Galen Carter, Felix Munoz Rodriguez, Justin Yu

EK 301 Section A1 Prof. Bunch

Fall 2024

Preliminary Design Report

Introduction:

Truss analysis by hand is a tedious and time-consuming process in which algebraic errors can easily arise. Using a computer program, such as MATLAB, for truss analysis greatly improves the accuracy and speed of calculations. The program will determine which member of the truss will buckle first and, in conjunction with the experimental critical force from the buckling lab, allows us to determine the maximum load the truss can support. Furthermore, optimization of cost and maximum load, and comparison between candidate trusses – tasks which would be otherwise hard by hand – are easily completed with a computer program. Using our truss-analysis program, we will evaluate several candidate truss designs. Our goal is to choose the design which both minimizes cost and supports the heaviest live load.

Methods & Analysis:

<u>Code:</u>

Our program depends on a provided input file requiring the user to revise 5 input matrices based on consistent numbering of joints and members by the user:

1. C: Binary connection matrix (j x m) which describes the connection of the joints (rows) and members (columns) of the truss. A "1" in position (j, m) means that joint j is connected to member m; a "0" means there is no connection.
2. X: Location vector of length j describing the x-locations (in.) of each joint.
3. Y: Location vector of length j describing the y-locations (in.) of each joint.
4. Sx: Binary support force matrix (j x s) describing the location of the horizontal support force(s), where s is the number of support forces. A "1" in position (j, 1) means there is a horizontal support force at joint j.
5. Sy: Binary support force matrix (j x s) describing the locations of the vertical support forces. A "1" in position (j, s) means there is a vertical support force at joint j. Note that between Sy and Sx there should be a "1" in each column.
6. L: Load vector of length 2j that represents the joints of the truss. The row that corresponds to the joint where the load is placed has a value of +mg. The rest of the vector consists of zeros as the load is placed at one joint.

Next, matrix A was constructed by concatenating matrices representing the static equilibrium equations for the X- and Y-directions (Ax and Ay) as well as inputted Sx and Sy matrices.

To obtain Ax and Ay (both j x m), the columns of connection matrix C were iterated through to identify which rows (joints) each member connects to. Then, for each identified pair of joints, their respective coordinate pairs are obtained and the Cartesian distance between them is easily calculated. The distances are then used to scale the horizontal and vertical components of the force at each joint.

For example, for a member spanning from $(x_1, y_1) \rightarrow (x_2, x_2)$, the "1" in position (1, 1) in the Ax matrix is scaled by $(x_2 - x_1) / r_{1,2}$ and the vertical component at the same position in the Ay matrix is scaled by $(y_2 - y_1) / r_{1,2}$. This procedure is repeated for each member. Since the absolute values of the normalized differences are the same for both connected joints of a member – one being $(x_2 - x_1)$ and the other $(x_1 - x_2)$ – the code simply negates the sign for the secondary row where the value is updated. The same logic applies to the y-differences.

To form the final A matrix, the Ax and Ay are vertically combined then horizontally concatenated with the Sx and Sy matrices. This A matrix (2j x m + s) is related to the load vector (L) by:

$$L = AT$$

where T is a force vector (length m + s) consisting of all of the unknown forces. Because A is a square matrix (as s = 3 and, for a normal truss, M = 2J - 3), it is invertible, and T can be solved as follows:

$$T = A^{-1} L$$

Once each of the forces have been obtained via the T matrix, the maximum load and the member which will buckle first (critical member) can be found. Because we assume the truss members can withstand infinite tension, only the compression forces are considered. Because a truss is a linear system, for each compressive member, m, we can obtain R_m which relates the external load and the force in the member by:

$$T_m = W \times R_m \Rightarrow R_m = T_m / W$$

Next, the buckling force (a function of length) in oz. for each member is obtained by:

$$P_{crit}(L) = 4863.346L^{-2.208}$$

The uncertainty in the buckling force is ± 1.54 oz (95% confidence, 2 standard deviations). From P_{crit} and R_m, the critical load for each member can be found by dividing P_{crit} / R_m. The smallest critical load is the maximum load for the truss and the member which determines it is the critical member, the member which will buckle first. The uncertainty in the max load can be calculated as follows, where ΔP_{crit} is 1.54 oz:

$$\Delta P_{crit} / P_{crit} = \Delta W_{max} / W_{max} \Rightarrow (\Delta P_{crit} \times W_{max}) / P_{crit} = \Delta W_{max}$$

Finally, the total cost of the truss and the weight-to-cost ratio can be calculated. The total cost is calculated by:

$$Cost = \$10(j) + \$1(l)$$

where j is the number of joints and l is the total length of members. Then, the weight-to-cost ratio is simply the maximum load divided by the total cost.

Practice Problem:

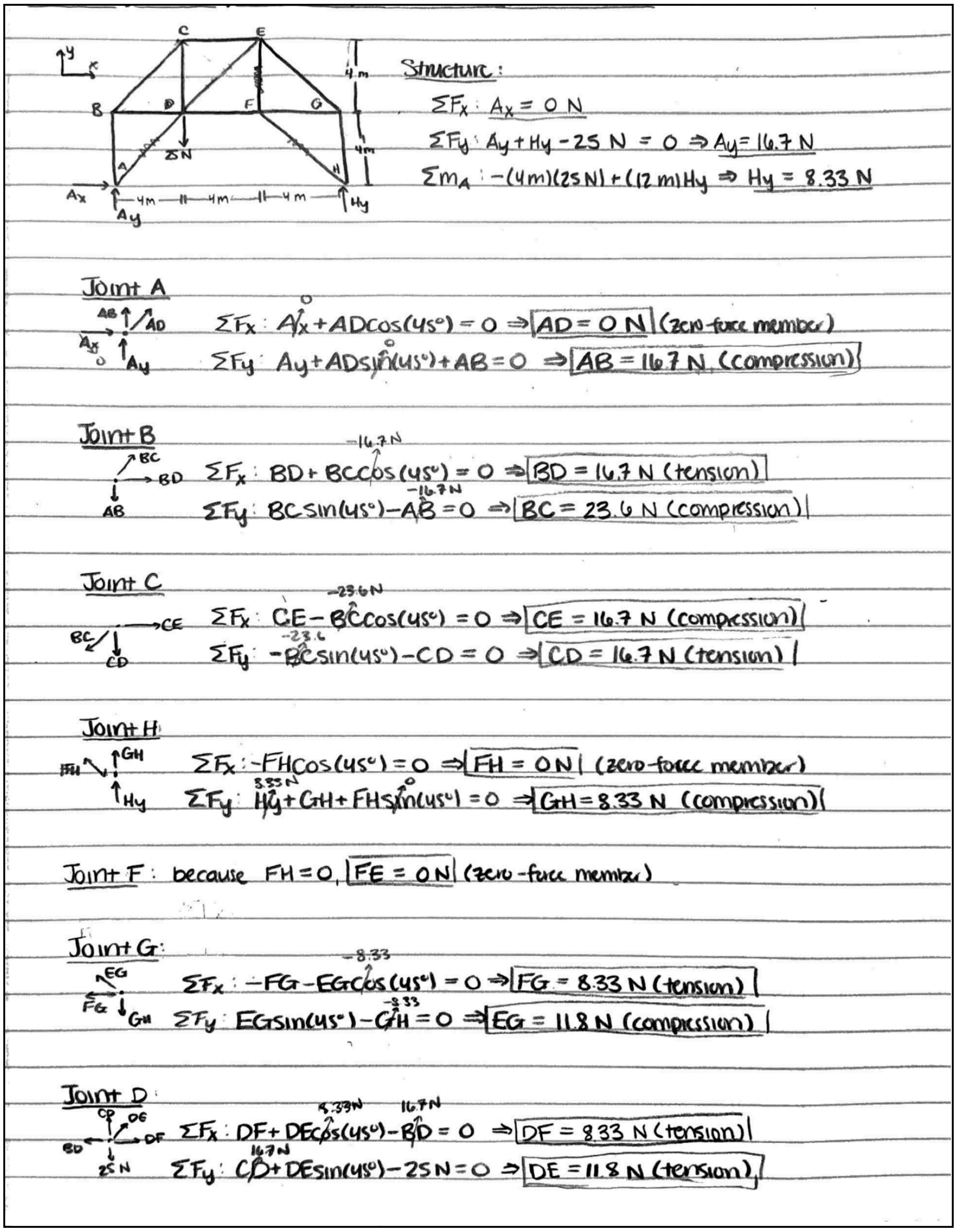

Figure 1. Manual solution of a practice truss using the method of joints. The structure is in the top left. Tensile forces are positive while compressive forces are negative (sign omitted).

Figure 1 displays the manual approach to a practice truss problem with the goal of confirming the accuracy of our program. First, for the entire structure, the sum of forces in the x- and y-directions and moment about A were set to zero. This determined the reaction forces at joints A and H. From there the individual forces in each member were solved by isolating the joints in the following order: A → B → C → H → F → G → D.

```
Load: 89.90 oz
Member Forces in oz
   m1: -59.93 oz (C)
   m2: 0 oz (N/A)
   m3: -84.76 oz (C)
   m4: 59.93 oz (T)
   m5: 59.93 oz (T)
   m6: -59.93 oz (C)
   m7: 42.38 oz (T)
   m8: 29.97 oz (T)
   m9: -42.38 oz (C)
   m10: 29.97 oz (T)
   m11: 0 oz (N/A)
   m12: -29.97 oz (C)
   m13: 0 oz (N/A)
Reaction Forces  in lbs
   Sx1: 0.00 oz
   Sy1: 59.93 oz
   Sy2: 29.97 oz
Cost of Truss: $2453.40
Theoretical Max Weight to Cost Ratio: 0.00001377
```

Figure 2. Computational solution of the same truss problem as in Figure 1. Here, m1 corresponds to AB; m2 to AD; m3 to BC; m4 to BD; m5 to CD; m6 to CE; m7 to DE; m8 to DF; m9 to EG; m10 to FG; m11 to FE; m12 to GH; m13 to FH; Sx1 to A_x; Sy1 to A_x; and Sy2 to H_y.

Figure 2 displays the computational solution of the exact same truss problem that Figure 1 solved manually. Although the units are in oz., when converted to N, the forces are identical. For example, m1 = -59.93 oz = 16.7 N, compression, which is exactly what the manual solution has. Table 1 summarizes these results and highlights the similarity between the computational and by-hand solutions.

Table 1. Forces for the practice truss problem solved manually and computationally. Forces are equivalent, showing the accuracy of the code.

Manual				**Computational**	
Member	Force (N)	Force (oz.)		Member	Force (oz.)
AB	-16.7	-59.9		m1	-59.9

AD	0	0		m2	0
BC	-23.6	-84.8		m3	-84.8
BD	16.7	59.9		m4	59.9
CD	16.7	59.9		m5	59.9
CE	-16.7	-59.9		m6	-59.9
DE	11.8	42.4		m7	42.4
DF	8.33	30.0		m8	30.0
EG	-11.8	-42.4		m9	-42.4
FG	8.33	30.0		m10	30.0
FE	0	0		m11	0
GH	-8.33	-30.0		m12	-30.0
FH	0	0		m13	0
A_x	0	0		Sx1	0
A_y	16.7	59.9		Sy1	59.9
H_y	8.33	30.0		Sy2	30.0

Results:
We choose to analyze two candidate trusses, both of which are based on non-right triangles. Both trusses assume a pin support at the left-most joint (forces S1x and S1y) and a roller support at the right-most joint (S2y).

<u>Candidate Truss 1:</u>
The design for the truss is displayed in Figure 3 with the members and joints labeled. Furthermore, the critical member is highlighted. Figure 4 displays the code output. Table 2 summarizes the forces in each of the members as well as provides information on member lengths, buckling forces, and force at max load. Only the buckling forces are noted for members under compression as the materials used can sustain (for our purposes) nearly infinite tensile force. Additionally, the critical member is highlighted according to the computational result.

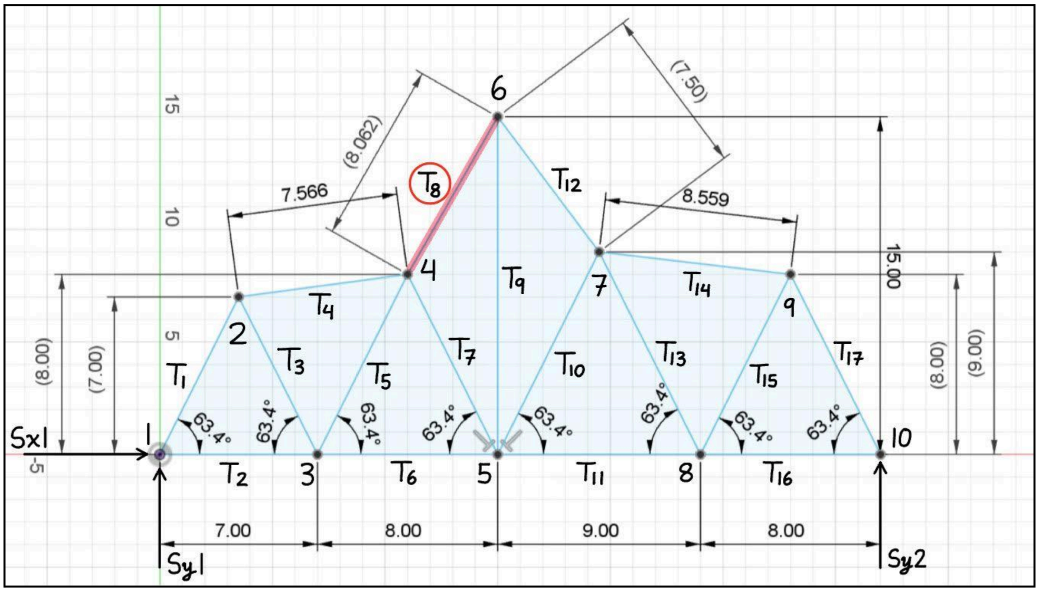

Figure 3. Diagram of candidate truss 1 with joints and members numbered and external forces labeled. The critical member (the member which will buckle first) is highlighted.

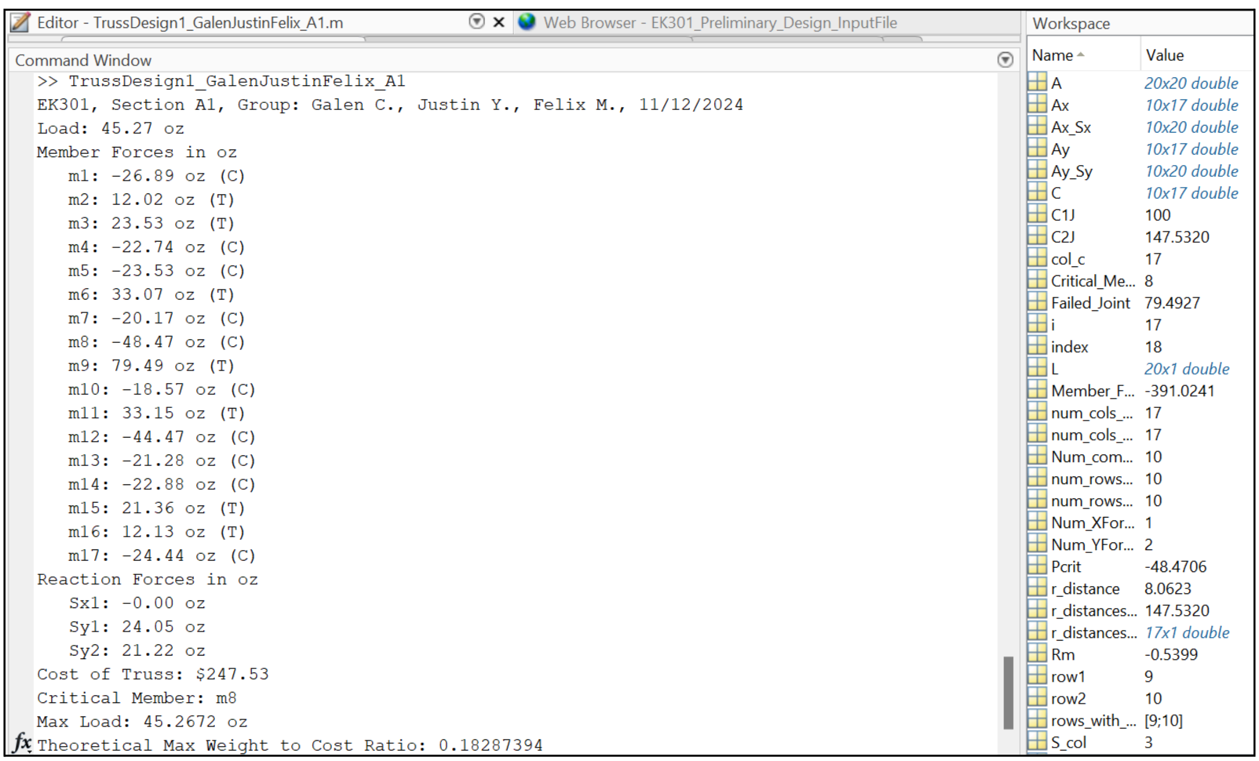

Figure 4. Computational output for candidate truss 1 with load, member forces (oz.), reaction forces (oz.), cost of truss, max load, and weight-to-cost ratio highlighted. A negative force indicates compression and a positive force indicates tension.

Table 2. Members, lengths, state of force (tension, compression, or zero-force member), buckling forces with uncertainties, and force under max load of members in candidate truss 1. A negative force indicates compression while a positive force indicates tension. Buckling forces for members in tension are omitted. The critical member is highlighted.

Member	Length (in)	Tension (T) or Compression (C)	Buckling Force with Uncertainty (oz)	Force at Max Load (oz)
T1	7.83	C	51.7 ± 1.54	-26.9
T2	7.83	T	n/a	12.0
T3	7.00	T	n/a	23.5
T4	7.57	C	55.7 ± 1.54	-22.7
T5	8.94	C	38.6 ± 1.54	-23.5
T6	8.00	T	n/a	33.1
T7	8.94	C	38.6 ± 1.54	-20.1
T8	8.06	C	48.5 ± 1.54	-48.5
T9	15.00	T	n/a	74.2
T10	10.1	C	29.5 ± 1.54	-12.1
T11	9.00	T	n/a	29.5
T12	7.50	C	56.9 ± 1.54	-40.1
T13	10.1	C	29.5 ± 1.54	-21.1
T14	8.56	C	38.6 ± 1.54	-20.2
T15	8.94	T	n/a	21.1
T16	8.00	T	n/a	10.6
T17	8.94	C	38.6 ± 1.54	-23.7

Table 3 below displays important information about buckling in candidate truss 1, with the critical member highlighted, buckling strength in the member, and the load the truss can support with uncertainties.

Table 3. The critical member, length, and buckling strength for candidate truss 1, as well as the maximum theoretical laid and load-to-cost ratio.

Crit. Member	Length (in)	Buckling Strength (oz.)	Max Theor. Load (oz.)	Load-to-Cost Ratio (oz. / $)
T8 (m8)	8.06	48.5 ± 1.54	45.3 ± 1.44	0.181

In total, the first candidate truss had a theoretical load (45.3 oz.) much higher than the required amount (32 oz.). The total cost ($250.24) was beneath the required $300, and the truss the required 32 inches (with a pin at 15 inches). Each joint was separated by ≥ 7 inches.

Candidate Truss 2:

Candidate truss 2 was a modified version of the first one. The design for the truss is displayed in Figure 5 with the members and joints labeled. Furthermore, the critical member is highlighted. Figure 6 displays the code output. Table 4 summarizes the forces in each of the members as well as provides information on member lengths, buckling forces, and force at max load. Note again, that only the buckling forces are used to determine the max load. Additionally, the critical member is highlighted according to the computational result.

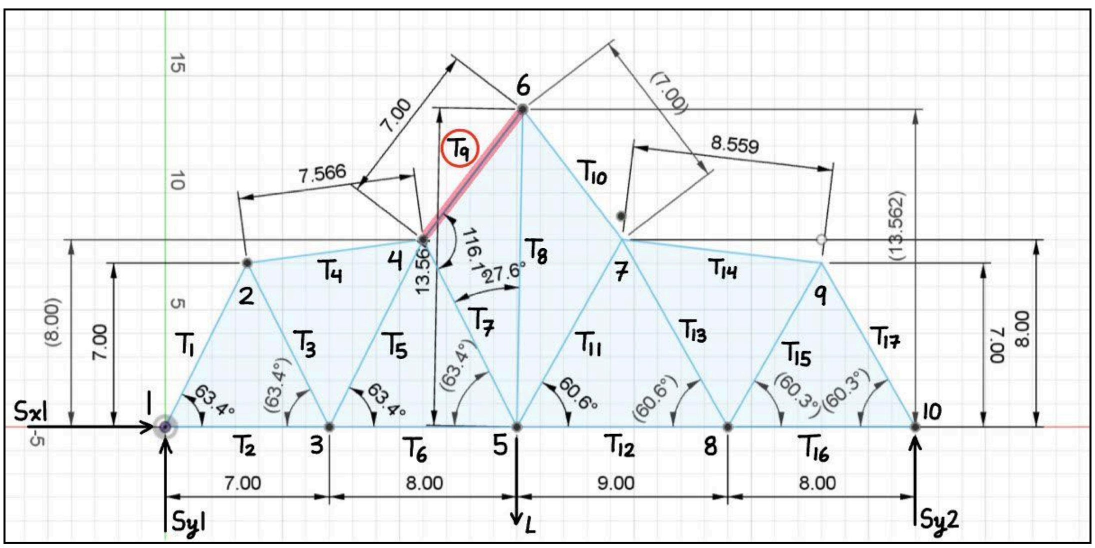

Figure 5. Diagram of candidate truss 2 with joints and members numbered and external forces labeled. The critical member (the member which will buckle first) is highlighted.

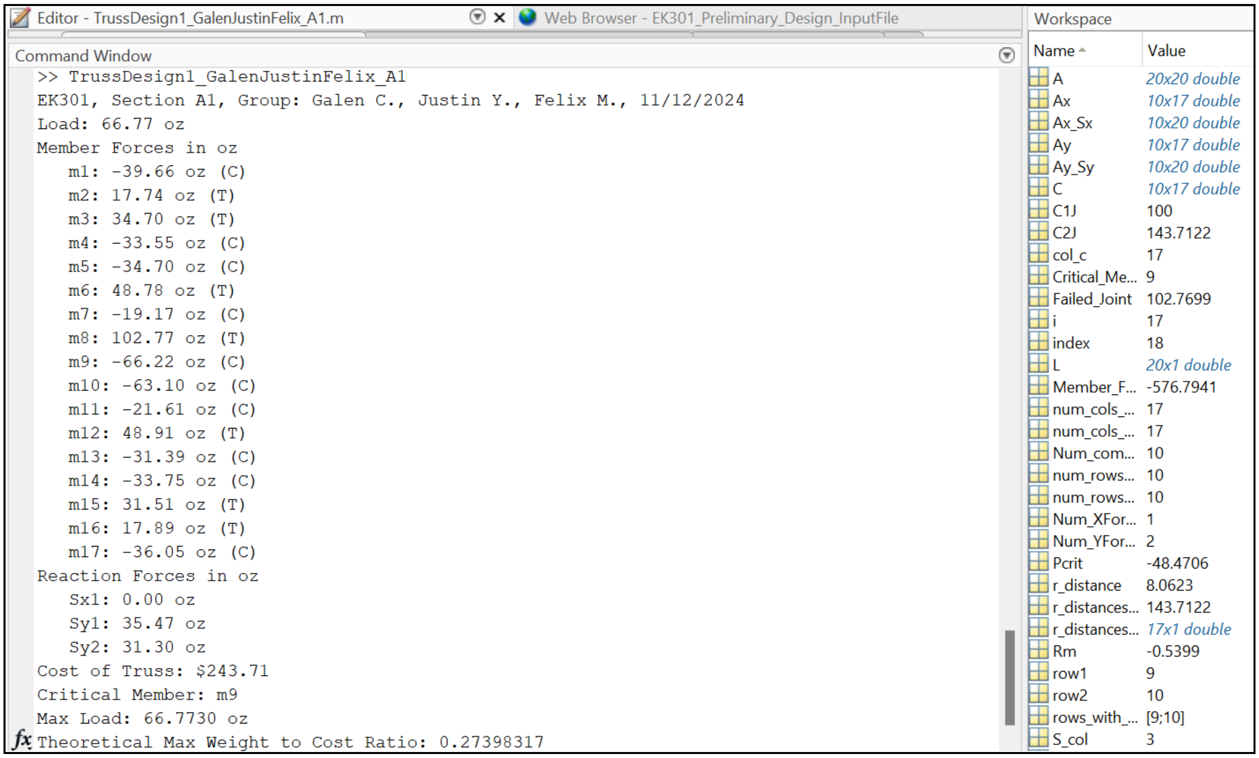

Figure 6. Computational output for candidate truss 2 with load, member forces (oz.), reaction forces (oz.), cost of truss, max load, and weight-to-cost ratio highlighted.

Table 4. Members, lengths, state of force (tension, compression, or zero-force member), buckling forces with uncertainties, and force under max load of members in candidate truss 2. The critical member is highlighted.

Member	Length (in)	Tension (T) or Compression (C)	Buckling Force with Uncertainty (oz)	Force at Max Load (oz)
T1	7.83	C	52.7 ± 1.54	-39.7
T2	7.00	T	n/a	17.7
T3	7.83	T	n/a	34.7
T4	7.57	C	55.7 ± 1.54	-33.6
T5	8.94	C	38.6 ± 1.54	-34.7
T6	8.00	T	n/a	48.8
T7	8.94	C	38.6 ± 1.54	-19.2
T8	13.56	T	n/a	103
T9	7.00	C	66.2 ± 1.54	-66.2

T10	7.00	C	66.2 ± 1.54	-63.1
T11	9.18	C	36.4 ± 1.54	-21.6
T12	9.00	T	n/a	48.9
T13	9.18	C	36.4 ± 1.54	-31.4
T14	8.56	C	46.4 ± 1.54	-33.8
T15	8.06	T	n/a	31.5
T16	8.00	T	n/a	17.9
T17	8.06	C	48.5 ± 1.54	-36.1

Table 5 below displays important information about buckling in candidate truss 2, with the critical member highlighted, buckling strength in the member, and the load the truss can support with uncertainties.

Table 5. The critical member, length, and buckling strength for candidate truss 2, as well as the maximum theoretical laid and load-to-cost ratio.

Crit. Member	Length (in)	Buckling Strength (oz.)	Max Theor. Load (oz.)	Load-to-Cost Ratio (oz. / $)
T9 (m9)	7.00	31.73 ± 1.54	66.7 ± 1.55	0.274

Discussion & Conclusion:

We began our truss designing process considering basic, well-known trusses such as the Pratt truss. However, when we adopted them to fit the design requirements for the project, we found that the loads they could hold were just below the required 32 oz. We next considered a truss based on equilateral triangles, since they are the most stable type of triangle. Some of these trusses were able to fulfill the criteria but had room for improvement. If we had more time, we would certainly continue investigating equilateral-triangle-based trusses. We then arrived at our two candidate truss designs, both of which are based on isosceles triangles. The two designs are fairly similar; the largest difference is that candidate truss 2 tries to minimize member lengths (i.e. in T9 and T10, see Figure 5 and Table 4) whereas candidate truss 1 did not. Considering this might explain our different results for two trusses; although both designs fit all of the requirements, the second design, based on the computational analysis, is predicted to perform much better than the first.

The goal of the design process was to create a truss which not only fulfilled several requirements but also maximized the load it could support without buckling and minimized the cost while doing it. Both of our trusses were below the required $300: our first truss had a total cost of $250.24 (Figure 4) while our second design had a total cost of $243.71 (Figure 6). The slight difference in cost simply comes from shortening the members connecting the central spike. The trusses also had max theoretical load greater than the minimum required 32 oz. The first truss had a max theoretical load of 45.3 oz (Table 3) which is solidly past the required amount, but our second truss had a max theoretical load of 66.7 oz (Table 5) which more than doubles the requirement. The difference in max load is likely due to the shortening of members T9 and T10 and the slight angle of T8 on Truss 2 (Figure 5). Since our second truss had both a lower cost and had a greater max load, it had a larger load-to-cost ratio of 0.274 oz/$ (Table 5) when compared to truss one's ration of 0.181 oz/$ (Table 3). Based on these metrics we decided that truss 2 was a better design choice.

For testing in the future, we plan on adjusting small parameters (e.g. width in the surrounding members, height) in the central spike to see if we can obtain an even larger max load. Furthermore, we want to investigate more designs with equilateral triangles because they should theoretically be stronger than the isosceles triangles that we used.

Appendix:

Input File Code

```
%---------------- Direct Input 'C' ----------------
% Connection Matrix | Members (Columns), Joints (Rows)

C = [1 1 0 0 0 0 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 0 0;
     0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0;
     0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0;
     0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0 0;
     0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0;
     0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0;
     0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0;
     0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1;
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1];

% C = [1 1 0 0 0 0 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 0 0;
%      0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0;
%      0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0;
%      0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0;
%      0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0;
%      0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0;
%      0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0;
%      0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1;
%      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1];


%----------------Direct Input 'Sx & Sy' --------------
% Support Matrix of X-direction | Members (Columns), Joints (Rows)
Num_XForces = 1;

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;
      0 0 0;
      0 0 0];

% Support Matrix of Y-direction | Members (Columns), Joints (Rows)
Num_YForces = 2;

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

```
      0 0 0;
      0 0 1];

%----------------- Direct Input 'X & Y' ----------------
% X-Coordinate (Chronologically Correspondant by Joint #)
X = [0, 3.5, 7, 11, 15, 15, 19.5, 24, 28, 32]; % Joint 1 --> Joint n
% X = [0, 3.5, 7, 11, 15, 15.25, 19.5, 24, 28, 32]; % Truss2

% Y-Coordinate (Chronologically Correspondant by Joint #)
Y = [0, 7, 0, 8, 0, 15, 8, 0, 7, 0]; % Joint 1 --> Joint n
% Y = [0, 7, 0, 8, 0, 13.562, 8, 0, 7, 0]; % Truss2

%------------------- Direct Input 'L' -------------------
% Load Vector | Order by Horizontal --> Vertical Joints

L = [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; % X
     0; 0; 0; 0; 45.2672; 0; 0; 0; 0; 0]; % Y

save('TrussDesign1_GalenJustinFelix_A1.mat', 'C', 'Sx', 'Sy', 'X', 'Y', 'L', 'Num_XForces'
run('EK301_Preliminary_Design_OutputFile.m')
```

```
EK301, Section A1, Group: Galen C., Justin Y., Felix M., 11/12/2024
Load: 45.27 oz
Member Forces in oz
   m1: -26.89 oz (C)
   m2: 12.02 oz (T)
   m3: 23.53 oz (T)
   m4: -22.74 oz (C)
   m5: -23.53 oz (C)
   m6: 33.07 oz (T)
   m7: -20.17 oz (C)
   m8: -48.47 oz (C)
   m9: 79.49 oz (T)
   m10: -18.57 oz (C)
   m11: 33.15 oz (T)
   m12: -44.47 oz (C)
   m13: -21.28 oz (C)
   m14: -22.88 oz (C)
   m15: 21.36 oz (T)
   m16: 12.13 oz (T)
   m17: -24.44 oz (C)
Reaction Forces in oz
   Sx1: -0.00 oz
   Sy1: 24.05 oz
   Sy2: 21.22 oz
Cost of Truss: $247.53
Critical Member: m8
Max Load: 45.2672 oz
Theoretical Max Weight to Cost Ratio: 0.18287394
```

Published with MATLAB® R2023a

Output File Code

```
% Clear variables and load data
clear all
load("C:\Users\madak\OneDrive\Desktop
\EK301\TrussDesign1_GalenJustinFelix_A1.mat");

% Calculate the dimensions for matrix Ax & Ay
num_rows_Ax = size(C, 1);
num_cols_Ax = size(C, 2);
num_rows_Ay = size(C, 1);
num_cols_Ay = size(C, 2);

% Create matrix Ax & Ay with dimensions of C
Ax = zeros(num_rows_Ax, num_cols_Ax);
Ay = zeros(num_rows_Ay, num_cols_Ay);

% Iterate through each column of C (each member) --> Ax
r_distances_Sum = 0;
r_distances_vector = eye(size(C, 1), 1);
index = 1;
for col_c = 1:size(C, 2)
    % Find rows (each joint) where there is a 1 in each column of C
    rows_with_ones_in_c = find(C(:, col_c) == 1);

    % Iterate through each column in C to find the 2 corresponding rows with 1
 respectively
    for i = 1:length(rows_with_ones_in_c)-1
        row1 = rows_with_ones_in_c(i);
        row2 = rows_with_ones_in_c(i+1);

        % Get (X,Y) of the two rows from Inputs X & Y (Coordinates)
        x1 = X(row1);
        x2 = X(row2);
        y1 = Y(row1);
        y2 = Y(row2);

        % Calculate the X, Y coordinates of joint differences respectively
        x_difference = x2 - x1;
        y_difference = y2 - y1;
        r_distance = sqrt(x_difference^2 + y_difference^2);

        % Append each Member Length for Cost Computation
        r_distances_Sum = r_distances_Sum + r_distance;
        r_distances_vector(index, 1) = r_distance;
        index = index + 1;

        % Update Ax matrix with the differences
        Ax(row1, col_c) = x_difference/ r_distance;
        Ax(row2, col_c) = -x_difference/ r_distance;
    end
end
```

```
% Iterate through each column of C (each member) --> Ay
for col_c = 1:size(C, 2)
    % Find rows (each joint) where there is a 1 in each column of C
    rows_with_ones_in_c = find(C(:, col_c) == 1);

    % Find rows (each joint) where there is a 1 in each column of C
    for i = 1:length(rows_with_ones_in_c)-1
        row1 = rows_with_ones_in_c(i);
        row2 = rows_with_ones_in_c(i+1);

        % Get (X,Y) of the two rows from Inputs X & Y (Coordinates)
        x1 = X(row1);
        x2 = X(row2);
        y1 = Y(row1);
        y2 = Y(row2);

        % Calculate the X, Y coordinates of joint differences respectively
        x_difference = x2 - x1;
        y_difference = y2 - y1;
        r_distance = sqrt((x_difference)^2 + (y_difference)^2);

        % Update Ay matrix with the differences
        Ay(row1, col_c) = y_difference/ r_distance;  % Place difference in
 first row
        Ay(row2, col_c) = -y_difference/ r_distance; % Negate the difference
 for the second row
    end

end

% Create the A Matrix
Ax_Sx = [Ax, Sx];
Ay_Sy = [Ay, Sy];
A = [Ax_Sx; Ay_Sy];

% Computing the T matrix
T = inv(A)*L;

%Results
for i = 1:length(L)
    if L(i) > 0
        W_index = i;
        W_oz = L(i);
        break;
    end
end

disp('EK301, Section A1, Group: Galen C., Justin Y., Felix M., 11/12/2024');
disp(sprintf('Load: %.2f oz', W_oz));
disp(sprintf('Member Forces in oz'));

[~, S_col] = size(Sx);
Num_compression_members = 0;
```

```
for i = 1:(length(T) - S_col)
    Member_Forces_oz = T(i, 1) * 16;
    T_or_C = T(i, 1);
    if T(i) > 0
        disp(sprintf('    m%d: %.2f oz (T)', i,T(i)));
    elseif T(i) < 0;
        disp(sprintf('    m%d: %.2f oz (C)', i, T(i)));
        Num_compression_members = Num_compression_members + 1;
    else
        disp(sprintf('    m%d: 0 oz (N/A)', i));
        T(i) = 0;
    end
end


disp('Reaction Forces in oz');
for i = 1: Num_XForces
    XReaction_Forces = T((length(T) - S_col) + i);
    disp(sprintf('    Sx%d: %.2f oz', i, XReaction_Forces));
end
for i = 1: Num_YForces
    YReaction_Forces = T((length(T) - S_col + Num_XForces) + i);
    disp(sprintf('    Sy%d: %.2f oz', i, YReaction_Forces));
end

C1J = 10 * size(C,1);
C2J = r_distances_Sum;
Total_Cost = C1J + C2J;
disp(sprintf('Cost of Truss: $%.2f', Total_Cost));


W_failure_array = eye(1, Num_compression_members);
for i = 1:(size(T, 1) - S_col)
    if T(i, 1) < 0
        Rm = T(i) / W_oz;
        Pcrit = -4863.346*((r_distances_vector(i, 1))^-2.208); %oz
        W_failure = (Pcrit / Rm);
        W_failure_array(1, i) = W_failure;
    else
        continue
    end
end

[~, Critical_Member] = min(T);
disp(sprintf('Critical Member: m%d', Critical_Member));

W_failure_value = min(W_failure_array(W_failure_array > 0));
disp(sprintf('Max Load: %.4f oz', W_failure_value));
Failed_Joint = max(abs(T));

Weight_Cost_Ratio_Value = W_failure_value/Total_Cost;
disp(sprintf('Theoretical Max Weight to Cost Ratio: %.8f',
 Weight_Cost_Ratio_Value));
```