Boston University

EK301-A1

Professor Farny

Spring 2025

Preliminary Design Lab Report

Jinyu Fang, Jessica Qiu, Riasat Audhy

Due Date: April 8, 2025

Submitted: April 8, 2025

I. Introduction

For our preliminary design report, this project aims to use computational analysis to evaluate and optimize truss designs, helping ensure they meet specified load and cost requirements. By using a computer program, we can efficiently calculate internal member forces, identify which members are most at risk of buckling, and assess the overall cost-effectiveness of each design. This method allows us to explore multiple configurations, compare their strengths and weaknesses, and ultimately select a design that offers the best balance between performance and cost. In this report, we verify our computational model and present two preliminary truss designs for comparison.

A simple truss, which is the focus of this project, relies on members that experience only tension or compression. This design approach uses material efficiently while maintaining stability. The structure's strength comes from forming triangular units, with members connected only at their endpoints. A stable, statically determinate truss follows Equation (1):

$$M = 2J - 3 \tag{1}$$

where M is the number of members and J is the number of joints. There are exactly 2J equations that can be obtained through equilibrium analysis of the horizontal and vertical forces at each joint, which means there must be exactly 2J unknown forces for the system to be solvable. There are M unknown forces within the members, and 3 unknown reaction forces from the pin (horizontal and vertical) and rocker (vertical) supports. Setting these equal and rearranging provides Equation (1) to determine the number of members needed in the truss.

For this project, our truss must span 31 inches and support a 32 oz. load placed 8.5-12 inches from the pinned support. Additionally, each member must be between 8.5 and 14 inches long. Cost is another key constraint: the total virtual cost must remain under $315, calculated using $10 per joint and $1 per inch of total member length. In developing our designs, we focused on the following key design principles:

1. Triangular geometry to maximize strength and stability.
2. Symmetry to ensure even load distribution and reduce stress imbalances.
3. Material efficiency by limiting the number of joints and total member length.
4. Shorter compression members to reduce the risk of buckling and increase load capacity.

These principles guided our design choices as we explored different truss layouts. In the sections that follow, we verify our analysis method and present a detailed comparison of our two preliminary designs.

II. Method and Analysis

In the Appendix A.1, the MATLAB script was developed to analyze multiple truss designs efficiently. It accepts user input for the number of joints, member-to-joint connections, support force locations, joint coordinates, and applied loads. Using this data, the code constructs the connection matrix (C) and support matrices, then calculates matrix A, which contains the direction cosines of each truss member based on the geometry and connectivity of the structure. The system of equations representing static equilibrium is then solved using the formula T = inv(A) * L, where T is the vector of internal member and support forces, and L is the external load vector. Additionally, the code estimates the maximum theoretical load the truss can support before any member fails due to buckling, and evaluates the cost-efficiency of the design by computing the load-to-cost ratio. A negative value corresponds to compression force, and a positive value corresponds to tension force. As a result, our result by hand matches our result calculated by computer code, indicating the accuracy of our computational code.

Our approach was consistent with the procedure described in the lab manual. First, we analyzed the diagram and numbered each joint (j) and each member (m) of the truss. From there, we created six separate matrices:

C Matrix- This matrix has j rows for joints and m columns for members. If a joint is connected to a member, a '1' is placed in the intersecting cell; all other cells contain '0'. We verified the matrix by ensuring each column has exactly two '1's.

$$\mathbf{C}_{j,m} = \begin{cases} 1, & \text{if member } m \text{ is connected to joint } j \\ 0, & \text{elsewhere} \end{cases}$$

Sx Matrix- A matrix with j rows and three columns, capturing only the reaction forces at the pin joint and roller joint at the base of the truss, specifically in the x-direction.

$$\mathbf{S_x} = \begin{array}{c} \begin{matrix} S_{x1} & S_{y1} & S_{y2} \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

Sy Matrix- Similar in size and purpose to Sx, but focused on y-axis interactions. Column 1 is filled with 0s, while column 2 has entries for the pin joint and roller joint.

$$\mathbf{S_y} = \begin{array}{c} \begin{matrix} S_{x1} & S_{y1} & S_{y2} \end{matrix} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

X Matrix- 1 x j matrix that stores the x-coordinates for each joint. The origin (0,0) was set at the bottom-left joint for this analysis.

$$\mathbf{X} = [x_1, x_2, x_3, x_4, ...]$$

Y Matrix- 1 x j matrix that stores the y-coordinates for each joint, using the same origin as the `X` matrix.

$$\mathbf{Y} = [y_1, y_2, y_3, y_4, ...]$$

L Matrix- 2j x 1 matrix to store the location and magnitude of external forces on the truss. The first j rows represent forces in the x-direction (none for this project), while the second j rows contain y-direction forces, where we recorded the weight's magnitude (in oz.) at the appropriate joint + j row.

$$\mathbf{L} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ +mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{array}{l} \text{no horizontal load at J1} \\ \text{no horizontal load at J2} \\ \text{...} \\ \text{...} \\ \text{...} \\ \text{no vertical load at J1} \\ \text{load } mg \text{ at J2} \\ \text{no vertical load at J3} \\ \text{...} \\ \text{...} \end{array}$$

With our matrices ready, we built the ‘A’ matrix, a 2j x (m+3) matrix. The first ‘j’ rows represent interactions in the x direction, and the second `j` rows cover interactions in the y direction. Each cell is filled with the force coefficients corresponding to the member tension at each joint, starting with forces along the x-axis in rows 1 to ‘j’ and ending with forces along the y-axis in rows ‘j+1’ to ‘2j’

$$\begin{aligned} \sum F_{x,1} \quad : \quad & \left(\frac{x_2 - x_1}{r_{1,2}}\right) T_1 + \left(\frac{x_3 - x_1}{r_{1,3}}\right) T_2 + 0 \cdot T_3 + 0 \cdot T_4 \\ & + 0 \cdot T_5 + 0 \cdot T_6 + 0 \cdot T_7 + 1 \cdot S_{x,1} + 0 \cdot S_{y,1} + 0 \cdot S_{y,2} = 0 \end{aligned}$$

$$\begin{aligned} \sum F_{y,1} \quad : \quad & \left(\frac{y_2 - y_1}{r_{1,2}}\right) T_1 + \left(\frac{y_3 - y_1}{r_{1,3}}\right) T_2 + 0 \cdot T_3 + 0 \cdot T_4 \\ & + 0 \cdot T_5 + 0 \cdot T_6 + 0 \cdot T_7 + 0 \cdot S_{x,1} + 1 \cdot S_{y,1} + 0 \cdot S_{y,2} = 0 \end{aligned}$$

Then using the values in A and L, we solved for the unknown forces of T:

$$\mathbf{T} = \mathbf{A}^{-1}(\mathbf{L}). \tag{2}$$

To validate the accuracy of the truss analysis program, a sample truss was analyzed manually, and its parameters (joint locations, member connections, and live load) were input into the program. The results generated by the program (Fig. 2) were compared with the manual calculations (Fig. 1), and both methods yielded identical results. This confirmed that the program was capable of accurately calculating the forces acting on each truss member.

Following this validation, the main truss design process was initiated using an iterative approach made possible by automating the analysis. In line with the given specifications, two truss designs were created. Each design focused on minimizing member length to enhance buckling strength while ensuring that members followed the parameters of the given project and support a 32 oz load:

Summary of distance and cost specifications

Joint-to-joint span L_{jj}	$8.5\ in. \leq L_{jj} \leq 14\ in.$
Truss span	$31\ in.$
Load to pin support span	8.5-12 $in.$
Total virtual cost	$< \$315$

$$\text{Cost} \quad = \quad C_1\, J + C_2\, L, \qquad \text{where: } C_1 = \$10/\text{joint} \quad \& \quad C_2 = \$1/\ in.$$

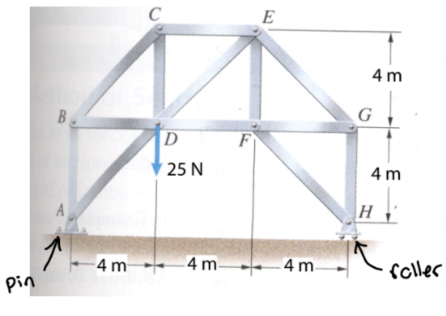
With this information we are able to run analysis on varying designs to determine, max force load, critical members, load to cot ratio, as well as calculate the overall cost of the trust.

By hand:

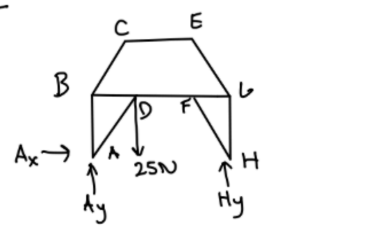
BOSTON UNIVERSITY
College of Engineering
EK 301 Engineering Mechanics I

Truss project computational method validation problem

Determine the loads in each of the members and whether they are in tension or compression. Analyze the loads yourselves using standard equilibrium analysis, and MATLAB (results should match!).



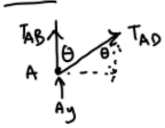
FBD



$\Sigma F_x = A_x = 0$
$\Sigma F_y = A_y + H_y - 25 = 0$
$\Sigma M_A = -25(4) + H_y \cdot 12 = 0$

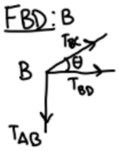
$H_y = 8.33$ N
$A_y = 16.67$ N
$A_x = 0$ N

FBD: A



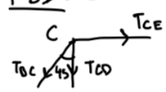
$\theta = \tan^{-1}\left(\frac{4}{4}\right) = 45°$
$\Sigma F_x = T_{AD} \sin\theta = 0$
$\Sigma F_y = T_{AD} \cos\theta + T_{AB} + A_y = 0$
$T_{AD} = 0$, $A_y = 16.67$ N
$T_{AB} = -16.67$ N

FBD: B



$\theta = \tan^{-1}\left(\frac{4}{4}\right) = 45°$
$\Sigma F_x = T_{BC} \cos\theta + T_{BD} = 0$
$\Sigma F_y = T_{BC} \sin\theta - T_{AB} = 0$
$-16.67 = T_{BC} \sin 45$
$T_{BC} = -23.57$ N, $T_{BD} = 16.67$ N

FBD: C

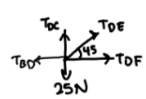


$\Sigma F_x = T_{CE} - T_{BC} \sin 45° = 0$
$\Sigma F_y = -T_{BC} \cos 45° - T_{CD} = 0$
$T_{CE} = -16.67$ N
$T_{CD} = 16.67$ N

FBD: H

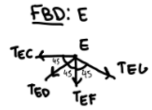
$\Sigma F_x: -T_{HF} \sin 45 = 0$
$T_{HF} = 0$ N
$\Sigma F_y: T_{HG} + H_y = 0$
$T_{HG} = -H_y$
$T_{HG} = -8.33$ N

FBD: D



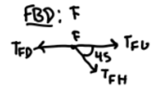
$\Sigma F_x: T_{DF} + T_{DE} \cos 45 - T_{BD} = 0$
$\Sigma F_y: T_{DC} + T_{DE} \sin 45 - 25 = 0$
$T_{DE} = 11.79$ N
$T_{DF} = 8.33$ N

FBD: E



$\Sigma F_x: T_{EG} \cos 45 - T_{ED} \cos 45 - T_{EC} = 0$
$T_{EG} = -11.79$ N
$\Sigma F_y: -T_{ED} \sin 45 - T_{EG} \sin 45 - T_{EF} = 0$
$T_{EF} = 0$ N

FBD: F



$\Sigma F_x: -T_{FD} + T_{FG} + T_{FH} \cos 45 = 0$
$\Sigma F_y: -T_{FH} \sin 45° = 0$
$T_{FH} = 0$ N
$T_{FG} = 8.33$ N

Results:

Force in member:

1. AB = 16.7 N in Compression
2. AD = 0 N
3. BC = 23.6 N in Compression
4. BD = 16.7 N in tension
5. CD = 16.7 N in tension
6. CE = 16.7 N in Compression
7. DE = 11.8 N in tension
8. DF = 8.33 N in tension
9. EF = 0 N
10. EG = 11.8 N in Compression
11. FG = 8.33 N in tension
12. FH = 0 N
13. GH = 8.33 N in Compression

Figure 1: Calculation performed for "Example Problem"

By computer code:

```
EK301, Section A1, Group6: Jessica Q. Jinyu F. Riasat A., 3/22/2025
Load: 25 oz
Member forces in oz:
m1: -16.667 (C)
m2: -0.000 (C)
m3: -23.570 (C)
m4: 16.667 (T)
m5: 16.667 (T)
m6: -16.667 (C)
m7: 11.785 (T)
m8: 8.333 (T)
m9: 0.000 (T)
m10: -11.785 (C)
m11: 8.333 (T)
m12: -0.000 (C)
m13: -8.333 (C)
Reaction forces in oz:
Sx1: 0.000
Sy1: 16.667
Sy2: 8.333
```

Figure 2: Example MATLAB Program Output

Figure 2 includes the final computer code output producing the information on the load applied, forces in each member, reaction forces, cost of the truss, and the theoretical max load/cost ratio. Given the 25 N load at joint D, the forces between each member are shown in *Table 2* as a final result.

Table 2: The forces of each member when experiencing 25N load

Member	Tension/Compression	Magnitude of the Force (N)
AB	Compression	16.7
AD	-	0.0
BC	Compression	23,6
BD	Tension	16.7
CD	Tension	16.7
CE	Compression	16.7
DE	Tension	11.8
DF	Compression	8.33
EF	-	0
EG	Compression	11.8
FG	Tension	8.33
FH	-	0
GH	Compression	8.33

III. Data & Design

<u>Design #1:</u>

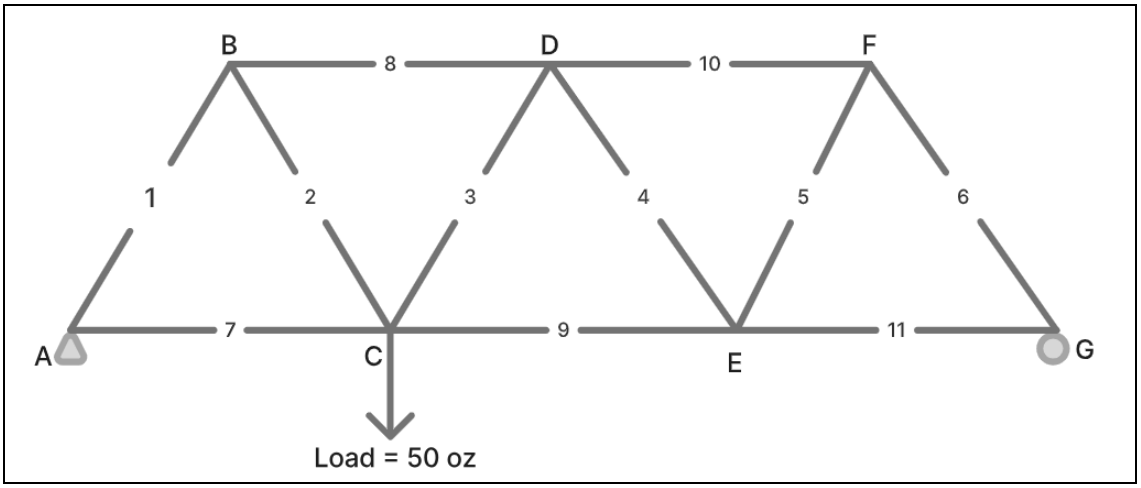


Figure 5: First Truss Design with A as a pin support, G as a roller, and a load of 50 oz on joint C.

From the Acrylic Data Fit provided, the fit error bar is a constant that results in U_fit = ± 1.35 oz (95%). Uncertainty was calculated based on this value, as a 5% of the buckling strength for each member. Using the code in Appendix A.2, we get the result in Table 3.

Table 3: Internal member forces, lengths, and buckling strengths for Design 1 under a 50 oz load.

Member Number	Member Length (inches)	Tension/ Compression	Buckling Strength and Uncertainty (± oz)	Magnitude of the Force at Max Truss load (oz)
1	9.304	Compression	42.09 ± 1.35	40.239
2	9.304	Tension	–	40.239
3	9.304	Tension	–	17.82
4	9.304	Compression	42.09 ± 1.35	17.82
5	9.343	Tension	–	18.069
6	10.63	Compression	33.06 ± 1.35	20.36
7	9.50	Tension	–	20.59

8	9.50	Compression	40.53 ± 1.35	41.179
9	9.50	Tension	–	32.082
10	9.75	Compression	38.66 ± 1.35	22.984
11	12.00	Tension	–	13.407

Maximum load: 50 oz ± 1.35 oz
Member to buckle first: Member 8
Reaction forces in Oz:

- Sx1 (Pin at A): 0.000 oz
- Sy1 (Pin at A): 34.677 oz
- Sy2 (Roller at G): 15.323 oz

Cost of the truss: $177.53
Theoretical Max Load/ Cost Ratio in Oz/$: 0.282 oz/$

```
EK301, Section A1, Group6: Jessica Q. Jinyu F. Riasat A., 3/22/2025
------------------------------------------------------------
Load: 50 oz
Member forces in oz:
m1: -40.329 (C)
m2: 40.329 (T)
m3: 17.820 (T)
m4: -17.820 (C)
m5: 18.069 (T)
m6: -20.360 (C)
m7: 20.590 (T)
m8: -41.179 (C)
m9: 32.082 (T)
m10: -22.984 (C)
m11: 13.407 (T)

Reaction forces in oz:
Sx1: 0.000
Sy1: 34.677
Sy2: 15.323

------------------------------------------------------------
The member to buckle first is member 8
Length of that member: 9.50 in
Predicted buckling strength: 40.53 oz ± 1.35 oz
The maximum load that the physical truss could support is 50 oz ± 1.35 oz

------------------------------------------------------------
Cost of the truss: $177.53
Theoretical max load/cost ratio in oz/$: 0.282
------------------------------------------------------------
```

```
Member Lengths (in inches):
Member  1: 9.304 in
Member  2: 9.304 in
Member  3: 9.304 in
Member  4: 9.304 in
Member  5: 9.434 in
Member  6: 10.630 in
Member  7: 9.500 in
Member  8: 9.500 in
Member  9: 9.500 in
Member 10: 9.750 in
Member 11: 12.000 in

Buckling Strengths for Compression Members:
-----------------------------------------------------------------
| Member | Length (in)  | Force (oz)  | Buckling Strength ± Unc.
-----------------------------------------------------------------
|   m1   |     9.304    |   -40.329   |      42.09 ± 1.35     |
|   m4   |     9.304    |   -17.820   |      42.09 ± 1.35     |
|   m6   |    10.630    |   -20.360   |      33.06 ± 1.35     |
|   m8   |     9.500    |   -41.179   |      40.53 ± 1.35     |
|   m10  |     9.750    |   -22.984   |      38.66 ± 1.35     |
-----------------------------------------------------------------
```

Figure 6: Design 1 MATLAB Program Output

Design #2:

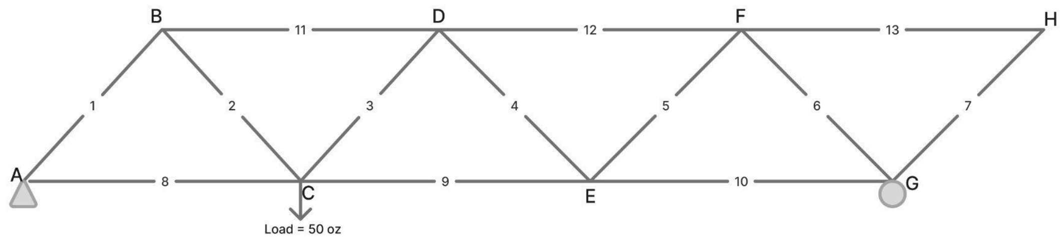


Figure 7: Second Truss Design with pin support at A, roller at G, and a load of 50 N on joint C.

From the Acrylic Data Fit provided, the fit error bar is a constant that results in U_fit = ± 1.35 oz (95% confidence). The uncertainty was calculated based on this value, as a 5% of the buckling strength for each member. Using the code provided in Appendix A.2, we get the result in Table 4.

Table 4: Internal member forces, lengths, and buckling strengths for Design 2 under a 50 oz load.

Member Number	Member Length (inches)	Tension/ Compression	Buckling Strength and Uncertainty (± oz)	Magnitude of the Force at Max Truss load (oz)
1	11.835	Compression	27.01 ± 1.35	27.551
2	11.835	Compression	27.01 ± 1.35	27.551
3	11.835	Tension	–	0
4	11.835	Compression	27.01 ± 1.35	0
5	11.835	Tension	–	0
6	11.835	Compression	27.01 ± 1.35	0
7	11.705	Tension	–	0
8	9.000	Tension	–	10.432
9	9.000	Tension	–	0

10	9.000	Tension	–	0
11	9.000	Compression	–	0
12	9.000	Tension	–	0
13	8.5	Compression	27.01 ± 1.35	0

Maximum load: 51 oz ± 1.35 oz
Member to buckle first: Member 2
Reaction forces in Oz:

- Sx1 (Pin at A): 0.000 oz
- Sy1 (Pin at A): 25.500 oz
- Sy2 (Roller at G): 25.500 oz

Cost of the truss: $216.51
Theoretical Max Load/ Cost Ratio in Oz/$: 0.236 oz/$

```
EK301, Section A1, Group6: Jessica Q. Jinyu F. Riasat A., 4/5/2025
-------------------------------------------------------------
Load: 51 oz
Member forces in oz:
m1: -27.551 (C)
m2: -27.551 (C)
m3: 0.000 (T)
m4: -0.000 (C)
m5: 0.000 (T)
m6: -0.000 (C)
m7: 0.000 (T)
m8: 10.432 (T)
m9: 0.000 (T)
m10: 0.000 (T)
m11: -0.000 (C)
m12: 0.000 (T)
m13: -0.000 (C)

Reaction forces in oz:
Sx1: -0.000
Sy1: 25.500
Sy2: 25.500

-------------------------------------------------------------
The member to buckle first is member 2
Length of that member: 11.88 in
Predicted buckling strength: 27.01 oz ± 1.35 oz
The maximum load that the physical truss could support is 51 oz ± 1.35 oz

-------------------------------------------------------------
Cost of the truss: $216.51
Theoretical max load/cost ratio in oz/$: 0.236
```

```
Member Lengths (in inches):
Member  1: 11.885 in
Member  2: 11.885 in
Member  3: 11.885 in
Member  4: 11.885 in
Member  5: 11.885 in
Member  6: 11.885 in
Member  7: 11.705 in
Member  8: 9.000 in
Member  9: 9.000 in
Member 10: 9.000 in
Member 11: 9.000 in
Member 12: 9.000 in
Member 13: 8.500 in

Buckling Strengths for Compression Members:
------------------------------------------------------------
| Member | Length (in) | Force (oz) | Buckling Strength ± Unc. |
------------------------------------------------------------
|   m1   |    11.885   |   -27.551  |      27.01 ± 1.35      |
|   m2   |    11.885   |   -27.551  |      27.01 ± 1.35      |
|   m4   |    11.885   |    -0.000  |      27.01 ± 1.35      |
|   m6   |    11.885   |    -0.000  |      27.01 ± 1.35      |
|   m13  |     8.500   |    -0.000  |      49.57 ± 1.35      |
------------------------------------------------------------
```

Figure 8: Design 2 MATLAB Program Output

IV. Discussion and Conclusion

We used both hand calculations and computational analysis to evaluate the internal member forces, buckling strengths, and cost-efficiency of truss structures. Our MATLAB code allowed us to verify the accuracy of manual calculations while enabling efficient evaluation of different truss configurations under the given design constraints. Our analysis of two preliminary truss designs allowed us to evaluate their stability, load capacity, cost-effectiveness, and overall structural efficiency. Both designs aimed to support a 32 oz load with minimal material cost while adhering to stability requirements and Equation (1).

Design 1 followed key structural principles such as symmetry, efficient triangulation, and relatively short compression members. The analysis showed that the truss could support a maximum load of 50 oz ± 1.35 oz before member 8 buckled. This member had a buckling strength of 40.53 oz, which was just barely exceeded by its internal compressive force of 41.179 oz. All other members remained safely within their strength limits. The cost of this truss was $177.53, leading to a theoretical load-to-cost ratio of 0.282 oz/$. These results indicate that Design 1 is structurally sound and efficient, though there is limited safety margin in the buckling capacity of some compression members. This suggests that minor improvements in geometry or member length distribution, such as shortening member 8 or distributing compressive forces more evenly, could further enhance the performance.

Design 2, while capable of supporting a similar maximum load of 51 oz ± 1.35 oz, did so at a significantly higher cost of $216.51, resulting in a lower load-to-cost ratio of 0.236 oz/$. The first member predicted to buckle was member 2, with a buckling strength of 27.01 oz and an internal force of 27.551 oz, again showing minimal margin before failure. Additionally, several members in Design 2 experienced no force, suggesting inefficient material usage and potential overdesign. This inefficiency in force distribution and member utilization contributed to a higher cost without an important increase in performance.

In conclusion, Design 1 is clearly the more cost-effective and structurally efficient option. It maintained a higher load-to-cost ratio, utilized more of its members effectively, and better adhered to key structural design principles like triangulation and load path continuity. Though both designs demonstrated the ability to meet the loading requirement, Design 1 did so with lower material use and cost, making it the preferred choice.

To improve future designs, we could implement optimization routines into our MATLAB code to adjust member lengths, joint positions, and loading points to maximize the load-to-cost ratio. Additionally, we could include a more robust buckling prediction model that accounts for material imperfections or misalignment, providing more realistic safety margins. Further testing could also assess the impact of small geometric changes on load distribution to avoid critical failures in individual members.

V. Appendix

A.1 — MATLAB Code Developed to Analyze Example Truss

Connection matrix C for joint

```
jointq = "How many joints? ";
numj = input(jointq);
numm = (2*numj) - 3;

C = zeros(numj, numm);

% Loop through each join and each member
for j = 1:numj;
    for m = 1:numm
        prompt = sprintf("Is member %d connected to joint %d? (1 for yes, 0 for no): ", m, j);
        C(j,m) = input(prompt);
    end
end
```

Connection matrix for support force Sx Sy

```
Sx1 = zeros(numj, 1);
Sy1 = zeros(numj, 1);
Sy2 = zeros(numj, 1);

Sx = zeros(numj, 3);
Sy = zeros(numj,3);

connection_matrixS_joint = zeros(numj,3);

% Loop through each joint and each force


for j = 1:numj
    support = sprintf("Is the x reaction force on joint %d? (1 for yes, 0 for no): ", j);
    Sx1 (j,1) = input(support);
end

for j = 1:numj
    support = sprintf("Is the first Y reaction force on joint %d? (1 for yes, 0 for no): ", j);
    Sy1 (j,1) = input(support);
end

for j = 1:numj
    support = sprintf("Is the second Y reaction force on joint %d? (1 for yes, 0 for no): ", j);
    Sy2 (j,1) = input(support);
end


Sx = [Sx1, zeros(numj,1), zeros(numj,1)];
Sy = [zeros(numj,1), Sy1, Sy2];
```

Joint Location

```
X = zeros(numj,1);
Y = zeros(numj,1);

for j = 1:numj
    x = input(sprintf("Location (x-axis) of joint %d from Origin",j));
    X(j,1) = x;
    y = input(sprintf("Location (y-axis) of joint %d from Origin",j));
    Y(j,1) = y;
end
```

Load

```
L = zeros(2*numj,1);

for j = 1:numj
    loadh = input(sprintf("horizontal load at joint %d",j));
    loadv = input(sprintf("vertical load at joint %d",j));
    L(j,1) = loadh;
    L(numj+j,1) = loadv;

end
```

Member Lengths

```
R = zeros(numm, 1);
for i = 1:numm
    joints = zeros(1,2);
    for j = 1:numj
        if C(j,i) == 1
            if joints(1) == 0
                joints(1) = j;
            else
                joints(2) = j;
            end
        end
    end
    if joints(2) == 0
        error("Member %d only connected to one joint - check connection matrix C!", i);
    end
    X1 = X(joints(1));
    Y1 = Y(joints(1));
    X2 = X(joints(2));
    Y2 = Y(joints(2));
    R(i) = sqrt((X2 - X1)^2 + (Y2 - Y1)^2);
end
```

Matrix A

```
A = zeros(2 * numj, numm);
for m = 1:numm
    connected_joints = find(C(:, m) == 1);
    joint1 = connected_joints(1);
    joint2 = connected_joints(2);
    dx = X(joint2) - X(joint1);
    dy = Y(joint2) - Y(joint1);
    length = sqrt(dx^2 + dy^2);
    A(joint1, m) = dx / length;
    A(joint2, m) = -dx / length;
    A(joint1 + numj, m) = dy / length;
    A(joint2 + numj, m) = -dy / length;
end
A = [A, S];

disp("Matrix A:");
disp(A);
```

Solve for Forces

```
T = A \ L;
```

Buckling Calculations

```
max_buckling = zeros(numm, 1);
for i = 1:numm
    max_buckling(i) = 2390 * R(i)^(-1.811);
end

[maxload, ind] = max(L);
mass = 0;
Q = 1;
breakload = 0;

while Q == 1
    mass = mass + 1;
    L(ind) = mass;
    T_crit = A \ L;
    for mm = 1:numm
        if abs(T_crit(mm)) > max_buckling(mm)
            Q = 0;
            breakload = mm;
        else
            max_th_load = max(L);
        end
    end
end
```

Uncertainty

```
U_fit = 1.35;  % From lab data
buckling_strength = max_buckling(breakload);
buckling_length = R(breakload);
uncertainty_max_load = U_fit;
```

Printing Results

```
fprintf("\nEK301, Section A1, Group6: Jessica Q. Jinyu F. Riasat A., 3/22/2025\n");
fprintf("Load: %d oz\n", 25);
fprintf("Member forces in oz:\n")
for i = 1:numm
    if T(i) > 0
        tc = 'T';
    else
        tc = 'C';
    end
    fprintf("m%d: %.3f (%c)\n", i, T(i), tc);
end

fprintf("Reaction forces in oz:\n");
fprintf("Sx1: %.3f\n", T(numm + 1));
fprintf("Sy1: %.3f\n", T(numm + 2));
fprintf("Sy2: %.3f\n", T(numm + 3));
```

Buckling + Uncertainty

```
fprintf("The member to buckle first is member %d\n", breakload);
fprintf("Length of that member: %.2f in\n", buckling_length);
fprintf("Predicted buckling strength: %.2f oz ± %.2f oz\n", buckling_strength, U_fit);
fprintf("The maximum load that the physical truss could support is %d oz ± %.2f oz\n", max_th_load, uncertainty_max_load);

fprintf("Cost of the truss: $%.2f\n", Cost);
fprintf("Theoretical max load/cost ratio in oz/$: %.3f\n", abs(max_th_load / Cost));
```

A.2 — MATLAB Code Developed to Analyze Design 1 & 2 Truss

Connection matrix C for joint

```
jointq = "How many joints? ";
numj = input(jointq);
numm = (2*numj) - 3;

C = zeros(numj, numm);

% Loop through each join and each member
for j = 1:numj;
    for m = 1:numm
        prompt = sprintf("Is member %d connected to joint %d? (1 for yes, 0 for no): ", m, j);
        C(j,m) = input(prompt);
    end
end
```

Connection matrix for support force Sx Sy

```
Sx1 = zeros(numj, 1);
Sy1 = zeros(numj, 1);
Sy2 = zeros(numj, 1);

Sx = zeros(numj, 3);
Sy = zeros(numj,3);

connection_matrixS_joint = zeros(numj,3);

% Loop through each joint and each force


for j = 1:numj
    support = sprintf("Is the x reaction force on joint %d? (1 for yes, 0 for no): ", j);
    Sx1 (j,1) = input(support);
end

for j = 1:numj
    support = sprintf("Is the first Y reaction force on joint %d? (1 for yes, 0 for no): ", j);
    Sy1 (j,1) = input(support);
end

for j = 1:numj
    support = sprintf("Is the second Y reaction force on joint %d? (1 for yes, 0 for no): ", j);
    Sy2 (j,1) = input(support);
end


Sx = [Sx1, zeros(numj,1), zeros(numj,1)];
Sy = [zeros(numj,1), Sy1, Sy2];
```

Joint Location

```
X = zeros(numj,1);
Y = zeros(numj,1);

for j = 1:numj
    x = input(sprintf("Location (x-axis) of joint %d from Origin",j));
    X(j,1) = x;
    y = input(sprintf("Location (y-axis) of joint %d from Origin",j));
    Y(j,1) = y;
end
```

Load

```
L = zeros(2*numj,1);

for j = 1:numj
    loadh = input(sprintf("horizontal load at joint %d",j));
    loadv = input(sprintf("vertical load at joint %d",j));
    L(j,1) = loadh;
    L(numj+j,1) = loadv;

end
```

Member Lengths

```
R = zeros(numm, 1);
for i = 1:numm
    joints = zeros(1,2);
    for j = 1:numj
        if C(j,i) == 1
            if joints(1) == 0
                joints(1) = j;
            else
                joints(2) = j;
            end
        end
    end
    if joints(2) == 0
        error("Member %d only connected to one joint – check connection matrix C!", i);
    end
    X_joint1 = X(joints(1));
    Y_joint1 = Y(joints(1));
    X_joint2 = X(joints(2));
    Y_joint2 = Y(joints(2));
    R(i) = sqrt((X_joint2 - X_joint1)^2 + (Y_joint2 - Y_joint1)^2);
end
```

Initialize matrix A

```
A = zeros(2 * numj, numm);
for m = 1:numm
    connected_joints = find(C(:, m) == 1);
    joint1 = connected_joints(1);
    joint2 = connected_joints(2);
    dx = X(joint2) - X(joint1);
    dy = Y(joint2) - Y(joint1);
    length = sqrt(dx^2 + dy^2);
    R(m) = length;
    A(joint1, m) = dx / length;
    A(joint2, m) = -dx / length;
    A(joint1 + numj, m) = dy / length;
    A(joint2 + numj, m) = -dy / length;
end

A = [A,S];

% Display matrix A
disp('Matrix A:');
disp(A);
```

Solve for Forces

```
T = A \ L;
```

Buckling calculations

```
max_buckling = zeros(numm, 1);
for i = 1:numm
    max_buckling(i) = 2390 * R(i)^(-1.811);
end

% Theoretical max load calculation
[maxload, ind] = max(L);
Q = 1;
mass = 0;
breakload = 0;

while Q == 1
    mass = mass + 1;
    L(ind) = mass;
    T_crit = A \ L;
    for mm = 1:numm
        if abs(T_crit(mm)) > max_buckling(mm)
            Q = 0;
            breakload = mm;
        else
            max_th_load = max(L);
        end
    end
end
```

Uncertainty (based on class average)

```
U_fit = 1.35;  % oz
buckling_strength = max_buckling(breakload);
buckling_length = R(breakload);
uncertainty_max_load = U_fit;
```

Failure analysis

```
load = max(L);
for i = 1:numm
    rr(i) = load / T_crit(i);
    P_load(i) = (2390) * R(i)^(-1.811);
    W_fail(i) = -P_load(i) / rr(i);
end
Wfail = min(W_fail);
```

Cost and efficiency

```
totallength = sum(R);
C1 = numj * 10;
C2 = totallength * 1;
Cost = C1 + C2;
```

Print Results

```matlab
fprintf("\nEK301, Section A1, Group6: Jessica Q. Jinyu F. Riasat A., 3/22/2025 \n");
fprintf("------------------------------------------------------------\n");
fprintf("Load: %d oz\n", load);
fprintf("Member forces in oz:\n");
for i = 1:numm
    if T_crit(i) > 0
        tc = 'T';
    else
        tc = 'C';
    end
    fprintf("m%d: %.3f (%c)\n", i, T_crit(i), tc);
end

fprintf("\nReaction forces in oz:\n");
fprintf("Sx1: %.3f\n", T_crit(numm+1));
fprintf("Sy1: %.3f\n", T_crit(numm+2));
fprintf("Sy2: %.3f\n", T_crit(numm+3));

fprintf("\n------------------------------------------------------------\n");
fprintf("The member to buckle first is member %d\n", breakload);
fprintf("Length of that member: %.2f in\n", buckling_length);
fprintf("Predicted buckling strength: %.2f oz ± %.2f oz\n", buckling_strength, U_fit);
fprintf("The maximum load that the physical truss could support is %d oz ± %.2f oz\n", max_th_load, uncertainty_max_load);

fprintf("\n------------------------------------------------------------\n");
fprintf("Cost of the truss: $%.2f\n", Cost);
fprintf("Theoretical max load/cost ratio in oz/$: %.3f\n", abs(max_th_load / Cost));
fprintf("------------------------------------------------------------\n");
```

```matlab
% Member lengths in inches

fprintf('\nMember Lengths (in inches):\n');
for i = 1:numm
    fprintf("Member %2d: %.3f in\n", i, R(i));
end


% Buckling Strengths for Compression Members

U_fit = 1.35; % from acrylic fit data
fprintf('\nBuckling Strengths for Compression Members:\n');
fprintf('-------------------------------------------------------------\n');
fprintf('| Member | Length (in) | Force (oz) | Buckling Strength ± Unc. |\n');
fprintf('-------------------------------------------------------------\n');

for i = 1:numm
    force = T_crit(i);
    if force < 0  % Only for compression members
        length = R(i);
        buckling = 2390 * length^(-1.811);
        fprintf('|   m%-2d   |   %7.3f  |  %8.3f  |   %8.2f ± %.2f    |\n', ...
            i, length, force, buckling, U_fit);
    end
end
fprintf('-------------------------------------------------------------\n');
```