

机器人学院项目报告



项 目 题 目： MADE小船

学 院 名 称： 机器人学院

专 业： 网络工程

班 级： 网络182

姓 名： 徐炜涛 学 号 18401190125

姓 名： 罗文荣 学 号 18404130277

指 导 教 师： 汪保

完成日期： 2019 年 11 月 10 日

目录

摘要¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨3

前言¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨4

一、项目背景¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨5

1.1设计理由（设计背景）¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨5

1.2项目要求介绍¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨5

二、小船材料与制造操作过程¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨5

2.1小船材料简介¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨5

2.2小船工艺与设计¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨5

三、小船的推导思路与代码¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨6

3.1推导模型¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨7

3.1.1整体思路¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨7

3.1.2重心、浮心¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨8

3.1.3四个阶段¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨8

3.1.3.1第一阶段0-1¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨8

3.1.3.2第二阶段1-90¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨10

3.1.3.3第三阶段90-2¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨11

3.1.3.3第四阶段2-180¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨12

3.2代码设计¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨13

4.1题目模型解答¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨14

四、结论与感悟¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨18

参考文献¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨¨19

MADE小船设计与实现

摘要：船舶在海上行驶会遇到海浪导致船舶飘忽不定，因此在设计船舶的时会将船设计地有较好地稳定性，船舶的稳定性是指船舶在有限的作用下不会倾覆，倾侧性消失后能恢复到正常状态的能力。

决定船舶稳定性的一大关键元素就是复原力矩， 复原力矩是指船舶受外力作用倾斜时由于重力和浮力的作用点不在同一铅垂线上所产生的使船恢复正态的恢复力矩当重心和浮心不在同一垂线上，重力和浮力形成力偶，促使船回到初始平衡位置，此时的复原力矩为正值。当重力和浮力形成的力偶，促使船继续横倾，此时的复原力矩为负值，当复原力矩达到负值的一瞬间就是最大AVS消失角。

对于船舶稳定性我们可以通过设计船的模型(船体方程)和改变船的重心来达到我们需要的最大恢复力矩，使船达到我们的要求。

模型设计与代码设计：本文采用了先推导公再结合代码迭代设计船体最后进行制作。

文中主要就对小船的相关结构设计与恢复力矩计算做了相关的综述。

关键词：微积分；船舶稳定性；恢复力矩；AVS稳定消失角

**前言**

为了让我们数学更贴近生活也为了增强我们对微积分认识，学院布置了小船项目，小船项目是通过教师给出一定要求，里面包含物理、数学、编程等一些列的要求，整合多种工具最后完成小船，小船项目要求分为几种测试，静平衡测试：船满载时放入水中，能够漂浮，且甲板平面与标识线（水面）平行。稳定性测试： 主要观测稳性消失角是否达到 135±10°。最大恢复力矩测试：根据 135°倾角时围绕浮心的转动惯量 J（单位用 kg·m2)和测量所得周期 T(单位用 s),计算得到该船的最大恢复力矩值，大于或等于 0.2 N·m 的该项合格，小于 0.2 N·m 的不合格。行进性能测试：单位时间内速度最大。

**一、项目背景**

**1.1设计背景**

为了加强我们对物理、微积分和编程与现实的联系，学院开展了小船项目：目的是培养综合运用所学的基础理论，基本知识和技能，提高分析解决实际问题的能力。提高实际工作能力和协作能力：如调查研究、查阅文献和收集资料并进行分析的能力；制定设计或实验方案的能力；设计、计算和绘图能力；总结提高撰写论文的能力。

**1.2项目要求介绍**

小船项目要求做出的小船满足以下要求：船满载时放入水中，能够漂浮，且甲板平面与标识线（水面）平行。主要观测稳性消失角是达到 135±10°。最大恢复力矩测试：在 135°倾角时围绕浮心的恢复力矩大于等于 0.2 N·m 的该项合格。行进性能测试：给予相同的力在一定距离内行驶时间越短得分越高。

**二、小船材料与制造操作过程**

**2.１小船材料简介**

学院提供硬质板、铁质重物、铝质桅杆、塑料薄膜、胶带、胶水、剪刀、美工刀。如图2-1：



图 2-1

**2.2小船工艺与设计**

小船船体设计需要方程，方程设计选择的是二次方程，因为二次方程有着良好的对称性，并且二元二次方程更容易计算，虽然最终计算是交给计算机计算，但是由于二次方程本身简单，我们能人为化简积分，可以较大减少计算机计算时间。然后我们可以对方程进行计算，通过计算以后得出相应数据后就可以设计船体，还可以对船体进行预留装饰位置，然后使用CAD画出平面切割图2-2，也可以使用solidwork建立3D图形模拟，如2-3图：(这里的图形为最终完成的图形设计)

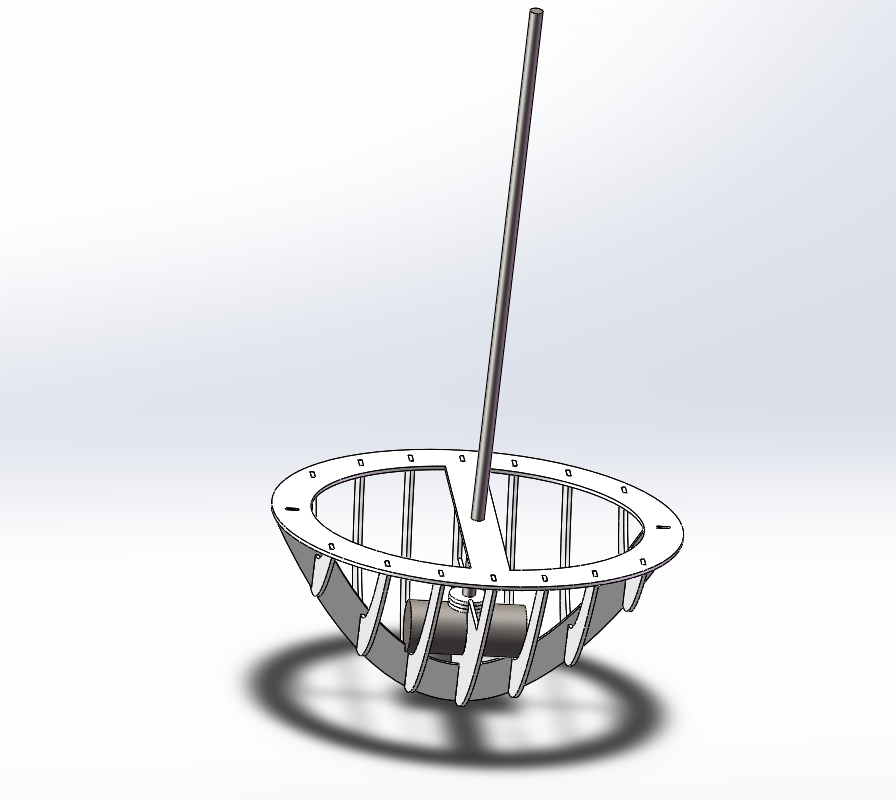
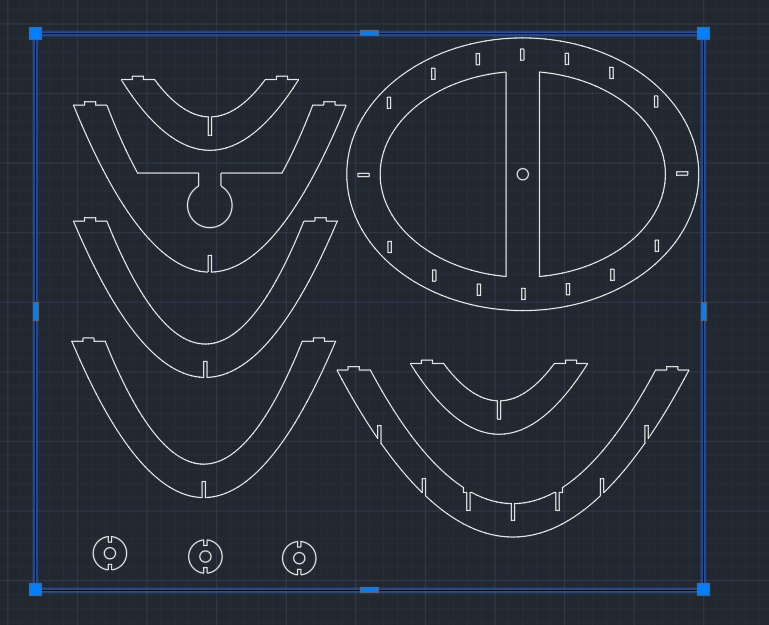


图2-2 图2-3

然后让老师审核切割图纸，在图纸不存在切割问题后就可以交给老师切割，切割过程不参与。最后组装、做好防水处理与装饰。

**三、小船方程推导思路与代码**

**3.1推导模型**

**3.1.1整体思路**

首先船体是三维立体图形，我们以（0，0，0）为原点这个点也是船的最低点，我们的目标是使得制造出来的船稳定消失角处于135±10°。所以这是最低的限定条件，由于物理知识我们可以知道船只要船在水中浮起，浮力永远等于重力（不进水的情况下）这样我们得到了一个重要的限定方程。①，因此无论后面吃水线如何变化都处于这个限定方程里面。

**整体思路的思维导图3-1：**

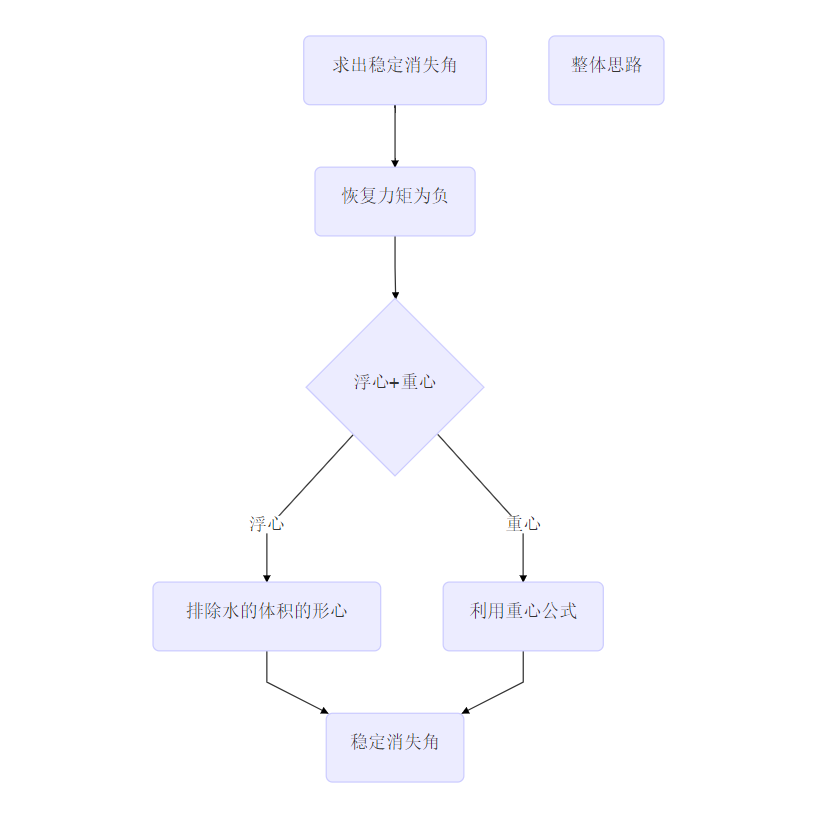


图 3-1

**3.1.2重心、浮心**

通过思维导图我们得知为了求出稳定消失角需要求出重心和排水的形心，根据测量我们得知材料的总重，我们可以通过粗略估计所需的船体重量就可以算出总重量，得出总体总量就可以利用①得出排水体积，我们知道我们的方程为我们可以粗略设定A、B和船高H，然后可以利用。

**重心方程求解：**

②(,,)

就可以得出船体的重心位置。

接下来就是浮心位置：浮心为排开水体积的形心，由于我们选择的是对称性良好的二元二次方程，所以我们在计算吃水线的时候我们可以选择对一个象限进行积分然后推广到其余四个象限。（由于我们是按照Y轴旋转的，所以可以忽略Y轴的积分，所以此处如果选择二重积分也是可以的，但是三重积分更为直观）

**浮心求解：**

我们需要得到排开水的体积，可以利用我们在前面的到重要限定方程使用①推导出来积分区域为：（，， ）

通过程序二分迭代得到吃水线，得到吃水线以后我们可以使用形心定义求得他在静水中的浮心，分别对x、z轴使用③和④就能得到浮心坐标

**3.1.3四个阶段**

求得静止时刻的浮心以后我们开始求解从0-180之间的复原力矩，求得复原力矩=复原力臂x力，力为重力，力臂我们可以用浮心和重心求得力臂长度。我们选择以船体为固定中心，让水面旋转，由于绕着Y轴旋转，我们将平立体船和水面看成二维图像，这样更直观，并且这个时候我们的模型可以简化成为一个平面与直线，求解这两个方程的解就能得到他们的交点类似图中的两个交点，我们知道排水体积不变，那么我们设从0°开始往上增加，每增加1°就能得到相对应的截距C这样，就是已知方程了，再联立得到下面图3-1-4

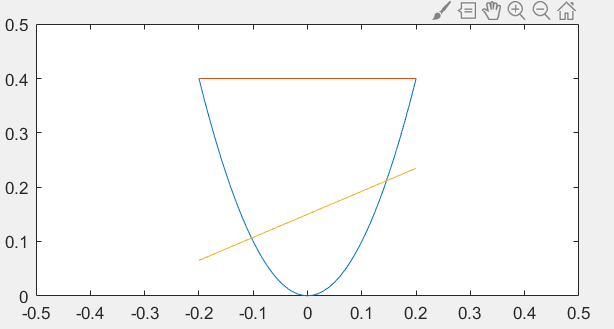


图3-1-4

而联立方程以后我们可得

方程的解为与，当船体到达吃水线附近的极限值也是约束条件与

这个过程需要分成4个阶段

这4个阶段的思维导图3-1-5：

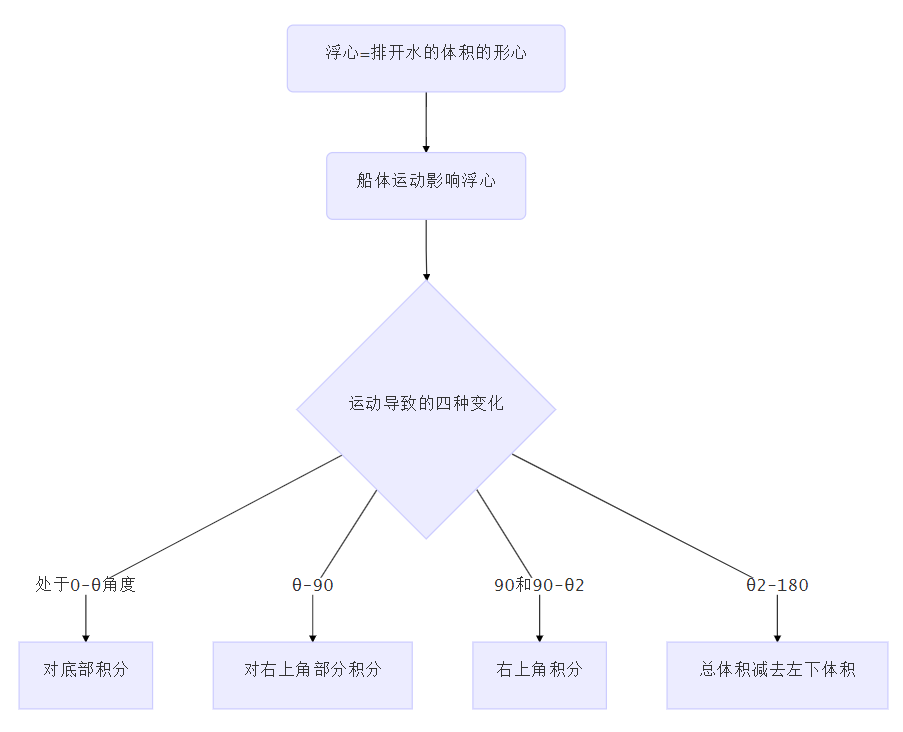
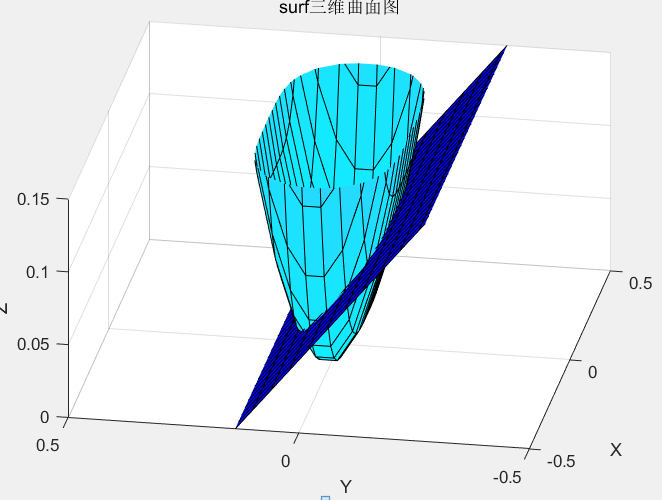
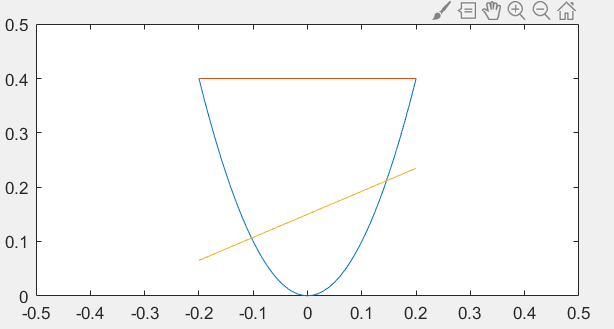


图 3-1-5

**3.1.3.1第一阶段0-****1**

阶段二维图3-1-3-1-1与三维图3-1-3-2-2



**图3-1-3-1-1 图3-1-3-1-2**

当角度小于的时候的积分区域为Z轴的上下限为积分区域为：

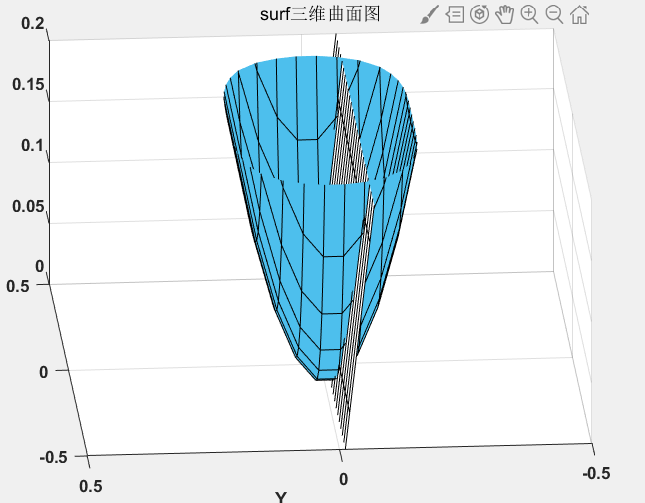
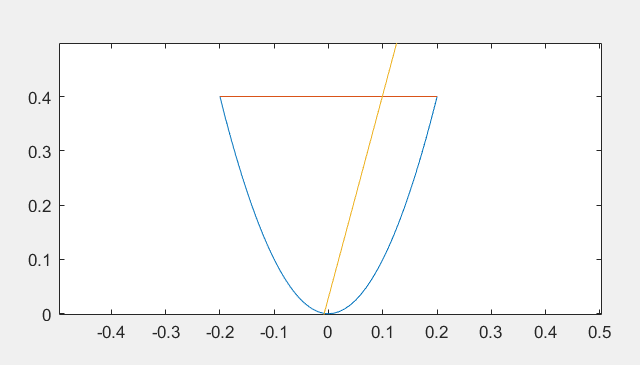
（、）

 （x的积分区域为）

然后再按照上面求浮心的方法，求得此时的浮心即可。

**3.1.3.2第二阶段****1-90**

阶段二维**图3-1-3-2-1** 与三维图3-1-3-2-2



**图3-1-3-2-1 图3-1-3-2-2**

当x大于小于90的时候积分区域为两部分：

第一部分： （x的积分区域为）

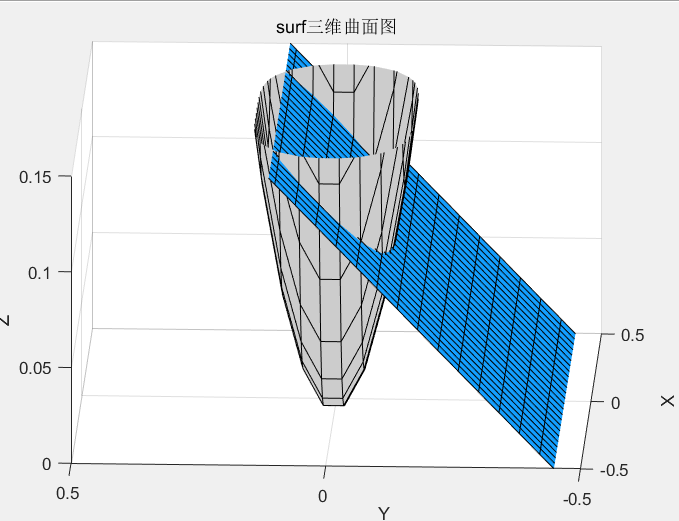
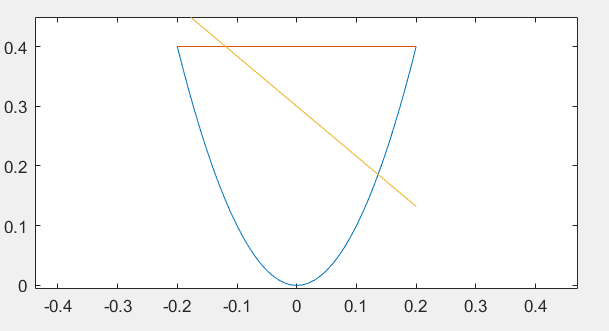
第二部分： （x的积分区域为

两部分积分加得到此时总积分再利用上面求浮心方法求得此时浮心

**3.1.3.3第二阶段90-****2**

**此时可以特判一次90°的吃水线，也可以跳过因为影响不大。**

阶段二维图**3-1-3-3-1**与三维图**3-1-3-3-2**



**图3-1-3-3-1 图3-1-3-3-2**

当x大于90小于2的时候积分区域为两部分：

第一部分： （x的积分区域为）

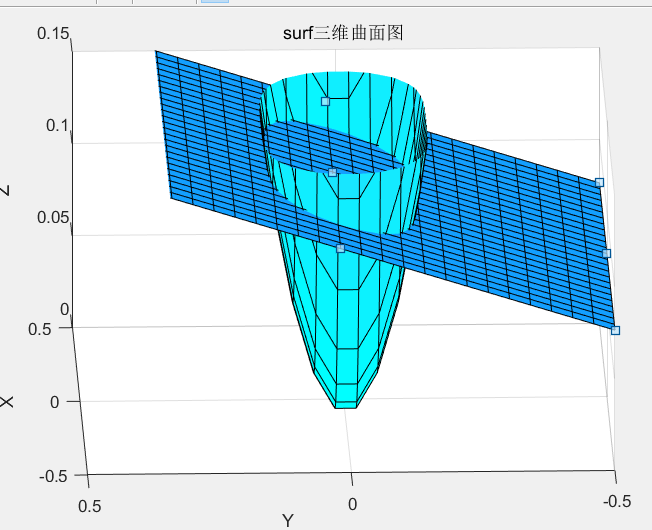
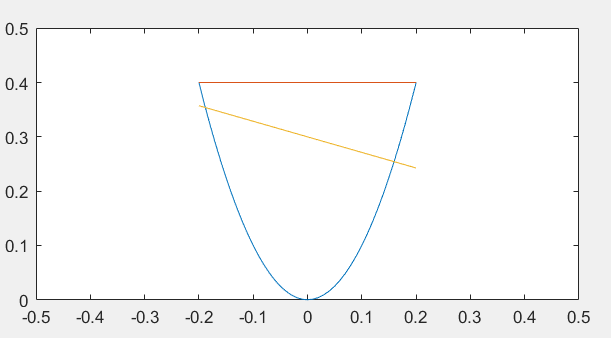
第二部分： （x的积分区域为

**两部分体积相加再利用上面求浮心方法求得浮心即可。**

**3.1.3.4第二阶段****2-180**

**这个阶段有两种方法求得体积，一种是求得上面部分的三部分积分，另外一种就是求得总体积然后减去下半部分的积分体积也能达到，我选择的是减法，会少一个积分。**

阶段二维**图3-1-3-4-1**与三维图**3-1-3-4-2**



**图3-1-3-4-1 图3-1-3-4-2**

当x大于2小于180的时候积分区域为两部分：

总体部分：（x的积分区域为）

减去下部分（x的区域为）

然后按照上面的方法求浮心。

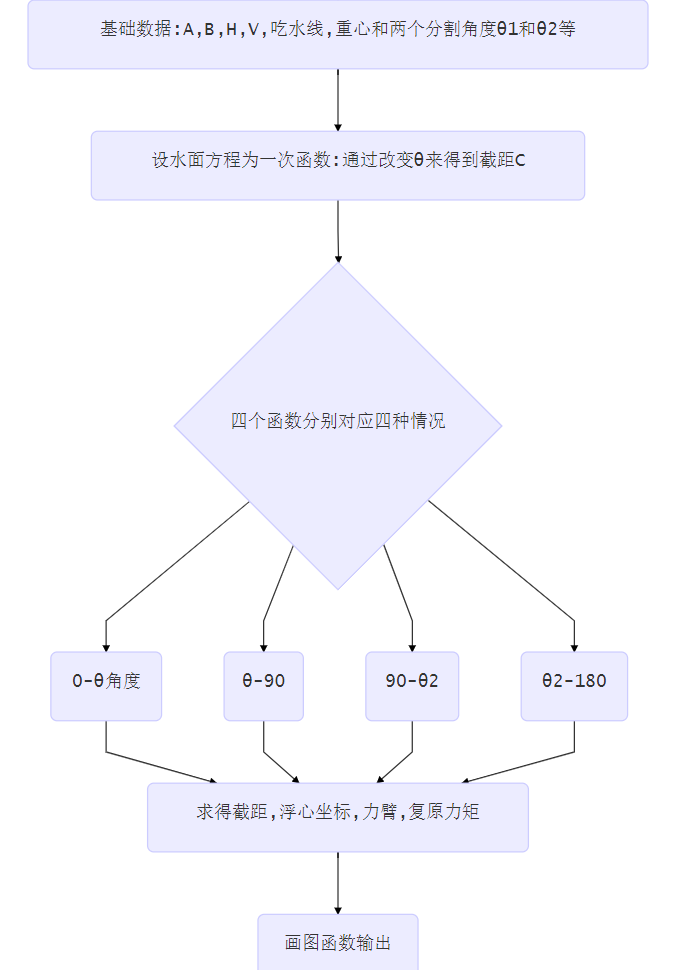
重心不变，并且我们有了个个阶段的浮心，直接利用两直线之间的距离就能求得力臂长度，复原力矩=力臂长度**x重力**

这样，每个阶段的复原力矩都能算出来。

**3.2代码设计**

**3.2.1总体设计思路**

代码的设计思路其实是与推导思路紧紧联系的，因为方程推出来之后代码只能进行计算，起到一个简化运算的效果，所以代码的思维导图为一个主函数控制基础数据然后下面的分支以下为思维导图**图3-2-1**



**图3-2-1**

**4.1题目模型解答**

**前提：完成代码和推导过程以后就可以确定方程确定的船体方程与水面方程为****其中实际小船=0.15m,A、B系数分别为10和6，根据方程可以得出实际长宽分别为0.31.6m、0.245m。**

**而水面方程可通过改变**的角度来得到截距C然后求解。

1：求出小船底部曲面方程及船高

小船方程：

2：求出龙骨方程、截面方程及甲板方程

龙骨方程：

截面方程：

甲板方程·：

3：求出重心及小船向 x 轴倾斜 135°（船体封闭）时的浮心及复原力矩以下的坐标均为笛卡尔坐标系，分为原点坐标为（0，0，0）船头方向为Y轴，船侧方向为X,桅杆正方向为Z轴。小船在倾斜的时候是沿着y轴倾斜的，所以我们可以直接使用y=0，并且由于我们的船是高度对称的小船，并且我们的A、B已经确定，它处于范围在90°-2°，此时模型为上面模型的第三种：

这时候的积分分为两段左边的两段分别为x的上下限：

 和使用上面的模型积分得出此时得到

浮心位置为（0.059398，0，0.133374）

重心位置：

重物位置=6cm（船底的距离）+2cm（重物宽度）

桅杆位置=重物z轴位置+桅杆长度/2

对船的薄板积分，算出重心z轴带入重心公式

重心公式;

(,,)

得到：（0，0，0.085）

力臂长度：利用两直线之间的距离公式

方向由角度决定。

力臂长度：0.001123m

计算得到恢复力矩：-0.012549N/m

4:求吃水线

已知的估算排水体积：1.14kg

船的方程，进行三重积分和二分迭代得到水平截距的吃水线：0.074975m

5：画数量关系图

5.1：画出重物质量与吃水线关系的曲线图4-5-1

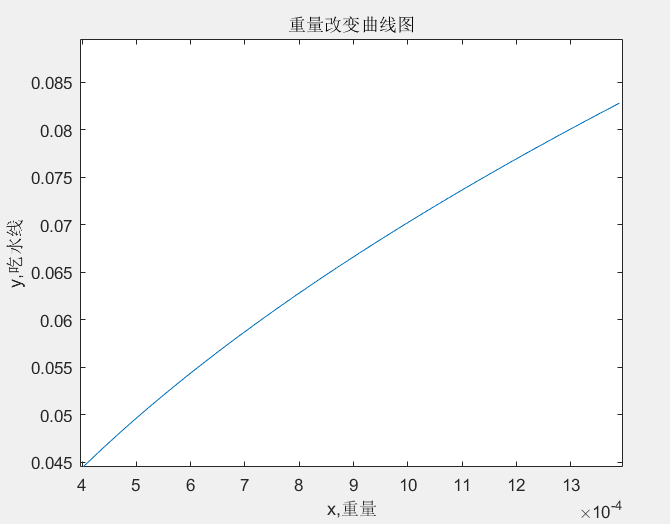


图4-5-1

5.2：画出船长、宽、高分别与复原力矩关系的曲线图

船体在0-180°的复原力矩图4-5-2：

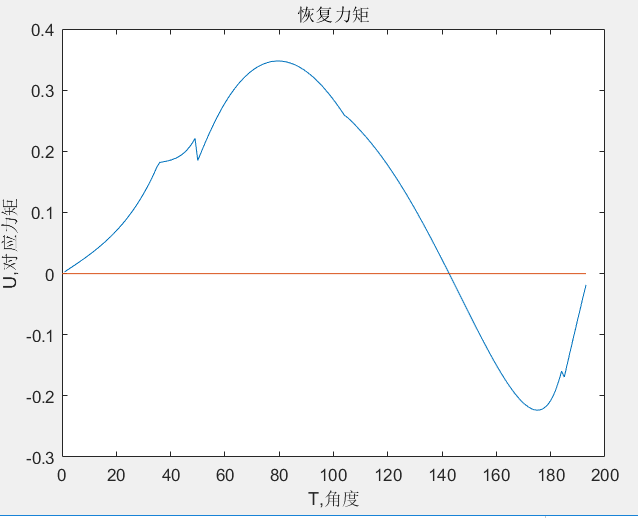


图4-5-2

5.3：画出重物质量与重心关系的曲线图4-5-3（曲线图的纵横坐标表示重心坐标，曲线上的点的）

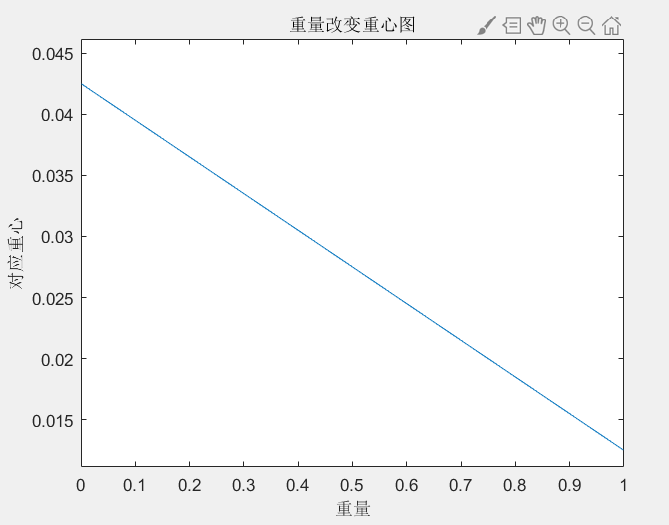


图4-5-3

**四、结论与感悟**

通过计算我们的小船在测试中达到了比较良好的效果，精准是吃水线、125°的AVS消失角度、较好的复原力矩、不错的速度，这次项目中由于选择了简单的二元二次函数，所以在计算的时候一开始使用了手算（matlab在三重积分中变现不佳造成的，速度巨慢，跑一个180的AVS角度需要20分钟）然后手动积分成一重积分以后速度提升了好多好多（这已经不能用快来说了，这是非常快），当然简单的二元二次函数不仅在我们化简中提供了很多方便，在求根的时候也是可以使用求根公式，而不用选择迭代求解（迭代求解虽然误差不大，但终归是误差），好的程序能让题目更快解决，但是一开始的方向很重要，在一开始的时候我们选择了以直代曲，然后在AVS哪里求不出浮心，因此耽搁了很久，最后发现曲面才是最简单的，因为软件可以算，所以，最重要的是数学思维，方向对了，那就离成功不远了。

**[[1]](#endnote-1)**

1. # 参考文献

   * 1. James Stewart. Calculus, 2014.

   附件：代码

   主函数：

   %/\*-----小船项目-------\*/

   %/\*-作者：徐炜涛-------\*/

   %/\*-功能：找寻复原力矩-\*/

   %/\*-时间：2019-11-6--\*/

   %/--------------------\*/

   %/\*-------首先确定方程为二次方程H=A\*x^2+B\*y^2---------\*/

   format long;

   syms x y;

   V\_all=0.002;%此处为船的体积m3--是指当船满排水的时候的排水体积

   %/\*-----最简方程 A B 为用来反推的数据，H为船高 h为吃水线----\*/

   F\_boat=@(x,y,H)A\*x^2+B\*y^2-H;

   A=10;%系数

   B=6;

   H=0.15;

   m\_L=0.084; %重物长度为后面cad画图需要的参数

   %由于良好的对称性，所以只需要考虑gz的位置

   m\_iron=0.8234;%重物质量

   m\_mast=0.1;%桅杆质量

   m\_focus\_z=0.01+0.02;%龙骨高度+重物宽 重物z轴重心

   m\_boat\_z=0.05;%船的重心

   m\_mast\_z=0.04+0.01+0.25;%桅杆重心位置

   %重心

   center=calculate\_center(m\_focus\_z,m\_boat\_z,m\_mast\_z);

   m\_total\_z=(m\_focus\_z\*0.8234+m\_boat\_z\*0.25+m\_mast\_z\*0.1);%此处的船体重心

   %Art=get\_add\_tie\_art(A,B,H);%画出重量增加的图

   V\_boat=0.0012;%此处为船的排除体积m3--实际排水体积（船的大致重量都可以测出）

   %求浮心公式

   %求出的浮心会在下面进行输出

   %while(A<20)

   %调用the\_main函数：

   %/\*--每个the\_main函数都代表个每个阶段求得的复原力矩--\*/

   %/\*--其中分为几个阶段：--------------\*/

   %/\*--第一阶段：0-左上方交点----------\*/

   %/\*--第二阶段：左上方交点-90---------\*/

   %/\*--第三阶段：90-右上方交点---------\*/

   %/\*--第四阶段：接近180的吃水---------\*/

   %/\*--

   %/\*--每个

   X=the\_main2(A,B,H,V\_boat,m\_total\_z);

   Y=the\_main3(A,B,H,V\_boat,m\_total\_z);

   %Z=the\_main4(A,B,H,V\_boat,m\_total\_z);

   F=the\_main5(A,B,H,V\_boat,m\_total\_z);

   T=the\_main6(A,B,H,V\_boat,m\_total\_z);

   %fprintf("%d\n",K);

   %把结果储存到矩阵中然后再制图

   % X=[X Y F];

   %end

   f\_water\_line=the\_0\_to\_xita(A,B,H,V\_boat);%吃水线

   fun\_float=@(x) (f\_water\_line^2-A^2\*x^4-(f\_water\_line^2-2\*A\*x^2\*f\_water\_line+A^2\*x^4)/5)\*(f\_water\_line-A\*x^2);

   float\_z=quadv(fun\_float,0,(f\_water\_line/A)^0.5)/0.0012;

   flaot\_z=float\_z\*2/sqrt(B);

   %画图函数

   % F=water\_line\_art(A,B,H,f\_water\_line);

   %输出

   % fprintf("吃水线为%d\n",f\_water\_line);

   % fprintf("船体重心位置(0,0,%d)\n",m\_total\_z);

   % fprintf("船体浮心位置为(0,0,%d)\n",float\_z);

   % fprintf("%f",F\_obtain);

   力臂函数：

   %/\*----求力臂长度----\*/

   %/\*----方向由角度定--\*/

   function L=get\_the\_L(float\_avs\_Z,float\_avs\_X,m\_mast\_z,z)

   L=(float\_avs\_Z-m\_mast\_z)\*sin(z)+float\_avs\_X\*cos(z);

   End

   主函数0-西塔角度

   %0-西塔角度的力矩和浮心

   %求取函数第一步找到西塔角再以角度为分界线无限逼近

   %每提升一个角度就会计算出一个截距值

   function F=the\_main2(A,B,H,V\_boat,m\_total\_z)

   X=[];

   theat=obtain\_theat(H,A,B,V\_boat);%1:找到角度

   i=1;

   while(i<100)

   z=i/180\*pi;

   if(z>theat)

   break;

   end

   %fprintf("这是第%f个i",i);

   %/\*------得到截距C----\*/

   %/\*------吊塔---------\*/

   %/\*------积分上下限---\*/

   %/\*------求取浮心坐标--\*/

   %/\*------力臂长度-----\*/

   %/\*------力矩大小-----\*/

   C=get\_intercept(A,B,H,V\_boat,z);

   b=(tan(z)^2+4\*A\*C)^0.5;

   x\_min=(tan(z)-b)/(2\*A);

   x\_max=(tan(z)+b)/(2\*A);

   float\_avs\_Z=float\_Z(A,B,H,V\_boat,z,x\_min,x\_max);

   float\_avs\_X=float\_X(A,B,H,V\_boat,z,x\_min,x\_max);

   Restoring\_L=get\_the\_L(float\_avs\_Z,float\_avs\_X,m\_total\_z,z);

   Restoring\_data=Restoring\_L\*11.7600000000000;

   X=[X Restoring\_data];

   i=i+1;

   fprintf("这是第%d:力臂%f 力矩%f\n",i-1,Restoring\_L,Restoring\_data);

   end

   F=X;

   End

   西塔到90：

   %西塔-90的力矩和浮心

   %/\*----此处的积分为两部分----\*/

   %/\*----积分区域为右下和右上--\*/

   %/\*----注意积分上下限-------\*/

   %/\*----其余原理和上一个函数一样-\*/

   function F=the\_main3(A,B,H,V\_boat,m\_mast\_z)

   X=[];

   theat=obtain\_theat(H,A,B,V\_boat);

   i=theat\*180/pi+1;

   while(i<90)

   %fprintf("这是第%f个i",i);

   %/\*------得到截距C----\*/

   %/\*------吊塔---------\*/

   %/\*------积分上下限---\*/

   %/\*------求取浮心坐标--\*/

   %/\*------力臂长度-----\*/

   %/\*------力矩大小-----\*/

   z=i/180\*pi;

   C=get\_intercept(A,B,H,V\_boat,z);

   float\_avs\_Z=float\_Z\_45(A,B,H,V\_boat,z);

   float\_avs\_X=float\_X\_45(A,B,H,V\_boat,z);

   %fprintf("浮心坐标为（%f，0,%f)\n",float\_avs\_X,float\_avs\_Z);

   Restoring\_L=get\_the\_L(float\_avs\_Z,float\_avs\_X,m\_mast\_z,z);

   Restoring\_data=Restoring\_L\*11.7600000000000;

   X=[X Restoring\_data];

   fprintf("这是第%d :力臂%f 力矩%f\n",i,Restoring\_L,Restoring\_data);

   i=i+1;

   end

   F=X;

   end

   90-西塔2：

   %/\*----此处的积分为两部分----\*/

   %/\*----积分区域为右下和右上--\*/

   %/\*----注意积分上下限-------\*/

   %/\*----其余原理和上一个函数一样-\*/

   function F=the\_main4(A,B,H,V\_boat,m\_total\_z)

   X=[];

   theat=obtain\_theat(H,A,B,V\_boat);

   j=theat\*180/pi+1;

   i=91;

   while(i<145)

   z=i/180\*pi;

   %fprintf("这是第%f个i\n",i);

   c=get\_intercept\_90\_180(A,B,H,V\_boat,z);

   %fprintf("%f\n",c);

   float\_avs\_Z=float\_Z\_90\_180(A,B,H,V\_boat,z);

   float\_avs\_X=float\_X\_90\_180(A,B,H,V\_boat,z);

   %fprintf("浮心坐标为（%f，0,%f)\n",float\_avs\_X,float\_avs\_Z) ;

   Restoring\_L=get\_the\_L(float\_avs\_Z,float\_avs\_X,m\_total\_z,z);

   Restoring\_data=Restoring\_L\*11.7600000000000;

   X=[X Restoring\_data];

   fprintf("这是第%d :力臂%f 力矩%f\n",i,Restoring\_L,Restoring\_data);

   i=i+1;

   end

   F=X;

   End

   西塔2-180

   %/\*----此处的积分为两部分----\*/

   %/\*----积分区域为总-右下--\*/

   %/\*----注意积分上下限-------\*/

   %/\*----其余原理和上一个函数一样-\*/

   function F=the\_main6(A,B,H,V\_boat,m\_total\_z)

   X=[];

   theat=max\_theat\_\_(H,A,B,V\_boat);

   j=theat\*180/pi;

   i=j;

   while(i<180)

   z=i/180\*pi;

   %fprintf("这是第%f个i\n",i);

   c=get\_intercept\_last(A,B,H,V\_boat,z);

   %fprintf("%f\n",c);

   float\_avs\_Z=float\_Z\_last(A,B,H,z,c);

   float\_avs\_X=float\_X\_last(A,B,H,z,c);

   %fprintf("浮心坐标为（%f，0,%f)\n",float\_avs\_X,float\_avs\_Z) ;

   Restoring\_L=get\_the\_L(float\_avs\_Z,float\_avs\_X,m\_total\_z,z);

   Restoring\_data=Restoring\_L\*11.7600000000000;

   X=[X Restoring\_data];

   % fprintf("这是第%d :力臂%f 力矩%f\n",i,Restoring\_L,Restoring\_data);

   i=i+1;

   end

   F=X;

   % W=0:0.1:0.35;

   % S=10\*W.^2;

   % plot(W,S);

   % hold on;

   % syms x;

   % R=tan(130/180\*pi)\*x+c;

   % plot(W,R);

   %fprintf("%f\n",j);

   end

   获取每一次得到的截距C

   function the\_theta=obtain\_theat(H,A,B,V\_boat)

   i=1;

   while(i<90)

   theta=i/180\*pi;

   %fprintf("这是第%f角度\n",i);

   c=H-tan(theta)\*(H/A)^0.5;

   F=@(x)(tan(theta)\*x+c-H/3-2\*A\*x^2/3)\*sqrt(H-A\*x^2);

   %2\*(H-A\*x.^2)^1.5/3+tan(theta)\*(x-sqrt(H/A)).\*sqrt(H-A\*x.^2);

   x\_min=(tan(theta)-(tan(theta)^2-4\*A\*c)^0.5)/(2\*A);

   x\_max=(H/A)^0.5;

   V\_temp=quadv(F,x\_min,x\_max);

   if(V\_temp<V\_boat)

   break;

   end

   i=i+1;

   end

   the\_theta=i/180\*pi;

   end

   得到西塔角2

   %/\*----得到最大的倾斜角度----\*/

   %/\*----总积分-下积分得到----\*/

   function F=max\_theat\_\_(A,B,H,V\_boat)

   i=91;

   cof=4/(3\*sqrt(B));%常系数

   while(i<180)

   theta=i/180\*pi;

   %fprintf("%dth calculate:",i);%

   c=H+tan(theta)\*sqrt(H/A);

   d=tan(theta)^2+4\*A\*c;

   %fprintf("d is %f,",d);%

   fun=@(x) (H-tan(theta)\*x+c)\*(tan(theta)\*x+c-A\*x^2)^0.5;

   xmin=-sqrt(H/A);

   xmax=(tan(theta)+sqrt(d))/(2\*A);

   volumn=quadv(fun,xmin,xmax);

   volumn=volumn\*cof;

   Fun=@(x) (H-A\*x.^2).^1.5;

   Cof=8/(3\*sqrt(B));%

   V=quadv(Fun,xmax,-xmin);

   V=V\*Cof;

   %fprintf("V is %f,v水上 is %f,",V,volumn);%测试输出

   volumn=V-volumn;

   %fprintf("volumn is %f\n",volumn);%测试输出

   if(volumn<V\_boat)

   break;

   end

   i=i+1;

   end

   F=theta;

   end

   第一阶段得到函数：

   %/\*-----固定积分上下限-------\*/

   %/\*-----得到了角度以后的计算-\*/

   %/\*-----迭代求C-------------\*/

   function F=get\_intercept(A,B,H,V\_boat,z)

   syms x y;

   c\_min=0;

   c\_max=H-tan(z)\*(H/A)^0.5;

   c\_center=(c\_min+c\_max)/2;

   j=1;

   while(j<100)

   fun=@(x)(tan(z)\*x+c\_center-H/3-2\*A\*x^2/3)\*(H-A\*x^2)^0.5;

   c\_x1=(tan(z)-(tan(z)^2-4\*A\*(tan(z)\*(H/A)^0.5-H))^0.5)/(2\*A);

   c\_x2=(tan(z)+(tan(z)^2-4\*A\*(tan(z)\*(H/A)^0.5-H))^0.5)/(2\*A);

   V\_temp=quadv(fun,c\_x1,c\_x2);

   V\_temp=V\_temp\*2/B^0.5;

   if(V\_temp>V\_boat)

   if(c\_center<c\_max)

   c\_max=c\_center;

   end

   end

   if(V\_temp<V\_boat)

   if(c\_center>c\_min)

   c\_min=c\_center;

   end

   end

   c\_center=(c\_min+c\_max)/2;

   j=j+1;

   end

   F=c\_center;

   End

   第一阶段X浮心：

   %/\*-----确定积分上下限此处为右下积分----\*/

   %/\*-----x的浮心位置坐标直接用积分体积比例表示--\*/

   %/\*-----求出体积质量后直接比总重量----\*/

   function obtain\_avs\_X=float\_X(A,B,H,V\_boat,z,x\_min,x\_max)

   C=get\_intercept(A,B,H,V\_boat,z);

   %fprintf("这里的C是%f",C)

   fun=@(x) x\*((tan(z)\*x+C-H/3-2\*A\*x^2/3)\*sqrt(H-A\*x^2));

   V\_temp=quadv(fun,x\_min,x\_max);

   V\_temp=V\_temp\*2/B^0.5;

   obtain\_avs\_X=V\_temp/V\_boat;

   %fprintf("体积%f\n",V\_temp);

   End

   第一阶段Z浮心：

   %/\*-----确定积分上下限此处为右下积分----\*/

   %/\*-----z的浮心位置坐标直接用积分体积比例表示--\*/

   %/\*-----求出体积质量后直接比总重量----\*/

   function obtain\_avs\_Z=float\_Z(A,B,H,V\_boat,z,x\_min,x\_max)

   C=get\_intercept(A,B,H,V\_boat,z);

   fun=@(x) ((tan(z)\*x+C)^2-A^2\*x^4-2\*A\*x^2/3\*(H-A\*x^2)-(H-A\*x^2)^2/5)\*(H-A\*x^2)^0.5;

   %(H^2-A^2\*x^4)\*sqrt(H-A\*x^2)-2\*A\*x^2.\*(H-A\*x^2)^1.5/3-(H-A\*x^2)^2.5/5;

   V\_temp=quadv(fun,x\_min,x\_max);

   V\_temp=V\_temp\*2/B^0.5;

   obtain\_avs\_Z=V\_temp/0.0012;

   %fprintf("Z的坐标%f\n",obtain\_avs\_Z);

   end

   第二阶段：

   function F=get\_intercept\_45(A,B,H,V\_boat,z)

   c\_min=H-tan(z)\*(H/A)^0.5;

   c\_max=100;%或0.15；

   c\_center=(c\_min+c\_max)/2;

   i=1;

   while(i<100)

   d=(tan(z)^2+4\*A\*c\_center)^0.5;

   x\_min=(tan(z)-d)/(2\*A);

   x\_max=(H-c\_center)/tan(z);

   x\_r=(H/A)^0.5;

   %左部分

   fun1=@(x) 2/3\*(tan(z)\*x+c\_center-A\*x^2)^1.5;

   Vo\_1=quadv(fun1,x\_min,x\_max);

   Vo\_1=Vo\_1\*2/(B^0.5);

   %右部分

   fun2=@(x) 2/3\*(H-A\*x^2)^1.5;

   Vo\_2=quadv(fun2,x\_max,x\_r);

   Vo\_2=Vo\_2\*(2/(B^0.5));

   Vo\_all=Vo\_1+Vo\_2;

   if(Vo\_all>V\_boat)

   if(c\_center<c\_max)

   c\_max=c\_center;

   end

   end

   if(Vo\_all<V\_boat)

   if(c\_center>c\_min)

   c\_min=c\_center;

   end

   end

   c\_center=(c\_min+c\_max)/2;

   i=i+1;

   end

   F=c\_center;

   End

   第二阶段浮心X：

   %/\*-----确定积分上下限此处为右下积分----\*/

   %/\*-----x的浮心位置坐标直接用积分体积比例表示--\*/

   %/\*-----求出体积质量后直接比总重量----\*/

   function F=float\_X\_45(A,B,H,V\_boat,z)

   c=get\_intercept\_45(A,B,H,V\_boat,z);

   d=(tan(z)^2+4\*A\*c)^0.5;

   x\_min=(tan(z)-d)/(2\*A);

   x\_max=(H-c)/tan(z);

   x\_right=(H/A)^0.5;

   %左部分

   fun1=@(x) 2/3\*(tan(z)\*x+c-A\*x^2)^1.5;

   Vo\_1=quadv(fun1,x\_min,x\_max);

   Vo\_1=Vo\_1\*2/(B^0.5);

   %右部分

   fun2=@(x) 2/3\*(H-A\*x^2)^1.5;

   Vo\_2=quadv(fun2,x\_max,x\_right);

   Vo\_2=Vo\_2\*(2/(B^0.5));

   Vo\_all=Vo\_1+Vo\_2;

   %左部分

   fun3=@(x) x\*2/3\*(tan(z)\*x+c-A\*x^2)^1.5;

   Vo\_3=quadv(fun3,x\_min,x\_max);

   Vo\_3=Vo\_3\*2/(B^0.5);

   %右部分

   fun4=@(x) x\*2/3\*(H-A\*x^2)^1.5;

   Vo\_4=quadv(fun4,x\_max,x\_right);

   Vo\_4=Vo\_4\*(2/(B^0.5));

   Vo\_all2=Vo\_3+Vo\_4;

   % fprintf("带X的%f %f\n",Vo\_all2,Vo\_all);

   F=Vo\_all2/Vo\_all;

   % %x轴积分体积区域

   % fun1=@(x) x\*((tan(z)\*x+c-A\*x^2)\*sqrt(tan(z)+c-A\*x^2)-(tan(z)+c-A\*x^2)^1.5/3);

   % Vo\_1\_x=quadv(fun1,x\_min,x\_max);

   % Vo\_1\_x=Vo\_1\_x\*2/(B^0.5);

   % fun2=@(x) x\*(H-A\*x^2)^1.5;

   % Vo\_2\_x=quadv(fun2,x\_max,x\_right);

   % Vo\_2\_x=Vo\_2\_x\*(4/(3\*B^0.5));

   % Vo\_all\_x=Vo\_1\_x+Vo\_2\_x;

   %

   % %整体区域

   % fun\_a1=@(x) ((tan(z)\*x+c-A\*x^2)\*sqrt(tan(z)+c-A\*x^2)-(tan(z)+c-A\*x^2)^1.5/3);

   % Vo\_x=quadv(fun\_a1,x\_min,x\_max);

   % Vo\_x=Vo\_x\*2/(B^0.5);

   % fun\_a2=@(x) (H-A\*x^2)^1.5;

   % Vo\_a2=quadv(fun\_a2,x\_max,x\_right);

   % Vo\_a2=Vo\_a2\*(4/(3\*B^0.5));

   % Vo\_all=Vo\_x+Vo\_a2;

   % F=Vo\_all\_x/Vo\_all;

   % fprintf("x的重心位置%f %f\n",Vo\_all\_x,Vo\_all);

   End

   第二阶段浮心Z:

   %/\*-----确定积分上下限此处为右下右上积分----\*/

   %/\*-----x的浮心位置坐标直接用积分体积比例表示--\*/

   %/\*-----求出体积质量后直接比总重量----\*/

   function F=float\_Z\_45(A,B,H,V\_boat,z)

   c=get\_intercept\_45(A,B,H,V\_boat,z);

   d=(tan(z)^2+4\*A\*c)^0.5;

   x\_min=(tan(z)-d)/(2\*A);

   x\_max=(H-c)/tan(z);

   x\_right=(H/A)^0.5;

   %左部分

   fun1=@(x) 2/3\*(tan(z)\*x+c-A\*x^2)^1.5;

   Vo\_1=quadv(fun1,x\_min,x\_max);

   Vo\_1=Vo\_1\*2/(B^0.5);

   %右部分

   fun2=@(x) 2/3\*(H-A\*x^2)^1.5;

   Vo\_2=quadv(fun2,x\_max,x\_right);

   Vo\_2=Vo\_2\*(2/(B^0.5));

   Vo\_all=Vo\_1+Vo\_2;

   %左边带Z

   fun3=@(x) ((tan(z)\*x+c)^2-(A\*x^2)^2-2\*A\*x^2\*((tan(z)\*x+c)-A\*x^2)-((tan(z)\*x+c)-A\*x^2)^2/5)\*((tan(z)\*x+c)-A\*x^2)^0.5;

   Vo\_3=quadv(fun3,x\_min,x\_max);

   Vo\_3=Vo\_3\*2/B^0.5;

   %右边带Z

   fun4=@(x) (H^2-(A\*x^2)^2-2\*A\*x^2\*(H-A\*x^2)-(H-A\*x^2)^2/5)\*(H-A\*x^2)^0.5;

   Vo\_4=quadv(fun4,x\_max,x\_right);

   Vo\_4=Vo\_4\*2/B^0.5;

   Vo\_all2=Vo\_3+Vo\_4;

   %fprintf("带Z的%f %f\n",Vo\_all2,Vo\_all);

   F=Vo\_all2/Vo\_all;

   % fprintf("V体积总：%f 分：%f\n",Vo\_all2,Vo\_all);

   % fun1=@(x) ((tan(z)\*x+c)^2-A^2\*x^4)\*sqrt(tan(z)\*x+c-A\*x^2)-2\*A\*x^2\*(tan(z)\*x+c-A\*x^2)^1.5/3-(tan(z)\*x+c-A\*x^2)^2.5/5;

   % Vo\_z1=quadv(fun1,x\_min,x\_max);

   % Vo\_z1=Vo\_z1\*1/B^0.5;

   % fun2=@(x) (H^2-A^2\*x^4)\*sqrt(H-A\*x^2)-2\*A\*x^2.\*(H-A\*x^2)^1.5/3-(H-A\*x^2)^2.5/5;

   % Vo\_z2=quadv(fun2,x\_max,x\_right);

   % Vo\_z2=Vo\_z2\*(2/(3\*B^0.5));

   % Vo\_z\_all=Vo\_z1+Vo\_z2;

   %

   %x的上下限

   % fun3=@(x) ((tan(z)\*x+c-A\*x^2)\*sqrt(tan(z)\*x+c-A\*x^2)-(tan(z)\*x+c-A\*x^2)^1.5/3);

   % Vo\_a=quadv(fun3,x\_min,x\_max);

   % Vo\_a=Vo\_a\*2/(B^0.5);

   %

   % fun4=@(x) 2/3\*(H-A\*x^2)^1.5;

   % Vo\_a2=quadv(fun4,x\_max,x\_right);

   % Vo\_a2=Vo\_a2\*(4/(3\*B^0.5));

   % Vo\_all=Vo\_a+Vo\_a2;

   % F=Vo\_z\_all/Vo\_all;

   End

   第三阶段：

   function F=get\_intercept\_90\_180(A,B,H,vp,z)

   cof=4/(3\*sqrt(B));

   i=1;

   cmin=H+tan(z)\*sqrt(H/A);%左上交点

   cmax=H-tan(z)\*sqrt(H/A);%右上交点

   c=double((cmin+cmax)/2);

   while(i<100)

   d=tan(z)^2+4\*A\*c;

   xmin=-sqrt(H/A);

   xL=(H-c)/tan(z);

   xR=(tan(z)+sqrt(d))/(2\*A);

   xmax=sqrt(H/A);

   fun1=@(x) (tan(z)\*x+c-A\*x.^2).^1.5;

   v1=quadv(fun1,xL,xR);

   v1=v1\*cof;%水上右

   fun2=@(x) (H-A\*x.^2).^1.5;

   v2=quadv(fun2,xmin,xL);

   v2=v2\*cof;%水上左

   v=quadv(fun2,xmin,xmax);

   v=v\*cof;%总体积

   volumn=v-v1-v2;

   if(volumn<vp)%

   if(c<cmax)

   cmax=c;

   end

   end

   if(volumn>vp)%

   if(c>cmin)

   cmin=c;

   end

   end

   if(volumn==vp)

   break;

   end

   c=double((cmin+cmax)/2);

   i=i+1;

   end

   F=c;

   end

   第三阶段浮心Z：

   %/\*-----确定积分上下限此处为右下积分和右上----\*/

   %/\*-----x的浮心位置坐标直接用积分体积比例表示--\*/

   %/\*-----求出体积质量后直接比总重量----\*/

   function F=float\_Z\_90\_180(A,B,H,V\_boat,z)

   c=get\_intercept\_90\_180(A,B,H,V\_boat,z);

   d=(tan(z)^2+4\*A\*c)^0.5;

   x\_min=(H-c)/tan(z);

   x\_max=(tan(z)+d)/(2\*A);

   x\_r=(H/A)^0.5;

   x\_l=-(H/A)^0.5;

   fun1=@(x) (H-tan(z)+c)\*(tan(z)+c-A\*x^2)^0.5;

   Vo\_1=quadv(fun1,x\_min,x\_max);

   Vo\_1=Vo\_1\*4/(3\*B^0.5);

   fun2=@(x) (H-A\*x^2-(H-A\*x^2)/3)\*(H-A\*x^2)^0.5;

   Vo\_2=quadv(fun2,x\_l,x\_min);

   Vo\_2=Vo\_2\*2/B^0.5;

   Vo\_all=Vo\_1+Vo\_2;

   Vo\_total=quadv(fun2,x\_l,x\_r);

   Vo\_total=Vo\_total\*2/(1\*B^0.5);

   Vo\_all=Vo\_total-Vo\_all;

   fun3=@(x) (H^2-A^2\*x^4-2\*A\*x^2\*(H-A\*x^2)/3-(H-A\*x^2)^2/5)\*(H-A\*x^2)^0.5;

   Vo\_3=quadv(fun3,x\_min,x\_max);

   Vo\_3=Vo\_3/B^0.5;

   fun4=@(x) (H^2-A^2\*x^4)\*sqrt(H-A\*x^2)-2\*A\*x^2\*(H-A\*x^2)^1.5/3-(H-A\*x^2)^2.5/5;

   Vo\_4=quadv(fun4,x\_l,x\_min);

   Vo\_4=Vo\_4\*2/(1\*B^0.5);

   Vo\_Z=quadv(fun4,x\_l,x\_r);

   Vo\_Z=Vo\_Z\*(2/(1\*B^0.5));

   Vo\_all2=Vo\_3+Vo\_4;

   Vo\_all2=Vo\_Z-Vo\_all2;

   %fprintf("不带%f 带%f\n",Vo\_all,Vo\_all2);

   F=Vo\_all2/Vo\_all;

   % fun1=@(x) 2/3\*(H-A\*x^2)^1.5;

   % Vo\_z1=quadv(fun1,x\_left,x\_min);

   % Vo\_z1=Vo\_z1\*2/B^0.5;

   %

   % fun2=@(x) (tan(z)\*x+c-A\*x^2-(tan(z)\*x+c-A\*x^2)/3)\*sqrt(tan(z)\*x+c-A\*x^2);

   % Vo\_z2=quadv(fun2,x\_min,x\_max);

   % Vo\_z2=Vo\_z2\*2/B^0.5;

   % Vo\_z\_all=Vo\_z1+Vo\_z2;

   %

   % fun3=@(x)(H^2-A^2\*x^4-(H-A\*x^2)^2/5)\*(H-A\*x^2)^0.5;

   % Vo\_a=quadv(fun3,x\_left,x\_min);

   % Vo\_a=Vo\_a\*1/(B^0.5);

   %

   % fun4=@(x) ((tan(z)+c)^2-A\*x^2\*x^4-2/3\*A\*x^2\*(tan(z)\*x+c-A\*x^2)-(tan(z)\*x+c-A\*x^2)^2/5)\*(tan(z)\*x+c-A\*x^2)^0.5;

   % Vo\_a2=quadv(fun4,x\_min,x\_max);

   % Vo\_a2=Vo\_a2\*1/(B^0.5);

   % Vo\_all=Vo\_a+Vo\_a2;

   % fprintf("Z这里的浮心：%f %f\n",Vo\_a,Vo\_a2);

   %

   % F=Vo\_all/Vo\_z\_all;

   End

   第三阶段浮心X:

   function F=float\_X\_90\_180(A,B,H,V\_boat,z)

   c=get\_intercept\_90\_180(A,B,H,V\_boat,z);

   d=(tan(z)^2+4\*A\*c)^0.5;

   x\_min=(H-c)/tan(z);

   x\_max=(tan(z)+d)/(2\*A);

   x\_l=-(H/A)^0.5;

   x\_r=(H/A)^0.5;

   %全部的

   fun1=@(x) (tan(z)-A\*x^2)\*(tan(z)+c-A\*x^2)^0.5;

   Vo\_1=quadv(fun1,x\_min,x\_max);

   Vo\_1=Vo\_1\*4/(3\*B^0.5);

   fun2=@(x) (H-A\*x^2-(H-A\*x^2)/3)\*(H-A\*x^2)^0.5;

   Vo\_2=quadv(fun2,x\_min,x\_l);

   Vo\_2=Vo\_2\*2/B^0.5;

   Vo\_total=quadv(fun2,x\_l,x\_r);

   Vo\_total=Vo\_total\*2/(1\*B^0.5);

   Vo\_all=Vo\_total-(Vo\_1+Vo\_2);

   %带x的 检查

   fun3=@(x) x\*(H-tan(z)+c)\*(tan(z)+c-A\*x^2)^0.5;

   Vo\_3=quadv(fun3,x\_min,x\_max);

   Vo\_3=Vo\_3\*4/(3\*B^0.5);

   fun4=@(x) 2/3\*x\*(H-A\*x^2)^1.5;

   Vo\_4=quadv(fun4,x\_max,x\_r);

   Vo\_4=Vo\_4\*2/B^0.5;

   Vo\_all2=Vo\_3+Vo\_4;

   %fprintf("不带%f 带%f\n ",Vo\_all,Vo\_all2);

   F=-Vo\_all2/Vo\_all;

   % %x轴积分体积区域

   % fun1=@(x) x\*2/3\*(H-A\*x^2)^1.5;

   % Vo\_x=quadv(fun1,x\_left,x\_min);

   % Vo\_x=Vo\_x\*2/B^0.5;

   %

   % fun2=@(x) x\*(tan(z)\*x+c-A\*x^2-(tan(z)\*x+c-A\*x^2)/3)\*sqrt(tan(z)\*x+c-A\*x^2);

   % Vo\_X=quadv(fun2,x\_min,x\_max);

   % Vo\_X=Vo\_X\*2/B^0.5;

   % Vo\_xall=Vo\_X+Vo\_x;

   % fprintf("X这里的浮心：%f %f\n",Vo\_x,Vo\_X);

   %

   % %整体区域

   %

   % fun1=@(x) 2/3\*(H-A\*x^2)^1.5;

   % Vo\_x1=quadv(fun1,x\_left,x\_min);

   % Vo\_x1=Vo\_x1\*2/B^0.5;

   %

   % fun2=@(x) (tan(z)\*x+c-A\*x^2-(tan(z)\*x+c-A\*x^2)/3)\*sqrt(tan(z)\*x+c-A\*x^2);

   % Vo\_x2=quadv(fun2,x\_min,x\_max);

   % Vo\_x2=Vo\_x2\*2/B^0.5;

   % Vo\_x\_all=Vo\_x1+Vo\_x2;

   %

   % F=Vo\_xall/Vo\_x\_all;

   end

   第四阶段：

   function F=get\_intercept\_last(A,B,H,vp,z)

   cof=4/(3\*sqrt(B));

   i=1;

   cmin=0;%左上交点

   cmax=H+tan(z)\*sqrt(H/A);%左上角交点

   c=double((cmin+cmax)/2);

   while(i<100)

   d=tan(z)^2+4\*A\*c;

   xL=(tan(z)-sqrt(d))/(2\*A);

   xR=(tan(z)+sqrt(d))/(2\*A);

   xmin=-sqrt(H/A);

   xmax=sqrt(H/A);

   fun1=@(x) (tan(z)\*x+c-A\*x.^2).^1.5;

   v1=quadv(fun1,xL,xR);

   v1=v1\*cof;%水下

   fun2=@(x) (H-A\*x.^2).^1.5;

   v2=quadv(fun2,xmin,xL);

   v2=v2\*cof;%水上左

   v=quadv(fun2,xmin,xmax);

   v=v\*cof;%水总

   volumn=v-v1-v2;

   if(volumn<vp)%体积小，太高了

   if(c<cmax)

   cmax=c;

   end

   end

   if(volumn>vp)%体积大，太低了

   if(c>cmin)

   cmin=c;

   end

   end

   if(volumn==vp)

   break;

   end

   c=double((cmin+cmax)/2);

   i=i+1;

   end

   F=c;

   %fprintf("%f\n",c);

   End

   第四阶段浮心X：

   function F=float\_X\_theat\_180(A,B,H,theta,c)

   d=tan(theta)^2+4\*A\*c;

   xmin=-sqrt(H/A);%左极限点x

   xL=(H-c)/tan(theta);%左交点x

   xR=(tan(theta)+sqrt(d))/(2\*A);%右交点x

   xmax=sqrt(H/A);%右极限点x

   cof=4/(3\*sqrt(B));

   fun1=@(x) (tan(theta)\*x+c-A\*x.^2).^1.5;

   v1=quadv(fun1,xL,xR);v1=v1\*cof;

   fun2=@(x) (H-A\*x.^2).^1.5;

   v2=quadv(fun2,xmin,xL);v2=v2\*cof;

   v=quadv(fun2,xmin,xmax);v=v\*cof;

   volumn=v-v1-v2;

   %小v为体积，大V是关于x的体积积分

   Fun1=@(x) x.\*(tan(theta)\*x+c-A\*x.^2).^1.5;

   V1=quadv(Fun1,xL,xR);V1=V1\*cof;

   Fun2=@(x) x.\*(H-A\*x.^2).^1.5;

   V2=quadv(Fun2,xmin,xL);V2=V2\*cof;

   %fprintf("volumn is %f, Vx1 is %f,",volumn,V1);%测试输出

   F=-(V1+V2)/volumn;

   % c=get\_intercept\_theat\_180(A,B,H,V\_boat,z);

   % d=(tan(z)^2+4\*A\*c)^0.5;

   %

   % x\_min=(tan(z)-d)/(2\*A);

   % x\_max=(tan(z)+d)/(2\*A);

   % x\_l=-(H/A)^0.5;

   % x\_r=(H/A)^0.5;

   % %全部的

   % fun1=@(x) (H-A\*x^2)^1.5;

   % Vo\_1=quadv(fun1,x\_min,x\_max);

   % Vo\_1=Vo\_1\*4/(3\*B^0.5);

   % fun2=@(x) (H-A\*x^2)^1.5;

   % Vo\_total=quadv(fun2,x\_l,x\_r);

   % Vo\_total=Vo\_total\*4/(3\*B^0.5);

   % Vo\_all=Vo\_total-Vo\_1;

   %

   % %带x的 检查

   % fun3=@(x) x\*(H-A\*x.^2)^1.5;

   % Vo\_3=quadv(fun3,x\_min,x\_max);

   % Vo\_3=Vo\_3\*4/(3\*B^0.5);

   % fun4=@(x) x\*(H-A\*x.^2)^1.5;

   % Vo\_4=quadv(fun4,x\_l,x\_r);

   % Vo\_4=Vo\_4\*4/(3\*B^0.5);

   % Vo\_all2=Vo\_4-Vo\_3;

   % %fprintf("X不带%f 带%f\n ",Vo\_1,Vo\_total);

   % F=Vo\_all2/Vo\_all;

   % %x轴积分体积区域

   % fun1=@(x) x\*2/3\*(H-A\*x^2)^1.5;

   % Vo\_x=quadv(fun1,x\_left,x\_min);

   % Vo\_x=Vo\_x\*2/B^0.5;

   %

   % fun2=@(x) x\*(tan(z)\*x+c-A\*x^2-(tan(z)\*x+c-A\*x^2)/3)\*sqrt(tan(z)\*x+c-A\*x^2);

   % Vo\_X=quadv(fun2,x\_min,x\_max);

   % Vo\_X=Vo\_X\*2/B^0.5;

   % Vo\_xall=Vo\_X+Vo\_x;

   % fprintf("X这里的浮心：%f %f\n",Vo\_x,Vo\_X);

   %

   % %整体区域

   %

   % fun1=@(x) 2/3\*(H-A\*x^2)^1.5;

   % Vo\_x1=quadv(fun1,x\_left,x\_min);

   % Vo\_x1=Vo\_x1\*2/B^0.5;

   %

   % fun2=@(x) (tan(z)\*x+c-A\*x^2-(tan(z)\*x+c-A\*x^2)/3)\*sqrt(tan(z)\*x+c-A\*x^2);

   % Vo\_x2=quadv(fun2,x\_min,x\_max);

   % Vo\_x2=Vo\_x2\*2/B^0.5;

   % Vo\_x\_all=Vo\_x1+Vo\_x2;

   %

   % F=Vo\_xall/Vo\_x\_all;

   End

   第四阶段浮心Z:

   function F=float\_Z\_theat\_180(A,B,H,theta,c)

   d=tan(theta)^2+4\*A\*c;

   xmin=-sqrt(H/A);%左极限点x

   xL=(H-c)/tan(theta);%左交点x

   xR=(tan(theta)+sqrt(d))/(2\*A);%右交点x

   xmax=sqrt(H/A);%右极限点x

   cof=4/(3\*sqrt(B));Cof=1/sqrt(B);

   fun1=@(x) (tan(theta)\*x+c-A\*x.^2).^1.5;%右边

   v1=quadv(fun1,xL,xR);v1=v1\*cof;

   fun2=@(x) (H-A\*x.^2).^1.5;%左边

   v2=quadv(fun2,xmin,xL);v2=v2\*cof;

   v=quadv(fun2,xmin,xmax);v=v\*cof;

   volumn=v-v1-v2;

   %小v为体积，大V是关于z的体积积分

   Fun1=@(x) ((tan(theta)\*x+c).^2-A^2\*x.^4).\*sqrt(tan(theta)\*x+c-A\*x.^2)-2\*A\*x.^2.\*(H-A\*x.^2).^1.5/3-(tan(theta)\*x+c-A\*x.^2).^2.5/5;

   V1=quadv(Fun1,xL,xR);V1=V1\*Cof;

   Fun2=@(x) (H^2-A^2\*x.^4).\*sqrt(H-A\*x.^2)-2\*A\*x.^2.\*(H-A\*x.^2).^1.5/3-(H-A\*x.^2).^2.5/5;

   V2=quadv(Fun2,xmin,xL);V2=V2\*Cof;

   V=quadv(Fun2,xmin,xmax);V=V\*Cof;

   Vz=V-V1-V2;

   F=Vz/volumn;

   % c=get\_intercept\_theat\_180(A,B,H,V\_boat,z);

   % d=(tan(z)^2+4\*A\*c)^0.5;

   % x\_min=(tan(z)-d)/(2\*A);

   % x\_max=(tan(z)+d)/(2\*A);

   % x\_r=(H/A)^0.5;

   % x\_l=-(H/A)^0.5;

   %

   % % fprintf("%f %f\n",z\*180/pi,c);

   % fun1=@(x) ((tan(z)\*x+c)-A\*x^2)^1.5;

   % Vo\_1=quadv(fun1,x\_min,x\_max);

   % Vo\_1=Vo\_1\*4/(3\*B^0.5);

   % fun2=@(x) (H-A\*x.^2)^1.5;

   % Vo\_2=quadv(fun2,x\_l,x\_r);

   % Vo\_2=Vo\_2\*4/(3\*B^0.5);

   % Vo\_all=Vo\_2-Vo\_1;

   %

   %

   % fun3=@(x) (H^2-A^2\*x^4-2\*A\*(H-A\*x^2)/3-(H-A\*x^2)^2/5)\*(H-A\*x^2)^0.5;

   % Vo\_3=quadv(fun3,x\_min,x\_max);

   % Vo\_3=Vo\_3/B^0.5;

   % fun4=@(x) (H^2-A^2\*x^4)\*sqrt(H-A\*x^2)-2\*A\*x^2\*(H-A\*x^2)^1.5/3-(H-A\*x^2)^2.5/5;

   % Vo\_4=quadv(fun4,x\_l,x\_r);

   % Vo\_4=Vo\_4\*1/B^0.5;

   % Vo\_all2=Vo\_3-Vo\_4;

   % % fprintf("Z不带%f 带%f\n",Vo\_1,Vo\_2);

   % F=Vo\_all2/Vo\_all;

   end

   寻找西塔角度：

   function F\_=the\_0\_to\_xita(A,B,H,V\_boat)

   pre=8/(3\*B^0.5);

   fun\_of\_tail=@(x) (H-A\*x^.2).^1.5;

   v=quadv(fun\_of\_tail,0,sqrt(H/A));

   v=v\*pre;

   H\_min=0;

   H\_max=H;

   H\_temp=double(H\_min+H\_max)/2;

   i=1;

   while(i<1000)

   F\_change=@(x) (H\_temp-A\*x^2)^1.5;

   x\_max=(H\_temp/A)^0.5;

   V\_temp=quadv(F\_change,0,x\_max);

   V\_temp=V\_temp\*pre;

   if(V\_temp>V\_boat)

   if(H\_temp<H\_max)

   H\_max=H\_temp;

   end

   end

   if(V\_temp<V\_boat)

   if(H\_temp>H\_min)

   H\_min=H\_temp;

   end

   end

   if(V\_temp==V\_boat)

   break;

   end

   H\_temp=double(H\_min+H\_max)/2;

   i=i+1;

   end

   F\_=H\_temp;

   End

   画图函数：

   铁的质量增加：

   function F=get\_add\_tie\_art(A,B,H)

   V\_boat=0.0004;

   T=[];

   U=[];

   i=1;

   while(V\_boat<0.0014)

   T=[T V\_boat];

   f\_water\_line=the\_0\_to\_xita(A,B,H,V\_boat);%吃水线

   U=[U f\_water\_line];

   plot(T,U);

   xlabel('x,重量');

   ylabel('y,吃水线');

   title('重量改变曲线图');

   V\_boat=V\_boat+0.00001;

   i=i+1;

   end

   F=1;

   End

   三维曲面图：

   %三维图

   colormap(hsv) ;

   t=-0.5:0.04:0.5;

   A=10;

   B=6;

   H=0.15;

   [x,y] =meshgrid(t);

   z=A\*x.^4+B\*y.^2;

   surf(x,y,z);

   hold on;

   colormap(cool) ;

   [X,Y] =meshgrid(t);

   Z=-Y\*tand(20)+0.05;%得到的角度和c

   surf(X,Y,Z);

   xlabel('X');

   ylabel('Y');

   zlabel('Z');

   title('surf三维曲面图');

   axis([-0.5 0.5 -0.5 0.5 0 H]);

   二维曲面图：

   %2D图

   clear;

   x=-0.2:0.01:0.2;

   A=10;

   H=0.12;

   y=A\*x.^2;

   plot(x,y);

   hold on;

   plot([-0.2 0.2],[0.4 0.4])

   y=x\*tand(66)+ 0.1;

   plot(x,y)

   axis equal;

   axis([-0.5 0.5 0 0.5]);

   吃水线方程：

   function F=water\_line\_art(A,B,H,water\_line)

   water\_line=water\_line\*1000;

   fsurf(@(x,y) A\*x.^2+B\*y.^2);

   zlim([0 H\*1000]);

   hold on;

   fsurf(water\_line);

   F=1;

   end [↑](#endnote-ref-1)