Truss Project:

Preliminary Design

Kelly Lam

Wenhao Cao

Xinyu Lei

Boston University

EK301: Engineering Mechanics I

Professor Albro

November 11th, 2022

Introduction

The goal of this computer program is to predict the performance of a candidate truss and verify that it can correctly analyze a truss. The intention behind doing the project is to take the understanding of systems of forces on a truss into another level. A program is more cost and time efficient by identifying any issues and defects (i.e. going over the cost limit, not meeting the minimum live load) and allows us to eliminate any faulty designs before building any truss. It evaluates the pros and cons of each design by predicting the maximum load of the truss, the critical member etc., enabling us not only to choose the best design but also to improve our design based on the data.

Methods & Analysis

The method of this project is to analyze truss design through a computational approach, therefore we have made a MATLAB script to help us analyze the designs. A truss analysis can also be performed by hand. To verify the validity of our script, we have decided to do a sample truss problem by hand and then by the code.

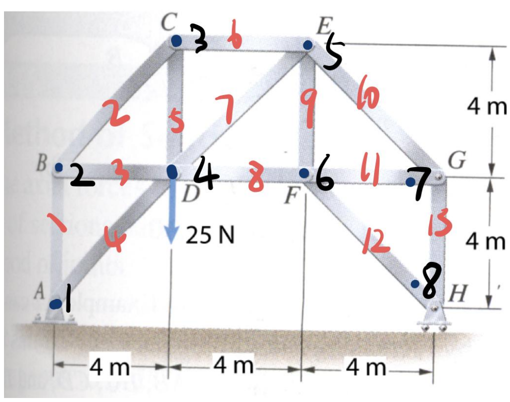


Figure1: Practice Truss Design

Figure1 is the sample design we use to confirm the validity of our code. The truss has eight joints and 13 members.

1) Handwritten solution

By calculating the forces at each joint and the overall reaction forces by hand[1], we get the truss analysis for the practice problem.

2) Computer code result

Figure 2: practice problem software output

```
Load: 98 oz
Member forces in oz
m1: 65.076 (C)
m2: 92.031 (C)
m3: 65.076 (T)
m4: 0.000 ( )
m5: 65.076 (T)
m6: 65.076 (C)
m7: 46.016 (T)
m8: 32.538 (T)
m9: 0.000 (T)
m10: 46.016 (C)
m11: 32.538 (T)
m12: 0.000 ( )
m13: 32.538 (C)
Reaction forces in oz:
Sx1: 0.000
Sy1: 65.076
Sy2: 32.538
Cost of truss: $140
Theoretical max load/cost ratio in oz/$: 0.69583
```

Figure3: members' resultant internal tension in code

```
T =

  -16.6667
  -23.5702
   16.6667
         0
   16.6667
  -16.6667
   11.7851
    8.3333
    0.0000
  -11.7851
    8.3333
         0
   -8.3333
         0
   16.6667
    8.3333
```

The internal forces of the members match up with the hand written answers, which means that the MATLAB calculation is correct.

In the MATLAB code, we manage to calculate the force in each member given a vertical load at a specific joint while identifying the compression and tension. The codes are divided into two scripts. The first script inputs the truss design and encoded the matrices : *Connection Matrix (C), Reaction Matrix (Sx, Sy) , Load Matrix (L) and Location Matrix (X, Y)* for calculation. It then saves all the matrics into a .mat file. The second script first reads in the matrix from the first script. Then, in order to calculate the tension and compression each truss member is experiencing, we have to make a coefficient matrix that tells us the relations between the members and joints. Our approach is that for each member, we loop through its connection using the connection matrix and find out which joining is connected to the member (when C==1). Then for each connection, we use the location matrix and a distance function to calculate the

[1] Practice problem's detailed calculation by hand is shown in Appendix

coefficient of connection. From there, we are able to build *A* matrix by concatenating the coefficient (unit vector) matrix in x and y direction with the reaction forces matrices *Sx, Sy*. The tension and compression forces can be found by solving linear equations of *A* matrix and *L* matrix. The result of the *T* matrix is a matrix with one column. The rows are determined by the number of members in the truss. The tensions are shown by the forces being positive and the compression vice versa. **Figure3** shows the matrix T for the practice problem.

Using the rule of linearity, we can then find the matrix Rm[2] by dividing *T* with the load that we have used. The smallest Rm tells us which member buckles first. In the code, we have used the function min() to find the smallest Rm then use find() to search for the index of the buckling member. In the sample case, the buckling member is #4.

The buckling force is the most force a member can take in compression, which is equal to negative Pcrit. Since different lengths acrylic members have different Pcrit, we have calculated the lengths for individual members. Then we use the equation GTAs provided by analyzing our buckling labs to get a vector of Pcrit[3]. By indexing the buckling member, we can find the force that it will take for the buckling member to break. Pcrit of that member dividing its Rc gives us the failure load. In the sample case, that is 98 oz.

The calculation of the truss is given by the project manual, which is ten dollars for each joint and one dollar for each inch of acrylic member that is used. The max load and cost ratio is maxload divides the cost. They are implemented in the code by two equations and displayed in the result[4].

[2] Matrix Rm of the practice problem is shown in the Appendix
[3] Vector Pcrit of the practice problem is shown in the Appendix
[4] Code for Cost and Max Load/Cost Ratio is shown in Appendix

Results:

Design 1:

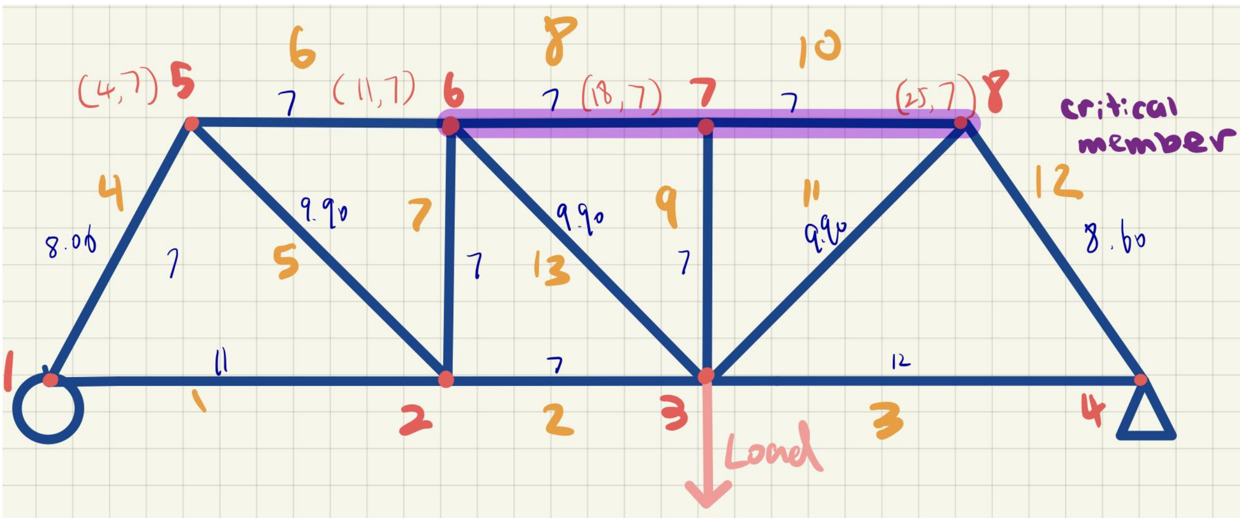


Figure 4: Design 1 with labeled pin (triangle), roller (circle), members (orange), joints (red), and highlighted critical member (purple).

Member_number	Member_Length_in	Tension_Compression	Buckling_Strength_oz	Uncertainty_oz	Forces_at_Maximum_Truss_Load_oz
1	11	"Tension"	NaN	NaN	13.356
2	7	"Tension"	NaN	NaN	36.729
3	12	"Tension"	NaN	NaN	25.043
4	8.0623	"Compression"	45.308	7.7712	26.92
5	9.8995	"Tension"	NaN	NaN	33.054
6	7	"Compression"	60.102	11.677	36.729
7	7	"Compression"	60.102	11.677	23.373
8	7	"Compression"	60.102	11.677	60.102
9	7	"Compression"	60.102	11.677	0
10	7	"Compression"	60.102	11.677	60.102
11	9.8995	"Tension"	NaN	NaN	49.582
12	8.6023	"Compression"	39.797	6.3165	43.085
13	9.8995	"Tension"	NaN	NaN	33.054

Figure 5: Table of Design 1's member number, its length, tension or compression, bucking strength, uncertainty, and maximum truss load.

```
\% EK301, Section A5, Group A: Kelly Lam, Xinyu Lei, Wenhao Cao, 11/11/2022.
Load: 58 oz
Member forces in oz
m1: 13.356 (T)
m2: 36.729 (T)
m3: 25.043 (T)
m4: 26.920 (C)
m5: 33.054 (T)
m6: 36.729 (C)
m7: 23.373 (C)
m8: 60.102 (C)
m9: 0.000 (C)
m10: 60.102 (C)
m11: 49.582 (T)
m12: 43.085 (C)
m13: 33.054 (T)
Reaction forces in oz:
Sx1: 0.000
Sy1: 35.060
Sy2: 23.373
Cost of truss: $191
Theoretical max load/cost ratio in oz/$: 0.30535
```

Figure 5: Design 1 software output

Design 2:

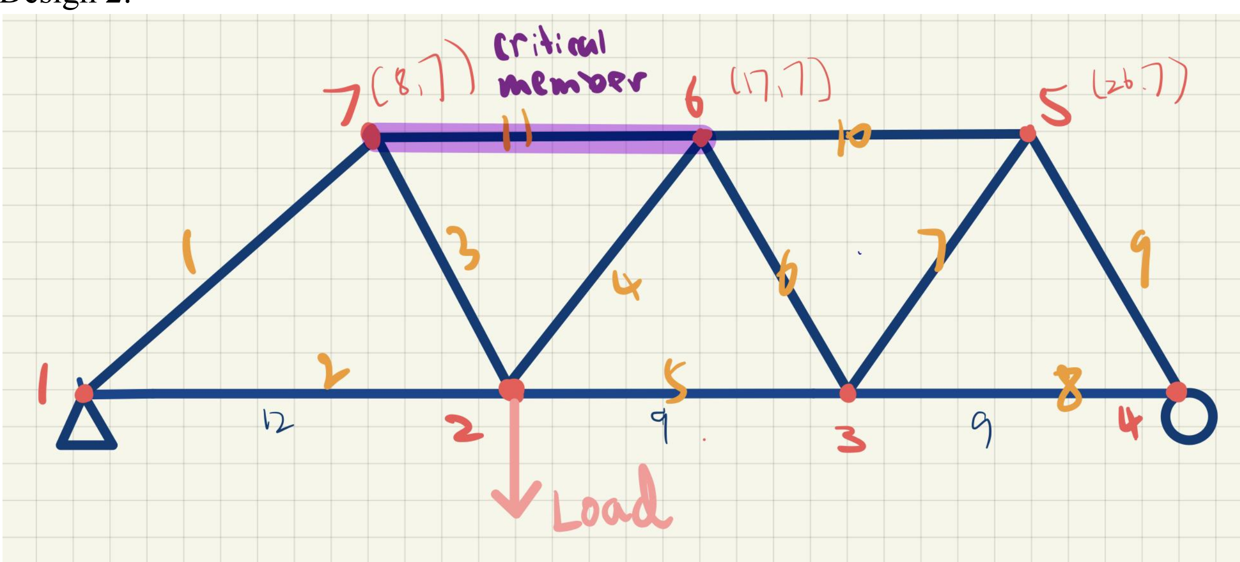


Figure 6: Design 2 with labeled pin (triangle), roller (circle), members (orange), joints (red), and highlighted critical member (purple).

Member_number	Member_Length_in	Tension_Compression	Buckling_Strength_oz	Uncertainty_oz	Forces_at_Maximum_Truss_Load_oz
1	10.63	"Compression"	26.062	2.6903	32.208
2	12	"Tension"	NaN	NaN	24.239
3	8.0623	"Tension"	NaN	NaN	24.427
4	8.6023	"Tension"	NaN	NaN	17.376
5	9	"Tension"	NaN	NaN	26.259
6	8.0623	"Compression"	45.308	7.7712	16.285
7	8.6023	"Tension"	NaN	NaN	17.376
8	9	"Tension"	NaN	NaN	8.0796
9	8.0623	"Compression"	45.308	7.7712	16.285
10	9	"Compression"	36.358	5.4085	18.179
11	9	"Compression"	36.358	5.4085	36.358

Figure 7: Table of Design 2's member number, its length, tension or compression, bucking strength, uncertainty, and maximum truss load.

```
\% EK301, Section A5, Group A: Kelly Lam, Xinyu Lei, Wenhao Cao, 11/11/2022.
Load: 35 oz
Member forces in oz
m1: 32.208 (C)
m2: 24.239 (T)
m3: 24.427 (T)
m4: 17.376 (T)
m5: 26.259 (T)
m6: 16.285 (C)
m7: 17.376 (T)
m8: 8.080 (T)
m9: 16.285 (C)
m10: 18.179 (C)
m11: 36.358 (C)
Reaction forces in oz:
Sx1: 0.000
Sy1: 21.209
Sy2: 14.139
Cost of truss: $170
Theoretical max load/cost ratio in oz/$: 0.20790
```

Figure 8: Design 2 software output.

Discussion & conclusion

Based on the two designs, Design 1 would bear the greatest load at 58 oz with Design 2 bearing 35 oz. Design 1 also had a better load-to-cost ratio at $0.30535 while Design 2 had $0.20790. The results indicate that Design 1 was the better design out of the two because it not only supports the most weight but also has the highest load-to-cost ratio. There are two critical members in Design 1, members 8 and 10, each with a compression of 60.102 N while Design 2 has one critical member at 11 with 36.358 N. To improve Design 1, vertical members can be added between members 8 and 13; and members 10 and 11, closer to pins 6 and 10 to decrease the compression of the critical members. The weight in Design 2 is more evenly distributed with every member bearing a load whether it is tension or compression while member 9 in Design 1 bears no load. To improve on design 1, member 9 can be replaced with another load-bearing member to increase the maximum load or member 9 can be removed to increase the load-out cost ratio.

Appendix

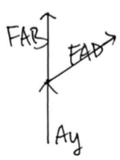
1. Practice problem's detailed calculation by hand

For the whole truss:

$\Sigma F_x = 0 \quad \therefore A_x = 0 \quad \Sigma M_A = 0,\ R_H \cdot 12 - 25 \cdot 4 = 0 \quad \therefore R_H = 8.33\,N\,(T)$

$\Sigma F_y = 0 \quad A_y - 25 + R_H = 0 \quad \therefore A_y = 16.67\,N\,(T)$

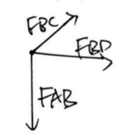
For A:



$\Sigma F_x = 0 \quad F_{AD}\sin 45° = 0 \quad \therefore F_{AD} = 0$

$\Sigma F_y = 0 \quad F_{AB} + A_y + F_{AD}\cos 45° = 0 \quad \therefore F_{AB} = -A_y = -16.67\,N\,(C)$

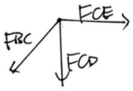
For B:



$\Sigma F_y = 0 \quad F_{BC}\sin 45° - F_{AB} = 0 \quad \therefore F_{BC} = -23.57\,N\,(C)$

$\Sigma F_x = 0 \quad F_{BC}\cos 45° + F_{BD} = 0 \quad \therefore F_{BD} = 16.67\,N\,(T)$

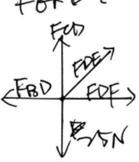
For C:



$\Sigma F_x = 0 \quad F_{CE} - F_{BC}\sin 45° = 0 \quad \therefore F_{CE} = -16.67\,N\,(C)$

$\Sigma F_y = 0 \quad F_{CD} - F_{BC}\cos 45° = 0 \quad \therefore F_{CD} = 16.67\,N\,(T)$

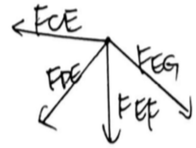
For D:



$\Sigma F_y = 0 \quad F_{CD} + F_{DE}\sin 45° - 25 = 0$
$\therefore F_{DE} = 11.78\,N\,(T)$

$\Sigma F_x = 0 \quad F_{DF} - F_{BD} + F_{DE}\sin 45° = 0$
$\therefore F_{DF} = 8.33\,N\,(T)$

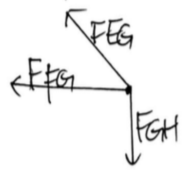
At E:



$\Sigma F_x = 0 \quad F_{EG}\sin 45° - F_{CE} - F_{DE}\sin 45° = 0$
$\therefore F_{EG} = -11.78\,N\,(C)$

$\Sigma F_y = 0 \quad -F_{EF} - F_{EG}\cos 45° - F_{DE}\cos 45° = 0$
$\therefore F_{EF} = 0\,N$

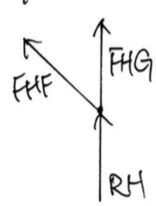
At G:



$\Sigma F_x = 0 \quad -F_{EG}\cos 45° - F_{FG} = 0$
$\therefore F_{FG} = 8.33\,N\,(T)$

$\Sigma F_y = 0 \quad F_{EG}\sin 45° - F_{GH} = 0$
$\therefore F_{GH} = -11.78\,N\,(C)$

At H:



$\Sigma F_x = 0 \quad \therefore F_{HF} = 0\,N$

2. Matrix Rm of the practice problem

```
Rm =

   -0.6667
   -0.9428
    0.6667
         0
    0.6667
   -0.6667
    0.4714
    0.3333
    0.0000
   -0.4714
    0.3333
         0
   -0.3333
         0
    0.6667
    0.3333
```

3. Vector Pcrit of the practice problem

```
Pcrit =

  184.0625   92.0312  184.0625   92.0312  184.0625  184.0625   92.0312  184.0625  184.0625   92.0312  184.0625   92.0312  184.0625
```

4. Code for Cost and Max Load/Cost Ratio

```
cost = sum(length)+10*j;

fprintf('Theoretical max load/cost ratio in oz/$: %0.5f\n',Wf(1)/cost)
```

5. Input File Code

```
j = input("How many joints are there? \n");
m = input("How many members are there? \n");

if (2*j-3 ~= m)
    fprintf('Joints and members don''t satisfied the requirements\n');
    return
end

C = zeros(j,m);
Sx = zeros(j,3);
Sy = zeros(j,3);
X = zeros(1,j);
Y = zeros(1,j);
L = zeros(1,2*j);
count = 1;
count2 = 1;
for i = 1:j
    fprintf('About joint %d\n',i);

    %reaction

    if count <= 2
        choice = input("Is there a reaction force? Y/N\n",'s');
        if choice == 'Y'
            count = count +1;
            choice = input("Is this a rollar or pin? R/P\n",'s');
            if choice == 'P'
                Sx(i,1) = 1;
                Sy(i,2) = 1;

            elseif choice == 'R'
                Sy(i,3) = 1;
            end

        else
            fprintf('joint%d has no reaction force.\n',i)
        end
    end
```

```
    %connected members
    num_mem = input("How many members is this joint connected to? \n");
    for k = 1:num_mem
        member = input(sprintf("What's the #%d member it is connected to? \n",k));
        C(i,member) = 1;
    end

    %joint location
    X(i) = input("what's the x coordinate of the joint? \n");
    Y(i) = input("what's the y coordinate of the joint? \n");

    %load applied
    if count2 <2
    choice = input("Is there load applied to this joint? Y/N \n",'s');
        if choice == 'Y'
            count2 = count2 +1;
            choice = input("Pick if the load is Horizontal or Vertical: H/V \n",'s');
            if choice == 'H'
                weight = input("Weight of the load? \n");
                L(i) = weight;
            elseif choice == 'V'
                weight = input("Weight of the load? \n");
                L(i+j) = weight;
            end
        elseif choice == 'N'
            sprintf("This joint has no load.");
        end
    end


end
L = L';
name = input('name the file: (with .mat)','s');
save(name,'C','Sx','Sy','X','Y','L')
```

6. Output File Code

```
load('Practice.mat');

[j, m] = size(C);
%makeT();
A = makeA(j,m,C,X,Y,Sx,Sy);
T = linsolve(A,L);
Wl = sum(L);
Rm = T/Wl;
length = make_length(m,X,Y,C);
Rmm = min((Rm));
index = find(Rm==Rmm);
Pcrit =  make_Pcrit(length,m);
Wf = -Pcrit(index)/Rmm;
Tfinal = Wf(1) * Rm;
cost = sum(length)+10*j;

%Print Analysis
fprintf('\\%% EK301, Section A5, Group A: Kelly Lam, Xinyu Lei, Wenhao Cao, 11/11/2022.\n')
fprintf('Load: %d oz\n',round(Wf(1)));
fprintf('Member forces in oz\n');
for i = 1:m
    if Tfinal(i,1) > 0
        f = 'T';
    elseif Tfinal(i,1) == 0
        f = ' ';
    else
        f = 'C';
    end
    fprintf('m%d: %.3f (%c)\n',i,abs(Tfinal(i,1)),f);
end

fprintf('Reaction forces in oz:\n')
fprintf('Sx1: %.3f\n',Tfinal(m+1,1))
fprintf('Sy1: %.3f\n',Tfinal(m+2,1))
fprintf('Sy2: %.3f\n',Tfinal(m+3,1))
fprintf('Cost of truss: $%d\n',round(cost))
fprintf('Theoretical max load/cost ratio in oz/$: %0.5f\n',Wf(1)/cost)
```

```
function Pcrit = make_Pcrit(length,m)
    Pcrit = zeros(1,m);
    for i =1:m
        Pcrit(i) = 2945/(length(i)^2);
    end
end

function length = make_length(m,X,Y,C)
    length = zeros(1,m);
    for i = 1:m
        CC = C(:,i);
        Col = find(CC==1)';
        length(i)=distance(Col(1),Col(2),X,Y);
    end
end
% j = #joints, m = #members

%make matrix A
% size [2j x m+3]
function A = makeA(j,m,C,X,Y,Sx,Sy)
    Coe = makeCoe(j,m,C,X,Y);
    S = [Sx;Sy];
    A = [Coe S];
end

% make coefficient matrix in A
function Coe = makeCoe(j,m,C,X,Y)
    CoeX = makeCoeX(j,m,C,X,Y);
    CoeY = makeCoeY(j,m,C,X,Y);
    Coe = [CoeX; CoeY];
end
```

```
% make coefficient matrix x in coefficient matrix
function CoeX = makeCoeX(j,m,C,X,Y)
    CoeX = C;
    for i = 1:m
        CC = C(:,i);
        Col = find(CC==1)';
        CoeX(Col(1),i) = (X(Col(2))-X(Col(1)))/distance(Col(1),Col(2),X,Y);
        CoeX(Col(2),i) = (X(Col(1))-X(Col(2)))/distance(Col(1),Col(2),X,Y);
    end

end

% make coefficient matrix y in coefficient matrix
function CoeY = makeCoeY(j,m,C,X,Y)
    CoeY = C;
    for i = 1:m
        CC = C(:,i);
        Col = find(CC==1)';
        CoeY(Col(1),i) = (Y(Col(2))-Y(Col(1)))/distance(Col(1),Col(2),X,Y);
        CoeY(Col(2),i) = (Y(Col(1))-Y(Col(2)))/distance(Col(1),Col(2),X,Y);
    end
end

function r = distance(m_1,m_x,X,Y)
    r = sqrt((X(m_x)-X(m_1))^2+(Y(m_x)-Y(m_1))^2);
end
```