Fall MatLab Programming Project

EAS230 Engineering Computations
Fall 2019

Partner #1: Rhianna Militello, Lab A4, and rhiannam
Partner #2: Lauren McLaughlin-Kelly, Lab B2, and lmclaugh
rhiannam_lmclaugh

Part	Description of the breakdown of tasks
Script file	Partner 1 and 2 both contributed equally to all parts.
Function file	Partner 1 and 2 both contributed equally to all parts.
Please list other groups/individual that you may have worked with:	
None	

Table of Contents

List of Tables and Figures

Member	Force(N)	Deformation(%)
1	-10045.5	-0.050227
2	4773.5	0.023868
3	4773.5	0.023868
4	-10045.5	-0.050227
5	1804.2	0.009021
6	1804.2	0.009021
7	-10825.3	-0.054127
8	3608.4	0.018042
9	3608.4	0.018042
10	-10825.3	-0.054127
11	-4811.3	-0.024056
12	-6697.0	-0.033485
13	-6697.0	-0.033485

Table 1: Case 1: Medium Steel Tower Data. Page 28

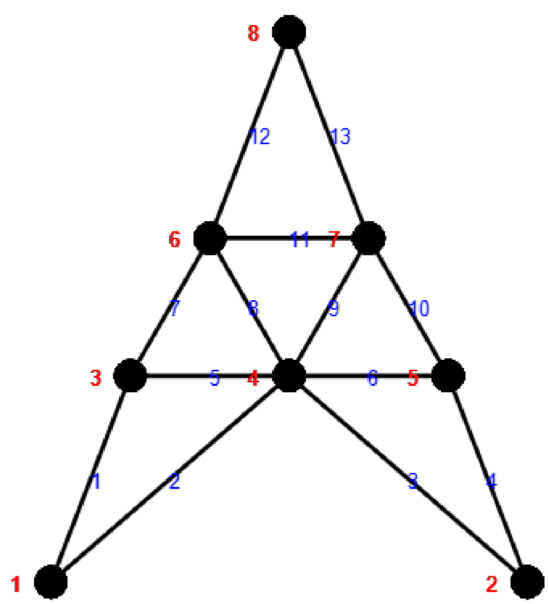


Figure 1: Case 1: Medium Steel Tower Structure. Page 28

Member	Force(N)	Deformation(%)
1	-25681.8	-0.229302
2	51363.6	0.458604
3	-12840.9	-0.114651
4	-64204.5	-0.573255

5	12840.9	0.114651
6	38522.7	0.343953
7	-12840.9	-0.114651
8	33973.8	0.303338
9	-51363.6	-0.458604

Table 2: Case 2: Titanium Crane with working Load of 5000lbs Data. Pages 29-30

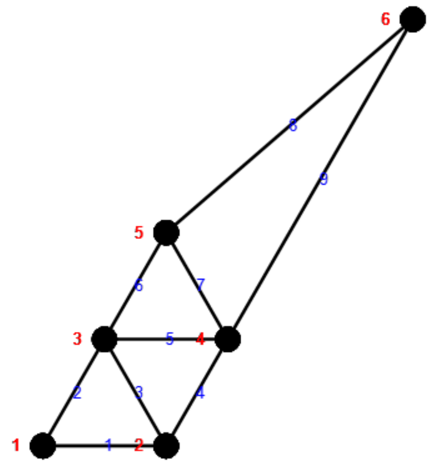


Figure 2: Case 2 Titanium Crane with working Load of 5000lbs Structure, Page 30

Member	Force(N)	Deformation(%)
1	-2742.4	-0.039745
2	1371.2	0.019873
3	2742.4	0.039745
4	-2742.4	-0.039745
5	-2742.4	-0.039745
6	4113.6	0.059618
7	2742.4	0.039745
8	-5484.8	-0.048972
9	-2742.4	-0.039745
10	6856.0	0.099363
11	2742.4	0.039745
12	-8227.2	-0.119235
13	-2742.4	-0.039745
14	9598.4	0.139108
15	2742.4	0.039745
16	-10969.7	-0.097943
17	-2742.4	-0.039745
18	12340.9	0.178853

19	2742.4	0.039745
20	-13712.1	-0.198726
21	-2742.4	-0.039745
22	15083.3	0.218598
23	2742.4	0.039745
24	-16454.5	-0.146915
25	-2742.4	-0.039745
26	17825.7	0.258343
27	2742.4	0.039745
28	-19196.9	-0.278216
29	2742.4	0.039745
30	17825.7	0.258343
31	-2742.4	-0.039745
32	-16454.5	-0.146915
33	2742.4	0.039745
34	15083.3	0.218598
35	-2742.4	-0.039745
36	-13712.1	-0.198726
37	2742.4	0.039745
38	12340.9	0.178853
39	-2742.4	-0.039745
40	-10969.7	-0.097943
41	2742.4	0.039745
42	9598.4	0.139108
43	-2742.4	-0.039745
44	-8227.2	-0.119235
45	2742.4	0.039745
46	6856.0	0.099363
47	-2742.4	-0.039745
48	-5484.8	-0.048972
49	2742.4	0.039745
50	4113.6	0.059618
51	-2742.4	-0.039745
52	-2742.4	-0.039745
53	2742.4	0.039745
54	1371.2	0.019873
55	-2742.4	-0.039745

Table 3: Case 3: Aluminum Bridge with Span of 40ft Data. Pages 31-32

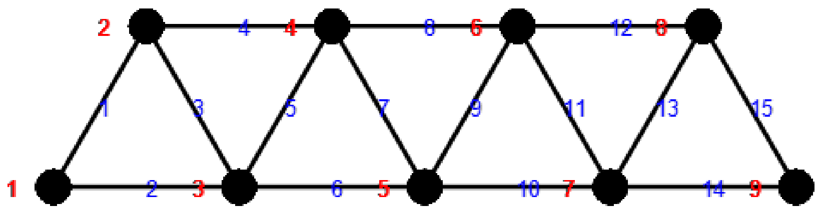


Figure 3: Case 3: Aluminum Bridge with Span of 40ft Structure. Page 33

Member	Force(N)	Deformation(%)
1	-41090.9	-0.366883
2	82181.8	0.733766
3	-20545.4	-0.183442
4	-102727.2	-0.917208
5	20545.4	0.183442
6	61636.3	0.550325
7	-20545.4	-0.183442
8	54358.1	0.485341
9	-82181.8	-0.733766

Table 4: Case 4: Titanium Crane with 8000lbs Data. Page 34

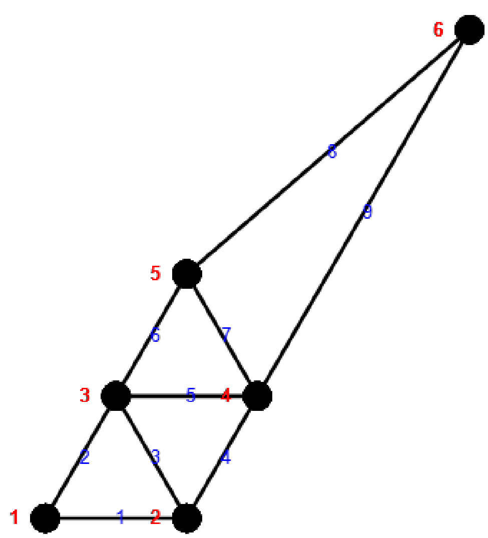


Figure 4: Case 4: Titanium Crane with 8000lbs Structure. Page 34

Member	Force(N)	Deformation(%)
1	-2742.4	-0.039745
2	1371.2	0.019873
3	2742.4	0.039745

4	-2742.4	-0.039745
5	-2742.4	-0.039745
6	4113.6	0.059618
7	2742.4	0.039745
8	-5484.8	-0.048972
9	-2742.4	-0.039745
10	6856.0	0.099363
11	2742.4	0.039745
12	-8227.2	-0.119235
13	-2742.4	-0.039745
14	9598.4	0.139108
15	2742.4	0.039745
16	-10969.7	-0.097943
17	-2742.4	-0.039745
18	12340.9	0.178853
19	2742.4	0.039745
20	-13712.1	-0.198726
21	-2742.4	-0.039745
22	15083.3	0.218598
23	2742.4	0.039745
24	-16454.5	-0.146915
25	-2742.4	-0.039745
26	17825.7	0.258343
27	2742.4	0.039745
28	-19196.9	-0.278216
29	-2742.4	-0.039745
30	20568.1	0.298088
31	2742.4	0.039745
32	-21939.3	-0.195887
33	-2742.4	-0.039745
34	23310.5	0.337833
35	2742.4	0.039745
36	-24681.7	-0.357706
37	-2742.4	-0.039745
38	26052.9	0.377579
39	2742.4	0.039745
40	-27424.1	-0.244858
41	-2742.4	-0.039745
42	28795.3	0.417324
43	2742.4	0.039745

44	-30166.5	-0.437196
45	-2742.4	-0.039745
46	31537.7	0.457069
47	2742.4	0.039745
48	-32909.0	-0.293830
49	-2742.4	-0.039745
50	34280.2	0.496814
51	2742.4	0.039745
52	-35651.4	-0.516686
53	-2742.4	-0.039745
54	37022.6	0.536559
55	2742.4	0.039745
56	-38393.8	-0.342802
57	-2742.4	-0.039745
58	39765.0	0.576304
59	2742.4	0.039745
60	-41136.2	-0.596177
61	-2742.4	-0.039745
62	42507.4	0.616049
63	2742.4	0.039745
64	-43878.6	-0.391773
65	-2742.4	-0.039745
66	45249.8	0.655794
67	2742.4	0.039745
68	-46621.0	-0.675667
69	-2742.4	-0.039745
70	47992.2	0.695539
71	2742.4	0.039745
72	-49363.4	-0.440745
73	-2742.4	-0.039745
74	50734.6	0.735285
75	2742.4	0.039745
76	-52105.8	-0.755157
77	-2742.4	-0.039745
78	53477.0	0.775030
79	2742.4	0.039745
80	-54848.3	-0.489717
81	-2742.4	-0.039745
82	56219.5	0.814775
83	2742.4	0.039745

84	-57590.7	-0.834647
85	-2742.4	-0.039745
86	58961.9	0.854520
87	2742.4	0.039745
88	-60333.1	-0.538688
89	2742.4	0.039745
90	58961.9	0.854520
91	-2742.4	-0.039745
92	-57590.7	-0.834647
93	2742.4	0.039745
94	56219.5	0.814775
95	-2742.4	-0.039745
96	-54848.3	-0.489717
97	2742.4	0.039745
98	53477.0	0.775030
99	-2742.4	-0.039745
100	-52105.8	-0.755157
101	2742.4	0.039745
102	50734.6	0.735285
103	-2742.4	-0.039745
104	-49363.4	-0.440745
105	2742.4	0.039745
106	47992.2	0.695539
107	-2742.4	-0.039745
108	-46621.0	-0.675667
109	2742.4	0.039745
110	45249.8	0.655794
111	-2742.4	-0.039745
112	-43878.6	-0.391773
113	2742.4	0.039745
114	42507.4	0.616049
115	-2742.4	-0.039745
116	-41136.2	-0.596177
117	2742.4	0.039745
118	39765.0	0.576304
119	-2742.4	-0.039745
120	-38393.8	-0.342802
121	2742.4	0.039745
122	37022.6	0.536559
123	-2742.4	-0.039745

124	-35651.4	-0.516686
125	2742.4	0.039745
126	34280.2	0.496814
127	-2742.4	-0.039745
128	-32909.0	-0.293830
129	2742.4	0.039745
130	31537.7	0.457069
131	-2742.4	-0.039745
132	-30166.5	-0.437196
133	2742.4	0.039745
134	28795.3	0.417324
135	-2742.4	-0.039745
136	-27424.1	-0.244858
137	2742.4	0.039745
138	26052.9	0.377579
139	-2742.4	-0.039745
140	-24681.7	-0.357706
141	2742.4	0.039745
142	23310.5	0.337833
143	-2742.4	-0.039745
144	-21939.3	-0.195887
145	2742.4	0.039745
146	20568.1	0.298088
147	-2742.4	-0.039745
148	-19196.9	-0.278216
149	2742.4	0.039745
150	17825.7	0.258343
151	-2742.4	-0.039745
152	-16454.5	-0.146915
153	2742.4	0.039745
154	15083.3	0.218598
155	-2742.4	-0.039745
156	-13712.1	-0.198726
157	2742.4	0.039745
158	12340.9	0.178853
159	-2742.4	-0.039745
160	-10969.7	-0.097943
161	2742.4	0.039745
162	9598.4	0.139108
163	-2742.4	-0.039745

164	-8227.2	-0.119235
165	2742.4	0.039745
166	6856.0	0.099363
167	-2742.4	-0.039745
168	-5484.8	-0.048972
169	2742.4	0.039745
170	4113.6	0.059618
171	-2742.4	-0.039745
172	-2742.4	-0.039745
173	2742.4	0.039745
174	1371.2	0.019873
175	-2742.4	-0.039745

Table 5: Case 5: Aluminum Bridge with 144ft Span Data. Pages 35-40

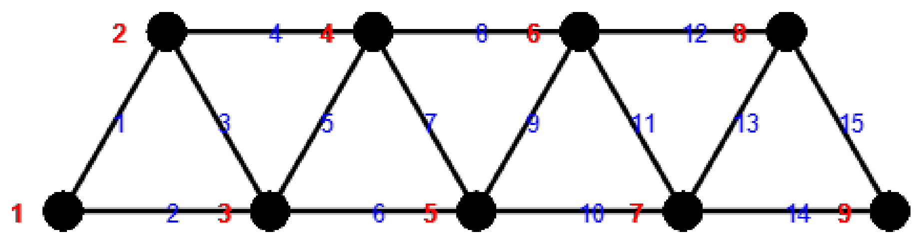


Figure 5: Case 5: Aluminum Bridge with 144ft Span Structure. Page 40

1. Summary/Abstract

The objective of the programming project was to design and construct metal truss structures varying in size from small to medium. The 3 different options of metal truss structures are: tower, crane, and a bridge/span structure, which can be constructed from aluminum, steel, and titanium. The problem in hand was each structure had different specificities about it, meaning they come in different sizes, have a different number of products, and every 8th member of the bridge/span must be titanium. We handled our approach by looking at the problem case by case at first, meaning we started with the bridge/span, then on the tower, and finally on the crane. The results showed the total price of each structure made from a specific material and any other constrictions, by adding material costs, fees, and taxes, to get a total cost. The results also showed the maximum load, the failure load, and maximum deformation each structure could withstand. The different structures varies a lot in cost and with their structural capabilities (maximum load and deformities).

2. Introduction

A company produces three truss structures that can be made of three different materials. Each structure has a minimum length which makes the cost different for each. Certain structures can have multiple minimum sizes based on the structure which can make finding the total distance for the entire structure difficult. We were asked to produce a MATLAB structure that can analyze these structures and compute an overall fabrication cost for the structure. These estimates, in turn, will be provided to clients.

3. Programming

3.1 F19_PP.m Script File

Inputs:

- Structure: This input defines the exact structure which is needed to find the lengths of all members.
- Material: The material type is needed in order to determine the entire price for the structure.
- Minimal Distance: For certain structures you have to enter the minimum length of a structure in order to determine the size of the overall structure.
- Working Load: Depending on the structure you need to enter the working load so we can determine the forces.
- Distance to Span: For certain structures the span is needed because they have a limit for how long they can be.

Loading Files:

- GeomtopoX: Each geomtop contains memnod, nodmem, x, and y. These are used to determine the number of members, nodes, and compute the distance of each member.
- Pricing: Contains the price per meter for each material type which is needed to determine the total cost.
- Materials: This provides Young's module, yield strength, and ultimate strength for each material type. This is needed to determine the max deformation of each member, the failure load, and the maximum load of each member.

Plotting:

In order to plot every structure, we needed to use memnod and nodmem as well as x and y for the certain geomtopo to determine where each node is located. This also allowed us to find how the members connected to each node.

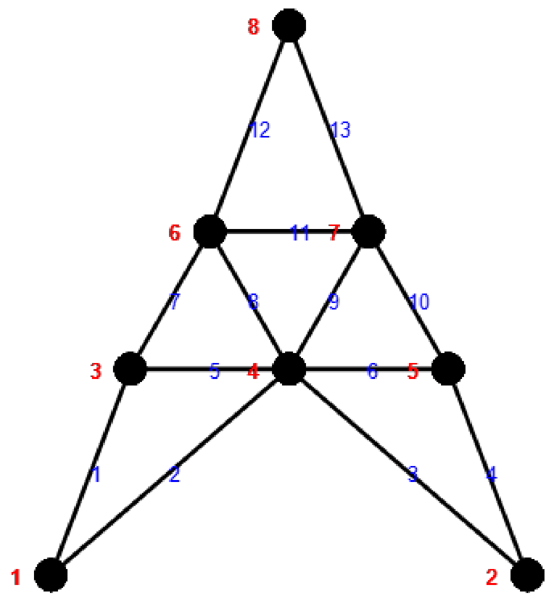
Calculations:

- Loads: Using the yield strength, the area factor, and the safety factor we can determine the max load. Using the ultimate yield, the area factor, and safety factor we can determine the failure load. In some cases, the max load equals the load but in other cases you have to input the load.
- Materials: Depending on the structure, the structure may be made out of multiple materials. The equation needed to determine this is given. This will mainly affect the cost.
- Clearance: In order to determine the clearance for each structure you need to use a certain, given clearance formula
- Footprints: In order to determine the footprint for each structure you need to use a certain, given footprint formula
- Costs: To calculate the cost you needed to use the given values from pricing.dat. In addition, you have to find the total distance of all members to multiply by that value. Then the tax can be found by multiplying this cost by .08. Finally, you are given a fee depending on the structure. All these values together give the total cost of the structure.

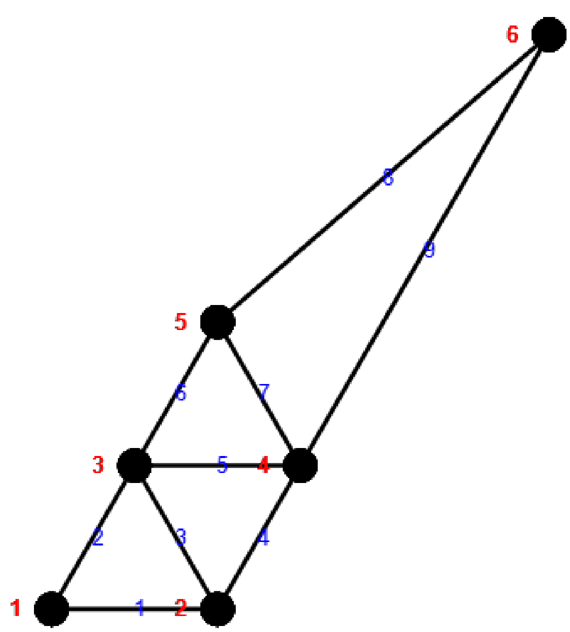
3.2 Structure_Analysis Function File

The Structure_Analysis calculate the different member forces, member deformations, and reaction forces of the different structures. The four different inputs are: structure ID, number of nodes N, estimated load, and a vector with Youngs Modulus of each member. The three outputs were: a vector of member deformations, a vector of member forces, and a vector of reaction forces. The structure ID has the scalar values of 1, 2, or 3, and allows the user to identify if the structure will be a tower, crane, or bridge/span. The number of nodes is used to identify the needed forces and to pair them with the right coefficient. The estimated load is used to calculate the forces needed by the user, which is applied on one node, vertically. The Young Modulus vector of each member is used after the forces are found, in order to find the deformations of each member. The vector of member deformation is based off the forces, meaning the member is either shrinking or stretching due the compression or tension, which is found using Youngs Modulus. The vector of member forces is each of the applied forces of the member structures, and how the length of the vector changes based off the number of nodes and the different structure used. The vector of reaction forces is required to always have a length of 3, because the reactions are at the supports, while the reaction forces are applied at two specific nodes.

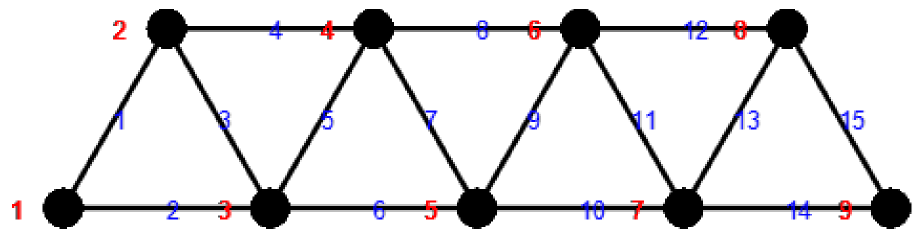
4. Results



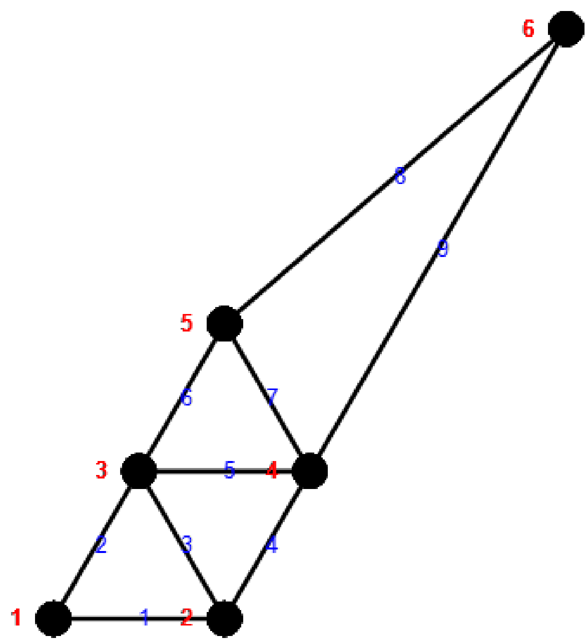
The Medium Steel Tower will have a cost of: Materials: $1471.08, Fees: $100.00, Tax: $117.69, Totaling out to: $1688.76. The maximum load is 2810lbs, the fail load is 4496lbs, and the maximum deformation is 0.0625%



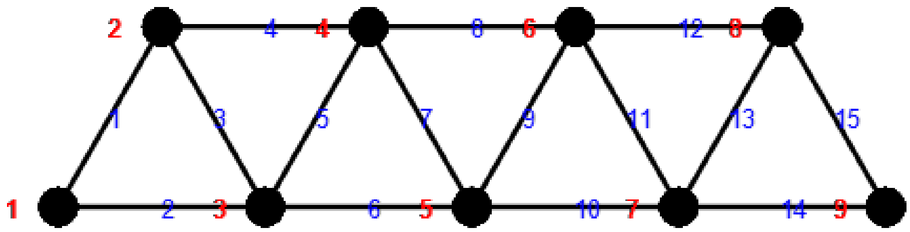
The Titanium Crane will have a cost of: Materials: $2617.04, Fees: $50.00, Tax: $209.36, Totaling out to: $2876.40. The maximum load is 5000lbs, the fail load is 10116lbs, and the maximum deformation is 0.3259%.



The Aluminum Bridge will have a cost of: Materials: $21799.80, Fees: $64.00, Tax: $1743.98, Totaling out to: $23607.78. The maximum load is 1068lb, the fail load is 1236lb, and the maximum deformation is 0.0688%.



The Titanium Crane will have a cost of: Materials: $2617.04, Fees: $50.00, Tax: $209.36, Totaling out to: $2876.40. The maximum load is 8000lbs, the fail load is 10116lbs, and the maximum deformation is 0.3259%.



The Aluminum Bridge will have a cost of: Materials: $ 69363.00, Fees: $94.00, Tax: $5549.04, Totaling out to: $75006.04. The maximum load is 1068lb, the fail load is 1236lb, and the maximum deformation is 0. 0688%.

5. Conclusions

We implemented the top-down design process by starting with what we felt most confident with, and then worked on the aspects we felt were more challenging. Some good programming practices we used for this project were utilizing our resources, such as our notes, the search bar in MatLab, and our teacher assistants. We also made sure to clearly define what each aspect of the code was supposed to do, in order to make sure we didn't forget anything or make simple mistakes. We also made sure to make no small errors. We did that by running our code very often to make sure the results were coming out properly, because if they weren't, it meant we made a mistake someplace. We checked our work very often so it would be easier to find where the error took place. These programming practices helped improve our scripts/functions because it allowed us to fix our errors right away, giving us a more accurate code. They also allowed us to see what we really struggled with, which allowed us to ask the teaching assistants for help with, right away so we could continue on with the project. We felt as if the structural analysis was definitely a harder aspect of the project due to the complexity of it. We also felt as if a lot of aspects of this project were very easy to overthink at first, but much simpler once we asked for clarification. The less challenging aspects were the initial steps, such as finding the clearance and footprint data, prompts, fabrication constraints, and loading the different files.

Appendices

Appendices

Appendix A

clc

```
%Step 1
load('materials.dat')
load('pricing.dat')
price=load('pricing.dat');
%Step 2
struct=input('********** MENU ********** \n\n Choose Structure:\n 1)Tower\n
2)Crane/Winch\n 3)Bridge/Span \n\n Enter a choice: ');
%Step 3
mater=input('\n********** MENU **********\n\nChoose Material:\n 1)Aluminum\n
2)Steel\n 3)Titanium \n\nEnter a choice: ');
%Step 4
yieldStrength = materials(:,3);
yieldStrength=yieldStrength*224808.943;
ultStrength = materials(:,4);
ultStrength=ultStrength*224808.943;
E=materials(:,2);
E=E*(2.248*(10^8));
a=1*(10^-4);
sf = 2;
switch mater
  case 1
    wMax = (a*yieldStrength(1))/sf;
    wFail =(a*ultStrength(1))/sf;
    defMax = wMax/(a*E(1));
    price=pricing(1,2);
    materString = 'Aluminum';
  case 2
    wMax = (a*yieldStrength(2))/sf;
    wFail =(a*ultStrength(2))/sf;
    defMax = wMax/(a*E(2));
    price=pricing(2,2);
    materString = 'Steel';
  case 3
    wMax = (a*yieldStrength(3))/sf;
    wFail =(a*ultStrength(3))/sf;
    defMax = wMax/(a*E(3));
    price=pricing(3,2);
    materString = 'Titanium';
end
%5to15
```

```
switch struct
  case 1
    structString='Tower';
    d=input('********* MENU **********\nChoose a size: \n 1)small (3.3 feet) \n 2)medium
(6.6 feet) \n 3)large (16.4 feet) \n\n Enter a choice: ');
    while d<1 || d>3
      d=input('********* MENU **********\nChoose a size: \n 1)small (3.3 feet) \n
2)medium (6.6 feet) \n 3)large (16.4 feet) \n\n Enter a choice: ');
    end
    switch d
      case 1
        d=1;
        fee=50;
      case 2
        d=2;
        fee=100;
      case 3
        d=5;
        fee=250;
    end
    S1=load('geomtopo1.mat');
    S1=struct2cell(S1);
    x1=cell2mat(S1(3,:));
    y1=cell2mat(S1(4,:));
    memnod=cell2mat(S1(1,:));
    M=length(memnod);
    for i = 1:M
      distance(i)=d*(sqrt((x1(memnod(i,2))-x1(memnod(i,1)))^2+(y1(memnod(i,2))-
y1(memnod(i,1)))^2));
    end
    W=wMax;
    h=(sqrt(3)*d)/2;
    H=4*h;
    L=3*d;
    matCost=price*sum(distance);
    tax = .08 * matCost;
    total = matCost+fee+tax;
    fprintf("\n****************************************\n\nYOU HAVE CHOSEN: %s %s
\n Clearance: %.0fft \n Footprint: %.0fft \n Max Load: %.0flb \n\n Max Rec'd Load: %.0flb
\n Max Deformation: %.4f%% \n Failure Load: %.0flb
```

```
\n\n*******************************************\n\n\nCost:\nMaterials: $ %.2f\nFees: $%.2f\nTax: $ %.2f\n---------------------\nTotal: $ %.2f\n ", materString, structString,round(H/.3048),round(L/.3048),wMax, wMax, defMax*100, wFail, matCost, fee, tax, total);
    wMax=4.44822*wMax;
    [Table,Deformation]=Structure_Analysis(struct,wMax,0, mater);
    Forces=Table(1:end-3);
    Reactions=Table(end-2:end);
    Deformation=Deformation(1:end-3);
    memberIndicator=[1:13]';
    disp('Member      Force(N)     Deformation(%)')
    for n=1:13
      fprintf('%i           %.1f        %.6f \n', memberIndicator(n), Forces(n), Deformation(n))
    end
    disp('Reactions(N)')
    reactionIndicator=[1:3]';
    for n=1:3
      if Reactions(n)<=1 & Reactions(n)>=-1
        fprintf('%i       %1.4e \n',reactionIndicator(n), Reactions(n))
      elseif Reactions(n)>0 | Reactions(n)<1
        fprintf('%i       %.0f \n',reactionIndicator(n), Reactions(n))
      end
    end
    load('geomtopo1.mat');
    for j=1:length(memnod(:,1))
      V= memnod(j,end);
      T= memnod(j,1);
      plot([x1(V),x1(T)],[y1(V),y1(T)],'k-','linewidth',2)
      chr=int2str(j);
      xTV= (x1(T) + x1(V))/2;
      yTV= (y1(T) +y1(V))/2;
      text(xTV,yTV,chr,'color','blue')
      hold on
      for i=1:length(nodmem)
        plot(x1(i),y1(i),'k.','MarkerSize',50)
        chr1=int2str(i);
        text(x1(i)-.25,y1(i),chr1,'Color','red')
      end
      axis equal
      axis off
```

```
        end
    case 2
        structString='Crane/Winch';
        W=input('\n********* MENU **********\n\nEnter the expected working load (In
LB)\n(0 < mass < 8203): ');
        while W<0 || W>wMax
            W=input('\n********* MENU **********\n\nEnter the expected working load (In
LB)\n(0 < mass < 8203): ');
        end
        d=1;
        S2=load('geomtopo2.mat');
        S2=struct2cell(S2);
        x2=cell2mat(S2(3,:));
        y2=cell2mat(S2(4,:));
        memnod=cell2mat(S2(1,:));
        M=length(memnod);
        for i = 1:M
            distance(i)=d*(sqrt((x2(memnod(i,2))-x2(memnod(i,1)))^2+(y2(memnod(i,2))-
y2(memnod(i,1)))^2));
        end
        fee=50;
        h=(sqrt(3)*d)/2;
        H=4*h;
        L=3*d;
        matCost=price*sum(distance);
        tax = .08 * matCost;
        total = matCost+fee+tax;
        fprintf("\n****************************************\n\nYOU HAVE CHOSEN: %s %s
\n Clearance: %.0fft \n Footprint: %.0fft \n Max Load: %.0flb \n\n Max Rec'd Load: %.0flb
\n Max Deformation: %.4f%% \n Failure Load: %.0flb
\n\n*******************************************\n\n\nCost:\nMaterials: $ %.2f\nFees:
$%.2f\nTax: $ %.2f\n---------------------\nTotal: $ %.2f\n ", materString,
structString,round(H/.3048),round(L/.3048),W,wMax, defMax*100, wFail, matCost, fee,
tax, total);
        load('geomtopo2.mat');
        W=4.44822*W;
        [Table,Deformation]=Structure_Analysis(struct, W, 0, mater);
        Forces=Table(1:end-3);
        Reactions=Table(end-2:end);
        Deformation=Deformation(1:end-3);
```

```
    memberIndicator=[1:9]';
    disp('Member      Force(N)     Deformation(%)')
    for n=1:9
      fprintf('%i           %.1f        %.6f \n', memberIndicator(n), Forces(n), Deformation(n))
    end
    disp('Reactions(N)')
    reactionIndicator=[1:3]';
    for n=1:3
      if Reactions(n)<=1 & Reactions(n)>=-1
        fprintf('%i      %1.4e \n',reactionIndicator(n), Reactions(n))
      elseif Reactions(n)>1 | Reactions(n)<1
        fprintf('%i      %.0f \n',reactionIndicator(n), Reactions(n))
      end
    end
    for j=1:length(memnod(:,1))
      V= memnod(j,end);
      T= memnod(j,1);
      plot([x2(V),x2(T)],[y2(V),y2(T)],'k-','linewidth',2)
      chr=int2str(j);
      xTV= (x2(T) + x2(V))/2;
      yTV= (y2(T) +y2(V))/2;
      text(xTV,yTV,chr,'color','blue')
      hold on
      for i=1:length(nodmem)
        plot(x2(i),y2(i),'k.','MarkerSize',50)
        chr1=int2str(i);
        text(x2(i)-.25,y2(i),chr1,'Color','red')
      end
      axis equal
      axis off
    end
  case 3
    structString='Bridge/Span';
    L=input('Enter the distance to span L (in feet): ')*1.1;
    while L>164/1.1
      L=input('DISTANCE TOO LONG OR TOO SHORT\nHOW WIDE DO YOU NEED YOUR
BRIDGE/SPAN? (< 164 FEET): ');
    end
    S3=load('geomtopo3.mat');
    S3=struct2cell(S3);
```

```
    x3=cell2mat(S3(3,:));
    y3=cell2mat(S3(4,:));
    memnod=cell2mat(S3(1,:));
    M=length(memnod);
    for i = 1:M
      distance(i)=L*(sqrt((x3(memnod(i,2))-x3(memnod(i,1)))^2+(y3(memnod(i,2))-
y3(memnod(i,1)))^2));
    end
    h=(sqrt(3)*1)/2;
    H=h;
    L1=L*.3048;
    Lfootprint=h*3.28084;
    if mod(L1,1)~=0
      L1=ceil(L1);
    end
    if mod(L1,2)~=0
      L1=L1+1;
    end
    L=L1*3.28084;
    u=L1/2;
    lengthTi=u-1;
    lengthOther=7*u;
    load('pricing.dat')
    if mater==1
      matCost=lengthTi*pricing(3,2)+lengthOther*pricing(1,2);
    elseif mater==2
      matCost=lengthTi*pricing(3,2)+lengthOther*pricing(2,2);
    elseif mater==3
      matCost=lengthTi*pricing(3,2)+lengthOther*pricing(3,2);
    end
    W=wMax;
    fee=52+2*(u-1);
    matCost=price*sum(distance);
    tax = .08*matCost;
    total = matCost+fee+tax;
    Lfootprint=L;
    fprintf("\n*******************************************\n\nYOU HAVE CHOSEN: %s %s
\n Clearance: %.0fft \n Footprint: %.0fft \n Max Load: %.0flb \n\n Max Rec'd Load: %.0flb
\n Max Deformation: %.4f%% \n Failure Load: %.0flb
\n\n*******************************************\n\n\nCost:\nMaterials: $ %.2f\nFees:
```

```
$%.2f\nTax: $ %.2f\n---------------------\nTotal: $ %.2f\n ", materString,
structString,round(H/.3048),Lfootprint,W,wMax, defMax*100, wFail, matCost, fee, tax,
total);
    wMax=wMax*4.44822;
    [Table,Deformation]=Structure_Analysis(struct,wMax,u,mater);
    Forces=Table(1:end-3);
    Reactions=Table(end-2:end);
    Deformation=Deformation(1:end-3);
    disp('Member      Force(N)      Deformation(%)')
    for n=1:length(Forces)
      fprintf('%i            %.1f        %.6f \n', n, Forces(n), Deformation(n))
    end
    disp('Reactions(N)')
    reactionIndicator=[1:3]';
    for n=1:3
      if Reactions(n)<=1 & Reactions(n)>=-1
        fprintf('%i       %1.4e \n', n, Reactions(n))
      elseif Reactions(n)>0 | Reactions(n)<1
        fprintf('%i       %.0f \n', n, Reactions(n))
      end
    end
    load('geomtopo3.mat');
    for j=1:length(memnod(:,1))
      V= memnod(j,end);
      T= memnod(j,1);
      plot([x3(V),x3(T)],[y3(V),y3(T)],'k-','linewidth',2)
      chr=int2str(j);
      xTV= (x3(T) + x3(V))/2;
      yTV= (y3(T) +y3(V))/2;
      text(xTV,yTV,chr,'color','blue')
      hold on
      for i=1:length(nodmem)
        plot(x3(i),y3(i),'k.','MarkerSize',50)
        chr1=int2str(i);
        text(x3(i)-.25,y3(i),chr1,'Color','red')
      end
      axis equal
      axis off
    end
end
```

Appendix B

```
function [Table,Deformation] = Structure_Analysis(System_type, w, u, Material)
switch System_type
  case 1
    alpha=atan((3*sqrt(3))/2); B=atan(sqrt(3)/2); t=pi/3;
    A=[cos(alpha) cos(B) zeros(1,11) 1 zeros(1,2)
      sin(alpha) sin(B) zeros(1,12) 1 zeros(1,1)
      zeros(1,2) -cos(B) -cos(alpha) zeros(1,12)
      zeros(1,2) sin(B) sin(alpha) zeros(1,11) 1
      -cos(alpha) zeros(1,3) 1 0 cos(t) zeros(1,9)
      -sin(alpha) zeros(1,5) sin(t) zeros(1,9)
      0 -cos(B) cos(B) 0 -1 1 0 -cos(t) cos(t) zeros(1,7)
      0 -sin(B) -sin(B) zeros(1,4) sin(t) sin(t) zeros(1,7)
      zeros(1,3) cos(alpha) 0 -1 zeros(1,3) -cos(t) zeros(1,6)
      zeros(1,3) -sin(alpha) zeros(1,5) sin(t) zeros(1,6)
      zeros(1,6) -cos(t) cos(t) 0 0 1 cos(alpha) zeros(1,4)
      zeros(1,6) -sin(t) -sin(t) zeros(1,3) sin(alpha) zeros(1,4)
      zeros(1,8) -cos(t) cos(t) -1 0 -cos(alpha) zeros(1,3)
      zeros(1,8) -sin(t) -sin(t) zeros(1,2) sin(alpha) zeros(1,3)
      zeros(1,11) -cos(alpha) cos(alpha) zeros(1,3)
      zeros(1,11) -sin(alpha) -sin(alpha) zeros(1,3)];
    b=[zeros(15,1);w];
    Table=A\b;
    load('materials.dat')
    if Material==1
      E=materials(1,2)*(10^9);
      a=0.0001;
      Deformation=(Table./(E*a))*100;
    elseif Material==2
      E=materials(2,2)*(10^9);
      a=0.0001;
      Deformation=(Table./(E*a))*100;
    elseif Material==3
      E=materials(3,2)*(10^9);
      a=0.0001;
      Deformation=(Table./(E*a))*100;
    end
  case 2
    lambda=atan(sqrt(3)/2); t=(pi/3);
    A=[1 cos(t) zeros(1,7) 1 zeros(1,2)
```

```
        0 sin(t) zeros(1,8) 1 zeros(1,1)
        -1 0 -cos(t) cos(t) zeros(1,8)
        0 0 sin(t) sin(t) zeros(1,7) 1
        0 -cos(t) cos(t) 0 1 cos(t) zeros(1,6)
        0 -sin(t) -sin(t) 0 0 sin(t) zeros(1,6)
        zeros(1,3) -cos(t) -1 0 -cos(t) 0 cos(t) zeros(1,3)
        zeros(1,3) -sin(t) 0 0 sin(t) 0 sin(t) zeros(1,3)
        zeros(1,5) -cos(t) cos(t) cos(lambda) zeros(1,4)
        zeros(1,5) -sin(t) -sin(t) sin(lambda) zeros(1,4)
        zeros(1,7) -cos(lambda) -cos(t) zeros(1,3)
        zeros(1,7) -sin(lambda) -sin(t) zeros(1,3)];
      b=[zeros(11,1);w];
      Table=A\b;
      load('materials.dat')
      if Material==1
        E=materials(1,2)*(10^9);
        a=0.0001;
        Deformation=(Table./(E*a))*100;
      elseif Material==2
        E=materials(2,2)*(10^9);
        a=0.0001;
        Deformation=(Table./(E*a))*100;
      elseif Material==3
        E=materials(3,2)*(10^9);
        a=0.0001;
        Deformation=(Table./(E*a))*100;
      end
    case 3
      t=(pi/3);
      Node_number=4*u+1;
      A=zeros(2*Node_number);
      A(1,1)=cos(t); A(1,2)=1; A(1,end-2)=1;
      A(2,1)=sin(t); A(2,end-1)=1;
      A(3,1)=-cos(t); A(3,3)=cos(t); A(3,4)=1;
      A(4,1)=-sin(t); A(4,3)=-sin(t);
      A(end,end)=1; A(end,end-3)=sin(t);
      A(end-1,end-4)=-1; A(end-1,end-3)=-cos(t);
      A(end-2,end-5)=-sin(t); A(end-2,end-3)=-sin(t);
      A(end-3,end-6)=-1; A(end-3, end-5)=-cos(t); A(end-3, end-3)=cos(t);
      for n=3:(Node_number-2)
```

```
        if mod(n,2)==0
          A((2*n-1),(2*n-4))=-1; A((2*n-1),(2*n-3))=-cos(t); A((2*n-1),(2*n-1))=cos(t);
A((2*n-1),(2*n))=1;
          A((2*n),(2*n-3))=sin(t); A((2*n),(2*n-1))=sin(t);
        elseif mod(n,2)~=0
          A((2*n-1),(2*n-4))=-1; A((2*n-1),(2*n-3))=-cos(t); A((2*n-1),(2*n-1))=cos(t);
A((2*n-1),(2*n))=1;
          A((2*n),(2*n-3))=sin(t); A((2*n),(2*n-1))=sin(t);
          if n==(Node_number+1)/2
            b=zeros((2*Node_number),1);
            b(2*n,1)=w;
          end
        end
      end
      Table=A\b; %%%%Bellow change all tables
      load('materials.dat')
      if Material==1
        E_Material_Chosen=materials(1,2)*(10^9);
        E_Ti=materials(3,2)*(10^9);
        a=0.0001;
        for n=1:length(Table)
          if mod(n,8)==0
            Deformation(n)=(Table(n)/(E_Ti*a))*100;
          else
            Deformation(n)=(Table(n)/(E_Material_Chosen*a))*100;
          end
        end
      elseif Material==2
        E_Material_Chosen=materials(2,2)*(10^9);
        E_Ti=materials(3,2)*(10^9);
        a=0.0001;
        for n=1:length(Table)
          if mod(n,8)==0
            Deformation(n)=(Table(n)/(E_Ti*a))*100;
          else
            Deformation(n)=(Table(n)/(E_Material_Chosen*a))*100;
          end
        end
      elseif Material==3
        E_Material_Chosen=materials(3,2)*(10^9);
```

```
        a=0.0001;
        Deformation=(Table./(E_Material_Chosen*a))*100;
    end
end
end
```

Case 1:

```
********** MENU **********
 Choose Structure:
 1)Tower
 2)Crane/Winch
 3)Bridge/Span
 Enter a choice: 1
********** MENU **********
Choose Material:
 1)Aluminum
 2)Steel
 3)Titanium
Enter a choice: 2
********* MENU **********
Choose a size:
 1)small (3.3 feet)
 2)medium (6.6 feet)
 3)large (16.4 feet)
 Enter a choice: 2
******************************************
YOU HAVE CHOSEN: Steel Tower
 Clearance: 23ft
 Footprint: 20ft
 Max Load: 2810lb

 Max Rec'd Load: 2810lb
 Max Deformation: 0.0625%
 Failure Load: 4496lb
*********************************************
Cost:
Materials: $ 1471.08
Fees: $100.00
Tax: $ 117.69
```

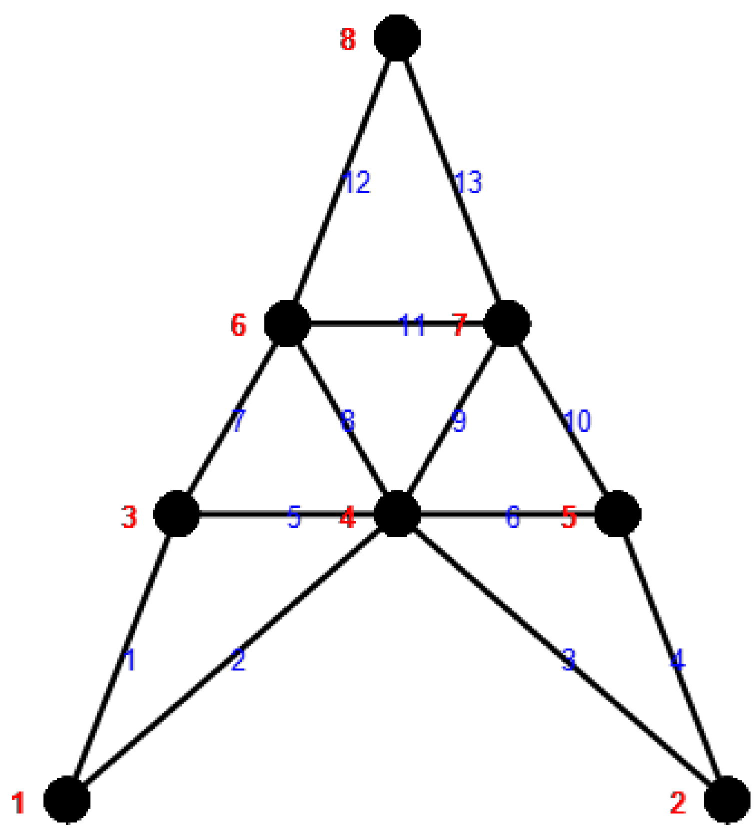
EAS230 Fall 2019 Programming Project
GroupName: rhiannam_lmclaugh

Total: $ 1688.76

Member	Force(N)	Deformation(%)
1	-10045.5	-0.050227
2	4773.5	0.023868
3	4773.5	0.023868
4	-10045.5	-0.050227
5	1804.2	0.009021
6	1804.2	0.009021
7	-10825.3	-0.054127
8	3608.4	0.018042
9	3608.4	0.018042
10	-10825.3	-0.054127
11	-4811.3	-0.024056
12	-6697.0	-0.033485
13	-6697.0	-0.033485

Reactions(N)

1	0.0000e+00
2	6250
3	6250



Case 2:

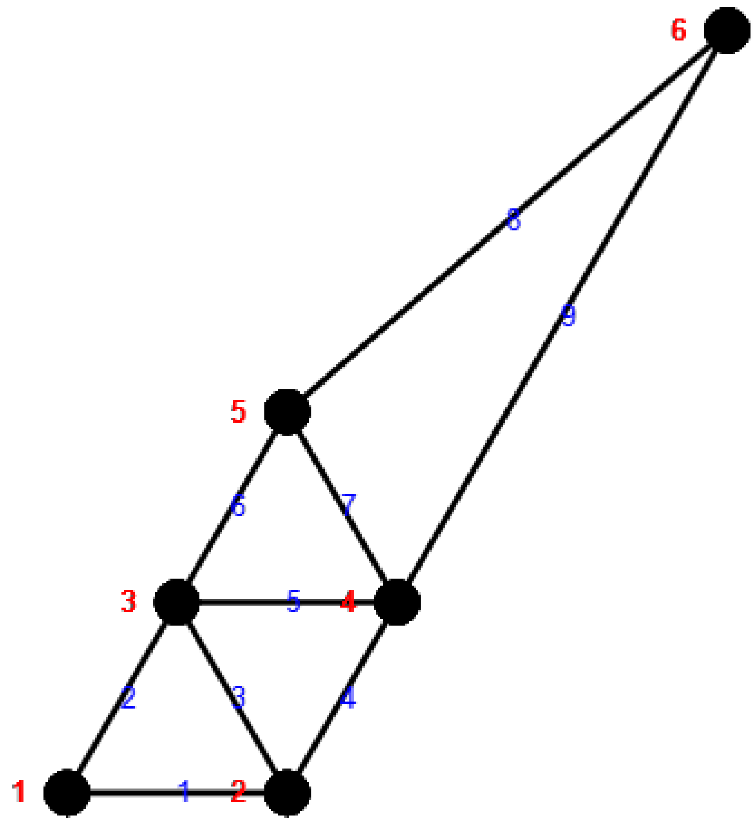
```
********** MENU **********
 Choose Structure:
 1)Tower
 2)Crane/Winch
 3)Bridge/Span
 Enter a choice: 2
********** MENU **********
Choose Material:
 1)Aluminum
 2)Steel
 3)Titanium
Enter a choice: 3
********* MENU **********
Enter the expected working load (In LB)
(0 < mass < 8203): 5000
*****************************************
YOU HAVE CHOSEN: Titanium Crane/Winch
 Clearance: 11ft
 Footprint: 10ft
 Max Load: 5000lb

 Max Rec'd Load: 8206lb
 Max Deformation: 0.3259%
 Failure Load: 10116lb
********************************************
Cost:
Materials: $ 2617.04
Fees: $50.00
Tax: $ 209.36
----------------------
Total: $ 2876.40
 Member     Force(N)     Deformation(%)
1          -25681.8       -0.229302
2          51363.6        0.458604
3          -12840.9       -0.114651
4          -64204.5       -0.573255
5          12840.9        0.114651
6          38522.7        0.343953
7          -12840.9       -0.114651
```

8	33973.8	0.303338
9	-51363.6	-0.458604

Reactions(N)

1	5.4570e-12
2	-44482
3	66723



Case 3:

********** MENU **********

Choose Structure:
1)Tower
2)Crane/Winch
3)Bridge/Span
Enter a choice: 3

********** MENU **********

Choose Material:
1)Aluminum
2)Steel
3)Titanium
Enter a choice: 1
Enter the distance to span L (in feet): 40

EAS230 Fall 2019 Programming Project
GroupName: rhiannam_lmclaugh

****

YOU HAVE CHOSEN: Aluminum Bridge/Span
Clearance: 3ft
Footprint: 46ft
Max Load: 1068lb

Max Rec'd Load: 1068lb
Max Deformation: 0.0688%
Failure Load: 1236lb
****

Cost:
Materials: $ 21799.80
Fees: $64.00
Tax: $ 1743.98

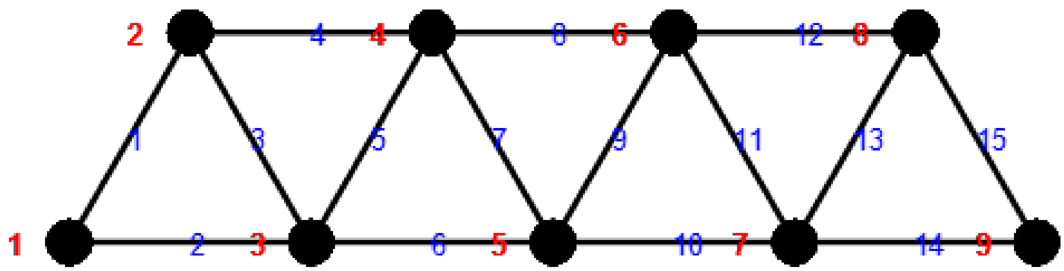
Total: $ 23607.78

Member	Force(N)	Deformation(%)
1	-2742.4	-0.039745
2	1371.2	0.019873
3	2742.4	0.039745
4	-2742.4	-0.039745
5	-2742.4	-0.039745
6	4113.6	0.059618
7	2742.4	0.039745
8	-5484.8	-0.048972
9	-2742.4	-0.039745
10	6856.0	0.099363
11	2742.4	0.039745
12	-8227.2	-0.119235
13	-2742.4	-0.039745
14	9598.4	0.139108
15	2742.4	0.039745
16	-10969.7	-0.097943
17	-2742.4	-0.039745
18	12340.9	0.178853
19	2742.4	0.039745
20	-13712.1	-0.198726
21	-2742.4	-0.039745
22	15083.3	0.218598
23	2742.4	0.039745

24	-16454.5	-0.146915
25	-2742.4	-0.039745
26	17825.7	0.258343
27	2742.4	0.039745
28	-19196.9	-0.278216
29	2742.4	0.039745
30	17825.7	0.258343
31	-2742.4	-0.039745
32	-16454.5	-0.146915
33	2742.4	0.039745
34	15083.3	0.218598
35	-2742.4	-0.039745
36	-13712.1	-0.198726
37	2742.4	0.039745
38	12340.9	0.178853
39	-2742.4	-0.039745
40	-10969.7	-0.097943
41	2742.4	0.039745
42	9598.4	0.139108
43	-2742.4	-0.039745
44	-8227.2	-0.119235
45	2742.4	0.039745
46	6856.0	0.099363
47	-2742.4	-0.039745
48	-5484.8	-0.048972
49	2742.4	0.039745
50	4113.6	0.059618
51	-2742.4	-0.039745
52	-2742.4	-0.039745
53	2742.4	0.039745
54	1371.2	0.019873
55	-2742.4	-0.039745

Reactions(N)

1	0.0000e+00
2	2375
3	2375



Case 4:

********** MENU **********

Choose Structure:

1)Tower

2)Crane/Winch

3)Bridge/Span

Enter a choice: 2

********** MENU **********

Choose Material:

1)Aluminum

2)Steel

3)Titanium

Enter a choice: 3

********* MENU **********

Enter the expected working load (In LB)

(0 < mass < 8203): 9000

********* MENU **********

Enter the expected working load (In LB)

(0 < mass < 8203): 8000

****

YOU HAVE CHOSEN: Titanium Crane/Winch

Clearance: 11ft

Footprint: 10ft

Max Load: 8000lb

Max Rec'd Load: 8206lb

Max Deformation: 0.3259%

Failure Load: 10116lb

****

Cost:

Materials: $ 2617.04

Fees: $50.00

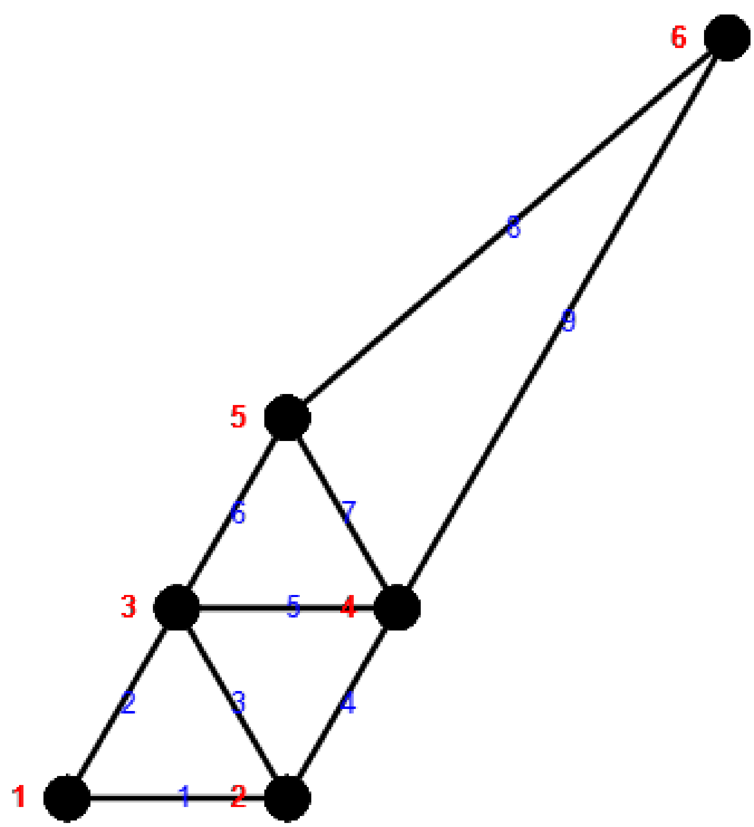
Tax: $ 209.36

Total: $ 2876.40

Member	Force(N)	Deformation(%)
1	-41090.9	-0.366883
2	82181.8	0.733766
3	-20545.4	-0.183442
4	-102727.2	-0.917208
5	20545.4	0.183442
6	61636.3	0.550325
7	-20545.4	-0.183442
8	54358.1	0.485341
9	-82181.8	-0.733766

Reactions(N)

1	-1.0914e-11
2	-71172
3	106757



Case 5:

********** MENU **********

Choose Structure:

1)Tower
2)Crane/Winch
3)Bridge/Span
Enter a choice: 3
********** MENU **********
Choose Material:
1)Aluminum
2)Steel
3)Titanium
Enter a choice: 1
Enter the distance to span L (in feet): 170
DISTANCE TOO LONG OR TOO SHORT
HOW WIDE DO YOU NEED YOUR BRIDGE/SPAN? (< 164 FEET): 160
DISTANCE TOO LONG OR TOO SHORT
HOW WIDE DO YOU NEED YOUR BRIDGE/SPAN? (< 164 FEET): 150
DISTANCE TOO LONG OR TOO SHORT
HOW WIDE DO YOU NEED YOUR BRIDGE/SPAN? (< 164 FEET): 140
****
YOU HAVE CHOSEN: Aluminum Bridge/Span
Clearance: 3ft
Footprint: 144ft
Max Load: 1068lb

Max Rec'd Load: 1068lb
Max Deformation: 0.0688%
Failure Load: 1236lb
****
Cost:
Materials: $ 69363.00
Fees: $94.00
Tax: $ 5549.04

Total: $ 75006.04

Member	Force(N)	Deformation(%)
1	-2742.4	-0.039745
2	1371.2	0.019873
3	2742.4	0.039745
4	-2742.4	-0.039745
5	-2742.4	-0.039745
6	4113.6	0.059618

7	2742.4	0.039745
8	-5484.8	-0.048972
9	-2742.4	-0.039745
10	6856.0	0.099363
11	2742.4	0.039745
12	-8227.2	-0.119235
13	-2742.4	-0.039745
14	9598.4	0.139108
15	2742.4	0.039745
16	-10969.7	-0.097943
17	-2742.4	-0.039745
18	12340.9	0.178853
19	2742.4	0.039745
20	-13712.1	-0.198726
21	-2742.4	-0.039745
22	15083.3	0.218598
23	2742.4	0.039745
24	-16454.5	-0.146915
25	-2742.4	-0.039745
26	17825.7	0.258343
27	2742.4	0.039745
28	-19196.9	-0.278216
29	-2742.4	-0.039745
30	20568.1	0.298088
31	2742.4	0.039745
32	-21939.3	-0.195887
33	-2742.4	-0.039745
34	23310.5	0.337833
35	2742.4	0.039745
36	-24681.7	-0.357706
37	-2742.4	-0.039745
38	26052.9	0.377579
39	2742.4	0.039745
40	-27424.1	-0.244858
41	-2742.4	-0.039745
42	28795.3	0.417324
43	2742.4	0.039745
44	-30166.5	-0.437196
45	-2742.4	-0.039745
46	31537.7	0.457069

47	2742.4	0.039745
48	-32909.0	-0.293830
49	-2742.4	-0.039745
50	34280.2	0.496814
51	2742.4	0.039745
52	-35651.4	-0.516686
53	-2742.4	-0.039745
54	37022.6	0.536559
55	2742.4	0.039745
56	-38393.8	-0.342802
57	-2742.4	-0.039745
58	39765.0	0.576304
59	2742.4	0.039745
60	-41136.2	-0.596177
61	-2742.4	-0.039745
62	42507.4	0.616049
63	2742.4	0.039745
64	-43878.6	-0.391773
65	-2742.4	-0.039745
66	45249.8	0.655794
67	2742.4	0.039745
68	-46621.0	-0.675667
69	-2742.4	-0.039745
70	47992.2	0.695539
71	2742.4	0.039745
72	-49363.4	-0.440745
73	-2742.4	-0.039745
74	50734.6	0.735285
75	2742.4	0.039745
76	-52105.8	-0.755157
77	-2742.4	-0.039745
78	53477.0	0.775030
79	2742.4	0.039745
80	-54848.3	-0.489717
81	-2742.4	-0.039745
82	56219.5	0.814775
83	2742.4	0.039745
84	-57590.7	-0.834647
85	-2742.4	-0.039745
86	58961.9	0.854520

87	2742.4	0.039745
88	-60333.1	-0.538688
89	2742.4	0.039745
90	58961.9	0.854520
91	-2742.4	-0.039745
92	-57590.7	-0.834647
93	2742.4	0.039745
94	56219.5	0.814775
95	-2742.4	-0.039745
96	-54848.3	-0.489717
97	2742.4	0.039745
98	53477.0	0.775030
99	-2742.4	-0.039745
100	-52105.8	-0.755157
101	2742.4	0.039745
102	50734.6	0.735285
103	-2742.4	-0.039745
104	-49363.4	-0.440745
105	2742.4	0.039745
106	47992.2	0.695539
107	-2742.4	-0.039745
108	-46621.0	-0.675667
109	2742.4	0.039745
110	45249.8	0.655794
111	-2742.4	-0.039745
112	-43878.6	-0.391773
113	2742.4	0.039745
114	42507.4	0.616049
115	-2742.4	-0.039745
116	-41136.2	-0.596177
117	2742.4	0.039745
118	39765.0	0.576304
119	-2742.4	-0.039745
120	-38393.8	-0.342802
121	2742.4	0.039745
122	37022.6	0.536559
123	-2742.4	-0.039745
124	-35651.4	-0.516686
125	2742.4	0.039745
126	34280.2	0.496814

127	-2742.4	-0.039745
128	-32909.0	-0.293830
129	2742.4	0.039745
130	31537.7	0.457069
131	-2742.4	-0.039745
132	-30166.5	-0.437196
133	2742.4	0.039745
134	28795.3	0.417324
135	-2742.4	-0.039745
136	-27424.1	-0.244858
137	2742.4	0.039745
138	26052.9	0.377579
139	-2742.4	-0.039745
140	-24681.7	-0.357706
141	2742.4	0.039745
142	23310.5	0.337833
143	-2742.4	-0.039745
144	-21939.3	-0.195887
145	2742.4	0.039745
146	20568.1	0.298088
147	-2742.4	-0.039745
148	-19196.9	-0.278216
149	2742.4	0.039745
150	17825.7	0.258343
151	-2742.4	-0.039745
152	-16454.5	-0.146915
153	2742.4	0.039745
154	15083.3	0.218598
155	-2742.4	-0.039745
156	-13712.1	-0.198726
157	2742.4	0.039745
158	12340.9	0.178853
159	-2742.4	-0.039745
160	-10969.7	-0.097943
161	2742.4	0.039745
162	9598.4	0.139108
163	-2742.4	-0.039745
164	-8227.2	-0.119235
165	2742.4	0.039745
166	6856.0	0.099363

167	-2742.4	-0.039745
168	-5484.8	-0.048972
169	2742.4	0.039745
170	4113.6	0.059618
171	-2742.4	-0.039745
172	-2742.4	-0.039745
173	2742.4	0.039745
174	1371.2	0.019873
175	-2742.4	-0.039745

Reactions(N)

1	0.0000e+00
2	2375
3	2375