Truss Design Project: Final Design Report

Prepared for

Professor Holt

Professor at Boston University

Prepared by

Just Truss Us

Team Members: Young-Chan Cho, Nicole Zacarias, and Susan Zhang

Engineering Students at Boston University

April 29th, 2022

ENG EK301: Engineering Mechanics I – Section A3
Spring 2022

Table of Contents

Introduction: The overall goal is to design a truss capable of supporting a predetermined load from given materials through the guidance of engineering analysis. As we approached the final stages of planning, we manually maximized a truss design's predicted strength and its load-to-cost ratio by tinkering with each joint position, and therefore the lengths of each member. Through slight changes in the original truss design, a structure capable of holding more weight — especially at the lower limit — and had a larger load-to-cost ratio was obtained.

Procedures: Under ideal circumstances, the optimized truss design would have been based on one from the preliminary design report. However, the spans of the two previous designs failed to meet the 32 in. span, due to a misinterpretation of the truss design constraints. We originally assumed that it was not required for the bottom part of the truss joints to be in line with one another, yet this was not true. As a result, we came up with a new base truss design (see Figure 2 in the Appendices) to then optimize. With this new base truss design in mind, we aimed to optimize the maximum weight it can support, which in turn helped maximize its load-to-cost ratio. By altering the joint positions of each member accordingly, a more optimal structure, with a greater maximum load supported, was acquired (see Figure 1).

Analysis: While our final truss design differs from the preliminary design report ones, our approach to choosing a preferred, optimal design, was the same. As we changed the joint positions to acquire a more optimal design, by simultaneously analyzing each updated version, we kept our priorities on having a greater maximum live load. Taking advantage of the pattern we noticed from previous design processes, we focused on reducing the number of compression members within the truss while also minimizing their lengths.The critical member, in particular, was the focus when optimizing to support more weight before failure. This also allowed for a higher load-to-cost ratio because shorter compression members resulted in a lower cost and a higher supported load, moving us further towards our goal. However, even though the maximum weight found using the nominal critical buckling force was increased, the lower limit of the strength of the truss should still be taken into account. The formula used to account for the uncertainty in critical buckling force is $P_{crit}(L) = P_{nom} \pm \Delta P$, where $\Delta P = 4.1$ oz, which is then used to calculate the maximum supportable weight. Our optimized design has a maximum lower bound weight limit of about 42.5 oz. compared to about 36.6 oz. from our original design.

Results:

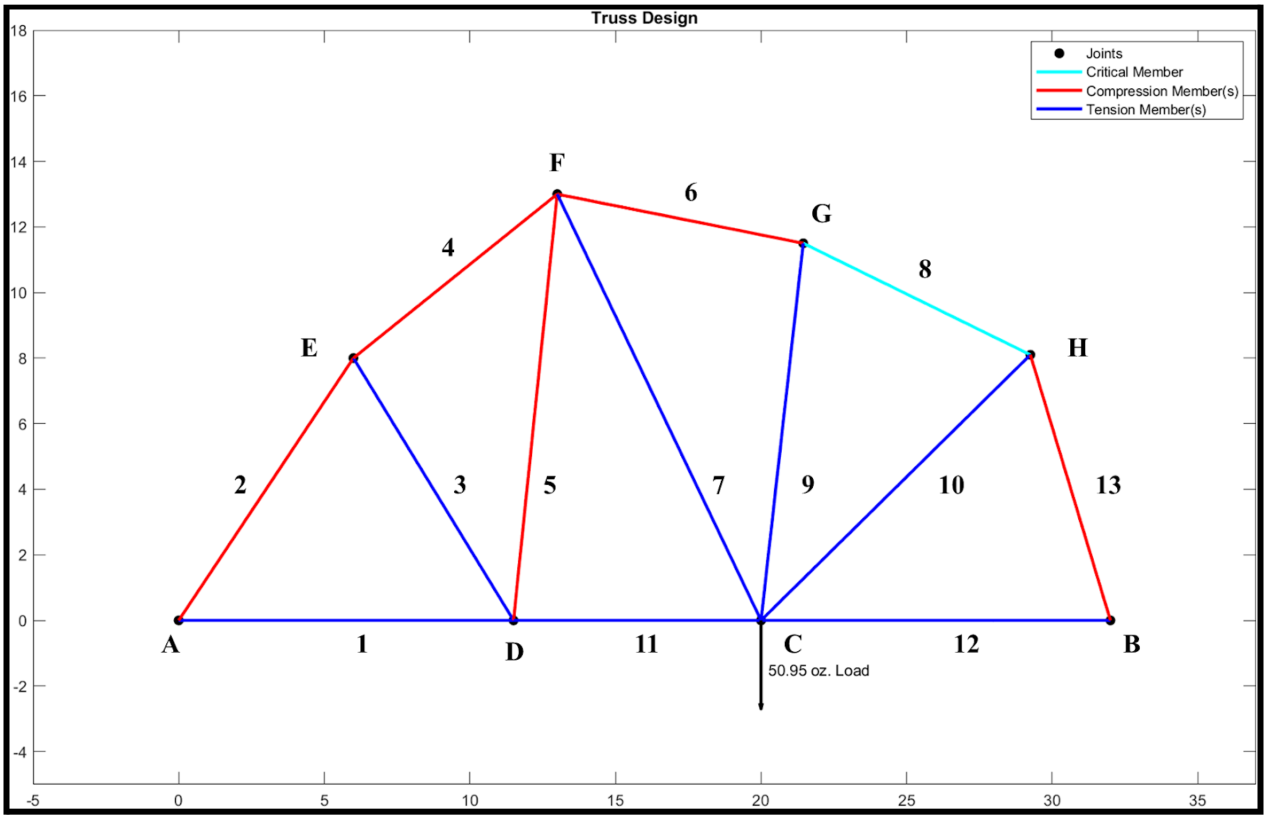

Figure 1: MATLAB generated diagram of optimized truss design with labeled joints and members

Member Number	Member	Joint-to-Joint Member Length (in.)	Tension / Compression	Buckling Strength (oz.)	Magnitude of Force at Max Truss Load (oz.)
1	AD	11.5	T	N/A	14.3293
2	AE	10	C	-24.0422 ± 4.1	23.8822
3	DE	9.7082	T	N/A	7.2194
4	EF	8.6023	C	-33.5382 ± 4.1	22.6355
5	DF	13.0863	C	-13.2647 ± 4.1	5.9886
6	FG	8.5821	C	-33.7131 ± 4.1	33.008
7	CF	14.7648	T	N/A	28.2519
8	GH	8.5088	C	-34.3585 ± 4.1	34.3585
9	CG	11.5911	T	N/A	8.023
10	CH	12.2952	T	N/A	27.4953
11	CD	8.5	T	N/A	19.1057
12	BC	12	T	N/A	10.8109
13	BH	8.5541	C	-33.9577 ± 4.1	33.628

Table 1: Description of optimized truss design member's connected joints, whether it is in tension or compression, and its magnitude of force experience (critical member highlighted in blue)

Maximum Load (oz.)	Maximum Load Range (oz.)	Total Length of all Members (in.)	Truss Cost	Load-to-Cost Ratio (oz. / $)
50.9486	42.5435 - 57.0283	137.6929	$217.69	0.234

Table 2: Summary of the results from the optimized truss design

Discussion and Conclusion:

Our final design's rationale stems from attempting to maximize the truss' maximum live load while also adhering to the restrictive constraints. Since the two structures presented during the preliminary design process were disqualified for not satisfying given criteria, a new base design that met the requirements was achieved on paper. It was then optimized through joint location modifications. The original truss design had a large diamond shape separated into two triangles near the loading joint and consisted of 7 compression members. Through slight alterations, the triangles near the loading joint were reduced in size, the number of members in compression was reduced to 6, and the critical member was changed to member 8 — whose length was then further minimized. Additionally, the final design looked sturdier and more physically attainable overall, while also surpassing the maximum live load and load-to-cost ratio of the base design. Through the process of truss designing, we were able to hone our skills in MATLAB, learn how to use computational analysis to facilitate the engineering design process, and connect classroom concepts to real-life applications. Nonetheless, if the project constraints were more flexible, there would have been more room for variety in truss designs and their respective maximum live load — allowing students to not only be ingenious but also more creative.

Appendices:

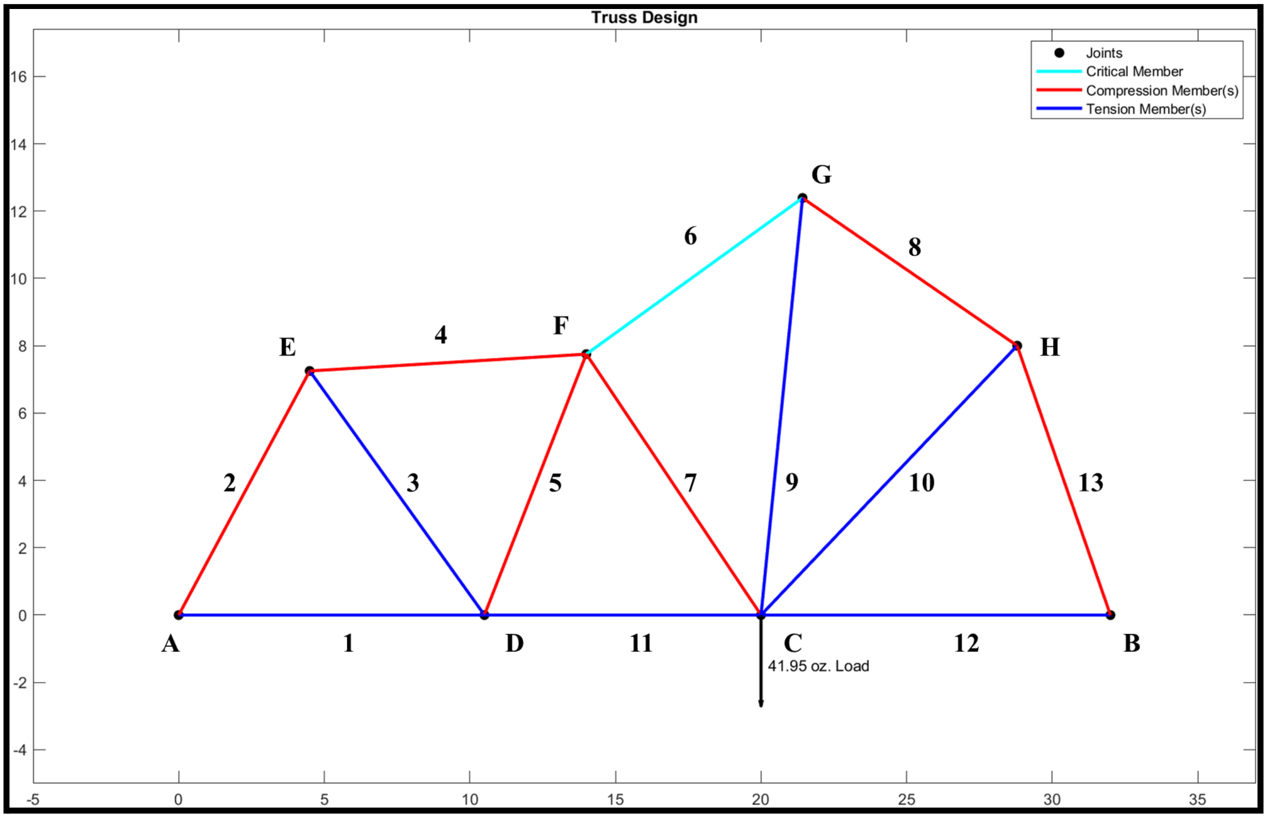

Figure 2: MATLAB generated diagram of original truss design with labeled joints and members

Member Number	Member	Joint-to-Joint Member Length (in.)	Tension / Compression	Buckling Strength (oz.)	Magnitude of Force at Max Truss Load (oz.)
1	AD	10.5	T	N/A	9.7639
2	AE	8.533	C	-34.1434 ± 4.1	18.5145
3	DE	9.4108	T	N/A	18.9275
4	EF	9.5131	C	-26.8473 ± 4.1	21.8616
5	DF	8.5037	C	-34.4044 ± 4.1	15.9997
6	FG	8.7546	C	-32.2622 ± 4.1	32.2622
7	CF	9.8011	C	-25.134 ± 4.1	1.7226
8	GH	8.579	C	-33.7401 ± 4.1	27.6612
9	CG	12.4699	T	N/A	31.4476
10	CH	11.8904	T	N/A	17.9378

11	CD	9.5	T	N/A	28.4167
12	BC	12	T	N/A	10.498
13	BH	8.6175	C	-33.4077 ± 4.1	28.2414

Table 3: Description of original truss design member's connected joints, whether it is in tension or compression, and its magnitude of force experience (critical member highlighted in blue)

Maximum Load (oz.)	Maximum Load Range (oz.)	Total Length of all Members (in.)	Truss Cost	Load-to-Cost Ratio (oz. / $)
41.9487	36.6175 - 47.2794	128.0731	$208.07	0.2016

Table 4: Summary of the results from the original truss design

_______________Hartford Civic Center Area Collapse Meeting Minutes_______________

Date: April 23rd, 2022

Time: 8:00 P.M. - 9:30 P.M.

Location: George Sherman Union

Chair: Nicole Zacarias

Recorder: Susan Zhang

Participants: YoungChan Cho, Nicole Zacarias, Susan Zhang

Planned Agenda:

1) Briefly summarize the case study

2) Discuss the dynamics of the engineers in the case study

3) Assess the strengths and weaknesses of the case study group

4) Discuss what we can take away from the case study

Important Points (Note Who Said What):

- Nicole: At the beginning, the inspection agency warned the engineers about “excessive deflections,” but this was ignored. Some time after, the deflection issue was brought up again when the roof deck was being installed and seemed to have a deflection measure of two times the predicted amount from their computer analysis.
- Susan: That’s right, and then the engineers said the discrepancy was fine and expected, which is very odd and irresponsible of them.
- YoungChan: If I remember correctly, the subcontractor also notified them about the difficulties this was causing, but were told to fix it otherwise they would be responsible for any delays this would cause.

- Nicole: Yes, that sounds about right. I think after that point in the text, it says how a citizen expressed concern for what they saw to be a downward deflection in the roof after it was put in place.
- Susan: That citizen's concern was ignored and got told that the roof was indeed safe when it was very clearly not. It provided a false sense of security.
- YoungChan: That was unfortunate and what's worse is that the roof did collapse 4 years later.
- Nicole: The first company to oversee this collapse thought it was due to the impactful snowstorm and compared what they thought was the roof's acceptable live load to what the snow's was.
- Susan: That company was Lev Zetlin. Another company that investigated the case is Loomis & Loomis. They thought the collapse was because of torsional buckling.
- YoungChan: Going back to Lev Zetlin, they also said that the roof's member slender ratios violated the American Institute of Steel Construction code provision.
- Nicole: Oh right, we forgot to mention that about Lev Zetlin. Then, Loomis & Loomis said that the members were already approaching its torsional buckling capacity. They got that from their computer analysis.
- YoungChan: Near the end of the text, it said the construction manager avoided hiring a structural engineer in order to not "waste money."
- Nicole: Yeah, it was so frustrating to read that the construction manager did not take responsibility for the collapse even though he was the one to oversee the project as a whole and coordinate the different subcontractors who helped in the design and building process.

- YoungChan: That can be considered a contradiction of Section III.8 of the Code of Ethics for Engineers which says that “Engineers shall accept personal responsibility for their professional activities.”
- Susan: I agree, I felt like he took on the leadership role, and did not take accountability for the result of the project which is very unprofessional of him. And it was even more concerning that the case was basically swept under the rug since it ended up being an out of court settlement. No one was punished for the collapse that threatened the lives of so many. They were lucky that it collapsed when there was no one in the arena, but if it had happened just a few hours earlier, it would have been a catastrophe.
- YoungChan: He also refused to hire a structural engineer even though no one else on the team was competent enough in that field of expertise to have provided insight into the overall sturdiness of the roof, including himself.
- Nicole: That is concerning since it violated Section II.2.b of the Code of Ethics for Engineers which basically states that “Engineers shall perform services only in the areas of their competence” and shall not sign off on any plans or documents dealing with subject matter not within their expertise. The construction manager shouldn’t have gone further with the project when he wasn’t able to guarantee the reliability of the design.
- Susan: I believe that the safety of civilians should have been prioritized over saving money in designs that’ll be used and affect the general public - although the design the engineers went with was cost effective, it was not a stable structure.
- Nicole: Their actions violated Section II.1 of the Code of Ethics for Engineers which states that “Engineers shall hold paramount the safety, health, and welfare of the public.”

- YoungChan: Although computer aided analysis is useful in helping to determine if designs are practical and fit within the constraints of the overall project, the designs should have been tested and evaluated thoroughly before implementing. Computers can only predict so much and may not consider all factors of real life that goes into how useful or strong the design may be when tested. It is so important to acknowledge that computers can help in the general design and idea process, but should not take priority over how the design reacts when physically constructed.
- Susan: Computer aided designs often assume ideal situations, but more often than not, ideal situations are not achievable in reality. Discrepancies in the theoretical and actual designs shouldn't be overlooked and should've been warnings to the engineers of the divergence from what they think is going to happen, which should be taken into account before proceeding and continuing with the project.
- Nicole: Also, the opinions expressed by others should not be disregarded. Ideas and concerns voiced by peers could help in making sure that the project goals will be reached. Because the engineers were overly confident in their design, due to the confirmation of their computer analysis, they ignored the signs of failure and the worries pointed out by professionals.

Conclusions:

From the case study, we acknowledge the potential dangers of being overly confident and reliant on computational design analysis. Thus, we will be considerate of other people's opinions, such as peer reviews and public concern, and be mindful of deviations from the theoretical design. Therefore, we will be model engineers by following the Code of Ethics for Engineers to the best of our abilities — to provide the most optimal solutions to given challenges.

MATLAB Code (as a .mlx live script file)**:**

Truss Design Project: Preliminary Design

```
clear all

clear

close all

clc
```

Read in .mat File with the Values of the C, Sx, Sy, X and Y, and L Matrices

```
filename = input('Please enter .mat file name without the extension: ','s');

filename = strcat(filename, '.mat');

load(filename)
```

Find out the Units and Convert to Oz. and In. if Necessary

```
Lunits = input('What units are the lengths in (m or in): ','s');
```

```
while ((Lunits ~= "m") && (Lunits ~= "in")) %Converting to inches if it is not already in inches

   Lunits = input('Invalid units, please enter m for meters or in for inches!\nWhat units are the lengths in (m or in): ','s');

end
```

```
if (Lunits == "m")

  X = X * 39.3701;

  Y = Y * 39.3701;

end
```

```
Funits = input('What units are the forces in (N, lb, or oz): ','s');


while ((Funits ~= "N") && (Funits ~= "lb") && (Funits ~= "oz"))
   Funits = input('Invalid units, please enter N for Newtons, lb for pounds, or oz for ounces!\nWhat units are the lengths in (N, lb, or oz): ','s');
end


if (Funits == "N") %Converting to ounces if it is not already in ounces
   L = L * 3.5969431019354;
elseif (Funits == "lb")
   L = L * 16;
end


Funits = 'oz.';
Lunits = 'in.';


%j rows (joints) x m columns (members)
[j, m] = size(C);
```

Method of Joints (to Determine the Forces at Each Joint)

```
%Creating Matrix Ax and Ay to Hold the Coefficients of the Force for the
%Respective Member Tension at Each Joint (in terms of x and y Components)
Ax = zeros((j),(m));     %Allocating Matrix A of Size j Rows x m Columns
Ay = zeros((j),(m));     %Allocating Matrix A of Size j Rows x m Columns
```

```
memlength = zeros(m,1);     %To hold the total length of all of the members of the truss


for i = 1:1:m %Loop through to analyze each of the 13 members (so that we can see how it is interacting with the joint)

       y = (find(C(:,i)))'; %Finds the indices of when C == 1 so that we know which joint it interacts with (1 = interacts with joint)

       [rows,columns] = size(y); %We know that the members of the truss are all 2 force members so size should always be (1,2) => this line is for sanity check


       if (sqrt(((X(max(y))-(X(min(y))))^2)+((Y(max(y)))-(Y(min(y))))^2) == 0) %To avoid a 0 in the denominator

            Ax(min(y),i) = 0;

            Ax(max(y),i) = 0;

            Ay(min(y),i) = 0;

            Ay(max(y),i) = 0;

       else

           Ax(min(y),i) = (X(max(y))-X(min(y)))/(sqrt(((X(max(y))-(X(min(y))))^2)+((Y(max(y)))-(Y(min(y))))^2));

           Ax(max(y),i) = (X(min(y))-X(max(y)))/(sqrt(((X(max(y))-(X(min(y))))^2)+((Y(max(y)))-(Y(min(y))))^2));

           Ay(min(y),i) = (Y(max(y))-Y(min(y)))/(sqrt(((X(max(y))-(X(min(y))))^2)+((Y(max(y)))-(Y(min(y))))^2));

           Ay(max(y),i) = (Y(min(y))-Y(max(y)))/(sqrt(((X(max(y))-(X(min(y))))^2)+((Y(max(y)))-(Y(min(y))))^2));

       end
```

```
       memlength(i,1) =
sqrt(((X(max(y))-(X(min(y))))^2)+((Y(max(y)))-(Y(min(y))))^2);
end
```

```
%Adding matrix S to the end of A
Ax = [Ax, Sx];
Ay = [Ay, Sy];
A = [Ax;Ay];
T = inv(A)*L;
```

```
R = T/sum(L);
```

```
Total_Length = sum(memlength);
```

Finding the Critical Member and the Maximum Load the Truss can Handle

```
Ttension = find((T(1:(length(T)-3)) > 0)); %Vector containing indices of
members in tension
Tcompression = find((T(1:(length(T)-3)) < 0)); %Vector containing
indices of members in compression
Tzero = find((T(1:(length(T)-3)) == 0)); %Vector containing indices of
zero force members
```

```
Fbuckle = 3908.184 .* ((memlength(Tcompression,1)).^(-1 * 2.211));
%Finding the Pcrit value for the members in compression with given
equation
Fbuckle_low = Fbuckle - 4.1; %Lower limit of Pcrit
Fbuckle_high = Fbuckle + 4.1; %Upper limit of Pcrit
```

```
Fbuckle = -1 * Fbuckle;
Fbuckle_low = -1 * Fbuckle_low;
Fbuckle_high = -1 * Fbuckle_high;


Weight = Fbuckle./R(Tcompression);
Weight_low = Fbuckle_low./R(Tcompression);
Weight_high = Fbuckle_high./R(Tcompression);


Weight_failure = min(Weight);
Weight_failure_low = min(Weight_low);
Weight_failure_high = min(Weight_high);


critMember = Tcompression (find(Weight == min(Weight_failure)));
```

Calculating Cost of Truss

```
C1 = 10; %$10/joint
C2 = 1; %$1/inch.


Cost = (C1*j) + (C2 * Total_Length);


LoadtoCost = Weight_failure/Cost;
```

Plotting the Truss

```
% If the member is a critical member, the member will be colored red
figure (1)

```

```
for d = 1:1:j
   points = plot (X(d), Y(d), 'ko', 'MarkerFaceColor', 'k',
'MarkerSize', 6, 'DisplayName', 'Joints');
   hold on
end
```

```
for r = 1:1:(m)
   interacts = (find(C(:,r)))'; %Finds the indices of when C == 1 so
that we know which joint it interacts with (1 = interacts with joint)
   [rows,columns] = size(interacts); %We know that the members of the
truss are all 2 force members so size should always be (1,2) => this
line is for sanity check
```

```
   if (r == critMember)
      crit = plot (X([interacts(1),interacts(2)]),
Y([interacts(1),interacts(2)]), 'c', 'LineWidth', 2,
'DisplayName','Critical Member');
   elseif (T(r) < 0)
      compress = plot (X([interacts(1),interacts(2)]),
Y([interacts(1),interacts(2)]), 'r', 'LineWidth', 2,
'DisplayName','Compression Member(s)');
   elseif (T(r) > 0)
      tens = plot (X([interacts(1),interacts(2)]),
Y([interacts(1),interacts(2)]), 'b', 'LineWidth', 2,
'DisplayName','Tension Member(s)');
   else
       zfm = plot (X([interacts(1),interacts(2)]),
Y([interacts(1),interacts(2)]), 'k', 'LineWidth', 2,
'DisplayName','Zero-Force Member(s)');
   end
```

```
    hold on
end

xlim ([min(X) - 5, max(X) + 5])
ylim ([min(Y) - 5, max(Y) + 5])

Arrow_start = [(X(find(L) - j)), (Y(find(L) - j))];
Arrow_end = [(X(find(L) - j)), (Y(find(L) - j) - 3)];
dp = Arrow_end - Arrow_start;

h = quiver(Arrow_start(1), Arrow_start(2), dp(1), dp(2));
set(h,'LineWidth', 2, 'color',  '#000000');

text((Arrow_end(1) + 0.25),((Arrow_start(2) + Arrow_end(2))/2),
sprintf('%.2f %s Load',sum(L), Funits))

title ('Truss Design')
legend([points, crit, compress, tens], 'Joints', 'Critical Member',
'Compression Member(s)', 'Tension Member(s)')
```

Print the Output

```
fprintf('EK301 Section A3, Team "Just Truss Us": YoungChan Cho, Nicole
Zacarias, Susan Zhang, Date: April 10th, 2022')
%Assuming that the L read in is in terms of oz. for our truss
[a,b] = size (T);
```

```
%All values were rounded 2 decimal places => so is approximate, not
%necessarily exact
fprintf ('Load: %.2f %s', L(find(L)), Funits)
fprintf ('Member Forces in %s\n', Funits)
for q = 1:1:(a-3)
   TorC = '';
   if T(q,1) < 0
       TorC = '(C)';
   elseif T(q,1) > 0
        TorC = '(T)';
   end
   fprintf ('member %i: %.2f %s\n', q, abs(T(q,1)),TorC)
end
fprintf ('Reaction Forces in %s\n', Funits)
fprintf ('Sx1: %.2f\n', abs(T(a-2,1)))
fprintf ('Sy1: %.2f\n', abs(T(a-1,1)))
fprintf ('Sy2: %.2f\n', abs(T(a,1)))
```

```
fprintf('\nThe sum of the lengths of all the members of the truss is
about: %.2f %s\n', Total_Length, Lunits)
fprintf('\nThe total cost of the truss is about: $%.2f\n',Cost)
fprintf('\nThe maximum load that can be supported by this truss is
about: %.2f %s\n', Weight_failure, Funits)
fprintf('The maximum load that can be supported by this truss ranges
from about %.2f %s to about %.2f %s\n', Weight_failure_low, Funits,
Weight_failure_high, Funits)
fprintf('\nThe theoretical max load/cost ratio in oz/$ is about:
%.4f\n', LoadtoCost)
```

```
fprintf('\nThe critical member is: Member %i\n', critMember)

fprintf('The length of the critical member is about: %.2f %s\n',
memlength(critMember), Lunits)

fprintf('The predicted buckling strength of the critical member is
about: %.2f %s\n', abs(Fbuckle(find(Tcompression == critMember))),
Funits)

fprintf('The predicted buckling strength of the critical member ranges
from about %.2f %s to about %.2f %s\n',
abs(Fbuckle_low(find(Tcompression == critMember))), Funits,
abs(Fbuckle_high(find(Tcompression == critMember))), Funits)
```

Does it Meet Project Criteria?

```
fails_criteria = 0; %Will let us know at the end whether all of the
project criterias have been met
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%

%%

%% For a simple truss to be staticly deterministic, it must satisfy the

%% relationship of: # of Members = (2 * # of Joints) + # of Reaction
Forces

%%

%% # of reaction forces in our case is 3 => from 1 pin joint and 1
roller support

%%

%% Note: This is a necessary condition for statical determinancy, but is
not a sufficient

%% condition => even if a truss satifies the above relation, it may NOT
be determinate

%% However, if it is determinate, then it MUST satisfy the condition

%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

simple_truss = 0;


if (m ~= ((2*j)-3))

   fprintf('The number of joints (J) and members (M) are not related by the equation M = 2J - 3. This truss design is not be statically deterministic.\n')

   fails_criteria = fails_criteria + 1;

   simple_truss = 1;

end


if (((min(memlength) < 8.5)) || (max(memlength > 15)))

   fprintf('The member lengths of this truss design is not within the constraint boundaries of 8.5 inches <= Member Length <= 15 inches.\n')

   fails_criteria = fails_criteria + 1;

end


if ((max(X) - min(X)) ~= 32 )

   fprintf('The span of this truss design is not 32 inches.\n')

   fails_criteria = fails_criteria + 1;

end


if (Weight_failure < 32)

   fprintf('This truss design is unable to support 32 oz!\n')

   fails_criteria = fails_criteria + 1;

end
```

```
if (Cost >= 295)
   fprintf('The total virtual cost of this truss design exceeds the maximum budget of $294.99.\n')
   fails_criteria = fails_criteria + 1;
end
```

```
if (fails_criteria > 0)
   fprintf('This truss design does not satisfy the project criteria.\n')
else
   fprintf('This truss design satisfies the project criteria.\n')
end
```

Finding Optimization of Truss Connection Matrix if it is a Simple Truss

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% If the Truss Design is Statistically Deterministic, we will try to optimize the design
%% of the truss
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (simple_truss == 0)
   [X_optimal, Y_optimal, Cost_optimal, Weight_optimal, Weight_optimal_low, LoadtoCost_optimal, iterations] = POS (C, L, Sx, Sy, X, Y)
else
```

```
    fprintf("Your truss may not be statistically deterministic or simple because it does not follow the relationship of M = 2J - 3. The truss design will not be optimized.")
end
```

Original Truss Design Matrix Inputs (as a .m script file)**:**

```
%% Connection Matrix
C = [  1   1   0   0   0   0   0   0   0   0   0   0   0;
       0   0   0   0   0   0   0   0   0   0   0   1   1;
       0   0   0   0   0   0   1   0   1   1   1   1   0;
       1   0   1   0   1   0   0   0   0   0   1   0   0;
       0   1   1   1   0   0   0   0   0   0   0   0   0;
       0   0   0   1   1   1   1   0   0   0   0   0   0;
       0   0   0   0   0   1   0   1   1   0   0   0   0;
       0   0   0   0   0   0   0   1   0   1   0   0   1];


%% Support Forces Matrix
Sx = [ 1   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0];

Sy = [ 0   1   0;
       0   0   1;
       0   0   0;
       0   0   0;
```

```
        0    0    0;
        0    0    0;
        0    0    0;
        0    0    0];


%% Position Vectors
X = [ 0,  32,  20, 10.5, 4.5,   14,   21.4249,  28.7967];
Y = [ 0,   0,   0,   0,  7.25,  7.75, 12.3882,      8];

L = [ 0;  0;  0;  0;  0;  0;  0;  0;  0;  0;  41.9487;  0;   0;  0;  0;
0];


%% Saving to a .mat File
save('TrussDesign_YoungChanNicoleSusan_A3.mat', 'C', 'Sx', 'Sy', 'X', 'Y',
'L');
```

Optimized Truss Design Matrix Inputs (as a .m script file)**:**

```
%% Connection Matrix
C = [  1   1   0   0   0   0   0   0   0   0   0   0   0;
       0   0   0   0   0   0   0   0   0   0   0   1   1;
       0   0   0   0   0   0   1   0   1   1   1   1   0;
       1   0   1   0   1   0   0   0   0   0   1   0   0;
       0   1   1   1   0   0   0   0   0   0   0   0   0;
       0   0   0   1   1   1   1   0   0   0   0   0   0;
       0   0   0   0   0   1   0   1   1   0   0   0   0;
       0   0   0   0   0   0   0   1   0   1   0   0   1];


%% Support Forces Matrix
Sx = [ 1   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0;
       0   0   0];

Sy = [ 0   1   0;
       0   0   1;
       0   0   0;
       0   0   0;
```

```
      0   0   0;
      0   0   0;
      0   0   0;
      0   0   0];


%% Position Vectors
X = [ 0   32   20  11.5  6  13  21.45   29.25];
Y = [ 0,   0,   0,   0,  8, 13, 11.5,     8.1];

L = [ 0;  0;  0;  0;  0;  0;  0;  0;  0;  0; 50.9486;  0;   0;  0;  0;
0];


%% Saving to a .mat File
save('OptimizedTrussDesign_YoungChanNicoleSusan_A3.mat', 'C', 'Sx', 'Sy',
'X', 'Y', 'L');
```