Calculating the Angle of Vanishing Stability

James Jagielski and Danny Burns

Boat Name: Whaler

Quantitative Engineering Analysis:
Boat Module

February 2022

1 Boat Design

When conducting our research into sailboat designs we realized that most mono-hull boat designs are similar. There aren't many boats with strong deviations away from the fundamental hull shape. Thus, we decided to look into sailboats with unusual designs. Our inspiration for our boat design comes from a combination of boats, the MC Scow (Figure 1a) and the Mini Transat (Figure 1b). We wanted to utilize a few key characteristics from these designs that didn't follow the traditional shapes, the rounded bow shape and the hull design (allows the MC Scow boat to sit flat on the water). These two types of boats are extremely different. The Mini Transat 6.50 is an offshore racing designed boat that sails on hydrofoils. The MC Scow is a one-design, flat water, inshore racing sailboat created for performance. Although these boats are made for distinct conditions and operating regimes they both are optimized for performance, a trait we sought to carry over into our design.

(a) MC Scow: inshore flat water racing sailboat.

(b) Mini Transat 6.50: offshore racing sailboat.

Figure 1: The inspirational sailboats for our design.

1.1 Presented Design

1.1.1 Mathematical Representation of the Design

Our boat design is characterized by a few mathematical expressions. For the cross sectional area the boat is defined by,

$$z = .05\left|\frac{x}{.07}\right|^4, \tag{1}$$

where $-0.035 \leq x \leq 0.035$. In the y-axis, we used the equation,

$$y = (7z - 0.37914397806)^8, \tag{2}$$

where $-0.0762 \leq z \leq 0.0762$.

$$x = \pm\sqrt{\frac{y - 0.0762}{-4.3913}}, \tag{3}$$

where $-0.0762 \leq y \leq 0.0762$.

Figure 2: 3D representation of our boat

Each equation was based upon our MATLAB calculations and were converted from the meters into inches. Therefore, our boat dimensions can be defined by a beam of 3.5 inches, height of 1.97 inches, and a length of 6 inches. The Equation (1) was calculated by inputting the height of 0.05 m and a width of 0.07 m into the equation. The width of the boat aids its ability to right itself. The degree of the polynomial was chosen as 4 because it would lower the COM with more mass at the bottom, and it was closest to the hull shapes in Figures (1). If the polynomial degree was decreased, the hull shape would've been too round at the bottom and the COM would have been raised. If the polynomial degree was increased, the hull corners at the bottom would be too sharp, lowering the COM. The polynomial degree of 4 gave us the optimal shape we wanted to achieve in the x-z plane.

The equation for the side profile of the boat (2) was determined by a combination of factors. We wanted a curve in the front that allows the boat to achieve it's planing state at as low of a speed as possible, as boats are substantially more efficient when in planing mode vs displacement mode. Since since this is a racing sailboat, a more efficient hull shape and longer waterline translates to a faster speed, which is a desirable characteristic for obvious reasons. The side profile that this equation produces pushes water under the bow as the boat moves forward, pushing the bow of the boat upwards according to Newton's Third Law of of Motion. This facilitates planing, making for an efficient and fast sailboat. We were considering a few shapes for the hull: elliptic, exponential, and a polynomial shape. At first, we tried an elliptic curve because the curve was more gradual. We pivoted away from this shape because when we implemented it in MATLAB and Solidworks the shape didn't appear as desired, and changing parameters for the curve did not improve it. We chose the polynomial derived profile because it gave the flat bottom to allow the boat to float upright and in a state of static equilibrium. This curve also gives the boat a forward pitch when it heels over.

The equation (3) for the top surface of the boat was chosen to match the shape of the full-scale boat we are emulating. The designers of the full scale boat reached this design decision with efficiency and hydrodynamics in mind, making it a useful feature to carry over to our model. The top profile of this boat is unconventional, however this is a design choice that the original manufacturer made to optimize performance within this particular design's niche, which is racing.

We chose low infill percentages because this allowed us to have a low displacement ratio. This ratio determines how deep the vessel will sit in the water, which has an effect on the AVS. The infill changes the COM of the boat, which changes the angle at which the boat will capsize.

1.1.2 CAD Concepts

When considering potential hull designs we had to comply with the design requirements as well as the performance requirements given to us for the project. The requirements were:

1. The boat must fit within a 6" x 6" x 3" volume (for 3d print build plate / volume).
2. The boat must weigh no more than 200 grams.
3. The boat must be a mono-hull displacement design (no outriggers or catamarans).
4. The boat must have a flat deck; it cannot have a mast.
5. The boat must float.
6. In equilibrium, the deck of the boat should be parallel to the surface of the water (off by < 5 degrees).
7. The angle of vanishing stability (AVS) for the boat must be between 120 degrees and 140 degrees.

We chose certain characteristics from the boats we are emulating that allow us to meet all the requirements while also trying to accurately represent our intended sailboats.

To complete this objective, we used the CAD program Solidworks. This allowed for comprehensive rendering and analysis of our hull shape, as well as the ability to seamlessly transition from the design phase to manufacturing.

Transferring our boat design from MATLAB to Solidworks proved to be challenging at times. The limitations of the equation driven curve function in Solidworks hampered our ability to directly transfer equations, however we were able to successfully reformat our equations to meet the requirements of Solidworks.

As mentioned above, our boat was modeled using three instances of the "equation driven curve" function. Each of these instances defines one plane of the Solidworks file, however these curves overlap to form a complex, three dimensional shape. (7)

The equations (as entered into Solidworks) are:

To describe the profile along the Y axis of the boat:

$$y = 0.05 \left| \frac{x}{0.07} \right|^4, \tag{4}$$

To describe the profile along the X axis of the boat:

$$y = (7x - 0.37914397806)^8, \tag{5}$$

To describe the profile along the Z axis:

$$y = -4.3913x^2 + 0.0762, \tag{6}$$

1.1.3 CAD Process

Our design started by defining the shape along the Y axis of the boat. This was accomplished by entering the first equation (4), setting the bounds according to the width of our boat, manually sketching the top surface of the boat, and finally extruding that shape to the correct length from the mid-plane of the sketch. This procedure resulted in an extruded curve along the length of the future boat, however the top and side profiles had yet to be defined.

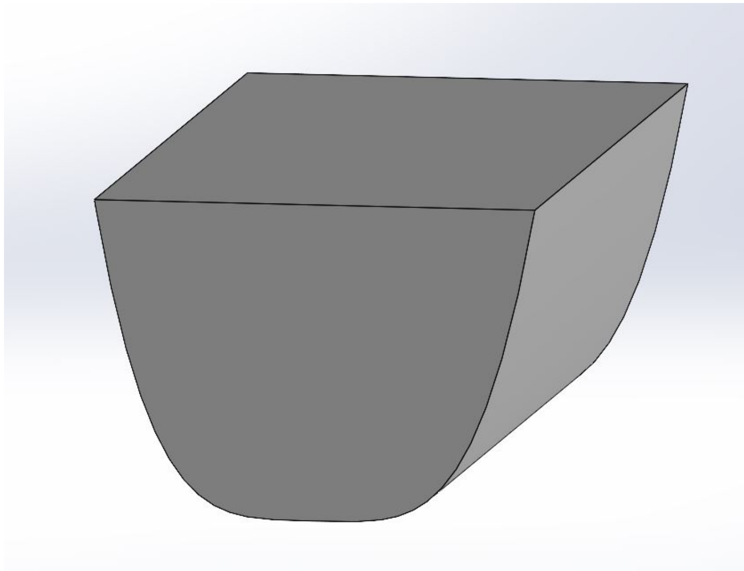


Figure 3: Step 1 in Solidworks CAD

Defining the side profile required a different technique from the procedure used to define the end profile. To complete this step, a reference plane was created perpendicular to the top surface of the boat and in line with one side of the boat. A sketch was initiated on this plane, and a curve was modeled according to the relevant equation (5). Instead of adding material, for this step, the "extruded cut" tool was used to remove all material forward and below of the defined curve. This left the boat with a sloping front, a much more functional and aesthetic form for a sailboat.

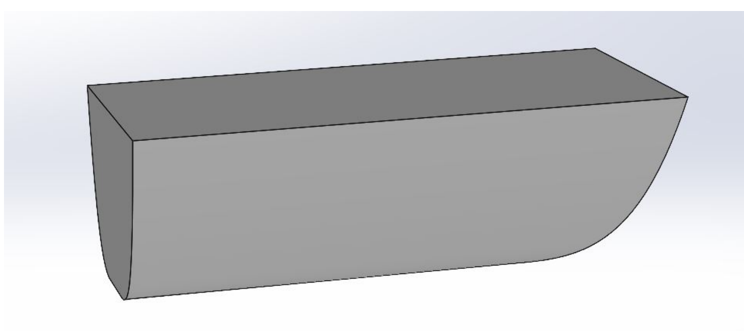


Figure 4: Step 2 in Solidworks CAD

The top profile of the boat was modeled in the same manner as the side profile, except in this case the sketch of the equation driven curve could be drawn directly on the top surface. Once the curve was defined according to the appropriate equation (6), the same method of removing material (extruded cut) was used.

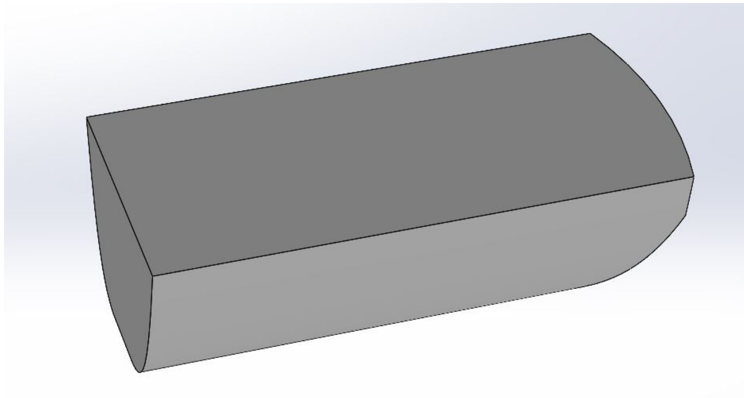


Figure 5: Step 3 in Solidworks CAD

While each of these curves is relatively simple on its own, they combine to form a complex and intricate shape that has been optimized for stability and function.

The final step in Solidworks was to define the point at which the infill percentage changes, which was accomplished by splitting the previously continuous hull. A reference plane was added parallel to the top surface of the boat and at the designated height (1 cm above the bottom surface) and the boat was bisected along this plane using the "split" tool. This produced two separate boat parts, which were then assigned materials with densities corresponding to the calculated infill percentage.

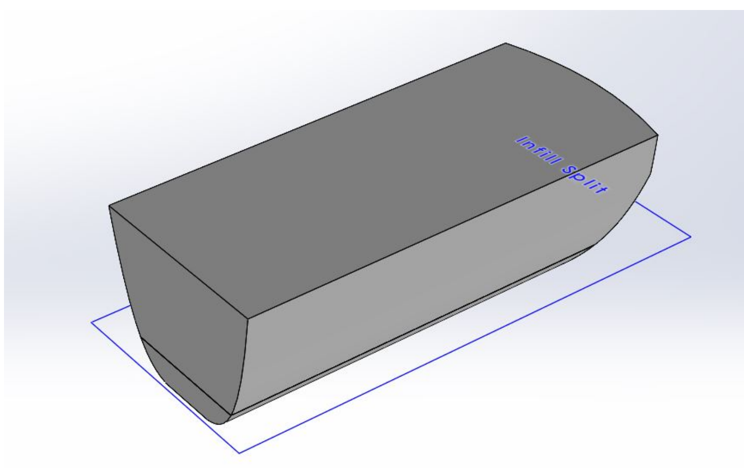


Figure 6: Step 4 in Solidworks CAD

2 Manufacturing

This boat is optimized for additive manufacturing processes such as 3D printing. For the sake of brevity, this report will cover only processes used to manufacture the boat using FDM 3D printing, however other 3D printing methods including SLA, SLS, or DLP could be used with equal success. Additionally, specific 3D printer settings will not be covered as these vary widely between different slicer programs and printers.

Our boat design has two flat faces (top surface and rear surface) which could be placed on the print surface. Placing either of the flat surfaces on the print surface prevents any overhangs, as these would require supports to be printed. This is undesirable as it increases print duration, increases the amount of filament used, and it generally degrades the surface finish of the completed boat.

While both flat surfaces could be theoretically be used, the limitations of the slicing software in use dictate that the top surface of our boat must be aligned with the print surface. This is because the slicing program cannot vary infill settings across the Z axis of the printer, making it impossible to use the rear surface as the base. This is unfortunate, as many more boats could be printed simultaneously on the same printer if the rear surface could be used.

Our design calls for a change in infill density from 40% to 10% across a plane 1 cm from the bottom of the boat. Since our boat will be printed upside down, this means that the slicer settings must be set to change the density at 4 cm from the build plate, which corresponds to the designated change level. Beyond this noteworthy adjustment, the print is relatively simple, as there are no overhangs, holes, or other features that would require supports.

3 Reason for Design

When designing a boat, safety is among the top concerns and considerations engineers need to be cognisant of. In order to ensure safety, measurements of the boats stability are closely analysed. The AVS (Angle of Vanishing stability) curve depicts the stability of the boat at every heel angle of the boat as well as the point at which the boat capsizes. High performance racing boats, like those we're trying to emulate, have more complex features. Speed and efficiency are among the top considerations for such boats, and we plan to produce a boat that would perform efficiently at high speeds.

3.1 Mathematical Theory behind the design

3.1.1 Center of Mass

The COM (center of mass) is the average position of a systems masses,

$$\vec{r} = \frac{m_1 r_1 + ... + m_n r_n}{m_1 + m_n}, \tag{7}$$

where the numerator of the equation is the moment of the system about the origin and the denominator of the equation represents the total mass. The m_n represents the mass of a selected portion of the system and r is the distance from the origin. This distribution of masses can be simplified as the equation for discrete masses,

$$\vec{r} = \frac{\sum_{i=1}^{n} m_n \vec{r_n}}{M_{total}}, \tag{8}$$

and the total mass comes from the equation,

$$M_{total} = \sum_{i=1}^{n} m_i. \tag{9}$$

Using these ideas of mass distribution, we can apply the equation of continuous masses to calculate the total mass bounded by the equations that define our boat in Figure 7. The continuous equation sums an infinite number of small sub-masses of the region in the bounds of integration,

$$M_{Total} = \iiint_R \rho \, dV, \tag{10}$$

where ρ is the density of the boat and integrating in respect to the density in the three axis results in the total mass of the boat. We can then integrate for the center of mass of each axis,

$$z_{COM} = \frac{1}{M_{total}} \iiint_R z\rho \, dV, \tag{11}$$

$$y_{COM} = \frac{1}{M_{total}} \iiint_R y\rho \, dV, \tag{12}$$

not including the x-axis because the boat is symmetrical across it.

3.1.2 Center of Buoyancy

The center of buoyancy is a point force, which is the center of mass of the submerged region under the waterline. We know that for static equilibrium,

$$\sum F = 0, \tag{13}$$

which means that the force of buoyancy and force of gravity when in static equilibrium are equal to each other. Knowing this we can cancel gravity from both forces and we are left with the mass of the displaced water set equal to the mass of the boat. Since we know the mass of the boat, we can solve the integrals for the draft of the boat by inputting the waterline equation (17) as the upper limits of the integrals with,

$$M_{Boat} = \iiint_R \rho_{water} * (x, y, z) \, dzdydz \tag{14}$$

In order to to calculate the center of buoyancy we need the mass of the submerged region and to integrate with upper bounds of the waterline from Equation (17) with the calculated draft. We can use the equation,

$$x_{COB} = \frac{1}{M_{Submerged\,region}} \iiint_R x\rho \, dV \tag{15}$$

$$y_{COB} = \frac{1}{M_{Submerged\,region}} \iiint_R y\rho \, dV \tag{16}$$

to solve for the x,y,z coordinates of the center of buoyancy respectively.

3.1.3 Waterline

The equation,

$$w\,(x) \;=\; \tan{(\theta)} * x \;+ D \tag{17}$$

is the waterline and it rotates with the boat at each θ value. The D is the draft of the boat, which is distance of the waterline to the base of the boat.

3.1.4 Righting Moment

$$\vec{\tau} = \vec{r} \times \vec{F_B} \tag{18}$$

$$\vec{r} = COB - COM \tag{19}$$

$$\vec{F_B} = 0\hat{i} + (sin\Theta)\hat{j} + (cos\Theta)\hat{k} \tag{20}$$

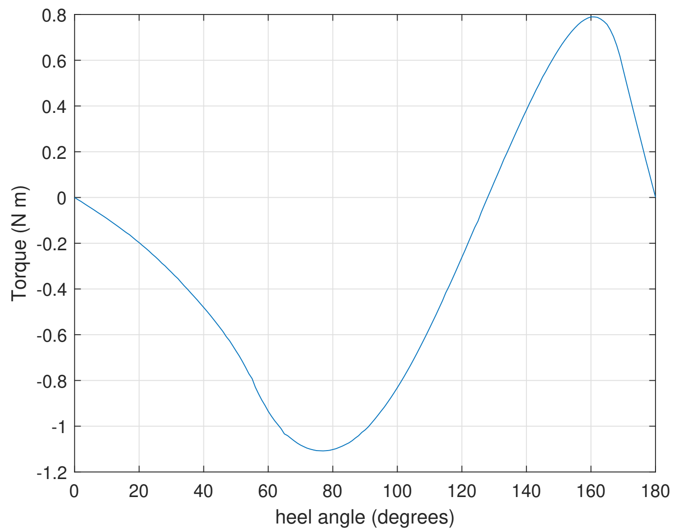
The force of buoyancy always acts perpendicularly to the waterline, consequently the buoyancy force vector maintains a positive magnitude. Due to the vertical misalignment between the COM and COB when the boat is at an angle, a righting moment occurs, returning the boat to its stable equilibrium at zero degrees. This phenomenon occurs until the boat surpasses its AVS (128 degrees), after which the righting moment does not stabilize the boat to upright equilibrium.

3.2 Mathematical Modeling of the Design

3.2.1 Fixed Global variables and assumptions

1. Width = 0.07 (meters)
2. Height = 0.05 (meters)
3. Length = 0.1524 (meters)
4. The density of the boat is 500 $\frac{kg}{m^3}$ below a height of 0.01 (meters) and 125 $\frac{kg}{m^3}$ above a height of 0.01 (meters).
5. The center of mass is [0, -0.0060, -0.0016]
6. The torque is given by the y-component as a result of Equation 16

3.2.2 AVS Curve



.

Figure 7: Angle of Vanishing stability. At 0° the boat has a negative slope and the graph crosses the x-axis at 128°.

The derivative of the AVS graph at y = 0 describes the stability of the boat. A negative derivative means that the boat is stable and a positive derivative indicates an unstable boat. The x intercept(s) of the AVS graph indicate(s) the point(s) at which the restoring torque is zero . Conceptually, this means the COB and COM forces equal each other and their torque vectors point directly at each other. The angle of vanishing stability is 128°, which falls under the performance requirement of capsizing in between the range of 120° and 140°.

3.2.3 Diagrams

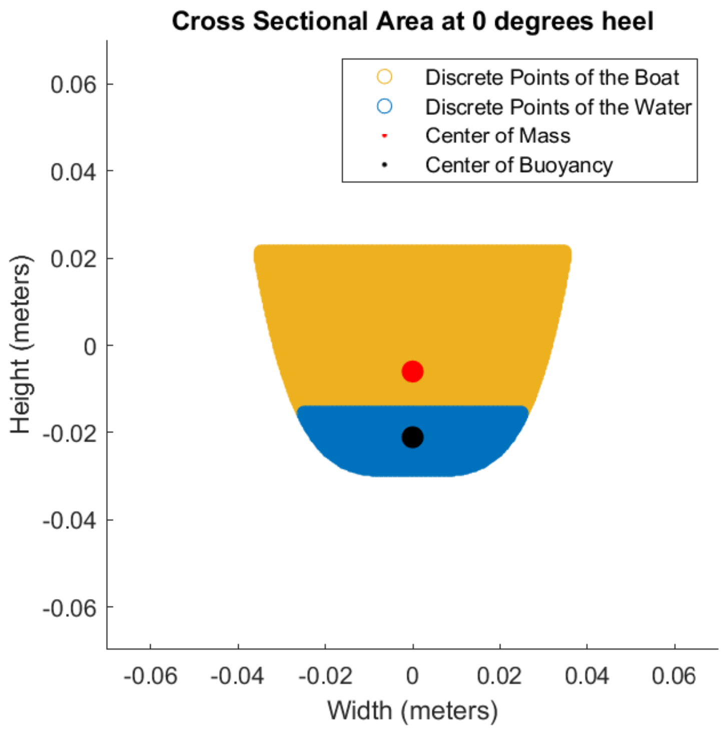


Figure 8: The cross sectional area of the boat in the axis x, z plane with $-0.035 \leq x \leq 0.035$ and $0 \leq z \leq 0.05$ at $0°$ heel.

This angle is significant because it shows what the boat looks like when it's not heeled over. We can also see that the relationship between the COB and COM in this axis for this angle are equal and cancel each other out. This means there is no torque on the boat in the x-direction, from which we can conclude that the boat is stable and won't have heel at $0°$. The AVS curve in figure 7 supports the argument that the boat is in an equilibrium state because because the graph shows the torque computed at $0°$ equals 0 $N * m$. This diagram also illustrates that the boat is floating above the waterline and in MATLAB we calculated the displacement ratio to be 0.1698. This makes sense because the displacement ratio is calculated by,

$$displacement\,ratio = \frac{M_{total}}{\frac{Area_{boat}}{\rho}}. \tag{21}$$

As long as the displacement ratio is less then 1 the boat will remain positively buoyant, which means our boat satisfies this requirement.

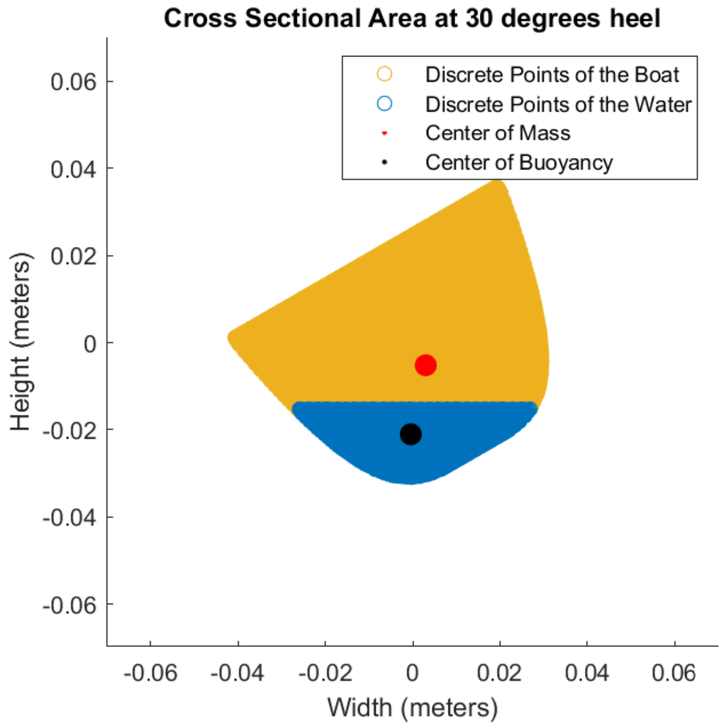


Figure 9: The cross sectional area of the boat in the axis x, z plane with $-0.035 \leq x \leq 0.035$ and $0 \leq z \leq 0.05$ at 30° heel.

This figure shows the boat at a heel of 30°. If we consider the force vectors and torques created by the COB and COM we can see the COB has a torque that will rotate the boat back to the stable equilibrium position at 0° heel. This is important to check because we can see that the boat acts the way we would expect and returns to its static equilibrium.

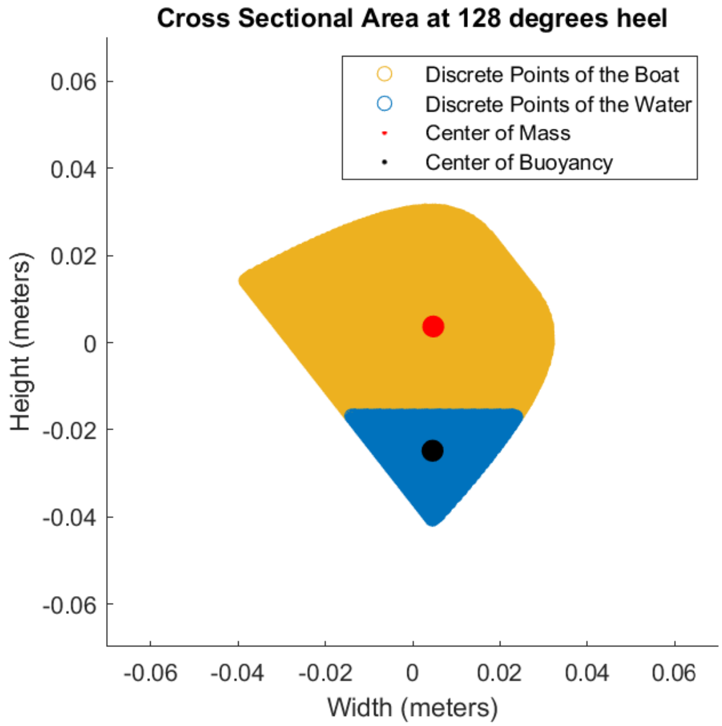


Figure 10: The cross sectional area of the boat in the axis x, z plane with $-0.035 \leq x \leq 0.035$ and $0 \leq z \leq 0.05$ at 128° heel.

When the boat is at a heel of 128° the COB and COM are in line with each other. This is the angle of vanishing stability, which is the angle where the boat will capsize and no longer return to static equilibrium at 0° heel.

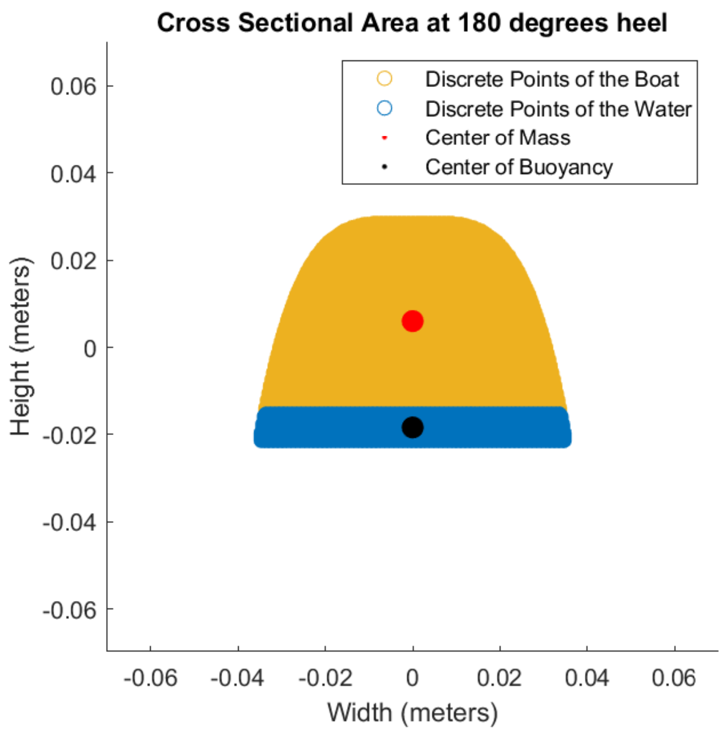


Figure 11: The cross sectional area of the boat in the axis x, z plane with $-0.035 \leq x \leq 0.035$ and $0 \leq z \leq 0.05$ at 180° heel.

The boat at the heel of 180° is fully capsized and again is in a static equilibrium where the COB and COM are in line and create a torque of zero.

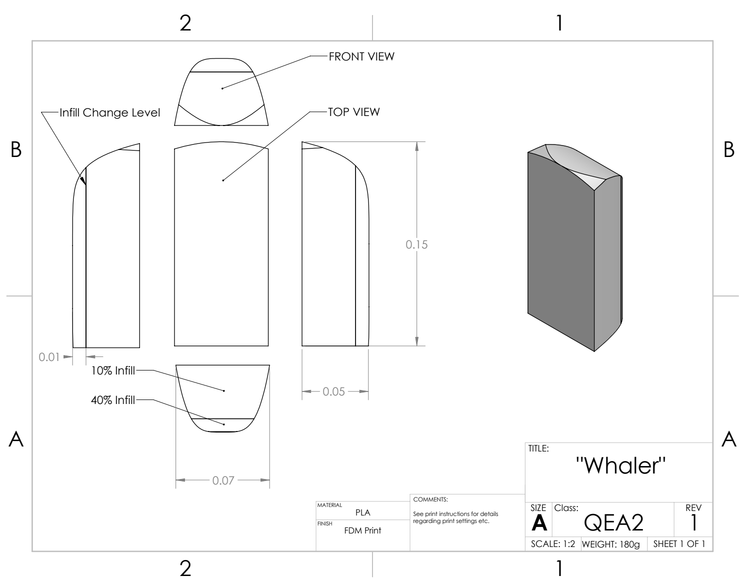


Figure 12: Technical Drawing

4 MATLAB code

```
  clf % clear the current figure
boatcolor = [0.9290 0.6940 0.1250]; % define the color of the boat
watercolor = [0 0.4470 0.7410]; % define the color of the water
% Design parameters
W = 0.07; % width (m)
H = 0.05; % height (m)
L = 0.1524; % length (m)
n = 4;  % shape parameter (-)
dispratio = 0.1698;  % ratio of boat displacement to max displacement
CoM_offset = [0; -0.0060; -0.0016]; % Offset of the center of mass, e.g. using ballast
% physical constants
wrho = 1000; % water density kg/m^3
g = 9.8;   % gravity m/s^2
%% One-time computations
% boat definition and key variables
Npts = 200;           % number of 1D spatial points
(^not a design variable; don't change^)
xPoints = linspace(-W/2,W/2,Npts); % set of points in the x direction (horizontal)
zPoints = linspace(0,H,Npts);   % set of points in the z direction (vertical)
ypoints = linspace(-L/2,L/2,Npts/20);
[X, Z, Y] = meshgrid(xPoints, zPoints, ypoints); % create the meshgrid
P = [X(:)'; Z(:)'; Y(:)'];         % pack the points into a matrix
% find all the points inside the boat - a logical array
insideBoat = transpose(P(2,:) >= H*abs((2*P(1,:)/W)).^n + ...
  (7.*P(3,:) - 0.37914397806).^8 &...
  P(2,:) <= H & P(3,:)<=L/2 + -abs(4.3913.*(P(3,:)).^2)+.1524 & ...
  P(3,:)>= -L/2 & P(1,:) >= -W/2 & ...
  P(1,:) <= W/2);
dx = xPoints(2)-xPoints(1); % delta x
dz = zPoints(2)-zPoints(1); % delta z
dA = dx*dz;         % define the area of each small section
boatmasses = insideBoat*wrho*dA*L; % find the water mass of each small section
maxdisp = sum(boatmasses);     % find the maximum displacement of the boat
boatdisp = dispratio*maxdisp;   % set the displacement of the boat
CoD = P*boatmasses/maxdisp;    % find the centroid of the boat
P = P - CoD;            % center the boat on the centroid
CoD = CoD - CoD;          % update the centroid after centering
CoM = CoD + CoM_offset;      % adjust the center of mass, e.g. for ballast
%% Looped computation
dtheta = 1; % define the angle step
R = [cosd(dtheta) -sind(dtheta) 0; % define rotation matrix
   sind(dtheta) cosd(dtheta) 0;
   0 0 1];
```

```
j = 1; % set the counter
for theta = 0:dtheta:180 % loop over the angles
  % Bound the waterline
  dmin = min(P(2, :)); % find the minimum z coordinate of the boat
  dmax = max(P(2, :)); % find the maximum z coordinate of the boat
  % Solve for the waterline of the rotated boat
  d = fzero(@waterline_error, [dmin, dmax]); % find the waterline d
  % Analyze the boat
  underWater = (P(2,:) <= d)';
  underWaterAndInsideBoat = insideBoat & underWater;
  watermasses = underWaterAndInsideBoat*wrho*dA*L;
  watermass = sum(watermasses);
  CoB = P*watermasses./watermass;
  % Store the results
  torque(j) = boatdisp * g * (CoB(1,1) - CoM(1,1)); % find the torque (N m)
  angle(j) = theta;                 % record the angle (deg)
%   % Uncomment these lines to display a rotated boat;
%   % useful for undertanding, but slows down the computation
   hold off
   %scatter3(P(1,insideBoat),P(3,insideBoat),P(2,insideBoat), [],boatcolor),
   axis equal, axis([-max(W,H) max(W,H) -max(W,H) max(W,H)]), hold on
    scatter3(P(1,insideBoat),P(3,insideBoat),P(2,insideBoat), [],boatcolor),
    axis equal, hold on
    scatter3(P(1,underWaterAndInsideBoat),P(3,underWaterAndInsideBoat),
    P(2,underWaterAndInsideBoat),[],watercolor)
   scatter3(CoM(1,1), CoM(3,1), CoM(2,1), 1000, 'r.'); % plot the COM
   scatter3(CoB(1,1), CoB(3,1) ,CoB(2,1), 1000, 'k.'); % plot the COB
   drawnow
  % Rotate the system
  P = R*P;   % rotate the boat by dtheta
  CoM = R*CoM; % rotate the center of mass too
  j = j + 1;  % update the counter
end
%% Plot the torque versus the angle curve
clf
plot(angle, torque)
xlabel('heel angle (degrees)')
ylabel('Torque (N m)')
grid on
%% TODO: Define the bouyancy error function - should be zero when balanced
function res = waterline_error(d)
  underWater = (P(2,:) <= d)'; % test if each part of the meshgrid is under the
  %water
  underWaterAndInsideBoat = insideBoat & underWater; % the & returns 1 if both
  %conditions are true
```

```
  watermasses = underWaterAndInsideBoat*wrho*dA*L;  % compute the mass of each
  %underwater section
  watermass = sum(watermasses);           % sum up the under-water %masses
  res = watermass - boatdisp; % difference between boat displacement and water
  %displacement
end
end
```