

ASEN 2001 - Lab 3 - Composite Beam Bending

Michael Burns* Jashan Chopra†, Aidan Kirby‡, Emma Markovich§, Michael Martinson¶,
ASEN 2001 Section 001

This lab focused on constructing a composite beam to represent a wing and testing it under simulated distributed aerodynamic loading with a wiffle tree system. The first step was performing strength analysis of the beam using moment and shear diagrams developed from bending test data on sample beams. Once these moment and shear diagrams were analyzed, the team used their findings to develop two functions for the width at any point on the beam: one for the width required to withstand the point's bending moment and the other to withstand the point's shear force. By taking the maximum width achieved between the two functions at every point on the beam, an appropriate width function for the entire beam was plotted. This width function was then used to cut the final wing shape. Next, using basic equilibrium equations, the locations of wiffle tree straps were determined to simulate an aerodynamic load. Upon completion of wiffle tree assembly, the wing was tested with increasing loads until it failed in bending moment under an total applied load of 36.7 lbf.

I. Nomenclature

F = Total load on the beam	M = Bending moment	V = Shear force
σ = Normal stress	τ = Shear stress	I_f = Moment of inertia foam
I_b = Moment of inertia balsa wood	E_f = Young's modulus of foam	E_b = Young's modulus of balsa wood
t_b = Thickness of balsa wood	t_f = Thickness of foam	L = Length of the bar
c = half of height of cross section	FOS = Factor of Safety	

II. Introduction

A wide range of beams used in aerospace engineering applications are composites, which means they are comprised of a variety of materials that better suit their specific purposes. Given the diversity of composite beams' applications and their complex structural responses to loading, engineers must analyze the failure conditions for these beams in detail. This lab was concerned with the design of a beam simulating the wing of a small model airplane. These simulated wings have a foam core with thin balsa wood layers on the top and bottom, bonded together by Gorilla Glue. The purpose of the lab was to analyze and interpret internal shear and bending moment test data from a set of composite beams, and apply this data to optimize a composite beam of the same materials. This beam's width was calculated to provide maximum resistance to loading with minimum weight. The lab required the constructed beam to be considered with computational failure analysis and then physically tested under distributed load applied by a wiffle tree system. Notably, the lab required finding a wiffle tree configuration for simulating a distributed aerodynamic load which was increased until the designed beam failed.

Background Theory

The fundamental physical concepts behind this lab are those describing the impact distributed loading has on a beam. Wing designers must consider distributed loading, normally in the form of load per unit area, as the pressure distribution acting on a wing in flight. A wing under distributed load faces two forms of stress which can cause failure: normal stress due to internal bending moment and shear stress.

*107578059

†107689146

‡105955098

§105086160

¶107190048

Assuming that a beam under a distributed load exhibits linear elastic behavior and is in static equilibrium, the following “flexure formula” describes the maximum internal normal bending stress encountered in the beam at a given cross section:

$$\sigma_{max} = \frac{Mc}{I} \quad (1)$$

Provided that the information in Eq. (1) is available, it offers a method for predicting what amount of normal stress due to loading will cause a design to fail. Importantly, beam failure due to normal bending stress caused by distributed loading occurs when the stress distribution modeled with Eq. (1) causes half of beam’s material either above or below the neutral axis to be torn apart in tension while the other half is crushed in compression.

A beam under distributed loading experiences internal shear stress in the form of transverse shear stress over its cross section and longitudinal shear stress along its length due to the complementary property of shear. The magnitude of shear stress at a point on a cross section of a beam is given with the “shear formula”:

$$\tau = \frac{VQ}{It} \quad (2)$$

Shear failure is recognizable by internal cracks forming and extending through the beam at angles not perpendicular to the beam’s longitudinal axis. These two equations allow the optimum width of the beam to be calculated.

III. Strength Analysis

Specimen Construction

As established in this report’s introduction, the behavior of composite materials under loading tends to be complex. Thus, before designing its optimized composite beam, the group needed to establish reliable data for the strength of this material against shear and normal bending internal stresses. Specifically, the group required estimates for τ_{fail} and σ_{fail} , respectively the shear stress and normal bending stress such a beam could withstand before failure.

A rectangular beam of the composite material applied in this lab with variable width w has the following cross section (note that Fig. 1 features thickness values in inches):

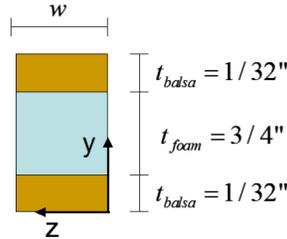


Fig. 1 Composite beam cross section

Furthermore, it is known that the modulus of elasticity of the foam used, E_f , is 0.035483 GPa while the modulus of elasticity of the balsa used, E_b , is 3.2953 GPa. Applying Eq. (1) to fit the composite beam situation yields the following equation for the bending normal stress of the beam if M_{fail} , the internal bending moment of the beam at failure, is known:

$$\sigma_{fail} = \frac{-M_{fail}c}{(I_b + (E_f/E_b)I_f)} \quad (3)$$

Notably, this equation requires I_b and I_f , respectively the moments of inertia of the cross section’s balsa and foam components about the cross section’s neutral axis. Applying the parallel axis theorem and the formula for the moment of inertia of a rectangle allows for calculating these two values with the following equations:

$$I_b = 2 \left(\left(\frac{1}{12} \right) (w)(t_b^3) + (w)(t_b) \left(\frac{t_b}{2} + \frac{t_f}{2} \right)^2 \right) \quad (4)$$

$$I_f = \left(\frac{1}{12}\right)(w)(t_f^3) \quad (5)$$

Likewise, the shear stress of the beam at failure, τ_{fail} , can be calculated for a lab beam with the following adaptation of Eq. (2):

$$\tau_{fail} = \left(\frac{3}{2}\right)\left(\frac{V_{fail}}{A_f}\right) \quad (6)$$

This equation requires knowing V_{fail} , the internal shear force acting on the beam at failure. A_f , the area of the foam section, is calculated by $A_f = (w)(t_f)$.

In order to obtain the values of the necessary variables to use Eq. (3) and Eq. (6), the data from the experiments conducted before the lab was made available. During each of the 24 experiments, a 36 in. long and 2 in. wide (from a top down perspective) beam with the cross section shown in Fig. 1 ($w = 2$ in. in this case) were loaded in the four-point bending test diagrammed below:

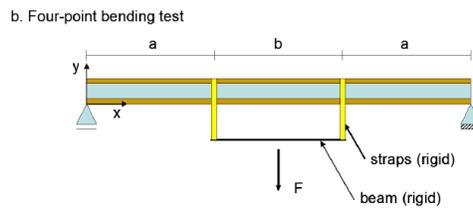


Fig. 2 Four point beam bending test

The supports shown in Fig. 2 were steel rollers; a is the distance from the left most support reaction to the left most applied force and the distance from the right most support reaction to the right most applied force. b is simply the length of the beam minus $2a$. During the tests the force F was increased until the beams failed. This was done by connecting a bucket to the center of the wiffle tree test component marked “beam (rigid)” in Fig. 2 and incrementally adding weights to it. For each test the values of F at failure, the width of the beam w , a , and the distance from the failure location of the beam to the closest support reaction were gathered.

Test Data Analysis

To determine reasonable values for τ_{fail} and σ_{fail} for the composite material, the group processed the given experimental data with following method of creating and analyzing shear and moment diagrams for each of the pre-lab bending tests. First, it is important to note that the four-point bending test significantly simplifies the shear and moment diagrams that would be produced in the distributed loading case. Instead of continuous functions, the shear and moment diagrams appear as piecewise functions. The shear stress was found using the equilibrium equation for the y -direction (vertical) across each section of the beam. Internal bending moment was found by numerically integrating the shear diagram results. This process resulted in the following set of diagrams:

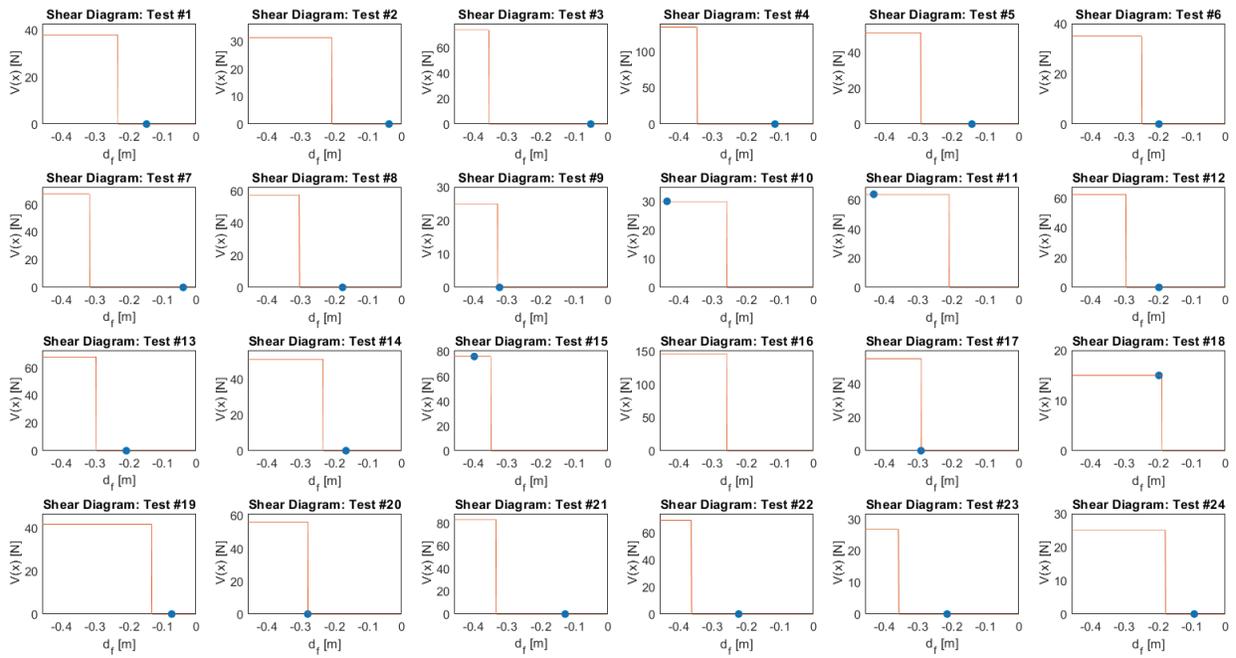


Fig. 3 Shear diagrams for each of the 24 pre-lab bending tests; the beams' points of failure are marked with blue dots

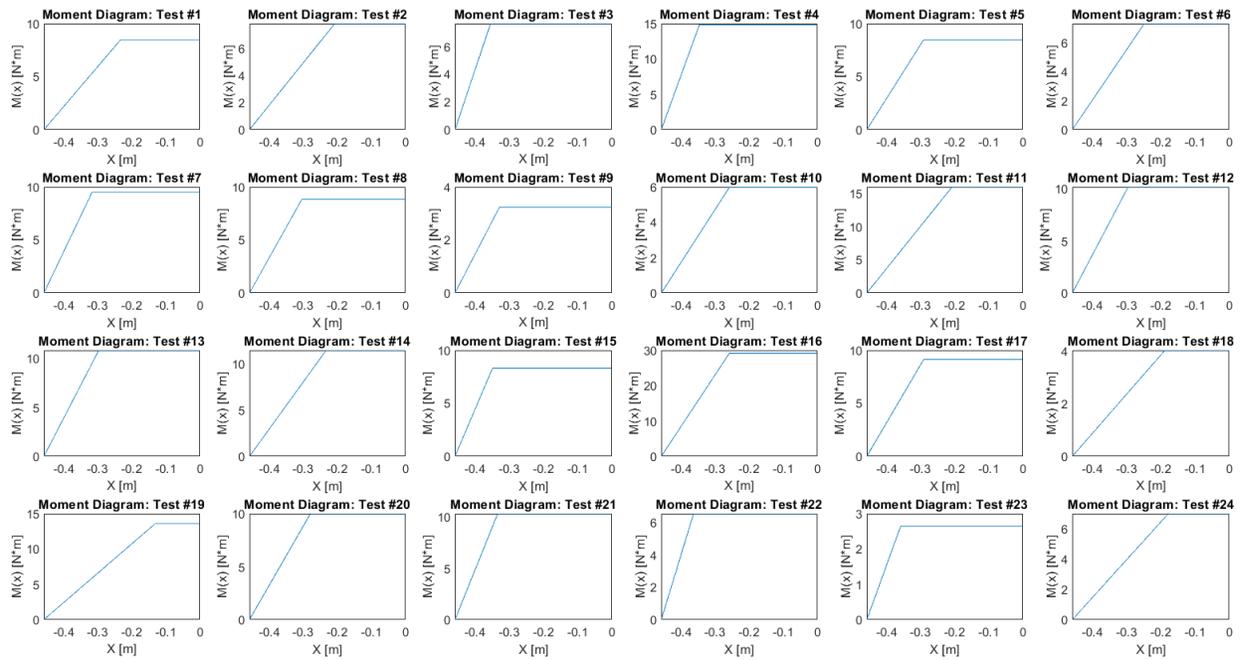


Fig. 4 Moment diagrams for each of the 24 pre-lab bending tests

With these figures created, simple inspection combined with the provided data on the beams' locations of failure was used to determine the mode of failure for each individual test. Prior to performing this analysis, erroneous data was identified and eliminated. Specifically, test 16 was recorded as breaking at a negative distance, therefore the test was ignored in further calculations. Although the group failed to recognize it at the time and still included it in their calculations, test 2 is also technically erroneous given that its width was roughly half of the widths of the other 23 beams.

To determine the mode of failure in each test the group considered the shear diagrams for the tests augmented with markers in each diagram for when failure was recorded to have occurred. These markers appear as the blue dots in Fig. 3. In cases where failure occurred when the beam was still experiencing shear force, the group recorded the test as a shear failure. All remaining tests were consequentially recorded as having failed due to bending.

The shear force and bending moment values at failure for the beams were then extracted and in each case applied with their corresponding test's data to either equation 3 or equation 6 depending on their type of failure. This created a set of τ_{fail} and σ_{fail} values. By averaging these values and then plugging the average τ_{fail} and average σ_{fail} into to (Eqn. 6) and (Eqn. 3) respectively, the values of τ_{max} and σ_{max} were approximated to be 10.0 MPa and 68.2 KPa for σ_{max} and τ_{max} , respectively.

IV. Wing Design

Load and Stress Modeling

The wing design was determined by applying the σ_{max} and τ_{max} values determined from the pre-lab testing data. First, the determined maximum stress values were divided by a factor of safety of 1.5 in order to account for potential discrepancies between a theoretical model and a physically imperfect beam. This process established a σ_{allow} value of 6.68 MPa and a τ_{allow} value of 45.5 KPa for the design.

The pressure applied to the wing per unit length was theoretically modeled with the following equation, where L is the wing's overall length, x expresses the distance along the wing at a given point and ranges from $-L/2$ at the left end of the wing to $L/2$ at the right end of the wing, P_0 is the pressure per unit area encountered at the center of the wing ($x=0$), and $w(x)$ expresses the width of the wing as a function of the x value:

$$q(x) = w(x)P_0\sqrt{1 - (2x/L)^2} \quad (7)$$

Modeling the wing width required to withstand the expected loading while maintaining a minimum weight required solving for P_0 . This was accomplished by considering Eqn. (7) with $w(x) = 4$ in. held constant. With pressure expressed as a function per unit length (in the $q(x)$ function) and leaving P_0 as a constant, a one dimensional integral was applied to solve for the total force on the planform due to the external distributed load expressed in Eqn. 7.

$$F = \int_{-(L/2)}^{(L/2)} 4P_0\sqrt{1 - (2x/L)^2} dx \quad (8)$$

Understanding that this total force F had to be evenly countered by the beam's two supports, and thus requiring a support reaction at each support of magnitude $F/2$ directed upwards, it was then possible to express the shear force acting within the beam, V , as a function of the same x value introduced earlier (via consideration of a free body diagram of the beam):

$$V(x) = (F/2) - \int_{-(L/2)}^{(x)} 4P_0\sqrt{1 - (2x/L)^2} dx \quad (9)$$

Next, understanding the relationship between the shear force and the moment acting in a beam enabled the development of a function for the bending moment in the beam as a function of x and $V(x)$:

$$M(x) = \int_{-(L/2)}^{(x)} V(x) dx \quad (10)$$

This final equation for $M(x)$ could then be evaluated at $x = 0$ to determine the moment acting at the center of the beam. As the bending moment at the center would be the maximum moment encountered in the beam, and thus where normal stress is expected to reach σ_{allow} , the following modification of the flexure formula can be used to solve for P_0 , which was determined to be 1.1129×10^3 Pa :

$$P_0 = \frac{\sigma_{allow} (I_b + \frac{E_f}{E_b} I_f)}{M(x = 0) c} \quad (11)$$

Moment and Shear Diagrams

Since equations for the shear force and internal moment had already been determined and P_0 has a numerical value, substituting a vector of x -values representing the length of the beam into (Eqn. 9) and (Eqn. 10) provided profiles of the shear force and bending moment acting along the length of the plan form. This is seen in the two figures below:

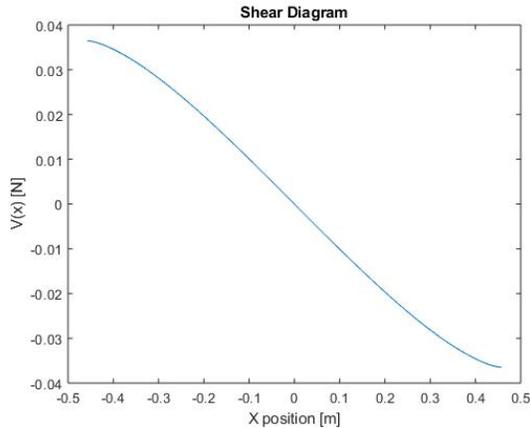


Fig. 5 Shear diagram for planform under a distributed load modeled by (Eqn. 7)

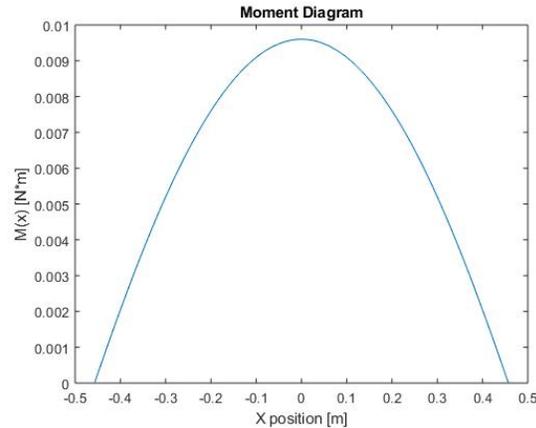


Fig. 6 Moment diagram for planform under a distributed load modeled by (Eqn. 7)

By inspection, one can tell that the magnitudes of shear stress in the planform are at their maximum near the end of the beam. If the planform breaks near either end, then the most likely method of failure is shear. The opposite is also

found to be true, closer to the center the internal moment reaches its maximum. If the planform breaks near the center, the most likely method of failure is due to an internal bending moment. Of course, these are speculation that would vary in real world testing due to internal inconsistencies in the material.

Solving Optimal Widths

In order to ensure that both the allowable normal and shear stresses were accounted for while still minimizing the beam's weight, optimal width functions were determined for both stresses. With P_0 , the optimal width was calculated and plotted both for moment (Eqn. 13) and shear (Eqn. 14).

A. Equations for Calculating Optimal Width

1. Normal Stress

A derivation of the flexure formula (Eqn. 1) is used to find the optimal width for normal stress/bending moment. Eqn. 13 combines the equations for moment of inertia (Equations 4 and 5) in terms of the width, $w(x)$. Ultimately, $w(x)$ optimized for normal stress is solved for from Equation 13, using the MATLAB solver *solve.m*.

$$-M(x) c = \sigma_{allow} \left(I_b + \frac{E_f}{E_b} I_f \right) \quad (12)$$

$$-M(x) c = \sigma_{allow} \left(2 \left(\frac{1}{12} w(x) t_b^3 + w(x) t_b \left(\frac{t_b}{2} + \frac{t_f}{2} \right) \right) + \frac{E_f}{E_b} \left(\frac{1}{12} w(x) t_f^3 \right) \right) \quad (13)$$

2. Shear Stress

A derivation of the shear formula (Equation 2) is used to solve for $w(x)$ optimized for shear stress (Equation 14).

$$w(x) = \frac{3}{2} \frac{V(x)}{\tau_{allow} t_f} \quad (14)$$

B. Width Diagram

The two optimized models for width were then overlayed to determine the required width, the maximum possible width produced by either of the two functions at a given value of x , to satisfy the allowable shear and bending stress requirements (Figure 7). Any width smaller than this maximum would cause either normal or shear stresses to exceed their allowable values. Since the calculated maximum widths are totals for the entire wingspan, the width values were divided by two and recorded at 1 in. increments along the 36 in. wing in order to draw a symmetric wing width shape with respect to the longitudinal centerline.

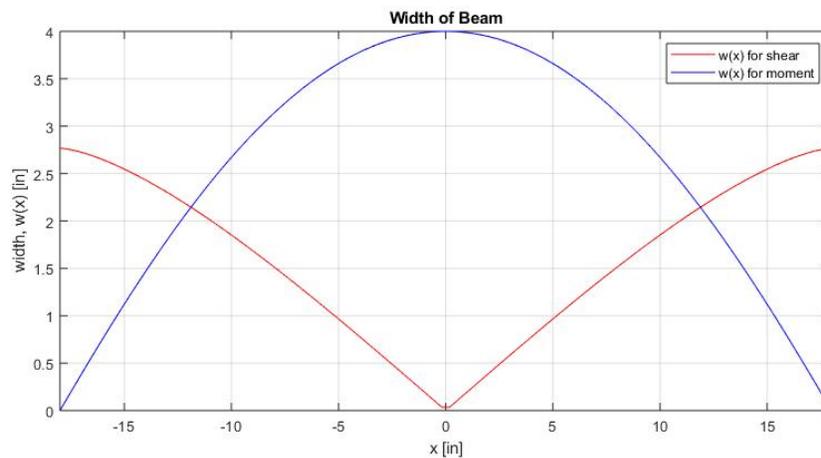


Fig. 7 Overlaid Optimal Widths for Allowable Normal and Shear Stresses

Design Procedure

The given planform was approximately 38 in. long and 4 in. wide, this was cut down to optimize the weight and load bearing capability based on the optimum width function discussed above. The final wing was cut to be 36 in. long as this was the length value we simulated in our code. The 1-inch width measurements were drawn on to the specimen using a pair of dial calipers. With width tick marks placed along the planform, a band saw was used to cut off the excess material.

1. Wiffle Tree Design

Since the wiffle tree design is inherently symmetric, only half of the design was mathematically analyzed. The first step was dividing this region into four smaller zones, since the top bar had four strap locations. The length of each region was the total length of the beam divided by eight. For each of these sections, the horizontal centroid was found. In this equation, the integral bounds are the extents of each region, in this case we take a look at the equation for the first section. (Eqn. 15)

$$\bar{x} = \frac{\int_{-L/2}^{-3L/8} xp(x) dx}{\int_{-L/2}^{-3L/8} p(x) dx} \quad (15)$$

Notice that the denominator of this equation is representative of the area under the distributed load. This is exactly identical to the force applied at the centroid, which helps simplify the equilibrium equations. The main goal of these equations is to determine the location of each strap. The weight of the sleeves and of each bar are factored into the net force equations. Equations 16 and 17 describe the distances used to place each strap along the first layer. Equation 18 describes the distance of the final layer strap. F_{1-4} represent the force from the distributed load for each of the four sections. w_b and w_s are the weight of the beam and the strap, respectively.

$$l_1 = \frac{L(w_s - F_2 - \frac{w_b L}{2})}{2w_s - F_1 - F_2 - \frac{w_b L}{2}} \quad (16)$$

$$l_2 = \frac{L(w_s - F_4 - \frac{w_b L}{2})}{2w_s - F_3 - F_4 - \frac{w_b L}{2}} \quad (17)$$

$$l_3 = \frac{L(w_s - F_3 - F_4 - \frac{w_b L}{2})}{2w_s - F_1 - F_2 - F_3 - F_4 - \frac{w_b L}{2}} \quad (18)$$

The final strap location (holding the bucket) was centered along the length of the total beam. A diagram of the wiffle tree is included below to further describe this scenario and the calculated lengths. This free-body diagram is not to scale, but it illustrates which length is associated with which value. For example, l_1 is the distance from the centroid of the first region (where F_1 is applied) to the strap location for that bar.

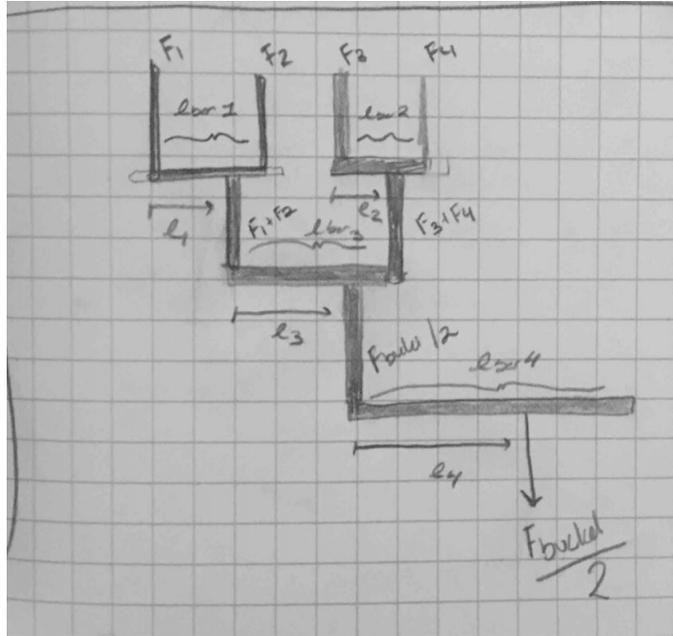


Fig. 8 Wiffle Tree Diagram

The following figure is a scale plot of the entire wiffle tree setup using the lengths calculated. This time, both sides of the wing are shown. The horizontal axis shows the strap locations in meters. The vertical axis has no scale because the lengths of the straps were not given, nor were they necessary in the placement of the straps.

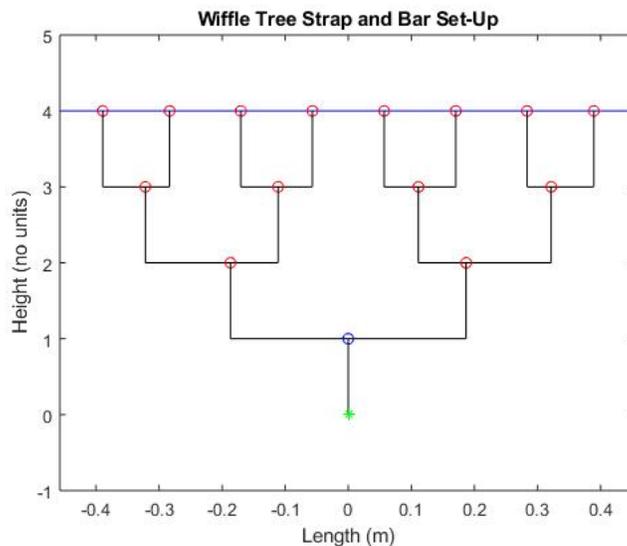


Fig. 9 Wiffle Tree Setup (to scale)

2. Wiffle Tree Loading Procedure

Once the location of the straps had been determined, the wiffle tree was constructed and used to test the wing. There were a few challenges with regard to interpreting the lengths calculated. One was the fact that the distance between centroids was less than the length of the bars. Thus, a convention was adopted that the bar would be centered between

the straps suspending it. This way, the weight of the bar would not affect the moment calculated previously. Secondly, it was assumed that the each length calculated would be measured from the centroid of each component (bar or strap).

With these clarifications in place, the wiffle tree was constructed. First, the distances for the straps were measured on the beams, and the hooks were set in the appropriate locations. The first layer of straps were placed onto the wing at the calculated centroid locations. Then the first layer bars were attached to the hooks, making slight modifications for uniformity. This process was repeated for the second layer, first calibrating the strap locations and the adding the bar. Finally the last bar was attached to the bottom two hooks. To fully complete the setup, the bucket was attached to the bottom long bar.

The wing was tested by placing weights inside the bucket one at a time until failure occurred. Initially, weight was added in 5 lbf increments, followed by 2.5 lbf increments. As deflection increased, weight was added in smaller increments in an attempt to pinpoint the maximum weight the wing could withstand. 1.25 lbf plates were added one at a time, and when two of these plates were added, they were removed and replaced with a 2.5 lbf plate. Towards the very end, small metal balls were placed in the bucket to further reduce the weight increments. Since the team overshot our initial weight estimate, we had to resort to adding 2.5 lbf weights at the end, since there were no more available smaller weights. These additions resulted in failure, after which the bucket and all of its contents were weighed to determine the total load that the wing supported.

Like any physical representation of a theoretical model, a handful of uncertainties and potential sources of error arose. It proved difficult to keep each strap at its exact location on the wing. It was noted during testing that the straps slid towards the center of the wing as bending increased. It was also difficult to place the hooks at their precise locations on the bar. In addition, there was slight torsion in the suspended bars due to the chains in the hooks not lining up exactly. These irregularities certainly contributed to the discrepancy between the predicted and recorded maximum loads as well as the uncertainty for the applied load, but in no way significantly altered the results.

V. Discussion

From the wiffle tree testing, the beam failed under 36.7 lbf. The mathematical model predicted the beam would withstand 18.3 lbf, under a factor of safety of 1.5, and could theoretically withstand 27.4 lbf, with a factor of safety of 1.0. The beam ultimately held 18.4 lbf more weight than designed for, and 9.3 lbf more weight than theoretically possible.

Only two failure modes were modeled in the theoretical portion of the lab, shear and moment. Since the predicted shear force is 0 lbf at the location of failure, it was concluded that the beam failed in moment. However, the actual situation is likely more complex than a simple moment failure due to potential imperfections inside the material. After application of a majority of the load, the top balsa wood cover started to buckle under compression. This slight buckling decreased the structural integrity, causing the balsa wood to crack before the foam and this buckling location, as confirmed by a video taken during testing. Fig. 10 shows the buckling balsa wood that occurred on the right side of the beam.



Fig. 10 Final Wiffle Tree Testing

The final failure weight was significantly more than predicted by the MATLAB code. There are two likely reasons for this high failure weight. The first is over extraneous loading. The group started to hear cracking sounds in the balsa wood around 15 – 20 seconds before failure. Additional weight was applied until failure, culminating with a 2.5 lbf weight. The beam failed instantly upon application of this final weight, and thus would likely have failed from adding a

smaller weight. In addition, if given more time, the beam likely would have fully cracked without adding the final 2.5 lbf weight. This means that the true failure weight was likely a few pounds below the final result.

The second likely cause of the near-10- lbf discrepancy is lack of perfect consistency between the theoretical versus actual models of the beam and its construction. All theoretical analysis was done on a perfect beam case, despite inconsistencies in the construction of the beams. Particularly the group noted that the specific wing tested in this case had an abundance of glue. This glue was pushed out of the sides between the foam core and the balsa cover, and this likely led to a higher strength of connectivity between the two layers. After considering both the extraneous added weight, and the increased strength from an abundance of glue, the difference between predicted and measured final loads is justifiable.

One failure mode that wasn't considered in this lab required for a more realistic wing model was fatigue. As studied in ASEN 1022, structures that undergo cyclic stress, such as a wing over multiple flights, are subject to growth of small micro structure cracks. Propagation of these small cracks in many cases eventually leads to large scale structural failure. Testing fatigue is difficult, and would require a different process than that tested in this lab. Instead, the group would have set a smaller amount of weight on the wing, something that adequately flexes the wing design. Then, the weight would be unloaded and loaded again over the course of multiple days. Realistically, fatigue testing is something that is better conducted with a automated process.

In addition to fatigue, an actual wing design is much more complex than the simple beam our team created. There are mounts for engines, bays for landing gear, attachments for munitions, and much more. These additional systems add more weight, and complicate the distributed loads on the wing. When additional attachments are included on the wing, there are more critical points where the wing can fail. These advanced wing designs can be tested with the wiffle tree method as well. However, point loads or attached weights should be incorporated to accurately represent the weight and moment effects from any additional attachments.

VI. Conclusion

The beam tested in this lab failed in moment after a tested weight of 36.7 lbf. This failure weight was larger than that theoretically predicted with both a factor of safety of 1.5 and 1. The best explanation for this increase in actual failure weight was an overabundance of adhesive between the foam core and the balsa cover, as well as extraneous added weight during wiffle tree testing. More generally, the simplified version of structural loading in this lab only considered failure from shear and moment, when in reality a wide variety of factors are incorporated in real structural failures. Cyclic fatigue, micro-structure imperfections, and additional loads all affect the location and cause of failure. Ultimately, multiple tests over a range of conditions are necessary to adequately model the failure conditions of wings in the aerospace field. Lives depend on proper wing design; therefore, engineers must use extensive testing and a factor of safety in their designs.

VII. References

- “ASEN 2002 Experimental Laboratory 2: Aerodynamics of a Cambered Airfoil”, *University of Colorado Boulder Aerospace Engineering Sciences*, Fall 2018. <https://canvas.colorado.edu/courses/21255/files/folder/Lab%203?preview=2349357>
- Campbell, G.S., and Lahey, R.T.C., (1983) A survey of serious aircraft accidents involving fatigue fracture, Vol. 1 Fixed-wing aircraft, National Aeronautical Establishment, Canada
- Çengel, Y. A., Cimbala, J. M., & Turner, R. H. (2017). *Fundamentals of Thermal-Fluid Sciences* (5th ed.). New York, NY: McGraw-Hill Education.
- Harrison, N. D,m & Findlay, S. J. (2002). Why aircraft fail - Review Feature. Cody Technology Park, Hampshire: QinetiQ Ltd. <http://www.ae.utexas.edu/courses/ase324huang/MT2002.pdf>
- Hibbeler, R. C. *Statics And Mechanics Of Materials*, 5th Edition, Pearson, 2017.

Appendix

Listings

1	Main Function	12
2	Function for Finding Allowable Stresses	15
3	Function for Plotting Shear Diagrams	17
4	Function for Modeling Wiffle Tree Loading	17

Listing 1 Main Function

```
1 %% Housekeeping
2 clc;
3 clear all;
4 close all;
5
6 %% Read in Test data
7 [data_num,~,data] = xlsread('TestData.xlsx');
8
9 %% Initialization
10 L = 36/39.37; %inches converted to meters
11 P0 = 1; %Total Pressure (Pa)?
12 Wx = 4/39.37; %Initial Width inches converted to meters
13
14 Ef = 0.035483*10^9; %Pa
15 Eb = 3.2953*10^9; %Pa
16
17 %% Plot Shear and Moment Diagrams for Test Cases
18 Outs = Sigma_Correct;
19
20
21
22 %% Finding Allowable Stresses
23
24 %Parse column data appropriately
25 tests = data_num(:,1);
26 F_fail = data_num(:,2);
27 a = data_num(:,3);
28 w = data_num(:,4);
29 d_f = data_num(:,5);
30 comments = data(:,6);
31
32 %defining other variables
33 L = 36/(39.37); %converting length of bar in inches to m
34 Ay = F_fail/2; %Ay = By = F/2
35 x_vec_test = linspace(-L/2,0);
36 t_b = (1/32)/39.37; %[m], thickness of balsa
37 t_f = (3/4)/39.37; %[m], thickness of foam
38 Af = t_f*w; %m^2, cross sectional area at failure location
39
40
41 Ib = 2*((1/12).*w*t_b^3 + w.*t_b*(t_b/2 + t_f/2)^2); %[m^4], moment of inertia of balsa wood
    at center cross section
42 If = (1/12)*w*t_f^3; %[m^4], moment of inertia of foam at center cross section
43 c = ((3/8)+(1/32))/39.37; %inches converted to meters half of cross section
44
```

```

45 %shear force and moment at failure
46 V_fail_test = zeros(size(tests));
47 M_fail_test = zeros(size(tests));
48
49 for i = 1:length(tests)
50     if d_f(i) < a(i) %failed before the weight hung
51         V_fail_test(i) = F_fail(i)/2; %in N
52         M_fail_test(i) = V_fail_test(i)*d_f(i); %[N*m], moment due to shear force
53     elseif d_f(i) >= a(i) %failed after or at the weight hung
54         V_fail_test(i) = 0; %in N
55
56         M_fail_test(i) = F_fail(i)/2*a(i); %[N*m], moment due to hung weight
57     end
58 end
59
60
61
62
63
64 %sigma and tau at failure
65 tau = zeros(size(tests));
66 sigma = zeros(size(tests));
67 for i = 1:length(tests)
68     tau(i) = (3/2)*(V_fail_test(i)/Af(i));
69     sigma(i) = (-c*M_fail_test(i))./(Ib(i)+(Ef/Eb)*If(i));
70 end
71
72
73 %useful tests for each
74
75 %Run everything except shear:
76 useful_shear = [10, 11, 15, 18];
77 useful_moment = [1:9,12:14,16,17,19:24];
78
79
80 tau_mean = mean(tau(useful_shear));
81 sigma_mean = mean(sigma(useful_moment));
82
83 %allowable sigma and tau
84 FOS = 1.5;
85 tau_allow = tau_mean/FOS;
86 sigma_allow = abs(sigma_mean/FOS);
87
88 %% Calculations for Design
89
90 %symbolically define necessary equations
91 width = 4/(39.37); %[m], width of beam (4 inches)
92 t_b = (1/32)/39.37; %[m], thickness of balsa
93 t_f = (3/4)/39.37; %[m], thickness of foam
94 c = t_f/2 + t_b; %[m], convert inches to meters
95
96
97
98

```

```

99 Ib = 2*((1/12)*width*t_b^3 + width*t_b*(t_b/2 + t_f/2)^2); %[m^4], moment of inertia of
    balsa wood at center cross section
100 If = (1/12)*width*t_f^3; %[m^4], moment of inertia of foam at center cross section
101
102
103 Ef = 0.035483e9; %[Pa], modulus of elasticity for foam
104 Eb = 3.2953e9; %[Pa], modulus of elasticity for balsa
105 x_vec = linspace(-L/2,L/2); %[cm], trial x locations
106
107 syms x
108 q = width*(sqrt(1-(2*x/L)^2)); %at center, width is 4 in
109 F = int(q,x,-L/2,L/2); %total force at center
110 V = F/2 - int(q, x, -L/2,x); % shear force with respect to distance
111 M = int(V,x, -L/2,x); % bending moment with respect to distance
112 %find M(x=0) - without P0 in equation
113 M_x = matlabFunction(M);
114 M0 = M_x(0); %bending moment at center
115 V_x = matlabFunction(V);
116
117 %shear and moment diagrams
118 figure
119 plot(x_vec,M_x(x_vec));
120 xlabel('X position [m]'); ylabel('M(x) [N*m]');
121 title('Moment Diagram');
122 figure
123 plot(x_vec,V_x(x_vec));
124 xlabel('X position [m]'); ylabel('V(x) [N]');
125 title('Shear Diagram');
126
127 % solve P0
128 P0 = sigma_allow*(Ib + (Ef/Eb)*If)/(M0*c);
129
130
131
132 %%
133 %symbolic equations with P0
134 syms x w_x
135 q_x = P0*(sqrt(1-(2*x/L)^2))*(width); %redefine q, distributed load over length
136 F = int(q_x,x,-L/2,L/2); %total distributed force
137 V = F/2 - int(q_x, x, -L/2,x); %defining shear force with respect to distance
138 M = int(V,x, -L/2,x); %definding bending moment with respect to distance
139
140 Ibforwxmom = 2*((1/12)*w_x*t_b^3 + w_x*t_b*(t_b/2 + t_f/2)^2);
141 Ifforwxmom = (1/12)*w_x*t_f^3;
142
143
144
145 %%
146 %negative here
147 wxCurveEqn = -(M*c) == sigma_allow * (Ibforwxmom + (Ef/Eb)*Ifforwxmom);
148
149 % find width functions for moment and shear
150 wx_moment = solve(wxCurveEqn,w_x);
151 wx_shear = (3/2)*V/(tau_allow*t_f); %[m], width of beam

```

```

152
153 %%
154
155
156 wx_plot_moment = double(subs(wx_moment,x,x_vec))*100; %[cm]
157 wx_plot_shear = double(subs(wx_shear,x,x_vec))*100; %[cm]
158 x_vec_cm = 100*x_vec; %[cm]
159
160 %convert from cm to in
161 x_vec_in = x_vec_cm/2.54; %[in]
162 wx_plot_moment = wx_plot_moment/2.54; %[in]
163 wx_plot_shear = wx_plot_shear/2.54; %[in]
164
165
166 %find intercept
167 x_in = -L/2:0.0254:L/2;
168 WX_MOMENT = double(subs(wx_moment,x,x_in))*39.3701; %[in]
169 WX_SHEAR = double(subs(wx_shear,x,x_in))*39.3701; %[in]
170 X_IN = x_in*39.37;
171 % print these results
172 final = {X_IN,WX_MOMENT,WX_SHEAR};
173 FINAL = cell2table(final,'VariableNames',{'X','wxMoment','wxShear'});
174
175 figure
176 plot(x_vec_in,abs(wx_plot_shear),'r')
177 hold on
178 xlabel('x [in]');ylabel('width, w(x) [in]');title('Width of Beam');
179 plot(x_vec_in,abs(wx_plot_moment),'b')
180 legend('w(x) for shear','w(x) for moment');
181 L_in = 36;
182 axis([-L_in/2,L_in/2,0,4]);
183 grid on
184
185 %% Interception point
186 int_in = 11.88; %[in]
187 int_m = int_in/39.3701; %[m]
188
189 width_int = double(subs(wx_shear,x,-0.3019))*100/2.54; %[in]

```

Listing 2 Function for Finding Allowable Stresses

```

1 function Outs = Sigma_Correct
2
3     %% Read in Test data
4     [data_num,~,data] = xlsread('TestData.xlsx');
5
6     %% Initialization
7     L = 36/39.37; %inches converted to meters
8     P0 = 1; %Total Pressure (Pa)?
9     Wx = 4/39.37; %Initial Width inches converted to meters
10
11     Ef = 0.035483*10^9; %Pa
12     Eb = 3.2953*10^9; %Pa
13

```

```

14 %% Shear Diagram
15
16
17 %Parse column data appropriately
18 tests = data_num(:,1);
19 F_fail = data_num(:,2);
20 a = data_num(:,3);
21 w = data_num(:,4);
22 d_f = data_num(:,5);
23 comments = data(:,6);
24
25 %defining other variables
26 L = 36/(39.37); %converting length of bar in inches to m
27
28 %shear force at failure
29 Ay = F_fail/2; %Ay = By = F/2
30
31
32 %% Shear Diagram
33 figure;
34
35 for i = 1:length(tests)
36     %V(x) test data
37     %determine the shear stress felt at failure during test depending on location of
        failure
38     x = d_f(i);
39     if x>=0 && x<a(i)
40         V_fail_test = F_fail(i)/2;
41     elseif x>= a(i) && x<= L/2
42         V_fail_test = 0;
43     end
44     %plot point (d_f,V_fail) on graph
45     subplot(4,6,i)
46     plot(x-L/2,V_fail_test, '.', 'MarkerSize',20)
47     hold on
48     y_a = linspace(1,(F_fail(i)/2));
49     xlabel('d_f [m]');
50     ylabel(' V(x) [N]');
51     title(['Test #',num2str(i)]);
52     axis([-L/2, 0,0,F_fail(i)/2 + 5]);
53
54     %V(x) theoretical – anonymous function
55     %assign anonymous function that models V(x) for each test
56     V_fail = @(x, a) V(x,a,F_fail(i),L);
57     %define variables for plotting
58     x = 0:1/1000:L/2;
59     for j = 1:length(x)
60         V_x(i,j) = V_fail(x(j),a(i)); %use function to assign V(x) vector
61     end
62     %plot V(x)
63     plot(x-L/2,V_x(i,:))
64 end
65
66 %% Moment Diagram

```

```

67     Ffail = data_num(:,2); %Newtons
68
69
70     c = ((3/8)+(1/32))/39.37; %inches converted to meters half of cross section
71
72     %Second Moment of inertia about the z-axis
73     Ib = ((data_num(:,4)*((3/4)+(2/32))/39.37)^3)/12)-((data_num(:,4)*((3/4)/39.37)^3)/12);
74     Outs(:,1) = Ib;
75     If = ((data_num(:,4)*((3/4)/39.37)^3)/12);
76     Outs(:,2) = If;
77
78     figure
79     for i = 1:length(tests)
80         xindex=find(min(abs(x-data_num(i,5)))==(abs(x-data_num(i,5))));
81         if i==16
82             M_fail(i)=0;
83         else
84             M_fail(i) = trapz(x(1:xindex),V_x(i,(1:xindex)));%Newton*Meters
85         end
86         M_x=cumtrapz(x,V_x(i,:));
87         hold on
88         subplot(4,6,i)
89         plot(x-L/2,M_x(1,:))
90
91         xlabel('X [m]');
92         ylabel('M(x) [N*m]')
93         title(['Test #',num2str(i)]);
94     end
95
96 end

```

Listing 3 Function for Plotting Shear Diagrams

```

1 function V_fail = V(x,a,F_fail,L)
2
3 if x>=0 && x<a
4     V_fail = F_fail/2;
5 elseif x>= a && x<= L/2
6     V_fail = 0;
7 end
8
9 end

```

Listing 4 Function for Modeling Wiffle Tree Loading

```

1 % Lab 3, Wiffle Tree Diagram Calculations
2 % 11/26/2018 - ASEN2001
3 % Jashan C, Emma M, Michael B, Micheal M, Aidan K
4
5 %% Housekeeping
6 clear all
7 close all
8 clc
9
10 %% Given Values

```

```

11
12 lengthbar1 = 6 / 39.37; %[m]
13 lengthbar2 = lengthbar1; % Same as bar 1
14 lengthbar3 = 12 / 39.37; %[m]
15 lengthbar4 = 18 / 39.37; %[m]
16
17 wbar1 = 1.77; %[N]
18 wbar2 = wbar1; % Same as bar 2
19 wbar3 = 2.94; %[N]
20 wbar4 = 3.92; %[N]
21 wstrap = 0.49; %[N] Weight of any strap
22
23 %% Find the x location of each divided section
24
25 L = 36/(39.37); % Length of the entire wingspan
26
27 % ! Note: This considers the left end of the bar as x = 0
28
29 point1 = 0-(L/2); % Left most end of the bar
30 point2 = (L / 8)-(L/2); % Difference between each point is L/8
31 point3 = (L / 4)-(L/2);
32 point4 = ((L * 3) / 8)-(L/2);
33 point5 = (L / 2)-(L/2);
34
35 %% Find the centroid of each section
36
37 % we know distributed load from Po * rest of the last section
38 % Need help from emma / other guys making sure we got the right distributed
39 % load and Po function
40
41 % Function for the distributed load
42 P0 = 1000;
43 px = @(x) P0*sqrt(1-((2*x)/L).^2);
44 pxx = @(x) P0*x.*sqrt(1-((2*x)/L).^2);
45
46 % Centroid 1
47 force1 = integral(px,point1,point2); % The area under distributed load is also the force
48 numerator1 = integral(pxx,point1,point2); % The numerator for the centroid equation
49 xbar1 = numerator1 / force1; % location in [m]
50
51 % Centroid 2
52 force2 = integral(px,point2,point3); % The area under distributed load is also the force
53 numerator2 = integral(pxx,point2,point3); % The numerator for the centroid equation
54 xbar2 = numerator2 / force2; % location in [m]
55
56 % Centroid 3
57 force3 = integral(px,point3,point4); % The area under distributed load is also the force
58 numerator3 = integral(pxx,point3,point4); % The numerator for the centroid equation
59 xbar3 = numerator3 / force3; % location in [m]
60
61 % Centroid 4
62 force4 = integral(px,point4,point5); % The area under distributed load is also the force
63 numerator4 = integral(pxx,point4,point5); % The numerator for the centroid equation
64 xbar4 = numerator4 / force4; % location in [m]

```

```

65 |
66 | %% Determining Location of Second Layer Straps
67 |
68 | % Distance from the left strap on bar 1 to its middle strap
69 | length1 = ((xbar2 - xbar1)*(wstrap - force2 - (wbar1 / 2))) / ...
70 |     (2*wstrap - force1 - force2 - (((xbar2-xbar1)*wbar1) / 2));
71 |
72 | % Distance from the left strap on bar 2 to its middle strap
73 | length2 = ((xbar4 - xbar3)*(wstrap - force4 - (wbar2 / 2))) / ...
74 |     (2*wstrap - force3 - force4 - (((xbar4-xbar3)*wbar2) / 2));
75 |
76 |
77 | %% Determined Location of Third (Final) Layer Strap
78 |
79 | % Length used in this scenario
80 | cdist3 = (xbar3 - xbar2) + (length2) + (xbar2-xbar1) - length1;
81 |
82 | % Numerator and Denominator assigned seperately to avoid confusion
83 | topeq = cdist3*(wstrap - force2 - wbar3/2);
84 | boteq = (2*wstrap - force1 - force2 - (wbar3/2)*cdist3);
85 |
86 | % Distance from the left strap on bar 3 to its middle strap
87 | length3 = topeq / boteq;
88 |
89 | %% Determined Location of the final strap / the bucket
90 | % center of the last bar
91 |
92 | %% Plot Locations
93 |
94 | % plot the wing
95 | x=(-L/2):0.01:(L/2);
96 | y=4;
97 | plot(x,y*ones(size(x)), 'b-')
98 | hold on
99 | axis([(-L/2) (L/2) -1 5]);
100 |
101 | % plot the first layer strap locations
102 | plot(xbar1,4, 'ro', xbar2,4, 'ro', xbar3,4, 'ro', xbar4,4, 'ro')
103 | plot(-xbar1,4, 'ro', -xbar2,4, 'ro', -xbar3,4, 'ro', -xbar4,4, 'ro')
104 |
105 | % plot first layer straps
106 | y=3:4;
107 | plot(xbar1*ones(size(y)),y, 'k-', xbar2*ones(size(y)),y, 'k-', xbar3*ones(size(y)),y, 'k-', xbar4*
    ones(size(y)),y, 'k-')
108 | plot(-xbar1*ones(size(y)),y, 'k-', -xbar2*ones(size(y)),y, 'k-', -xbar3*ones(size(y)),y, 'k-', -
    xbar4*ones(size(y)),y, 'k-')
109 |
110 | % plot first layer bars
111 | x1=xbar1:0.001:xbar2;
112 | x2=xbar3:0.001:xbar4;
113 | x3=-xbar2:0.001:-xbar1;
114 | x4=-xbar4:0.001:-xbar3;
115 | y=3;
116 | plot(x1,y*ones(size(x1)), 'k-', x2,y*ones(size(x2)), 'k-', x3,y*ones(size(x3)), 'k-', x4,y*ones(

```

```

    size(x4)), 'k-')
117
118 % plot second layer strap locations
119 plot(xbar1+length1,3, 'ro',xbar3+length2,3, 'ro',-(xbar1+length1),3, 'ro',-(xbar3+length2),3, '
    ro')
120
121 % plot second layer straps
122 y=2:3;
123 plot((xbar1+length1)*ones(size(y)),y, 'k-',(xbar3+length2)*ones(size(y)),y, 'k-',(-(xbar1+
    length1))*ones(size(y)),y, 'k-',(-(xbar3+length2))*ones(size(y)),y, 'k-');
124
125 % plot second layer bars
126 x1=(xbar1+length1):0.001:(xbar3+length2);
127 x2=(-(xbar3+length2)):0.001:-(xbar1+length1);
128 y=2;
129 plot(x1,y*ones(size(x1)), 'k-',x2,y*ones(size(x2)), 'k-')
130
131 % plot third layer strap locations
132 plot(xbar1+length1+length3,2, 'ro',-(xbar1+length1+length3),2, 'ro')
133
134 % plot third layer straps
135 y=1:2;
136 plot((xbar1+length1+length3)*ones(size(y)),y, 'k-',-(xbar1+length1+length3)*ones(size(y)),y, '
    k-')
137
138 % plot third layer bars
139 x=(xbar1+length1+length3):0.001:-(xbar1+length1+length3);
140 y=1;
141 plot(x,y*ones(size(x)), 'k-')
142
143 % plot location of bucket
144 plot(0,1, 'bo')
145
146 % plot bucket strap
147 y=0:1;
148 plot(0*ones(size(y)),y, 'k-')
149
150 % plot bucket
151 plot(0,0, 'g*')

```