EK301 Preliminary Design Report

Kailan Pan, James Conlon, Austin Zhang

Section A2 - Professor Joseph Bunch

Spring 2024, Due April 5th, 2024

Introduction

Determining each truss member's load given any truss design is time-consuming if done by hand, a way to make this process faster is with computer programs designed to calculate the load experienced by every truss member. The next step in a complete analysis of a truss involves finding both the critical member and the theoretical maximum force. Implementing those two steps in a computer program is practically a requirement given the time-consuming nature of these calculations by hand. Using a MATLAB program and the information on each truss member's lengths, joint locations, and the load at the specific joint, MATLAB will be able to calculate the load on each member of the truss, the cost of the truss, the member in the truss that will buckle first at a theoretical maximum force, as well as the theoretical maximum load-to-cost ratio. The overall goal is to test different designs that will have high maximum buckling force as well as a high maximum load-to-cost ratio for a truss, while also meeting the specific truss design requirements.

Methods and Analysis

To be able to use the computer program and analyze the load of each member of the truss, the first thing that must be found is $\boldsymbol{A}$ which represents the coefficients (i.e. direction cosines) of each member. To calculate the matrix $\boldsymbol{A}$, the required inputs are matrices $\boldsymbol{C}$, $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{S_x}$, and $\boldsymbol{S_y}$. $\boldsymbol{C}$ is a connection matrix that will be determined from the number of joints and members in the truss, where rows in the matrix correspond to the total number of joints in the truss, and the columns in the matrix correspond to the total number of members in the truss. Thus, the size of $\boldsymbol{C}$ is $j \times m$. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are matrices that will be the coordinates of each joint location, corresponding to some origin and coordinate system which is ultimately arbitrary (as long as it is consistent). $\mathbf{S_x}$ and $\mathbf{S_y}$

are the matrices that define the support force locations of the truss. To visualize the calculation of matrix ***A***, Equation 1 shows the calculations that will be performed to achieve ***A***.

Equation 1[1]:

$$\begin{array}{rl}
& \begin{array}{ccccccc|ccc} m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 & & \mathbf{S} & \end{array} \\
\mathbf{A} = & \begin{bmatrix}
\frac{x_2-x_1}{r_{1,2}} & \frac{x_3-x_1}{r_{1,3}} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\frac{x_1-x_2}{r_{1,2}} & 0 & \frac{x_3-x_2}{r_{2,3}} & 0 & \frac{x_4-x_2}{r_{2,4}} & \frac{x_5-x_2}{r_{2,5}} & 0 & 0 & 0 & 0 \\
0 & \frac{x_1-x_3}{r_{1,3}} & \frac{x_2-x_3}{r_{2,3}} & \frac{x_4-x_3}{r_{3,4}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{x_3-x_4}{r_{3,4}} & \frac{x_2-x_4}{r_{2,4}} & 0 & \frac{x_5-x_4}{r_{5,4}} & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & \\
- & - & - & - & - & - & - & - & - & - \\
\frac{y_2-y_1}{r_{1,2}} & \frac{y_3-y_1}{r_{1,3}} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\frac{y_1-y_2}{r_{1,2}} & 0 & \frac{y_3-y_2}{r_{2,3}} & 0 & \frac{y_4-y_2}{r_{2,4}} & \frac{y_5-y_2}{r_{2,5}} & 0 & 0 & 0 & 0 \\
0 & \frac{y_1-y_3}{r_{1,3}} & \frac{y_2-y_3}{r_{2,3}} & \frac{y_4-y_3}{r_{3,4}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{y_3-y_4}{r_{3,4}} & \frac{y_2-y_4}{r_{2,4}} & 0 & \frac{y_5-y_4}{r_{5,4}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{y_2-y_5}{r_{2,5}} & \frac{y_4-y_5}{r_{5,4}} & 0 & 0 & 1
\end{bmatrix} & \begin{array}{l} \text{x components} \\ - \\ \text{y components} \end{array}
\end{array}$$

For Equation 1, the strategy is to use the locations of the joints to determine the lengths of each member in the truss as well as the coefficients of the forces in both the x and y directions. This is done in MATLAB by taking the indices of the two 1s in each column (corresponding to joints), and mapping those indices to ***X*** and ***Y*** (coordinate vectors) to calculate the joint-to-joint distance which is needed for the final ***A*** as seen in Equation 1. Also note that the last 3 columns of ***A*** are the support reactions (they act in one direction so they are not defined by a direction cosine, just with 1). This calculation is shown in Appendix Fig 3 as the MATLAB code.

Now the matrices ***A*** and ***L*** (known external forces) are solved for the unknowns. This is where MATLAB is especially helpful as it can be done with the following equation $T = A^{-1}(L)$ to get the ***T*** matrix which is the force matrix that represents the tensions/compressions on each member. The equation that will be needed is:

Equation 2[1]: $[A][T] = [L] \rightarrow T = A^{-1}(L)$

In Equation 2, ***T*** is the force matrix that has one column for all the internal forces experienced by each member and all reaction forces of the truss, ***A*** is the matrix populated from the coefficient of forces from ***T***, and ***L*** is the load vector that represents the known external forces acting on the truss. Using linear algebra, ***T*** can be solved which gives the forces experienced by each member of the truss as well as the supports of the truss.

The next equation needed to calculate the buckling force is

Equation 3[2]: $F = 3054.789 \bullet L^{-2.009}$

Where ***L*** is the length(in) of a member in a truss, and ***F*** is the force(oz) that the member will buckle. Figure 1 from the Appendix shows the power best-fit data from all EK301 groups in the Spring 2024 semester and gives the relationship of the buckling force vs the lengths of an acrylic strip. The uncertainty in determining the buckling force from Figure 1 using the fit error bar is ±1.36 oz, this is the uncertainty that will be used when making a truss design for the project. The specification for the minimum amount of weight that the truss has to handle is a live load of 32 oz, which is the force that will be applied when testing the truss design in the MATLAB program. This live load is ultimately arbitrary for calculating the max buckling load and critical member since R_m scales linearly with the applied load. However, the choice of 32 oz is relevant for the calculation of tension and compression forces in each member for that given load but is not important for the actual buckling member since that is based on the ratio of internal force to applied load (R_m).

After the truss has been designed, to determine the overall cost of the truss, the equation that will be used is

Equation 4[1]: $Cost = C_1J + C_2L$

Where $\boldsymbol{C_1}$ is the cost per joint of the truss at \$10/joint, $\boldsymbol{J}$ is the number of joints in the truss, $\boldsymbol{C_2}$ is the cost per inch of the truss at \$1/inch, and $\boldsymbol{L}$ is the total lengths of all members combined in the truss. The overall goal is to design a truss that has a cost within the range of \$300. Another equation to test if the truss is within the specification is

Equation 5[1]: $M = 2J - 3$

Where $\boldsymbol{J}$ is the number of joints in the truss, and $\boldsymbol{M}$ is the total number of members in the truss. Equation 5 holds as the specification truss design to make sure the relationship between the total number of members and the total number of joints is accurate before beginning the data analysis. This is algorithmically done with the MATLAB code in Appendix Figure 4.

In Figures 3-6, the MATLAB code shown will calculate the load in each member of the truss as well as specify if the member is in tension, compression, or is a zero-force member. It will also calculate the maximum load that a specific member in the truss will hold before buckling, the theoretical maximum load-to-cost ratio of the truss, and ensure the specifications are met from Equations 4 and 5.

To test whether the MATLAB code is accurate, a practice example truss[3] shown in Figure 2 from the Appendix will be used where the code will calculate all the forces that each member experiences as well as a handwritten calculation to compare both results.

Data Table 1 for the Practice Example Truss

Member Number	Lengths (m)	Tension (T) or Compression (C) or Zero-Force Member (ZFM)	$F_{applied}$ (N) = 25N	$F_{maximum}$ (N) = 99.69N
1	4	C	16.67	66.46
2	5.66	ZFM	0	0
3	4	T	16.67	66.46
4	5.66	C	23.57	93.99
5	4	T	16.67	66.46
6	4	C	16.67	66.46
7	5.66	T	11.78	46.99
8	4	T	8.33	33.23
9	4	ZFM	0	0
10	5.66	C	11.78	46.99
11	4	T	8.33	33.23
12	5.66	ZFM	0	0
13	4	C	8.33	33.23

Table for the practice example truss that includes forces in each member, whether each member is in tension, compression, or is a zero-force member, as well as the lengths of the members, and maximum force for a member to buckle.

Using the data table above for the information collected for the practice example truss, each member of the truss will be either in tension, compression, or a zero-force member. In Figure 7 from the Appendix, calculations of the forces in each member given a load of 25 N are done through the MATLAB code. In Figure 8 from the Appendix, calculations of the forces in each member are done by hand. Comparing the results from both MATLAB code calculations and handwritten calculations, both results match as well as determining whether each member is in tension, compression, or is a zero-force member, therefore, the verification of both results means that the MATLAB code used is accurate. In the table above, member 4 with a length of 5.66 meters will be the first member to buckle when a theoretical maximum load of 99.69 N is applied with an uncertainty of ±1.36 N. The predicted buckling strength for member 4 is 93.99 N in compression once that load is applied. In the truss design for the project, only members that are in compression for a truss can buckle. The assumption is that using acrylic strips as the material, members that are in tension for the truss and the joints of the truss will have infinite strength which will not fail under an applied load. The overall cost of the practice example truss is \$140.28 and the maximum load-to-cost ratio is 0.7106 $\frac{N}{\$}$. Since the MATLAB code is verified to be accurate, the code will be used to calculate different truss designs as well as the theoretical maximum force that the truss can handle for the preliminary design phase.

Results

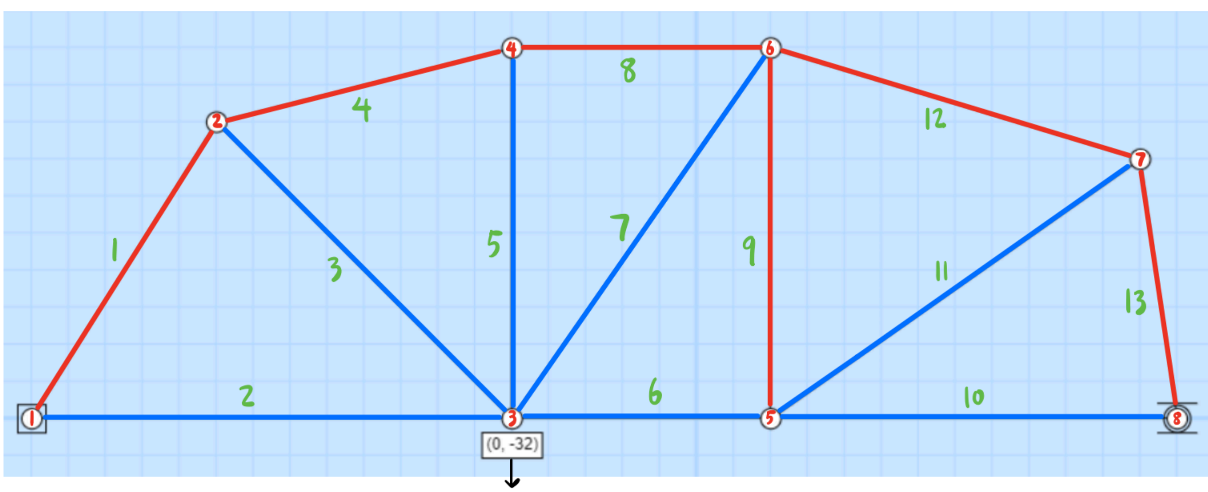


Figure: Truss Design 1

First truss design, members and joints are labeled with load applied at joint 3, and members that are in tension are blue and members that are in compression are red.

Data Table 2 for Truss Design 1

Member Number	Lengths (in)	Tension (T) or Compression (C)	$F_{applied}$(oz) = 32 oz	$F_{maximum}$(oz) = 49.13 oz
1	9.434	C	21.911	33.641
2	13	T	11.613	17.829
3	11.314	T	17.737	27.232
4	8.246	C	24.898	38.227
5	10	T	6.039	9.271
6	7	T	14.761	22.663
7	12.207	T	16.380	25.149
8	7	C	24.155	37.085
9	10	C	8.911	13.804
10	11	T	1.917	2.943
11	12.207	T	15.678	24.071
12	10.44	C	15.411	23.661
13	7.071	C	13.556	20.812

Table for the first truss design that includes forces in each member, whether each member is in tension or compression, as well as the lengths of the members, and maximum force for a member to buckle.

Using the calculations from Figure 9 in the Appendix and the figure of truss design 1, a load of 32 oz is put on the truss at joint 3 and the maximum theoretical load before the truss buckles is 49.13 oz with an uncertainty of ±1.36 oz. The critical member in the truss that will buckle is member 1 highlighted in blue from Table 2 with a buckling strength of 33.641 oz. The overall cost of the truss design is \$208.92 and has a maximum load-to-cost ratio of \$0.2351 $\frac{oz}{\$}$.

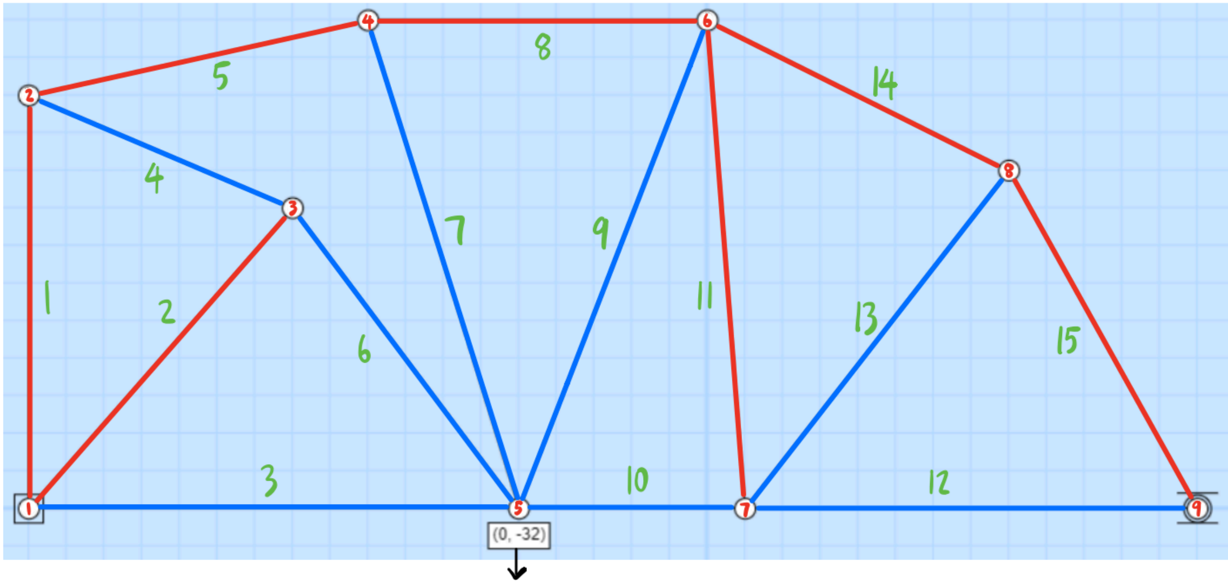


Figure: Truss Design 2

Second truss design, members and joints are labeled with load applied at joint 5, and members that are in tension are blue and members that are in compression are red.

Data Table 3 for Truss Design 2

Member Number	Lengths (in)	Tension (T) or Compression (C)	$F_{applied}$(oz) = 32 oz	$F_{maximum}$(oz) = 63.27 oz
1	11	C	11.318	22.378
2	10.63	C	9.650	19.080
3	13	T	6.355	12.564
4	7.616	T	18.921	37.411
5	9.22	C	17.816	35.225
6	10	T	18.395	36.370
7	13.601	T	4.044	7.995
8	9	C	18.395	36.737
9	13.928	T	14.378	28.427
10	7	T	13.419	26.533
11	13.153	C	8.097	16.008
12	11	T	5.964	11.792
13	10.817	T	10.138	20.045
14	8.944	C	13.338	26.371
15	10.296	C	14.685	29.035

Table for the second truss design that includes forces in each member, whether each member is in tension or compression, as well as the lengths of the members, and maximum force for a member to buckle.

Using the calculations from Figure 10 in the Appendix and the figure of truss design 2, a load of 32 oz is put on the truss at joint 5 and the maximum theoretical load before the truss buckles is 63.27 oz with an uncertainty of ±1.36 oz. The critical member in the truss that will buckle is member 5 highlighted in blue from Table 3 with a buckling strength of 35.225 oz. The overall cost of the truss design is $250.25 and has a maximum load-to-cost ratio of $0.2528 $\frac{oz}{\$}$.

Discussion & Conclusion

Both of the truss designs were successful in being able to withstand a minimum live load of 32 oz as required for the truss along with meeting the required specifications, such as having the cost of the truss within less than \$300 using Equation 4, as well as having the members within range using Equation 5.

The first truss can withstand a theoretical maximum load of 49.13 oz with an uncertainty of ±1.36 oz whereas the second truss can withstand 63.27 oz with an uncertainty of ±1.36 oz. The cost of the second truss is also more expensive, coming in at \$250.25, whereas the first truss is \$208.92. The maximum cost-to-load ratio for the first truss is \$0.2351 $\frac{oz}{\$}$ and for the second truss, it's \$0.2528 $\frac{oz}{\$}$. In the first truss, member 1 from the left will buckle first, while in truss two, member 5 from the top of the truss will be the first to buckle. Initially, the first design is based on the Baltimore truss design where the goal is to split the diagonals of the truss into triangles. For the second truss, the design of the truss making member one vertical eliminates any horizontal force. The design was focused on making the top part of the truss taller so that the members would experience less compression force before buckling. Applying the triangle from the idea of the Baltimore truss, the maximum load of truss two was higher than truss one. If the design in truss one was vertical in member 1, it would decrease the compression force, however, it would increase the compression force experienced by member 4. The biggest challenge with the project was the fact that the pin-to-load length has to be from 12.5 - 13.5 inches, which means that the truss is not able to have any node from the pin to the load. It is also important to think about how easy it is to build the physical truss. The process of building the physical truss must be accurate like the design or it will not hold a load nearly as high. Small changes in the truss' design will lead to a completely different maximum force at which a specific member will buckle. Even though the theoretical maximum load of the first design is lower than the second design, the process of building the actual truss will be simpler so it has a higher chance to withstand the calculated theoretical maximum load. Since building the physical truss is important to the final project, the factor of how easy building the truss is plays an important role.

Appendix

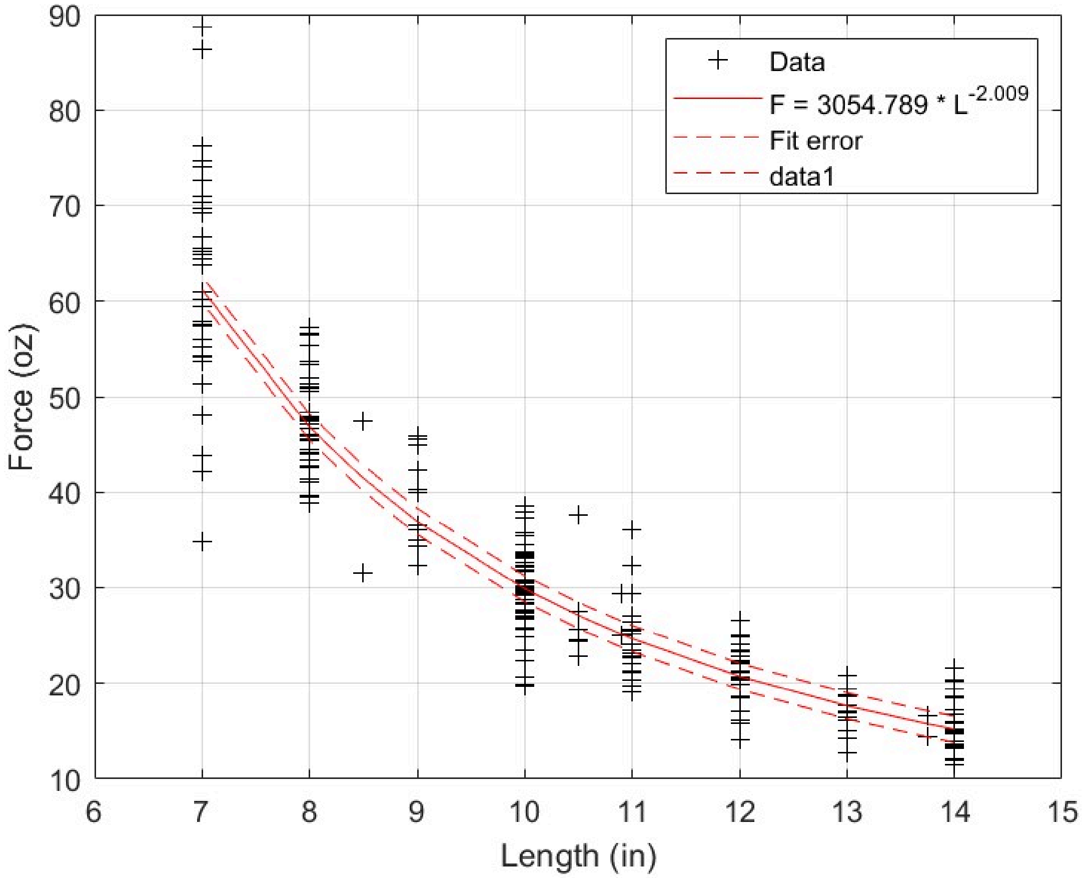


Figure 1: The buckling force vs the lengths of an acrylic strip.

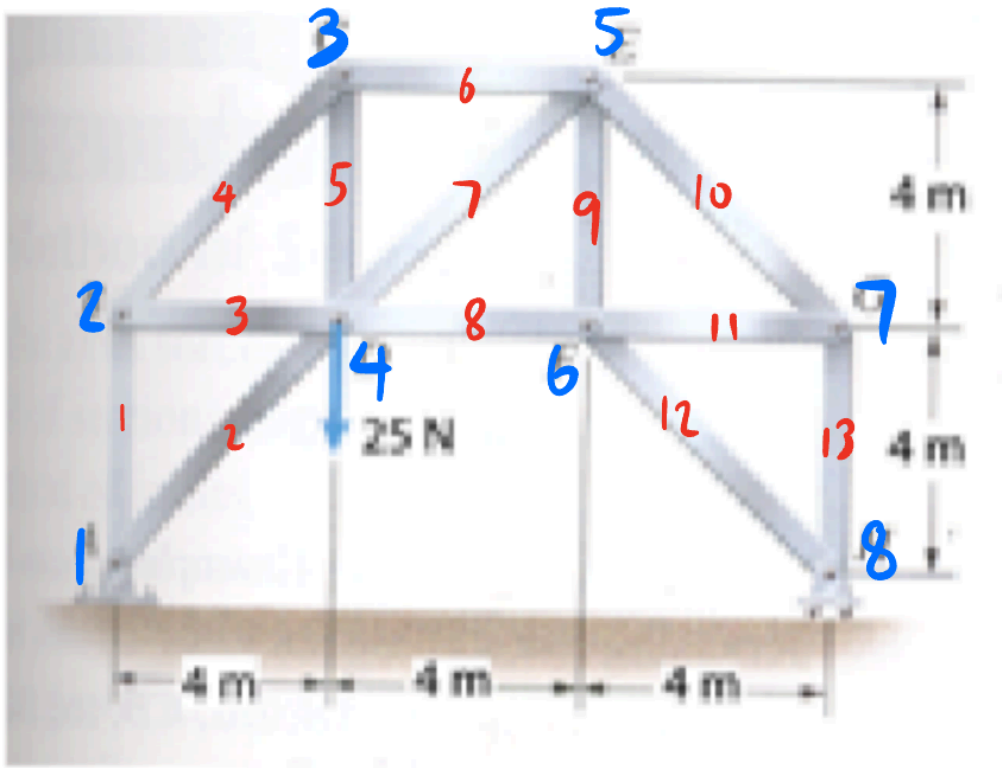


Figure 2: The example truss from the practice problem.

truss_import.m | buckme.m | checkCostAndMembers.m | eq_eqns.m | pcritCalc.m | Truss Data.m

```matlab
% make the equations
function [A, L] = eq_eqns(C, Sx, Sy, X, Y, L)
    % x,y are the num of joints and num of members
    [numJoints, numMembers] = size(C);

    % create A matrix of proper size in zeros
    A = zeros(2*numJoints, numMembers + 3);

    % go through the entire C matrix looking for 1s
    for joint = 1:numJoints
        for member = 1:numMembers
            if C(joint, member) == 1 % ie if connected at this point
                % important! have to find the other joint which is connected to
                % that member (meaning the other 1 in that col)
                otherJoint = find(C(:, member) & (1:numJoints)' ~= joint);

                % direction cosines for the current member
                deltaX = X(otherJoint) - X(joint);
                deltaY = Y(otherJoint) - Y(joint);
                memberLen = sqrt(deltaX^2 + deltaY^2);
                cosTheta = deltaX / memberLen; % exactly same calculation as on the manual
                sinTheta = deltaY / memberLen;

                % now add both to A matrix
                A(joint, member) = cosTheta; % Contribution to sum of forces in x-direction
                A(joint + numJoints, member) = sinTheta; % sum of F_y has to be offset by numJoints to match A's format
            end
        end

        % only 1 Sx support reaction below, add it to A matrix
        if Sx(joint)
            A(joint, numMembers+1) = 1; % adds to numMembers+1 col since thats where Sx goes in x component of A
        end

        % there are 2 Sy support reactions added to last 2 cols
        for k = 1:size(Sy, 2)
            if Sy(joint, k) == 1
                A(joint+numJoints, numMembers+k) = 1;  % adds needed 1 where its supposed to be in y component of A
            end
        end

    end

end
```

Figure 3: MATLAB code that will generate the equilibrium equations and place them in ***A***

```
% function that checks cost, member/joints, and returns the member lengths

function [totalCost, totalLength, memberLengths] = checkCostAndMembers(C, X, Y)
    % length of each member in an array (zeros for now)
    memberLengths = zeros(size(C,2), 1);

    % go thru every member (col) -- to find member lens
    for i = 1:size(C,2)
        joints = find(C(:,i)); % Find the joints connected to that member and save to joints
        if length(joints) == 2 % if there are two joints connected then you are good
            diffx = X(joints(2)) - X(joints(1));
            diffy = Y(joints(2)) - Y(joints(1));
            memberLengths(i) = sqrt(diffx^2 + diffy^2); % add the length of the member to the memberLengths array
        else
            fprintf('member %d is not connected to two joints!\n', i);
        end
    end

    totalLength = sum(memberLengths); % Total length of all members
    numJoints = size(C,1); % Total number of joints

    % total cost
    totalCost = 10 * numJoints + totalLength;

    % check member/joint ratio
    % M = 2J - 3
    if size(C,2) > 2 * size(C,1) - 3
        fprintf('too many members!: %d > %d\n', size(C,2), 2 * size(C,1) - 3);
    else
        fprintf('Members are within range: %d <= %d\n', size(C,2), 2 * size(C,1) - 3);
    end
end
```

Figure 4: MATLAB code that will check if the cost is within $300 and check if the members are within range.

```matlab
% find the critical member and MIN value of all the Wfailures among the members.
% therefore it is the max theoretical load that the truss can support before the weakest member buckles

function [critical_member, W_failure_min] = buckme(Pcrit, Rm, memberLens)
    % W_failure for each member
    Wfailure = -Pcrit ./ Rm;

    % ignore 0 force members (or positive ie tension) by setting value to inf
    Wfailure(Rm >=0) = Inf;
    % [min Wfailure value, index] of the critical member
    [W_failure_min, critical_member] = min(Wfailure);

    fprintf(['The member m%d with length %.3fin will theoretically buckle at maximum ' ...
        'force: %.2f oz +/-1.36 oz\n'], critical_member, memberLens(critical_member), W_failure_min);
end

```

```matlab
% calculates Pcrits and returns the matrix
function Pcrit = pcritCalc(memberLengths)
    % pcrit matrix
    Pcrit = zeros(length(memberLengths), 1);

    % Calculate Pcrit for each member
    for i = 1:length(memberLengths)
        Pcrit(i) = 3054.789 * (memberLengths(i)^(-2.009));
    end
end

```

Figure 5: MATLAB code that will determine the Pcrit and the maximum load for a member before failure.

truss_import.m | buckme.m | checkCostAndMembers.m | eq_eqns.m | pcritCalc.m | Truss Data.m

```matlab
fprintf('EK301, Section A2, Group 2: Kailan Pan, James Conlon, Austin Zhang, Due 4/5/2024\n\n');

%% TRUSS MATRIX SETUP

% connection matrix is 1 if there is a connection at that joint
% rows are joints, columns are members

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

% cols sum to 2 (down)
% rows sum to number of members attached to a joint (across)

% support matrices (only 3 ones total for 3 unknown support reactions)
%  Sx1, Sy1, Sy2

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

% joint coords

X = [0; 0; 4; 4; 8; 8; 12; 12];
Y = [0; 4; 8; 4; 8; 4; 4; 0];

% load vector
my_load = 99.69;

% create an empty load vector ( size(C,1) is j)
L = zeros(2*size(C,1), 1);

% Only one live load at a joint
% (numJoints + j_load) replace the joint where the load is at
L(12) = my_load;  % change as needed!!!
```

```matlab
%% calculations

% generate equalibrium equations
[A, L] = eq_eqns(C, Sx, Sy, X, Y, L);

% solve for member forces and the 3 reaction forces
% T is in this format: [ T_1-m, S_x1, S_y2, S_y2 ] (all internal forces)
T = A \ L;

% check cost and member/joint reqs
[totalCost, totalLength, memberLengths] = checkCostAndMembers(C, X, Y);

% Rm
Rm = T / my_load;
Rm_membersOnly = Rm(1:size(C,2)); % no reaction forces in Rm

% make pcrit matrix
[Pcrits] = pcritCalc(memberLengths);

% critical member and max theoretical load
[critical_member, W_failure_min] = buckme(Pcrits, Rm_membersOnly, memberLengths);

%% printing
fprintf('\nLoad: %.3f oz\n', my_load);
fprintf('Member forces in oz\n');
for i = 1:size(C,2)
    if T(i) == 0
        fprintf('m%d: %.3f (0 force member)\n', i, abs(T(i)));
    elseif T(i) < 0
        fprintf('m%d: %.3f (C)\n', i, abs(T(i))); % we in compression
    else
        fprintf('m%d: %.3f (T)\n', i, T(i)); % we in tension
    end
end


fprintf('Reaction forces in oz:\n');
fprintf('Sx1: %.2f\n', T(size(C,2)+1));
fprintf('Sy1: %.2f\n', T(size(C,2)+2));
fprintf('Sy2: %.2f\n', T(size(C,2)+3));


fprintf('Cost of truss: $%0.2f\n', totalCost);
load_to_cost = abs(W_failure_min) / totalCost;

fprintf('Theoretical max load/cost ratio in oz/$: %.4f\n', load_to_cost);
```

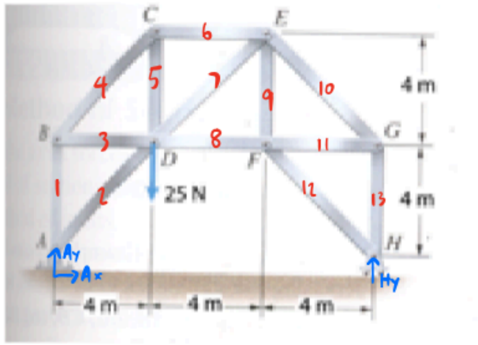
Figure 6: MATLAB code to import the truss design and print out if members are within range, the calculated forces experienced by each member, the maximum force a specific member will buckle, the cost of the truss, and theoretical max load-to-cost ratio.

EK301, Section A2, Group 2: Kailan Pan, James Conlon, Austin Zhang, Due 4/5/2024

Members are within range: 13 <= 13
Load: 25.000 N
Member forces in N
m1: 16.667 (C)
m2: 0.000 (0 force member)
m3: 16.667 (T)
m4: 23.570 (C)
m5: 16.667 (T)
m6: 16.667 (C)
m7: 11.785 (T)
m8: 8.333 (T)
m9: 0.000 (0 force member)
m10: 11.785 (C)
m11: 8.333 (T)
m12: 0.000 (0 force member)
m13: 8.333 (C)
Reaction forces in N:
Sx1: 0.00
Sy1: 16.67
Sy2: 8.33
Cost of truss: $140.28
The member m4 with length 5.657in will theoretically buckle at maximum force: 99.69 N +/-1.36 N
Load: 99.690 N
Member forces in N
m1: 66.460 (C)
m2: 0.000 (0 force member)
m3: 66.460 (T)
m4: 93.989 (C)
m5: 66.460 (T)
m6: 66.460 (C)
m7: 46.994 (T)
m8: 33.230 (T)
m9: 0.000 (0 force member)
m10: 46.994 (C)
m11: 33.230 (T)
m12: 0.000 (0 force member)
m13: 33.230 (C)
Reaction forces in N:
Sx1: 0.00
Sy1: 66.46
Sy2: 33.23
Theoretical max load/cost ratio in N/$: 0.7106

Figure 7: MATLAB calculation results (printed to console) for the practice example truss.

Determine the loads in each of the members and whether they are in tension or compression. Analyze the loads using yourselves (yes, that means do it out by hand) and MATLAB (results should match!).



Reaction Forces:

$\xrightarrow{+}\Sigma F_x=0$: $A_x=0$

$+\uparrow\Sigma F_y=0$: $A_y+H_y=25$ $\quad A_y=25-H_y$

$\circlearrowleft+\Sigma M_A=0$: $-25(4)+H_y(12)=0$

$H_y=8.33$ N

$A_y=16.67$ N

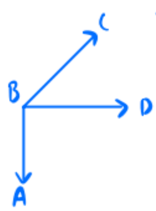
Joint A: $4^2+4^2=x^2$ $\quad x=5.657$

$\xrightarrow{+}\Sigma F_x=0$: $\overset{0}{\cancel{A_x}}+AD\frac{4}{5.657}=0$ $\quad AD=0$ N

$+\uparrow\Sigma F_y=0$: $A_y+AB+\overset{0}{\cancel{AD}}\frac{4}{5.657}=0$

$AB=-A_y$ $\quad AB=-16.67\text{N}=16.67\text{N (C)}$

Joint B:



$+\uparrow\Sigma F_y=0$: $-BA+BC\frac{4}{5.657}=0$

$BC=BA\frac{5.657}{4}$

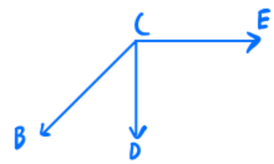
$BC=-23.57\text{N}=23.57\text{N (C)}$

$\xrightarrow{+}\Sigma F_x=0$: $BC\frac{4}{5.657}+BD=0$

$BD=-BC\frac{4}{5.657}$

$BD=16.67\text{N (T)}$

Joint C:



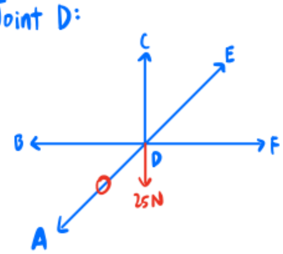
$\xrightarrow{+}\Sigma F_x=0$: $-CB\frac{4}{5.657}+CE=0$

$CE=CB\frac{4}{5.657}$ $\quad CE=-16.67\text{N}=16.67\text{N (C)}$

$+\uparrow\Sigma F_y=0$: $-CB\frac{4}{5.657}-CD=0$

$CD=-CB\frac{4}{5.657}$ $\quad CD=16.67\text{N (T)}$

Joint D:



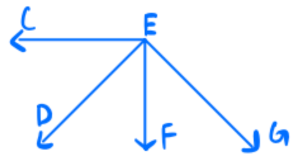
$+\uparrow\Sigma F_y=0$: $DC+DE\frac{4}{5.657}-25=0$

$DE=(-DC+25)\frac{5.657}{4}=11.78\text{ N (T)}$

$\xrightarrow{+}\Sigma F_x=0$: $-DB+DE\cdot\frac{4}{5.657}+DF=0$

$DF=DB-DE\cdot\frac{4}{5.657}=8.33\text{ N (T)}$

Joint E:



$$\xrightarrow{+}\sum F_x = 0: -EC - ED\frac{4}{5.657} + EG\frac{4}{5.657} = 0$$

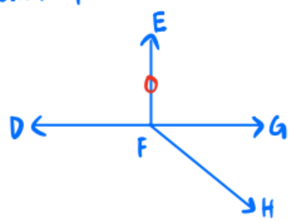
$$EG = \left(EC + ED\frac{4}{5.657}\right)\frac{5.657}{4}$$

$$EG = -11.78N = 11.78(C)$$

$$+\uparrow\sum F_y = 0: -ED\frac{4}{5.657} - EF - EG\frac{4}{5.657} = 0$$

$$EF = -ED\frac{4}{5.657} - EG\frac{4}{5.657} \qquad EF = 0\,N$$

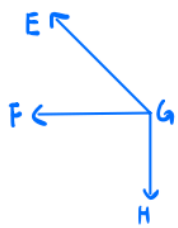
Joint F:



$$+\uparrow\sum F_y = 0: \overset{0}{\cancel{EF}} - FH\frac{4}{5.657} = 0 \qquad FH = 0\,N$$

$$\xrightarrow{+}\sum F_x = 0: FD = FG \qquad FG = 8.33N\,(T)$$

Joint G:



$$+\uparrow\sum F_y = 0: GE\frac{4}{5.657} - GH = 0$$

$$GH = GE\frac{4}{5.657} \qquad GH = -8.33N = 8.33N(C)$$

AB = m1 = 16.67N (C)	EF = m9 = 0
AD = m2 = 0	EC = m10 = 11.78N(C)
BD = m3 = 16.67 N(T)	FG = m11 = 8.33 N(T)
BC = m4 = 23.57 N (C)	FH = m12 = 0
CD = m5 = 16.67 N (T)	GH = m13 = 8.33N(C)
CE = m6 = 16.67 N (C)	
DE = m7 = 11.78N(T)	
DF = m8 = 8.33 N (T)	

Figure 8: Handwritten calculation results of the practice example truss.

EK301, Section A2, Group 2: Kailan Pan, James Conlon, Austin Zhang, Due 4/5/2024
Members are within range: 13 <= 13
Load: 32.000 oz
Member forces in oz
m1: 21.911 (C)
m2: 11.613 (T)
m3: 17.737 (T)
m4: 24.898 (C)
m5: 6.039 (T)
m6: 14.761 (T)
m7: 16.380 (T)
m8: 24.155 (C)
m9: 8.991 (C)
m10: 1.917 (T)
m11: 15.678 (T)
m12: 15.411 (C)
m13: 13.556 (C)
Reaction forces in oz:
Sx1: 0.00
Sy1: 18.58
Sy2: 13.42
The member m1 with length 9.434in will theoretically buckle at maximum force: 49.13 oz +/-1.36 oz
Load: 49.130 oz
Member forces in oz
m1: 33.641 (C)
m2: 17.829 (T)
m3: 27.232 (T)
m4: 38.227 (C)
m5: 9.271 (T)
m6: 22.663 (T)
m7: 25.149 (T)
m8: 37.085 (C)
m9: 13.804 (C)
m10: 2.943 (T)
m11: 24.071 (T)
m12: 23.661 (C)
m13: 20.812 (C)
Reaction forces in oz:
Sx1: 0.00
Sy1: 28.53
Sy2: 20.60
Cost of truss: $208.92
Theoretical max load/cost ratio in oz/$: 0.2351

Figure 9: MATLAB calculations for truss design 1

EK301, Section A2, Group 2: Kailan Pan, James Conlon, Austin Zhang, Due 4/5/2024
Members are within range: 15 <= 15
Load: 32.000 oz
Member forces in oz
m1: 11.318 (C)
m2: 9.650 (C)
m3: 6.355 (T)
m4: 18.921 (T)
m5: 17.816 (C)
m6: 18.395 (T)
m7: 4.044 (T)
m8: 18.581 (C)
m9: 14.378 (T)
m10: 13.419 (T)
m11: 8.097 (C)
m12: 5.964 (T)
m13: 10.138 (T)
m14: 13.338 (C)
m15: 14.685 (C)
Reaction forces in oz:
Sx1: 0.00
Sy1: 18.58
Sy2: 13.42
The member m5 with length 9.220in will theoretically buckle at maximum force: 63.27 oz +/-1.36 oz
Load: 63.270 oz
Member forces in oz
m1: 22.378 (C)
m2: 19.080 (C)
m3: 12.564 (T)
m4: 37.411 (T)
m5: 35.225 (C)
m6: 36.370 (T)
m7: 7.995 (T)
m8: 36.737 (C)
m9: 28.427 (T)
m10: 26.533 (T)
m11: 16.008 (C)
m12: 11.792 (T)
m13: 20.045 (T)
m14: 26.371 (C)
m15: 29.035 (C)
Reaction forces in oz:
Sx1: 0.00
Sy1: 36.74
Sy2: 26.53

Cost of truss: $250.25
Theoretical max load/cost ratio in oz/$: 0.2528

Figure 10: MATLAB calculations for truss design 2

References

[1] - Preliminary Design Manual from the preliminary design folder on Blackboard

[2] - Acrylic Data Fit from the preliminary design folder on Blackboard

[3] - Truss Practice Problem from the preliminary design folder on Blackboard