208MAE Analytical Modelling Finite Element Analysis Coursework

By Derrano Green (SID:5466500)

Abstract

Introduction

A truss structure with properties of 70Gpa and 250Mpa, assumed to be aluminium (Kulper et al., 2017, p.1115) will be loaded at node F with a 100N force (see figure 1). The forces in each element; nodal forces and nodal displacements will be examined and compared using three methods: Method of joints; Finite Element Method (FEM); and Finite Element Analysis (FEA) software from Abaqus CAE.

The results will be examined to determine which method produces the most accurate estimate.

Method of Joints: Static Equilibrium Calculations

The method of joints technique assumes the truss is in equilibrium. Therefore, moments and forces in the X and Y planes are balanced and equate to zero.

As node F has two unknown forces and a known force this is used as a starting point to use the equilibrium conditions to simultaneously solve the unknown forces. Once solved in MATLAB using the matrix method, another node can then be equated to solve the other unknown forces using the equilibrium conditions till all unknowns are solved.

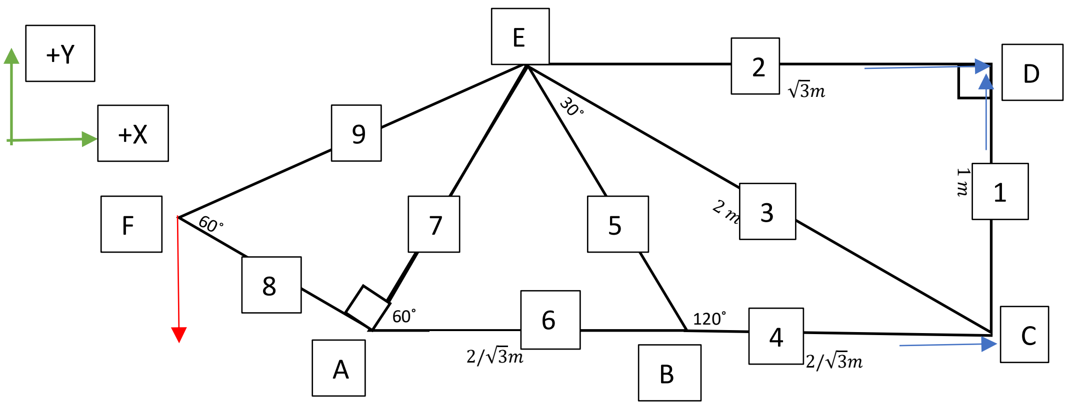

Figure 1: Free Body Diagram (FBD) of model truss structure. Nodes labelled A-F. Elements labelled 1-9. Reaction forces denoted by blue arrows. Applied force represented by red arrow.

Support reaction forces

$$\sum F_x: C_x = D_x$$

$$\sum F_y: D_y = 100\ N$$

$$Moments\ @\ D: C_x \times 1 + \frac{5 \times 100}{\sqrt{3}} = 0$$

$$C_x = \frac{-500}{\sqrt{3}}\ N\ (-288.7\ N);\ D_x = \frac{500}{\sqrt{3}}\ N\ (288.7N\ )$$

Table 1: Shows calculated tension force values, for each element, from using method of joints. Negative values represent compressive forces.

Equilibrium: Method of joints									
Element	CD (1)	DE (2)	CE (3)	BC (4)	BE (5)	AB (6)	AE (7)	FA (8)	FE (9)
Force, N	33.33	173.21	-66.67	57.74	0	57.74	57.74	-100	100

Sample calculation at node E shown below.

$$\sum F_x: F_{ED} = F_{FE} \times \cos 30 + F_{AE} \times \sin 30 - F_{BE} \sin 30 - F_{CE} \cos 30$$

Finite Element Calculations ("Mathematical Model")

Each element was indexed to the six nodes on the truss. As each node has two degrees of freedom, thus 12 degrees of freedom for the structure, a 4x4 stiffness matrix K was created for all nine elements to describe the potential displacement of each element. To describe the full movement of the truss, a 12x12 matrix was formed, using the previous indexing to correctly place the values.

The formula, U = K^{-1}*F, was used to first solve for nodal displacements, resulting in a reduced K matrix as unknown forces were removed from the equation. As all unknown displacements were then solved, F=K*U, using the original matrices, was used to calculate the unknown nodal forces. The results of these calculation are shown below. (Bofang, 2014, p.189).

Table 2: Shows nodal forces and nodal displacements of truss structure, see figure 1 for indexing.

$\begin{pmatrix} F_{C_x} \\ F_{C_y} \\ F_{D_x} \\ F_{D_y} \\ F_{E_x} \\ F_{E_y} \\ F_{B_x} \\ F_{B_y} \\ F_{A_x} \\ F_{A_y} \\ F_{F_x} \\ F_{F_y} \end{pmatrix} (N) = \begin{pmatrix} -287.4906 \\ 0.1329 \\ 287.5779 \\ 99.5099 \\ -0.1658 \\ -0.8125 \\ -0.7504 \\ 0.2001 \\ 0.4554 \\ 0.8387 \\ 0.3735 \\ -99.8690 \end{pmatrix} (N)$	$\begin{pmatrix} U_{C_x} \\ U_{C_y} \\ U_{D_x} \\ U_{D_y} \\ U_{E_x} \\ U_{E_y} \\ U_{B_x} \\ U_{B_y} \\ U_{A_x} \\ U_{A_y} \\ U_{F_x} \\ U_{F_y} \end{pmatrix} (m) = 1 \times 10^{-3} \times \begin{pmatrix} 0 \\ -0.0181 \\ 0 \\ 0 \\ -0.0906 \\ -0.249 \\ 0.0181 \\ -0.1862 \\ 0.0363 \\ -0.3326 \\ 0.0049 \\ -0.4416 \end{pmatrix} (m)$
A) Forces at nodes	B) Displacement at nodes

Numerical (FEA) Model

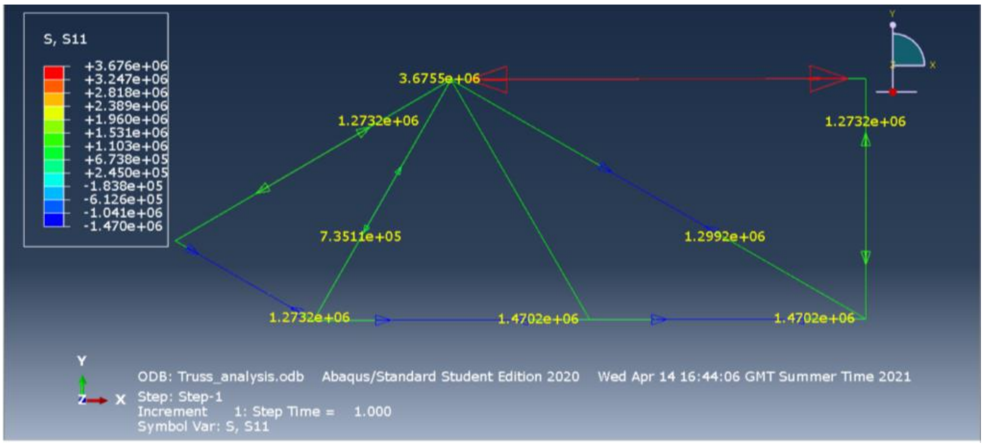

Figure 2: Shows Stress (in Pascals) for each element from simulation.

Table 3: Shows Stress in each element from simulation and subsequently the calculated Force using F = σ (stress, Pa) *A (cross sectional area, m^2).

	FEA Elements								
Element	CD (1)	DE (2)	CE (3)	BC (4)	AB (6)	BE (5)	AE (7)	FE (9)	FA (8)
Stress, Pa	1.27E+06	3.68E+06	- 1.30E+06	- 1.47E+06	- 1.47E+06	0.0	7.35E+05	1.27E+06	-1.27E+06
Force, N	100.0	288.7	-200.0	-115.5	-115.5	0.0	57.7	100.0	-100.0

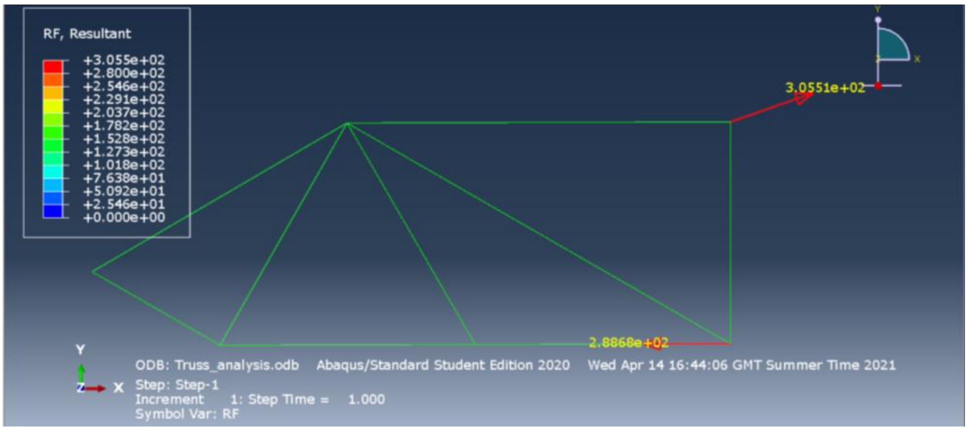

Figure 3: Shows reaction forces and (in this case) nodal forces from simulation.

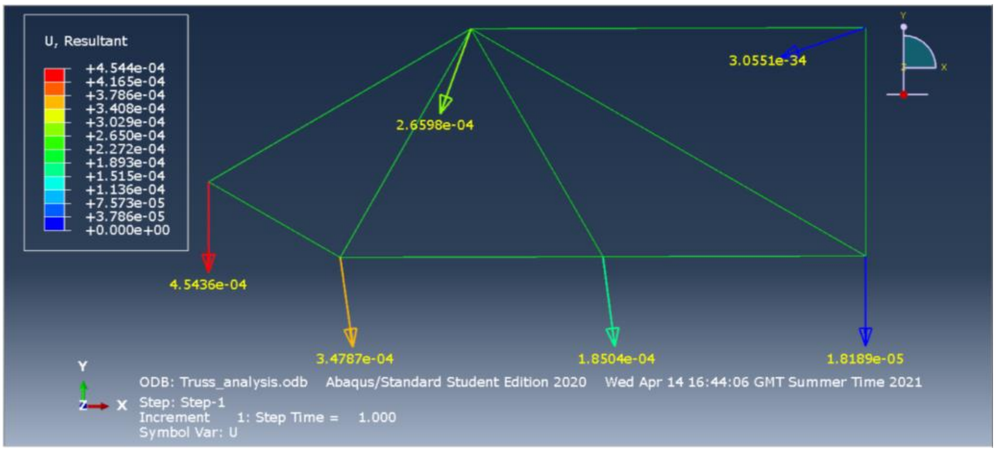

Figure 4: Shows resultant nodal displacements from simulation.

Table 4: Shows nodal forces, nodal displacements, and reaction forces in both the X and Y directions.

FEA Nodes						
Node	C (1)	D (2)	E (3)	B (4)	A (5)	F (6)
displacement, x (m)	2.89E-34	-2.89E-34	-9.09E-05	2.43E-05	4.85E-05	9.38E-07
displacement, y (m)	1.82E-05	1.00E-34	2.50E-04	1.83E-04	3.44E-04	4.54E-04
Reaction Force, x (N)	-288.68	2.89E+02	0	0	0	0
Reaction Force, y (N)	0	1.00E+02	0	0	0	0
Nodal Force, x (N)	-288.68	2.89E+02	0	0	0	0
Nodal Force, y (N)	0	1.00E+02	0	0	0	0

Analysis & Discussion

As member BE, indexed as element 5, has zero stress or no load acting upon it, member CE indexed as element 3 supports the truss from collapsing upon itself with a compressive force as member CD and DE are tensile and hold the truss together, as the bar CE has a larger surface are the rest of the other bars(elements) the stress on it is reduced compared if the cross sectional area was lower.

Engineering Yield strain is 0.2% above a material’s elastic limit. (Lee et al., 2005, p.191).

A safety factor of 2 will be applied.

$$factor\ of\ safty = \frac{UTS}{working\ stress} \quad (Hannah\ \&\ Hillier, 1962, p. 280)$$

Conclusions

Reference List

Bofang, Z. (2014). *Thermal Stresses and Temperature Control of Mass Concrete*. (1st ed.). Butterworth-Heinemann.

Hannah, J., and Hillier, M. (1962) *Mechanical engineering science*. (3rd ed.) Prentice Hall.

Kulper, S., Fang, C., Ren, X., Guo, M., Sze, K., Leung, F., & Lu, W. (2017). Development and Initial Validation of a Novel Smoothed-ParticleHydrodynamics-Based Simulation Model of Trabecular BonePenetration by Metallic Implants. *Journal of Orthopaedic Research*, 36(4), 1114-1123. https://doi.org/10.1002/jor.23734.

Lee, Y., Pan, J., Hathaway, R., & Barkey, M. (2005). *Fatigue Testing and Analysis: Theory and Practice.* (1st ed.). Elsevier Butterworth-Heinemann.

Appendix A.1 – MATLAB Transcript for mathematical model & method of joints equilibrium equations

```
%calculate cross sectional area of truss elements
a_CE= pi*((.014)^2)*0.25;
a_rest = pi*.01^2*.25;


% define element stiffness matrices
k1 = ((70*10^9)*a_rest)*[0 0 0 0; 0 1 0 -1; 0 0 0 0; 0 -1 0 1];
k2 = (70*10^9*a_rest / sqrt(3))* [1 0 -1 0; 0 0 0 0; -1 0 1 0; 0 0 0 0];
k3 = (70*10^9*a_CE / 2)* [3/4, -1*sqrt(3)/4, -3/4, sqrt(3)/4; -1*sqrt(3)/4,
1/4, sqrt(3)/4, -1/4; -3/4, sqrt(3)/4, 3/4, -1*sqrt(3)/4; sqrt(3)/4, -1/4, -
1*sqrt(3)/4, 1/4];
k4 = (2*70*10^9*a_rest/ sqrt(3))* [1 0 -1 0; 0 0 0 0; -1 0 1 0; 0 0 0 0];
k5 = (2*70*10^9*a_rest / sqrt(3))* [1/4, -1*sqrt(3)/4, -1/4, sqrt(3)/4; -
1*sqrt(3)/4, 3/4,sqrt(3)/4, -3/4; -1/4, sqrt(3)/4, 1/4, -1*sqrt(3)/4;
sqrt(3)/4, -3/4, -1*sqrt(3)/4, 3/4 ];
k6 = (2*70*10^9*a_rest / sqrt(3))* [1 0 -1 0; 0 0 0 0; -1 0 1 0; 0 0 0 0];
k7 = (2*70*10^9*a_rest / sqrt(3))* [1/4, sqrt(3)/4, -1/4, -1*sqrt(3)/4;
sqrt(3)/4, 3/4, -1*sqrt(3)/4, -3/4; -1/4, -1*sqrt(3)/4, 1/4,sqrt(3)/4; -
1*sqrt(3)/4, -3/4, sqrt(3)/4, 3/4 ];
k8 = (2*70*10^9*a_rest / 3)* [3/4, -1*sqrt(3)/4, -3/4, sqrt(3)/4; -
1*sqrt(3)/4, 1/4, sqrt(3)/4, -1/4; -3/4, sqrt(3)/4, 3/4, -1*sqrt(3)/4;
sqrt(3)/4, -1/4, -1*sqrt(3)/4, 1/4];
k9 = (4*70*10^9*a_rest / 3)* [3/4, sqrt(3)/4, -3/4, -1*sqrt(3)/4; sqrt(3)/4,
1/4, -1*sqrt(3)/4, -1/4; -3/4, -1*sqrt(3)/4, 3/4, sqrt(3)/4; -1*sqrt(3)/4, -
1/4, sqrt(3)/4, 1/4 ];


% create global stiffens matrix
K=zeros(12);


% indexing individual stiffness matrices to global stiffness matrix
vec1 = [1 2 3 4];
K(vec1,vec1) = K(vec1,vec1) + k1;


vec2 = [5 6 3 4];
K(vec2,vec2) = K(vec2,vec2) + k2;


vec3 = [1 2 5 6];
K(vec3,vec3) = K(vec3,vec3) + k3;


vec4 = [7 8 1 2];
K(vec4,vec4) = K(vec4,vec4) + k4;
```

```
vec5 = [7 8 5 6];
K(vec5,vec5) = K(vec5,vec5) + k5;
vec6 = [9, 10, 7, 8];
```

Appendix A.2 – MATLAB Transcript for mathematical model & method of joints equilibrium equations

```
K(vec6,vec6) = K(vec6,vec6) + k6;


vec7 = [9, 10, 5, 6];
K(vec7,vec7) = K(vec7,vec7) + k7;


vec8 = [9, 10, 11, 12];
K(vec8,vec8) = K(vec8,vec8) + k8;


vec9 = [11, 12, 5, 6];
K(vec9,vec9) = K(vec9,vec9) + k9;


disp(K)


% solve for U; reduced stiffness matrix; U=k^-1*f


%Indexing reduced matrix

kr1 = 1*10^7 * [0.6845, 0.2333   -0.1347         0         0         0
0         0         0];
kr2 = 1*10^7 * [0.2333 1.5887    0.0841   -0.1587    0.2749   -0.1587   -
0.2749   -0.5498   -0.3174];
kr3 = 1*10^7 * [-0.1347 0.0841    1.2702    0.2749   -0.4761   -0.2749   -
0.4761   -0.3174   -0.1833];
kr4 = 1*10^7 * [0,-0.1587    0.2749    1.4284   -0.2749   -0.6348         0
0         0];
kr5 = 1*10^7 * [0 0.2749   -0.4761   -0.2749    0.4761         0         0
0         0];
kr6 = 1*10^7 * [0, -0.1587   -0.2749   -0.6348         0    1.0684    0.1162
-0.2749    0.1587];
kr7 = 1*10^7 * [0 -0.2749   -0.4761         0         0    0.1162    0.5678
0.1587   -0.0916];
kr8 = 1*10^7 * [0 -0.5498   -0.3174         0         0   -0.2749    0.1587
0.8247    0.1587];
kr9 = 1*10^7 * [0 -0.3174   -0.1833         0         0    0.1587   -0.0916
0.1587    0.2749];


% Calculate and show reduced matrix

Kr= [kr1;kr2;kr3;kr4;kr5;kr6;kr7;kr8;kr9];
disp(Kr);
```

Appendix A.3 – MATLAB Transcript for mathematical model & method of joints equilibrium equations

```
% solve unknown displacements with reduced matrix

u = inv(Kr)*[0;0;0;0;0;0;0;0;-100];
disp(u);


% input solved values from into U to solve for unknown Forces, F

U = 1*10^-3* [0; -0.0181; 0; 0; -0.0906; -0.2490; 0.0181; -0.1862; 0.0363; -
0.3326; 0.0049; -0.4416]
F = K * U;
disp(F);

% Method of Joints equilibrium equations


% node F
F = [cosd(30),cosd(30); sind(30), -sind(30)];
F_FE_and_F_FA = inv(F)*[0;100];
F_FE = F_FE_and_F_FA(1,:);
F_FA = F_FE_and_F_FA(2,:);
disp(F_FE_and_F_FA);


% node A
A = [1, cosd(60); 0 sind(60)];
F_AB_and_F_AE = inv(A)*[100*cosd(30); 100*sind(30)];
F_AB = F_AB_and_F_AE(1,:);
F_AE = F_AB_and_F_AE(1,:);
disp(F_AB_and_F_AE);


% node B
B = [1, -cosd(60); 0, sind(60)];
F_BC_and_F_BE = inv(B) * [F_AB;0];
F_BC = F_BC_and_F_BE(1,:);
F_BE = F_BC_and_F_BE(2,:);
disp(F_BC_and_F_BE);


% node C
C = [0, -cosd(30); 1, sind(30)];
F_CD_and_F_CE = inv(C) * [F_BC; 0];
F_CD = F_CD_and_F_CE(1,:);
F_CE = F_CD_and_F_CE(2,:);
disp(F_CD_and_F_CE);
```

```
% node E
F_ED = F_FE*cosd(30) + F_AE*sind(30) - F_BE*sind(30) - F_CE*cosd(30);
disp(F_ED);
```