

Summary

We are discussing how to search the lost plane that may dropping in the ocean like Malaysia Airlines MH370. We divide our paper into three parts

First part, we will determine the circle of the possible crashed plane. By expansion and application of Markov Process, we will narrow the circle so that the searching power can be focused. It's the base of all searching works. Without predict the dropping position, searching work can't be started.

Second part, we will build an ocean current model to estimate the effects of ocean current to the black boxes or other plane parts. In this model we have two assumption and build two models. The first one is for the coastal sea and the other one is for the open sea. We are building this because the electronic devices like sonars have different searching area in these two situation. We will use the area find before and by connecting the Bayes Theorem and the area we get in the first part to create a grip map with possibility.

Third part, we have determined the area that need to be searched and by building a searching model based on the optimal search theory. We need to decide the arrangement and use of different search methods including planes, boats and radars. We will what is the most efficient combination of these methods to search in every small area is.

Key words: splash-down MH370 search

摘要

我们正在讨论如何寻找丢失的飞机可能在海洋般的马来西亚航空公司 MH370 坠。我们把我们的论文分为三个部分

第一部分，我们将确定可能的坠毁飞机的圆。通过马尔可夫过程的扩展和应用，缩小了搜索范围，提高了搜索能力。这是所有搜索工作的基础。没有预测下落位置，搜索工作不能启动。

第二部分，我们将建立一个海流模型来估计海流对黑匣子或其他平面部分的影响。在这个模型中，我们有两个假设，并建立两个模型。第一个是沿海海域，另一个是公海。我们建立这个因为电子设备如声纳在这两种情况有不同的搜索区域。我们将使用之前发现的区域，并通过连接贝叶斯定理和我们在第一部分得到的区域创建一个有可能的抓地力地图。

第三部分，确定了需要搜索的区域，并建立了基于最优搜索理论的搜索模型。我们需要决定安排和使用不同的搜索方法，包括飞机，船只和雷达。我们将什么是最有效的组合，这些方法在每一个小面积搜索。

关键词：溅落 MH370 搜寻

1.Introduction

1.1Background

2014 年 3 月 8 日凌晨 1 时 22 分(世界标准时间 3 月 7 日下午 17 时 22 分)一班从马来西亚吉隆坡前往中国北京的波音 777-200ER, 经马来西亚与越南空域交界处 IGARI 导航点后, 与空中交通管制失去联系。客机失踪 16 天后的 3 月 24 日, 马来西亚总理纳吉布宣布, 马航 MH370 航班“终结于”南印度洋, 至今未找到飞机残骸, 但机上的 227 名乘客、2 名飞行员和 10 名乘务人员可能已全部遇难。马来西亚航空 370 号班机空难被认为是有史以来“最离奇”的飞机失联案列。在客机失踪后期间, 各国积极参与搜寻工作, 搜寻与救援工作涉及来自 25 个国家的 160 艘/架舰机(包括 65 架飞机和 95 艘舰船)及专家。尽管投入大量人力、物力和财力, 搜救工作并没有得到任何进展, 最终飞机失联而告终。因此有效的预测失事区域和合理安排搜索工作是寻找失事飞机和援救失事人员的好法子。建立起一个很好援救策略是寻找失事飞机和救援的有效途径。

1.2Problem Restatement

PROBLEM B: Searching for a lost plane Recall the lost Malaysian flight MH370. Build a generic mathematical model that could assist "searchers" in planning a useful search for a lost plane feared to have crashed in open water such as the Atlantic, Pacific, Indian, Southern, or Arctic Ocean while flying from Point A to Point B. Assume that there are no signals from the downed plane. Your model should recognize that there are many different types of planes for which we might be searching and that there are many different types of search planes, often using different electronics or sensors. Additionally, prepare a 1-2 page non-technical paper for the airlines to use in their press conferences concerning their plan for future searches

本文针对已经失联并且确定掉进某块海域的飞机, 通过建立了确定飞机高度求飞机可能沉降范围, 和最优的搜救方案模型。

模型一: 飞机从一定高度发生故障失去动力在空中作类平抛运动坠入海里, 根据牛顿第二定律, 建立飞机从高度 H 降落的模型:

$$\begin{cases} m \frac{d^2 x}{dt^2} = -k_1 \left(\frac{dx}{dt} \right)^2 \\ m \frac{d^2 y}{dt^2} = -mg + k_1 \left(\frac{dy}{dt} \right)^2 + k_2 \left(\frac{dy}{dt} \right)^2 \end{cases}$$

本文通过给定飞机不同的飞行高度 H : $1000m \sim 10000m$ 多组数据, 得到飞机到海平面对应水平位移 $10180 \sim 10204m$ 和对应的水平速度 $45.71 \sim 46.00m/s$ 。

模型二: 飞机进入海洋, 考虑无洋流作用, 飞机继续作类平抛运动。根据动力学知识建立模型:

$$\begin{cases} m \frac{d^2 x}{dt^2} = -k_3 \frac{dx}{dt} \\ m \frac{d^2 y}{dt^2} = -mg + k_3 \frac{dy}{dt} + F_{bou} \end{cases}$$

得到飞机在无洋流作用下，在海中的运动轨迹如图 6。

模型三：对模型二进行优化，考虑有洋流作用，建立模型：

$$\begin{cases} x = \frac{(v_{x1} + v_1)m + k_3 x_1}{k_3} - \frac{(v_{x1} + v_1)m e^{-\frac{k_3}{m}t}}{k_3} \\ y = \frac{gm^2 - F_{bou}m + F_{bou}k_3t - gk_3mt}{k_3^2} - \frac{e^{-\frac{k_3}{m}t}(-F_{bou}m + gm^2)}{k_3^2} \\ z = \frac{v_2}{k_3} - \frac{v_2 m e^{-\frac{k_3}{m}t}}{k_3} \end{cases}$$

得到飞机在海里运动轨迹如图 9。通过给定不同的海底深度，确定了飞机沉降最大面积 $S_b = 109627.2m^2$ ，考虑有洋流或助推或阻碍作用，得到了飞机在总的搜索范围内，最可能出现的沉降范围如图 12。

problem analysis

针对题意，需要解决两个问题，归纳如下：

问题一，坠落的飞机失去联系时会有一个最后信号点，同时结合飞机预定的航线，分析坠落的飞机可能跌落的海洋区域。在假设的条件下，对失去动力的飞机作受力分析，对飞机到落海的过程，考虑了不同类型的坠机，建立起通用动力学的微分方程模型确定区域求解。由于飞机跌落海中收到来至海洋的各方面因素影响。其中洋流对于坠落的飞机是影响最大的，并且由于各个海洋洋流情况不同，飞机落海后确定的区域受到洋流流速的影响，确定的区域产生误差。结合建立起动力学微分方程。进一步将确定区域的动力学微分方程模型进一步优化。建立洋流型的动力学微分方程模型。最终求解得到了比较准确的飞机坠落水域区域范围。

问题 2，

在问题一已经求得了帮助“搜索者”规划合理准确的水域区域范围。为了能快速找到坠落的飞机。合理的调度“搜索者”在已确定的区域进行搜索，实现搜索时间最短，救援效率最高。考虑了不同类型的飞机携带不同电子扫描的设备。会有不同行的扫描范围和巡航时间。建立了时间最少的优化模型。进一步分析发现，坠落的区域呈现连续性的正态分布，为了缩小时间和有效搜索，进一步将时间最少的优化模型进一步讨论优化。最终求解出搜索时间。

2.Assumptions

- 1.假设失去联系的飞机没有被犯罪分子劫持
- 2.假设坠落到海洋飞机的重量没有改变
- 3.飞机落到水平面上，发生碰撞速度竖直方向速度几乎为零。保留水平方向上的速度。

由过程 a 得到飞机落入水中水平方向速度 $v_{x1} = v_1$ 。

3.Symbols

f	飞机受到阻力
m	飞机的重量
k_1	飞机坠落时阻力系数
ρ	空气密度
C	飞机在水中进行类平抛运动
P_0	为标准大气压
C_D	飞机收到的风阻力系数
S_b	为索飞机的总面积应该
G	重力

4.Models and Methodology

4.1Model One

模型一

模型分析:

飞机高速飞行的过程,发生故障时突然失去动力,于是飞机开始降落。飞机在降落的过程中,为了防止飞机因为失去速度而垂直掉落,驾驶员会在发动机出现故障以后,压低机头,主动放弃一部分飞机的高度来获取足够的速度,进行迫降。这时飞机的受力有重力G、风对飞机的阻力f、竖直向上的升力F。如图一所示:

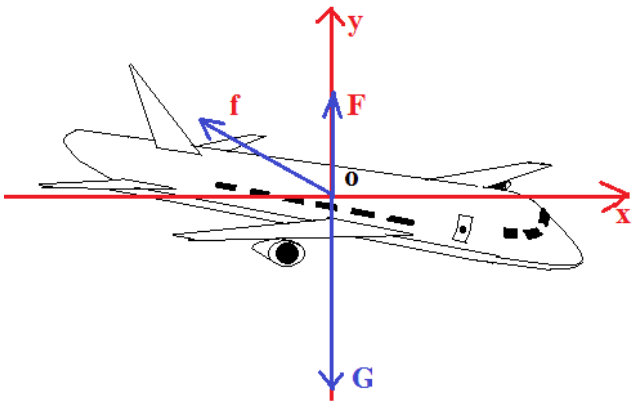


Figure 1:飞机下降时受力分析

一般在物体不大且运动速度比较低,可假定在较小的情况下,阻力与速度的一次方成正比即 $f = kv$;在一般的相当大的运动速度和物体尺寸的范围,可假定较大到接近

或超过音速时,阻力与速度的二次方成正比即 $f = kv^2$; 当物体速度超过音速时,阻力急剧增加,阻力与速度的高次方成正比^[1]即 $f = kv^n$ 。

所以飞机在空中受到的阻力:

$$f = \frac{1}{2} C_D \rho S v^2$$

其中飞机受到的空气阻力的系数为

$$k_1 = \frac{1}{2} C_D \rho S$$

同理可以得到飞机的升力的计算公式为:

$$F = \frac{1}{2} C_L \rho S v^2$$

飞机受到的升力的系数为

$$k_2 = \frac{1}{2} C_L \rho S$$

飞机重力的计算公式为:

$$G = mg$$

飞机发生故障到落水,直至静止。本文将其分为三个部分 a:飞机在空中进行类平抛运动。b. 飞机落到水平面上,竖直方向速度为零。只保留水平方向上的速度。C:飞机在水中进行类平抛运动。

飞机在空中运动过程,如图2所示:

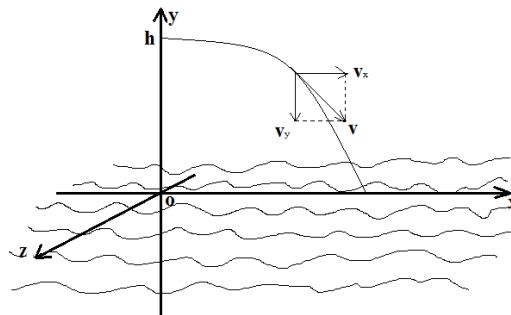


Figure 2: 在空中飞机运动过程以及坐标轴的建立

根据如上图1、图2所示,我们由牛顿第二定律可以得到飞机的运动过程动力学方程:

水平方向上的动力学的方程:

$$\vec{f}_x = m\vec{a}_x$$

竖直方向上动力学的方程:

$$\vec{G} + \vec{F} + \vec{f}_x = m\vec{a}_y$$

物体在水平方向运动学方程为 $\vec{a} = \frac{d^2\vec{r}}{dt^2}$, $v = \frac{dr}{dt}$

综上所述可以得到飞机从失事点坠落到水面的动力学微分方程模型:

$$\begin{cases} m \frac{d^2 x}{dt^2} = -k_1 \left(\frac{dx}{dt} \right)^2 \\ m \frac{d^2 y}{dt^2} = -mg + k_1 \left(\frac{dy}{dt} \right)^2 + k_2 \left(\frac{dy}{dt} \right)^2 \end{cases}$$

其中由初始条件:

$$\begin{cases} y(0) = H \\ x(0) = 0 \end{cases}$$

运用 Matlab 对微分方程模型进行符号数值求解得:

$$x = \frac{m}{k_1} \log\left(\frac{m + v_{x0} k_1 t}{m}\right)$$

$$y = H + \frac{m}{k_2 + k_1} \log\left[\tanh\left(\sqrt{\frac{mg}{k_2 + k_1}} \times t\right) + 1\right] - \sqrt{\frac{mg}{k_2 + k_1}} \times t$$

模型求解与结果分析:

对飞机坠落入水点进行求解, 以一般经典的飞机为例。机翼的参考面积 S 取为 $75m^2$ 。

飞机重量 $560t$, 飞行速度 $v_0 = 800km/h$ 约等于 $222m/s$ 。

对于一般的飞机而言, 低速和亚音飞机的升力和阻力大小比可达 $17 \sim 18^{[2]}$ 。本文取 $|F| = 17|f|$ 。风力阻力系数^[3]: 一些物体的风阻, 垂直平面体风阻系数大约 1.0、球体风阻系数大约 0.5、一般轿车风阻系数 0.28-0.4、好些的跑车在 0.25、赛车可以达到 0.15、飞禽在 0.1-0.2、飞机达到 0.08、目前雨滴的风阻系数最小在 0.05 左右。因此本文中飞机收到的风阻力系数 $C_D = 0.08$ 。

空气密度 ρ 与大气压、温度、湿度有关。根据密度的计算公式^[4]:

$$P = P_0(1 - 0.02257h)5.256 \text{ (kPa)}$$

$$\rho = P * 29 / (8314 * T)$$

其中 h 为海拔高度, P_0 为标准大气压, T 为温度单位是 $K = 273.15 + t$ 。计算得到一定海拔高度下的空气密度如下表 1 所示:

表 1: 空气密度和海拔高度

The air density and the altitude		
Serial number	Altitude(m)	Air density (kg/m)
1	1000	1.148
2	2000	1.0918
3	3000	0.9918
4	4000	0.8918
5	5000	0.7918
6	6000	0.6918

7	7000	0.5918
8	8000	0.4918
9	9000	0.3918
10	10000	0.2918

综合以上数据和阻力升力的对应关系和性质：

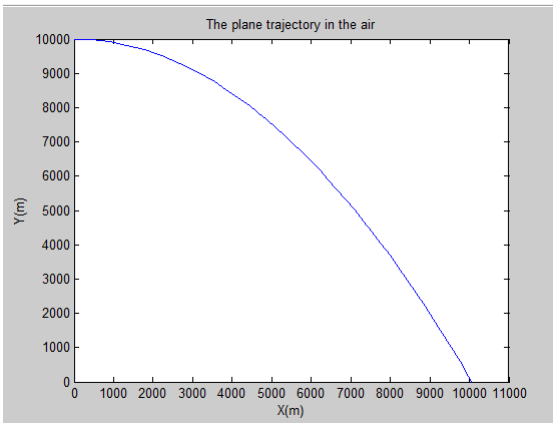
将数据代入求解得到系数 k_1, k_2 与海拔高度的对应关系，如表 2 所示：

表 2：海拔高度和 k_1, k_2 对应关系

Altitude with k1,k2 tables		
Altitude(m)	k1	k2
1000	3.44	58.55
2000	3.28	55.68
3000	2.98	50.58
4000	2.68	45.48
5000	2.38	40.38
6000	2.08	35.28
7000	1.78	30.18
8000	1.48	25.08
9000	1.18	19.98
10000	0.88	14.88

本文取飞机最高飞行高度 $10000m$ [1]得到：通过 Matlab 编程可以得到飞机在空中的运动轨迹如图 3 所示：

图 3：飞机在空中的运动轨迹



将上述的各个物理量带入竖直位移：

$$y = y_0 + \frac{m}{k_2 + k_1} \log \left[\tanh \left(\sqrt{\frac{mg}{k_2 + k_1}} \times t \right) + 1 \right] - \sqrt{\frac{mg}{k_2 + k_1}} \times t$$

求解得到飞机落到海平面的时间：

$$t = 46s$$

求得落水点的水平位移： $x_1 = 10204m$

记飞机水平方向上的速度 v_x 则由：

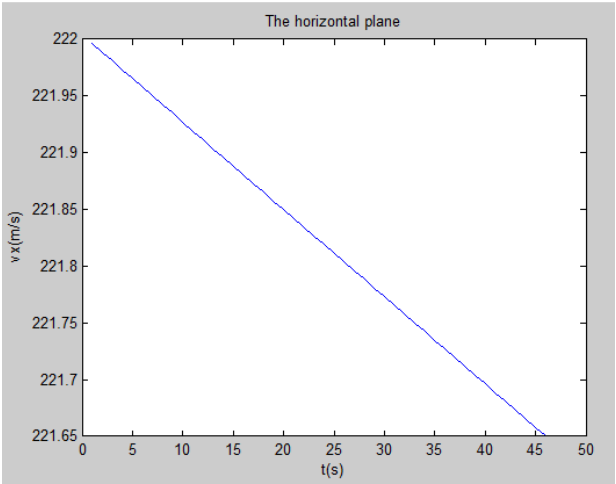
$$x = \frac{m}{k_1} \log\left(\frac{m + v_{x0} k_1 t}{m}\right)$$
$$v_x = \frac{dx}{dt}$$

得到飞机水平方向的速度表达式为：

$$v_x = \frac{mt}{k_1 t v_{x0}}$$

则可以得到飞机在空中运动水平方向上的速度如图 4：

图 4 ：飞机在空中的水平速度



由求解得到的飞机落入水的时间 $t = 46s$ 代入公式中可以求得落入水时的水平方向上的速度：

$$v_1 = 45.927$$

同理可以求得落入水时竖直方向的速度： $v_y = -419.2044m/s$

根据飞机不同的飞行高度，得到飞机落水点处不同的水平位移和水平速度如下表 3：

表 3：飞机不同飞行高度下的落水时的水平位移，和水平速度

Altitude with x1,v1 tables		
Altitude(m)	x1(m)	v1(m/s)
1000	10180	45.71
2000	10182	45.72
3000	10183	45.75
4000	10192	45.89
5000	10197	45.82
6000	10197	45.83
7000	10198	45.90
8000	10201	45.98

9000	10203	40.98
10000	10204	46.00

从上面的表 3 可以看出：水平位移在 $10180 \sim 10204m$ ，对应的水平速度 $45.71 \sim 46.00m/s$ 之间。

4.2 Model Two

模型分析：

飞机坠落到水中后会受到来自水的浮力和海水的黏附阻力的影响，飞机自身的重力影响由假设【3】可以得知道飞机在水一个减速平抛过程，首先飞机在水中不考虑洋流影响，则飞机在水中的受力过程如图 5：

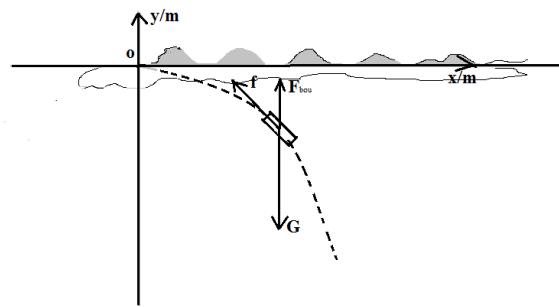


Figure 5: 水中无洋流作用下飞机受力分析以及坐标轴的建立

如上面图 5 所示,这时飞机受到重力 G 、水流对飞机的阻力、水中飞机受到的浮力。由于落水后飞机的速度较小，所以飞机在水中受到的阻力：

$$f = k_3 v$$

由浮力定律可以知道飞机在水中受到的浮力：

$$F_{bou} = \rho V g$$

由牛顿定律可得到飞机的重力：

$$G = mg$$

由动力学的牛顿第二定律基本知识得到：

飞机在水平方向上的受力情况：

$$\vec{f}_x = m \vec{a}_x$$

飞机在竖直方向上的受力情况：

$$\vec{G} + \vec{F}_{bou} + \vec{f}_x = m \vec{a}_y$$

联立上述方程我们可以得到由此得到飞机在水下运动过程的微分方程模型：

$$\begin{cases} m \frac{d^2 x}{dt^2} = -k_3 \frac{dx}{dt} \\ m \frac{d^2 y}{dt^2} = -mg + k_3 \frac{dy}{dt} + F_{bou} \end{cases}$$

由初始条件:

$$\begin{cases} \dot{x}(0) = v_{x1}; x(0) = x_1 \\ \dot{y}(0) = 0; y(0) = 0 \end{cases}$$

运用 Matlab 对微分方程模型进行符号数值求解得:

$$\begin{cases} x = \frac{v_{x1}m + k_3x_1}{k_3} - \frac{v_{x1}me^{\frac{-k_3}{m}t}}{k_3} \\ y = \frac{gm^2 - F_{bou}m + F_{bou}k_3t - gk_3mt}{k_3^2} - \frac{e^{\frac{-k_3}{m}t}(-F_{bou}m + gm^2)}{k_3^2} \end{cases}$$

模型求解:

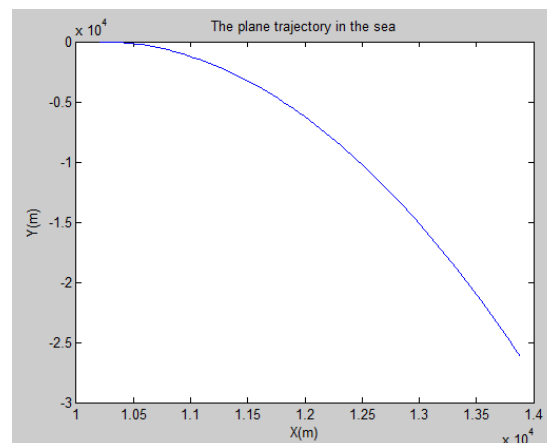
一般地取海水的阻力系数【5】 $k_3=0.12$ ，取飞机的体积 V 为 $900m^3$ ，海水的密度

$\rho=1.025 \times 10^3 kg/m^3$ ，代入公式可以求得飞机所受到的浮力。

$$F_{bou} = \rho V g = 9040500 kg$$

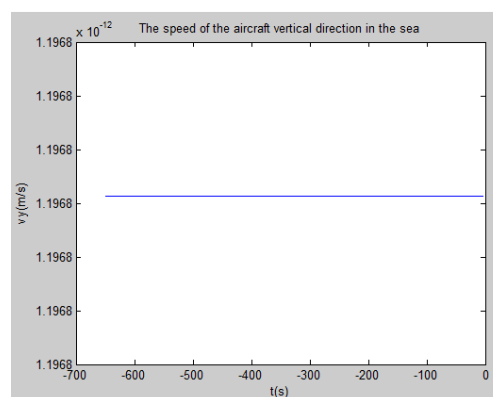
得到飞机在海中的运动轨迹如图 6

图 6 飞机在水中的运动轨迹



通过对坠落的飞机在海水中竖直方向的运动分析计算机仿真可以得到的飞机在竖直方向的速度变化如图所示:

图 7 飞机在海中竖直方向的速度



4.3 Model Three

模型分析:

对模型三进行优化，在海洋中飞机实际受到了不同情况的洋流作用。设入水点为原点，题中沿虚线入海的方向为 X 轴正方向，竖直指向天空的方向为 Y 轴的正方向，黑匣子入水时候的时间为零。设入水点为原点，题中沿虚线入海的方向为 X 轴正方向，竖直指向天空的方向为 Y 轴的正方向， Z 轴为 X 轴逆时针旋转直角角度。 X 轴与 Y 轴、 Z 轴构成三维立体空间如图 8。

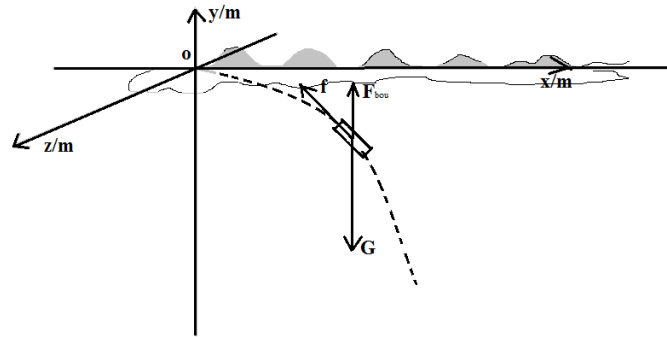


Figure 8: 水中洋流作用下飞机受力分析以及坐标轴的建立

由于海域的不同或季节的不同，洋流的方向也会有很多种。飞机在水中受到洋流或推动作用或阻碍作用。为了简化模型，我们只考虑水平面上的洋流。假设洋流在水平面上 xOz 的流速为 v 。其中在 x 轴上的分速度为 v_1 ，在 z 轴上的分速度为 v_2 。这样飞机在水中的初速度就可得出 x 轴上的速度 $v_{x1} + v_1$ ， y 轴上的速度为 0 ， z 轴上的初速度为 v_2 。具体受力情况如图 4 所示：

因此可得到：

飞机在洋流的作用下的模型：

$$\begin{cases} m \frac{d^2 x}{dt^2} = -k_3 \frac{dx}{dt} \\ m \frac{d^2 y}{dt^2} = -mg + k_3 \frac{dy}{dt} + F_{bou} \\ m \frac{d^2 z}{dt^2} = -k_3 \frac{dz}{dt} \end{cases}$$

由初始条件：

$$\begin{cases} x'(0) = v_{x1} + v_1; x(0) = x_1 \\ y'(0) = 0; y(0) = 0 \\ z'(0) = v_2; z(0) = 0 \end{cases}$$

运用 Matlab 对微分方程模型进行符号数值求解得：

$$\begin{cases} x = \frac{(v_{x1} + v_1)m + k_3 x_1}{k_3} - \frac{(v_{x1} + v_1)m e^{\frac{-k_3}{m}t}}{k_3} \\ y = \frac{gm^2 - F_{bou}m + F_{bou}k_3t - gk_3mt}{k_3^2} - \frac{e^{\frac{-k_3}{m}t}(-F_{bou}m + gm^2)}{k_3^2} \\ z = \frac{v_2}{k_3} - \frac{v_2 m e^{\frac{-k_3}{m}t}}{k_3} \end{cases}$$

模型求解：

这里为了方便求解我们取洋流的速度与水平面的夹角为 45 度。大小为 $2\sqrt{2}m/s$ 。则

$v_1 = 2m/s$ ， $v_2 = 2m/s$ 。得到飞机在海里的运动轨迹：如图 9 所示：

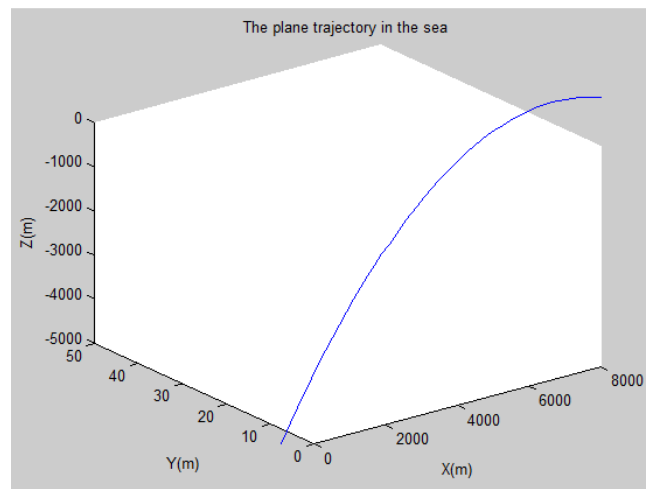


图 9：飞机在海底的运动轨迹

竖直平面飞机在有洋流和无洋流的对比下的运动轨迹如图 10 所示：

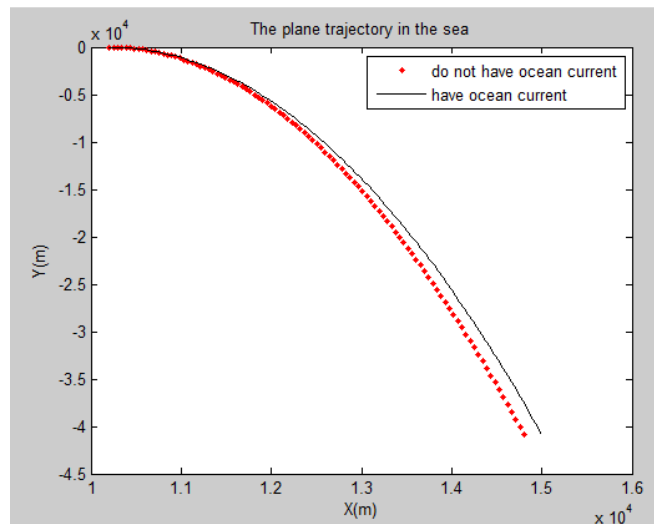


图 10：飞机在无洋流和有洋流下竖直方向运动轨迹对比图

根据以上所示，我们的得出结论：洋流的作用下，随着海底的深度不同。为此我们得出了当飞机飞行高度为两个极值时，飞机在洋流的推动和阻碍作用下。随着海底深度不同，得到飞机的沉降位置。如表 4 所示：

表 4：洋流的作用下随着海底深度的不同，飞机的沉降位置

10000(m)Ocean currents push			10000(m)Ocean currents hamper		
y(m)	z(m)	x(m)	y(m)	z(m)	x(m)
1000	31.4	10934	1000	31.4	10895
2000	44.3	11244	2000	44.2	11150
3000	54.2	11481	3000	54.2	11396
4000	62.6	11682	4000	62.6	11581
5000	70	11859	5000	67	11744

1000(m)Ocean currents push			1000(m)Ocean currents hamper		
y(m)	z(m)	x(m)	y(m)	z(m)	x(m)
1000	31.4	10929	1000	31.4	10866
2000	44.3	11237	2000	44.3	11148
3000	54.2	11473	3000	54.2	11365
4000	62.6	11673	4000	62.6	11548
5000	70	11849	5000	70	11709

通过表 4 数据我们得到了飞机沉降范围如图 11：

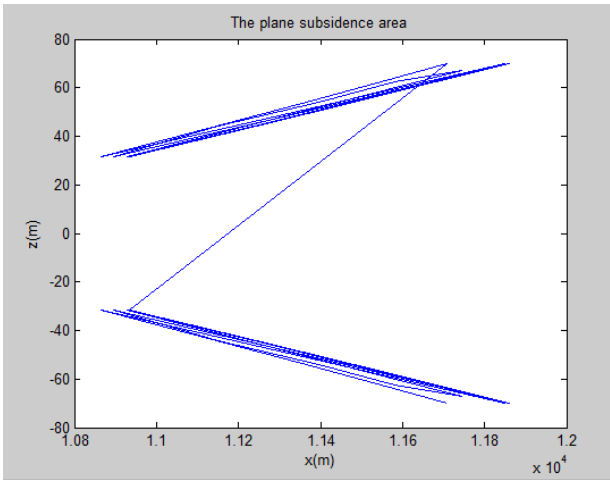


图 11：飞机沉降范围

通过飞机 10000m 降落在洋流的推动，阻碍作用下和飞机不在洋流作用下，与飞机在 1000m 降落在洋流的推动阻碍作用下和飞机不在洋流作用下，得到了飞机沉降位置概率面积大的部分 S_s 如图 12。

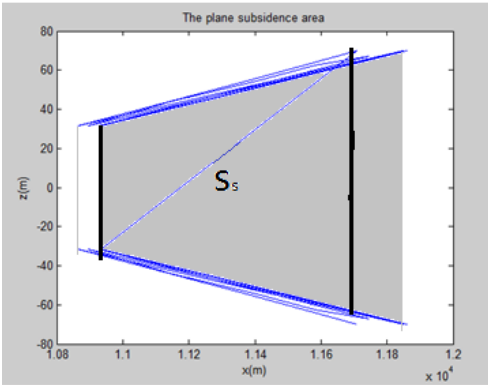


图 12：飞机降落概率分布

从图 12 可以看出，从海平面直至海底深度 5000m。飞机沉降的最大范围为大梯形，飞机沉降概率大的区域为小梯形。为此搜索飞机的总面积应该为 S_b ，其中应该重点搜索小梯形，搜索面积为 S_s 。则搜索面积最大面积为：

$$h = 11859 - 10866 = 993m$$

$$a = 31.4 \times 2 = 62.8m$$

$$b = 70 \times 2 = 140m$$

$$S_b = \frac{1}{2}(a+b)h = 109627.2m^2$$

重点搜索面积大小为：

$$h = 11849 - 10895 = 954m$$

$$a = 44.3 \times 2 = 88.6m$$

$$b = 67 \times 2 = 134m$$

$$S_s = \frac{1}{2}(a+b)h = 106180.2m^2$$

$$\frac{S_s}{S_b} = 0.9686$$

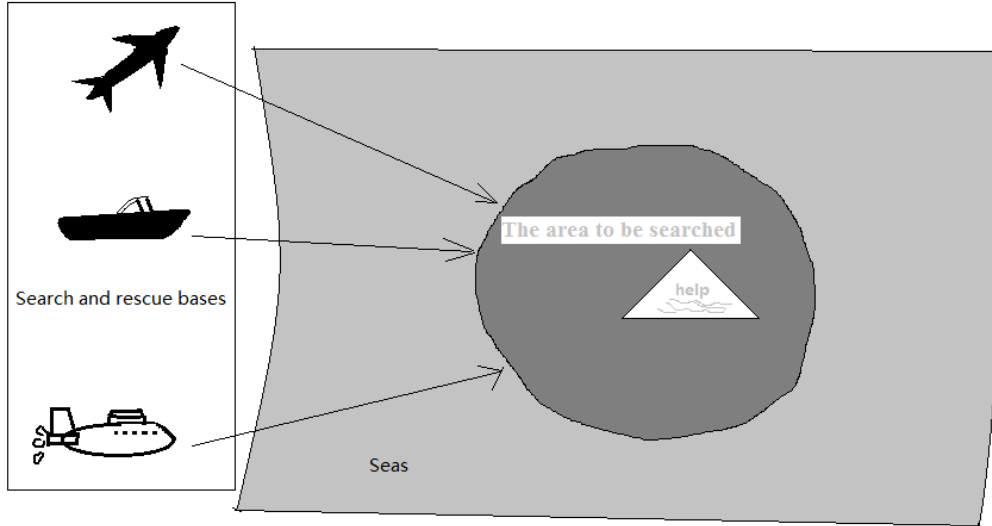
结论分析：通过对上述三个模型进行仿真模拟，得到了已知飞行高度的飞机，在发生故障时沉降海底的范围。从而也就确定了搜救范围。同时从数据也可以看出由于洋流的作用飞机在不断的移动。所以为了能够准确的找到失联的飞机就必须把握时间，时间越久飞机的沉降范围就越大。因此最优的搜救方案就是搜索最快。

4.4 Model Four

模型四：

模型的提出：

在对模型一中得出的可行区域 S 进行搜寻的时候，首先我们得及时、准确地获取可行区域 S 附近的搜寻力量信息。然后从所得到的搜寻力量中选出可用的力量并给出一个最合理的方案，从而能够在最短的时间内完成搜寻任务。要解决这个问题需要完成两个方面的工作：先得对搜寻力量进行删选，选出符合条件的搜寻力量，此处需要综合考虑可行区域 S 的环境状况，以及遇险人员的实际情况结合搜寻力量自身的情况进行安排。再从满足条件的搜寻力量中选择最佳搜寻力量参与搜寻，只是本模型的重点研究对象，即通过建立数学模型实现对可用搜寻力量的优选。



如图所示：假设该可行域 S 内没有任何可以参与搜寻任务的条件（如飞机、船舶、潜艇），即所有参与搜寻任务的力量都是从可行域 S 附近的不同救援基地派出。对于救援基地派遣出来的搜寻力量：飞机、船舶、潜艇共同构成了海空立体搜寻力量。其中专业搜救飞机、船舶、潜艇的最大航速、搜寻能力、最大续航时间和救援基地到可行域 S 的初始距离也可能不同。如何选择可用的船舶和飞机并使其协同工作。以最短的时间对可行域 S 实施高效率的搜寻覆盖，是海上搜救实践中常常遇到的优化问题。该问题是一个追求时间最快的优化问题。所以可以建立如下的数学模型再进行求解。

首先，对所需符号进行说明：待搜海域即可行域的面积 $S(n\text{ mile}^2)$ ；被派往参与搜寻的飞机、船舶、潜艇的数量分别为 O, P, Q ，且每架参与搜寻任务的飞机距离可行域的初始距离为 $D_i(n\text{ mile}^2), i = 1, 2, \dots, O$ ；每艘船舶距离可行域的初始距离为 $D_j(n\text{ mile}^2), j = 1, 2, \dots, P$ ；每艘潜艇距离可行域的初始距离为 $D_k(n\text{ mile}^2), k = 1, 2, \dots, Q$ ；每架飞机、每艘船舶、每艘潜艇的最大航速分别为： $V_i(kn/h), i = 1, 2, \dots, O$ ； $V_j(kn/h), j = 1, 2, \dots, P$ ； $V_k(kn/h), k = 1, 2, \dots, Q$ ；每架飞机、每艘船舶、每艘潜艇的搜寻能力分别为： $A_i(n\text{ mile}^2/h), i = 1, 2, \dots, O$ ； $A_j(n\text{ mile}^2/h), j = 1, 2, \dots, P$ ； $A_k(n\text{ mile}^2/h), k = 1, 2, \dots, Q$ ；每架飞机的最大续航时间为 $T_i, i = 1, 2, \dots, O$ 。每架飞机在整个搜寻过程中的出动架次为 $L_i, i = 1, 2, \dots, O$ ；可行域能够容纳的搜寻飞机数、船舶数、潜艇数的上限分别为 $O_1(1 \leq O_1 \leq O)$ ； $P_1(1 \leq P_1 \leq P)$ ； $Q_1(1 \leq Q_1 \leq Q)$ 。整个搜寻过程消耗的时间为 $T(h)$ 。

根据上述符号说明，可求得第 i 架飞机全速往返可行域所需的时间为 $T_{wi} = \frac{2D_i}{V_i}(h)$ ，所以第 i 架

飞机在可行域开展搜寻作业的时间为： $T_{si} = T_i - T_{wi}(h)$ ，第 i 架飞机共出动的架次 $L_i = \frac{T}{T_i}$ ，第 j

艘船舶和第 k 艘潜艇能够在可行域 S 内展开搜寻的时间为 $T_{sj} = T - \frac{D_j}{V_j}(h)$ ， $T_{sk} = T - \frac{D_k}{V_k}(h)$ 。

模型的建立：

该模型的目标是要求选择最佳的搜救力量，并使其协同工作，从而使得完成对整个可行域 S 的覆盖搜寻所消耗的时间 $T(h)$ 最少，所以可引入如下决策变量：

$$x_i = \begin{cases} 0, & \text{If the aircraft does't participate in the search,} \\ 1, & \text{If the aircraft participating in search.} \end{cases}$$

$$y_j = \begin{cases} 0, & \text{If the ship does't participate in the search,} \\ 1, & \text{If the ship participating in search.} \end{cases}$$

$$z_k = \begin{cases} 0, & \text{If the submarine does't participate in the search,} \\ 1, & \text{If the submarine participating in search.} \end{cases}$$

为了以最短的时间对可行域实施快速高效的搜寻覆盖，需要分析空中力量、水面力量、水下力量在整个搜寻行动中所用时间的构成情况。

首先分析空中力量参与行动的时间：并非所有搜救飞机都有机会参与行动，只有那些在搜寻行动结束前能够赶到可行域的飞机可以参与搜寻任务，对于每一架有机会参与行动的飞机 a_i 而言，由于它的续航时间有限，需要执行多个架次的搜寻，且每个架次所用时间均等于其最大续航时间 T_i 。本文中为了方便处理，假设飞机往返可行区域与救援基地的时间相等，并且不考虑回基地加油的时间，所以飞机总的搜寻时间为

$$T_f = L_i T_{si}(h) = L_i \left(T_i - \frac{2D_i}{V_i} \right).$$

再来分析水面力量参与行动的时间：考虑到每艘船舶的最大航速、距可行域的初始距离各不相同，所以其到达可行域的时刻就不一样。对于每一艘有机会参与搜寻的船舶 b_j 而言，其

搜寻所用时间为 $T_{sj} = T/j - \frac{D_j}{V_j}(h)$ ，所以总的搜寻时间为 $T_c = \left(T - \frac{2jD_j}{V_j} \right)$ ；同理可以求出潜

艇总的搜寻时间为 $T_q = \left(T - \frac{2kD_k}{V_k} \right)$ 。

通过以上的分析可知，要实现对可行域的完全覆盖需要满足：

$$\sum_{i=1}^O T_f \cdot A_i \cdot x_i + \sum_{j=1}^P T_c \cdot A_j \cdot y_j + \sum_{k=1}^Q T_q \cdot A_k \cdot z_k = S$$

即

$$\sum_{i=1}^O \left(T - \frac{2iD_i}{V_i} \right) \cdot A_i \cdot x_i + \sum_{j=1}^P \left(T - \frac{2jD_j}{V_j} \right) \cdot A_j \cdot y_j + \sum_{k=1}^Q \left(T - \frac{2kD_k}{V_k} \right) \cdot A_k \cdot z_k = S$$

由此可解出 T ，

$$T = \frac{\sum_{i=1}^O \frac{2iD_i}{V_i} \cdot A_i \cdot x_i + \sum_{j=1}^P \frac{2jD_j}{V_j} \cdot A_j \cdot y_j + \sum_{k=1}^Q \frac{2kD_k}{V_k} \cdot A_k \cdot z_k + S}{\sum_{i=1}^O A_i \cdot x_i + \sum_{j=1}^P A_j \cdot y_j + \sum_{k=1}^Q A_k \cdot z_k}$$

所以模型可以表示成：

$$\left\{ \begin{array}{l} global \ min T = \frac{\sum_{i=1}^O \frac{2iD_i}{V_i} \cdot A_i \cdot x_i + \sum_{j=1}^P \frac{2jD_j}{V_j} \cdot A_j \cdot y_j + \sum_{k=1}^Q \frac{2kD_k}{V_k} \cdot A_k \cdot z_k + S}{\sum_{i=1}^O A_i \cdot x_i + \sum_{j=1}^P A_j \cdot y_j + \sum_{k=1}^Q A_k \cdot z_k} \quad (1) \\ s. \ t. \quad 1 \leq \sum_{i=1}^O x_i = O_1 \leq O \quad (2) \\ \quad \quad \quad 1 \leq \sum_{j=1}^P y_j = P_1 \leq P \quad (3) \\ \quad \quad \quad 1 \leq \sum_{k=1}^Q z_k = Q_1 \leq Q \quad (4) \end{array} \right.$$

上述模型就是所要求的关于时间的优化模型，记为 M1，(1)是求整个搜寻行动所用时间 T 的全局最小值，并带有三个约束条件，即参与搜寻任务飞机总量的约束条件(2)，船舶总量的约束条件(3)，以及参与搜寻潜艇总量的约束条件(4).增加搜救力量总量约束主要考虑两个方面：一是客观上，由于受到可行域面积的大小以及参与搜救飞机的类型、船舶的吨位、潜艇的操纵性能限制，不便于全部力量都参与行动。而是从主观上，通过加入力量总量的限制，以便于求得不同数量搜救力量参与行动所需的最少时间，从而使搜救指挥者在搜寻代价和所需时间上综合考虑，从而制定出耗时较少且动用搜寻力量较少的最优行动方案。

模型求解：

该模型可以利用 Dinkelbach 算法求解，该算法的思想是：首先构造一个与该模型具有相同最优解的含参数的辅助问题，然后通过迭代搜索方法求解，具体如下：设

$$f_1(x) = \sum_{i=1}^O \frac{2iD_i}{V_i} \cdot A_i \cdot x_i + \sum_{j=1}^P \frac{2jD_j}{V_j} \cdot A_j \cdot y_j + \sum_{k=1}^Q \frac{2kD_k}{V_k} \cdot A_k \cdot z_k + S$$

$$f_2(x) = \sum_{i=1}^O A_i \cdot x_i + \sum_{j=1}^P A_j \cdot y_j + \sum_{k=1}^Q A_k \cdot z_k$$

则原模型可表示成 M2 形式：

$$\begin{cases} f(x) = \frac{f_1(x)}{f_2(x)} \\ x \in S \end{cases}$$

其中： $S = \{x | x \in (0,1), \sum_{i=1}^O x_i = 0, 1 \leq O_i \leq O\}$ 为可行域，并且有 $f_1(x) > 0, f_2(x) > 0$ 。再构造与 M2 有相同最优解的含参数的辅助问题 M3 为：

$$\begin{cases} g(\tau) = \max g(\tau) = f_1(x) - \tau f_2(x) \\ x \in S \end{cases}$$

所以其详细步骤如下：

第一步：任取 M2 中的一个可行解 $x_0 \in S$ ，令 $\tau_0 = f(x_0)$ ；转入第二步。

第二步：解辅助问题你 M3，得其最优解 x^{τ^k} 和最优值 $g(x_0)$ ，若 $g(x_0) = 0$ ，则 x^{τ^k} 是最优值，终止，否则转第三步。

第三步：令 $\tau^{k+1} = f(x^{\tau^k})$ ， $S = S + 1$ ，返回第二步。

模型仿真：

由模型三可得：待搜寻海域即可行域 $S = 109627.2m^2$ ，设有 2 艘可用船舶，5 架飞机，和 3 艘潜艇。其对应能力信息如表 5 所示：

Figuer5 : Search power information

Search power information		
Ship number	Start distance(m)	Largest search area (m^2/h)

1	1000	80
2	21000	100
Plane number	Start distance(m)	Largest search area (m^2/h)
1	21000	155
2	35000	175
3	2250000	135
4	41200000	155
5	71700000	175
Dunker number	Start distance(m)	Largest search area (m^2/h)
1	0	1
2	100	2
3	200	3

将数据带入模型求解，有如下方案可行：

Plan 1		
Ship number	Search time(h)	search area (m^2/h)
1	172.22	34444
2	122.22	36666
3	72.22	28888
Plan 2		
Plane number	Search time(h)	search area (m^2/h)
1	55.55	100000
2	0	0
3	0	0
4	0	0
5	0	0
Plan 3		
Dunker number	Search time(h)	search area (m^2/h)
1	164.2	100000
2	0	0
Plan 3		
Tool	Search time(h)	search area (m^2/h)
Ship number 1	50.4	10080
Dunker number 1	101.5	20300
Dunker number 2	49.6	14880
Others	0	0

Figure 6 : Search and rescue plan

通过营救方案结果可以看出方案二用时间最短，则可以看出采用搜索最快的飞机作为搜索工具更快。综合结果可以看出为了得到最佳的搜索将结果，除了搜索工具重要。速度也是很重要的根据模型三，可以看出随着时间的不断延长飞机不断往下沉降，当海底深

度很大时，搜索工具也就无法搜索的到。所以确定准确的搜索位置，和搜索工具，搜索方案的合理利用，是十分重要的。

5.Strength and Weakness

5 优缺点分析

5.1 Strength 优点

Our model can calculate the possible “splash-down” area using only the altitude, speed, and last known coordinates of the plane, which is much less data than most searches typically require. Using those three parameters, the model is able to generate a more realistic search shape than just a circle or rectangle. It can also use real-life ocean current data to have the debris search area drift accordingly, allowing it to change shape and size. Other models might just have a static search area, and debris might drift out of that. A third strength of the model is that, regardless of the shape of the area, we are able to model and predict how long it will take to search that area sufficiently to have a 95% chance of finding a piece of debris, if there is any debris to be found.

我们的模型可以计算出可能的“飞溅”面积仅使用的高度，速度，和最后已知的平面坐标，这是少得多的数据比大多数搜索通常需要。使用这三个参数，该模型能够产生一个更逼真的搜索形状比只是一个圆形或矩形。它也可以使用现实生活中的洋流数据，使碎片搜索区域相应漂移，允许它改变形状和大小。其他模型可能只是有一个静态搜索区域，碎片可能会漂移出来。三分之一强度的模型，无论该区域的形状，我们能够模拟和预测多长时间搜索区域有 95% 的机会找到一块碎片，如果有被发现的任何碎片。

5.2 Weakness 缺点

While our model is quite robust, it has a few important weaknesses. One of them is that, in order to run our search patterns, the model has to search a rectangle that bounds the search area, rather than only the area itself. This means that, depending on the orientation and shape of the actual search area, the search planes may spend excessive amounts of time searching outside where the debris could be. Another weakness at the moment is that the model can only calculate current drift in the vicinity of where flight MH370 went down; we lack the data necessary to model ocean currents globally. Our model does not account for survivors floating on debris or in rafts moving the debris against the current and outside the search area as they head for land. Also, our model does not take into account the speed of the crashing plane when it loses power and begins to glide, so the actual search area might be larger than what we are currently estimating.

虽然我们的模型是相当强大的，它有一些重要的弱点。其中之一是，为了运行我们的搜

索模式,该模型必须搜索一个矩形的边界搜索区域,而不仅仅是该地区本身。这意味着,根据实际搜索区域的方向和形状,搜索飞机可能会花费大量的时间搜索外面的碎片可能。另一个弱点此刻是模型只能计算附近哪里 fl 飞行 MH370 下降电流漂移; 我们缺乏必要的全局数据模型的洋流。我们的模型没有考虑幸存者 fl 浮在废墟或筏移动碎片对当前搜索区域之外的人的土地。此外,我们的模型不考虑的崩溃飞机的速度时,它失去了权力,并开始滑行,所以实际的搜索区域可能比我们目前估计的大。

6.Reference

- [1] 王立华, 空气阻力对自由落体运动的影响, 沧州师范专科学校学报, 第 16 卷第 2 期: 2000, 51。
- [2] 马 鹏, 丰家辉. 风力发电机翼型升阻比的计算[J]. 华 电技术, 2008, (7): 76.
- [3]<http://baike.baidu.com/view/95279.htm>
- [4]<http://wenku.baidu.com/view/9bc9350876c66137ee061944.html>
- [5] 王宪杰, 海水中物体下落速度的新算法, 高等数学典型应用实例与模型[M], 北京: 科学出版社, 2004。
http://blog.sina.com.cn/s/blog_6383bcba0100xf4z.html
- [6] 邢胜伟 海上立体搜索全局优化模型及仿真研究[D]
- [7]http://wenku.baidu.com/link?url=YOyixrJXWj0ZunlJGqdUFdLv8wkF1KCXxXcEkHGpaulHwlsXPwjR29GxGBxQ-AhMVIKIMDm3045K5kP6xvrabtnGbPDzk1oI5rS-BWhYbe_

Appendix

附录 1:

```

x=dsolve('D2y*m+k1*Dy*Dy=0','y(0)=0,Dy(0)=vx0','t');
x =
(m*log((k1*(t*vx0 + m/k1))/m))/k1
y=dsolve('m*D2y=-m*g+(k1+k2)*Dy*Dy','y(0)=h,Dy(0)=0','t');
y =
(m*(log(2) - log(g^(1/2)*m^(1/2)*exp(-h/m*(k1 + k2))/(k1 + k2)^(1/2) +
g^(1/2)*m^(1/2)*exp(-h/m*(k1 + k2))*exp(2*g^(1/2)/m^(1/2)*t*(k1 + k2)^(1/2))/(k1 + k2)^(1/2))
- log(k1 + k2)/2 + log(g)/2 + log(m)/2 + (g^(1/2)*t*(k1 + k2)^(1/2)/m^(1/2)))/(k1 + k2)
t=0:46;
m=5600000;
k1=0.8754;
k2=14.8818;
vx0=222;
h=10000;
g=9.8;
x =(m*log((k1*(t*vx0 + m/k1))/m))/k1;
y =(m*(log(2) - log(g^(1/2)*m^(1/2)*exp(-h/m*(k1 + k2))/(k1 + k2)^(1/2) +
g^(1/2)*m^(1/2)*exp(-h/m*(k1 + k2))*exp(2*g^(1/2)/m^(1/2)*t*(k1 + k2)^(1/2))/(k1 + k2)^(1/2))
- log(k1 + k2)/2 + log(g)/2 + log(m)/2 + (g^(1/2)*t*(k1 + k2)^(1/2)/m^(1/2)))/(k1 + k2);
plot(x,y);
axis([0,12000,0,10000]);
xlabel('X(m)');
ylabel('Y(m)');
title('The plane trajectory in the air');
x =(m*log((k1*(t*vx0 + m/k1))/m))/k1;
diff(x);
vx=(m*t)/(k1*(t*vx0 + m/k1))
t=0:46;
m=5600000;
k1=0.8754;
k2=14.8818;
vx0=222; h=10000;
g=9.8; x =(m*log((k1*(t*vx0 + m/k1))/m))/k1;
vx=diff(x);
plot(vx);
xlabel('t(s)');
ylabel('vx(m/s)');
title('The horizontal plane');
t=46;
m=5600000;
k1=0.8754;

```

```

k2=14.8818;
vx0=222;
h=10000;
g=9.8;
x=(m*log((k1*(t*vx0+m/k1))/m))/k1
x =
    1.0204e+004
t=46;%
m=5600000;
k1=0.8754;
k2=14.8818;
vx0=222;
h=10000;
g=9.8;
vx=(m*t)/(k1*(t*vx0+m/k1))
vx =
    45.9267

```

附录二：

```

x=dsolve('D2y*m+k3*Dy*Dy=0','y(0)=x1,Dy(0)=vx1','t');
x=(m*log((k3*((m*exp((k3*x1)/m))/k3+t*exp((k3*x1)/m)*(v1+vx1)))/m))/k3;
y=dsolve('m*D2y=-m*g+k3*Dy*Dy+F','y(0)=0,Dy(0)=0','t');
y=-(m*(log(k3^(1/2))+log(-i/2)+log(((F-g*m)^(1/2)*i)/k3^(1/2)+(exp((2*k3^(1/2)*t*(F-g*m)^(1/2)*i)/m)*(F-g*m)^(1/2)*i)/k3^(1/2))+log(1/(F-g*m)^(1/2)))-k3^(1/2)*t*(F-g*m)^(1/2)*i)/k3;
z=dsolve('D2y*m+k3*Dy*Dy=0','y(0)=0,Dy(0)=v2','t');
z=(m*log((k3*(t*v2+m/k3))/m))/k3;
t=39;
m=5600000;
k3=0.12;
vx1=46;
v1=2;
v2=2;
x1=10204;
F=9040500;
g=9.8; x=(m*log((k3*((m*exp((k3*x1)/m))/k3+t*exp((k3*x1)/m)*(v1+vx1)))/m))/k3;
y=-(m*(log(k3^(1/2))+log(-i/2)+log(((F-g*m)^(1/2)*i)/k3^(1/2)+(exp((2*k3^(1/2)*t*(F-g*m)^(1/2)*i)/m)*(F-g*m)^(1/2)*i)/k3^(1/2))+log(1/(F-g*m)^(1/2)))-k3^(1/2)*t*(F-g*m)^(1/2)*i)/k3;
z=(m*log((k3*(t*v2+m/k3))/m))/k3;

```

附录三：

```

t=0:100;
    m=5600000;
k3=0.12;
vx1=46;

```

```

v1=2;
v2=2;
x1=10204;
F= 9040500;
g=9.8;
x0=(m*log((k3*((m*exp((k3*x1)/m))/k3 + t*vx1*exp((k3*x1)/m)))/m))/k3;
x=(m*log((k3*((m*exp((k3*x1)/m))/k3 + t*exp((k3*x1)/m)*(v1 + vx1)))/m))/k3;
y=-(m*(log(k3^(1/2)) + log(-i/2) + log(1/(F - g*m)^(1/2)) + log(-((F - g*m)^(1/2) +
k3^(1/2)*vy*i)/k3^(1/2) + (2*(F - g*m)^(1/2)*i)/k3^(1/2) + (exp((2*k3^(1/2)*t*(F -
g*m)^(1/2)*i)/m)*((F - g*m)^(1/2) + k3^(1/2)*vy*i)/k3^(1/2))) - k3^(1/2)*t*(F -
g*m)^(1/2)*i)/k3;
z=(m*log((k3*(t*v2 + m/k3))/m))/k3;
plot3(x,z,y);
axis([0,8000,0,50,-5000,0]);
xlabel('X(m)');
ylabel('Y(m)');
zlabel('Z(m)');
title('The plane trajectory in the sea');
t=22.20;
m=5600000;
k3=0.12;
vx1=45.71;
v1=-2;
v2=2;
x1=10180;
F= 9040500;
g=9.8;
x=(m*log((k3*((m*exp((k3*x1)/m))/k3 + t*exp((k3*x1)/m)*(v1 + vx1)))/m))/k3
y=-(m*(log(k3^(1/2)) + log(-i/2) + log(((F - g*m)^(1/2)*i)/k3^(1/2) + (exp((2*k3^(1/2)*t*(F -
g*m)^(1/2)*i)/m)*(F - g*m)^(1/2)*i)/k3^(1/2)) + log(1/(F - g*m)^(1/2))) - k3^(1/2)*t*(F -
g*m)^(1/2)*i)/k3
z=(m*log((k3*(t*v2 + m/k3))/m))/k3
a=[31.4  44.3 54.2 62.6 70.0 31.4 44.2 54.2 62.6 67.0 31.4 44.3 54.2 62.6 70.0 31.4 44.3 54.2
    62.6 70.0 -31.4   -44.3   -54.2   -62.6   -70.0   -31.4   -44.2   -54.2
    -62.6   -67.0   -31.4   -44.3   -54.2   -62.6   -70.0   -31.4   -44.3
    -54.2   -62.6   -70.0];
b=[10934 11244  11481  11682  11859  10895  11150  11396  11581  11744
    10929  11237  11473  11673  11849  10866  11148  11365  11548
    11709 10934  11244  11481  11682  11859  10895  11150  11396  11581
    11744  10929  11237  11473  11673  11849  10866  11148  11365
    11548  11709];

plot(b,a)
xlabel('x(m)');

```



```
ylabel('z(m)');
title('The plane subsidence area');

#include <cstdio>
#include <cmath>
#include <algorithm>
#include <iostream>
using namespace std;

#define N 1100
#define EPS 1e-6
#define EQ(a, b) (fabs(a-b)<=EPS)
#define FFOR(i, n) for(i=0; i<n; ++i)

struct info{ double v; int i; }c[N];

double a[N], b[N];
int n, k;

bool compare(info a, info b){
return a.v>b.v;
}

double ques(double x){
int i;
FFOR(i, n) { c[i].v=a[i]-x*b[i]; c[i].i=i; };
sort(c, c+n, compare);

double fz, fm;
fz=fm=0;
FFOR(i, n-k) { fz+=a[c[i].i]; fm+=b[c[i].i];}
return fz/fm;
}

int solve(){
double k, q;

k=0.0, q=rand();

while(!EQ(k, q)){
k=q;
q=ques(k);
}
return (int)(k*100+0.5);
```

```
}
```

```
int main(){  
    int i;  
    while(scanf("%d%d", &n, &k) && n|k){  
        FFOR(i, n) scanf("%lf", &a[i]);  
        FFOR(i, n) scanf("%lf", &b[i]);  
        printf("%d\n", solve());  
    }  
    return 0;  
}
```