

宁波工程学院

机器人学院

MADE 项目 I 小船项目报告

专业、班级 网络工程(机器人)、18-1 班

姓名 曹鹏霄 学号 18401190120

姓名 张凯科 学号 18402050209

日期 2019 年 11 月 10 日

指导老师 张浩向

目录

摘要:	3
1. 小船设计简介.....	5
2. 符号的定义.....	5
3. 小船设计方法和理由.....	5
3.1 设计概述.....	5
3.2 小船曲面方程的确定.....	6
3.3 龙骨和甲板的确定.....	6
3.4 倾斜不同角度水面方程的确定	7
3.4.1 整体关系式的确定	7
3.4.2 积分方程区域的确定	7
3.4.3 记忆化区间二分查找确定水面方程系数	10
3.5 吃水线的计算.....	11
3.6 浮心和重心的计算.....	11
3.6.1 浮心(COB)	11
3.6.2 重心(COM)	11
4. 制作方案.....	13
5. 比较性能.....	14
6. 结论.....	14
参考文献.....	15

MADE 项目 I 小船项目

曹鹏霄¹，张凯科²

摘要：针对如何使用一块 600mm x500mm 的木板制造一艘在承载一定质量的重物的情况下能够平稳的漂浮起来切具有稳定消失角(AVS)的方案进行了研究。通过数学曲面公式来简单的表示船体的曲面，通过曲面积分来计算船体体积、吃水线性、船体的浮心和重心，确定浮心、重心之后计算出倾斜一定角度时的复原力矩。通过使倾斜一定角度时船体重心在浮力方向和船体中心线的交点上来确定 AVS(135°)。

以小船最低点为坐标原点，垂直垂直向上为 Z 轴，船头至船尾的连线为 Y 轴建立直角坐标系以方程 $z = Ax^2 + By^2$ 为船体底部曲面方程，最终确定 $A=8.33$ ， $B=3.33$ ，计算得船长 length 为 0.379m，宽 width 为 0.240m 高 height 为 0.120m，桅杆重心(0,0,0.292)，重物重心(0,0,0.042)，船体重心为(0,0,0.080)。

关键字：木板；小船；复原力矩；AVS；曲面积分；吃水线；浮心；重心；

Abstract: This paper studies how to use a 600mm x 500mm board to make a vessel which can float smoothly and have a stable vanishing angle (AVS) under the condition of carrying a certain weight. The curved surface of the hull is simply represented by mathematical curved surface formula, and the hull volume, draught linearity, buoyancy center and center of gravity are calculated by curved area division. After the buoyancy center and center of gravity are determined, the restoring moment at a certain angle is calculated. The AVS (135°) is determined by making the center of gravity of the hull at the intersection of the buoyancy direction and the hull centerline at a certain angle.

Taking the lowest point of the boat as the coordinate origin, the vertical vertical upward as the z-axis, and the line from the bow to the stern as the y-axis, the rectangular coordinate system was established, and the equation was taken as the curved surface equation of the bottom of the boat. Finally, $a = 8.33$, $B = 3.33$ was determined. The length of the ship was 0.379m, the width of the ship was 0.240m, the H eight of the ship was 0.120m, the center of gravity of the mast (0,0.292), the center of gravity of the heavy object (0,0.042), and the center of gravity of the ship was (0,0.080).

Keywords: plank; boat; restoring moment; AVS; surface integral; waterline; center of buoyancy; center of gravity;

1. 小船设计简介

机器人学院第一个 MEAD 项目小船项目作为机器人学院大二学生的其中考核项目旨在锻炼学生的运用数理知识和计算机相关软件来解决实际问题的能力。小船模型的设计要求理解重心、浮心、复原力矩、AVS、船舶稳性的概念。需要确定小船的长宽高的比例，它们在一定程度上决定着你小船大部分的性能。其中数学的曲面积分、物理的力矩知识将是项目的最大挑战。

2. 符号的定义

表(3-1) 符号说明

符号	说明
Ma	桅杆质量
Mb	重物质量
Mc	船体质量
h	船高
x_1	水面与船体曲面正半轴交点
x_2	水面与船体曲面负半轴交点
x_{mid}	水面与船体甲板交点
h	水平吃水线高度
M_s	复原力矩
r	复原力臂

3. 小船设计方法和理由

3.1 设计概述

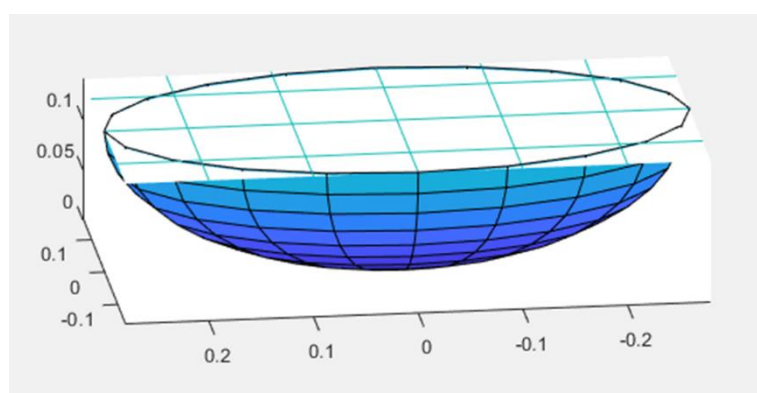
首先以小船最低点为坐标原点，垂直垂直向上为 Z 轴，船头至船尾的连线为 Y 轴建立直角坐标系。之后通过经验确定小船的曲面方程模型和小船的长宽高，进而计算出曲面方程的参数。计算出小船的甲板方程和倾斜各个角度的水面方程，从而确定了各倾斜角度的计算体积和重心时的积分上下限。计算出 135° 时的浮心坐标，将浮力方向与 Z 轴坐标的交点

定为的船体的整体重心坐标 $COM(0,0,Z_{all})$ ，如此便可将小船 AVS 确定为 135° 。计算出纯船体的重心坐标通过组合体重心公式来确定桅杆和重物的坐标。

3.2 小船曲面方程的确定

通过对船舶资料的阅读和现实中船舶的形状来选择小船底部的曲面方程。绘制出 $z = Ax^2 + By^2$ 的曲面图，如图(3-1)，发现与船体形状十分相像决定使用 $z = Ax^2 + By^2$ 作为船体的曲面方程。

据经验确定长宽比值为 1.58，高宽比值为 1，船高 H 确定为 0.12m。



图(3-1) $z = Ax^2 + By^2$ 的曲面图

3.3 龙骨和甲板的确定

船舶龙骨是在船体的基底中央连接船首柱和船尾柱的一个纵向构件，它位于船的底部。通过对龙骨方程的计算即可确定船体龙骨。令船体曲面方程 $z = Ax^2 + By^2$ 中的 $x=0$ 即可获得龙骨方程

$$z = By^2 \quad (3-1)$$

从而确定龙骨大小。

甲板是船体的重要构件，是船舶结构中，位于内底板以上的平面结构，用于封盖船内空间而我们小船模型中的甲板位于船体最上方。对于甲板的确定也可同确定龙骨大小一样计算甲板的方程。令 $z = H$ 可以获得甲板方程

$$H = Ax^2 + By^2 \quad (3-2)$$

3.4 倾斜不同角度水面方程的确定

3.4.1 整体关系式的确定

当小船浸入水中时根据阿基米德浮力定律有

$$G_{\text{总}} = M_{\text{all}} = F_{\text{浮}} = \rho_{\text{水}} v_{\text{排}} g \quad (3-3)$$

其中

$$v_{\text{排}} = \iiint_E dv$$

3.4.2 积分方程区域的确定

当小船倾斜不同角度时 x, y, z 的积分上下限确定方程不同, 可以分为 4 种情况, 如图 (3-2)

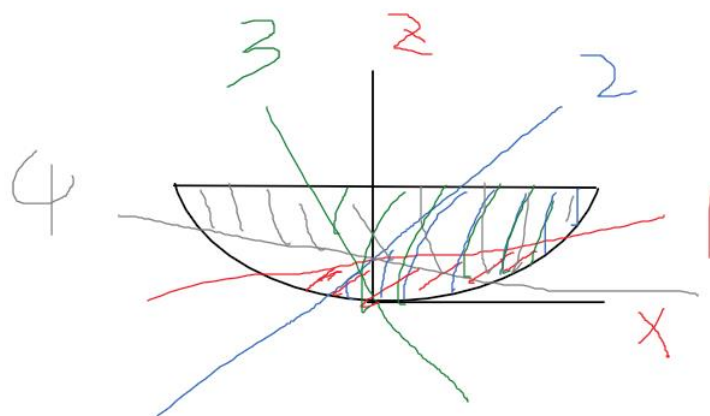


图 (3-2) 四种不同情况下的水面方程

定义水面方程与船底曲面相交于 x 轴正半轴的 x 坐标为 x_1 , 交负半轴坐标为 x_2 , 交甲板方程为 x_{mid} , 水面方程为 $z = px + z_0$, $p = \tan \theta$, θ 为船体倾斜角。

船体无倾斜时

$$v_{\text{排}} = \iiint_E dv \quad \text{中的} \quad E = \{(x, y, z) | x_2 \leq x \leq x_1, -y_0 \leq y \leq y_0, Ax^2 + By^2 \leq z \leq h\} \quad \text{其中 } h \text{ 为吃}$$

水线高度, $y_0 = \sqrt{\frac{h - Ax^2}{B}}$, x_1, x_2 由方程 $\begin{cases} z = Ax^2 + By^2 \\ z = px + z_0 \end{cases}$ 确定, 其中 $z = px + z_0$ 为水面

方程, 此时 $p = \tan \theta, \theta = 0^\circ$, $z_0 = h$

船体倾斜时

假设水面方程过甲板最右端点 $(\sqrt{\frac{H}{A}}, 0, H)$ 时的角度为 θ_R ，过左端点 $(-\sqrt{\frac{H}{A}}, 0, H)$ 时的角度为 θ_L 。则水面可有分为以下四种情况：

(1)

$$0 < \theta < \theta_R;$$

由于此时小船排开水的体积积分区域的 y 的范围会随着水面方程的改变而改变不方便计算，我们可以积分计算出船体水上体积 $V_{\text{上}}$ ，再通过总体积 V_{all} 减去船体水上体积即可得船体排开水的体积 $V_{\text{排}}$ 。

$$\begin{cases} V_{\text{上}} = \iiint_E dv = 2 \left(\int_{-\sqrt{\frac{H}{A}}}^{x_2} \int_0^{y_0} \int_{Ax^2+By^2}^H dz dy dx + \int_{-\sqrt{\frac{H}{A}}}^{x_2} \int_0^{y_0} \int_{px+z_0}^H dz dy dx + \int_{x_1}^{\sqrt{\frac{H}{A}}} \int_0^{y_0} \int_{Ax^2+By^2}^H dz dy dx \right) \\ V_{\text{排}} = V_{\text{all}} - V_{\text{上}} \end{cases} \quad (3-4)$$

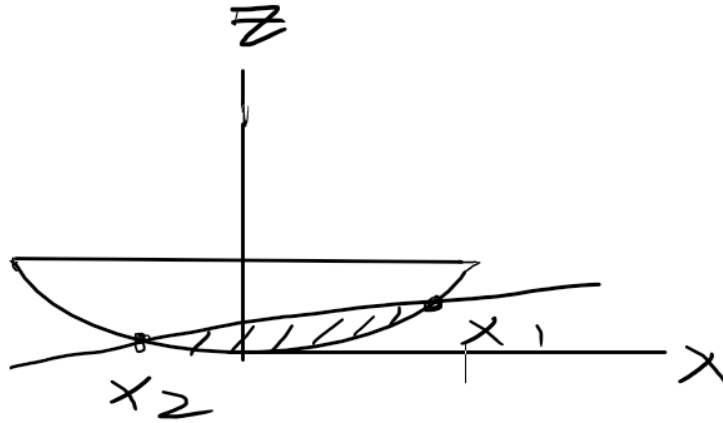


图 (3-3) $0 < \theta < \theta_R$ 时水面与船体 xoz 平面图

(2)

$$\theta_R < \theta < \frac{\pi}{2};$$

此时和 a 相同 y 的积分区域不好确定我们可以通过总体积来减去船体水上体积来得到

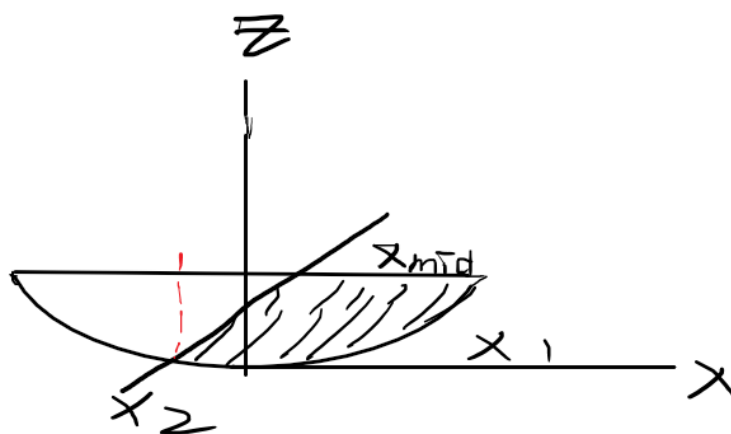


图 (3-4) $\theta_R < \theta < \frac{\pi}{2}$ 时水面与船体 xOz 平面图

$$\begin{cases} v_{\text{上}} = \iiint_E dv = 2 \left(\int_{-\sqrt{\frac{H}{A}}}^{x_2} \int_0^{y_0} \int_{Ax^2+By^2}^H dz dy dx + \int_{x_{\text{mid}}}^{x_2} \int_0^{y_0} \int_{px+z_0}^H dz dy dx \right) \\ v_{\text{排}} = v_{\text{all}} - v_{\text{上}} \end{cases} \quad (3-5)$$

(3)

$$\frac{\pi}{2} < \theta < \theta_L :$$

时 z 轴的积分上限为 H 所以 y 的上限不会受到 z 轴坐标的影响直接计算比较方便计算

$$v_{\text{排}} = \iiint_E dv = 2 \left(\int_{x_1}^{x_{\text{mid}}} \int_0^{y_0} \int_{px+z_0}^H dz dy dx + \int_{x_1}^{\sqrt{\frac{H}{A}}} \int_0^{y_0} \int_{Ax^2+By^2}^H dz dy dx \right) \quad (3-6)$$

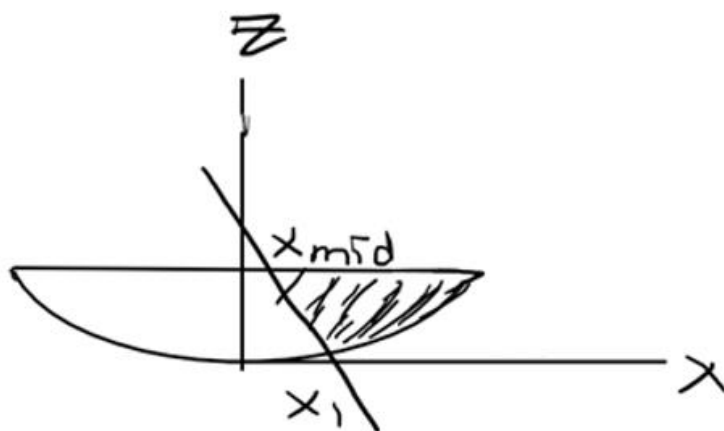


图 (3-5) $\frac{\pi}{2} < \theta < \theta_L$ 时水面与船体 xOz 平面图

(4)

$$\theta_L < \theta < \pi;$$

此时 z 轴的积分上限为 H 所以 y 的上限不会受到 z 轴坐标的影响直接计算比较方便计算

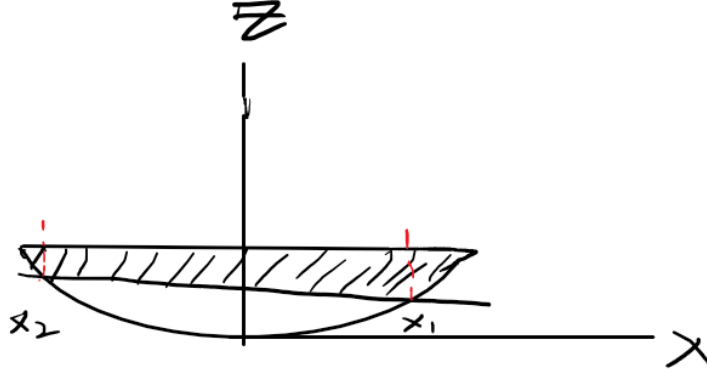


图 (3-6) $\theta_L < \theta < \pi$ 时水面与船体 xoz 平面图

$$v_{\text{排}} = \iiint_E dv = 2 \left(\int_{-\sqrt{\frac{H}{A}}}^{x_2} \int_0^{y_0} \int_{Ax^2+By^2}^H dz dy dx + \int_{-\sqrt{\frac{H}{A}}}^{x_2} \int_0^{y_0} \int_{px+z_0}^H dz dy dx + \int_{x_1}^{\sqrt{\frac{H}{A}}} \int_0^{y_0} \int_{Ax^2+By^2}^H dz dy dx \right) \quad (3-7)$$

3.4.3 记忆化区间二分查找确定水面方程系数

通过记忆化的区间二分查找可以确定水面方程 $z = px + z_0$ ($p = \tan \theta$) 在不同倾斜角度下, z_0 的值。

当遇到方程无法直接求解的情况下, 可以通过程序迭代来试符合要求的方程解。一般迭代算法可以使用梯度下降法来迭代, 但是梯度下降法的时间复杂度是 $O(N)$ 的, 再加上 matlab 的内置积分函数耗时也是十分大的, 如果使用这种方法甚至会导致计算一个倾斜角度的水面方程需要花费 10 分钟, 这是无法接受。而可以发现函数是一个单调递增的, 这就可以使用二分查找的方法来优化迭代算法, 二分查找的时间复杂度是 $O(\log N)$, 这样计算一个角度可以缩短到 1 分钟以内。但是这还是不够快。经过观察发现当船体的倾斜角度 θ 增大, x 的区间也是在单调变化的。所以我们可以利用上一次的 x 的区间值来缩减二分查找的区间范围, 这样平均能提高 3 倍的运算速度, 具体根据数据的分布不同会有不同的差别。

3.5 吃水线的计算

当小船浸入水中时根据阿基米德浮力定律有 $G_{\text{总}} = M_{\text{all}} = F_{\text{浮}} = \rho_{\text{水}} v_{\text{排}} g$ 其中 $v_{\text{排}} = \iiint_E dv$ 有

$$\begin{cases} v_{\text{排}} = \iiint_E dv = 2 \int_{-\sqrt{\frac{h}{A}}}^{\sqrt{\frac{h}{A}}} \int_0^{y_0} \int_{Ax^2+By^2}^h dz dy dx \\ y_0 = \sqrt{\frac{h-Ax^2}{B}} \end{cases} \quad (3-8)$$

3.6 浮心和重心的计算

3.6.1 浮心(COB)

在确定了水面方程和排开水的积分区域之后计算浮心会十分便捷。假设浮心 $COB(x_b, y_b, z_b)$ 有

$$\begin{cases} x_b = \frac{1}{v_{\text{排}}} \iiint_E x dv \\ y_b = 0 \\ z_b = \frac{1}{v_{\text{排}}} \iiint_E z dv \end{cases} \quad (3-9)$$

积分区域 E 在 3.4.2 积分方程区域的确定已经计算完毕。

无法直接进行积分的需要使用组合体重心计算公式来进行推算

$$\frac{\sum_{i=1}^n m_i \cdot com_i}{\sum_{i=1}^n m_i} \quad (3-10)$$

3.6.2 重心(COM)

要使 AVS 等于 135° ，需要使复原力矩等于零，根据公式 $\begin{cases} M_s = r \times F_{\text{浮}} \\ r = COM - COB \\ F_{\text{浮}} = \langle \tan \theta, 0, -1 \rangle \lambda \end{cases}$ 要使重心在浮力的作用线上。首先计算出其浮心 $COB(x_b, y_b, z_b)$ ， $y_b = 0$ 。已知水面方程 $z = px + z_0 (p = \tan \theta)$ 。假设重心 $COM(x_m, y_m, z_m)$ 根据 $M_s = 0$ 便可得

$$\begin{cases} x_m = 0 \\ y_m = 0 \\ \tan \theta (z_b - z_m) + x_b = 0 \end{cases} \quad (3-10)$$

3.7 数量关系图的绘制

3.7.1 重物质量与吃水线关系的曲线图

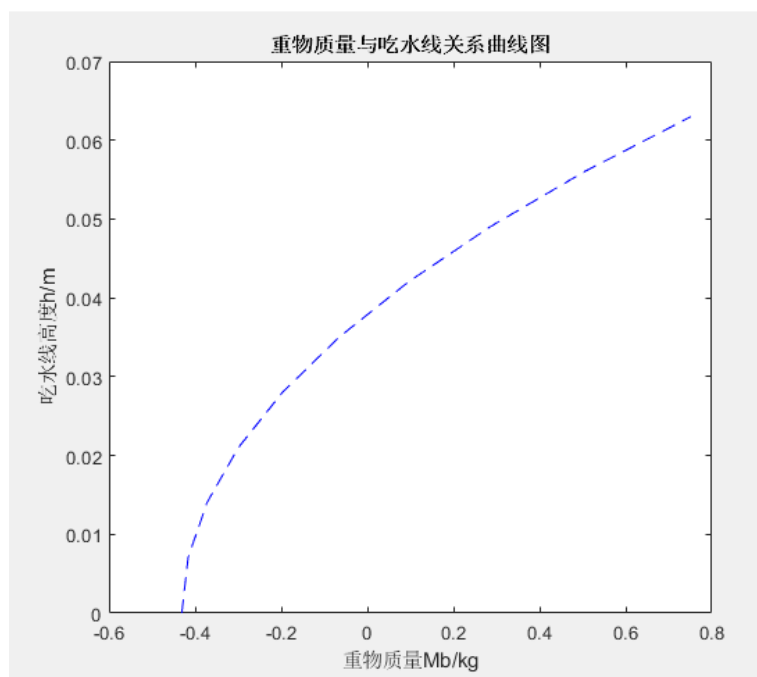


图 (3-6) 重物质量与吃水线关系的曲线图

3.7.2 重物质量与重心关系的曲线图

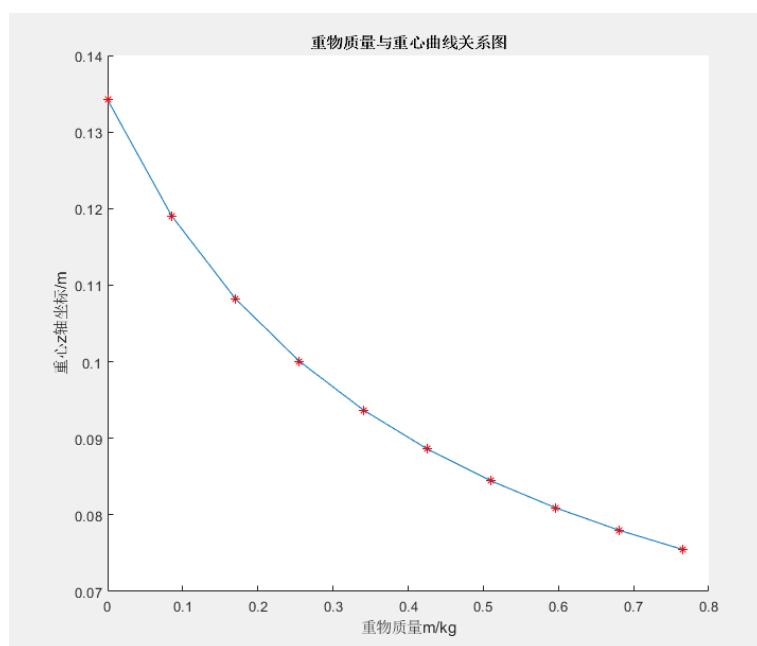
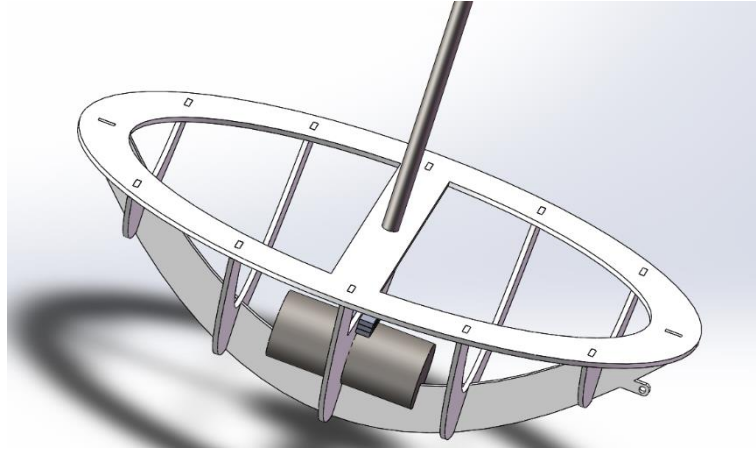


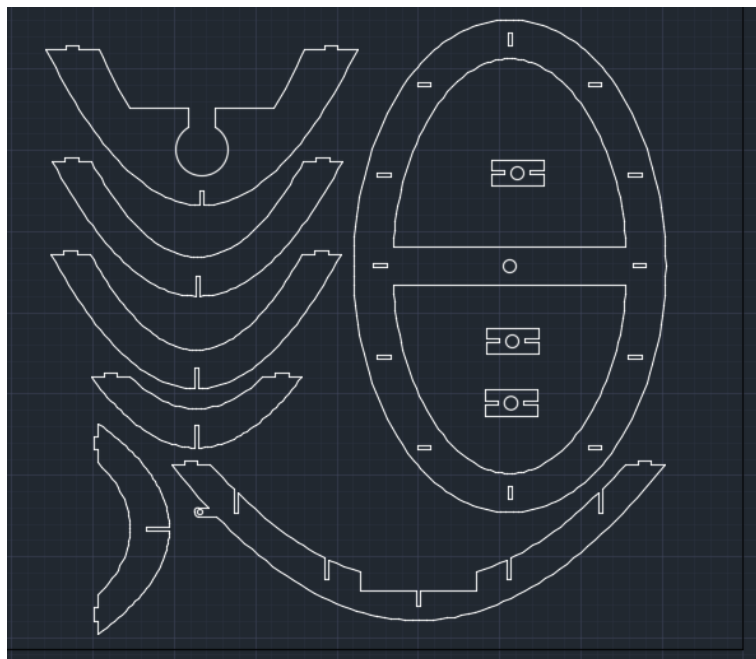
图 (3-7) 重物质量与重心关系的曲线图

4. 制作方案

经过计算确定了小船的长宽高之后通过 solidworks 绘制小船骨架的三维模型，再将三维模型转成 cad 可读的二维工程图，最后通过激光从木板中切割出骨架零件进行拼装。



图（4-1）小船 solidwords 三维模型图



图（4-2）小船 CAD 二维工程图



图（4-3）小船实体图

5. 比较性能

理论性能：AVS 为 135°

实际性能：AVS 为 125°

意义：理论计算和实际测试一定会有所差别，在计算的过程中应该考虑到理论与实际的误差尽可能的减小这种误差。设计方案和计算应该有一定的容错性。

6. 结论

小船项目最重要的一部分便是对于多重积分的理解和掌握还有能够使用程序辅助自己进行计算。在积分过程中最难点便是如何去确定小船体积和浮心的积分区域。在此，我们将其积分区域的确定按照交点位置的不同分成了 4 种情况分别计算，直接积分计算有难度的使用间接法来简化计算。

这次的小船并没有考虑小船的速度性能，想要将小船的行驶速度提高就绝对不能向我们这次一样用简单的曲面进行确定船体形状，最好把船头部分进行分段优化减小船体行进时受到的阻力。

参考文献

- [1] Kyle Combes:Boat hull design. (2015). <http://portfolio.kylecombes.com/>
- [2] 524890689: 船舶稳性教案. 百度文库 (2015-04-25).
- [3] 李子富, 杨盐生, 杜嘉立. 静水中船舶大倾角横摇稳性的理论计算方法. 《大连海事大学学报》, (2004)

附录

```
function [A, B, H, volume,V_all,M_all,rho,g,thetaL,thetaR] = setValue()
%[A, B, H, volume,V_all,M_all,rho,g,thetaL,thetaR]
%定义了小船模型的基本参数单位均为国际制单位
format short
H = 0.12;
% zg = H*4.1/(1.75+4.10);
Phw = 1;%height:width=Phw 为高和宽的比值
A = Phw^2/H;
Plw = sqrt(2.5);%length:width=Plw 为长和宽的比值
B = A/Plw^2;
Mc = 0.321;%小船的质量
Ma =0.11;%桅杆的质量
Mb = 0.766;%重物的质量
M_all = Ma+Mb+Mc;
g = 9.8;
rho = 1 * 10^3;
volume = M_all / rho;
% h = 0.0635;
V_all = 0;%vpa(CalcV_all(), 6);%计算小船全部的体积
thetaL =147.320 * pi/180;%Calc_theta_1();
thetaR =23.407 * pi/180;%Calc_theta_0(A,B,H,volume);
% COB_135 = [ 0.0607, 0, 0.09736];
End
```

```
function xLim3 = xLimCOB(theta,x1,xMid,x2)
[A,~,H,volume,V_all,~,~,~,thetaL,thetaR] = setValue();
tol = 10^-6;%精度
runtimes = 100;
precision = 6;%分数转小数保留的位数
if nargin == 1
    x1 = 0;%初始化 x1 作为 5 的第一次迭代下界
```



```

        x2 = -sqrt(H/A);%初始化 x2 作为 1 的第一次迭代下界
        xMid = sqrt(H/A);
    end
    xLim3 = [x1 xMid x2];
    % fileID =fopen('迭代信息.txt','w');
    cnt = 0;
    res = 100;
    p = tan(theta);
    if theta > thetaL %1
        xMax = sqrt(H/A);
        xMin = 0;%DP
        fprintf('1: ');
        while abs(res) > tol && cnt < runtimes
            cnt=cnt+1;
            % fprintf('xMax xMin: %4.f %4.f\n',xMax,xMin);
            x1 = (xMax+xMin)/2;
            % fprintf('x1 = %4f\n',x1);
            xLim = Calc_xLim(theta,false,x1);%函数中进行了分段
            x1 = xLim(1);
            x2 = xLim(2);
            % xMid = xMid;
            z0 = -A*xLim(1)*xLim(2);
            res = real( CalcVwater(theta,x1,xMid,x2,z0 ))-volume;%res 为当前 V-V 排
            if res > tol%waterVolme 大了 需将 x1 变大增大积分范围
                xMin = x1;
            elseif res < -tol
                xMax = x1;
            end
            % fprintf(fileID,'x1 = %4f xMid = %8f x2 = %4f res = %6f\n',x1,xMid,x2,res);
            % fprintf('x1 = %4f xMid = %8f x2 = %4f res = %6f\n',x1,xMid,x2,res);

        end
    elseif theta > pi/2 %2
        fprintf('2: ');

```

```

xMidMax = xMid;
xMidMin = -sqrt(H/A);
while abs(res) > tol && cnt < runtimes
    cnt=cnt+1;
    xMid = (xMidMin+xMidMax) / 2;
    xLim = Calc_xLim(theta,false,xMid);
    x1 = xLim(1);
    z0 = H - p*xMid;
    Vdown = CalcVwater(theta,x1,xMid,x2,z0);
    res = Vdown - volume;
    if res < -tol % 体积太小将 xMid 变小
        xMidMax = xMid;
    elseif res > tol
        xMidMin = xMid;
    end
    %          fprintf(fileID,'x1 = %.4f xMid = %.8f res = %.6f\n',x1,xMid,res);
end
elseif theta == pi/2 %3
    xMax = -sqrt(H/A);
    xMin = -sqrt(H/A);
    fprintf('3: ');
    while abs(res) > tol && cnt < runtimes
        cnt=cnt+1;
        xMid = (xMax+xMin) / 2;
        xLim = [sqrt(A/H) xMid];
        res = CalcVwater(theta,x1,xMid,x2,z0) - volume;
        if res < -tol% V_down 小了 需将 xMid 变小增大积分范围
            xMax = xMid;
        elseif res > tol
            xMin = xMid;
        end
    end
end
elseif theta > thetaR %4
    fprintf('4: ');

```

```

xMidMax = xMid;
xMidMin = -sqrt(H/A);
while abs(res) > tol && cnt < runtimes
    cnt=cnt+1;
    xMid = (xMidMax+xMidMin) / 2;
    xLim = Calc_xLim(theta,false,xMid);
    x2 = xLim(2);
    z0 = H - p*xMid;
    Vdown = CalcVwater(theta,x1,xMid,x2,z0);
    res = Vdown - volume;
    res = vpa(res,precision);
    if res > tol % 体积太大将 x1 增大
        xMidMin = xMid;
    elseif res < -tol
        xMidMax = xMid;
    end
    %          fprintf(fileID,'xMid = %.8f res = %.6f\n',xMid,res);
end

else %5
    xMax = sqrt(H/A);
    xMin = x1;%0;动态规划思想
    fprintf('5: ');
    while abs(res) > tol && cnt < runtimes
        cnt=cnt+1;
        xMid = (xMax+xMin) / 2;
        xLim = Calc_xLim(theta,false,xMid);%函数中进行了分段
        z0 = -A*xLim(1)*xLim(2);
        res = CalcVwater(theta,x1,xMid,x2,z0)-volume;%res 为当前 V-V 排
        if res < -tol%V_down 小了 需将 x1 变大增大积分范围
            xMin = xMid;
        elseif res > tol
            xMax = xMid;
        else

```

```

        break;
    end

    %          fprintf(fileID,'xMid = %.8f res = %.6f\n',xMid,res);
    end
    x1 = xMid;
end
xLim3 = [x1 xMid xLim(2)];
%          fprintf('%d-th xMid = %.4f xMid = %.8f\n',i,xMid,xMid);
End

%计算得出小船的总体积、长、宽、高和左右两边的特殊角[thetaR thetaL]
format short;

[A, B, H, volume,~,M_all,rho,g] = setValue();
length = sqrt(H / B);
width = sqrt(H / A);
V_all = CalcV_all();% 计算小船全部的体积
%---求船型重心--
syms x
y0 = sqrt( (H - A*x.^2) / B);
funZg = y0.*( H^2 - A^2*x.^4) - 2/3*A*B*x.^2.*y0.^3 - B^2*y0.^5/5 ;
ZgMin = -sqrt(H / A);
ZgMax = sqrt(H / A);
Zg = int(funZg, ZgMin, ZgMax)/V_all;
fprintf('V_all = %.5f length = %.5fm width = %.5fm height = %.5fm\n',V_all,2*length,
2*width, H);
fprintf('COM (0, 0, %.3f)\n',Zg);

thetaR=Calc_thetaR() * 180/pi;
thetaL=Calc_thetaL() * 180/pi;
fprintf('thetaR = %.2f° thetaL = %.2f°\n', thetaR, thetaL);

```

```

% function showWaterlinRes()
% 计算并画出重物与吃水线的关系图

clear

Ma = 0.110;% 桅杆
Mb = 0.321;% 船体
% Mc = 0.770;% 重物
[A,B] = setValue();
h = linspace(0, 0.063,10);
sizeH = size(h);
% syms x
% y0 = sqrt( (h - A*x.^2)/B);

res = linspace(1,1,10);
rho = 1e3;% 水的密度
for i = 1:sizeH(2)
%     VolMin = -sqrt(h(i) / A);
%     VolMax =  sqrt(h(i) / A);
%     funVol = (h(i) - A*x.^2) .* y0 - y0.^3 * B/3;
%     aa = rho*2*int(funVol,VolMin,VolMax)
res(i) = rho * CalcV_all_( h(i) );
end

res = res - Ma - Mb;
% size(h)
% size(res)

plot(res,h,'b--');
xlabel('重物质量 Mb/kg');
ylabel('吃水线高度 h/m');
title('重物质量与吃水线关系曲线图');
% end

function res = CalcV_all_(H)
[A,B] = setValue();
% H = 0.0635;
syms x

```

```

y0 = sqrt( (H - A*x.^2)/B);
VolMin = vpa(-sqrt(H/A),4);
VolMax = vpa(sqrt(H/A),4);
funVol = (H-A*x.^2) .* y0 - y0.^3 * B/3;
res = 2*int(funVol,VolMin,VolMax);
end

```

%绘制船体优化后

```

function wLine = dispBoat(theta,z0)
[A, B, H] = setValue();
xc = 0;
yc = 0;
zc = H;
xr = sqrt(H/A);
yr = sqrt(H/B);
zr = H;
xUp = xr*0.5;
yUp = yr*0.5;
zUp = zr*0.5;
hold on;
xlabel('x');
ylabel('y');
zlabel('z');
[X, Y, Z] = ellipsoid(xc,yc,zc,xr,yr,zr,100);
Size = size(X);
colSize = Size(1);
z1 = Z;
z1(ceil(colSize/2) + 1:colSize, : ) = ones(size(z1(ceil(colSize/2) + 1:colSize, : ) ) ) * H;
mesh(X,Y,z1,z1*0.5);
axis equal
axis([-xr-xUp xr+xUp -yr-yUp yr+yUp 0 zr+zUp]);
x = ones(colSize,1)*linspace(-xUp*3,xUp*3,Size(2) );
y = ones(colSize,1)*linspace(-yUp*3,yUp*3,Size(2) );
% theta = i * pi / 180;

```

```

p = tan(theta);%水面方程斜率
if nargin == 1%default z0
    if p < 0
        z0 = H + p*sqrt(H/A);%水面方程截距
    else
        z0 = H - p*sqrt(H/A);
    end
end
z2 = p*x+z0;
fprintf('z0 = %.4f\n',z0);
wLine = surf(x,y',z2);
axis([-xr-xUp xr+xUp -yr-yUp yr+yUp 0 zr+zUp]);
title(theta*180/pi);
% pause(1);
End

```

```

function diffWaterlin()
%根据 theta 和 z0 绘制水面
filename = 'Waterline.xlsx';
T = readtable(filename);
z0 = T.z0;
Theta = T.theta*pi/180;
for i = 1:4
    hold on
    subplot(2,2,i);
    dispBoat( Theta(i), z0(i) );
end

```

```

% function z0 = CalcZ0(xLim)
%迭代求解积分的上下限 【x1 xMid x2】和 水面方程截距
%即可得到水面方程
clear
[A, B, H, volume,~,~,~,~,thetaL,thetaR] = setValue();
stepAngle = 20;

```

```

lenX = length(1:stepAngle:179);
z0 = zeros(lenX,1);
xLimData = zeros(lenX,3);
x1 = sqrt(H/A);
x2 = -sqrt(H/A);
xMid = sqrt(H/A);
Angle = [15 50 125 178]*pi/180;
for i = 4:4%enX
    theta = Angle(i);% 135*pi/180;% (5 + (i-1)*stepAngle)*pi/180;
    if i == 1
        xLim = xLimCOB(theta);
    else
        xLim = xLimCOB(theta,x1,xMid,x2);
    end
    p = tan(theta);
    %x1 xMid x2 分别为水面与
    x1 = xLim(1);
    xMid = xLim(2);
    x2 = xLim(3);
    if theta < thetaR
        %5. 船倾斜角度小与 theta_0，即吃水线与
        船地面相交
        z0(i) = -A*x1*x2;
    elseif theta < pi/2
        %4. 船倾斜角度大与 theta_0 小于 90°，即吃
        水线与船甲板相交
        z0(i) = H - p*xMid;
    elseif theta == pi/2
        %3. 船倾斜角度为 90°
        z0(i) = x2;
    elseif theta < thetaL
        %2. 船倾斜角度小与 theta_1 大于 90°，即吃水
        线与船甲板相交
        z0(i) = H - p*xMid;
    elseif theta < pi
        %1. 船倾斜角度小与 theta_0，即吃水线与船
        地面相交
        z0(i) = -A*x1*x2;
    end
end

```



```

xLimData(i, :) = [ x1 xMid x2];

fprintf('%0.6f-%0.6f-%0.6f\n',x1,xMid,x2);

end

function res = CalcVwater(theta,x1,xMid,x2,z0)
%计算过点(x0,0,H)的船体水下的体积
[A, B, H, ~,~,~,~,~,thetaL,thetaR] = setValue();
syms x
p = tan(theta);
y0 = sqrt( (H - A*x.^2) / B);
funVol_0 = y0 .* (H-A*x.^2) - B/3*y0.^3;%(tan_theta .* x - 2/3*A*x.^2 + z0 - H / 3) .* z0;
funVol_1 = y0 .* (p*x+z0-A*x.^2) - B/3*y0.^3;
funVol_2 = y0.*(H - p*x- z0);
if theta <= thetaR %5
    res = 2*int(funVol_1,x2,x1);
elseif theta < pi/2 %4
    res = 2*int(funVol_1,x2,xMid) + 2*int(funVol_0,xMid,sqrt(H/A) );
elseif theta == pi/2 %3
    res = 2*int(funVol_0,xMid,sqrt(H/A) );
elseif theta < thetaL %2
    res = 2*int(funVol_2,xMid,x1) + 2*int(funVol_0,x1,sqrt(H/A) );
else
    res = 2*int(funVol_0,-sqrt(H/A),x2)+2*int(funVol_2,x2,x1)+2*int(funVol_0,x1,sqrt(H/A) );
end
end

function Res = CalcV_all(H)
%计算船的总容积
[A,B] = setValue();

```

```

if nargin == 0
    [~,~,H] = setValue;
end
syms x
y0 = sqrt( (H - A*x.^2)/B);
VolMin = vpa(-sqrt(H/A),4);
VolMax = vpa(sqrt(H/A),4);
funVol = (H-A*x.^2) .* y0 - y0.^3 * B/3;
Res = 2*int(funVol,VolMin,VolMax);
End

function Mom = CalcMom(theta, COB, COM)
% 已知小船倾斜 theta 角度后的重心，浮心计算恢复力矩
rMon = COB - COM;
p = tan(theta);
Buo_val= volume * rho;%浮力 Buoyancy
lambda = Buo_val / (p^2+1);
Buo = [p 0 1]*lambda;
Mom = sum(cross(rMon,Buo) );
End

function xLim = Calc_xLim(theta, option, x0)
%when option == true Calc the thetaL and thetaR
%when option == false Calc the waterline x1 and x2
[A, ~, H, ~,~,~,~,thetaL,thetaR] = setValue();
% fprintf('x0 = %.4f\n',x0);
p = tan(theta);
if nargin < 2
    option = true;
else
    Delta = p^2-4*A*(p*x0 - H);
end
if option%Calc the thetaL and thetaR

```

```

if theta < pi/2
    xLim(1) = sqrt(H / A);
    xLim(2) = p / A - xLim(1);
else
    xLim(2) = -sqrt(H / A);
    xLim(1) = p / A - xLim(2);
end
else
    %Calc the waterline x1 and x2          1
    if theta > thetaL %thetaL~pi
        xLim(1) = x0;
        xLim(2) = p/A - xLim(1);
    elseif theta > pi/2 %pi/2~thetaL      2
        xLim(2) = x0;
        xLim(1) = (p+sqrt(Delta) ) / (2*A);
    elseif theta == pi/2 %pi/2          3
        xLim(1) = sqrt(H/A);
        xLim(2) = x0;
    elseif theta > thetaR %thetaR~pi/2    4
        xLim(1) = x0;
        xLim(2) = (p-sqrt(Delta) ) / (2*A);
    else                                %    5
        xLim(1) = x0;%sqrt(H / A);
        xLim(2) = p / A - xLim(1);
    end
end
% disp(xLim);
% (p-sqrt(Delta) ) / (2*A)
% (p+sqrt(Delta) ) / (2*A)
End

```

```

function Calc_wLine()
%求无倾斜时小船的吃水线
[~, ~, H, volume] = setValue();%导入小船模型参数

```

```

hMin = 0;%设置迭代区间
hMax = H;
tol = 10^-6;%精度
runtimes = 100;%限制迭代最大次数防止死循环
i = 0;
while i <= runtimes
    i = i+1;
    h = (hMin +hMax) / 2;
    res = CalcV_all(h);
    res = res - volume;
    if res > tol%积分区域过大减小 h 使积分区域减小
        hMax = h;
    elseif res < -tol
        hMin = h;
    else
        break;
    end
end
% fprintf('res = %.5f volume = %.5f\n', res+volume,volume);
fprintf('waterline = %.5fm \n', h);
end

function thetaR = Calc_thetaR()
%二分查找计算吃水线过点（sqrt(H/A),0,H）时的角度 thetaR
[A,~,H,volume] = setValue();
thetaMin = 0;
thetaMax = pi*57/180;
tol = 10^-6;%精度
runtimes = 100;%最大迭代次数
i = 0;
syms x;
while i <= runtimes
    i=i+1;
    thetaR = (thetaMin+thetaMax) / 2;

```

```

xLim = Calc_xLim(thetaR);
z0 = -A*xLim(1)*xLim(2);
res = CalcVwater(thetaR, xLim(1), sqrt(H/A), xLim(2), z0);
res = res - volume;
if res < -tol
    thetaMax = thetaR;%角度变小， res 变大
elseif res > tol
    thetaMin = thetaR;%角度变大， res 变小
else
    break;
end
end
end

```

```

function thetaL = Calc_thetaL()
%二分查找计算吃水线过点（-sqrt(H/A),0,H）时的角度 thetaL
[A,~,H,volume] = setValue();
thetaMin = pi*91/180;
thetaMax = pi;
tol = 10^-6;%精度
runtimes = 100;%最大迭代次数
i = 0;
syms x;
while i <= runtimes
    i=i+1;
    thetaL = (thetaMin+thetaMax) / 2;
    xLim = Calc_xLim(thetaL);
    z0 = -A*xLim(1)*xLim(2);
    res = CalcVwater(thetaL,xLim(1), -sqrt(H/A), xLim(2), z0);
    rho = real(vpa(res - volume, 4));
    if rho > tol
        thetaMin = thetaL;%角度变大， res 变大 if res > tol
    elseif rho < -tol
        thetaMax = thetaL;%角度变小， res 变小
    end
end

```

```

        else
            break;
        end
    end
end
end

% [A, B, H, volume,V_all,M_all,rho,g,thetaL,thetaR] = setValue();
% Mc = 0.321;
% Ma =0.11;
% Mb = 0.766;
% COMc_z = 0.08;
% COB_135 = [ 0.0607, 0, 0.09736];
% COG = [ 0 0 0.05277];
% M_all = Ma + Mb + Mc;
% Cz = 0.08;
% Ra = 0.005;
% Rb = 0.02;
% Zb = ( COG(3)*M_all - Cz*Mc - Ra*Ma - Rb*Ma) / ( Mb + Ma);
% fprintf('重物重心 Zb = %.5f\n', Zb);
mAndCOG();
function mAndCOG()
    Mc = 0.321;
    Ma =0.11;
    % Mb = 0.766;
    Zb = 0.0424;
    Za = 0.0224+0.02+0.25;
    Zc = 0.08;
    Mb = linspace(0,0.766,10);
    Size = size(Mb);
    Z_all = linspace(1,1,Size(2) );
    for i = 1: Size(2)
        Z_all(i) = (Ma*Za+Mb(i)*Zb+Mc*Zc)/(Mc+Ma+Mb(i));
    end
    hold on;

```

```

plot(Mb,Z_all,'r*');
plot(Mb,Z_all);
xlabel('重物质量 m/kg');
ylabel('重心 z 轴坐标/m');
title('重物质量与重心曲线关系图');
end

function COB = Calc_COB(theta,x1,xMid,x2,z0)
format short
p = tan(theta);
[A, B, H, volume,~,~,~,~,thetaL,thetaR] = setValue();
syms x;

y0 = sqrt( (H - A*x.^2) / B);

xbFun_0 = x.*(y0.*(H-A*x.^2)-B/3*y0.^3);
xbFun_1 = x.*(y0.*(p*x+z0-A*x.^2)-B/3*y0.^3);% 1/4*y0.^2 .* ( (p*x+z0).^2-Z.^2- 8*B*Z);
xbFun_2 = x.*(H-p*x-z0).*y0;
zbFun_0 = 1/2*y0 .* ( H.^2 - A^2*x.^4 - 2/3*A*B*x.^2.*y0.^2 - 1/5*B^2*y0.^4 );
zbFun_1 = 1/2*y0 .* ( (p*x+z0).^2 - A^2*x.^4 - 2/3*A*B*x.^2.*y0.^2 - 1/5*B^2*y0.^4 );
zbFun_2 = 1/2*y0 .* (H^2-(p*x+z0).^2);

if theta <
thetaR

%5

xb = 2*int(xbFun_1,x2,x1);
zb = 2*int(zbFun_1,x2,x1);

elseif theta <
pi/2

%4

xb = 2*int(xbFun_1,x2,xMid)+2*int(xbFun_0,xMid,sqrt(H/A) );
zb = 2*int(zbFun_1,x2,xMid)+2*int(zbFun_0,xMid,sqrt(H/A) );

```

```

elseif theta ==
pi/2
    %3
    xb = 2*int(xbFun_0,xMid,sqrt(H/A) );
    zb = 2*int(zbFun_0,xMid,sqrt(H/A) );
elseif theta <
thetaL
    %2
    xb = 2*int(xbFun_2,xMid,x1)+2*int(xbFun_0,x1,sqrt(H/A) );
    zb = 2*int(zbFun_2,xMid,x1)+2*int(zbFun_0,x1,sqrt(H/A) );
else
    %1
    xb = 2*int(xbFun_0,-sqrt(H/A),x2 )+2*int(xbFun_2,x2,x1)+2*int(xbFun_0,x1,sqrt(H/A) );
    zb = 2*int(zbFun_0,-sqrt(H/A),x2 )+2*int(zbFun_2,x2,x1)+2*int(zbFun_0,x1,sqrt(H/A) );
end
fprintf('%0.6f-%0.6f  %0.6f-%0.6f-%0.6f\n',xb,zb,x1,xMid,x2);
COB = vpa([xb 0 zb]/volume , 6);
end

```