8*u4x == 0;
soln = solve([eqn1, eqn2, eqn3, eqn4, eqn5, eqn6, eqn7], [u1x, u1y, u2x, u2y, u3x, u3y u4x]);
Au1x = soln.u1x;      % node 1 x-direction displacement
Au1y = soln.u1y;      % node 1 y-direction displacement
Bu2x = soln.u2x;      % node 2 x-direction displacement
Bu2y = soln.u2y;      % node 2 y-direction displacement
Cu3x = soln.u3x;      % node 3 x-direction displacement
Cu3y = soln.u3y;      % node 3 y-direction displacement
Du4x = soln.u4x;      % node 4 x-direction displacement
Au1x
Au1y
Bu2x
Bu2y
Cu3x
Cu3y
Du4x
```

Displacement results (in meter)

Au1x =

1.5679e-04

Au1y=

-8.5717e-04

Bu2x=

-1.0052e-04

Bu2y=

-5.5506e-04

Cu3x=

1.5284e-04

Cu3y=

-5.5506e-04

Du4x=

1.4595e-04

<u>Nodal Forces result (in N)</u>

```
% Displacements at each node
Au1x = 1.5679*10^-4
Au1y = -8.5717*10^-4
Bu2x = -1.0052*10^-4
Bu2y = -5.5506*10^-4
Cu3x = 1.5284*10^-4
Cu3y = -5.5506*10^-4
Du4x = 1.4595*10^-4
% Forces at each node
F1x = 10.2764*10^8*Au1x + 2.6781*10^8*Au1y - 2.6781*10^8*Bu2x - 2.6781*10^8*Bu2y - 7.5983*10^8*Cu3x;
F1y = 2.6781*10^8*Au1x + 2.6781*10^8*Au1y - 2.6781*10^8*Bu2x - 2.6781*10^8*Bu2y;
F2x = -2.6781*10^8*Au1x - 2.6781*10^8*Au1y + 8.762*10^8*Bu2x + 1.0519*10^8*Bu2y - 2.8166*10^8*Du4x;
F2y = -2.6781*10^8*Au1x - 2.6781*10^8*Au1y + 1.0519*10^8*Bu2x + 11.2147*10^8*Bu2y - 7.5983*10^8*Cu3y +
1.6262*10^8*Du4x;
F3x = -7.5983*10^8*Au1x + 11.9546*10^8*Cu3x - 4.3563*10^8*Du4x;
F3y = -7.5983*10^8*Bu2y + 7.5983*10^8*Cu3y;
F4x = -2.8166*10^8*Bu2x + 1.6262*10^8*Bu2y - 4.3563*10^8*Cu3x + 8.8065*10^8*Du4x;
F4y = 1.6262*10^8*Bu2x - 0.9389*10^8*Bu2y + 1.2033*10^8*Du4x;
F5x = -3.2673*10^8*Bu2x - 1.6336*10^8*Du4x;
F5y = -2.8295*10^8*Du4x;
F1x
F1y
F2x
F2y
F3x
F3y
F4x
F4y
F5x
F5y
```

<u>Force at node A (1) → X in (N)</u>
F1x =

-8.9966e+03

<u>Force at node A (1) → Y in (N)</u>
F1y =

-1.1998e+04

<u>Force at node B (2) → X in (N)</u>
F2x =

-1.8946

Force at node B (2) → Y in (N)
F2y =

  -2.4404

Force at node C (3) → X in (N)
F3x =

   0.1622

Force at node C (3) → Y in (N)
F3y =

     0

Force at node D (4) → X in (N)
F4x =

  -2.2157
Force at node D (4) → Y in (N) [Reaction force at support D]
F4y =

  5.3330e+04

Force at node E (5) → X in (N) [Reaction force at support E]
F5x =

  9.0005e+03

Force at node E (5) → Y in (N) [Reaction force at support E]
F5y =

 -4.1297e+04

Other illustrations of the given truss using numerical method (Abacus)

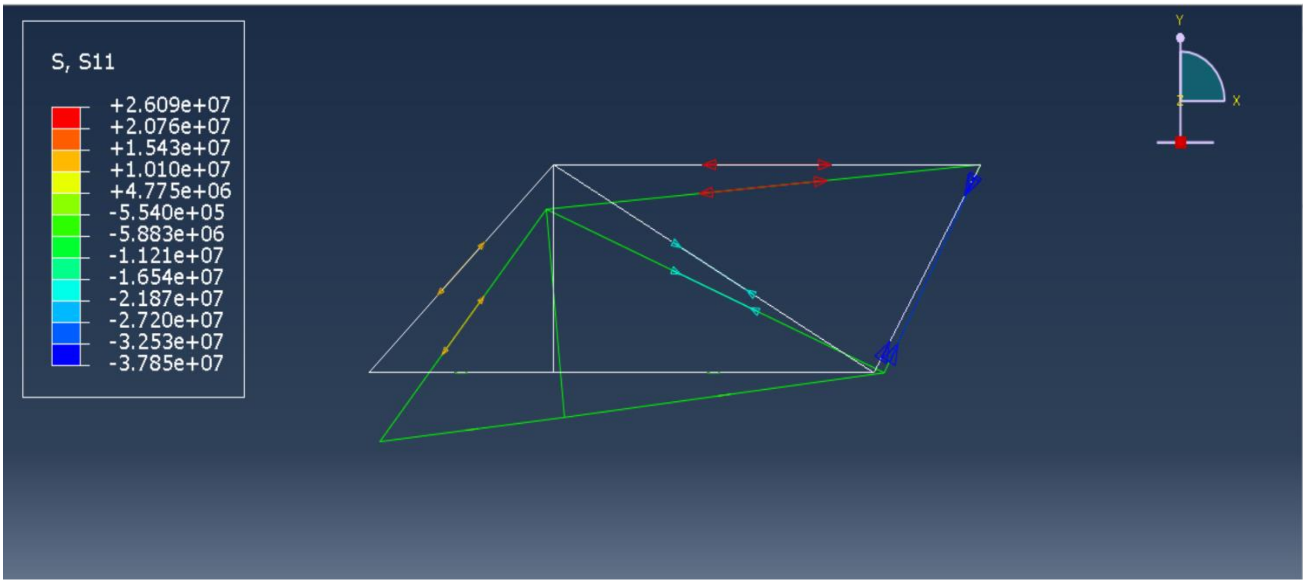


Fig 6: Truss before and after deformation due to 15000 N load applied on node A

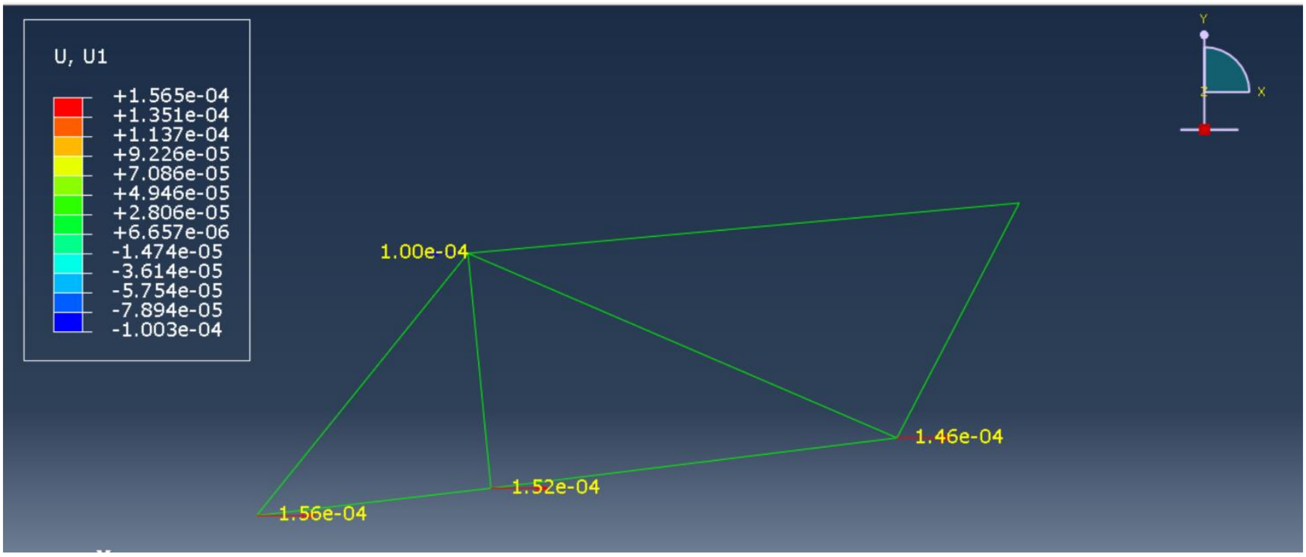


Fig 7: Nodal displacements in X-direction

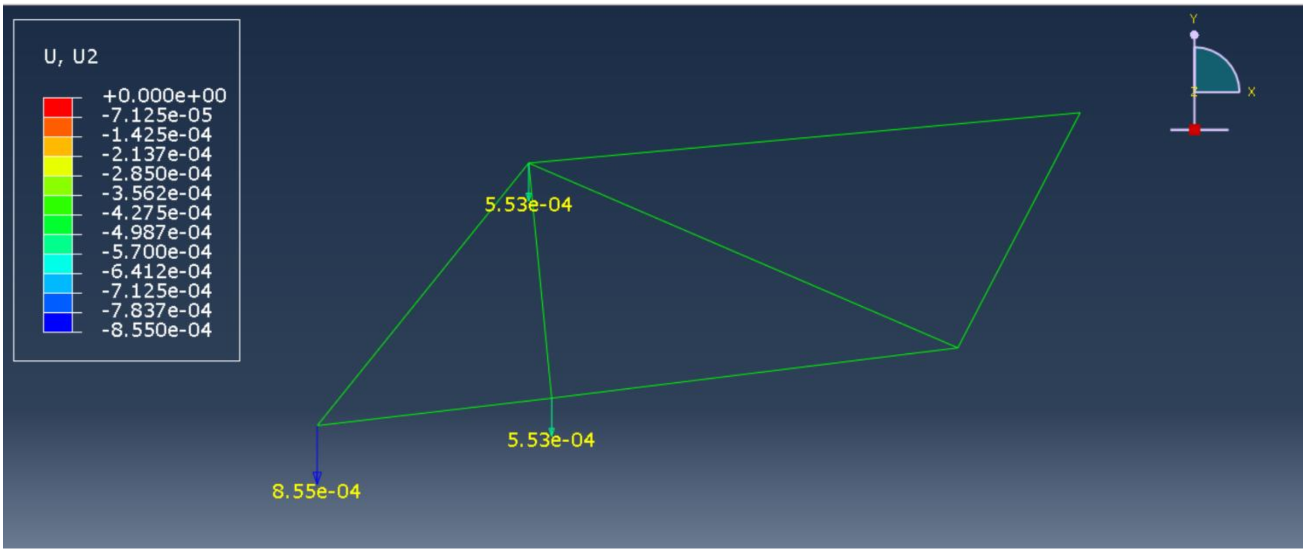
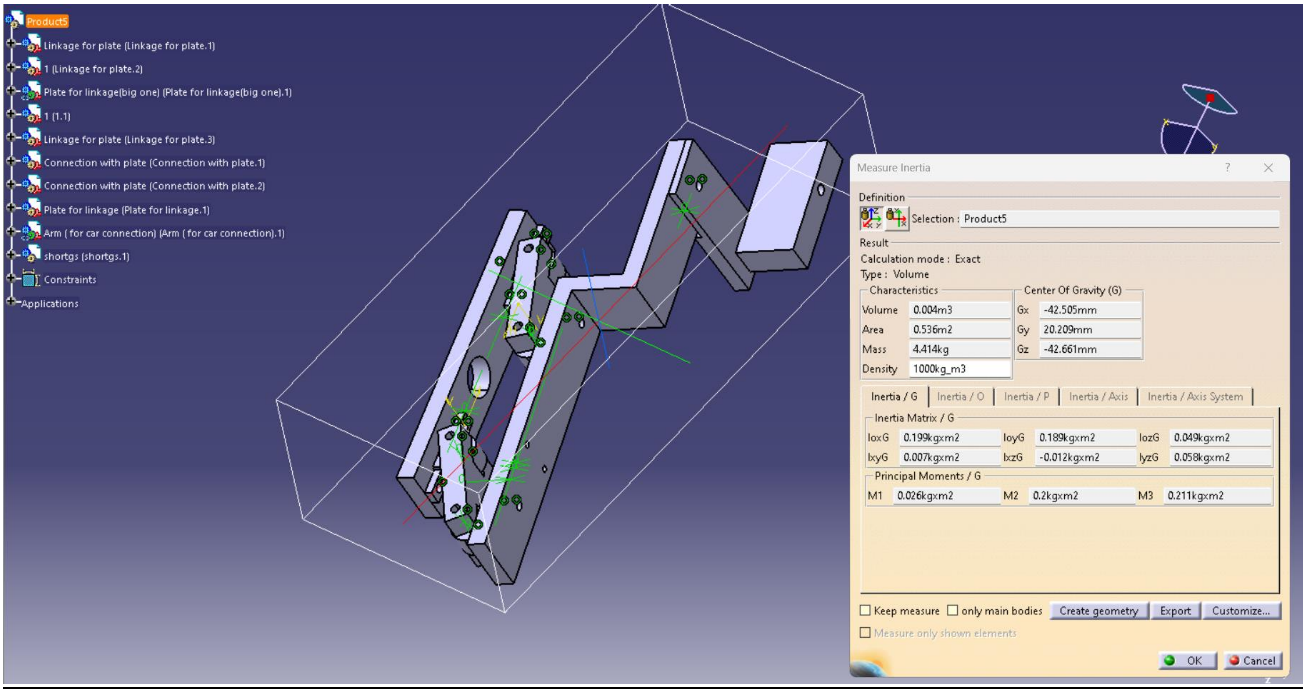


Fig 8: Nodal displacements in Y-direction


Product5
Linkage for plate (Linkage for plate.1)
1 (Linkage for plate.2)
Plate for linkage(big one) (Plate for linkage(big one).1)
1 (1.1)
Linkage for plate (Linkage for plate.3)
Connection with plate (Connection with plate.1)
Connection with plate (Connection with plate.2)
Plate for linkage (Plate for linkage.1)
Arm (for car connection) (Arm (for car connection).1)
shortgs (shortgs.1)
Constraints
Applications
Measure Inertia
Definition
Selection : Product5
Result
Calculation mode : Exact
Type : Volume
Characteristics
Volume 0.004m3
Area 0.536m2
Mass 4.414kg
Density 1000kg_m3
Center Of Gravity (G)
Gx -42.505mm
Gy 20.209mm
Gz -42.661mm
Inertia / G
Inertia / O
Inertia / P
Inertia / Axis
Inertia / Axis System
Inertia Matrix / G
IoxG 0.199kgxm2
IoyG 0.189kgxm2
IozG 0.049kgxm2
IxyG 0.007kgxm2
IxzG -0.012kgxm2
IyzG 0.058kgxm2
Principal Moments / G
M1 0.026kgxm2
M2 0.2kgxm2
M3 0.211kgxm2
Keep measure
only main bodies
Create geometry
Export
Customize...
Measure only shown elements
OK
Cancel