ASEN 2001 Lab 3: Composite Beam Bending

Ryan Hughes[*], Roland Bailey[†], Will Faulkner[‡], Abdulla Al Ameri[§], Johann Kailey-Steiner[¶]

This lab investigated the structural loading of a foam-balsa composite. Based on loading experiment results, a wing shape was derived to optimize structural integrity under typical aerodynamic forces. A foam-balsa planform was then created following the derived shape and tested under simulated aerodynamic forces. A "Wiffle Tree" test was conducted to achieve these loading conditions. The planform withstood 116.25 N of force and failed in shear at the narrowest section as predicted.

I. Nomenclature

FOS	=	Factor Of Safety
p_0	=	Total pressure
$q(x)$	=	Pressure Distribution
V	=	Shear Force
M	=	Bending Moment
σ_{fail}	=	Failure Stress
M_{fail}	=	Failure Bending Moment
c	=	Chord Length of Plan Form
I_b	=	Moment of Inertia of Balsa Wood
I_f	=	Moment of Inertia of Foam
E_b	=	Young's Modulus of Balsa Wood
E_f	=	Young's Modulus of Foam
τ_{fail}	=	Failure Shear Stress
V_{fail}	=	Failure Shear Force
A_f	=	Cross-Sectional Area of Foam

II. Introduction

Composites are used in numerous applications across the aerospace industry. One example is a foam-balsa composite comprised of a foam core covered by thin sheets of balsa wood. This type of composite along with other similar composite materials is commonly used in the construction of wings for small unmanned aircraft. These wings undergo aerodynamic forces resulting in shear stress and bending moment that must be understood and accounted for in order to develop a safe and effective design. In this lab, data from testing a foam-balsa composite beam consisting of a 3/4" foam core between two 1/32" sheets of balsa wood was provided. This data reflected the strength of the composite and was used to optimize a planform of the composite beam. A width function describing the ideal planform shape was derived using the relationships between the following formulas: pressure distribution described by

$$q(x) = p_0\sqrt{1 - (2x/L)^2}\,w(x) \tag{1}$$

where w(x) is the width function of the planform, along with the relationships

$$\frac{dV}{dx} = q(x) \tag{2}$$

---
[*]107498319
[†]107467065
[‡]103620482
[§]109364560
[¶]108077503

$$\frac{dM}{dx} = V(x) \tag{3}$$

where V is the shear force and M is the bending moment, the Flexure formula

$$\sigma_{fail} = -\frac{M_{fail}c}{(I_b + (E_f/E_b)I_f)} \tag{4}$$

where σ_{fail} is the failure stress of the beam, M_{fail} is the failure bending moment, c is the distance from the neutral axis to the outside edge of balsa. I_b and I_f are the moments of inertia of the balsa and the foam respectively, and E_f and E_b are the Young's Moduli of the foam and balsa respectively and the shear formula

$$\tau_{fail} = \frac{3}{2}\frac{V_{fail}}{A_f} \tag{5}$$

where τ_{fail} is the failure shear stress of the beam, V_{fail} is the failure shear force, and A_f is the cross-sectional area of the foam.

Additional details regarding the analysis of the composite beam and the derivation of the width function are included in the following sections

III. Strength Analysis

The sample being tested was a composite beam of foam and balsa. The first and third layers were 1/32" strips of balsa and the middle layer was 3/4" of foam. This was a 4" by 36" beam. The group used data from similar experiments to determine the strength of the specimen. In this scenario, the bars were loaded with one support on each end, and two straps used to hang weight (Fig. 3). Weight was added until the beam snapped. The force at which the specimen failed was then recorded along with whether the beam snapped in shear or bending. We used the formulas for Flexure and shear based on the moment and shear failures in the data. Figures 1 and 2 show the moment and shear diagrams for the data given. The calculated value for failure from shear was $4.19 * 10^4$ (Pascals) and was $9.70 * 10^6$ (Pascals) for the normal stress. Dividing these by the factor of safety gave us our maximum allowable shear stress and bending moment (τ_{fail} and M_{fail}). From the test data, we were able to determine that the sample was more likely to break under shear. We derived a w(x) function for width. The foam-balsa composite beam was then cut using the w(x) function for testing. Using wiffle tree loading, we were able to simulate aerodynamic forces on the wings.
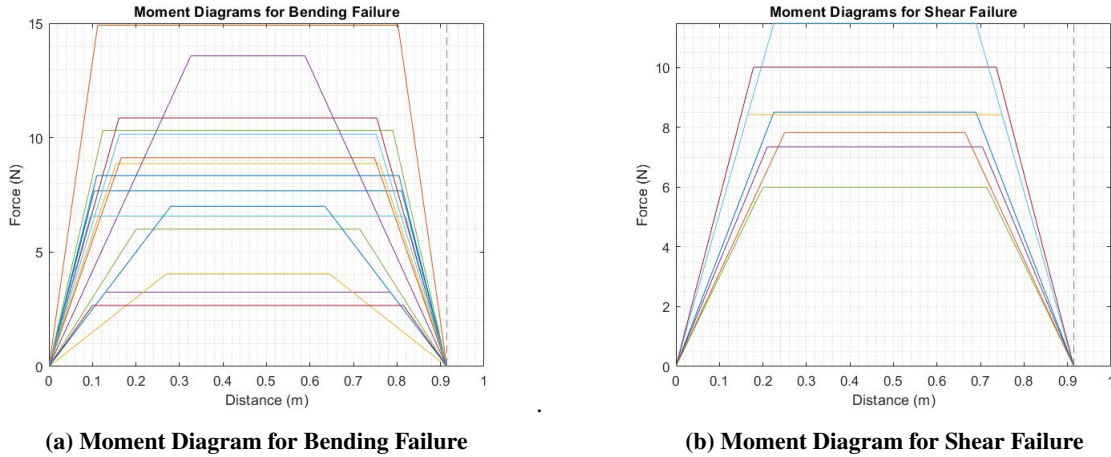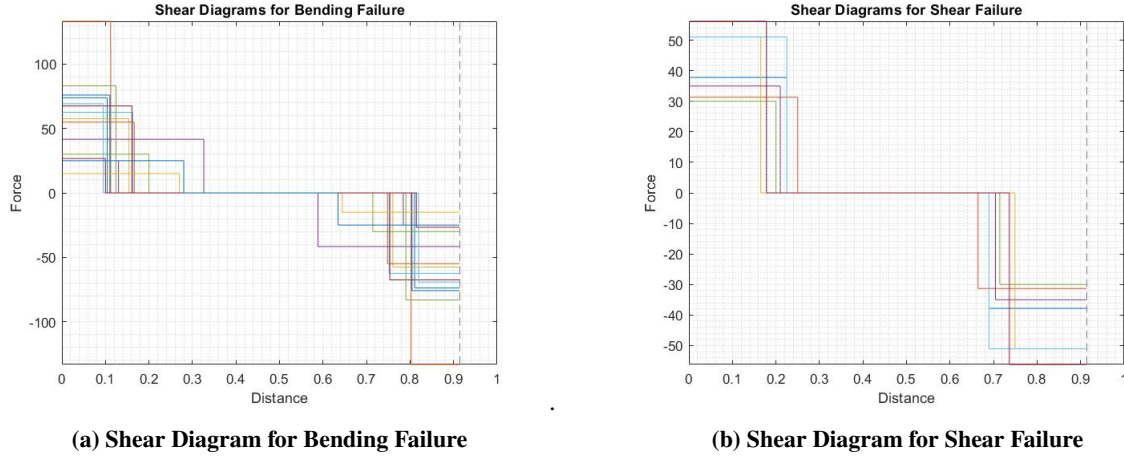

(a) Moment Diagram for Bending Failure   (b) Moment Diagram for Shear Failure

Fig. 1   Moment Diagrams

(a) Shear Diagram for Bending Failure

(b) Shear Diagram for Shear Failure

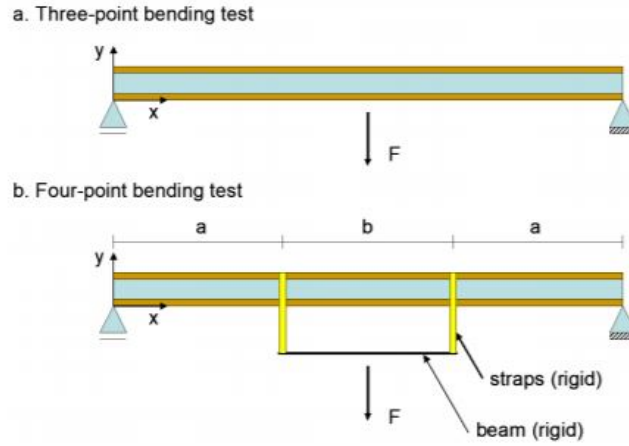**Fig. 2   Shear Diagrams**

Fig. 3   Preliminary Strength Test Setup

IV. Wing Design

Once there was enough information to approximate the composite material strength, a design and load simulation was then considered. Initially, after excluding all outliers, a safety factor was cho a safety factor was chosen between 1.3 to 1.5, which is the typical range for aerospace applications. For this lab, 1.3 was chosen as the safety factor.

FOS	σ_{fail}	τ_{fail}	$\sigma_{allowed}$	$\tau_{allowed}$
1.3	9.70e+06	4.19e+04	7.46e+06	3.22e+04

Table 1   Normal and Shear Stresses and Safety Factor. Allowed values are after considering the safety factor. All values are in Pascals.

Table 1 displays the values for safety factor along with normal and shear stresses before and after considering the safety factor. Once a safety factor was decided upon, values for maximum allowed shear and normal stresses were determined. From there, given that the aerodynamic loading was given by equation (1) and knowing the relationship between the applied load, shear, and moment distribution, given by the relationships in equations (2) and (3), the loading function was integrated using MATLAB built in symbolic integration function to derive the shear diagram. Once the shear diagram was obtained, the same process of integration was repeated to derive the moment diagram.

It is important to note that it was assumed that the width in the load function is 4 inches. This was done just to obtain

the value of the term p_0, and once the value of p_0 was obtained, given the chosen maximum allowed shear and normal stresses, the Flexure formula and shear formula (equations 4 and 5 respectively) were both rearranged to arrive at a width function across the whole length of the beam; one with respect to moment, and one with respect to shear. Afterwards, the maximum absolute values for width with respect to both shear and moment width functions were chosen and a final width function for the plan form was created. The moment and shear diagrams for the beam are shown in Figure 4.
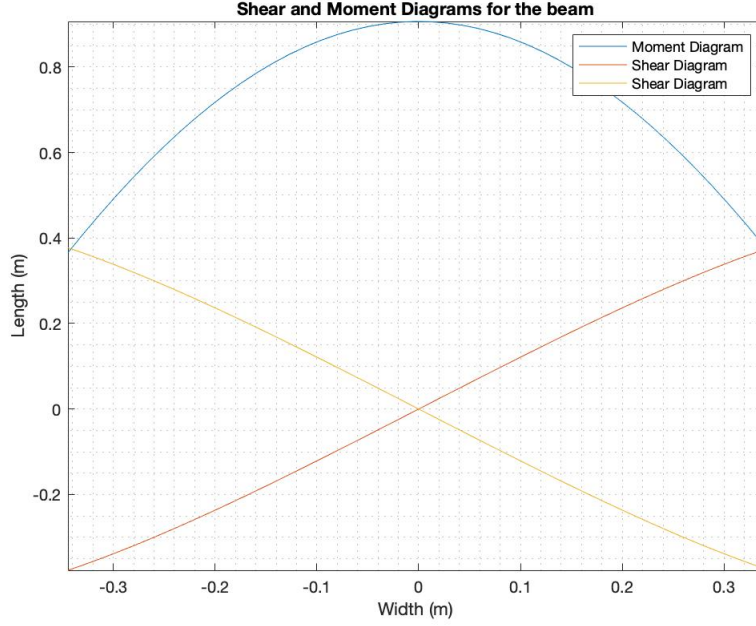

Fig. 4   Shear and Moment Diagrams for the Beam

To simulate typical aerodynamic loading, the "Wiffle Tree" approach was used. Simply, the centroid under each $\frac{1}{8}$th of the aerodynamic load function (equation 1) was determined, and this served as each location of a strap. These centroid locations were determined for half the wing and then symmetry was applied for the whole wing. Once that was known equation 6 was applied to find the location of the next part of the wiffle tree. This method is also shown in figure 5. It was determined that the force at each strap would be equal to $\frac{1}{8}$th of the integral of equation 1 for that section as well as the portion of the wiffle tree test setup that that section would have to withhold (Fig. 5).

$$l_1 = \frac{F_2 l}{F_1 + F_2} \tag{6}$$

Once the locations of all parts of the wiffle tree were found by the method in figure 5, the wiffle tree was constructed on the wing. Figure 8 shows the final setup. From there, the wing was loaded by adding loads in increments of 1.5 pounds to the bucket (Fig. 8) until failure.

V. Discussion

It was hypothesized that the beam is more likely to fail in shear than bending as the shear stress failure values were lower. As predicted, the plan form appeared to fail in shear. Figure 6 shows the break clearly happening in shear, with the balsa failing first.

Furthermore, the wing was able to support a greater force than predicted. The theoretical predictions and actual results are discussed more extensively in the conclusion. It is suspected that this difference is due to the fact that the expected force the plan form was expected to handle came based on initial strength test, which was based on the 4-point configuration test. The data from these tests was analyzed and the minimum bending and shear failure values were used to find the width function. In practice the material used was stronger than the weakest material from the test data as material properties vary significantly.
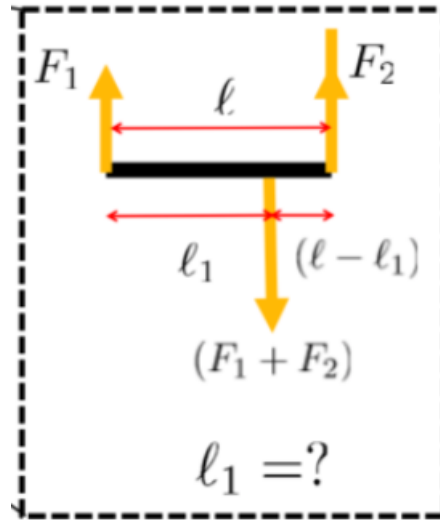
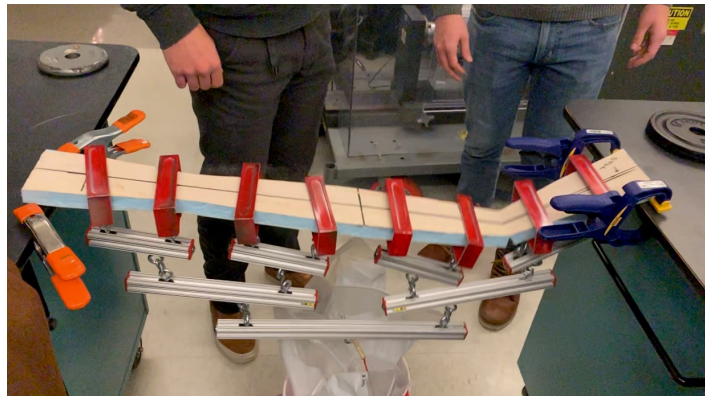**Fig. 5   Finding the location of the straps**

Fig. 6   Planform Fail

Additional failure modes that would require consideration in realistic applications are fatigue, thermal variations, and corrosion. Fatigue failure can be prevented by cyclic testing prior to implementation of a wing design, and thermal variations can be accounted for analytically by finding atmospheric variations due to altitude and re-calculating the aerodynamic forces given these variations. Failure due to corrosion can largely be written off for the wings themselves, as they are mostly made of non-corrosive composites, but bolts and screws on the wing are still susceptible to this mode of failure. This happened in 1979, when an engine pylon broke due to lack of proper maintenance and caused the deadliest aviation accident to occur in the US

Fig. 7   Shear of Planform

Fig. 8   Wiffle Loading Setup

VI. Conclusions

The results of the experiment and the lab followed what was expected. Based off of the given test data the wing plan form was predicted to hold 6.90 kilograms, but in practice the wing held 11.85 Kg. The wing was designed with a FOS of 1.3 in mind which means that the max force that the beam would experience in normal use would be 9.12 Kg. This follows the reasoning presented in the discussion on the strength of the material. The fact that the wing held more weight than was predicted also strongly suggests that the wing design followed proper practices and the equations used to derive the wing width were indeed applicable. The wing ended up failing at the right side thinnest point, when viewing as in Figure 6. This may be due to the fact that the stress was concentrated at the sharp corner as shown in the width function which would have meant the material at that point could have been experiencing more stress than it should have. Another reason that the thinnest point failed first could also be due to manufacturing errors where the area was thinner than it was intended to be.

References

[1] Statics and Structures, ASEN 2001, "Composite Beam Bending ASEN 2001 Lab 3, Session 3, Fall 2018," Lab 3 Session 2 2018.pptx.

[2] Statics and Structures, ASEN 2001, "Composite Beam Bending ASEN 2001 Lab 3, Session 3, Fall 2018," Lab 3 Session 3 2018.pptx.

[3] Hibbeler, R. C., and Yap, K. B., Statics and mechanics of materials, Harlow, United Kingdom: Pearson Education, Inc., 2017.

[4] Statics and Structures, ASEN 2001, "ASEN 2001 Lab 3 Fall 2018.pdf," ASEN 2001: Statics and Structures, Fall 18.

Appendix A: MATLAB Code

```matlab
1  % info:
2
3  %{
4
5  This codes is part of CU's ASEN 2001: Statics and Structures
6
7   This lab is concerned with the analysis and design of composite beams.
8  Beams are frequently used componentsin aerospace structures, such as wings,
9  booms, etc. To reduce their weight and to tailor their stiffness and strength
10 toparticular loading conditions, beams are often made of several materials.
11 Such beams are called composite beams. Incomparison with truss structures,
12 the structural response of beams can be rather complex and various forms of
13 failureneed to be carefully considered in the design process.
14
15 This code will take Test data for the the strength of the beam and get the
16 faliure normal and shear stresses and then plot optimal width function for
17 that beam, for more info read The files in AssignmentInfo .
18
19 Done by:
20
21 - Ryan Hughes
22 - Roland Bailey
23 - Will Faulkner
24 - Johann Kailey-Steiner
25 - Abdulla Al Ameri
26
27
28
29 %}
30
31 %% Housekeeping
32
33 clear;
34 clc;
35 close all;
36
37 %% read and extract file
38
39 E_Foam = 0.035483*10^9 ; %Pa
40 E_Bals = 3.2953*10^9 ; %Pa
41
42 % Wiffle Tree Weights:
43
44 Sleeve = 0.49 ; %N sleeve
45 six_in_bar = 1.77 ; %N 6 inch bar
46 twelve_in_bar = 2.94 ; %N 12 inch bar
47 eighteen_in_bar = 3.92 ; %N 18 inch bar
48
49 LengthCS = 0.0206375; %in meter.
50 %% read test data:
51
52 % read file
53 [ TestData Comments ] = xlsread('TestData.xlsx');
54
55 %get the comments as strings to see how it failed.
56 FailType = cellstr(Comments(2:length(Comments),6));
57
58 % Check if you failing bending or shear
59
60 % the contains is case-sensitive;
61 BendFail = contains(FailType,{'Bending','bending'});
62 ShearFail = contains(FailType,{'Shear','shear','perpendicular','both'});
63
64
65 %% excludeds :
```

```matlab
66
67  % test 7 is excluded, no breakage
68  % test 11 : no comment
69
70  %%
71  % get the exact data related to bend and ehar
72
73  BendData = TestData(BendFail,:);
74  ShearData = TestData(ShearFail,:);
75
76  %% add data manually:
77
78  %{
79  if some if borken on both shear and moment or might be shear but is
80  reported as moment, it should be added manually.
81
82  %}
83
84
85  %% get the length of the middle bar b:
86
87  Barlength = 36*0.0254 ; % in m
88  bBend = (Barlength) - 2*BendData(:,3); % in meter
89  bShear = (Barlength) - 2*ShearData(:,3); % in meter
90
91  FoamLength = 3/4 * 0.0254 ; % in m
92  BalsaLength = 1/32 * 0.0254 ; % in m
93  WidthCS = 4 * 0.0254; %in m CS = Cross-section
94  WidthCSBendData = BendData(:,4);
95  WidthCSShearData = ShearData(:,4);
96
97
98
99
100
101 %% Shear/moment daiagram : TEST DATA
102
103
104 % the purpose of this is just to plot.
105
106
107 % do the shear/moment daiagrams for shear fail
108
109 ShearDiagram_ShearFail = {};
110
111 %place holders
112
113 ShearStress_ShearFail = zeros(1,length(bShear));
114 ShearStress_BendFail = zeros(1,length(bBend));
115 RootFunction_ShearFail = {};
116
117 %place holders for maximas
118 Max_Shear_ShearFail = zeros(1,length(bShear));
119 Max_Shear_BendFail = zeros(1,length(bBend));
120 Max_Moment_ShearFail = zeros(1,length(bShear));
121 Max_Moment_BendFail = zeros(1,length(bBend));
122
123
124 for i =1:length(bShear)
125 syms x
126 ShearDiagram_ShearFail{i} = piecewise ( 0<x<ShearData(i,3) , ShearData(i,2)/2 , ...
        ShearData(i,3)<x<bShear(i)+ShearData(i,3) , 0 , bShear(i)+ShearData(i,3)<x<Barlength , ...
        -ShearData(i,2)/2 );
127 Max_Shear_ShearFail(i) = ...
        double(max(abs(subs(cell2sym(ShearDiagram_ShearFail(i)),[0:0.01:Barlength]))));
128 MomentDiagram_ShearFail(i) = piecewise ( 0<x<ShearData(i,3) , ShearData(i,2)/2 * x, ...
        ShearData(i,3)<x<bShear(i)+ShearData(i,3) , ShearData(i,3) * ShearData(i,2)/2 , ...
        bShear(i)+ShearData(i,3)<x<Barlength , (-ShearData(i,2)/2 * (x-Barlength)));
```

9

```matlab
129  %ShearStress_ShearFail(i) = ((3/2) * (Max_Shear_ShearFail(i)) )/ (WidthCS*FoamLength) ;
130
131
132  end
133
134  % bending fail
135
136  ShearDiagram_BendFail = {};
137
138  for i =1:length(bBend)
139  syms x
140  ShearDiagram_BendFail{i} = piecewise ( 0<x<BendData(i,3) , BendData(i,2)/2 , ...
         BendData(i,3)<x<bBend(i)+ BendData(i,3) , 0 , bBend(i)+BendData(i,3)<x<Barlength , ...
         -BendData(i,2)/2 );
141  Max_Shear_BendFail(i) = ...
         double(max(abs(subs(cell2sym(ShearDiagram_BendFail(i)),[0:0.01:Barlength])))) ;
142  MomentDiagram_BendFail(i) = piecewise ( 0<x<BendData(i,3) , BendData(i,2)/2 * x, ...
         BendData(i,3)<x<bBend(i)+ BendData(i,3) ,  BendData(i,3) * BendData(i,2)/2 , ...
         bBend(i)+BendData(i,3)<x<Barlength , (-BendData(i,2)/2 * (x-Barlength)));
143  %ShearStress_BendFail(i) = ((3/2) * (Max_Shear_BendFail(i)) )/ (WidthCS*FoamLength) ;
144
145  end
146
147  %moment diagrams =
148
149  %not needed:
150  %{
151  %loop to get max
152  % remove ABS if you do not need the absloute value of max-min!
153  for i =1:length(bShear)
154
155  Max_Moment_ShearFail(i) = ...
         double(max(abs(subs(MomentDiagram_ShearFail(i),[0:0.01:Barlength])))) ;
156
157  end
158
159  for i =1:length(bBend)
160
161  Max_Moment_BendFail(i) = ...
         double(max(abs(subs(MomentDiagram_BendFail(i),[0:0.01:Barlength])))) ;
162
163  end
164  %}
165  %% PLOT HERE
166  figure(1)
167  fplot(MomentDiagram_BendFail, [0, 1])
168  title('Moment Diagrams for Bending Failure')
169  xlabel('Distance (m)')
170  ylabel('Force (N)')
171  grid minor
172
173  figure(2)
174  fplot(ShearDiagram_BendFail, [0, 1])
175  title('Shear Diagrams for Bending Failure')
176  xlabel('Distance (m)')
177  ylabel('Force (N)')
178  grid minor
179
180  figure(3)
181  fplot(MomentDiagram_ShearFail, [0, 1])
182  title('Moment Diagrams for Shear Failure')
183  xlabel('Distance (m)')
184  ylabel('Force (N)')
185  grid minor
186
187  figure(4)
188  fplot(ShearDiagram_ShearFail, [0, 1])
189  title('Shear Diagrams for Shear Failure')
```

```matlab
190  xlabel('Distance (m)')
191  ylabel('Force (N)')
192  grid minor
193
194
195
196  %% Find Maximum moment: the moment at the location of faliure
197
198  for i =1:length(bBend)
199
200  Max_Moment_BendFail(i) = double(abs(subs(MomentDiagram_BendFail(i),x,BendData(i,5)))) ;
201
202  end
203
204  for i =1:length(bShear)
205
206  Max_Moment_ShearFail(i) = double(abs(subs(MomentDiagram_ShearFail(i),ShearData(i,5)))) ;
207
208  end
209
210
211
212  %}
213
214  %Largest moment
215  [ r c ] = size(BendData) ;
216  for i=1:r
217
218      if 0 ≤ BendData(i,5) < BendData(i,3)
219
220          Mfail(i) = (BendData(i,2)/2)*BendData(i,5);
221
222      elseif BendData(i,3) ≤ BendData(i,5) && BendData(i,5) < BendData(i,3)+bBend(i)
223
224          Mfail(i) = (BendData(i,2)/2)*BendData(i,3);
225
226      else
227
228          Mfail(i) = (BendData(i,2)/2)*BendData(i,5);
229
230      end
231
232
233  end
234
235  %Largest shear
236  [ r c ] = size(ShearData) ;
237  for i=1:r
238
239      if 0 ≤ ShearData(i,5) < ShearData(i,3)
240
241          Vfail(i) = ShearData(i,2)/2;
242
243      elseif ShearData(i,3) ≤ ShearData(i,5) && ShearData(i,5) < ShearData(i,3)+ShearData(i)
244
245          Vfail(i) = ShearData(i,2)/2; % It should be 0 but they didn't account
246          %for the strap in the test, so we will just use the value before it.
247          %Because there's no way it'll break @ 0 shear.
248
249      else
250
251          Vfail(i) = -ShearData(i,2)/2;
252
253      end
254
255
256  end
257
```

```matlab
258
259  %% Moment of Inertia: from centroidal axis.
260
261  % the neutral axis is the z axis, thus moment about z axis for foam will be
262  % just at the axis itself,
263
264  CentroidShape = ((FoamLength+(2*BalsaLength))/2) ;
265  CnetroidTopBalsa = ( (FoamLength + BalsaLength) + BalsaLength/2 );
266
267  IFoam_Bend = (1/12).*( WidthCSBendData ).*(FoamLength)^3 ;
268  IBalsa_Bend = 2*((1/12).*(WidthCSBendData).*(BalsaLength)^3 + ( CentroidShape - ...
          CnetroidTopBalsa ) ^2 ).*(WidthCSBendData.*BalsaLength) ;
269
270  IFoam_Shear = (1/12).*( WidthCSShearData ).*(FoamLength)^3 ;
271  IBalsa_Shear = 2*((1/12).*(WidthCSShearData).*(BalsaLength)^3 + ( CentroidShape - ...
          CnetroidTopBalsa ) ^2 ).*(WidthCSShearData.*BalsaLength) ;
272
273  %% safety factor
274
275  FOS = 1.3;
276
277  %% Max Normal Stress: Flexural Formula
278
279  MaxNormalStress_BendFail = -(Mfail' .* (BalsaLength+(FoamLength/2)))./ ( IBalsa_Bend + ...
          (E_Foam/E_Bals).*IFoam_Bend) ;
280  %remove outlayers
281
282  MaxNormalStress_BendFail(5) = [];
283
284  % they're both the same, it's sample 10;
285
286  %max alloweable normal stress
287  MaxAllowNormal = mean(MaxNormalStress_BendFail)./ FOS ;
288
289  %% Max Shear Stress: Shear Formula
290
291  % the ShearStress_BendFail takes the whole area instead of half, double
292  % check.
293
294  %we checked, it's the full area.
295
296  for i=1:length(Vfail)
297  MaxShearStress_ShearFail = (3/2) * (abs(Vfail)'./((FoamLength).*ShearData(i,4))) ;
298  end
299
300  %remove outlayers
301
302  %get picket values
303
304  %all of them are 0 here but this needs to be checked later.
305
306  PickedAllow = min(MaxShearStress_ShearFail);
307
308  %{
309  % determine which width of the beam you have picked:
310
311  if isnumeric(find(MaxShearStress_ShearFail == PickedAllow))==1
312
313      PickedWidth = find(MaxShearStress_ShearFail == PickedAllow);
314
315  else
316
317      PickedWidth = find(MaxShearStress_BendFail == PickedAllow);
318
319  end
320
321  %}
322
```

```matlab
323  MaxAllowShear = ( PickedAllow /1.3);
324
325  %%
326
327
328  %% get p(0), V, and M : DESIGN
329
330  syms p0 x
331  qx = (4*0.0254)*p0 * sqrt ( 1 - ((2*x)/Barlength)^2) ;
332
333  %Get F:
334  F = int(qx,x,(-Barlength/2),(Barlength/2));
335  %get V
336
337  Vx = -F/2 + int(qx,x,(-Barlength/2),x);
338
339  Mx = int(Vx,x,(-Barlength/2),x);
340
341  %solve for M as function of p0, then solve for p0;
342
343
344  % here If and Ib will use the width of the cross section, 4 inches or
345  % what's stored as WidthCS;
346
347  If = (1/12).*( WidthCS ).*(FoamLength)^3 ;
348
349  %we had to mulyiply by 2 to make it work, check why:
350  Ib = 2*((1/12).*(WidthCS).*(BalsaLength)^3 + ( CentroidShape - 2*CnetroidTopBalsa ) ^2 ...
         ).*(WidthCS.*BalsaLength) ;
351
352
353
354  MP0 = subs(Mx,x,0); %sub 0 for x in Mx;
355
356  Equation = MaxAllowNormal == (MP0 .* (BalsaLength+(FoamLength/2)))/ ( Ib + ...
         (E_Foam/E_Bals)* If ) ;
357  p0Value = solve(Equation,p0);
358
359  %SUB back P0 in the moment function
360
361  Mx_WithP0 = subs(Mx,p0,p0Value);
362  Vx_WithP0 = subs(Vx,p0,p0Value);
363  % C = distance from neutral axis
364  Cvalue = (BalsaLength+(FoamLength/2));
365
366
367  %moment of inertia of foam and balasa for bending data, width now isn't
368  %constant width is function will be solved for o
369
370  If = (1/12)*(FoamLength)^3 ;
371  Ib = 2*((1/12)*(BalsaLength)^3 + ( CentroidShape - CnetroidTopBalsa ) ^2 *BalsaLength);
372
373
374  Width_Moment_Function = Mx_WithP0 * (BalsaLength+(FoamLength/2))/((Ib + ...
         (E_Foam/E_Bals).*If)*MaxAllowNormal);
375  Width_Shear_Function = (((MaxAllowShear*(3/2))/(Vx_WithP0))^-1)/(FoamLength);
376
377  %loop over the two plots to get width function
378  j = 1; %index of storing
379  for i=(-Barlength/2):0.01:(Barlength/2)
380      %width is maximum of both graphs of shear and moment
381      WidthFunction(j) = max(abs(subs(Width_Moment_Function,i)), ...
             abs(subs(Width_Shear_Function,i)));
382      iValues(j) = i; %store x values
383      j = j+1;
384  end
385
386  WidthFunction = WidthFunction/2;
```

```matlab
387  %Just to half the width because it gives us width for full span of 4 inches
388  % and we're doing half top and half bottom.
389
390  ForCut = double([ iValues' WidthFunction' ]);
391  % this just for us to cut the wing in practice.
392
393  ForCut = ForCut * 100 ; %convert
394  %% plotting
395
396  figure(5)
397  fplot(Width_Moment_Function)
398  hold on
399  fplot(Width_Shear_Function)
400  hold on
401  fplot(-Width_Shear_Function)
402  legend('Moment Diagram', 'Shear Diagram','Shear Diagram');
403  title('Shear and Moment Diagrams for the beam')
404  xlabel('Width (m)')
405  ylabel('Length (m)')
406  grid minor
407  hold off
408
409
410  figure(6);
411  plot(iValues,WidthFunction)
412  hold on
413  plot(iValues,-WidthFunction)
414  title('Width function');
415  xlabel('Length')
416  ylabel('Width')
417  hold on
418  grid minor
419
420  %% Determine the testing
421
422
423  % Loading function
424
425  syms x
426  Px = p0Value  * sqrt(1-(2*x/Barlength)^2);
427  ItegralBounds = 0:(Barlength/8):(Barlength/2);
428  % Integrate the weighted x values as well as the function
429  for i = 1:length(ItegralBounds)-1
430      wint(i)  = int(Px*x,x,ItegralBounds(i),ItegralBounds(i+1));
431      nwint(i) = int(Px,x,ItegralBounds(i),ItegralBounds(i+1));
432      centroid(i) = wint(i)/nwint(i);
433  end
434
435  %location of straps
436  l1 = centroid(2)-centroid(1);
437  l2 = ...
          ((nwint(1)+Sleeve+(.5*six_in_bar))*l1)/((nwint(1)+Sleeve+(.5*six_in_bar))+(nwint(2)+Sleeve+(.5*six_in_bar
438  l3 = centroid(4)-centroid(3);
439  l4 = ...
          ((nwint(3)+Sleeve+(.5*six_in_bar))*l3)/((nwint(3)+Sleeve+(.5*six_in_bar))+(nwint(4)+Sleeve+(.5*six_in_bar
440
441  l5 = (centroid(4)-l4)-(centroid(2)-l2);
442
443  l6 = ...
          (((nwint(1)+Sleeve+(.5*six_in_bar))+(nwint(2)+Sleeve+(.5*six_in_bar))+(.5*twelve_in_bar))*l5)/((nwint(1)+
444
445  %% Theoretical:
446
447  % how much force we excpect the wing to handle.
448
449  MaxNormalForce = MaxAllowNormal * LengthCS * WidthCS;
450
451  MaxShearForce = MaxAllowShear * LengthCS * WidthCS;
```