

基于多种方法的系泊系统方案设计

摘要

本文针对系泊系统的状态分析、部件规格设计、方案选择等问题进行合理的数学抽象建立基于力学平衡的物理模型，应用试算法、二分法、改进的理想点法、遗传算法等求解方法对模型进行求解得到结果并加以分析。

问题一中，针对物理受力运用改进的集中质量多边形近似法（坐标系分解法）建立力学模型，选取合适的试算间隔运用试算法编写 C 语言程序进行模型求解，得到在风向为 0 度，风速为 12 m/s 和 24 m/s 时钢桶和各节钢管的倾斜角度如表 1、锚链形状如图 6、浮标的吃水深度分别为 0.736 m 和 0.751 m 及游动区域半径为 13.8576 m 和 16.9451 m。物体在风向法平面的投影面与风向有关，讨论后得到各个风向下的上述待求量，本文展示出浮标吃水深度和游动区域随风向的变化曲线如图 9-11，并对规律曲线进行最大值最小值分析，模型检验采用偏差分析如图 12、13。

问题二中，根据第一问的物理建模，首先试算风速为 3600 m/s，风向为 0 度及重物球质量为 1200 kg 的系统状态如图 14，表 2，且不满足角度约束，而重物球质量为 3000 kg 时满足角度约束，进而在 1200~3000 kg 范围内由单调性通过二分法得到 2228 kg 时铁桶和竖直面的夹角达到 86.439 度而锚链末端和海床的夹角为 15.7161 度，为恰好满足角度的约束条件的重物球的最小质量。对结果左右侧进行角度约束灵敏度分析如图 15，表 3。

问题三中，在第一问的模型中考虑海水的作用力及物体在海水流法平面的投影面得到改进的物理模型并进行合理的近似处理，针对系泊系统设计的三个目标通过改进的理想点算法将多目标转化为单一目标贴近度，在满足钢桶轴线和竖直面的夹角小于 5 度，锚链末端、海床的夹角小于 16 度、水深介于 16~20 m 及锚链不接触海床的四个约束条件下，给定的风速，流速和风向基础上运用基于二进制编码的遗传算法对锚链型号，锚链环数目（锚链长度）和重物球质量进行筛选，最后本文仅给出在初始条件风速为 12 m/s，风向为 0 度，海水流速为 1.5 m/s 时符合约束的最优方案，其余初始条件对应的方案由 EXE 软件呈现。本文展示遗传算法计算 3 次 1000 和 10000 繁衍代数下的所求目标值如表 5、6。最后通过灵敏度分析对结果可靠性进行检验。

综上所述，本文在对系泊系统深入了解的基础上针对提及的监测系泊系统通过建立合理的物理模型，运用多种不同的计算算法对体系进行求解并进行深入分析，在得到相应的结果同时，为深入的研究奠定了基础。

关键字：试算法 EXE 软件 二分法 改进的理想点算法 遗传算法

1 问题重述

1.1. 背景

传统的单点系泊系统可以分为单点系泊储油装置和单点系泊卸油装置两种，以实现将 FPSO（浮式生产储油卸油装置）定位于预定海域，起到输送井流，电力、通信等功能^[1]。

现如今在海洋观测领域，实施海洋调查、监测、海洋环境预报等各项活动的国家地区越来越多。带有海洋测量仪器的各种系泊系统被广泛应用于各大洋的海洋科考调查，系泊系统也被公认为测量某一海域环境参数的最佳选择^[2]。单点系泊系统的姿态可随风、浪、流的变化而自由转向，且基本趋向于系统受力最小的方位，因而具有风标的作用。同时，造价低廉，应用方便也使其发展和应用越来越广泛。

本题中提到近浅海观测网的传输节点由浮标系统、系泊系统和水底通讯系统组成，其中最为重要的即为浮标系统和系泊系统，其中对浮标系统和系泊系统的分析和相关模型研究势必将会为系泊系统的发展起到关键作用。基于此，本文将基于合理假设对该系统进行力学分析和建模计算，从而在解决问题的基础上为后续研究提供方便。

1.2. 问题

1) 某型传输节点选用 II 型电焊锚链 22.05 m (长度为 105 mm 单位长度质量为 7 kg/m)，选用的重物球的质量为 1200 kg。现将该型传输节点布放在水深 18 m、海床平坦、海水密度为 $1.025 \times 10^3 \text{ kg/m}^3$ 的海域。如果海水静止，分别计算海面风速为 12 m/s 和 24 m/s 时钢桶和各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域。

2) 在问题 1 的假设下，计算海面风速为 36 m/s 时钢桶和各节钢管的倾斜角度、锚链形状和浮标的游动区域。请调节重物球的质量，使得钢桶的倾斜角度不超过 5 度，锚链中锚点与海床的夹角不超过 15 度。

3) 由于潮汐等因素的影响，布放海域的主测水深介于 16~20 m 之间。布放点的海水速度最大可达到 1.5 m/s、风速最大可达到 36 m/s。请给出考虑风力、水流力和水深情况下的系泊系统设计，分析不同情况下钢桶、钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域。

2 模型假设

1. 模型假设

- 1) 假设体系中的各个部分均为刚体，不考虑刚体内部结构的变化等。
- 2) 假设第一根钢管（钢管从上到下为第一根到第四根）连接浮标的底部中央且考虑到浮标吃水深度的定义、浮标的稳定性、浮标受到的风力相对自重和海水对其作用力的相对大小，假设浮标不会翻转，仅可以做水平游动。
- 3) 假设海面上气压变化是广域的，从而海风是均匀的，不考虑旋风。
- 4) 假设海水的流动是水平的，且海水的流动为均匀流，不考虑漩涡流。
- 5) 假设在其轴线和竖直面夹角较小情况下钢管、钢桶正对海水流动方向。

3 符号说明

符号	符号意义及单位
x	相接触两个物体之间的作用力的水平分量 (N)
x'	相接触两个物体之间的作用力的水平分量反作用力 (N)
y	相接触两个物体之间的作用力的竖直分量 (N)
y'	相接触两个物体之间的作用力的竖直分量的反作用力 (N)
w	多目标规划贴合度 (遗传算法适应度)
θ	该物体轴向方向和水平面的夹角 (度)
l	该物体的长度或者高度 (m)
v	风速或海水流速 (m/s)
H	海平面到潜标深度 (m)
S	浮标的游动区域半径 (m)
$F_{\text{风}}$	风对浮标的作用力 (后分解为水平方向和竖直方向) (N)
$F_{\text{水}}$	海水对物体的作用力 (N)
$F_{\text{浮}}$	物体所受浮力 (N)
F_i	集中质量的多边形近似法中相邻锚链链环之间的作用力 (N)
N	种群规模
$X(t)$	种群进化代数
t	进化代数计数器
T	终止进化代数

4 第一问的建模求解

4.1. 问题的分析

原问题：某型传输节点选用 II 型电焊锚链 22.05 m (长度为 105 mm, 单位长度质量为 7 kg/m)，选用的重物球的质量为 1200 kg。现将该型传输节点布放在水深 18 m、海床平坦、海水密度为 $1.025 \times 10^3 \text{ kg/m}^3$ 的海域。如果海水静止，分别计算海面风速为 12 m/s 和 24 m/s 时钢桶和各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域。

分析：原问题中要求求解各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域，首要的是基于合理的假设对整个体系进行受力分析，对于浮标，钢桶、重物球和锚，其受力分析较为简单，而连接的钢管和锚链的受力较为复杂。由于钢管的连接方式主要为焊接连接、法兰连接、灌浆套筒等连接方式^[3]，其特点为均为固定连接，并不适用于本问题。基于此，针对钢管本文采用坐标轴分解法对钢管的受力进行分析，同时考虑到钢管不同的倾斜角。针对锚链，本文对提到的基于集中质量的多边形近似法^[4]进行改进，从而对锚链进行受力分析。

经过分析，得到以吃水深度 h 为主要自变量的方程组。基于题目给定水深 18 米的条件可以列出关于吃水深度的另一个约束方程，与之前的方程组结合可以求得所有的变量值，包括各个钢管和锚链的受力和角度，同时也就求出了其相应的形状，得到问题的答案。

概念约定：

✚ **角度**：本文中所有的角度除非特别说明，均为物体轴线和水平面的夹角。

✚ **G_i** ：第一问中表示钢管（锚链）和钢桶所受浮力和重力的过程中，用 G_i 表示二者的合力，不仅仅指代重力。

4.2. 基于单元力学分析模型的建立

受力分析：首先将体系分成各个部分分别进行考虑，然后画出各个部分的受力分析图，进而对各个部分分别用不同的方法进行受力分析，从而列出基本的受力平衡方程和力矩平衡方程，联立得到的方程组可以得到总的以浮标吃水深度为主要自变量的平衡方程组，带入数据求解。

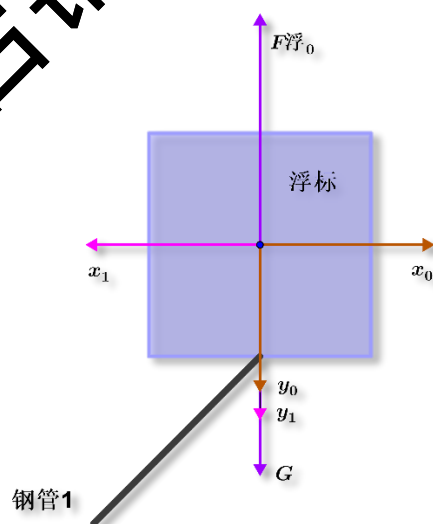


图 1 浮标受力分析图

浮标受力分析：浮标受到风力，竖直向上的浮力 $F_{\text{浮}0}$ ，竖直向下的重力 G ，钢管的水平拉力 x_l 和竖直拉力 y_l 的作用而平衡。浮标的浮力和重力相对简单，基于假设浮标不会翻转，故其仅仅用浮力和重力的定义就可以简单计算：

$$\begin{cases} x_l = x_0 \\ F_{\text{浮}0} = G + y_l + y_0 \end{cases}$$

基于之前的问题分析，第一根钢管对浮标的作用力为第一根钢管上端所受力的反作用力，分解为水平方向 x_l 和竖直方向 y_l 。

经分析可以得到风吹向浮标，浮标受风的部分为水面上的部分，风的风荷载不用原题中给出的公式，其中受风面积即为物体在风向法平面的投影面积，接下来进行讨论。而风荷载进行水平和竖直分解后可以得到：

$$\begin{cases} F_{\text{风}} = \sqrt{x_0^2 + y_0^2} \\ F_{\text{风}} = 0.625 \times s v^2 \\ x_0 = F_{\text{风}} \cdot \cos \theta_0 \\ y_0 = F_{\text{风}} \cdot \sin \theta_0 \end{cases}$$

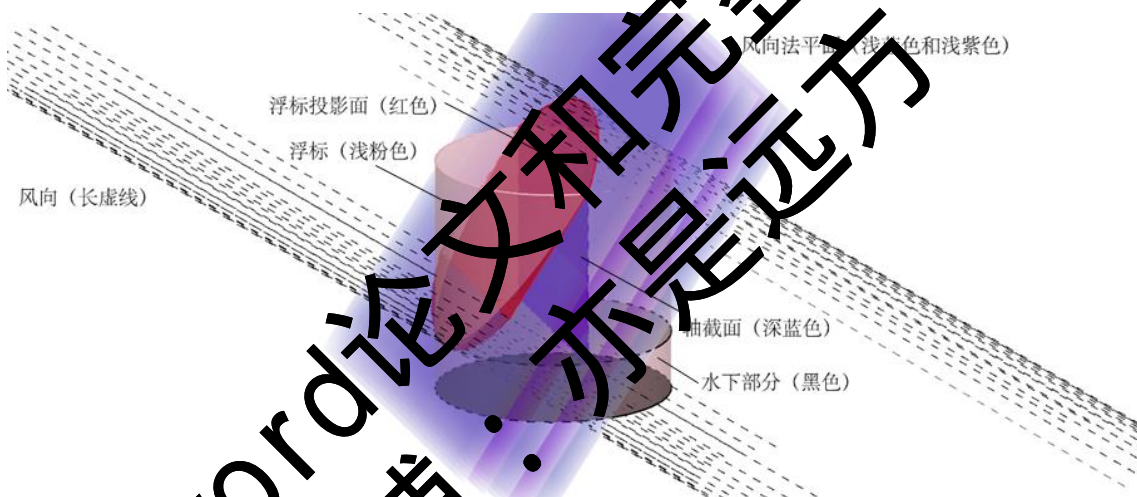


图2 浮标风向投影图

如图2所示 (GeoGebra 作图软件)，相应的颜色代表的实际含义如下：浮标 (浅粉色) 为所研究的浮标，其大小未按照严格的比例进行绘制；风向，即为风吹向浮标的方向 (本例为斜向下) 用长虚线表示；水下部分即为被海水淹没的部分，用黑色横截面加以区分，此部分不受风的影响；风向法平面集合 (浅蓝色和浅紫色间隔表示) 实际上有无数多个，图中仅为示意；轴截面 (深蓝色)，即为纵切圆柱得到的竖直截面；浮标投影面即物体在风向法平面上的投影面，它由三部分组成，第一部分为圆柱横截面在法平面的投影面 s_1 ，第二部分为圆柱下底面 (圆柱与海平面相交的横截面) 左半圆在法平面的投影面 s_2 ，第三部分为圆柱上底面右半圆在法平面的投影面 s_3 ，其中 s_2 、 s_3 之和为圆柱底面面积。设风向和水平面的夹角为 θ (本文中所有的 θ 均为物体与水平面的夹角)，则有：

$$\begin{cases} s = s_1 \cdot \cos \theta + s_2 \cdot \sin \theta + s_3 \cdot \sin \theta \\ s_2 + s_3 = \pi r^2 \end{cases}$$

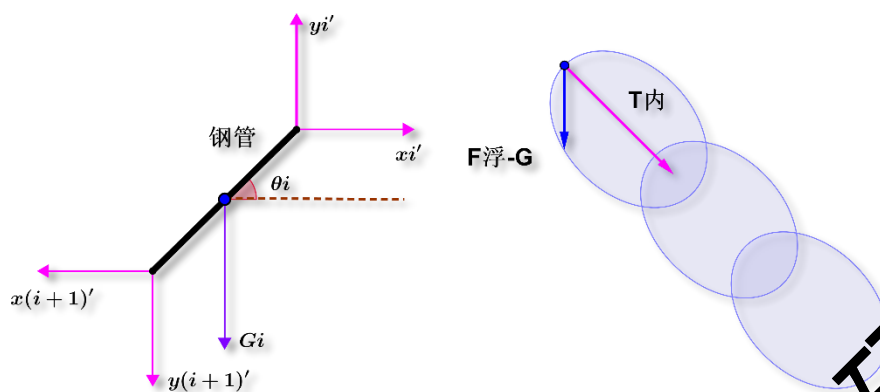


图3 钢管受力和集中质量的多边形近似法的锚链受力分析图

第 i 根钢管（第 i 段锚链）受力分析：四根同样的钢管受力的情况基本相同，竖直方向上受到重力和浮力的作用（二者合力为 G_i ）；而钢管的两端则受到钢管和下一根钢管的作用力 $x_{(n+1)}$ ， $y_{(n+1)}$ ，基于假设将其水平和竖直进行投影得到如图2中所示下标的分力。对于锚链，本文采用水平和竖直分解的方法对每段的锚链的受力进行分解，取代原集中质量的多边形近似法中考虑沿每段锚链的内力的做法（ G_i 为浮力），从而更为精确（原方法中锚链受力的方向可以不沿锚链，但是由于锚链较小可以忽略，本文则未忽略）。

集中质量的多边形近似法的受力平衡如下所示：

$$\begin{cases} B_i - G_i + T_{i-1} \sin \theta_{i-1} = T_i \sin \theta_i \\ T_{i-1} \cos \theta_{i-1} = T_i \cos \theta_i \end{cases}$$

改进法的受力平衡和力矩方程如下所示：

$$\begin{cases} x_i = x_{i-1}' \\ y_{i+1} = y_i' - G_i \\ G_i \cdot \frac{l}{2} \cdot \cos \theta_i + x_i' \cdot l \cdot \sin \theta_i = y_i' \cdot l \cdot \cos \theta_i \end{cases}$$

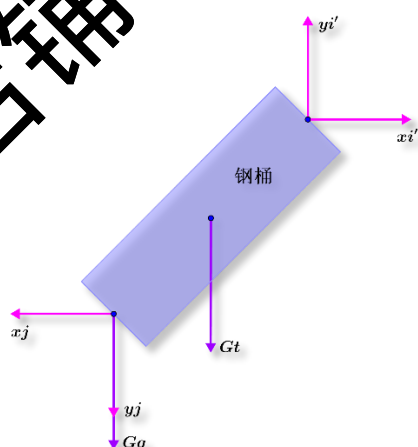


图4 钢桶受力分析图

钢桶受力分析：钢桶受到的力有重力 G_t ，第四根钢管和锚链的作用力 x_5 ， y_5 ， x_j ， y_j 和重物球的拉力 G_g 。且锚链对钢桶的作用力不是沿锚链的开始部分的切线等（原集中质

量的多边形近似法), 而是被分解为竖直方向和水平方向。如下的公式分别为

$$\begin{cases} x_j = x_5' \\ y_j + G_g + G_t = y_5' \\ G_t \cdot \frac{l}{2} \cdot \cos \theta_5 + x_5' \cdot l \cdot \sin \theta_5 = y_5' \cdot l \cdot \cos \theta_5 \end{cases}$$

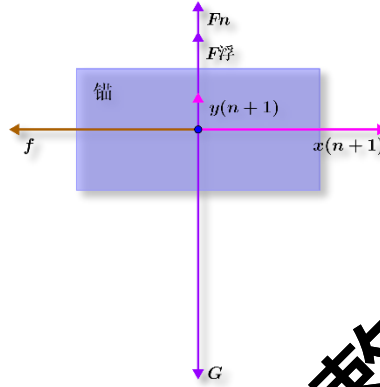


图 5 锚受力分析图

重力锚受力分析: 锚泊受力为最后一段锚链施加的水平作用力 $x(n+1)$ 和竖直作用力 $y(n+1)$, 重力 G , 海床对其的支持力 F_n 和静摩擦力 f 。由于锚的体积较小, 且作用力多数可以集中在质量中心, 同样不考虑其翻转的情况, 列出水平和竖直方向的受力平衡方程得到:

$$\begin{cases} F_n + F_{浮n+1} + y_{n+1} = G_n \\ f = x_{n+1} \end{cases}$$

4.3. 模型的试算法^[4-6]求解

考虑风的方向后, 风向的变化会使得原问题中要求求解的结果变得有无穷多个, 本论文中仅仅给出风向水平时候的钢桶和各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域, 对于风向变化的问题求解作者给出部分图示, 并在附录和附件中给出相应的程序, 进行微调后可以求解出任意风向下钢桶和各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域。

上述模型的建立实际上仅仅给出了以浮标吃水深度 h 为主要自变量的等式集合, 还要有一个约束条件即水深为 18 m 的条件:

$$H = \sum_{j=1}^n l_j \cdot \sin \theta_j + l \cdot \sin \theta_5 + \sum_{i=1}^4 l \cdot \sin \theta_i + h$$

考虑到计算的方便性, 本文采用穷举试算法进行求解, 即设定一个吃水深度 h , 带入之前的方程组得到一个水深 H , 将之与 18 m 进行比较, 程序自动筛选出最接近 18 m 相对应的吃水深度 (风向水平), 进而将改吃水深度代入方程组中可以求得四根钢管的倾斜角度受力情况, 锚链的形状以及钢桶的倾斜角度。对于浮标的游动区域 R 采用以下的公式进行求解:

$$R = \sum_{j=1}^n l_j \cdot \cos \theta_j + l \cdot \cos \theta_5 + \sum_{i=1}^4 l \cdot \cos \theta_i$$

在求解的过程中穷举吃水深度时以 **0.001 m** 为间隔（吃水深度的范围为 **0** 到 **2 m**），风速 **1 度** 为间隔（风速方向的范围为 **0** 到 **90 度**），锚链链环的个数为 **210** 个，用 C 语言编写相应的程序：Q1_12m_shape（出风速为 12 m/s 且风向水平下的钢管钢桶等的倾角数据等）、Q1_24m_shape（出风速为 24 m/s 且风向水平下的钢管钢桶等的倾角数据等）、Q1_12m（出风速为 12 m/s 下，不同风向下的浮标吃水深度散点数据）、Q1_24m（出风速为 24 m/s 下，不同风向下的浮标吃水深度散点数据）、Q1_12m_calculate（出风速为 12 m/s 下，不同风向下的游动范围半径散点数据）、Q1_24m_calculate（出风速为 24 m/s 下，不同风向下的游动范围半径散点数据）、Q1_12m_function_picture（出风速为 12 m/s 时偏差图数据）、Q1_24m_function_picture（出风速为 24 m/s 时偏差图数据），并用 MATLAB 作图：picture1.m（出吃水深度和游动范围随风向的变化情况图）、picture2.m（出风速为 12 m/s、24 m/s 和 36 m/s 时钢管和锚链的形状示意图）、picture3.m（风速为 12 m/s 时候的偏差图）、picture4.m（风速为 24 m/s 时候的偏差图）可以得到结果（附录包中有相应的附录包说明）。

同时当迭代计算锚链的角度的时候，如果出现了负值，则说明锚链出现负值的部分与海床接触，此时要重新迭代链环的个数进行计算，舍去拖在海床上的锚链部分。

4.4. 结果及其分析（趋势、最值）

根据模型求解方法对模型进行求解可以得到钢桶和各市钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域：

表 1 问题 1 求解表格

风速 $v(m/s)$	12	24
浮标吃水深度 h 米	0.736	0.751
从上到下四根钢管倾斜角度 α 度	89.0264	86.2878
	89.0207	86.2668
	89.0149	86.2456
	89.009	86.2241
钢桶倾斜角度 β 度	88.9957	86.1753
锚链末端夹角 γ 度	0	3.01811
浮标范围半径 R 米	13.8576	16.9451

如表 1 所示，在风速为 12 m/s 和 24 m/s 的时候对应的四根钢管的倾斜角度变化较小，基本接近 90 度，与风速为 12 m/s 相比，风速变大后钢管的倾斜角度相应地变小；对于钢桶，其倾斜角度和钢管的倾斜角度接近，风速增大其倾斜角度也相应地减小，且都符合角度要求即钢桶倾斜角度大于 85 度，即为与竖直面的夹角小于 5 度；对于锚链，由于在风速为 12 m/s 的时候锚链末端在海床上，所以倾斜角度为 0，而对于风速增大为 24 m/s 的时候锚链末端的倾角为 3 度左右，小于 16 度，符合要求；对于浮标的浮动范围，为风向和水平面平行的前提下得到的最大游动半径，而其浮动范围为以 **13.8576 m** 为半径的圆内，当风速为 24 m/s 的时候，浮动的范围为以 **16.9451 m** 为半径的圆内。

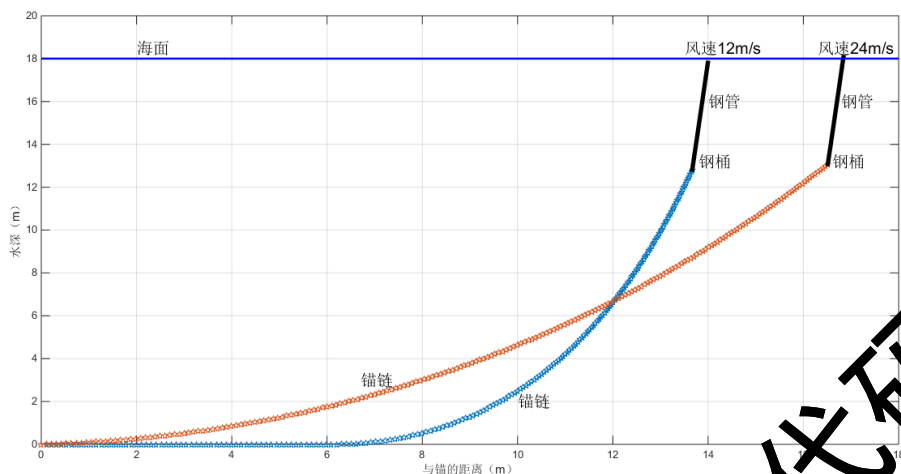


图 6 风速为 12 m/s 和 24 m/s 时钢管和锚链的形状示意图

如图所示，锚点的位置处于零点位置，风速为 12 m/s 的时候，一部分的锚链是拖在海床上的，而风速为 24 m/s 的时候，锚链则没有拖在海床上。经过计算得到拖在地上的部分长度为 6.3 m。对于钢管，每个钢管倾斜角度都不同，但是由于每个钢管的倾斜角度差别不大，故在图中拟合后显示基本上为直线。

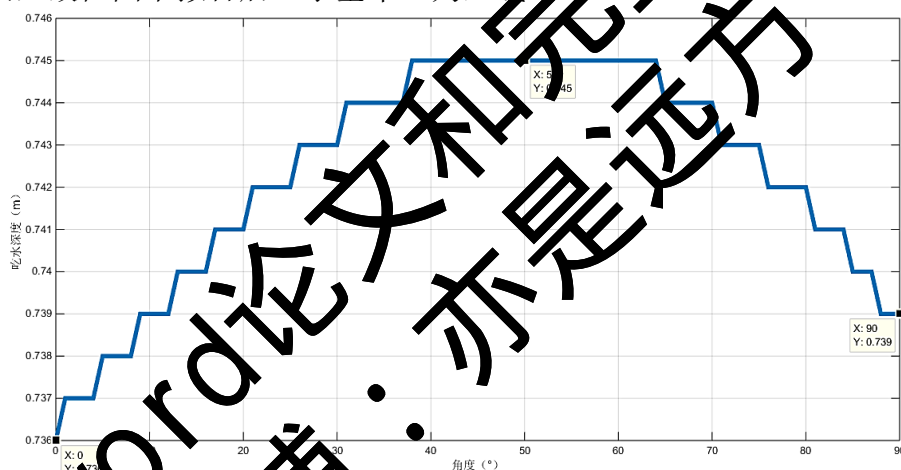


图 7 风速为 12 m/s 时浮标吃水深度随风向的变化情况（散点分段拟合趋势图）

如图 7 所示，该图为风速为 12 m/s 时浮标吃水深度随风向的变化情况。由于在穷举的时候会有间隔，所以最后的结果是不连续的，该图是通过散点图进行分段拟合得到，从该图可见，随着风向和水平面夹角的增大，吃水深度先增大后减小，在 40 度到 60 度之间达到最大值，当风向和水平面夹角为 0 度的时候即为求得的风向水平的吃水深度，而由于此物理模型考虑的是稳态状态，所以在风向和水平面垂直的时候吃水深度并不是最大的，浮标的运动区域相应地在风向和水平面垂直的时候也不是最小的。经过几何分析可以得到当海水淹没一部分浮标后浮标受风的底面变成了海平面和圆柱体的相交横截面，该横截面边缘和圆柱体上底面边缘相连形成的斜截面为一特殊平面，经验证如果风向法平面和该截面重合则会出现最大值。

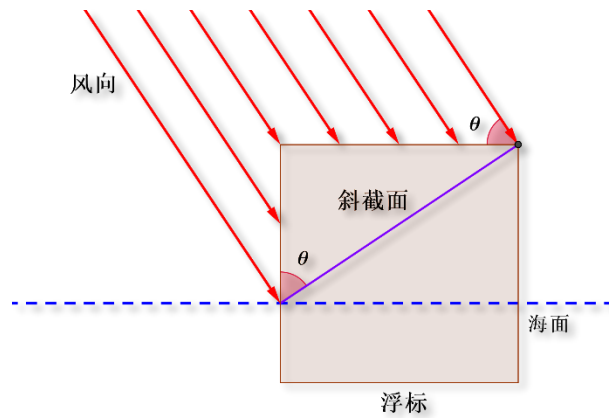


图 8 标吃水深度随风向的变化中特殊情况

如图所示，该风向下风向与水平角的夹角应该和斜截面与竖直面的夹角相同，即可验证风向法平面和该截面重合则会出现最大值。

验证：计算得到风速为 12 m/s 的时候吃水范围为 0.736~0.745 m，而计算得到斜截面与竖直面的夹角范围为 57.7~57.9 度，由风速为 12 m/s 时浮标吃水深度随风向的变化情况图可得在该范围内为最大值。同时计算得到风速为 24 m/s 的时候吃水范围为 0.751~0.786 m，而计算得到斜截面与竖直面的夹角范围为 58.0~58.7 度，由风速为 24 m/s 时浮标吃水深度随风向的变化情况图可得在该范围内为最大值。最大值即为图中所示 0.745 m。

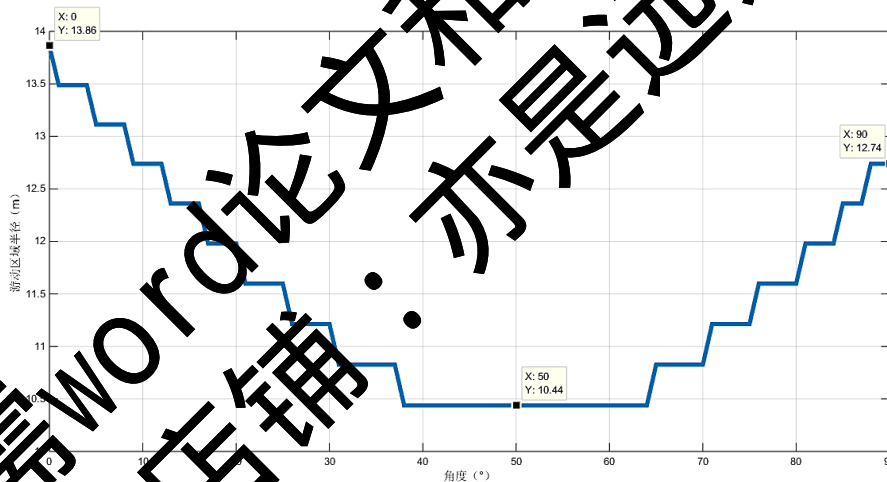


图 9 风速为 12 m/s 时浮标游动范围随风向的变化情况（散点分段拟合趋势图）

如图所示，该图中浮标的游动范围随着风向和水平面的夹角增大，先减小后增大。在 40 到 60 度之间会出最小值，而在 0 度和 90 度即为风向水平和风向垂直浮标的情况。最小值即为图中所示 10.44 m。

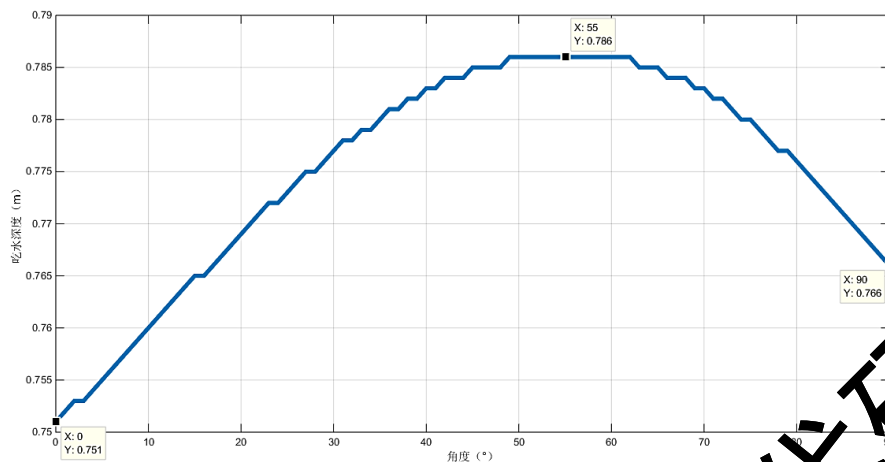


图 10 风速为 24 m/s 时浮标吃水深度随风向的变化情况（散点分段拟合趋势图）

如图所示，该图为风速为 24 m/s 时浮标吃水深度随风向的变化情况。随着风向和水平面夹角的增大，吃水深度先增大后减小，在 50 度到 60 度之间达到最大值，在风向和水平面夹角为 0 的时候即为求得的风向水平的吃水深度，最大值为图中所示 **0.786 m**。相对风速为 12 m/s 的时候，散点图分段拟合后的图形更加光滑。

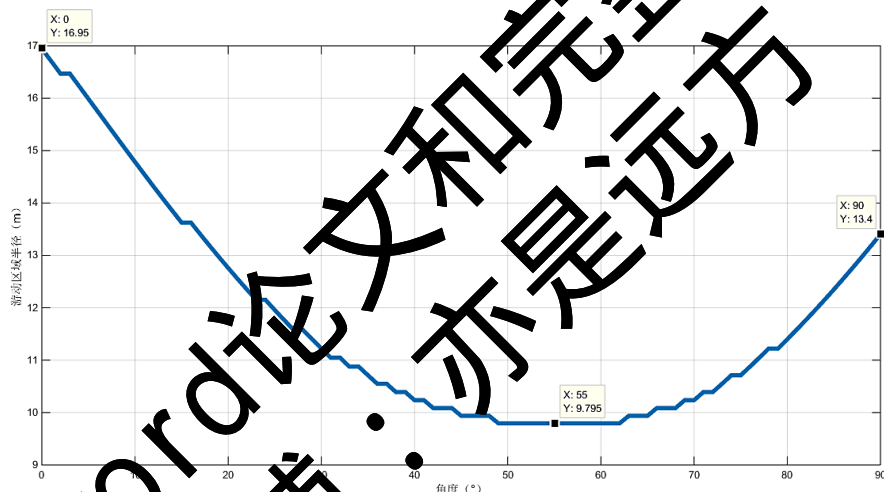


图 11 风速为 24 m/s 时浮标游动范围随风向的变化情况（散点分段拟合趋势图）

如图所示，当风速为 24 m/s 时散点图分段拟合后的图形更加光滑，最小值为 **9.795 m**。同时对比浮标的游动范围和吃水深度图可以发现，随着风向角度的增大，吃水深度的先增大后减小，和游动范围的先减小后增大的变化趋势是不一致的，甚至有相反的变化关系，这对第三目标规划时考虑吃水深度和游动范围的两个目标的多目标规划带来了困难。

关于风场的补充说明：本问模型结果中仅仅是为了显示吃水深度随风向的变化关系和浮标浮动范围随风向的变化关系（间接显示），实际上随着风向角度的变化钢管，锚链，钢桶的形状也会发生改变（甚至钢桶和锚链的角度可能不满足约束条件），但本文由于篇幅和时间关系并未一一展示，根据附录的程序或附件进行微调都可以获得。

4.5. 偏差分析

本方法求解的误差主要来源于进行试算的过程中，枚举的吃水深度的间隔并不是连续的，虽然作者在程序运行的过程中进行调试后间隔的减小基本对结果不会产生影响，但由于不连续性的存在，故给出偏差分析结果加以检验。

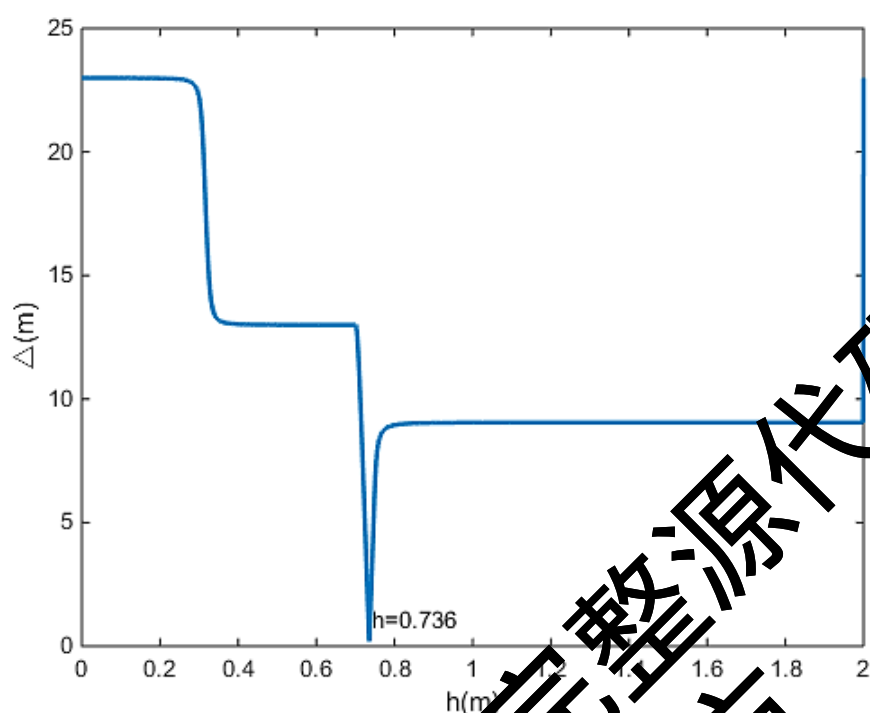


图 12 风速为 12 m/s 时吃水深度偏差图

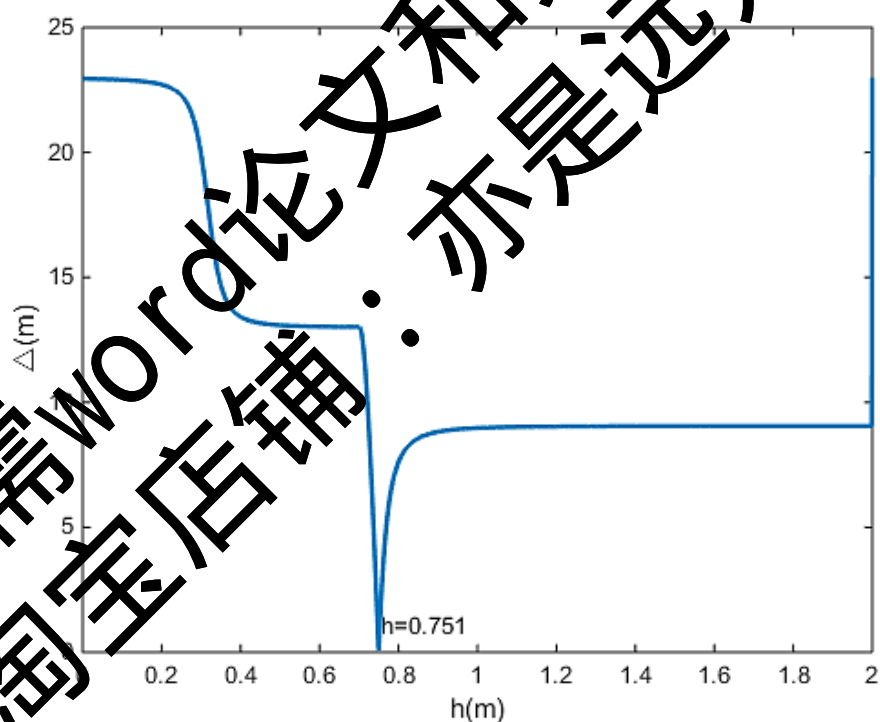


图 13 风速为 24 m/s 时吃水深度偏差图

从偏差图中可以看出,当风速(水平)为 12 m/s 的时候,偏差最小的吃水深度为 **0.736 m**; 风速(水平)为 24 m/s 的时候,偏差最小吃水深度为 **0.751 m**。

5 第二问的建模求解

5.1. 问题的分析

原问题：在问题 1 的假设下，计算海面风速为 36 m/s、风向为 0 度时钢桶和各节钢管的倾斜角度、锚链形状和浮标的游动区域。请调节重物球的质量，使得钢桶的倾斜角度不超过 5 度，锚链中锚点与海床的夹角不超过 16 度。

分析：从题目中可以得知，在风速为 36 m/s 的时候钢桶锚链的角度可能不满足设计要求，故首先要根据第一建立的模型求取 36 m/s 时候的钢桶和各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域，从而判断是否满足。如果满足角度约束则重物球的质量可以不变，如果不满足要求，要改变重物球的质量使钢桶的倾斜角度不超过 5 度，锚链中锚点与海床的夹角不超过 16 度。

5.2. 模型的求解和结果分析

本问的建模沿用第一问的物理方程，所不同的是重物球的质量也成为变量，其值并不知道，所以本问的重点就是找到符合约束条件的重物球的质量。作者先对风速为 36 m/s 时候的钢桶和各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域进行求解，C 语言：Q2_36m_shape（出风速为 36 m/s 时钢管和锚链的形状数据）、Q2_find_g（二分法找重物球质量）、Q2_36m_check（检验二分法找到的是否符合），MATLAB：picture5.m（出散点拟合图）得到：

表 2 风速为 36 m/s 时的钢桶和各节钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域

风速 $v(m/s^2)$	12	24	36
游标吃水深度 米	0.736	0.751	0.774
钢管倾斜角度 α 度	89.0264	86.2878	82.2461
	89.0207	86.2668	82.2048
	89.0149	86.2456	82.163
	89.009	86.2241	82.1208
钢桶倾斜角度 β 度	88.9957	86.1753	82.0253
锚链末端夹角 γ 度	0	3.01811	21.2719
浮标范围半径 r 米	13.8576	16.9451	18.1864

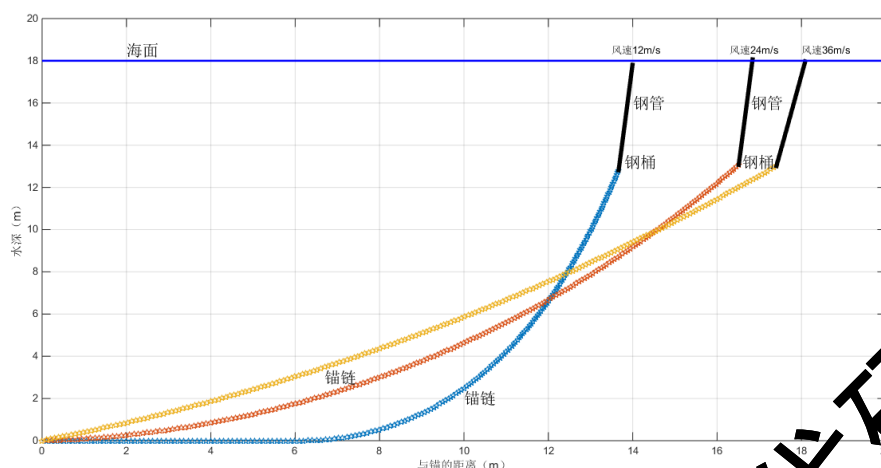


图 14 风速为 36 m/s 时钢管和锚链的形状示意图

从结果可以看出当风速为 36 m/s 时钢桶倾斜角度为 82.023° ，小于 85° 即和竖直方向的夹角大于 5° ，锚链末端夹角为 21.2719° ，即大于 16° ，使得设备无法正常工作，故需要对重物球的质量进行调节使得角度约束满足条件。同时，和之前求得的风速为 12 m/s 和风速为 24 m/s 时候的钢管倾斜角度也明显减小，游标的浮动范围也变得更大。由于此时重物球的质量为 1200 kg 不能满足要求，故试算 3000 kg 能否满足要求，试算结果如下：吃水深度为 1.314 m，钢桶倾斜角度为 87.9612° ，大于 85° 即和竖直方向的夹角小于 5° ，锚链末端夹角为 8.9065° ，即小于 16° 。显然，重物球的质量为 3000 kg 的时候可以满足要求，由于随着重物球质量的增大，钢桶水平角度增大（竖直角减小），锚链末端水平角度减小，基于该单调关系，本文采用二分法在 1200~3000 kg 之间筛选重物球的质量使得角度约束满足条件。

数列查找常用的算法有顺序查找算法、二分算法、索引查找算法、分块查找、散列查找。其中对有序序列采用的查找算法二分算法性能最好^[8]。

筛选结果如下：在重物球的质量为 2228 kg 时铁桶和竖直面的夹角达到 86.439° 而锚链末端和海床的夹角为 15.7161° ，为恰好满足角度的约束条件的最小重物球的质量。

5.4. 灵敏度分析

表 3 重物球的质量变化的灵敏度分析表

重物球质量 (kg)	2225	2226	2227	2228	2229	2230	2231
锚链末端夹角 (度)	16.16	16.41	16.06	15.72	15.36	15.01	14.66

从表格中可以看出，当重物球的质量大于等于 2228 kg 的时候，锚链末端和海床的夹角都满足要求，而当重物球的质量小于 2228 kg 的时候，锚链末端和海床的夹角不满足要求，故 2228 kg 为所求的准确重物球的质量。

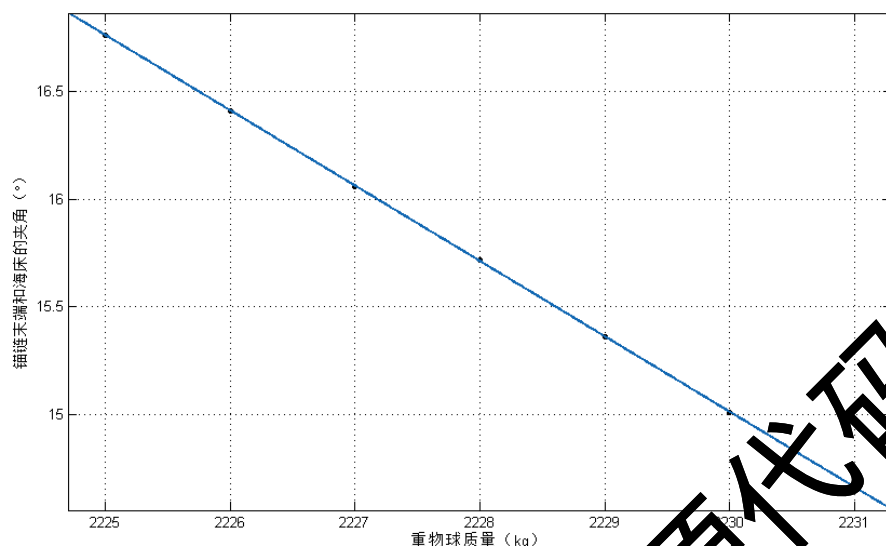


图 15 锚链和海床的夹角随重物球质量变化散点拟合图

由图表可见，随着重物球质量的增大，锚链末端和海床的夹角也逐渐变小，线性关系十分明显，拟合后的 $R\text{-square}$ 值为 1，方程为 $Y = -0.567955X + 1249.5$ ，从而说明 2228 kg 为所求重物球质量的准确值。

6 第三问的建模求解

6.1. 问题的分析

原题：由于潮汐等因素的影响，布放海域的水深介于 16~20 m 之间。布放点的海水速度最大可达到 1.5 m/s，风速最大可达到 36 m/s。请给出考虑风力、水流力和水深情况下的系泊系统设计，分析不同情况下钢桶、钢管的倾斜角度、锚链形状、浮标的吃水深度和游动区域。系泊系统的设计问题就是确定锚链的型号、长度和重物球的质量，使得浮标的吃水深度和游动区域及钢桶的倾斜角度尽可能小。

分析：系泊系统设计要求复杂，概括起来就是在四个约束条件下（钢桶轴线和竖直面的夹角小于 5 度，锚链末端和海床的夹角小于 16 度，水深介于 16~20 m 和锚链不接触海床），以及给定的风速、水速和风向基础下，选取使得浮标吃水深度和游动区域及钢桶的倾斜角度尽可能小的锚链型号，锚链环数目（锚链长度）和重物球质量。

显而易见，浮标吃水深度、游动区域及钢桶的倾斜角度是三个目标，该问题即为多目标规划问题，多目标规划问题的一般解法^[9]有理想点法、等级权重法、加权算数平均法、加权几何平均法、风险偏好系数法、乘除法和模糊规划法等。本文首先对三个目标进行分析，钢桶的倾斜角度是约束条件之一，可先满足该目标在 5 度以内，即将三个目标转化为两个目标，而剩下的两个目标在第一问的结果中看出，其变化规律不是正相关的，故无法同时使得此两个目标同时达到最小，本文采用改进的理想点法^[12]进行方案的选取。

接下来是锚链型号，锚链环数目（锚链长度）和重物球质量的筛选，本文采用基于二进制编码的遗传算法^[10-11]进行选取，通过对三者的范围进行合理的限定缩小搜索范围。最后得出符合条件的方案并编写 EXE 文件方便应用。（打开 Q3_step1 根据提示输入相应的值，运行后得到 h_{min} , h_{max} , R_{min} , R_{max} ，再打开 Q3_step2 根据提示输入之前得到的结果运行后得到方案）

6.2. 考虑海水作用力的单元模型的建立

受力分析：在第一问的基础上，考虑海水的作用力，重新对物体进行受力分析。同样先将体系分成各个部分分别进行考虑，然后画出各个部分的受力分析图，进而对各个部分分别用不同的方法进行受力分析，从而列出基本的受力平衡方程和力矩平衡方程，联立得到的方程组可以得到总的以浮标吃水深度为主要自变量的平衡方程组，带入数据求解。

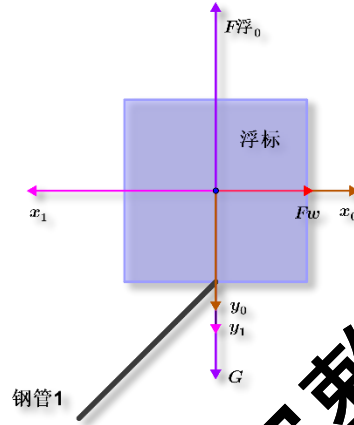


图 16 浮标受力分析图 (考虑海水作用力)

浮标受力分析：浮标受到风力，竖直向上的浮力 $F_{\text{浮}0}$ ，竖直向下的重力 G ，钢管的水平拉力 x_1 和竖直拉力 y_1 的作用以及海水水平的作用力（设其水平向右为正）而平衡。基于之前的问题分析，第一根钢管对浮标的作用力为第二根钢管上端所受力的反作用力，分解为水平方向 x_1 和竖直方向 y_1 。对于风力根据第一问的讨论有：

$$\begin{cases} F_{\text{风}} = \sqrt{x_0^2 + y_0^2} \\ F_{\text{风}} = 0.625 \times s v^2 \\ x_0 = F_{\text{风}} \cdot \cos \theta_0 \\ y_0 = F_{\text{风}} \cdot \sin \theta_0 \end{cases}$$

对于截面积，浮标投影面即物体在风向法平面的投影面，它由三部分组成，第一部分为圆柱横截面在法平面的投影面 s_1 ，第二部分为圆柱下底面（圆柱与海平面相交的横截面）在法平面的投影面 s_2 ，第三部分为圆柱上底面右半圆在法平面的投影面 s_3 ，其中 s_2 、 s_3 之和为圆柱底面面积。设风向和水平面的夹角为 θ （本文中所有的 θ 均为物体与水平面的夹角），则有：

$$\begin{cases} s = s_1 \cdot \cos \theta + s_2 \cdot \sin \theta + s_3 \cdot \sin \theta \\ s_2 + s_3 = \pi r^2 \end{cases}$$

由于浮标水平则截面面积仅仅为被浸没的那部分圆柱的轴截面的面积。

海水对浮标的水平作用力需要加入讨论得到：

$$\begin{cases} F_w = 374 \cdot 2R \cdot h \cdot v^2 \\ F_{\text{浮}0} = \rho_{\text{水}} \cdot g \cdot \pi R^2 \cdot h \\ x_1 = F_w + x_0 \\ F_{\text{浮}0} = G + y_1 + y_0 \end{cases}$$

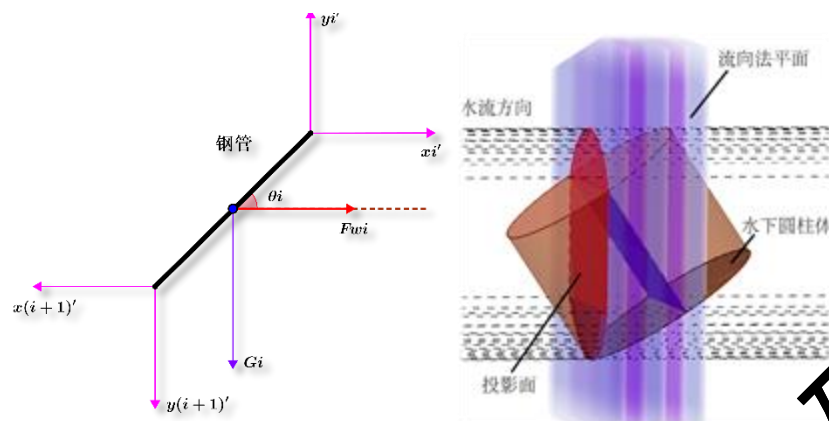


图 17 钢管受力分析（考虑海水作用力）和在海水流向法平面投影面图

第 i 钢管受力分析：四根同样的钢管受力的情况基本相同。竖直方向上受到重力和浮力的作用（二者合力为 G_i ）；水平方向上受相邻钢管的作用力和海水的水平作用力 F_{wi} 而钢管的两端则受到钢管和下一根钢管的作用力 $x_{(n+1)}$, $y_{(n+1)}$ 。基于假设将其水平和竖直进行投影得到如图 2 中所示下标的分力。对于钢管的截面积，根据第一问对风向的分析，类比可以得到，海水均匀流就相当于风，其截面面积也是三个部分，对于钢管，由于其长度过长，两个底面的面积相对可以忽略不计得：

$$\begin{cases} F_{wi} = 374 \cdot v^2 \cdot 0.05 \cdot \sin \theta_i \\ x_{i+1} = x_i + F_{wi} \\ y_{i+1} = y_i - G_i \\ G_i \cdot \frac{l}{2} \cdot \cos \theta_i = x_i \cdot l \cdot \sin \theta_i = y_i \cdot l \cdot \cos \theta_i \end{cases}$$



图 18 锚链结构示意图

表 4 补充后的锚链信息汇总表

型号	锚链径 D(mm)	外长 L(mm)	外宽 B(mm)	单位长度的质量 (kg/m)	平均横截面积 S (m ²)
I	12.5	78	43	3.2	1.7×10^{-3}
II	17.5	105	60	7	3.2×10^{-3}
III	24	120	82	12.5	4.9×10^{-3}
IV	30	150	102	19.5	7.6×10^{-3}
V	36	180	122	28.12	1.09×10^{-2}

$$\begin{cases} S_1 = 2 \cdot (L-B) \cdot (B-2 \cdot D) + \pi \cdot \left(\frac{B}{2}\right)^2 - \pi \cdot \left(\frac{B-2 \cdot D}{2}\right)^2 \\ S_2 = D \cdot (L-D) + \pi \cdot \left(\frac{D}{2}\right)^2 \\ S = \frac{S_1 + S_2}{2} \end{cases}$$

由图表对照计算可得，对于二维坐标系下锚链分为正对型（图示左边锚链环）和侧对型（图示右边锚链环）两种，正对型的链环正对面积为 s_1 ，侧对型链环的正对面积为 s_2 ，求取平均值得到为 s 。计算最大型号的锚链环所受的海水浮力与钢管所受浮力相差一个数量级，本文对锚链的受力采用均值法计算，即每个锚链均受到水平方向海水的作用力为 s ，即不考虑锚链角度变化带来的其所受海水水平作用力的影响。

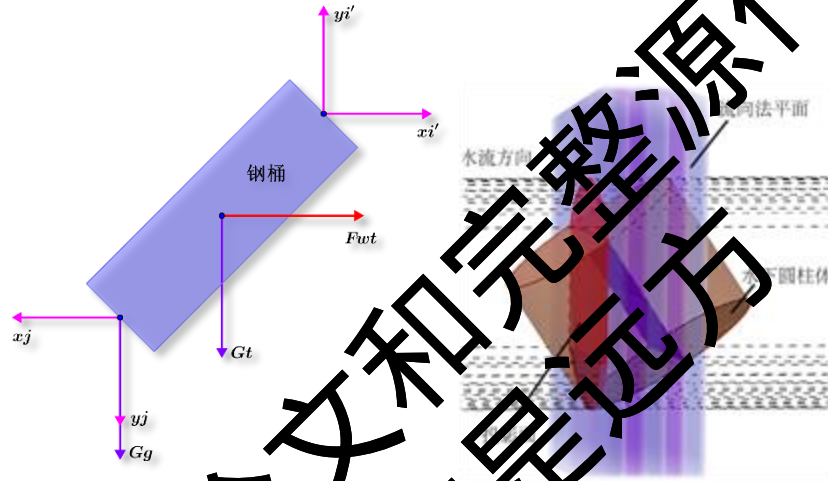


图 19 钢桶受力分析（考虑海水作用力）和在海水流向法平面投影面图

钢桶受力分析：钢桶受到的力有重力 G ，第四根钢管和锚链的作用力 x_5, y_5, x_j, y_j 和重物球的拉力 G_g 以及海水的水平作用力。对于钢桶的截面积，包含了上下底面在海水流向法平面的投影面积和圆柱轴截面在法平面上的投影面积，从而根据受力平衡和力矩平衡得到：

$$\begin{cases} F_{wt} = 3.4 \times 10^2 \cdot (0.3 \cdot 1 \cdot \sin \theta_5 + \pi \cdot 0.15^2 \cdot \cos \theta_5) \\ x_j = x_5' + F_{wt} \\ y_j + G_g + G_t = y_5' \\ G_t \cdot \frac{l}{2} \cdot \cos \theta_5 + x_5' \cdot l \cdot \sin \theta_5 = y_5' \cdot l \cdot \cos \theta_5 \end{cases}$$

海水深度和吊标的游动范围计算依然按照以下公式：

$$\begin{aligned} H &= \sum_{j=1}^n l_j \cdot \sin \theta_j + l \cdot \sin \theta_5 + \sum_{i=1}^4 l \cdot \sin \theta_i + h \\ R &= \sum_{j=1}^n l_j \cdot \cos \theta_j + l \cdot \cos \theta_5 + \sum_{i=1}^4 l \cdot \cos \theta_i \end{aligned}$$

6.3. 模型的求解

关于程序，C 语言：Q3_find_minmax（找正负理想点）、Q3_find_w_GA（遗传算法求最优解）、Q3_shape（出钢管的倾斜角度数据）；MATLAB：calculate1（计算锚链环的投影面积）。

关于物理模型的近似：
在模型计算的过程中如果：

$$\begin{cases} F_{wt} = 374 \cdot v^2 \cdot (0.3 \cdot 1 \cdot \sin \theta_5 + \pi \cdot 0.15^2 \cdot \cos \theta_5) \\ x_j = x_5' + F_{wt} \\ y_j + G_g + G_t = y_5' \\ G_t \cdot \frac{l}{2} \cdot \cos \theta_5 + x_5' \cdot l \cdot \sin \theta_5 = y_5' \cdot l \cdot \cos \theta_5 \end{cases}$$

此公式在求解的过程中会出现含有三角函数的高次方程，编写程序计算的过程中出现问题，故对海水作用力进行假设，在锚链的受力分析中沿用之前的近似的结果，对于钢桶和钢管，根据第一问和第二问的求解结果，其几乎垂直，即钢桶和竖直面的夹角要求小于 5 度，而钢桶和竖直面的夹角大于四根钢管和水平面的夹角，故可以考虑将钢桶和钢管做近似垂直处理，从而其在均匀海水流速平面方向上的投影为轴截面。

求解流程图：

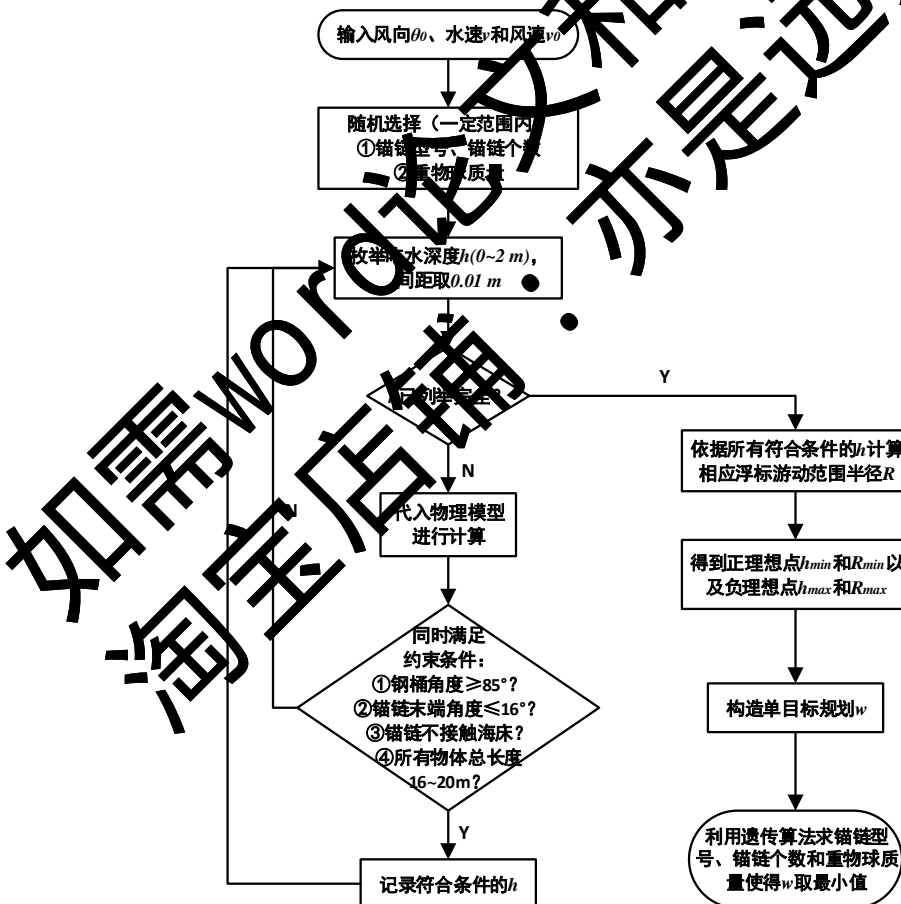


图 20 第三问求解流程图

由流程图，首先要给定风速、风向和海水流速等环境因素的值，其中，风速的范围为 **0~36 m/s**，风向为 **0~90 度**，海水流速为 **0~1.5 m/s**。程序内部设定的选取范围如下：锚链的型号为 **1~5**，个数范围为 **60~500**（60 个对应用的锚链个数最小的情况，即锚链垂直，对应海水深度为 16 m，用的形状为 5 号锚链；500 个位本文选取的最多的锚链个数），重物球质量范围为 **500~4000 kg**。程序自动从中选取锚链的型号，数目以及重物球质量，然后枚举吃水深度，吃水深度的范围为 **0~2 m**，枚举间隔为 **0.1 m**，倘若枚举的 h 带入物理模型求解的结果满足两个角度约束条件，即铁桶和垂直方向的夹角小于 5 度，锚链末端和海床的夹角小于 16 度，海水深度在 **16~20 m** 之间，且由于锚链末端拖在海床上浪费，所以还要满足锚链没有拖在海床上的条件，则记录符合条件的吃水深度 h ，并记录将此 h 带入物理模型后得到的浮标游动范围 s ，由理想点法得到理想点 h_{min} 和 h_{max} ，以及 R_{min} 和 R_{max} 。

紧接着，进而计算目标函数贴近度为：

$$w = \sqrt{(h - h_{min})^2 + (R - R_{min})^2}$$

由于原来的贴近度公式中并未考虑 h 的范围和 R 的范围相差较大，所以本文对二者进行标准化，从而得到新的贴近度计算公式：

$$w = \sqrt{\left(\frac{h - h_{min}}{h_{max} - h_{min}}\right)^2 + \left(\frac{R - R_{min}}{R_{max} - R_{min}}\right)^2}$$

从而将多目标规划转化为单一的目标贴近度。

而后运用遗传算法在程序内部设定的选取范围内进行搜索：

个体编码：

遗传算法的运算对象是表示个体的符号串，所以必须把决策变量（锚链型号、锚链个数和重物质量）采用无符号二进制整数的编码方式表示为符号串。本文采用 24 位无符号二进制编码表示个体基因组，其中前 3 位无符号二进制整数表示锚链型号，即为 **I、II、III、IV、V** 五种锚链，其后 9 位无符号二进制整数表示锚链个数，即为 **0~511** 个锚链，最后 12 位无符号二进制整数表示重物球质量，即为 **0~4095 kg** 的重物。例如，001 000 000 001 000 000 000 001 表示一个型号 **I** 锚链，重物球质量 1 kg 的个体。

个体评价：

遗传算法内以个体适应度的大小来评定个体的优劣程度，从而决定其遗传机会的大小。由于目标函数为非负值，且是以求函数最小值为优化目标，故而可以利用目标函数值作为个体的适应度。

遗传算法种群操作方式：

(1) 交叉计算：交叉运算是遗传算法中产生新个体的主要操作过程，随机交换两个相邻个体之间的部分染色体产生交叉个体。首先，产生随机数 K ，确定两个相邻个体染色体交换的位置；其次，相互交换相邻染色体之间的部分基因；最后，遍历 $X(t)$ 代所有相邻个体，保证产生 $N-1$ 个交叉个体。

(2) 变异计算：变异运算是对于个体的某一个或某一些基因座上的基因值进行改变，也是产生新个体的一种操作方法。首先，产生随机数，确定各个个体的基因变异位置；然后，将变异点的原有基因值取反；最后，遍历 $X(t)$ 代所有个体，保证产生 N 个变异个体。

(3) 选择计算：选择计算把当前种群中适应度较高的个体按某种规则遗传到下一代

种群中,要求适应度较高的个体将有更多的机会遗传到下一代。对 $X(t)$ 代父代所有个体、交叉计算得到所有个体和变异计算得到所有个体组成的个体集合的适应度进行排序, 选出适应度排名靠前的 N 个个体组成新一代种群 $X(t+1)$ 。

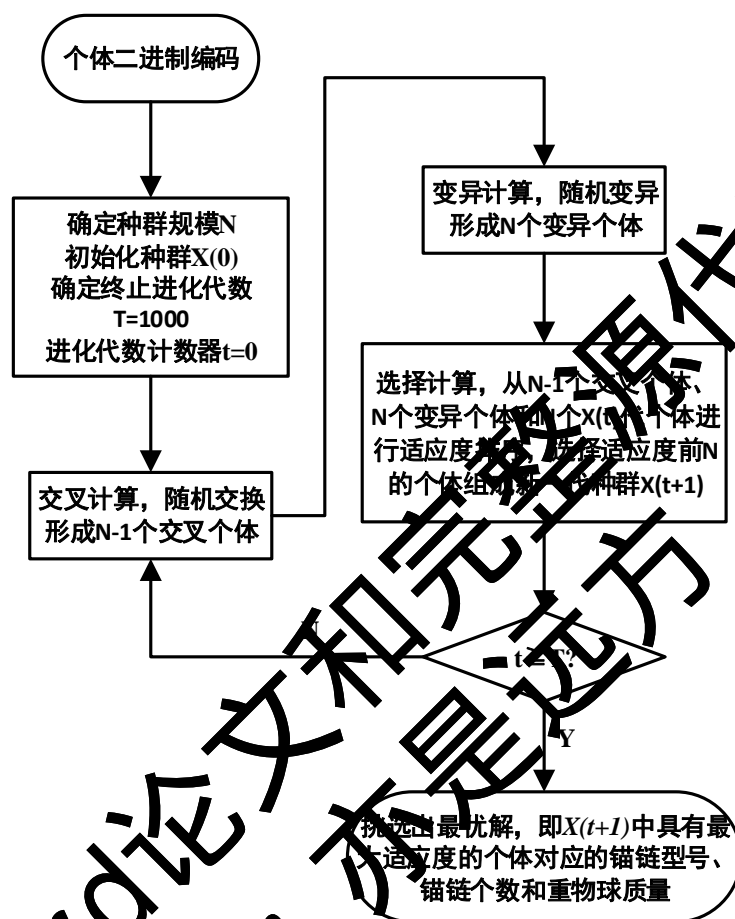


图 21 遗传算法流程图

遗传算法实现步骤

✚ 初始化种群

确定种群规模 $N=10$, 确定终止进化代数 $T=1000$; 随机生成 N 个个体作为初始种群 X_0 ; 进化代数计数器 $t=0$ 。

✚ 种群进化

(1) 交叉计算: 遍历 $X(t)$ 代所有个体, 随机交换两个相邻个体的部分基因, 形成 $N-1$ 个交叉个体。

(2) 变异计算: 遍历 $X(t)$ 代所有个体, 随机使每个个体的某部分基因发生变异, 形成 N 个变异个体。

(3) 选择计算: 从上述所形成的 $N-1$ 个交叉个体, N 个变异个体和原本 $X(t)$ 代的 N 个父代个体进行适应度排序, 选择适应度前 N 的个体组成新一代种群 $X(t+1)$ 。

✚ 终止检验

如进化代数 t 已达到终止进化代数 T , 则输出 $X(t+1)$ 中具有最大适应度的个体对应的锚链型号、锚链个数和重物球质量作为最优解, 终止运算; 否则置 $t=t+1$ 并转至种群进

化的步骤。

程序运行最终求解得到锚链型号，锚链环数目（锚链长度），重物球质量，同时输出浮标吃水深度和浮标游动区域。带入求解钢管钢桶形状的程序进行求解表 6 结果。

6.4. 结果及其分析

针对风速为 12 m/s，海水流速为 1.5 m/s，风向为 0 度的情况进行计算，求得 $h_{min}=1.24$ m； $h_{max}=1.454$ m； $R_{min}=12.32784$ m； $R_{max}=96.56464$ m。

表 5 遗传算法两种繁殖代数对应的结果表 1

	第 1 次 1000 代 运算	第 2 次 1000 代	第 3 次 1000 代	第 4 次 10000 代
锚链型号（类）	5	4	4	3
锚链环数目（个）	91	112	130	162
重物球质量（kg）	2594	2700	2626	2816
浮标吃水深度（m）	1.3	1.3	1.3	1.3
浮标游动区域半径（m）	12.34	13.24	15.29	16.83

注：C 语言程序运行 1000 代大概需要几分钟，运行 10000 代大概要十几分钟

从结果中可以看出，锚链型号可以选择 3、4 或 5 型号；锚链环数目分布在 125.25 个左右；重物球质量集中在 2684 个左右；浮标吃水深度保持 1.3 m 不变；浮标游动区域半径集中在 14.425 左右（求取平均值）。

表 6 遗传算法两种繁殖代数对应的结果表 2

	第 1 次 1000 代运算	第 2 次 1000 代运算	第 3 次 1000 代运算	第 4 次 10000 代运算
第 1 根钢管倾斜角度	85.72	85.72	85.72	85.72
第 2 根钢管倾斜角度	85.63	85.63	85.63	85.63
第 3 根钢管倾斜角度	85.54	85.54	85.54	85.54
第 4 根钢管倾斜角度	85.45	85.45	85.45	85.45
钢桶倾斜角度	85.15	85.16	85.15	85.15
锚链末端倾斜角度	11.59	15.95	15.85	11.15

将表 4 的数据带入另一个程序可以计算得出四根钢管的倾斜角度、钢桶的倾斜角度和锚链末端的倾斜角度，从而得表 5，从该表中不难发现，四根钢管的倾斜角度分别稳定在 85.72、85.63、85.54 和 85.45 度。而钢桶的倾斜角度会发生变化但是也不是很大，锚链末端的倾斜角度会发生变化但是都在要求的范围之内。

6.5. 稳定性分析

表 7 遗传算法稳定性分析

	第 1 次 1000 代运算	第 2 次 1000 代运算	第 3 次 1000 代运算	1 次 10000 代 运算
第 1 根钢管倾斜角度	85.72	85.72	85.72	85.72
第 2 根钢管倾斜角度	85.63	85.63	85.63	85.63
第 3 根钢管倾斜角度	85.54	85.54	85.54	85.54
第 4 根钢管倾斜角度	85.45	85.45	85.45	85.45
钢桶倾斜角度	85.15	85.16	85.15	85.15
锚链末端倾斜角度	11.59	15.95	15.85	11.15

分析得到的表格可见系统的形态基本不发生变化, 仅仅钢桶的倾斜角度发生了 **0.01** 度的变化, 其变化比例为 **0.012%**, 锚链末端由于其变化范围极大所以浮动范围相对也较大, 总体来说遗传算法的稳定性较好。

7 模型&算法优缺点和改进

7.1. 模型&算法优点

第一问

考虑风向对浮标吃水深度、浮动区域及其它形态的影响并找出了相应的影响关系曲线, 在完成题目要求的情况下进行了较为深入的讨论, 尤其是对产生最大吃水深度和最浮动范围的情况进行分析, 使模型的建立更为完整。

在物理建模的过程中, 针对钢管、钢桶和锚链的受力采用坐标系分解的方法, 对基于集中质量力的多边形近似法进行改进, 从而更为准确地描述稳态时刻的体系情况。

在计算的过程之中, 采用试算法进行计算, 通过程序调试不同的试算间隔, 使计算结果更为精确, 并对最后的曲线进行拟合, 得到变化规律。

第二问

根据第一问的讨论, 在计算的过程中采用二分法简化程序搜索, 从而可以更为快捷地得到最终结果。

第三问

考虑海水的流动, 多种约束和多重目标的综合问题, 最终给出了求解程序 EXE 从而为实际应用的过程带来了方便。

在物理建模的过程中, 针对海水流动的问题, 类比第一问对风向的讨论, 使得物理问题更为明确, 同时在合理的简化下实现程序的优化求解问题。

在求解的过程中对于多目标规划问题采用改进的理想点法使得结果更为精确, 同时搜索的算法采用较为稳定的遗传算法进行求解, 最终求解出在给定条件下的结果。

7.2. 模型&算法缺点

第一问求解的过程中试算法的结果并不连续, 必须通过拟合来得到规律曲线

第三问的物理方程含三角函数高次方程求解较为困难, 在合理假设的基础上进行求解会和真实值有一定的误差。

7.3. 模型&算法改进

第一问中对随风向变化而改变的浮标吃水深度和浮标游动区域的变化规律曲线可以

通过其他方式求取其连续变化规律。

第三问可以考虑海水的方向变化,可以通过改进算法计算含有三角函数的高次方程。

8 参考文献

- [1] 刘志刚, 何炎平. FPSO 转塔系泊系统的技术特征及发展趋势[J]. 中国海洋平台, 2006, 21(5):1-6.
- [2] 沈琪, 徐召利. 海洋工程中单点系泊系统的应用探讨[J]. 船舶标准化与质量, 2015(6).
- [3] 于欢, 李运鹏. 圆钢管连接方式及应用现状[J]. 山西建筑, 2016, 42(16).
- [4] 钟平安, 邹长国, 李伟, 等. 水库防洪调度分段试算法及应用[J]. 水利水电科技进展, 2003, 23(6):21-23.
- [5] 何勇. 最优关联群试算法[J]. 浙江大学学报(工学版), 1994(2):324-330.
- [6] 朱美玲, 徐永梅. 浅析试算法在项目决策与分析中的应用[J]. 知识经济, 2015(6):103-103.
- [7] GB/T 549-2008, 电焊锚链[S]
- [8] 邹国霞, 何南. 一种改进的二分查找算法[J]. 软件导刊, 2010, 09(3).
- [9] 杨桂元, 郑亚豪. 多目标决策问题及其求解方法研究[J]. 数学的实践与认识, 2012, 42(2):108-115.
- [10] 葛继科, 邱玉辉, 吴春明, 等. 遗传算法研究综述[J]. 计算机应用研究, 2008, 25(10):2911-2916.
- [11] 苏日娜. 基于二进制编码的遗传算法的研究[J]. 宁波工程学院学报, 2005, 17(2):11-13.
- [12] 艾正海. 关于多目标决策问题的理想点法研究[J]. 西南交通大学, 2007.

9 附录

一、Q1_12m_shape（出风速为 12 m/s 且风向水平下的钢管钢桶等的倾角数据等）

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0;
double sum2=0;
double ans;
double windj;
double Abs(double a)
{
    if(a<0)
        return -a;
    return a;
}
void fb()
{
    Vf=12;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
```

```

}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    cout<<"四根钢管夹角:"<<endl;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        cout<<theat[i-1]<<endl;
        sum2+=L*cos(theat[i-1]/180*pi);
    }
    double Gg=1200*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gg-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum2+=L*cos(Theat/180*pi);
    cout<<"钢桶夹角:"<<endl;
    cout<<Theat<<endl;
}
void chain()
{
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=0.105;
    double G=7*g;
    cout<<"链节夹角:"<<endl;
    for(int i=1;i<=10;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        if(jiao[i]<=0)
        {
            cout<<0<<endl;
            sum2+=l;
            continue;
        }
        cout<<jiao[i]<<endl;
        sum2+=l*cos(jiao[i]/180*pi);
    }
}

```

```

    }
}
int main()
{
    windj=0;
    h=0.736;
    fb();
    steel();
    chain();
    cout<<"游动范围半径:"<<endl;
    cout<<sum2<<endl;
    return 0;
}

```

二、Q1_24m_shape（出风速为 24 m/s 且风向水平下的钢管钢桶等的倾角数据等）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double L=1,Rfb=1,rou=1025,lfb=1000;
double Ffengx,Ffengy,Ff,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double r[MAXN];
double r1[MAXN];
double theat;
double Fx[MAXN],Fy[MAXN];
double sum=0;
double sum1;
double ans;
double windj;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}

```

```

void fb()
{
    Vf=24;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
}

```

```

void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    cout<<"四根钢管夹角:"<<endl;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G)/(x[i-1]*L))/pi*180;
        cout<<theat[i-1]<<endl;
        sum2+=L*cos(theat[i-1]*pi);
    }

```

```

    double Gg=1200*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gg-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt)/(x[5]*L))/pi*180;
    sum2+=L*cos(Theat/180*pi);
    cout<<"前桶夹角:"<<endl;
    cout<<Theat<<endl;
}

```

```

void chain()
{
    Fx[0]=0;
    Fy[0]=Yj;
    double l=0.105;
    double G=7*l*g;
    cout<<"锚链夹角:"<<endl;
    for(int i=1;i<=210;i++)
    {
        Fy[i]=Fy[i-1]-G;

```

```

        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        if(jiao[i]<=0)
        {
            cout<<0<<endl;
            sum2+=1;
            continue;
        }
        cout<<jiao[i]<<endl;
        sum2+=l*cos(jiao[i]/180*pi);
    }
}
int main()
{
    windj=0;
    h=0.751;
    fb();
    steel();
    chain();
    cout<<"游动范围半径:"<<endl;
    cout<<sum2<<endl;
    return 0;
}

```

三、Q1_12m（出风速为12m/s下，不同风向下的浮标吃水深度散点数据）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1000
#define g 9.8
#define pi 3.1415
double eps=1e-999;
double L=1,Kfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;

```



```

double Fx[MAXN],Fy[MAXN];
double sum=0;
double ans;
double windj;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
void fb()
{
    Vf=12;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        cout<<theat[i]<<endl;
        y[i]=y[i-1]+L*sin(theat[i]/180*pi);
    }
    double Gt=2200*g;
    double Gc=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum+=L*sin(Theat/180*pi);
    //cout<<Theat<<endl;
}
void chain()
{

```

```

    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=0.105;
    double G=7*l*g;
    //cout<<G<<endl;
    for(int i=1;i<=210;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        //cout<<jiao[i]<<endl;
        if(jiao[i]<=0)
            break;
        sum+=l*sin(jiao[i]/180*pi);
    }
}
int main()
{
    for(double i=0;i<=90;i++)
    {
        windj=i/180*pi;
        INF=99999;
        h=0;
        while(1)
        {
            if(h>2)
                break;
            sum=0;
            h=h+0.001;
            el();
            steel();
            chain();
            if(Abs(sum-18)<INF)
            {
                INF=Abs(sum-18);
                ans=h;
            }
        }
        cout<<ans<<endl;
    }
    return 0;
}

```

四、Q1_24m（出风速为 24 m/s 下，不同风向下的浮标吃水深度散点数据）

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double INF=99999;
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0;
double ans;
double windj;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
void steel()
{
    Vf=24;
    double S=(2*h+2*cos(windj)+pi*1*1*sin(windj));
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{

```

```

double G=10*g-rou*g*pi*0.025*0.025*L;
for(int i=2;i<=5;i++)
{
    y[i]=y[i-1]-G;
    x[i]=x[i-1];
    theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
    //cout<<theat[i-1]<<endl;
    sum+=L*sin(theat[i-1]/180*pi);
}
double Gg=1200*g;
double Gt=100*g-rou*g*pi*0.15*0.15*L;
Yj=y[5]-Gg-Gt;
Xj=x[5];
Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
sum+=L*sin(Theat/180*pi);
//cout<<Theat<<endl;
}
void chain()
{
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=0.105;
    double G=7*l*g;
    //cout<<G<<endl;
    for(int i=1;i<=21;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiaod[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        //cout<<jiaod[i]<<endl;
        if(jiaod[i]<0)
            break;
        sum+=L*sin(jiaod[i]/180*pi);
    }
}
int main()
{
    for(double i=0;i<=90;i++)
    {
        windj=i/180*pi;
        INF=99999;
        h=0;
    }
}

```

```

while(1)
{
    if(h>2)
        break;
    sum=0;
    h=h+0.001;
    fb();
    steel();
    chain();
    if(Abs(sum-18)<INF)
    {
        INF=Abs(sum-18);
        ans=h;
    }
}
cout<<ans<<endl;
}
return 0;
}

```

五、Q1_12m_calculate（出风速为 12 m/s 下，不同风向下的游动范围半径散点数据）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define  $\alpha$  9.
#define  $\beta$  3.14159
double  $\alpha$ ,  $\beta$ , Rfb=1,  $\alpha$ ou=12, Mfb=1000;
double Ffengx, Ffengy, Ffd, h, Vf, Vs;
double theat[MAXN], jiao[MAXN];
double x[MAXN];
double y[MAXN];
double t[MAXN];
double Xj, Yj;
double Theat;
double Fx[MAXN], Fy[MAXN];
double sum=0;
double sum2;
double ans;
double windj;

```

```

double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
void fb()
{
    Vf=12;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    //cout<<"四根钢管夹角:"<<endl;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        //cout<<theat[i-1]<<endl;
        sum2+=1*cos(theat[i-1]/180*pi);
    }
    double Gt=1200*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gt;
    Xj=x[5];
    Theat=atan((Yj*L-Gt*L/2)/(Xj*L))/pi*180;
    sum2+=1*cos(Theat/180*pi);
    //cout<<"钢桶夹角:"<<endl;
    //cout<<Theat<<endl;
}
void chain()
{
    Fx[0]=Xj;
    Fy[0]=Yj;
}

```

```

double l=0.105;
double G=7*l*g;
//cout<<"锚链夹角:"<<endl;
for(int i=1;i<=210;i++)
{
    Fy[i]=Fy[i-1]-G;
    Fx[i]=Fx[i-1];
    jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
    if(jiao[i]<=0)
    {
        //cout<<0<<endl;
        sum2+=1;
        continue;
    }
    //cout<<jiao[i]<<endl;
    sum2+=l*cos(jiao[i]/180*pi);
}
}
int main()
{
    freopen("12m.in","r",stdin);
    freopen("12m.out","w",stdout);
    for(int i=0;i<=90;i++)
    {
        sum2=0;
        cin>>h;
        windj=i/180*pi;
        fb();
        //cout<<sum2<<endl;
    }
    return 0;
}

```

六、Q1_2m_calculate（出风速为 24 m/s 下，不同风向下的游动范围半径散点数据）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005

```



```

#define g 9.8
#define pi 3.14159
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0;
double sum2;
double ans;
double windj;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
void fb()
{
    Vf=24;
    double S=(2-h)*2*cos(windj)+pi*L*cos(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*L;
    double Gfb=1000*g;
    x[0]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=(rou*g*pi*0.025*0.025*L);
    //cout<<"四根钢管夹角:"<<endl;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        //cout<<theat[i-1]<<endl;
    }
}

```

```

        sum2+=L*cos(theat[i-1]/180*pi);
    }
    double Gg=1200*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gg-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum2+=L*cos(Theat/180*pi);
    //cout<<"钢桶夹角:"<<endl;
    //cout<<Theat<<endl;
}
void chain()
{
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=0.105;
    double G=7*l*g;
    //cout<<"锚链夹角:"<<endl;
    for(int i=1;i<=210;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        if(jiao[i]<=0)
        {
            //cout<<0<<endl;
            sum2+=l;
            continue;
        }
        //cout<<jiao[i]<<endl;
        sum2+=l*cos(jiao[i]/180*pi);
    }
}
int main()
{
    freopen("24m.in","r",stdin);
    freopen("24m.out","w",stdout);
    for(int i=0;i<=90;i++)
    {
        sum2=0;
        cin>>h;
        windj=i/180*pi;
    }
}

```

```

        fb();
        steel();
        chain();
        cout<<sum2<<endl;
    }
    return 0;
}

```

七、Q1_12m_function_picture（出风速为 12 m/s 时偏差图数据）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double INF=99999;
double L=1,Rfb=1,rou=1025,Mfb=100;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0;
double abs;
double abs2;
double Abs(double a)
{
    if(a>0)
        return a;
    else
        return -a;
}
void fb()
{
    Vf=12;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);

```

```

Ffu=rou*g*pi*Rfb*Rfb*h;
double Gfb=1000*g;
x[1]=Ffengx;
y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        //cout<<theat[i-1]<<endl;
        sum+=L*sin(theat[i-1]/180*pi);
    }
    double Gg=1200*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gg-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum+=L*sin(Theat/180*pi);
    //cout<<Theat<<endl;
}
void chain()
{
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=0.05;
    double G=7*l*g;
    //cout<<G<<endl;
    for(int i=1;i<=20;i++)
    {
        Fx[i]=Fx[i-1]-G;
        Fy[i]=Fy[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        //cout<<jiao[i]<<endl;
        if(jiao[i]<=0)
            break;
        sum+=l*sin(jiao[i]/180*pi);
    }
}

```

```

int main()
{
    freopen("12m_pic.out","w",stdout);
    windj=0;
    INF=99999;
    h=0;
    while(1)
    {
        if(h>2)
            break;
        sum=0;
        h=h+0.001;
        fb();
        steel();
        chain();
        cout<<Abs(sum-18)<<endl;
        if(Abs(sum-18)<INF)
        {
            INF=Abs(sum-18);
            ans=h;
        }
    }
    return 0;
}

```

八、Q1_24m_function_picture (出风速为 24 m/s 时偏差图数据)

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double INF=99999;
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;

```

```

double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0;
double ans;
double windj;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
void fb()
{
    Vf=24;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        cout<<theat[i-1]<<endl;
        sum+=L*sin(theat[i-1]/180*pi);
    }
    double Gg=1200*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gg-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum+=L*sin(Theat/180*pi);
    //cout<<Theat<<endl;
}
void chain()

```

```

{
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=0.105;
    double G=7*l*g;
    //cout<<G<<endl;
    for(int i=1;i<=210;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        //cout<<jiao[i]<<endl;
        if(jiao[i]<=0)
            break;
        sum+=l*sin(jiao[i]/180*pi);
    }
}

int main()
{
    freopen("24m_pic.out","w",stdout);
    windj=0;
    INF=99999;
    h=0;
    while(1)
    {
        if(h>2)
            break;
        sum=0;
        h=h+0.1;
        Fx[0]=0;
        Fy[0]=0;
        chain(0);
        cout<<abs(sum-18)<<endl;
        if(Abs(sum-18)<INF)
        {
            INF=Abs(sum-18);
            ans=h;
        }
    }
    return 0;
}

```

九、Q3_find_minmax（找正负理想点）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<ctime>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double INF=99999;
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0,sum2=0;
double ans;
double windj;
double Fw,Vw,weigh;
double type[10][5];
double Ds[10];
int Type,number;
void fb()
{
    double s=(2-h)*pi*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Fw=374*2*h*Vw*Vw;
    Ffu=rou*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx+Fw;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    for(int i=2;i<=5;i++)

```



```

{
    Fw=374*Vw*Vw*0.05;
    theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L+Fw*L/2))/pi*180;
    y[i]=y[i-1]-G;
    x[i]=x[i-1]+Fw;
    sum+=L*sin(theat[i-1]/180*pi);
    sum2+=L*cos(theat[i-1]/180*pi);
    //cout<<theat[i-1]<<endl;
}
double Gg=weigh*g;
double Gt=100*g-rou*g*pi*0.15*0.15*L;
Fw=374*Vw*Vw*0.3;
Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L+Fw*L/2))/pi*180;
Yj=y[5]-Gg-Gt;
Xj=x[5]+Fw;
sum+=L*sin(Theat/180*pi);
sum2+=L*cos(Theat/180*pi);
//cout<<Theat<<endl;
}
void chain()
{
    Fw=Ds[Type]*374*Vw*Vw;
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=type[Type][1];
    double G=type[Type][0]*l*g;
    for(int i=1;i<=number;i++)
    {
        Fx[i]=Fy[i-1]-G;
        Fy[i]=Fx[i-1]+Fw;
        thet[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l+Fw*l/2))/pi*180;
        if(jiao[i]<=0)
        {
            jiao[i]=0;
            sum=0;
            sum2+=l;
            //cout<<jiao[i]<<endl;
            continue;
        }
        sum+=l*sin(jiao[i]/180*pi);
        sum2+=l*cos(jiao[i]/180*pi);
        //cout<<jiao[i]<<endl;
    }
}

```

```

    }
}
void initial()
{
    type[1][1]=0.078;
    type[1][2]=3.2;
    type[2][1]=0.105;
    type[2][2]=7;
    type[3][1]=0.120;
    type[3][2]=12.5;
    type[4][1]=0.150;
    type[4][2]=19.5;
    type[5][1]=0.180;
    type[5][2]=28.12;
    Ds[1]=0.0017;
    Ds[2]=0.0032;
    Ds[3]=0.0049;
    Ds[4]=0.0076;
    Ds[5]=0.0109;
}
double hmin=99999,hmax=-1;
double Rmin=99999,Rmax=-1;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
int main()
{
    TIME=0.0001;
    srand(time(NULL));
    initial();
    cout<<"请输入风的速度 (m/s) : "<<endl;
    cin>>V;
    cout<<"请输入风向与水平面的夹角 (°) : "<<endl;
    cin>>windj;
    cout<<"请输入海水流速 (m/s) : "<<endl;
    cin>>Vw;
    for(int i=1;i<=1000;i++)
    {
        cout<<i<<endl;
    }
}

```

```

Type=rand()%5+1;
weigh=500+rand()%2500;
number=60+rand()%500;//存疑
h=0;
while(1)
{
    if(h>2)
        break;
    sum=0;
    sum2=0;
    h=h+0.01;
    fb();
    steel();
    chain();

    if(sum+h>=16&&sum+h<=20&&Theat>=85&&jiao[number]<=0&&jiao[number]>=0)
    {
        if(h<hmin)
            hmin=h;
        if(h>hmax)
            hmax=h;
        if(sum2<Rmin)
            Rmin=sum2;
        if(sum2>Rmax)
            Rmax=sum2;
    }
}

cout<<"min"<<" " <<Rmin<<endl;
cout<<"max"<<" " <<Rmax<<endl;
return 0;
}

```

十一、Q3_find_w_GA (遗传算法求最优解)

```

#include<iostream>
#include<stdio>
#include<ctime>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;

#define MAXN 1005
#define g 9.8

```

```

#define pi 3.14159
struct node
{
    int num[50];
    double H;
    double S;
    double w;
};
node gene[MAXN];
double INF=99999;
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0,sum2=0;
double windj;
double Fw,Vw;
double type[10][5];
double hmin,hmax,Rmin,Rmax;
double deltah,deltaR;
int n=100;//种群数量
int dai=1000;//代数
int temp;
double res[10];
void main()
{
    type[1][1]=0.078;
    type[1][2]=5.07;
    type[2][1]=0.105;
    type[2][2]=7;
    type[3][1]=0.120;
    type[3][2]=12.5;
    type[4][1]=0.150;
    type[4][2]=19.5;
    type[5][1]=0.180;
    type[5][2]=28.12;
    Ds[1]=0.0017;

```

```

Ds[2]=0.0032;
Ds[3]=0.0049;
Ds[4]=0.0076;
Ds[5]=0.0109;
}
void value_before(int Type,int number,int weigh)
{
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Fw=374*2*h*Vw*Vw;
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx+Fw;
    y[1]=Ffu-Gfb-Ffengy;
    double G=10*g-rou*g*pi*0.025*0.025*L;
    for(int i=2;i<=5;i++)
    {
        Fw=374*Vw*Vw*0.05;
        theat[i-1]=atan((y[i-1]*L-G*(L/2))/(x[i-1]*L+Fw*L/2))/pi*180;
        y[i]=y[i-1]-G;
        x[i]=x[i-1]+Fw;
        sum+=L*sin(theat[i-1]/180*pi);
        sum2+=L*cos(theat[i-1]/180*pi);
    }
    double Gg=weigh*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Fw=374*Vw*Vw*0.03;
    Theat=atan(y[5]*L-Gt*(L/2))/(x[5]*L+Fw*L/2))/pi*180;
    Yj=y[5]-Gg-Gt;
    Xj=x[5]+Fw;
    sum+=L*sin(Theat/180*pi);
    sum2+=L*cos(Theat/180*pi);
    Fw=Ds[Type]*374*Vw*Vw;
    Fx[0]=Fx;
    Fy[0]=Yj;
    double l=type[Type][1];
    G=type[Type][2]*l*g;
    for(int i=1;i<=number;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1]+Fw;
    }
}

```

```

        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l+Fw*l/2))/pi*180;
        if(jiao[i]<=0)
        {
            jiao[i]=0;
            sum+=0;
            sum2+=1;
            continue;
        }
        sum+=l*sin(jiao[i]/180*pi);
        sum2+=l*cos(jiao[i]/180*pi);
    }
}
double value(int m,int Type,int number,int weigh)
{
    double ans=999999;
    h=0;
    while(1)
    {
        if(h>2)
            break;
        sum=0;
        sum2=0;
        h=h+0.1;
        value_before(Type,number,weigh);

        if(sum+h>=16&&sum+h<=20&&Theat>=85&&jiao[number]<=16&&jiao[number]>=0)
        {
            if(sqrt(((h-hmin)/deltah)*((h-hmin)/deltah)+((sum2-
            Rmin)/deltaR)*(sum2-Rmin)/deltaR))<ans)
            {
                gene[m].h=h;
                gene[m].s=sum2;
                an=sqrt(((h-hmin)/deltah)*((h-hmin)/deltah)+((sum2-
            Rmin)/deltaR)*(sum2-Rmin)/deltaR));
            }
        }
    }
    return ans;
}
void bianma(int i,int Type,int number,int weigh)
{
    for(int k=3;k>=1;k--)

```

```

{
    gene[i].num[k]=Type%2;
    Type=Type/2;
}
for(int k=12;k>=4;k--)
{
    gene[i].num[k]=number%2;
    number=number/2;
}
for(int k=24;k>=13;k--)
{
    gene[i].num[k]=weigh%2;
    weigh=weigh/2;
}
}
void chushi()
{
    for(int i=1;i<=n;i++)
    {
        int Type=rand()%5+1;
        int weigh=500+rand()%350;
        int number=60+rand()%40;
        bianma(i,Type,number,weigh);
    }
}
void jiaocha()//交叉操作
{
    for(int i=0;i<=n-1;i++)
    {
        i=rand()%n+1;
        for(int j=1;j<=k;j++)
            gene[temp].num[j]=gene[i].num[j];
        for(int j=k+1;j<=n;j++)
            gene[temp].num[j]=gene[i+1].num[j];
        temp++;
    }
}
void bianyi()//变异
{
    for(int i=1;i<=n;i++)
    {
        int k=rand()%n+1;

```

```

        for(int j=1;j<=n;j++)
            gene[temp].num[j]=gene[i].num[j];
        gene[temp].num[k]=1-gene[i].num[k];
        temp++;
    }
}
int Pow(int m)
{
    int ans=1;
    for(int i=1;i<=m;i++)
        ans*=2;
    return ans;
}
void jiema()
{
    for(int i=1;i<=temp-1;i++)
    {
        int Type=0,number=0,weigh=0;
        for(int j=1;j<=3;j++)
            Type+=gene[i].num[j]*Pow(3-j);
        for(int j=4;j<=12;j++)
            number+=gene[i].num[j]*Pow((12-j));
        for(int j=13;j<=20;j++)
            weigh+=gene[i].num[j]*Pow(22-j);
        gene[i].w=value(i,Type,number,weigh);
    }
}
bool cmp(node a,node b)
{
    return a.w<b.w;
}
void ga()
{
    chushu();
    for(int i=1;i<=dai;i++)
    {
        temp=n+1;
        jiaocha();
        bianyi();
        jiema();
        sort(&gene[1],&gene[temp],cmp);
        //cout<<i<<endl;
    }
}

```



```

    }
}
int main()
{
    srand(time(NULL));
    cout<<"请输入风的速度 (m/s) : "<<endl;
    cin>>Vf;
    cout<<"请输入风向与水平面的夹角 (°) : "<<endl;
    cin>>windj;
    cout<<"请输入海水流速 (m/s) : "<<endl;
    cin>>Vw;
    initial();
    hmin=1.24;
    hmax=1.45;
    deltah=hmax-hmin;
    Rmin=12.3278;
    Rmax=96.5646;
    deltaR=Rmax-Rmin;
    GA();
    int Type=0,number=0,weigh=0;
    for(int j=1;j<=3;j++)
        Type+=gene[1].num[j]*Pow((3-j));
    for(int j=4;j<=12;j++)
        number+=gene[1].num[j]*Pow((1-j));
    for(int j=13;j<=24;j++)
        weigh+=gene[1].num[j]*Pow((24-j));
    cout<<Type<<endl<<number<<endl<<weigh<<endl;
    cout<<gene[1].S<<" "<<gene[1].S<<endl;
    return 0;
}

```

十一、Q2Shape（出钢管的倾斜角度数据）

```

#include<iostream>
#include<stdio>
#include<string>
#include<math>
#include<algorithm>
#include<ctime>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double INF=99999;

```

```

double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0,sum2=0;
double ans;
double windj;
double Fw,Vw,weigh;
double type[10][5];
double Ds[10];
int Type,number;
void fb()
{
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Fw=374*2*h*Vw*Vw;
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx+Fw;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    cout<<"钢管的角度"<<endl;
    double G=10*g-rou*g*pi*0.025*0.025*L;
    for(int i=1;i<=5;i++)
    {
        Fw=374*Vw*Vw*0.05;
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L+Fw*L/2))/pi*180;
        y[i]=y[i-1]-G;
        x[i]=x[i-1]+Fw;
        sum+=L*sin(theat[i-1]/180*pi);
        sum2+=L*cos(theat[i-1]/180*pi);
        cout<<theat[i-1]<<endl;
    }
    double Gg=weigh*g;

```

```

double Gt=100*g-rou*g*pi*0.15*0.15*L;
Fw=374*Vw*Vw*0.3;
Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L+Fw*L/2))/pi*180;
Yj=y[5]-Gg-Gt;
Xj=x[5]+Fw;
sum+=L*sin(Theat/180*pi);
sum2+=L*cos(Theat/180*pi);
cout<<"钢桶的角度:"<<endl;
cout<<Theat<<endl;
}
void chain()
{
    cout<<"锚链的角度:"<<endl;
    Fw=Ds[Type]*374*Vw*Vw;
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=type[Type][1];
    double G=type[Type][2]*l*g;
    for(int i=1;i<=number;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1]+Fw;
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l+Fw*l/2))/pi*180;
        if(jiao[i]<=0)
        {
            jiao[i]=0;
            sum+=0;
            sum2+=l;
            cout<<jiao[i]<<endl;
            continue;
        }
        sum+=l*sin(jiao[i]/180*pi);
        sum2+=l*cos(jiao[i]/180*pi);
        cout<<jiao[i]<<endl;
    }
}
void initial()
{
    type[1][1]=0.078;
    type[1][2]=3.2;
    type[2][1]=0.105;
    type[2][2]=7;
}

```

```

type[3][1]=0.120;
type[3][2]=12.5;
type[4][1]=0.150;
type[4][2]=19.5;
type[5][1]=0.180;
type[5][2]=28.12;
Ds[1]=0.0017;
Ds[2]=0.0032;
Ds[3]=0.0049;
Ds[4]=0.0076;
Ds[5]=0.0109;
}
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
int main()
{
    Type=3;
    number=162;
    weigh=2816;
    h=1.3;
    initial();
    cout<<"请输入风的速度(m/s):"<<endl;
    cin>>Vf;
    cout<<"请输入风向与水平面的夹角(°):"<<endl;
    cin>>alpha;
    cout<<"请输入海水流速(m/s):"<<endl;
    cin>>Vw;
    fb=1;
    steel();
    chain();
    cout<<"锚链范围半径:"<<endl;
    cout<<sum2<<endl;
    return 0;
}

```

十二、calculate1（计算锚链环的投影面积）

```

L=[78 105 120 150 180];
B=[43 60 82 102 122];

```

```

D=[12.5 17.5 24 30 36];
S=[0 0 0 0 0];
for i=1:1:5
    S1=2*(L(i)-B(i))*(B(i)-2*D(i)) + pi*(B(i)/2)^2 - pi * ((B(i)-
2*D(i))/2)^2;
    S2=D(i)*(L(i) - D(i)) + pi * (D(i)/2)^2;
    S(i) = (S1 + S2)/2 * 10^(-6);
end
S

```

十三、Q2_36m_shape（出风速为 36 m/s 时钢管和锚链的形状数据）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double vx=0;
double vy=0;
double as;
double windj;
double Ab;double a)
{
    if(a>0)
        return a;
    return -a;
}
void fb()
{
    Vf=36;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);

```

```

Ffengx=0.625*S*Vf*Vf*cos(windj);
Ffengy=0.625*S*Vf*Vf*sin(windj);
Ffu=rou*g*pi*Rfb*Rfb*h;
double Gfb=1000*g;
x[1]=Ffengx;
y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    cout<<"四根钢管夹角:"<<endl;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        cout<<theat[i-1]<<endl;
        sum2+=L*cos(theat[i-1]/180*pi);
    }
    double Gg=1200*g;
    double Gt=100*g-rou*g*pi*0.05*0.05*L;
    Yj=y[5]-Gg-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum2+=L*cos(Theat/180*pi);
    cout<<"钢桶夹角:"<<endl;
    cout<<Theat<<endl;
}
void chuan()
{
    Fx[1]=Xj;
    Fy[1]=Yj;
    double l=0.1;
    double G=7*l*g;
    cout<<"钢筒夹角:"<<endl;
    for(int i=1;i<=210;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        if(jiao[i]<=0)
        {

```

```

        cout<<0<<endl;
        sum2+=1;
        continue;
    }
    cout<<jiao[i]<<endl;
    sum2+=1*cos(jiao[i]/180*pi);
}
}
int main()
{
    windj=0;
    h=0.774;
    fb();
    steel();
    chain();
    cout<<"游动范围半径:"<<endl;
    cout<<sum2<<endl;
    return 0;
}

```

十四、Q2_36m_check（检验二分法找到的是否符合）

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
#define MAXN 1005
#define  $\alpha$  9.
#define  $\beta$  3.14159
double L, R, Rb=1, r0u=1/2, Mfb=1000;
double Ffengx, Ffongj, Ffd, h, Vf, Vs;
double theat[MAXN], jiao[MAXN];
double x[10];
double r[10];
double t[MAXN];
double Xj, Yj;
double Theat;
double Fx[MAXN], Fy[MAXN];
double sum=0;
double sum2;
double ans;
double windj;

```

```

double gball=2228;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
void fb()
{
    Vf=36;
    double S=(2-h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffengx=0.625*S*Vf*Vf*cos(windj);
    Ffengy=0.625*S*Vf*Vf*sin(windj);
    Ffu=rou*g*pi*Rfb*Rfb*h;
    double Gfb=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Gfb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.02*0.02*L;
    cout<<"四根钢管夹角:"<<endl;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        cout<<theat[i-1]<<"°<<endl;
        sum2+=L*cos(theat[i-1]/180*pi);
    }
    double Gg=gball*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-G-Gg;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum2+=L*cos(Theat/180*pi);
    cout<<"钢桶夹角:"<<endl;
    cout<<Theat<<endl;
}
void chain()
{
    Fx[0]=Xj;

```



```

    Fy[0]=Yj;
    double l=0.105;
    double G=7*l*g;
    cout<<"锚链夹角:"<<endl;
    for(int i=1;i<=210;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i-1]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        if(jiao[i]<=0)
        {
            cout<<0<<endl;
            sum2+=l;
            continue;
        }
        cout<<jiao[i]<<endl;
        sum2+=l*cos(jiao[i]/180*pi);
    }
}

int main()
{
    windj=0;
    h=1.082;
    fb();
    steel();
    chain();
    cout<<sum2<<endl;
    return 0;
}

```

十五、plot5.m (出散点拟合图)

```

x=[2220,2226,2227,2228,2229,2230,2231];
y=[15.76,16.42,16.53,15.72,15.36,15.01,14.66];
plot(x,y,'o');
cftool;
xlabel('重物球质量 (kg) ');
ylabel('锚链末端和海床的夹角 (°) ');

```

十六、Q2_find_g (二分法找重物球质量)

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>

```

```

#include<algorithm>
using namespace std;
#define MAXN 1005
#define g 9.8
#define pi 3.14159
double INF=99999;
double L=1,Rfb=1,rou=1025,Mfb=1000;
double Ffengx,Ffengy,Ffu,h,Vf,Vs;
double theat[MAXN],jiao[MAXN];
double x[10];
double y[10];
double T[MAXN];
double Xj,Yj;
double Theat;
double Fx[MAXN],Fy[MAXN];
double sum=0;
double ans;
double windj;
int gball;
double Abs(double a)
{
    if(a>0)
        return a;
    return -a;
}
void fb()
{
    Vf=36;
    double S=0.5*h)*2*cos(windj)+pi*1*1*sin(windj);
    Ffu=0.625*S*Vf*cos(windj);
    Ffengy=0.625*S*Vf*sin(windj);
    Ffb=rou*g*L*Rfb*Rfb*h;
    double G=1000*g;
    x[1]=Ffengx;
    y[1]=Ffu-Ffb-Ffengy;
}
void steel()
{
    double G=10*g-rou*g*pi*0.025*0.025*L;
    for(int i=2;i<=5;i++)
    {
        y[i]=y[i-1]-G;
    }
}

```

```

        x[i]=x[i-1];
        theat[i-1]=atan((y[i-1]*L-G*L/2)/(x[i-1]*L))/pi*180;
        //cout<<theat[i-1]<<endl;
        sum+=L*sin(theat[i-1]/180*pi);
    }
    double Gg=gball*g;
    double Gt=100*g-rou*g*pi*0.15*0.15*L;
    Yj=y[5]-Gg-Gt;
    Xj=x[5];
    Theat=atan((y[5]*L-Gt*L/2)/(x[5]*L))/pi*180;
    sum+=L*sin(Theat/180*pi);
    //cout<<Theat<<endl;
}
void chain()
{
    Fx[0]=Xj;
    Fy[0]=Yj;
    double l=0.105;
    double G=7*l*g;
    //cout<<G<<endl;
    for(int i=1;i<=210;i++)
    {
        Fy[i]=Fy[i-1]-G;
        Fx[i]=Fx[i-1];
        jiao[i]=atan((Fy[i]*l-G*l/2)/(Fx[i-1]*l))/pi*180;
        //cout<<jiao[i]<<endl;
        if(jiao[i]<=0)
        {
            break;
        }
        sum+=L*sin(jiao[i]/180*pi);
    }
}
void check(double H)
{
    h=H;
    fb();
    steel();
    chain();
}
int main()
{
    int gballmin=1300;
    int gballmax=4000;

```

```

windj=0;
while(1)
{
    if(gballmax-gballmin<=1)
        break;
    gball=(gballmax+gballmin)/2;
    INF=99999;
    h=0.5;
    while(1)
    {
        if(h>2)
            break;
        sum=0;
        h=h+0.001;
        fb();
        steel();
        chain();
        if(Abs(sum-18)<INF)
        {
            INF=Abs(sum-18);
            ans=h;
        }
    }
    check(ans);
    if(Theat>=85&&Jiao[10]<=16)
        gballmax=gball;
    else gballmin=gball;
}
cout<<ans<<endl;
return 0;
}

```