



## “华为杯”第十五届中国研究生 数学建模竞赛

学 校

上海交通大学

参赛队号

18102480211

队员姓名

1.闫晓桀

2.王成罡

3.吴迪

# “华为杯”第十五届中国研究生 数学建模竞赛

## 题 目      基于卫星高度计海面高度异常资料 获取潮汐调和常数方法及应用

---

### 摘            要：

本文以**数据插值**理论为基础，研究了高度异常值提取潮汐调和常数、潮汐调和常数沿轨道方向变化规律、正压潮和内潮分离、数据插值绘制同潮图以及插值函数最优阶次确定等问题。首先，通过“线性映射”的数学思想，将资料中所给数据格式进行统一化处理，将无序数据格式转化为有序的矩阵元素，为后续数据处理提供便利。利用不同的算法如**最近点跟踪法**、**高阶多项式轨迹插值法**、**有效网格点邻近搜索高维插值法**等求解潮汐调和常数沿轨道方向变化规律，数据插值法绘制同潮图等问题。而本文的创新点在于对原始数据增加**数据清洗**环节，滤掉可信度不高的数据点，排除偶然因素带来的扰动，从而保证模型及结果的准确性。

问题一中，解决了高度异常值提取潮汐调和常数的问题。利用调和分析法求解观测点潮汐调和常数时，充分考虑闰年、大小月份、格林尼治时间与北京时间关系等因素，撰写专用程序处理时间转换问题，采用**最小二乘法**进行求解，确保结果准确性；采用**最近点匹配比较法**验证结果时，确定验潮站最近观测点时，充分考虑地球自身球体形状，计算两点间**球面距离**；验证评价采用**振幅偏差 $\Delta H$** ，**振幅偏差均方差 $RMS_{\Delta H}$** ，**矢量差 $\Delta$** ，**矢量差均方差 $RMS_{\Delta}$** 四个角度综合进行评价，更全面更客观。

问题二中，解决了潮汐调和常数沿轨道方向变化规律的问题，同时对正压潮和内潮进行了分离。选取一条轨道上的观测点为例进行分析，等间隔选取若干观测点作为控制点进行后续**高阶多项式轨迹插值**，得到插值函数。并根据映射关系，将潮汐调和常数沿轨道方向变化的三维问题转化为随纬度方向变化的二维问题，方便进行正压潮和内潮的分离。

问题三中，解决了同潮图的绘制问题。首先通过“化整为零”的数学思想，将海洋进行网格化划分，采用**最近点跟踪算法**，进行**数据清洗**去除无效点，同时

为后续有效网格点邻近搜索高维插值法进行面插值提供数据，最终绘制南海区域同潮图。

问题四中，解决了插值函数最优阶次确定的问题。通过引入均方差参数作为衡量插值函数效果优劣的性能指标，最终得到 15 次为最佳阶次。

## 目 录

1.问题重述.....	5
1.1 问题背景.....	5
1.2 要解决的问题.....	5
2.模型基本假设.....	6
3.符号说明.....	7
4.问题一 星下观测点各主要分潮潮汐调和常数求解及评价检验.....	8
4.1 问题分析.....	8
4.2 星下观测点主要分潮潮汐调和常数的求解.....	8
4.2.1 数据处理 .....	8
4.2.2 模型准备 .....	10
4.2.3 模型建立与求解 .....	13
4.2.4 求解结果 .....	14
4.3 利用验潮点调和常数对计算结果进行检验评价.....	15
4.3.1 数据处理 .....	15
4.3.2 模型准备 .....	17
4.3.3 模型建立与求解 .....	19
4.3.4 求解结果 .....	19
5.问题二 观测点调和常数沿轨道作图及正压潮、内潮分离.....	24
5.1 问题分析.....	24
5.2 星下观测点各主要分潮潮汐调和常数沿轨道方向分布.....	24
5.2.1 数据处理 .....	24
5.2.2 模型准备 .....	27
5.2.3 模型建立与求解 .....	27
5.2.4 求解结果 .....	27
5.3 沿轨道各分潮的潮汐调和常数的正压潮和内潮的分离.....	32
5.3.1 数据处理 .....	32
5.3.2 模型准备 .....	33

5.3.3 模型建立与求解 .....	34
5.3.4 求解结果 .....	37
6.问题三 插值求出各分潮的同潮图，并利用验潮站检验 .....	41
6.1 问题分析.....	41
6.2 采用高阶多项式轨迹插值法绘制南海主要分潮的同潮图.....	41
6.2.1 数据处理 .....	41
6.2.2 模型准备 .....	44
6.2.3 模型建立与求解 .....	45
6.2.4 求解结果 .....	47
6.3 利用验潮站数据对插值法得到的同潮图进行评价.....	50
6.3.1 数据处理 .....	50
6.3.2 模型准备 .....	51
6.3.3 模型建立与求解 .....	51
6.3.4 求解结果 .....	51
7.问题四 插值求出各分潮的同潮图，并利用验潮站检验 .....	55
7.1 问题分析.....	55
7.2 数据处理.....	55
7.3 模型准备.....	55
7.4 模型建立与求解.....	56
7.5 求解结果.....	56
8.模型评价 .....	60
8.1 模型的优点.....	60
8.2 模型的缺点.....	60
9.参考文献.....	61
10.附录.....	62

# 1. 问题重述

## 1.1 问题背景

海洋潮汐潮流运动作为海洋中的基本运动之一，直接影响着波浪、风暴潮、环流等其他海洋现象，其在海岸及河口区域附近尤为明显<sup>[1]</sup>。潮汐潮流领域的相关研究，对加强沿海国防建设、促进经济发展及开发沿海海洋资源具有重要意义。

区域海洋潮汐的数值模拟需要在开边界的水位调和常数基础之上进行，目前获取开边界水位调和常数的途径主要分为：①实际观测；②全球海洋潮汐数值模拟。但这两种途径目前都由于各种限制导致效果差强人意<sup>[2]</sup>。

目前，区域海洋潮汐的数值模拟中，除四个主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的单一分潮的数值模拟与同化能达到较好效果外，其他较弱分潮及长周期分潮的数值模拟精度受限于客观条件一直难以提高。

调和常数的获取起初主要是通过对已有观测结果进行插值，后又逐渐被数值模拟的方法所取代。随着卫星观测技术的发展，高度计资料的出现又使得插值法在调和常数的获取中发挥重要作用。

## 1.2 要解决的问题

已知南海地形图，具体区域为  $2^\circ \sim 25^\circ \text{N}$ ， $99^\circ \sim 122^\circ \text{E}$ ；中国近海及周边海域 770 个验潮点和 56 个验潮点的资料，包括 9 个分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ 、 $N_2$ 、 $K_2$ 、 $P_1$ 、 $Q_1$ 、 $S_a$ ）的潮汐调和常数；卫星测量轨道以及南海该区域的海面高度异常数据文件。

通过建立数学模型，解决以下问题：

**问题一：**利用星下观测点海面高度异常值数据，计算全体星下观测点主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的潮汐调和常数；并利用已知的验潮点的调和常数，对得到的星下观测点的潮汐调和常数进行评价或检验。

**问题二：**将问题一中求得的全体星下观测点的主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的潮汐调和常数沿测量卫星轨道进行作图，可发现潮汐调和常数由于内潮对正压潮的调制而导致的在沿轨道方向上存在细结构。需要利用模型，对沿轨道的各分潮的潮汐调和常数进行正压潮和内潮的分离。

**问题三：**基于数据插值或者拟合方法绘制南海各主要分潮的同潮图，并利用验潮点的潮汐调和常数对所绘制的同潮图进行评价或分析。

**问题四：**如果在对沿着卫星轨道的潮汐调和常数分离、插值或拟合中，利用了特定函数进行拟合，请确定该特定函数的最优（高）次数；并考虑与第三问的联系。

## 2. 模型基本假设

为方便问题研究，对题目中某些条件进行简化及合理化假设。

1. 假设各分潮的振幅与迟角为常数，与时间无关；
2. 假设地球为标准球体，半径为 6371km；
3. 假设各分潮的振幅、迟角与地理位置之间满足连续函数关系；
4. 假设实际水位仅是有限主要分潮叠加的结果；
5. 假设忽略交点因子与交点订正角随时间的变化；
6. 假设忽略轨迹弯曲度对潮汐调和常数随轨迹变化带来的影响；
7. 在计算星下观测点与选定轨迹间距离时，忽略空间弧度，假设为平面内点到直线距离；

### 3. 符号说明

符号	意义
$h$	实际水位
$S_0$	余水位
$f_j$	交点因子
$h_j$	调和常数—振幅
$v_j$	分潮相位
$u_j$	交点订正角
$g_j$	调和常数—迟角
$J$	主要分潮数
$\sigma_j$	分潮角速度
$t$	时刻
$v_{0j}$	第 $j$ 个分潮的初相位
$\mu$	Doodson 数
$R$	地球半径 6731km
$L$	地球上已知精度、纬度两点间的距离(km)
$\Delta H$	潮汐调和常数的振幅差
$\Delta$	潮汐矢量差
$RMS_{\Delta H}$	潮汐调和常数的振幅差的均方差
$RMS_{\Delta}$	潮汐矢量差的均方差



## 4. 问题一 星下观测点各主要分潮潮汐调和常数求解及评价检验

### 4.1 问题分析

问题一中要求利用星下观测点海面高度异常值数据集，计算全体 4520 个星下观测点主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的潮汐调和常数，即分潮的振幅和迟角；并利用已知的验潮点的调和常数，对计算得到的星下观测点的潮汐调和常数进行检验评价。

问题一可以划分为两个小问题：

1. 全体 4520 个星下观测点的主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的潮汐调和常数  $h_j$  和  $g_j$  的求解；
2. 利用验潮点调和常数对计算得到星下观测点调和常数进行检验和评价。

### 4.2 星下观测点主要分潮潮汐调和常数的求解

#### 4.2.1 数据处理

##### （一）观测点观测数据存储格式整理

题目中所给出的各个星下观测点的水位检测数据是按照第一行三个数据分别代表该星下观测点的经度、纬度、该观测点的观测数量，其余行两个数据分别代表观测时刻，对应水位值的数据格式来进行存储的。

按照该类型存储的观测数据，并不适用于后续计算处理，因此本文中利用 Matlab 软件对原有数据格式进行处理，将各个星下观测点的测量数据以观测点的经度、纬度为区分标志进行分别存储。最终将全部观测点存入一个  $1 \times 4520$  的矩阵之中，矩阵中的每个元素由各观测点的经度、纬度和在该观测点测量数目构成。该矩阵中的每个元素分别一一映射一个  $n \times 1$  的矩阵，其中  $n$  为每个观测点的测量数目， $n \times 1$  矩阵中的每个元素由该测量点的测量时刻、测量的水位数值构成。具体处理程序源代码详见附录 1。具体处理流程如下图 1 所示：

经处理后，可以方便地获取每一个观测点所观测到的水位高度的时间变化序列，为后续进一步计算奠定基础。

##### （二）观测数据清洗

为避免观测数据过少可能导致的数据失真问题，同时考虑模型求解过程中的方程组未知数个数等因素，因此本文中将测量次数小于 9 次的观测点数据从观测数据中去除，只选取观测次数不低于 300 次的观测点数据进行计算。

经统计，4520 个观测点中共有 871 个观测点的测量次数小于 300 次，去除

该 871 个观测点后，利用剩余的 3649 个进行计算。

### （三）测量数据中的时间转换

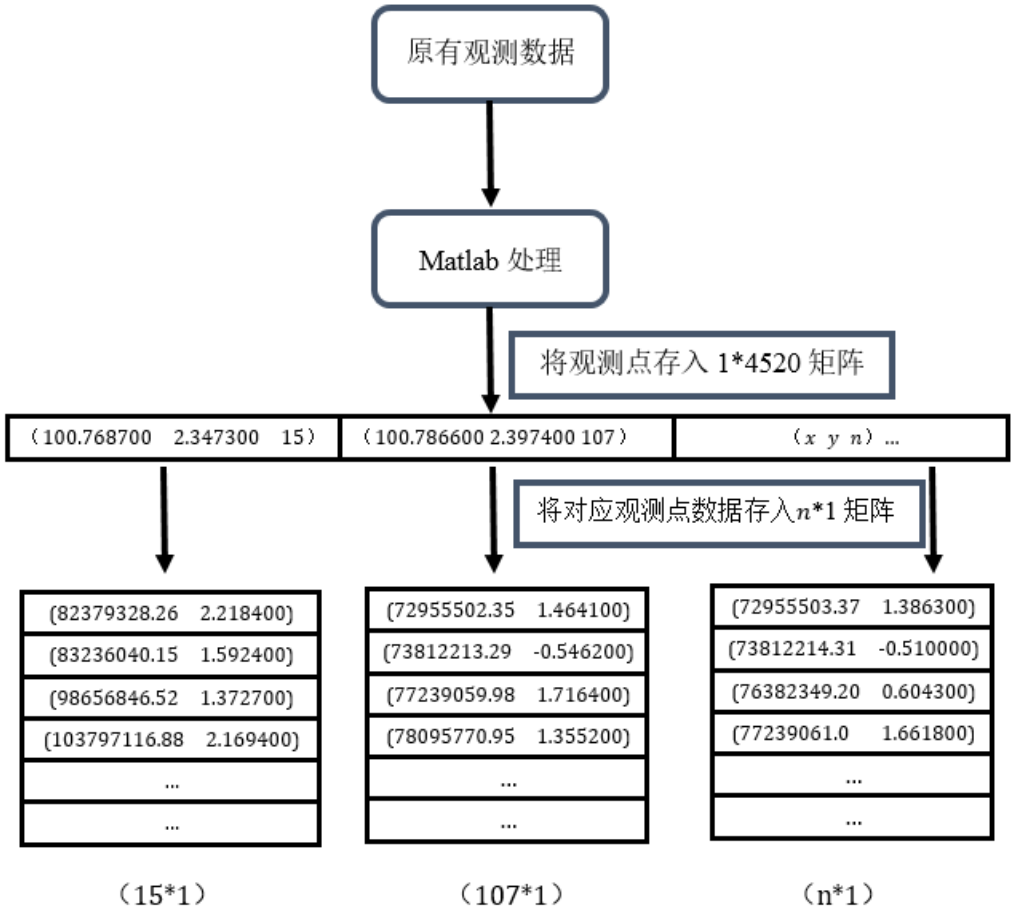
注意到题目高度计资料中所记录的观测时刻是以 2000 年 1 月 1 日 0 点 0 分 0 秒（格林尼治时间）开始计算，题目中验潮站资料中所给出的振幅迟角的求取采用的是北京时间，而调和分析方法中分潮初相位的计算中，则是利用  $Y$  年  $M$  月  $D$  日  $t$  时刻进行的计算。

因此需要将星下观测站中的时刻由以 2000 年 1 月 1 日 0 点 0 分 0 秒（格林尼治时间）先转换为 2000 年 1 月 1 日 8 点 0 分 0 秒（北京时间），然后根据总秒数将其转换为  $Y$  年  $M$  月  $D$  日  $t$  时刻。具体计算方法为：

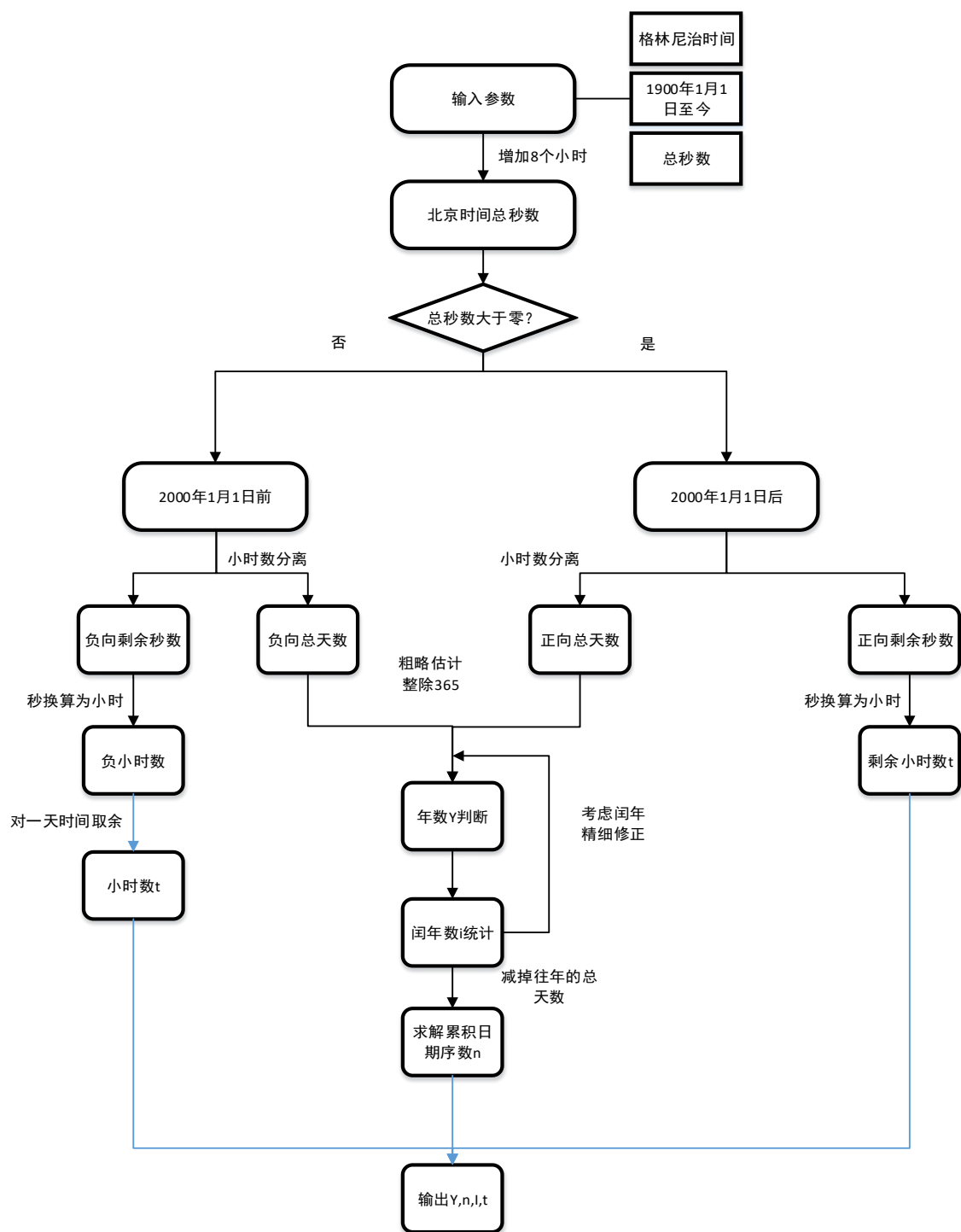
一天中总共的秒数为 86400 秒，将观测数据中记录的总秒数除以 86400，得到总天数。考虑到月份中的大小月份以及闰年、平年等因素，计算得到确切的年月日及时刻。具体计算流程如下图 2 所示，详细程序见附录 2。

### （四）水深数据的处理

题目中所给出的水深数据存储顺序左上角位置为（99E，2N），右下角位置为（122E，25N），与实际图片方位相反，因此需要将水深数据集中的地理位置存储顺序进行调换，从而保证绘制地形图与实际情况相符。



（图 1 原有数据格式处理流程）



(图 2 时间由秒数转换为Y年M月D日i时刻逻辑流程图)

#### 4.2.2 模型准备

由题目中所给的参考资料可知,实际水位可以近似看作有限个主要分潮的叠加。假设有 $J$ 个分潮,则根据题目所给资料可知:

$$h = S_0 + \sum_{j=1}^J f_j h_j \cos(v_j + u_j - g_j) = S_0 + \sum_{j=1}^J f_j h_j \cos(\sigma_j t + v_{0j} + u_j - g_j) \quad (4-1)$$

其中 $h$ 为实际测得的水位， $S_0$ 为余水位， $f_j$ 为交点因子， $h_j, g_j$ 为分潮的调和常数（振幅和迟角）， $\sigma_j$ 为分潮的角速度， $t$ 为时间（单位： $h$ ）， $v_{0j}$ 为第 $j$ 个分潮的初始相位， $u_j$ 为交点订正角。

### （一）交点因子 $f_j$ 与交点订正角 $u_j$ 的计算

其中，交点因子 $f_j$ 与交点订正角 $u_j$ 忽略其随时间的变化，选取资料序列的中间时刻对上述两变量进行计算，公式如（4-2）所示：

$$\begin{cases} f \cos u = \sum_{m=1}^M \rho_m \cos(\Delta\mu_4^m p + \Delta\mu_5^m N') \\ f \sin u = \sum_{m=1}^M \rho_m \sin(\Delta\mu_4^m p + \Delta\mu_5^m N') \end{cases} \quad (4-2)$$

相关参数取值见下表所示：

（表 1  $\rho_m$ 、 $\Delta\mu_4^m$ 、 $\Delta\mu_5^m$  列表）

$\Delta\mu_4$	$\Delta\mu_5$	$M_m$	$M_f$	$O_1$	$P_1$	$K_1$	$J_1$	$OO_1$	$M_2$	$L_2$	$k_2$
-2	-1					0.0002					
-2	0		0.0432					0.1496			
-2	1							0.0296			
0	-2	0.0008			0.0008	0.0001			0.0005		
0	-1			0.1885							
0	0	1	1	1	1	1	1	1	1	1	1
0	1		0.4143			0.1356	0.1980	0.6398			0.2980
0	2		0.0387					0.1342			0.0324
0	3		-					0.0086			
2	-1			0.0002						0.0047	
2	0								0.0006		
2	1								0.0002		
2	2										

上述求解交点因子 $f_j$ 与交点订正角 $u_j$ 公式不适用于 $M_1$ 分潮， $M_1$ 分潮的求解公式如下所示：

$$\begin{aligned} f \cos u = & -0.008 \cos(-p-2N') + 0.094 \cos(-p-N') + 0.510 \cos p - 0.041 \cos(p-N') \\ & + 1.418 \cos p + 0.284 \cos(p+N') - 0.008 \cos(p+2N') \end{aligned}$$

(4-3)

$$f \sin u = -0.008 \sin(-p - 2N') + 0.094 \sin(-p - N') - 0.510 \sin p - 0.041 \sin(p - N') \\ + 1.418 \sin p + 0.284 \sin(p + N') - 0.008 \sin(p + 2N')$$

(4-4)

## (二) 分潮角速度的计算

分潮角速度 $\sigma_j$ 的计算主要通过如下公式:

$$\sigma = \mu_1 \dot{\tau} + \mu_2 \dot{s} + \mu_3 \dot{h}' + \mu_4 \dot{p} + \mu_5 \dot{N}' + \mu_6 \dot{p}' \quad (4-5)$$

其中 $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$ 为 Doodson 数:

相关参数取值见下表 2:

(表 2 分潮角速度计算公式中的相关参数取值)(单位: 度/平太阳时)

$\dot{\tau}$	$\dot{s}$	$\dot{h}'$	$\dot{p}$	$\dot{N}'$	$\dot{p}'$
14.4920521	0.5490165	0.0410686	0.0046418	0.0022064	0.0000019
1	3	4	3	1	6

## (三) 分潮初相位的计算

$Y$  年  $M$  月  $D$  日  $t$  时刻 (实际计算中是观测数据的起始时间) 的天文初相角:

$$v_0 = \mu_1 \tau + \mu_2 s + \mu_3 h' + \mu_4 p + \mu_5 N' + \mu_6 p' + \mu_0 90 \quad (4-6)$$

其中, 各个参数的取值如下表所示:

(表 3 初相位计算参数取值)

$\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$  为 Doodson 数,

$$\begin{cases} s = 277.02 + 129.3848(Y - 1900) + 13.1764(n + i + \frac{t}{24}) \\ h' = 280.19 - 0.2387(Y - 1900) + 0.9857(n + i + \frac{t}{24}) \\ p = 334.39 + 40.6625(Y - 1900) + 0.1114(n + i + \frac{t}{24}) \\ N' = 100.84 + 19.3282(Y - 1900) + 0.0530(n + i + \frac{t}{24}) \\ p' = 281.22 + 0.0172(Y - 1900) + 0.00005(n + i + \frac{t}{24}) \\ \tau = 15t - s + h' \end{cases}$$

式中  $i$  为 1900 年至  $Y$  年的闰年数,  $i = \text{int}(\frac{Y-1901}{4})$ ;  $n$  为从  $Y$  年 1 月 1 日开始计算的累积日期序数, 1 月 1 日的日期序数为 0,  $t$  为时间 (单位: 小时)。以上各式中的单位是度。

#### 4.2.3 模型建立与求解

根据公式 (4-1), 结合题目所要求解的四个主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的潮汐调和常数: 振幅和迟角。可以知道, 每一方程中含有 9 个未知数; 而不同时刻对应不同的测量值, 因此每一个星下测量点可以列出相应的含有  $n$  个方程的方程组。

观察题目中所给出的 4520 个星下观测点的测量数据, 98.87% 的观测点有 9 个以上的测量值; 意味着有 98.87% 的观测点所列的方程组为超静定方程组, 即方程组中方程数目大于未知数个数。

##### (一) 最小二乘法

基于上述问题, 因此在进行上述方程组的计算求解时, 采用最小二乘法<sup>[3]</sup>进行求解。首先对公式 (4-1) 利用三角函数中和差化积公式进行变换:

$$f_j \cos(\sigma_j t + V_{0j} + u_j) H_j \cos g_j + f_j \sin(\sigma_j t + V_{0j} + u_j) H_j \sin g_j \quad (4-7)$$

对应  $J$  个分潮, 则有:

$$h = S_0 + \sum_{j=1}^J f_j \cos(\sigma_j t + V_{0j} + u_j) H_j \cos g_j + \sum_{j=1}^J f_j \sin(\sigma_j t + V_{0j} + u_j) H_j \sin g_j \quad (4-8)$$

假设某一星下观测点有  $n$  个时刻的观测数据, 则此时对应的方程组为:

$$\begin{cases} S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_1 + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_1 + V_0 + u_j) = h_1 \\ S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_2 + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_2 + V_0 + u_j) = h_2 \\ \vdots \\ S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_n + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_n + V_0 + u_j) = h_n \end{cases} \quad (4-9)$$

根据最小二乘法的求解思想,通过计算使得拟合值与实际值之差的平方和最小,即下面公式的数值达到最小:

$$\Delta = \sum_{i=1}^n (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{im}x_m + S_0 - h_i)^2 \quad (4-10)$$

即要求 $\Delta$ 相对于各未知数的偏导数均为0,即:

$$\frac{\partial \Delta}{\partial S_0} = \frac{\partial \Delta}{\partial x_1} = \frac{\partial \Delta}{\partial x_2} = \cdots = \frac{\partial \Delta}{\partial x_m} = 0 \quad (4-11)$$

进而可以利用最小二乘法求得上述方程组的解,最终得到星下观测点的四个主要分潮的潮汐调和常数。

#### 4.2.4 求解结果

利用 Matlab 编程计算得到星下观测点处对应的各主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的潮汐调和常数。所求得的星下观测点的四个主要分潮的潮汐调和常数存储于 Excel 表格中,数据格式为 $[m*10]$ 的矩阵 ( $m$  代表全体参与计算的星下观测点的个数), 每一行的 13 个数据分别代表: 星下观测点的经度、纬度、 $M_2$  分潮振幅、 $S_2$  分潮振幅、 $K_1$  分潮振幅、 $O_1$  分潮振幅、 $M_2$  分潮迟角、 $S_2$  分潮迟角、 $K_1$  分潮迟角和  $O_1$  分潮迟角。

部分计算结果如下表 4 所示,整体计算结果见附件 1,求解源代码见附录 3。

(表 4 部分星下观测点四个主要分潮的潮汐调和常数)

	A	B	C	D	E	F	G	H	I	J
1	观测点经度	观测点纬度	M2分潮振幅	S2分潮振幅	K1分潮振幅	O1分潮振幅	M2分潮迟角	S2分潮迟角	K1分潮迟角	O1分潮迟角
2	100.7687	2.3473	24.14012974	15.54630977	32.13740449	14.23455468	0.33290453	29.22357686	331.0350417	298.9235193
3	100.7866	2.3974	10.91734684	13.60543792	30.65828146	26.89177264	279.8059249	203.0605117	323.7188777	73.25584276
4	100.8045	2.4474	20.18621927	16.10344914	18.43265718	26.4509707	351.6930855	213.6201613	18.63733788	92.79841123
5	100.8224	2.4974	17.82437323	14.20711313	12.55810474	17.70617772	61.49675557	256.5635462	45.39174145	129.5555701
6	100.8402	2.5474	20.71137871	11.12534141	12.95016519	17.88361911	117.5631829	333.250321	71.27968054	142.3971457
7	100.8581	2.5974	21.57984527	14.42189179	13.22801105	15.87766127	185.0748345	48.1146494	135.9594334	174.5905895
8	100.876	2.6474	21.76070946	19.17185803	14.01480204	20.09632878	226.493328	109.9259464	132.7535194	225.5125708
9	100.8939	2.6975	21.08080637	19.87153714	13.39965466	21.76101127	297.8887476	185.9244138	177.2959391	236.386212
10	100.9118	2.7475	18.50496502	18.78902529	10.35120415	20.60573978	356.4238172	235.1051959	214.4003835	270.3191165
11	100.9297	2.7975	23.60172914	17.55148327	11.89880032	21.20868375	44.25205166	312.9111168	249.6822462	303.9814603
12	100.9475	2.8475	15.85444651	23.54052554	14.37754877	15.9393135	112.6653663	2.758773868	269.2297508	340.3183978
13	100.9654	2.8975	15.03379045	17.49449481	15.44181438	23.41155444	164.163575	48.07221863	283.00044	8.463741832
14	100.9833	2.9475	10.47210339	16.83960524	15.97898511	15.8123229	250.6993058	112.8392367	302.3933395	45.31950093
15	101.0012	2.9975	16.55361795	18.56425679	17.33221651	23.32474978	326.0075571	177.6198071	6.289493262	64.72837985
16	101.0191	3.0476	21.10139246	21.90389909	16.30085263	27.34958093	117.193287	258.1021556	15.49290713	91.06126369
17	101.037	3.0976	20.31728253	12.49483878	11.83186393	24.15591996	59.7904184	337.8722245	92.02269828	105.4840824
18	101.0549	3.1476	19.10555437	17.75023601	16.90484727	25.97950872	145.477789	13.31338619	104.2905889	144.6230928
19	101.0728	3.1976	19.52076415	21.48644003	13.27118722	24.01623755	201.4235591	82.58140639	150.6155653	170.360313
20	101.0907	3.2476	17.468833	18.70667755	10.309245	18.23947072	260.7311843	135.9928771	168.3850887	219.7901832
21	101.1086	3.2976	14.16519226	9.178649364	18.32676043	12.29044208	262.485796	190.6006259	152.7199044	210.1356577
22	101.1265	3.3476	36.59571363	21.24614222	36.19105858	10.80279497	180.331592	238.0663428	222.1791675	190.6302456
23	102.1982	6.3242	8.124400165	3.735075293	0.512662891	3.644424652	259.5504582	79.49704096	255.6491245	116.8559838
24	102.2163	6.3741	2.664894158	1.76723146	6.20010472	0.775559269	279.5296364	120.8206751	108.2144343	28.05204746

### 4.3 利用验潮点调和常数对计算结果进行检验评价

#### 4.3.1 数据处理

##### (一) 处理验潮站数据存储格式

题目中所给出的验潮点的数据格式是按照 4 个分潮分别存储,每个分潮中采取[tide\_station\_num\*4]矩阵的形式对数据进行存储(tide\_station\_num 代表验潮点个数),每一行四个数据分别表示该位置点的振幅(厘米)、该位置点的迟角(度)、该位置点的纬度(度)、该位置点的经度(度)。

此种数据存储格式不利于后续计算,为方便验潮点数据与星下观测点数据之间进行比对,将 4 个分潮的验潮点数据按照 4.2.4 中所述的星下观测点的数据存储格式整合到一个[234\*10]的矩阵中,部分数据如表 5 所示,完整数据见附件 2。



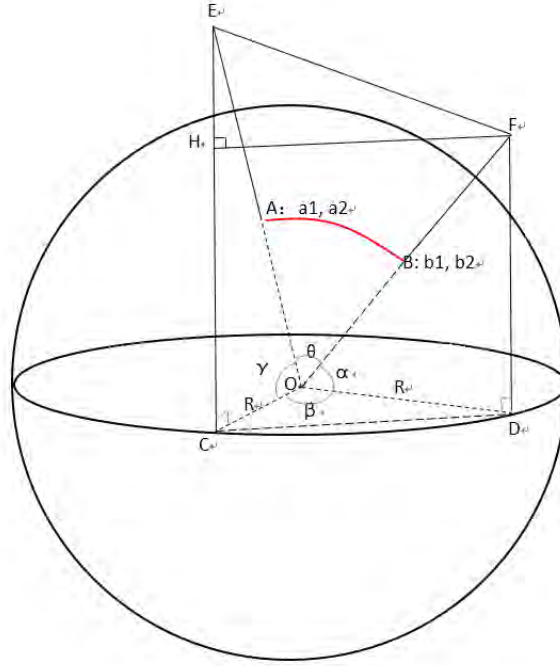
(表 5 部分验潮站四个主要分潮的潮汐调和常数)

	A	B	C	D	E	F	G	H	I	J
1	验潮站经度	验潮站纬度	M2分潮振幅	S2分潮振幅	K1分潮振幅	O1分潮振幅	M2分潮迟角	S2分潮迟角	K1分潮迟角	O1分潮迟角
2	99.816667	11.795	5.874719	1.409046	50.438524	33.652766	164.893211	225.494503	181.951332	131.015764
3	102.106667	6.226667	17.164304	8.168535	25.576541	13.060076	254.802101	292.665494	351.879808	300.160507
4	103.186667	5.265	29.832596	12.057419	49.077744	30.222835	239.382426	282.960128	1.330298	315.67556
5	103.43	3.975	53.40618	17.862157	52.686445	35.556355	263.519854	310.835634	12.744196	330.349817
6	104.14	2.806667	60.065477	18.777993	45.327471	33.823347	273.006208	323.337368	22.286495	342.494515
7	106.283333	18.083333	24.606623	6.668539	32.552167	40.291182	225.204327	302.448283	218.611127	142.050117
8	106.816667	20.666667	5.872028	4.412443	70.259777	78.916583	272.804735	344.333854	212.765094	139.906187
9	107.066667	10.333333	77.099657	28.672405	60.212331	45.233673	271.65762	325.034765	74.31877	15.382044
10	108.616667	19.1	18.949943	5.675352	55.49388	66.145438	61.268197	118.795747	71.574329	10.658705
11	109.083333	21.483333	48.473637	11.301431	93.504572	102.484722	179.158539	236.858753	95.596823	32.558698
12	110.283333	20.016667	23.352377	13.127846	41.494247	51.850537	254.089606	322.990843	96.272854	32.625544
13	111.833333	21.583333	64.538267	28.21417	41.182904	35.059085	292.37297	332.217591	313.585985	263.711409
14	113.066667	3.216667	19.210519	6.167435	43.618599	35.962842	52.779287	46.270158	319.073571	268.621419
15	113.55	22.166667	46.347852	18.580124	37.466875	30.699787	288.25837	321.965912	307.229345	257.723587
16	113.971667	4.391667	17.754224	8.761419	37.750717	32.767426	334.396923	5.653918	316.167805	266.897604
17	114.216667	22.3	40.167327	16.014656	37.643063	29.867347	239.962346	266.554256	285.215364	236.975608
18	115.35	22.75	27.750799	11.002604	32.622999	26.294924	255.264306	277.565197	296.78834	249.674214
19	116.066667	5.983333	23.932879	10.269386	36.059625	30.992014	318.910849	354.35997	314.504963	264.376795
20	117.75	2.1	96.569961	59.04559	19.266846	14.578782	174.538082	225.391968	270.718885	239.444654
21	117.883333	4.233333	85.711096	52.440036	21.544203	15.226164	167.68507	215.778715	266.663376	232.991143
22	118.066667	24.45	183.344075	52.945566	34.18706	28.274387	351.438608	43.285717	281.481107	237.357997
23	118.066667	5.81	41.868064	22.977829	36.1285	32.951563	318.237019	0.375936	325.517475	281.295496
24	118.733333	9.75	30.695443	16.30078	34.654662	29.85402	311.870758	354.220801	325.410191	279.735325
25	120.291667	22.61	17.384856	6.834445	16.997122	16.107078	219.436801	231.211139	286.429464	240.082189

### 4.3.2 模型准备

#### （一）地球上已知经度纬度的两点间的球面距离

本文中假设地球是一个 $R$ 半径为 6371km 的标准球体，如图 3 所示。



（图 3 地球上已知经纬度的两点之间的距离）

已知地球表面 $A(a_1, a_2)$ 与 $B(b_1, b_2)$ 两点的经度、纬度，则 $A$ 、 $B$ 两点之间的最短距离公式可以通过三角几何关系求得为：

$$L = R * \arccos(\sin(a_2) \sin(b_2) + \cos(a_2) \cos(b_2) \cos(a_1 - b_1)) \quad (4-12)$$

#### （二）验潮站与星下观测站最近点匹配

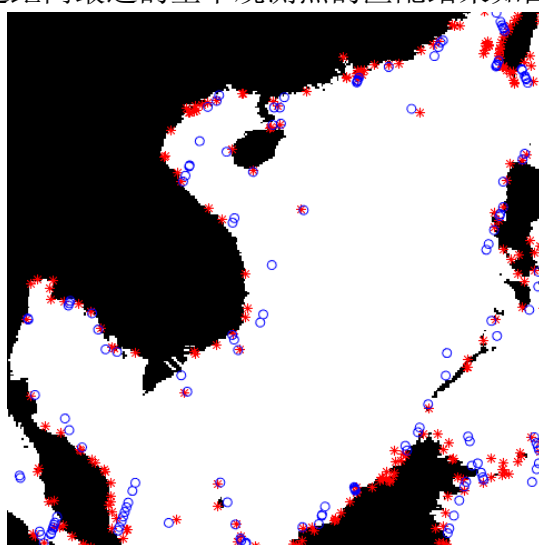
采用循环遍历算法，依次计算每一个验潮站与全体星下观测点之间的距离，提取距离最小值，找到每一个验潮站对应的最接近的星下观测点，为后续利用验潮站分潮的潮汐调和常数评价求得的星下观测点的潮汐调和常数做准备。

将所得数据结果存入一个 $[243*5]$ 的矩阵中，具体数据格式为：每一行的五个数据分别代表验潮站经度、纬度，星下观测点经度、纬度以及两点之间的球面距离。部分结果如下表 6 所示，完整验潮站与最近点匹配结果见附件 3。

(表 6 验潮站和与之距离最近的星下观测点的部分匹配表)

	A	B	C	D	E
1	验潮站经度	验潮站纬度	观测点经度	观测点纬度	球面距离 (km)
2	99.816667	11.795	99.8952	11.8018	8.581358076
3	102.106667	6.226667	102.1982	6.3242	14.83144761
4	103.186667	5.265	104.4848	4.8065	152.5591971
5	103.43	3.975	104.1077	3.7565	79.01347479
6	104.14	2.806667	103.8033	2.9063	38.99947895
7	106.283333	18.083333	106.6337	17.9749	38.95882527
8	106.816667	20.666667	107.5923	20.924	85.55464495
9	107.066667	10.333333	106.457	9.4134	122.1643832
10	108.616667	19.1	108.4006	19.0005	25.26146831
11	109.083333	21.483333	108.0793	21.4281	104.0877176
12	110.283333	20.016667	110.3713	20.156	18.01191978
13	111.833333	21.583333	110.8747	21.338	102.8869339
14	113.066667	3.216667	112.4976	3.4365	67.73523433
15	113.55	22.166667	113.985	21.9779	49.49581335
16	113.971667	4.391667	113.9263	4.4692	9.981098844
17	114.216667	22.3	114.0707	22.1745	20.50487809
18	115.35	22.75	115.3524	22.6379	12.46738256
19	116.066667	5.983333	116.1391	6.192	24.54599852
20	117.75	2.1	117.8824	2.6459	62.4581404
21	117.883333	4.233333	118.3839	4.0462	59.28682136
22	118.066667	24.45	117.6804	23.776	84.57933289
23	118.066667	5.81	118.9206	5.5378	99.21711159
24	118.733333	9.75	117.8001	9.4016	109.4127509
25	120.291667	22.61	119.9769	22.7089	34.11967441
26	120.966667	14.583333	121.7926	13.8511	120.6437836
27	121	6.066667	121.7473	6.3614	88.87113604
28	100	8.533	100.182	8.5719	20.47455692
29	100	13.35	99.8764	11.8516	167.1534383
30	100.1	2.617	100.3801	2.7153	32.97620858
31	100.2	2.5	100.4338	2.5653	26.96787332
32	100.283	13.533	101.6893	12.6087	183.7513182
33	100.283	5.25	99.5549	5.0157	84.7406103

相应验潮点和与之距离最近的星下观测点的匹配结果如图 4 所示:



(图 4 验潮站和与之距离最近的星下观测点的分布图  
红色星号: 验潮站; 蓝色圆圈: 星下观测站)

### 4.3.3 模型建立与求解

#### (一) 检验评估参数及求解

为验证通过上述 4.2 章节中所建模型的准确性,本文中利用验潮站的各主要分潮的潮汐调和常数对星下观测点求得数据进行对比评估。

对比评估以各主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的潮汐调和常数的振幅差 $\Delta H$ 和潮汐矢量差 $\Delta$ 、振幅差均方差 $RMS_{\Delta H}$ 和矢量差均方差 $RMS_{\Delta}$ ,以上四个参数来进行衡量。

上述四个参数的计算,均为每个验潮站和与之距离最近的星下观测点的相应参数之间的计算。振幅差 $\Delta H$ 的计算公式为:

$$\Delta H = |H_r - H_c| \quad (4-13)$$

其中,  $H_r$ 为验潮站的各分潮的振幅;  $H_c$ 为星下观测点的各分潮的振幅。

分潮潮汐矢量差 $\Delta$ 的计算公式为:

$$\Delta = [(H_r \cos G_r - H_c \cos G_c)^2 + (H_r \sin G_r - H_c \sin G_c)^2]^{1/2} \quad (4-14)$$

振幅差均方差 $RMS_{\Delta H}$ 和矢量差均方差 $RMS_{\Delta}$ 的计算公式<sup>[4]</sup>为:

$$RMS_{\Delta H} = ((\sum_{i=1}^n (\Delta H_i - \Delta H^a)^2)/n)^{1/2} \quad (4-15)$$

$$RMS_{\Delta} = ((\sum_{i=1}^n (\Delta_i - \Delta^a)^2)/n)^{1/2} \quad (4-16)$$

其中,  $\Delta H^a$ 为振幅差平均值,  $\Delta^a$ 为平均值,  $n$ 为验潮站总个数。

### 4.3.4 求解结果

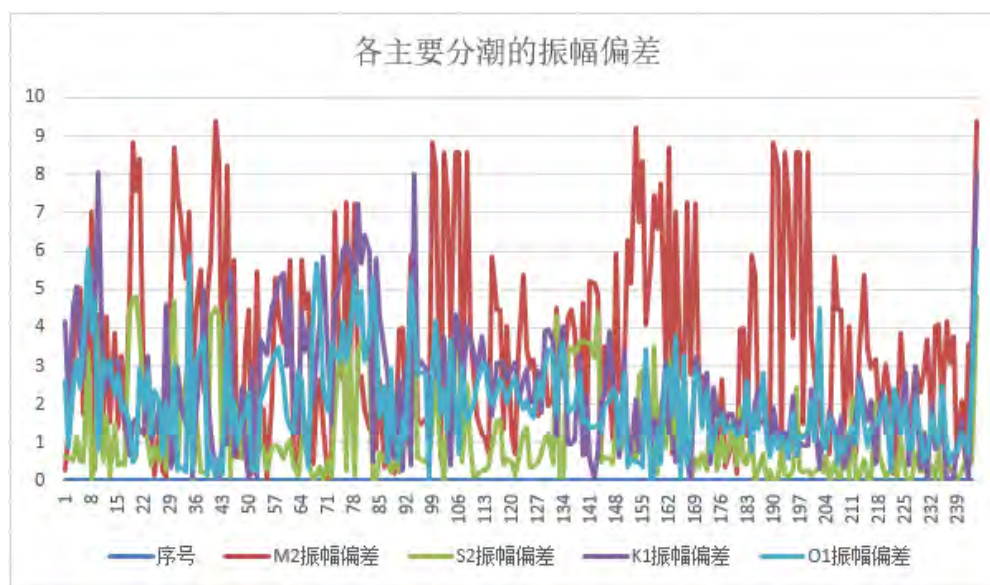
利用 4.3.1 和 4.3.2 中整理的数据格式,通过 MATLAB 进行编程,分别计算各个验潮点不同分潮的潮汐调和常数所对应的上述四个参数:振幅差 $\Delta H$ ,潮汐矢量差 $\Delta$ ,振幅差均方差 $RMS_{\Delta H}$ 以及矢量差均方差 $RMS_{\Delta}$ 。

#### (一) $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ 分潮的振幅差 $\Delta H_{M_2}$ 、 $\Delta H_{S_2}$ 、 $\Delta H_{K_1}$ 、 $\Delta H_{O_1}$

将运算结果存储进一个[243\*9]的矩阵中,具体的数据格式为:每一行的 9 个数据分别代表验潮站经度、纬度,与之对应的距离最近的观测点的经度、纬度,两点之间的球面距离<sup>[5]</sup>,以及  $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$  分潮的振幅差 $\Delta H_{M_2}$ 、 $\Delta H_{S_2}$ 、 $\Delta H_{K_1}$ 、 $\Delta H_{O_1}$ 。部分数据结果见表 7 所示,处理数据结果折线图见图 5,完整处理数据见附件 4。

(表 7  $\Delta H_{M2}$ 、 $\Delta H_{S2}$ 、 $\Delta H_{K1}$ 、 $\Delta H_{O1}$  部分对比数据)

	A	B	C	D	E	F	G	H	I
1	验潮站经度	验潮站纬度	观测点经度	观测点纬度	球面距离 (km)	M2振幅偏差	S2振幅偏差	K1振幅偏差	O1振幅偏差
2	99.816667	11.795	99.8952	11.8018	8.581358076	0.275836991	0.637868459	4.187819909	2.605049037
3	102.106667	6.226667	102.1982	6.3242	14.83144761	1.179968721	0.610690736	2.103670752	0.903717776
4	103.186667	5.265	104.4848	4.8065	152.5591971	2.707550029	0.545588949	4.56861646	2.506654782
5	103.43	3.975	104.1077	3.7565	79.01347479	4.968956269	1.130648972	5.068629846	3.17585351
6	104.14	2.806667	103.8033	2.9063	38.99947895	5.046486424	0.548631919	3.568596704	2.455864799
7	106.283333	18.083333	106.6337	17.9749	38.95882527	1.752382992	0.489729539	2.689553506	3.705630684
8	106.816667	20.666667	107.5923	20.924	85.55464495	2.629312536	3.407462105	5.799327464	6.015434109
9	107.066667	10.333333	106.457	9.4134	122.1643832	7.003073028	0.054513386	5.203964129	3.866163632
10	108.616667	19.1	108.4006	19.0005	25.26146831	0.304983116	0.38046544	4.315447497	4.724098285
11	109.083333	21.483333	108.0793	21.4281	104.0877176	4.358893141	6.889171085	8.057833769	0.856261169
12	110.283333	20.016667	110.3713	20.156	18.01191978	1.518521667	0.548029021	2.254000189	3.225012319
13	111.833333	21.583333	110.8747	21.338	102.8869339	4.311228908	1.714858385	2.415311881	2.802192665
14	113.066667	3.216667	112.4976	3.4365	67.73523433	1.516144171	0.097845983	3.060451629	3.121673855
15	113.55	22.166667	113.985	21.9779	49.49581335	3.858585	1.51038751	3.173294478	2.271075581
16	113.971667	4.391667	113.9263	4.4692	9.981098844	1.400684187	0.399203307	3.114641294	2.799486522
17	114.216667	22.3	114.0707	22.1745	20.50487809	3.272159017	0.488045099	2.219354035	1.896539727
18	115.35	22.75	115.3524	22.6379	12.46738256	1.337178941	0.407710326	1.64667178	1.980441377
19	116.066667	5.983333	116.1391	6.192	24.54599852	4.858825404	5.450232603	0.689158736	1.4962371
20	117.75	2.1	117.8824	2.6459	62.4581404	8.8255672	4.774838806	1.541076107	0.505681537
21	117.883333	4.233333	118.3839	4.0462	59.28682136	7.582811737	4.796594588	1.600733274	0.679982952
22	118.066667	24.45	117.6804	23.776	84.57933289	17.39789954	3.926516236	1.27818358	2.903096198
23	118.066667	5.81	118.9206	5.5378	99.21711159	1.173542021	1.800483612	1.360778155	1.913160712
24	118.733333	9.75	117.8001	9.4016	109.4127509	2.233971729	1.012414845	3.244616758	2.852528702
25	120.291667	22.61	119.9769	22.7089	34.11967441	1.464129553	0.425152219	1.084538918	1.241800966
26	120.966667	14.583333	121.7926	13.8511	120.6437836	0.147138625	0.894423905	1.304704053	2.357707413



(图 5 各主要分潮的振幅偏差分布)

## (二) $M2$ 、 $S2$ 、 $K1$ 、 $O1$ 的 $RMS_{\Delta H_{M2}}$ 、 $RMS_{\Delta H_{S2}}$ 、 $RMS_{\Delta H_{K1}}$ 、 $RMS_{\Delta H_{O1}}$

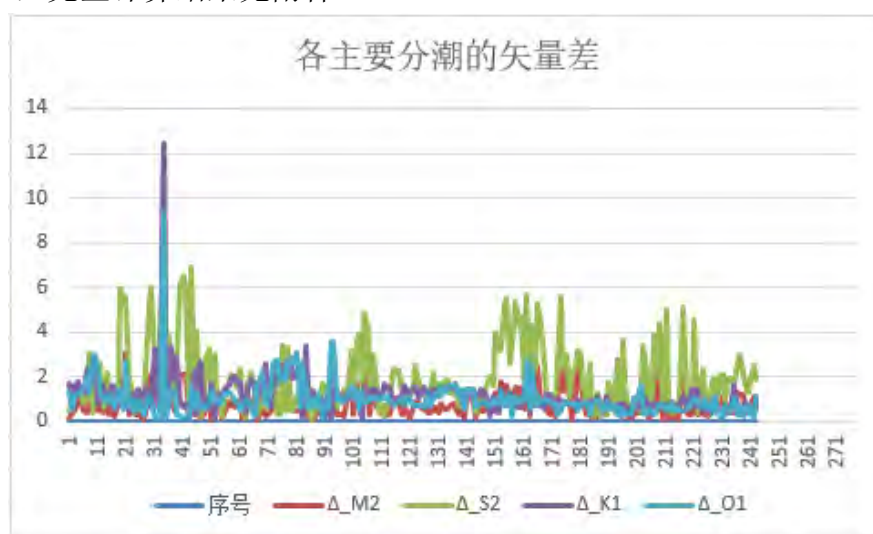
通过 Matlab 计算得到验潮站与之对应的距离最近的星下观测点的四个分潮的潮汐调和常数的振幅差的均方差:  $RMS_{\Delta H_{M2}}$ 、 $RMS_{\Delta H_{S2}}$ 、 $RMS_{\Delta H_{K1}}$ 、 $RMS_{\Delta H_{O1}}$ 。计算结果如下表 8 所示:

(表 8  $RMS_{\Delta HM2}$ ,  $RMS_{\Delta HS2}$ ,  $RMS_{\Delta HK1}$ ,  $RMS_{\Delta HO1}$ )

$RMS_{\Delta HM2}$	$RMS_{\Delta HS2}$	$RMS_{\Delta HK1}$	$RMS_{\Delta HO1}$
4.899813123	1.811295531	2.664940198	2.251510839

### (三) M2、S2、K1、O1 的矢量差 $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$

通过 Matlab 计算得到的验潮站与之对应距离最近的星下观测点的四个分潮的矢量差 $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$ ，部分计算结果如下表 9 所示，处理数据结果折线图见图 6，完整计算结果见附件 5。



(图 6 各主要分潮的矢量差)

(表 9  $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$  部分数据)

	A	B	C	D	E	F	G	H	I
▲	验潮站经度	验潮站纬度	观测点经度	观测点纬度	球面距离 (km)	$\Delta_{M2}$	$\Delta_{S2}$	$\Delta_{K1}$	$\Delta_{O1}$
1	99.816667	11.795	99.8952	11.8018	8.581358076	0.990882415	0.495157144	5.160452444	3.779084088
2	102.106667	6.226667	102.1982	6.3242	14.83144761	2.231571318	0.928628076	2.119875141	1.708217905
3	103.186667	5.265	104.4848	4.8065	152.5591971	2.864105431	1.611125194	4.798842541	3.402242354
4	103.43	3.975	104.1077	3.7565	79.01347479	5.689185227	1.321158768	5.352393155	3.556600341
5	104.14	2.806667	103.8033	2.9063	38.99947895	6.692763003	1.424593425	4.4486025	3.794744787
6	106.283333	18.083333	106.6337	17.9749	38.95882527	2.930267678	0.791521915	3.820645086	4.351875171
7	106.816667	20.666667	107.5923	20.924	85.55464495	2.660391554	1.093725739	6.996006295	6.098118426
8	107.066667	10.333333	106.457	9.4134	122.1643832	8.416849262	3.056876372	5.222832953	3.884229977
9	108.616667	19.1	108.4006	19.0005	25.26146831	3.058727709	0.423170067	6.617455864	8.49954866
10	109.083333	21.483333	108.0793	21.4281	104.0877176	8.716085651	1.064931117	8.304994976	9.040531602
11	110.283333	20.016667	110.3713	20.156	18.01191978	2.984042004	1.221255508	2.39871126	7.087581838
12	111.833333	21.583333	110.8747	21.338	102.8869339	7.812016088	2.57475505	5.486503687	3.613623804
13	113.066667	3.216667	112.4976	3.4365	67.73523433	2.318502519	0.665776304	4.472378747	3.408504283
14	113.55	22.166667	113.985	21.9779	49.49581335	5.409888936	2.191554085	3.597267623	2.447381859
15	113.971667	4.391667	113.9263	4.4692	9.981098844	1.966301767	1.240300754	3.80614343	3.72014351
16	114.216667	22.3	114.0707	22.1745	20.50487809	4.005765353	1.229443889	4.966591502	2.46351301
17	115.35	22.75	115.3524	22.6379	12.46738256	1.394998324	0.820278905	1.774568582	1.999300047
18	116.066667	5.983333	116.1391	6.192	24.54599852	5.20964371	1.086871741	4.11518413	4.491993311
19	117.75	2.1	117.8824	2.6459	62.4581404	10.48570578	5.949646466	2.169949229	3.072122129
20	117.883333	4.233333	118.3839	4.0462	59.28682136	7.58324452	5.313798743	2.400969974	1.801131358
21	118.066667	24.45	117.6804	23.776	84.57933289	18.61215596	5.615944141	5.503376493	8.108195306
22	118.066667	5.81	118.9206	5.5378	99.21711159	1.908874523	1.936548862	1.363124901	2.178083525
23	118.733333	9.75	117.8001	9.4016	109.4127509	3.15222856	1.283621621	3.684899683	3.006145185
24	120.291667	22.61	119.9769	22.7089	34.11967441	1.720817667	0.833631474	1.337993923	1.243229048
25	120.966667	14.583333	121.7926	13.8511	120.6437836	1.967735029	0.472094078	4.393165419	2.874576399
26	121	6.066667	121.7473	6.3614	88.87113604	1.918907972	1.21676051	1.918413313	2.128448173
27	100	8.533	100.182	8.5719	20.47455692	0.685152755	0.694764492	3.511546317	1.033294642
28	100	13.35	99.8764	11.8516	167.1534383	2.574218867	2.333407008	4.61376294	2.15377503
29	100.1	2.617	100.3801	2.7153	32.97620858	12.15651652	5.166109973	2.013632123	3.527323462
30	100.2	2.5	100.4338	2.5653	26.96787332	12.89735165	5.991978325	2.756658261	3.124069236
31	100.283	13.533	101.6893	12.6087	183.7513182	5.563826772	2.848444642	9.65983726	0.636183117
32	100.283	5.25	99.5549	5.0157	84.7406103	5.064137014	2.972843723	1.983999893	0.385439324
33									

#### (四) M2、S2、K1、O1 矢量差均方差 $RMS_{\Delta M2}$ 、 $RMS_{\Delta S2}$ 、 $RMS_{\Delta K1}$ 、 $RMS_{\Delta O1}$

通过 Matlab 计算得到验潮站与之对应的距离最近的星下观测点的四个分潮的潮汐调和常数的矢量差的均方差： $RMS_{\Delta M2}$ 、 $RMS_{\Delta S2}$ 、 $RMS_{\Delta K1}$ 、 $RMS_{\Delta O1}$ 。计算结果如下表 10 所示：

(表 10  $RMS_{\Delta M2}$ 、 $RMS_{\Delta S2}$ 、 $RMS_{\Delta K1}$ 、 $RMS_{\Delta O1}$ )

$RMS_{\Delta HM2}$	$RMS_{\Delta HS2}$	$RMS_{\Delta HK1}$	$RMS_{\Delta HO1}$
4.899813123	1.811295531	2.664940198	2.251510839

#### (五) 根据计算结果对提取得到的潮汐调和常数进行分析、评价

通过对上文中计算得到的验潮站和与之距离最近的星下观测点的四个主要分潮 M2、S2、K1、O1 的振幅差 $\Delta H$ ，潮汐矢量差 $\Delta$ ，振幅差均方差 $RMS_{\Delta H}$ 以及矢量差均方差 $RMS_{\Delta}$ 分析可得：

所求得的星下观测点的潮汐调和常数的振幅差 $\Delta H_{M2}$ 、 $\Delta H_{S2}$ 、 $\Delta H_{K1}$ 、 $\Delta H_{O1}$ 分别对应的最小值、最大值、均方差和平均值如下表 11 所示：

(表 11  $\Delta H_{M2}$ 、 $\Delta H_{S2}$ 、 $\Delta H_{K1}$ 、 $\Delta H_{O1}$ 的最小值、最大值、均方差和平均值)

	$\Delta H_{M2}$	$\Delta H_{S2}$	$\Delta H_{K1}$	$\Delta H_{O1}$
<i>min</i>	0.011690255	0.000380105	2.783234548	0.006520839
<i>max</i>	9.39789954	4.796594588	8.057833769	6.015434109
<i>RMS</i>	3.899813123	1.811295531	2.664940198	2.251510839
<i>average</i>	4.162029565	1.521909977	2.452920851	2.18011436

所求得的星下观测点的潮汐调和常数的矢量差 $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$ 分别对应的最小值、最大值、均方差和平均值如下表 12 所示：

(表 12  $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$ 的最小值、最大值、均方差和平均值)

	$\Delta_{M2}$	$\Delta_{S2}$	$\Delta_{K1}$	$\Delta_{O1}$
<i>min</i>	0.305116664	0.071881215	0.391730165	0.343244727
<i>max</i>	12.21448139	8.841530676	10.44549929	10.04872624
<i>RMS</i>	5.982136586	1.522593902	2.832918769	2.45432506
<i>average</i>	5.283253318	1.932630333	3.576777708	3.097343358

由表 11 可以看出，全部验潮站各个分潮的潮汐调和常数振幅差 $\Delta H$ 的最小值为 0.000380105，最大值为 9.39789954，振幅均方差 $RMS_{\Delta H}$ 的最小值为 1.811295531，最大值为 3.899813123。

由表 12 可以看出，全部验潮站各个分潮的潮汐调和常数矢量差 $\Delta$ 的最小值为 1.522593902，最大值为 12.21448139，振幅均方差 $RMS_{\Delta H}$ 的最小值为 1.522593902，最大值为 5.982136586。

由此可得，所求得的星下观测点的四个主要分潮的潮汐调和常数与验潮站的已知数据吻合度较高，误差较小，同时证明验证了本模型的准确性。



## 5. 问题二 观测点调和常数沿轨道作图及正压潮、内潮分离

### 5.1 问题分析

问题二中要求沿卫星轨道绘制星下观测点各主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的潮汐调和常数, 然后根据所绘制图形可以发现潮汐调和常数沿着轨道方向的细结构, 并将沿轨道的各分潮的潮汐调和常数进行正压潮和内潮的分离。

问题二可以划分为两个小问题:

1. 绘制星下观测点四个主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的潮汐调和常数在轨道上的分布变化情况;
2. 沿轨道方向对各分潮的潮汐调和常数进行正压潮和内潮分离。

### 5.2 星下观测点各主要分潮潮汐调和常数沿轨道方向分布

#### 5.2.1 数据处理

##### (一) 水深资料数据处理

将题目中所给的水深资料数据, 利用 Matlab 进行处理, 将其存储为一个  $[277 \times 277]$  的矩阵, 元素在矩阵中的位置  $[m, n]$  对应着地图上从  $(99E, 2N)$  到  $(122E, 25N)$  之间以 5 分为分辨率的地图上的地理位置, 元素的数值对应代表着该点的水深深度。

将矩阵中全部元素的数值按照是否等于 NaN 进行分类, 划分为等于 NaN 和 不等于 NaN 的两大类。并依次为依据来区分陆地和海洋, 方便后续绘制陆地和海洋图。如下图 7 所示:



(图 7 根据水深数据绘制的陆地、海洋分布图)

## （二） 星下观测点求解数据处理

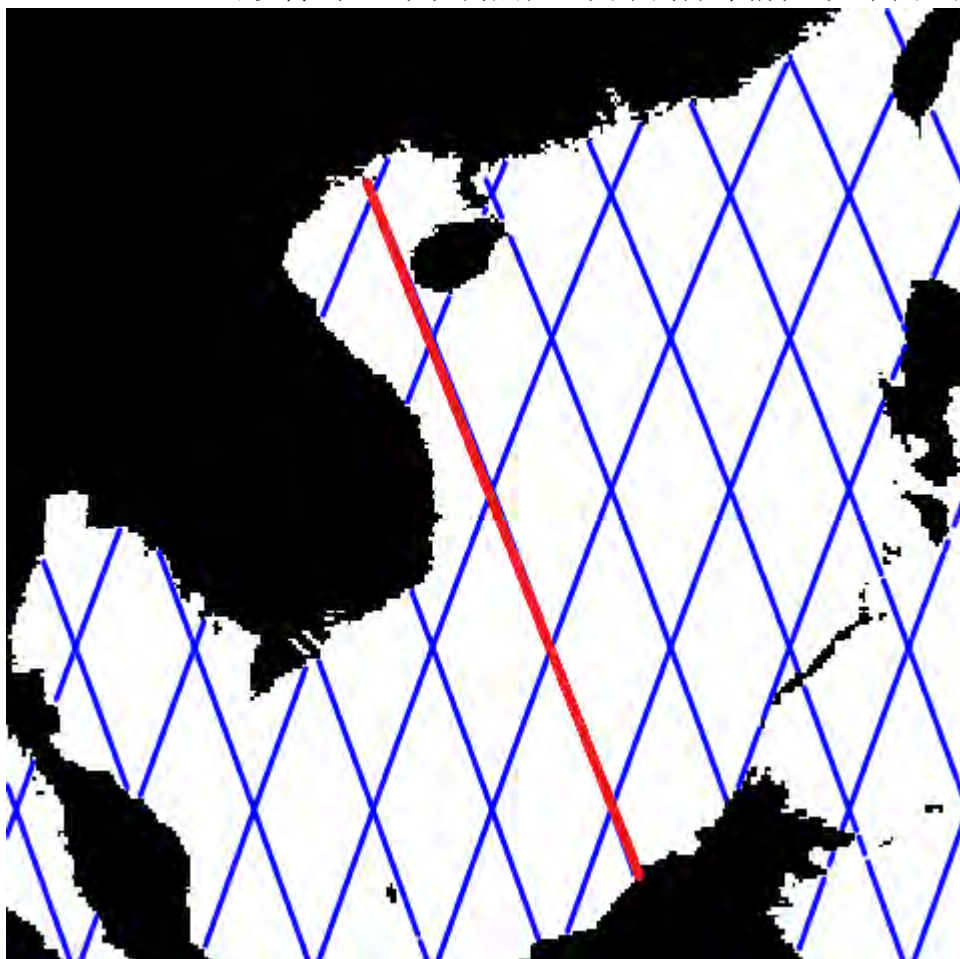
对 4.2.4 章节中按照 $[m*10]$ 的矩阵（ $m$  代表全体参与计算的星下观测点的个数），每一行的 13 个数据分别代表：星下观测点的经度、纬度、M2 分潮振幅、S2 分潮振幅、K1 分潮振幅、O1 分潮振幅、M2 分潮迟角、S2 分潮迟角、K1 分潮迟角和 O1 分潮迟角，格式进行存储的数据中的星下观测点的坐标数据进行处理，从而保证星下观测点坐标数据与上文中绘制的陆地、海洋分布图中的坐标处于同一量纲。

具体转化公式如下：

$$x_{map} = (x_{obs} - 99) * 12 + 1 \quad (5-1)$$

$$y_{map} = (25 - y_{obs}) * 12 + 1 \quad (5-2)$$

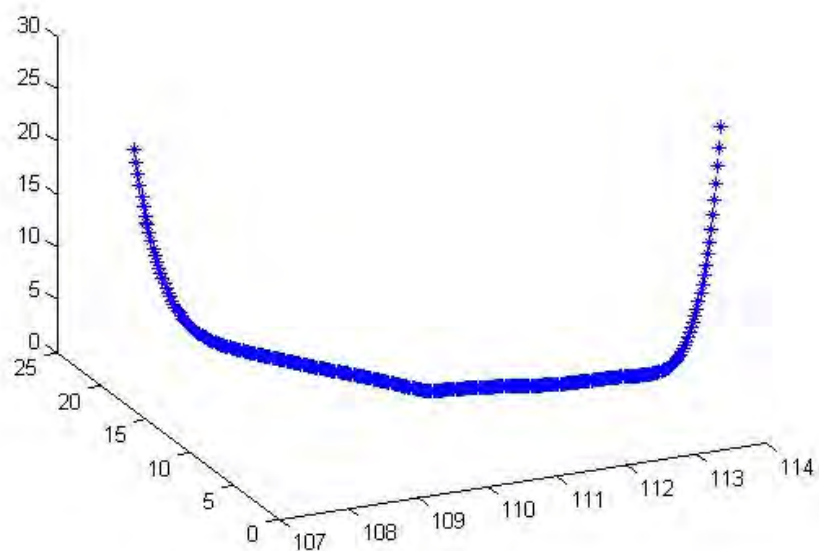
通过 MATLAB 可以得到，星下观测点在地图中的分布情况，如下图 8 所示：



（图 8 星下观测点在地图中的分布情况）

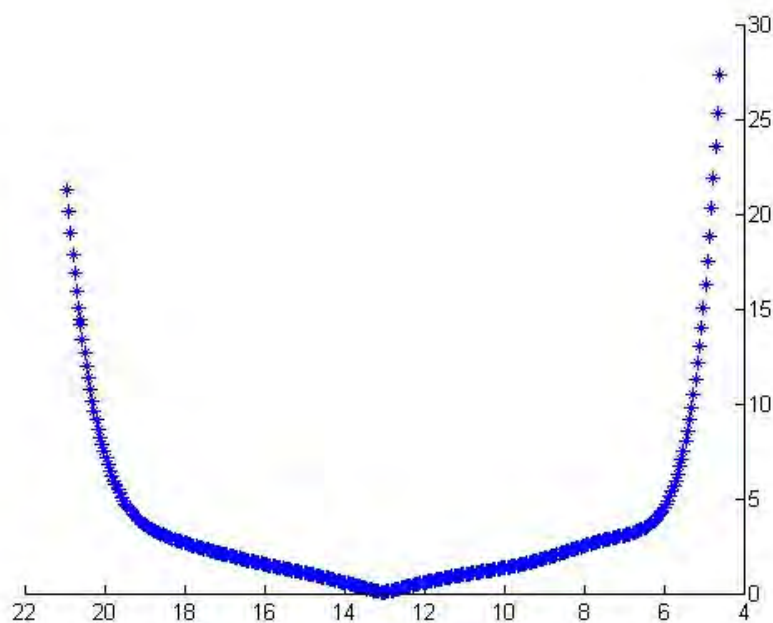
## （三） 潮汐调和常数延轨迹方向分布图的坐标处理

若直接通过三维图描述潮汐调和常数沿轨迹方向变化规律，则会如图 9 所示，不能较好直观地分析潮汐调和常数沿轨迹方向的变化分布规律。



（图 9 O1 分潮的振幅沿选定轨迹变化情况三维图）

为形象描述表达各个分潮的潮汐调和常数沿着轨迹方向的分布变化趋势，通过对比分析可知，轨迹在经度方向的变化率大于轨迹在纬度方向的变化率，为提高潮汐调和参数延轨迹方向变化的分辨率，保证结果准确性。我们选取轨迹变化率较小的纬度方向，作为研究四个主要分潮的潮汐调和常数沿轨迹变化坐标系的横坐标。如下图 10 所示：

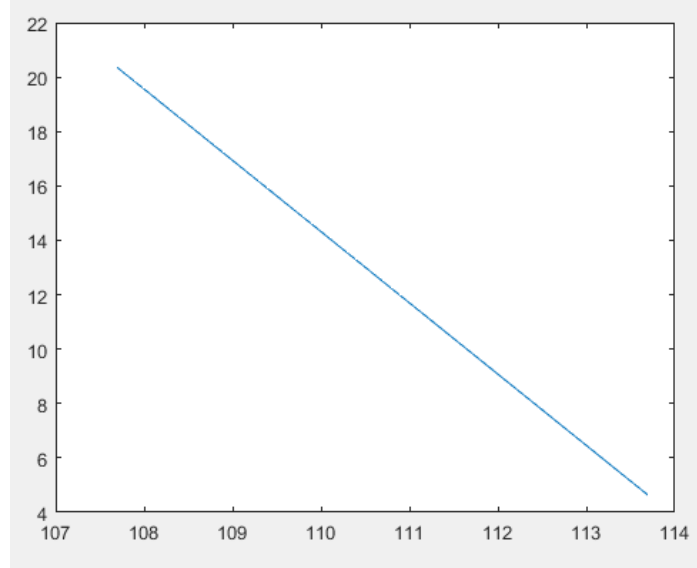


（图 10 O1 分潮的振幅沿选定轨迹变化情况的二维图）

### 5.2.2 模型准备

本文中以卫星运行轨道的一条下行轨迹为例进行分析,具体如图 8 中红色轨迹所示。通过提取位于该条轨迹上两点的经纬度坐标(107.6973,20.6777) 和 B[113.7104,5.0691], 确定轨迹所在方程为:

$$y = -2.62x + 302.52 \quad (5-3)$$



(图 11 轨迹方程图像)

### 5.2.3 模型建立与求解

在判断星下观测点是否位于卫星运行轨迹上时,忽略空间三维因素,将轨迹视为平面内的一条直线。同时,考虑到卫星观测的定位误差因素,将与轨迹相距在 0.001 范围内的观测点,视为处于卫星运行轨迹上。

平面内点到直线的距离公式为:

$$d = \left| \frac{Ax_0 + By_0 + C}{\sqrt{A^2 + B^2}} \right| \quad (5-4)$$

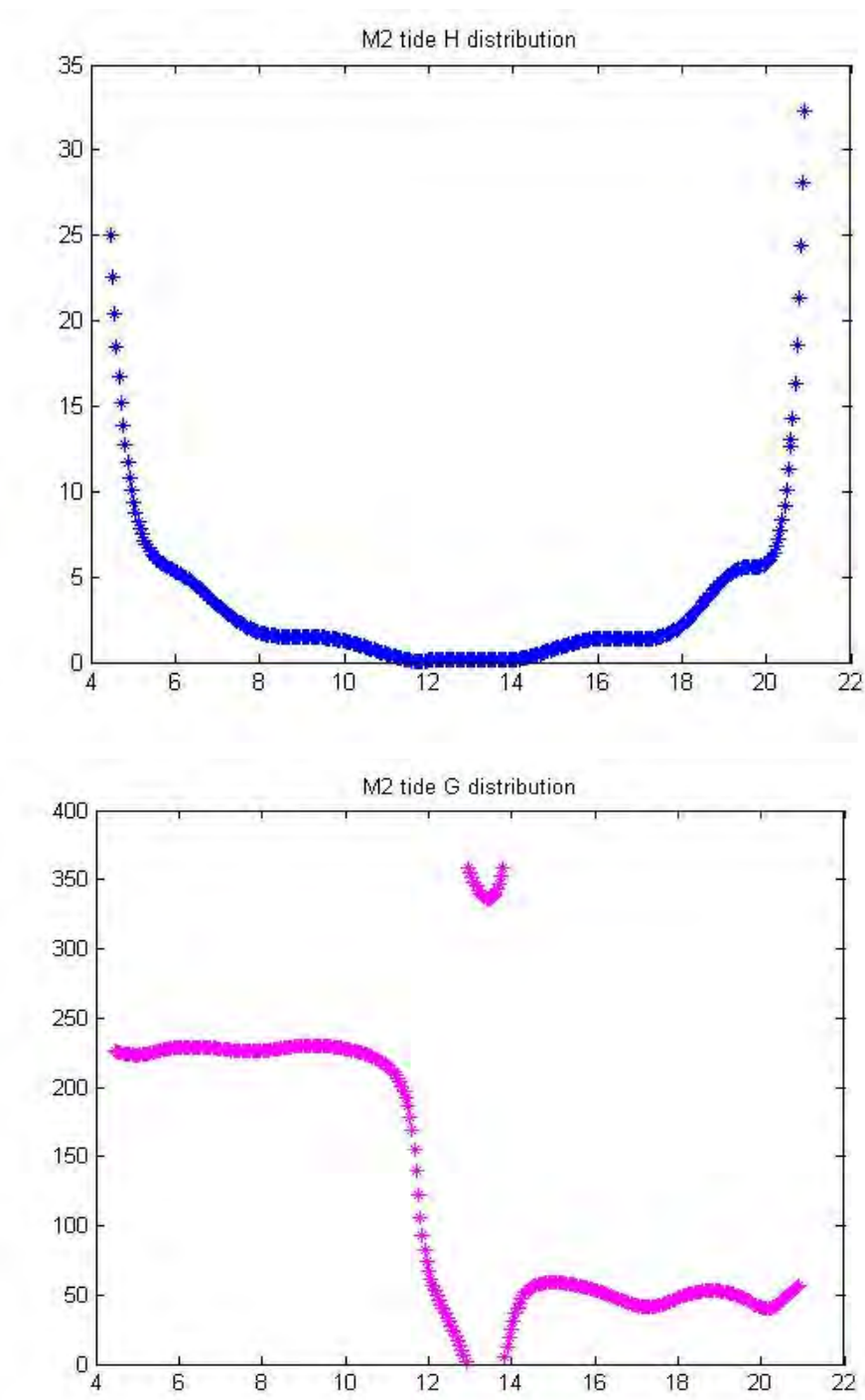
其中  $(x_0, y_0)$  为待检测的星下观测点的经纬度坐标。

利用 MATLAB 通过循环遍历算法,求解确定位于选定轨迹上的全部星下观测点。并将其经纬度坐标以及对应的四个主要分潮的潮汐调和常数存储入  $(k*10)$  的矩阵之中。

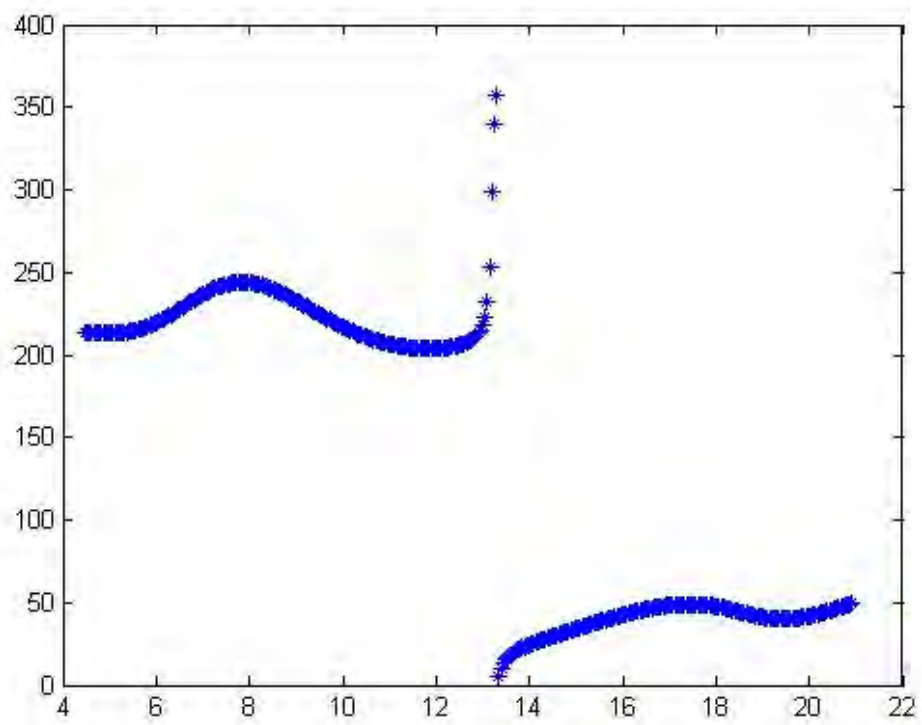
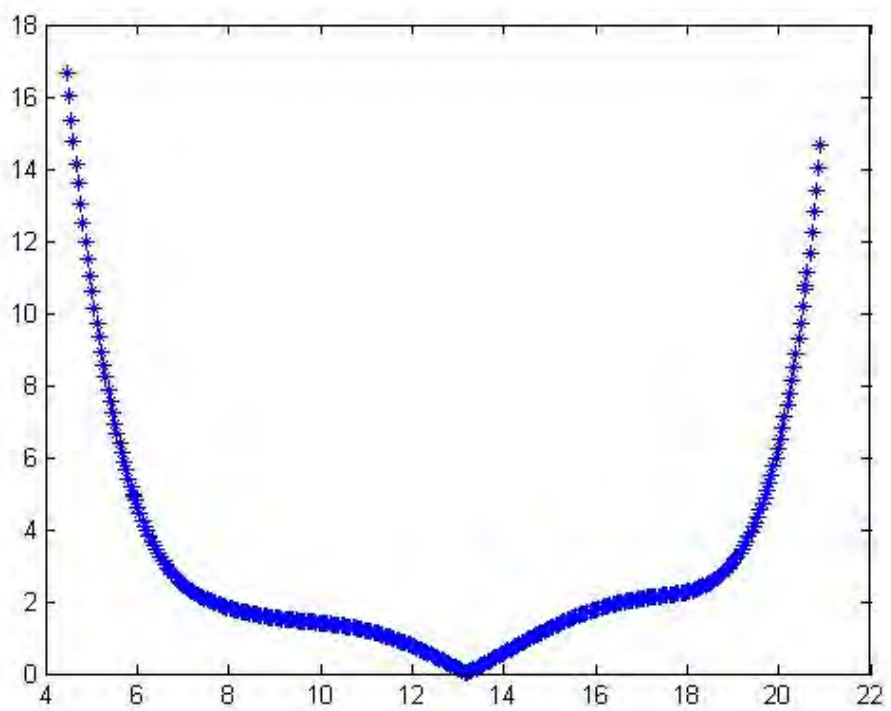
### 5.2.4 求解结果

利用 MATLAB 将上述计算结果,即选定轨迹上星下观测点的四个主要分潮  $(M_2, S_2, K_1, O_1)$  的潮汐调和常数分别绘制在以纬度为横坐标的二维坐标系

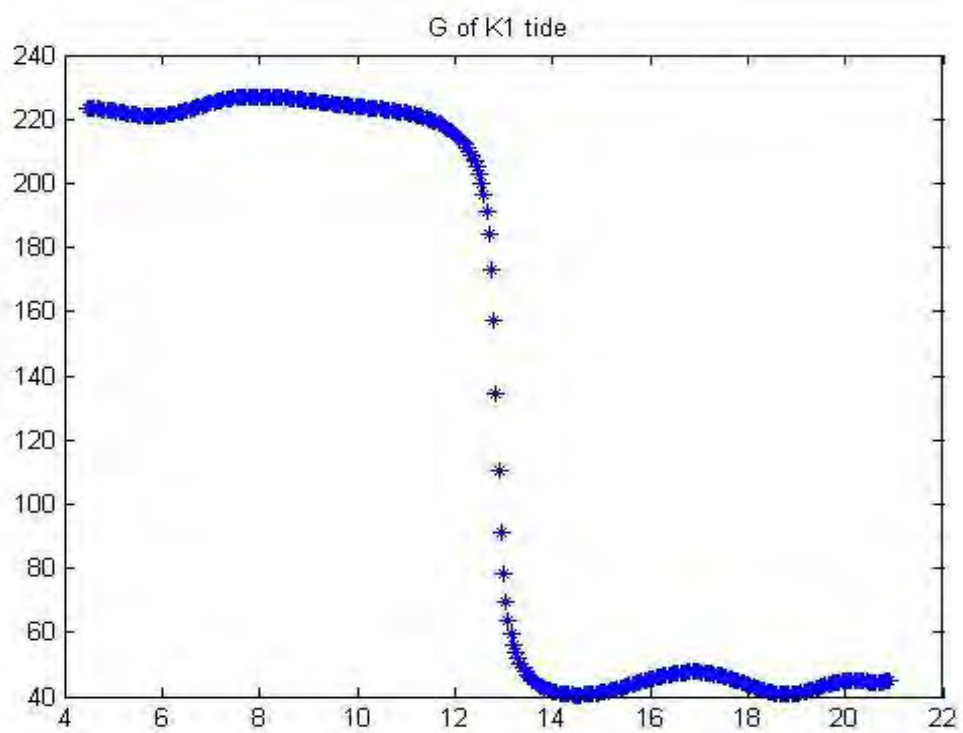
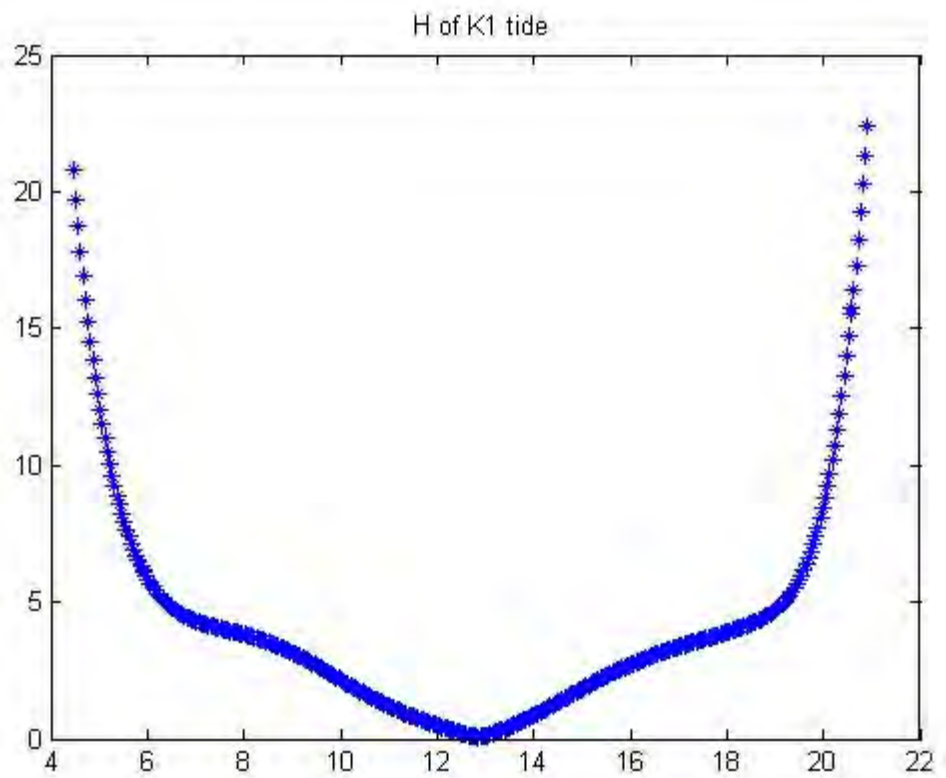
中，分别如下图 12、13、14、15 所示：



（图 12 M2 分潮振幅 H 与迟角 G 沿纬度分布图）

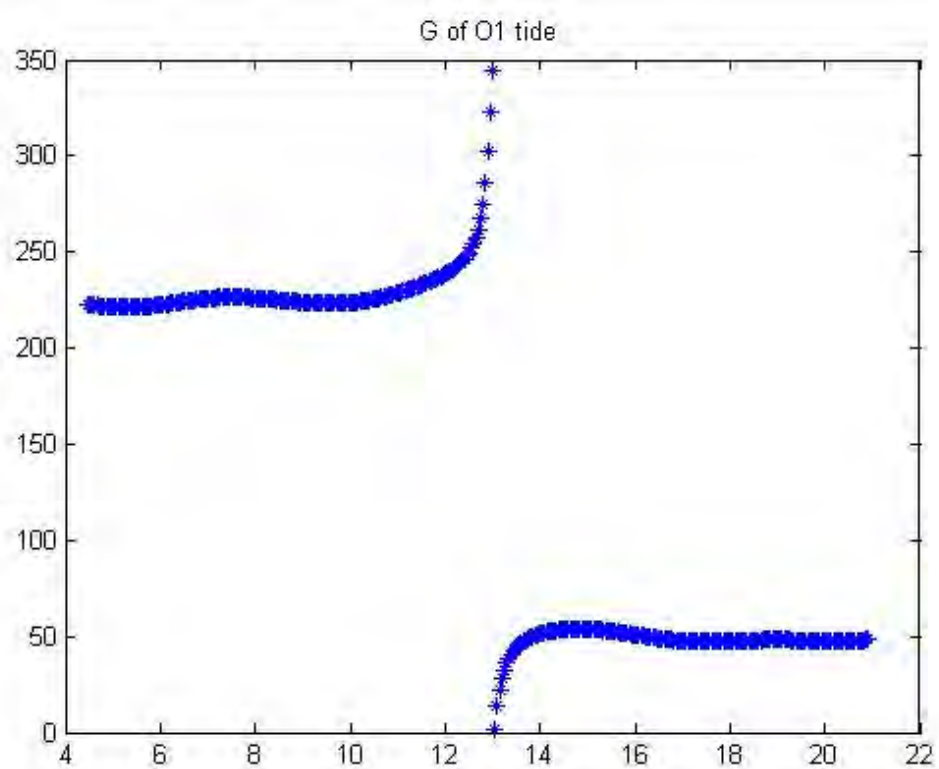
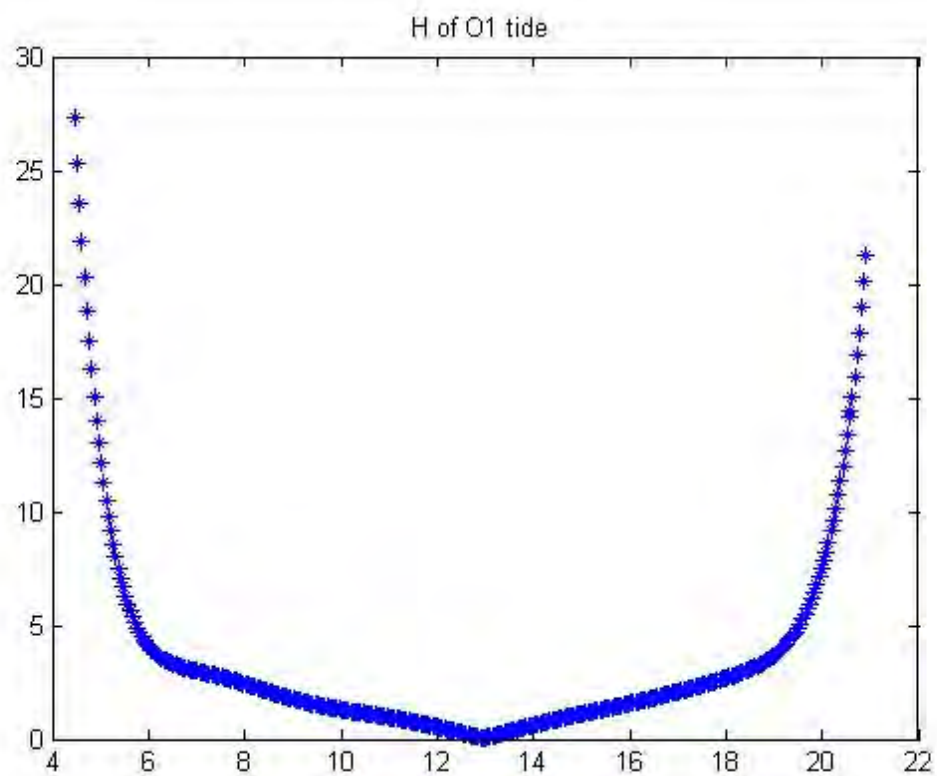


(图 13 S2 分潮振幅  $H$  与迟角  $G$  沿纬度分布图)



(图 14 K1 分潮振幅  $H$  与迟角  $G$  沿纬度分布图)





(图 15 O1 分潮振幅 H 与迟角 G 沿纬度分布图)



结合实际地理位置如图 16 所示，对上述计算结果进行分析可知：



(图 16 选定轨迹对应的实际地理图)

本文中所选定的卫星运行下行轨迹的两个端点分别靠近陆地（越南）与岛屿（文莱达鲁萨兰）。潮汐调和常数在沿纬度的函数分布中，高纬度与低纬度地区由于地处海洋与陆地交界处，海洋水流潮汐情况复杂，振幅 $H$ 表现出较大幅值。

中间纬度地区基本为海洋，海洋水流潮汐情况相较于海陆交界处较为稳定<sup>[6]</sup>，振幅 $H$ 幅度较小。

以上分析结果，与潮汐现象在沿海地区较为明显的生活常识正好吻合，同时也从侧面印证上述模型的准确性。

### 5.3 沿轨道各分潮的潮汐调和常数的正压潮和内潮的分离

#### 5.3.1 数据处理

##### （一）选定轨道上星下观测点的正压潮和内潮提取

正压潮又称为水面潮，内潮又称为斜压潮，以上两种分潮的不同划分是按照潮波引起的水体运动形式的不同而进行的划分。一般而言，海洋中正压流动代表水体垂向保持恒定的水平流动，同频率的内潮比正压潮的波长小的多。

查阅文献及题目中所给的相关参考资料可知，正压潮为潮汐调和常数中振幅  $H$  与迟角余弦值  $\cos G$  的乘积<sup>[7, 8]</sup>，内潮为潮汐调和常数中振幅  $H$  与迟角正弦值  $\sin G$  的乘积。

利用 MATLAB 将所选定的轨迹上的星下观测点的经度、纬度以及四个主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的正压潮和内潮存储入一个  $[n*10]$  的矩阵中 ( $n$  代表所选定轨迹上星下观测点的个数)，具体数据格式为每一行的 10 个数据分别代表：星下观测点的经度、纬度以及  $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$  的正压潮和内潮。部分数据如表 13 所示：

(表 13 部分轨迹上星下观测点的各分潮的正压潮及内潮数据)

	A	B	K	L	M	N	O	P	Q	R
1	经度	纬度	M2正压潮	S2正压潮	K1正压潮	O1正压潮	M2内潮	S2内潮	K1内潮	O1内潮
2	107.7213	20.5912	-5.880205992	6.983471378	4.706034345	1.643363813	-5.403530316	2.840137954	2.062175241	-1.791574231
3	110.2635	14.3434	1.057641992	0.529011503	-0.903266564	-1.065784196	-2.287156391	1.355402504	-2.677734399	2.367730324
4	110.2827	14.2937	0.715844107	0.685850834	1.80823242	-1.602102344	0.621134308	1.951930967	-3.238009568	0.240341504
5	110.302	14.244	-0.099577976	-0.674995842	1.707151891	-2.286989038	0.479132076	0.842826931	-1.614736471	-0.526646737
6	110.3212	14.1943	-0.831669203	-1.375736618	2.694434279	-0.5241455	-0.030775341	-0.414810286	0.338805967	-1.043386479
7	110.3404	14.1447	0.193873933	-0.761306466	1.92754686	-0.182888348	-1.01934688	-0.808291166	2.016746518	-1.287116
8	110.3597	14.095	0.748341217	0.474234396	0.891006013	0.193041679	0.22466588	-0.947716282	2.422482004	-1.330146598
9	110.3789	14.0453	0.205671095	1.758870773	-0.645981617	0.631845621	0.634267018	-0.67047033	2.488510971	-1.549507893
10	110.3981	13.9956	0.183588704	0.90904639	-1.645891257	2.207213308	0.295515069	1.262090326	1.460334985	-0.771569171
11	110.4173	13.9459	-0.489601114	-1.033954648	-2.042235658	1.712740835	-0.347707021	1.41994575	1.55013437	0.094253544
12	110.4365	13.8962	-0.044711544	-1.540354517	-2.372883192	1.245631746	0.212383989	-0.119103079	-0.246394212	1.236403376
13	110.4556	13.8465	-0.21058161	-0.712006292	-1.853017275	0.924716644	-0.358041634	-1.539325469	-1.119590245	1.946845018
14	110.4748	13.7968	0.03715483	0.750240748	-1.256649874	0.255681981	-0.005815691	-0.823993024	-2.563688066	2.161175121
15	110.4939	13.7471	-0.32644985	1.692099282	-0.065998463	-1.333948232	0.014683592	0.247093017	-2.839779404	2.299153158
16	110.5131	13.6974	-0.627180452	-0.124188661	0.565931292	-2.694353005	0.100747589	1.330025012	-1.71127927	1.629957198
17	110.5322	13.6476	-0.54011247	-1.05649802	2.493779749	-1.990232401	-0.542353305	0.880168851	-1.355290031	0.55151698
18	110.5513	13.5979	0.948985356	-1.233656909	2.779925264	-2.765545545	-0.395958682	-1.224972351	0.358847656	-1.241109759
19	110.5705	13.5482	-0.236297266	0.033072426	2.541642874	-1.570419987	-0.531275264	-1.247349056	1.947419297	-2.525982784
20	110.5896	13.4985	0.29261072	0.281793207	0.82587488	-0.474221324	-0.905316416	-0.581662311	2.049639333	-2.413903068
21	110.6087	13.4488	0.156159642	1.617646947	-0.737587528	0.94541717	0.197851524	0.416975615	3.109158332	-3.137856616
22	110.6277	13.399	0.269385609	-0.396976989	-1.116882492	2.172478794	0.19693917	0.999147107	1.527650736	-2.732331126
23	110.6468	13.3493	-0.934351195	-0.700914714	-2.860450493	2.732858757	1.093179979	0.870469648	1.103690867	-0.800926629
24	110.6659	13.2996	-1.156430747	-0.750798903	-3.024352503	2.767683852	-0.581185893	-0.304456404	-0.479839832	0.962431476

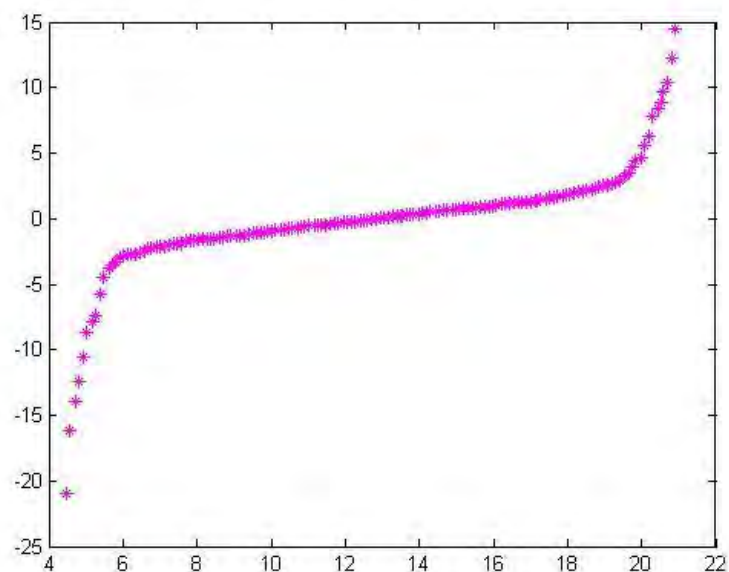
## (二) 提取轨迹上等间距 150 个观测点

同样利用 MATLAB 软件，提取上述选定卫星运行轨迹上的 150 个等间距的观测点，作为控制点用于后续正压潮、内潮沿轨迹方向变化函数的插补，数据的存储格式为  $[150 * 9]$  的矩阵中，每一行存储一个星下观测点的 9 个数据[纬度， $M2\_HcosG$ ,  $M2\_HsinG$ ,  $S2\_HcosG$ ,  $S2\_HcosGK1HcosG$ ， $K1HsinG$ ,  $O1\_HcosG$ ,  $O1\_HsinG$ ]，分别代表：星下观测点的纬度，四个分潮的正压潮、内潮。

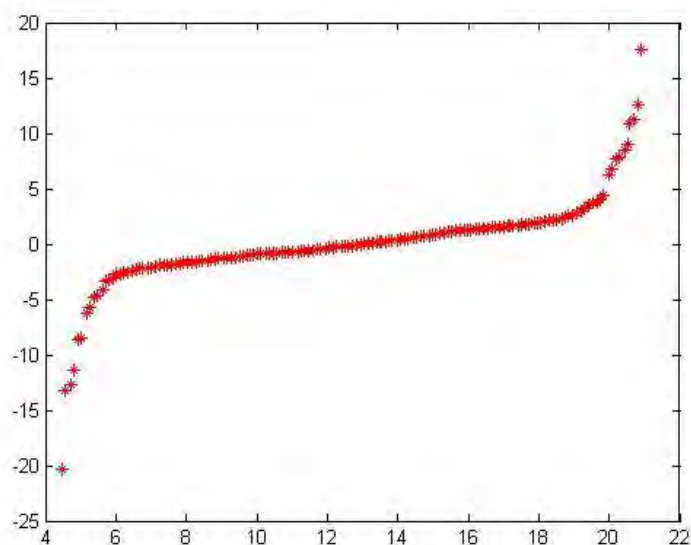
### 5.3.2 模型准备

以 5.3.1 数据处理阶段准备的 150 个等间距的选定轨迹上的星下观测点的  $O1$  分潮的正压潮和内潮为例，将其绘制在以纬度为横坐标的图表上，如图 17、18

所示：



(图 17 选定轨迹上 O1 分潮的正压潮随轨迹分布图 )



(图 18 选定轨迹上 O1 分潮的内潮随轨迹分布图)

### 5.3.3 模型建立与求解

经查阅相关文献，了解到采用高阶多项式轨线插值法对正压潮 $H \cos G$ 和内潮 $H \sin G$ 进行处理得出的插值结果要优于直接运用振幅 $H$ 和迟角 $G$ 插值得到的结果。本文中采用 12 阶次的多项式，对卫星运行轨迹上的  $H \cos G$ ， $H \sin G$  进行插值。最后通过坐标变换，反推出振幅  $H$  与迟角  $G$ 。

现将高阶多项式轨线插值原理介绍如下：

高次多项式定义为如下所示公式：

$$\hat{f}(y) = a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n, \quad (5-5)$$

其中  $y$  表示 TP 轨迹上任意一点的纬度。代价函数可以写为：

$$J(a_n) = \sum_{k=1}^N [(a_0 + a_1 y_k + a_2 y_k^2 + \cdots + a_n y_k^n) - H_k \cos G_k]^2, \quad (5-6)$$

其中  $y_k$  表示 TP 轨迹上观测点的纬度。通过是代价函数对每个多项式系数的偏微分为 0，可以得到：

$$\begin{cases} \sum_{k=1}^N [(a_0 + a_1 y_k + a_2 y_k^2 + \cdots + a_n y_k^n) - H_k \cos G_k] = 0 \\ \sum_{k=1}^N y_k [(a_0 + a_1 y_k + a_2 y_k^2 + \cdots + a_n y_k^n) - H_k \cos G_k] = 0 \\ \sum_{k=1}^N y_k^2 [(a_0 + a_1 y_k + a_2 y_k^2 + \cdots + a_n y_k^n) - H_k \cos G_k] = 0 \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ \sum_{k=1}^N y_k^n [(a_0 + a_1 y_k + a_2 y_k^2 + \cdots + a_n y_k^n) - H_k \cos G_k] = 0. \end{cases} \quad (5-7)$$

本文中引入如下标注

$$M_{i,j} = \sum_{k=1}^N y_k^{i+j-2}, \quad b_i = \sum_{k=1}^N y_k^{i-1} H_k \cos G_k, \\ i = 1, 2, \dots, n+1, j = 1, 2, \dots, n+1, \quad (5-8)$$

那么方程可以重新写为：

$$\begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} & \cdots & M_{1,n+1} \\ M_{2,1} & M_{2,2} & M_{2,3} & \cdots & M_{2,n+1} \\ M_{3,1} & M_{3,2} & M_{3,3} & \cdots & M_{3,n+1} \\ \vdots & \cdots & \cdots & \cdots & \cdots \\ M_{n+1,1} & M_{n+1,2} & M_{n+1,3} & \cdots & M_{n+1,n+1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \cdots \\ a_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_{n+1} \end{bmatrix}. \quad (5-9)$$

通过求解上述线性方程组即可求解多项式系数，于是正压潮系数可以解出：

$$\hat{f}(y) = a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n. \quad (5-10)$$

同理，对  $H \sin G$  进行拟合可以求出内潮系数：

$$\hat{g}(y) = a_0 + a_1 y + a_2 y^2 + \cdots + a_n y^n, \quad (5-11)$$

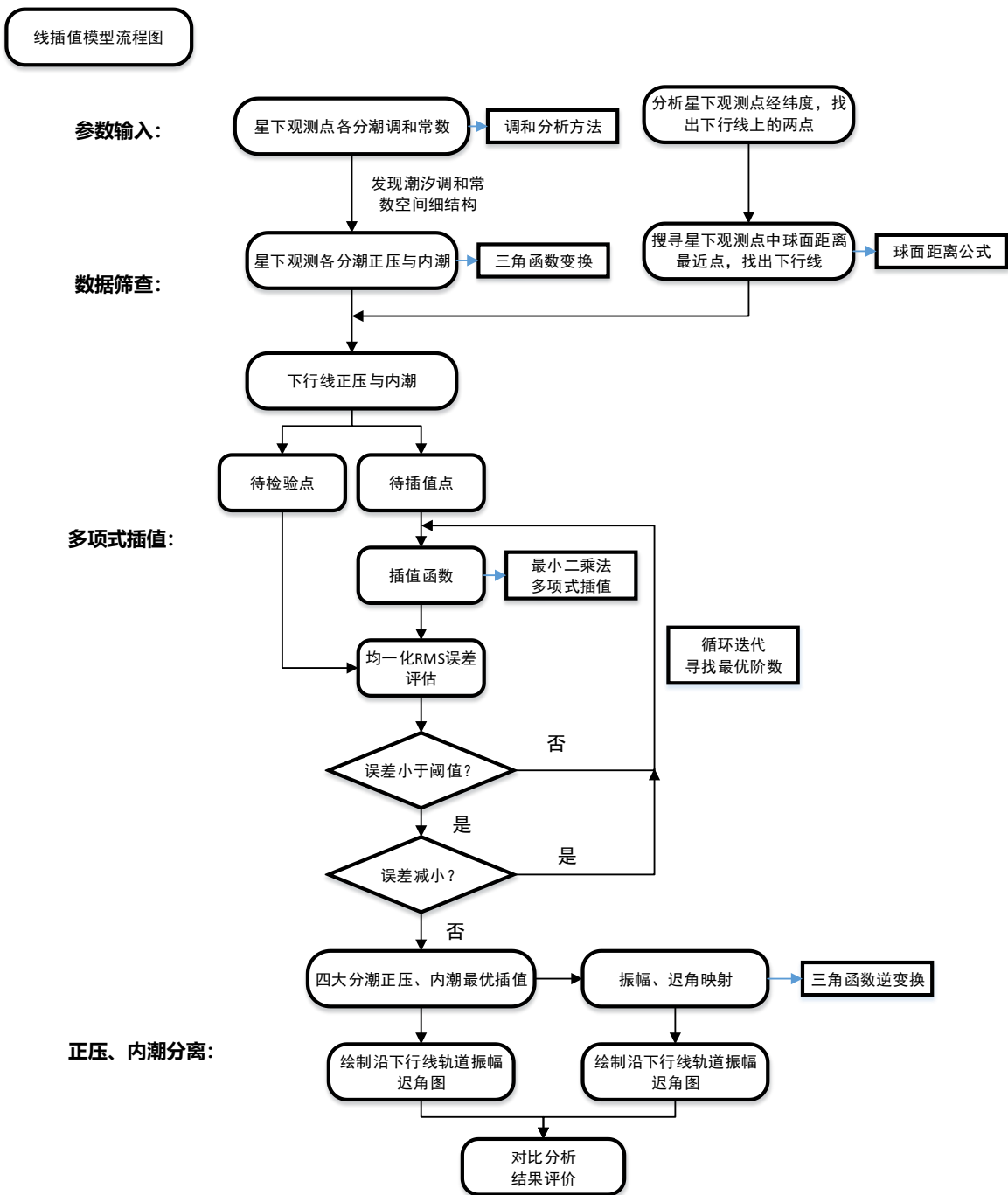
在数值分析中，高阶多项式插值是一种常用插值算法，即利用存在某种内在联系或规律的数据点作为控制点，以多项式作为插值函数，拟合出多项式函数，使得该函数通过所有的控制点。由于不同次数的多项式可以组成一组基，理论上，任意高阶多项式的线性组合可以表达所有的连续函数，即若控制点的内在规律满

足连续的性质，皆可使用高阶多项式插值算法进行插值。在本例中，海洋潮汐分潮的正压和内潮在物理空间上明显满足连续函数的性质，理论论证可以使用高阶多项式插值求解轨迹线上的其他正压和内潮<sup>[9]</sup>。

高阶多项式插值存在多种形式的插值函数，如拉格朗日插值函数、牛顿插值函数等，而插值函数在本质上并无区别，其核心思想为插值函数的误差最小，即最小二乘法，针对本问题的具体算法如下。

首先，根据前述的调和分析方法，通过一系列的数据转换与方程组求解，得到所有星下观测点四大主要分潮的调和常数，以此作为高阶多项式轨线插值方法的输入参数。根据相关文献与本例计算得到的数据，经过分析和讨论可以发现，潮汐调和常数关于空间的分布存在某些细结构，使用正压与内潮的方法表示分潮更加合理，因此，通过三角函数变换，将插值目标改为行下观测点的各分潮正压与内潮。测量过程中，卫星按照上行线和下行线的规律进行海洋空间点的测量，因此插值对象应保持统一在上行或下行线上，考虑到地图中包含的海洋区域较广，地球的球面效应较为显著，我们使用了球面距离公式对所有星下观测点进行判别，设置最小距离阈值，找出一条下行线上的所有星下观测点。接下来，将该下行线的测量点分为待插值点和待检验点两类，使用待插值点应用高阶多项式插值算法，使用待检测点对该插值函数进行 RMS 误差评估，设置误差阈值对插值函数进行评估。由于不同分潮的正压和内潮范围有差别，为了采用统一的误差阈值评价参数对不同插值函数进行评估，对插值后的 RMS 进行了均一化处理。建立以统一误差阈值为评价标准的体系后，通过最小二乘算法对星下观测点下行线进行高阶多项式插值，得到该下行线上所有点的正压和内潮值，再通过三角函数逆变换得到其分潮的振幅与迟角<sup>[10]</sup>，最后绘制轨线上的振幅迟角图，对最终结果进行分析评价。

高阶多项式轨线插值算法流程图如下图 19 所示：

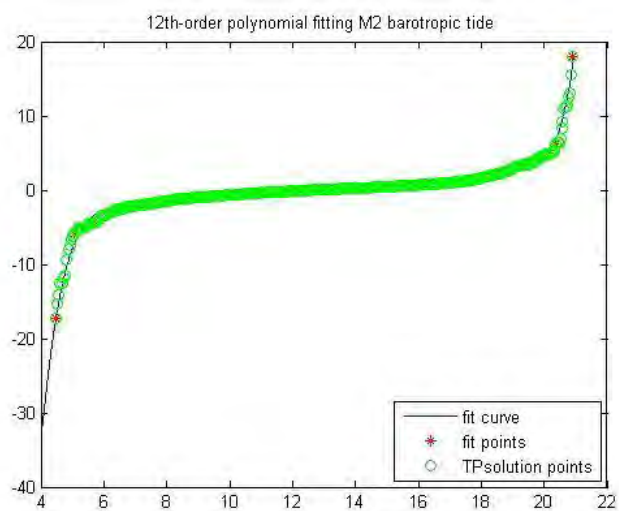


（图 19 高阶多项式轨线插值算法流程图）

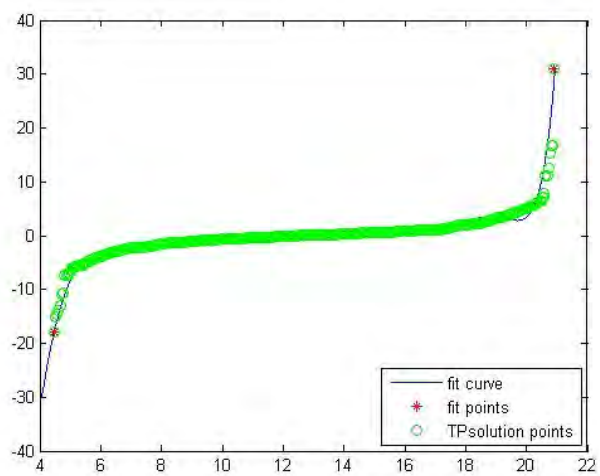
### 5.3.4 求解结果

选定轨迹上，星下观测点四个主要分潮的正压潮和内潮通过 12 次多项式插值计算后，其沿着轨迹方向的变化分布图如下图所示。

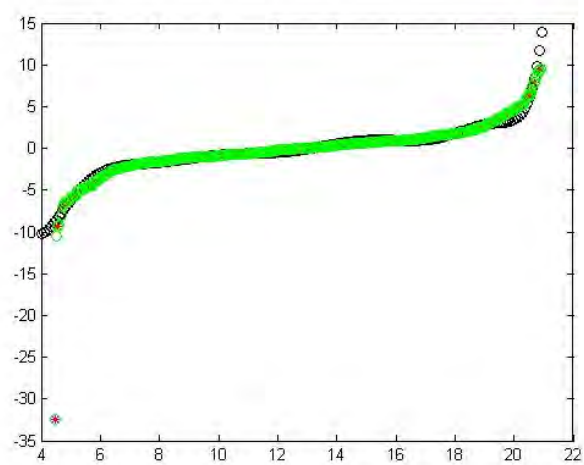




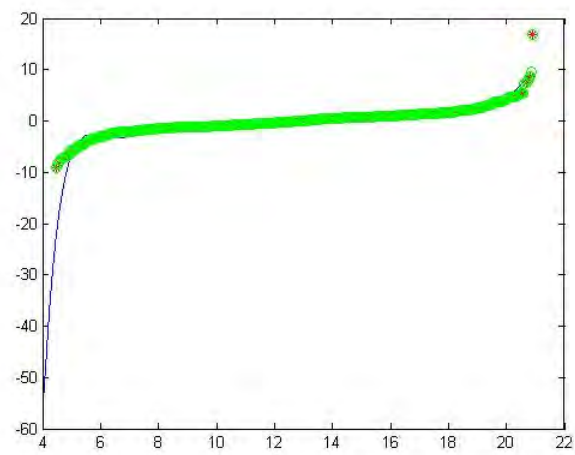
(图 20 M2 正压潮沿轨迹分布图)



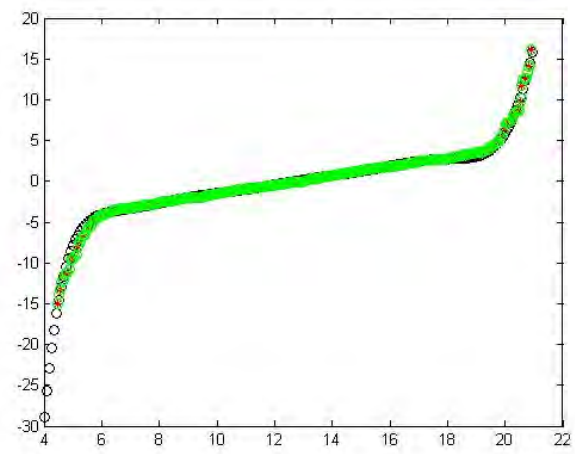
(图 21 M2 内潮沿轨迹分布图 )



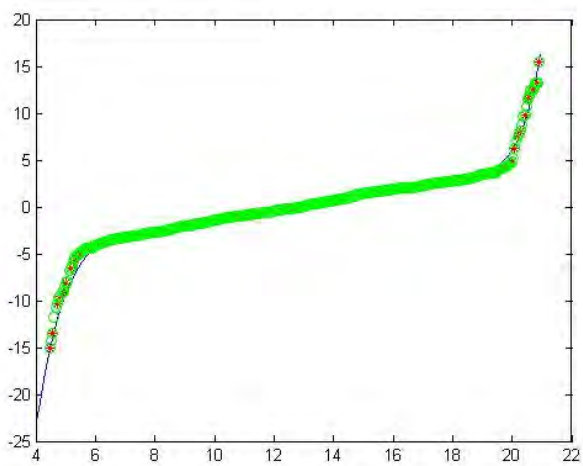
(图 22 S2 正压潮沿轨迹分布图)



(图 23 S2 内潮沿轨迹分布图)

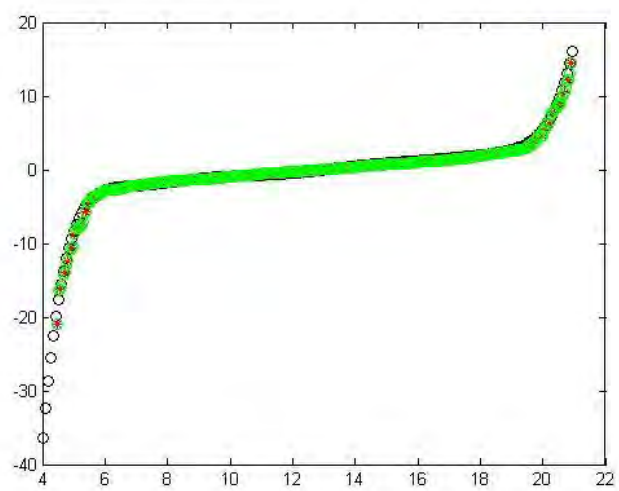


(图 24 K1 正压潮沿轨迹分布图)

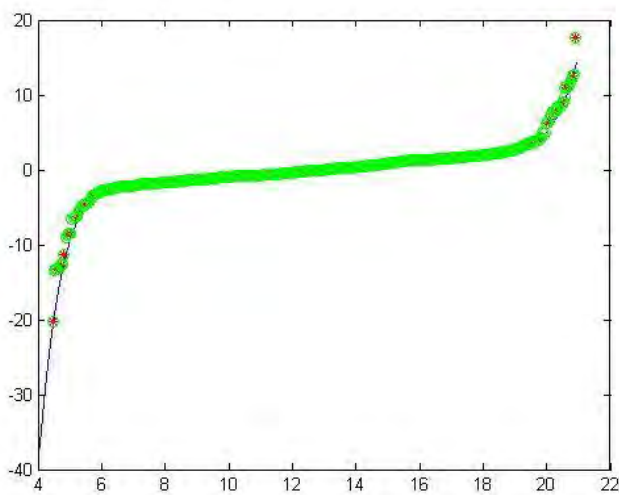


(图 25 K1 内潮沿轨迹分布图)





(图 26 O1 正压潮随轨迹分布图)



(图 27 O1 内潮随轨迹分布图)

## 6. 问题三 插值求出各分潮的同潮图，并利用验潮站检验

### 6.1 问题分析

问题三中要求采用数据插值或者拟合的方法绘制出南海四个主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的同潮图，同时利用已知的验潮站的数据对所求得同潮图进行评价，并对评价结果进行分析。

问题三可以划分为两个小问题：

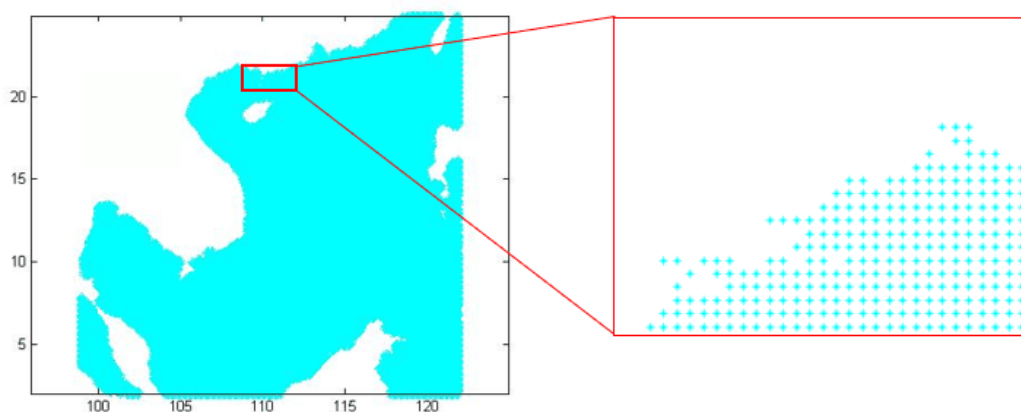
1. 采用高阶多项式轨迹插值的方法绘制南海主要分潮的同潮图；
2. 利用验潮站数据对同潮图进行评价，并对评价结果进行分析。

### 6.2 采用高阶多项式轨迹插值法绘制南海主要分潮的同潮图

#### 6.2.1 数据处理

##### （一）提取南海海洋交叉点坐标

根据题目中所给出的水深数据，提取地图上以（99E,2N）、（122E,25N）两点为顶点的矩形范围内除陆地之外的分辨率为 5 分的海洋网格交叉点，如图 28 所示，并将所得的海洋网格交叉点的经度、纬度存入一个 $[2*n]$ 的矩阵之中（ $n$ 为海洋交叉点的个数），部分数据如表 14 所示。



（图 28 南海海洋网格交叉点）

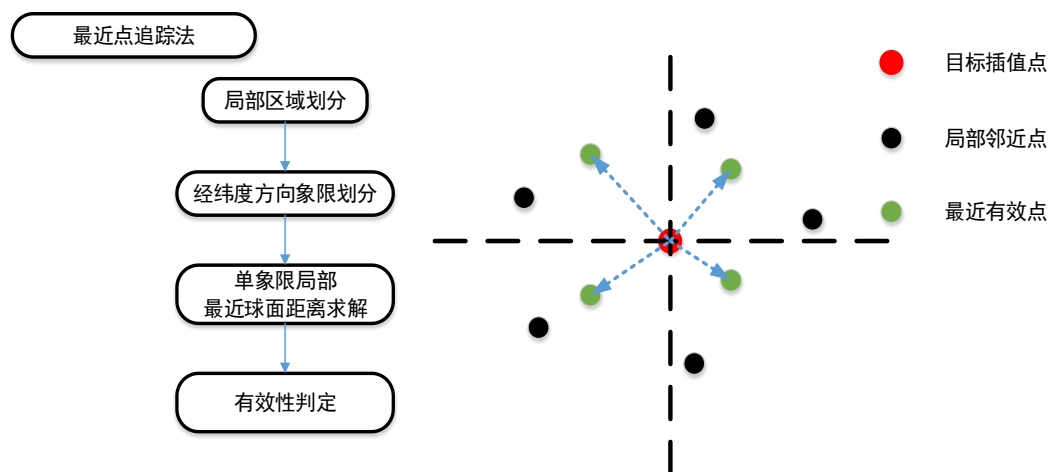
（表 14 南海海洋交叉点坐标信息）

纬度	2	2	2	2	2	2	2	2	2	2	2	
经度	101.324	101.739	101.822	101.905	101.988	102.071	102.154	102.237	102.32	102.403	102.486	104.063

## （二）数据清洗：去除无效点

在第六章节内，本文将沿着网格线的四个方向上不存在完整相邻点的网格交叉点称之为无效点。去除无效点为后续进行高阶次多项式轨迹插值排除噪点，确保所得结果的准确性。

本文中提出了一种去除无效点进行数据清洗的方法——最近点追踪法。逻辑流程图如图 29 所示：



（图 29 最近点追踪法逻辑流程图）

同潮图能直观的反映海洋中潮水的流动规律，在海洋洋流分析中有着至关重要的作用，绘制分潮的同潮图需要已知海洋中所有点潮水振幅与迟角，而由于实际情况限制，不可能将所有点的相关参数测量出来，需要通过面内插值的方法，将未测量点的潮水振幅与迟角计算出来。但由于实际情况的复杂性，往往会存在测量数据不完整、测量区域不规整等引起的难以插值的现象，针对这种现象，需要设计一种算法筛除掉这些坏点<sup>[7, 11]</sup>。

对于某一具体的待插值点  $P$ ，其影响插值可行性可能出现的情况大体可以分为四种，如图 30 所示。

（1）目标插值点某一象限无可插值节点：针对平面插值方法，通过四个象限的局部邻近点进行插值是较好的选择，若某一象限无可插值节点，使用其他邻近点进行插值可能会产生较大的误差，如图 30（a）中所示，需要将这些点筛除掉。

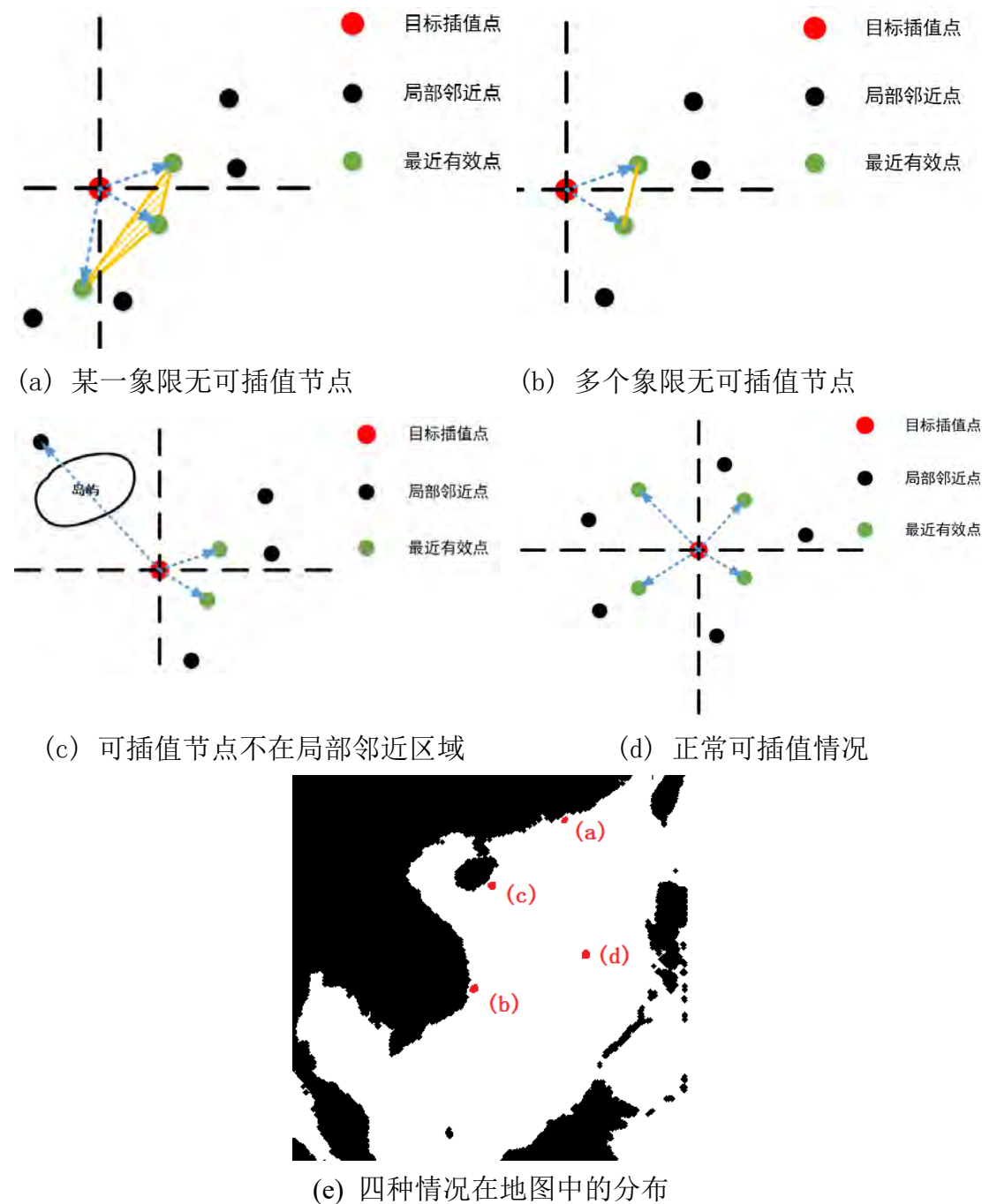
（2）目标插值点多个象限无可插值节点：与情况（1）类似，多个象限无可插值节点，使用剩余邻近点插值会产生更严重的误差，如图 30（b）中所示，需要将这些点筛除掉。

（3）目标插值点四个象限内都有可插值节点，但不分布在邻近区域：如图 30（c）所示，若目标插值点在岛屿附近，岛屿侧可能存在可插值节点，但由于这些节点分布在岛屿对侧，目标插值点与其物理上的联系关系不大，若仍然使用岛屿对侧节点插值，必然造成结果失真，需要将这种情况排除。

（4）目标插值点在四个象限全部存在局部邻近点：如图 30（d）所示，目标插值点与四个象限内的局部邻近点在物理上存在较为紧密的联系，并且邻近点连线构成的四边形将目标插值点包容在内，符合平面插值的基本思路，这种情况可以

使用面内插值算法。

基于上述分析，我们可以发现，在进行插值算法前，需要对目标插值点四个象限内是否存在局部邻近点准则进行判别，筛除掉不符合要求的情况，这样才能得到较为准确的插值结果。



(图 30 目标插值点的不同情况)

针对这种局部邻近点分布的点集筛选，我们设计了一种最近点追踪法来进行筛选，并找到有效点的邻近插值节点。其具体步骤如下。1. 局部区域划分：设置局部范围半径阈值，找出目标插值点局部区域内的点集。2. 经纬度方向象限划分：

按照经纬度的方向，将局部区域内的点集按照坐标系象限的方式进行划分。3. 单象限局部最近球面距离求解：分别对上述划分的四个象限进行最近点求解，采用球面距离的方法求解更加精确，球面距离的具体计算方法在后面会详细给出。4. 有效性判定：经过上述 3 个步骤后，我们可以区分出目标插值点属于 (a), (b), (c), (d) 中的何种情况，从而对目标插值点进行筛选。经过最近点追踪法计算后，我们不仅能筛选有效的目标插值点，还能将其局部邻近点找到，便于后续插值<sup>[12]</sup>。

运用最近点追踪法筛选数据时，需要理由球面距离计算两点的真实距离，从而进行局部性与最近判断。由于球面几何与实际平面几何有所区别，我们查阅了相关文献，得到了球面距离的计算公式。

如图 31 所示，已知球面上两点经纬度分别为 $(a_1, b_1)$ ,  $(a_2, b_2)$ ，两点之间的球面距离如下：

$$L = R * \arccos(\sin(a_2) \sin(b_2) + \cos(a_2) \cos(b_2) \cos(a_1 - b_1)) \quad (6-1)$$

确定两点之间的距离公式后，还需计算球面上点到线的距离。以  $p$  到  $p_1 p_2$  为例。

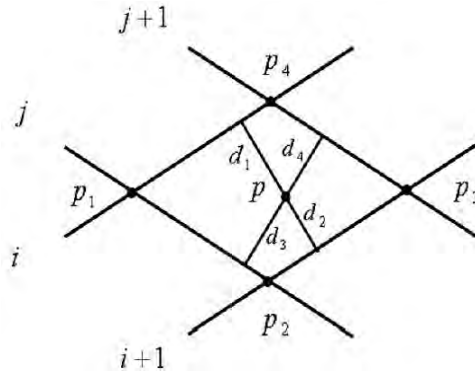
首先，计算三角形  $\Delta p p_1 p_4$  面积，面积公式如下：

$$s = \sqrt{l(l - p p_1)(l - p p_4)(l - p_1 p_4)} \quad (6-2)$$

其中  $l$  为三角形半周长。

接下来，计算距离：

$$d_1 = 2\sqrt{l(l - p p_1)(l - p p_4)(l - p_1 p_4)} / p_1 p_4 \quad (6-3)$$



(图 31 球面距离示意图)

## 6.2.2 模型准备

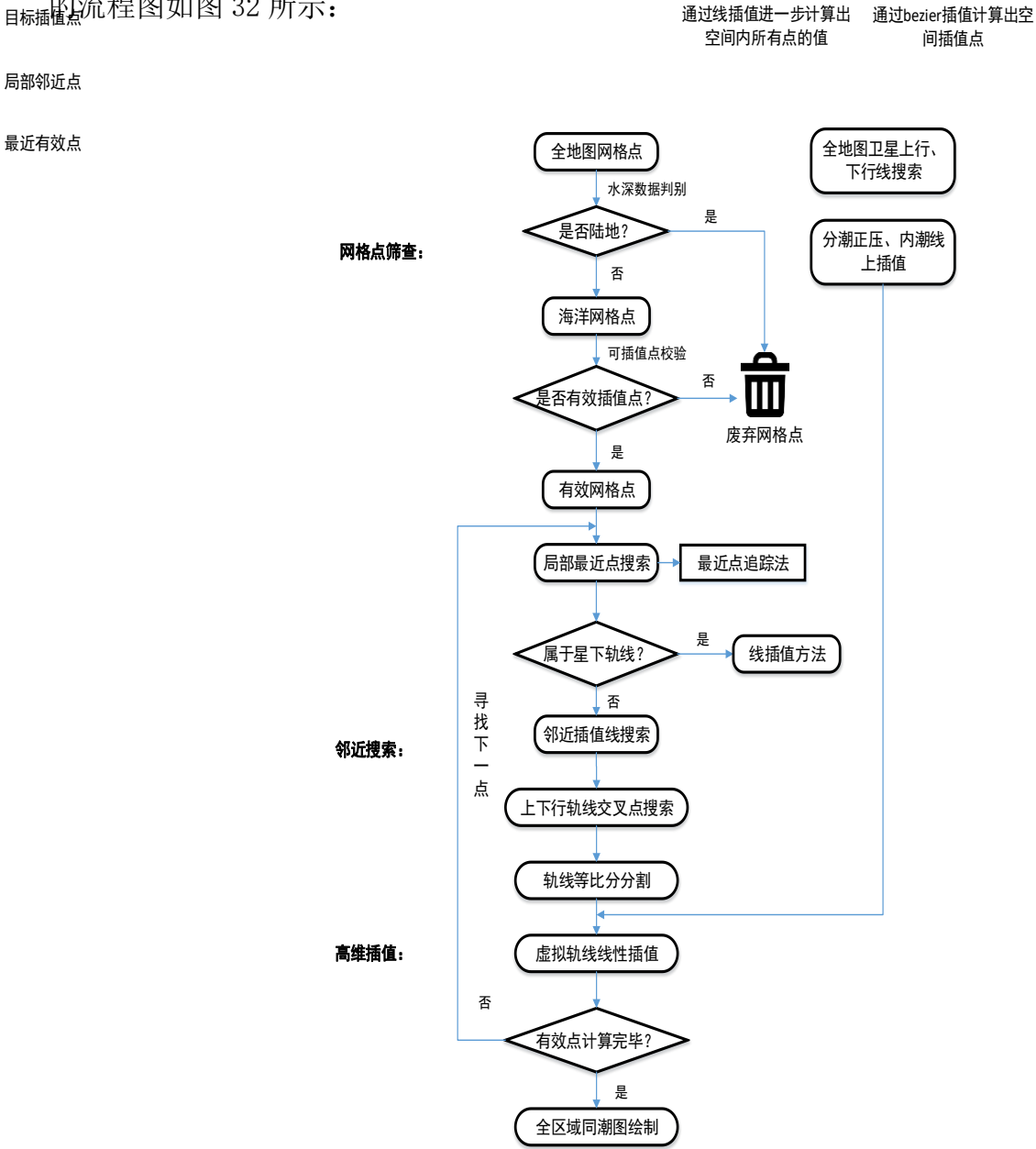
由第五章节可知得，通过高次多项式轨线插值法可以求出地图上全部的卫星轨迹上行线和下行线上所分布点的四个主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 调和常数的插值函数。

通过最近点跟踪法判断该点是否位于轨迹上，当该点位于轨迹线上时，采用第五章节中的高阶多项式轨线插值法进行处理求解，相关公式见 5.3.3；当该点

不在轨迹上时，通过有效网格点邻近搜索高维插值法进行求解。

6.2.3 模型建立与求解

有效网格点邻近搜索高维插值法，实质上，同潮图是将海洋平面上所有点的振幅或迟角绘制在地图上，经过问题一的求解，我们获得了星下观测点的潮汐调和常数，经过问题二的求解，我们获得了卫星上行线与下行线的振幅与迟角<sup>[13]</sup>，而为了获得整个海洋平面内所有点的潮汐调和常数，需要对海洋平面内所有点进行插值，为此，我们设计了一种名为“有效网格点邻近搜索高维插值”算法，将地图中其余点的潮汐调和常数进行插值，从而绘制出海洋平面的同潮图。该算法



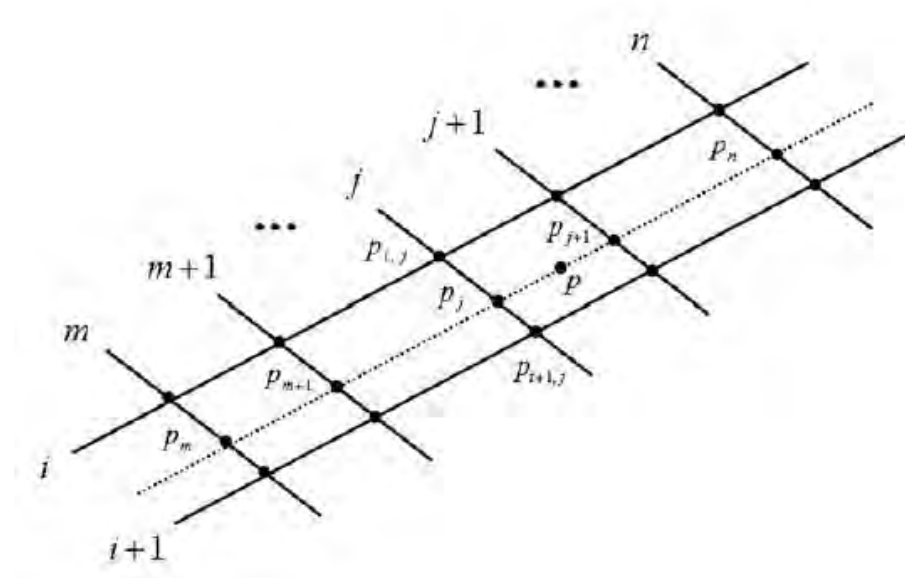
(图 32 有效网格点邻近搜索高维插值法逻辑流程图 )

首先，利用资料中所给的深度计数据，我们获得了全图的经纬度范围，根据最大范围设计的原则，本文将经度  $2\sim 25^\circ$ ，纬度  $99\sim 122^\circ$  范围内所有点按照  $5'$  的分辨率进行网格点划分。然后通过是否陆地、是否有效插值点的一系列判断，将网格点中废弃的点舍去，从而得到了海洋平面内的有效网格点。与此同时，利用第二问的算法，将所有的星下轨迹线通过高阶多项式轨线插值的方法求出，便于后续再次插值使用。

对于有效网格点，本文采用了逐点插值的方法分别求解，对每一个有效的网格点分别进行以下操作。首先使用最近点追踪法进行局部最近点搜索，得到四个象限内的局部邻近点。接下来，我们对目标插值点是否属于星下轨线进行判断，使用了球面距离，求解该点到上下行线的距离，若小于设置的距离阈值，则认为该点属于星下轨线，类似第二问使用线插值的方法对该点进行插值，否则认为该点属于空间中的任意点，使用高维的插值方法<sup>[14]</sup>。

高纬度的插值算法利用了等比分的性质，寻找附近的星下轨迹线插值函数，对目标插值点函数值进行估计，具体实施如下。搜索目标插值点邻近的插值线，找到该点附近的两条上行轨迹线，如图 33 所示。找到上行轨迹线与其他下行轨迹线的交点，根据点  $p$  到最近两个上行轨迹距离的比值，在  $n + m - 1$  下行轨迹线上，可以获得拥有相同距离比值的一系列点  $(p_m, p_{m+1}, \dots, p_j, p_{j+1}, \dots, p_n)$ 。

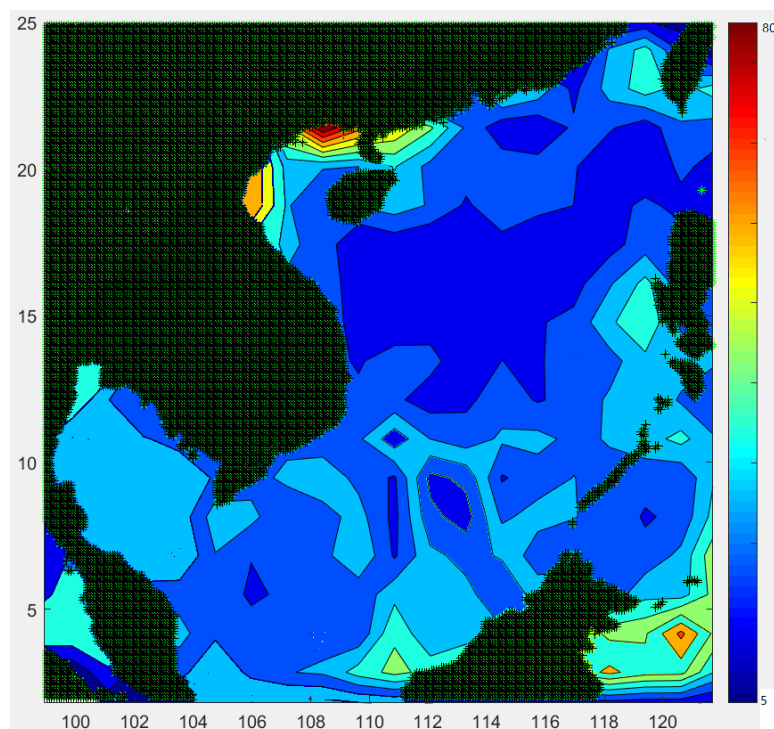
基于第二问的高次多项式插值方法，轨迹上的  $\hat{f}(y)$  与  $\hat{g}(y)$  可以通过多项式插值得到，我们将此不在星下观测点的轨迹线称之为虚拟轨线。该虚拟轨线上的  $(p_m, p_{m+1}, \dots, p_j, p_{j+1}, \dots, p_n)$  上的正压与内潮可以通过再次插值得。通过这种方式不断递推求解，计算出所有有效网格点的正压与内潮值，从而绘制出个各分潮下的同潮图。



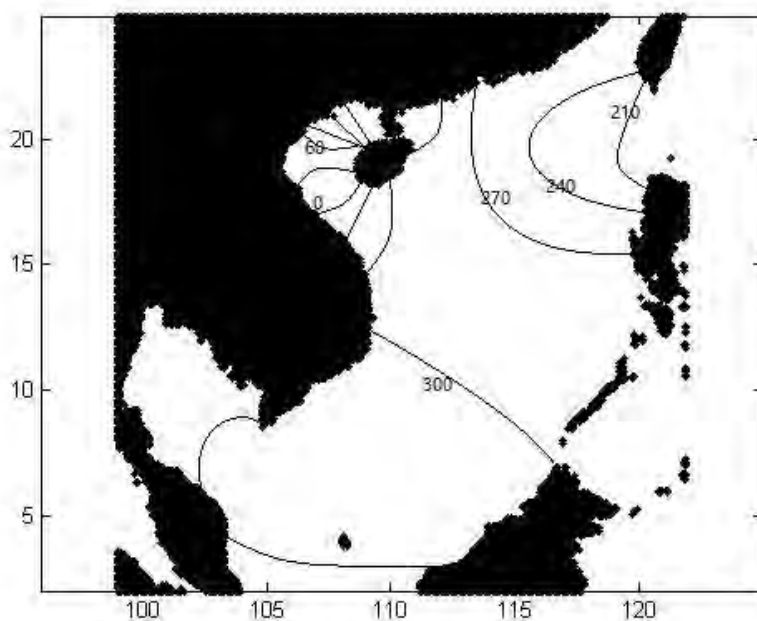
(图 33 等比分上轨线插值示意图)

#### 6.2.4 求解结果

利用 MATLAB 软件，结合上述建立的模型，可以求得南海区域的四个主要分潮的同潮图，如下图所示：

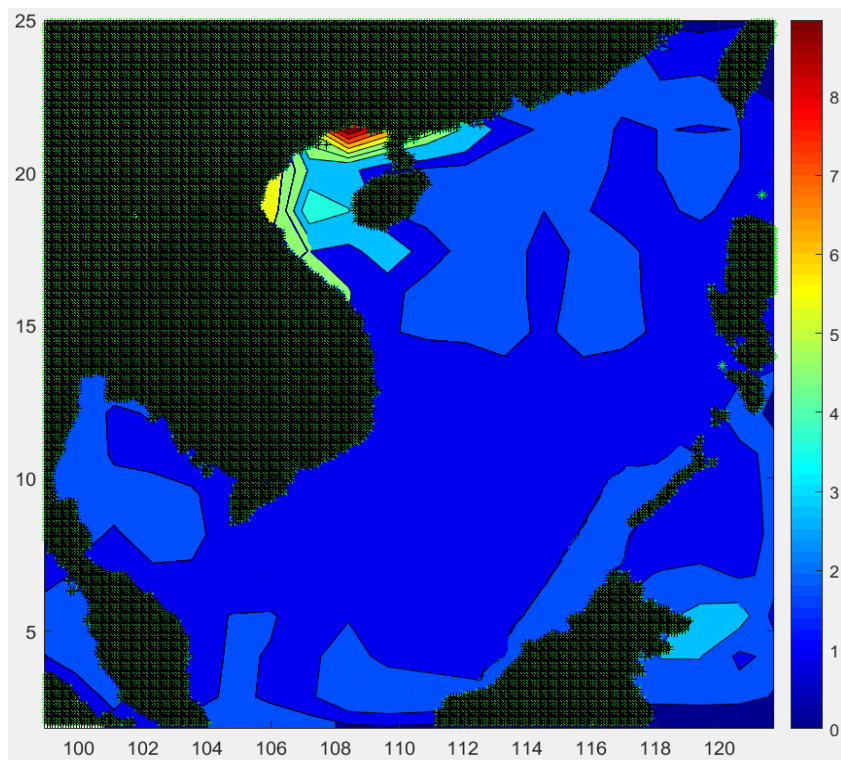


(图 34 南海区域 M2 分潮振幅 $H$ 的同潮图)

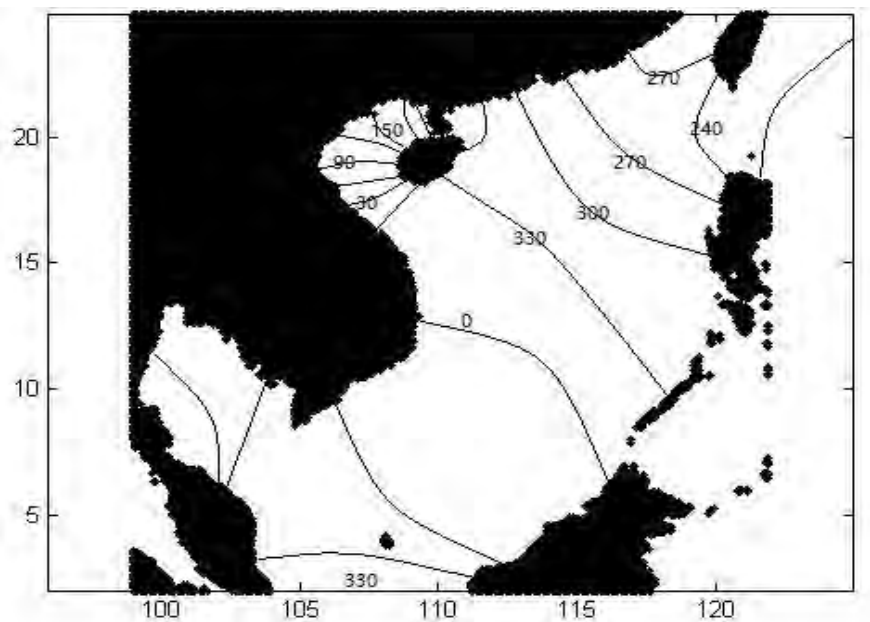


(图 35 南海区域 M2 分潮迟角 $G$ 的同潮图)

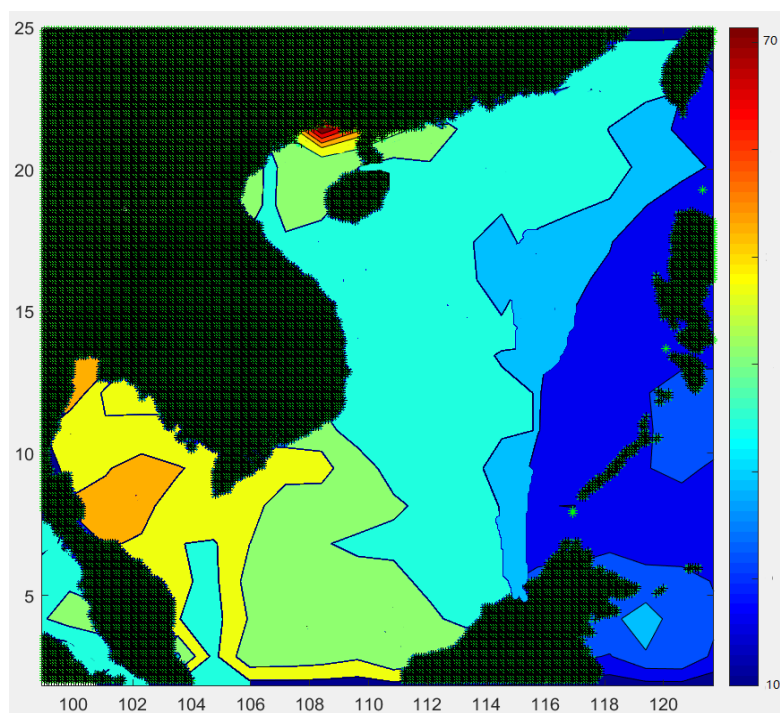




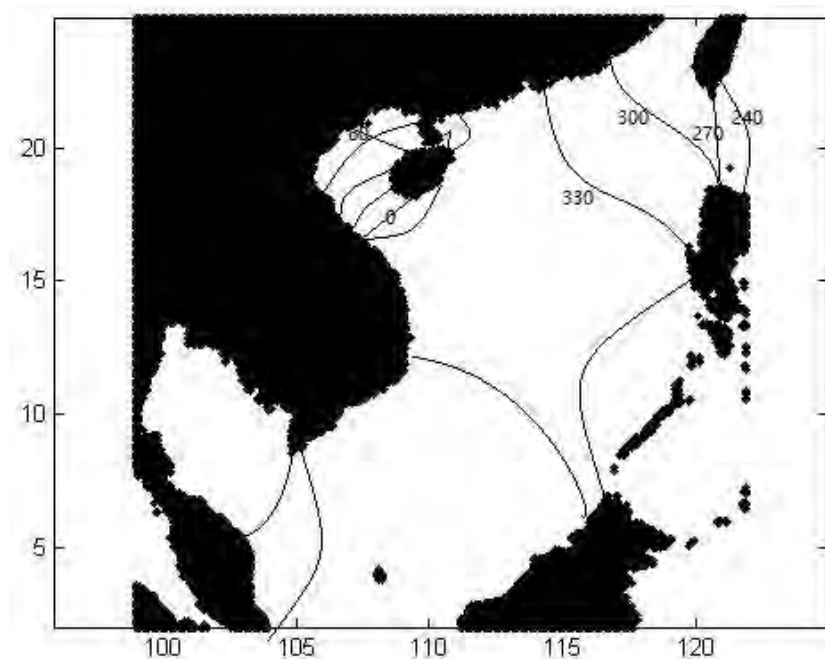
(图 36 南海区域 S2 分潮振幅 $H$ 的同潮图)



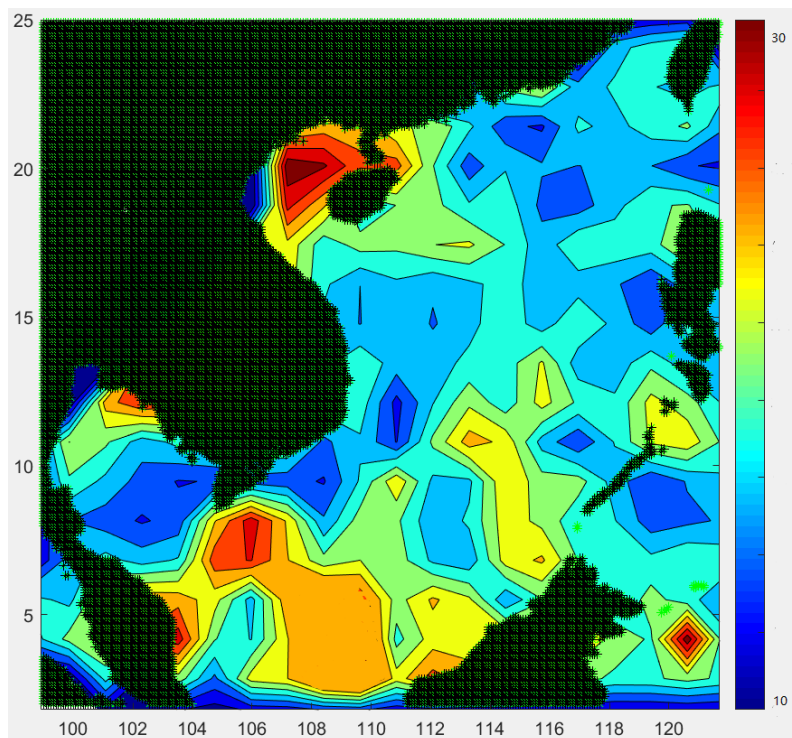
(图 37 南海区域 S2 分潮迟角 $G$ 的同潮图)



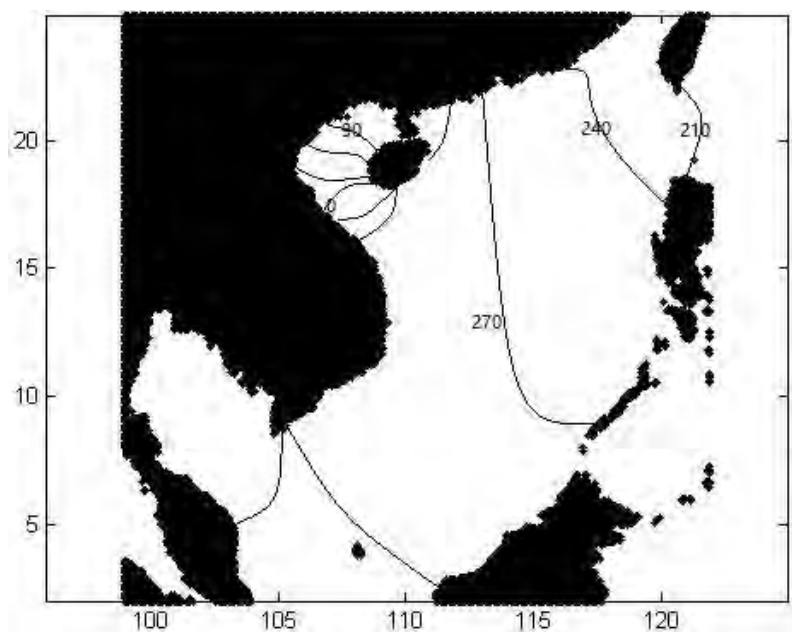
(图 38 南海区域 K1 分潮振幅 $H$ 的同潮图)



(图 39 南海区域 K1 分潮迟角 $G$ 的同潮图)



(图 40 南海区域 O1 分潮振幅 $H$ 的同潮图)



(图 41 南海区域 O1 分潮迟角 $G$ 的同潮图)

### 6.3 利用验潮站数据对插值法得到的同潮图进行评价

#### 6.3.1 数据处理

利用 MATLAB 处理题目中给出的验潮站数据，将其存储放置在一个 $[n \times 10]$ 的矩阵中，其中 $n$ 代表验潮站的总个数，每一行的十个数据分别代表验潮站的经度、纬度以及四个主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的潮汐调和常数振幅 $H$ 和迟角

$G$ 。以方便后续的调用计算。

### 6.3.2 模型准备

在通过验潮站数据对所绘制的同潮图准确性的验证过程中,需要借助 6.2 中的通过高阶多项式轨线插值法以及有效网格点邻近搜索高维插值法求得的模型进行验证。

### 6.3.3 模型建立与求解

将 6.3.1 中处理的验潮站数据中的经纬度代入上述模型中,计算得到对应的验潮站的各个主要分潮的潮汐调和函数的理论值。通过 MATLAB 将计算到的各个验潮站的主要分潮的潮汐调和常数振幅和迟角存入 $[n * 18]$ 的矩阵中,其中 $n$ 代表验潮站的个数,每一行的 18 个数分别代表验潮站的经度、纬度、四个主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的潮汐调和常数振幅 $H$ 和迟角 $G$ 的实际记录值、四个主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的潮汐调和常数振幅 $H$ 和迟角 $G$ 的理论计算值。

以同一个验潮站的四个主要分潮的相应潮汐调和常数的理论计算值与实际测量值之间的振幅偏差 $\Delta H$ ,潮汐矢量差 $\Delta$ ,振幅差均方差 $RMS_{\Delta H}$ 以及矢量差均方差 $RMS_{\Delta}$ 为评价标准对所绘制的同潮图进行评价,相应参数的计算公式见公式 4-13、4-14、4-15、4-16。

### 6.3.4 求解结果

#### (一) $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ 分潮的振幅差 $\Delta H_{M_2}$ 、 $\Delta H_{S_2}$ 、 $\Delta H_{K_1}$ 、 $\Delta H_{O_1}$

利用 MATLAB 可以分别计算得到同一验潮站的四个主要分潮的潮汐调和常数的理论值与实际值之间的振幅差 $\Delta H_{M_2}$ 、 $\Delta H_{S_2}$ 、 $\Delta H_{K_1}$ 、 $\Delta H_{O_1}$ ,部分数据如下表 15 所示:

(表 15  $\Delta H_{M2}$ 、 $\Delta H_{S2}$ 、 $\Delta H_{K1}$ 、 $\Delta H_{O1}$  部分对比数据)

	A	B	C	D	E	F	G
	序号	验潮站经度	验潮站纬度	M2振幅偏差	S2振幅偏差	K1振幅偏差	O1振幅偏差
1	1	99.816667	11.795	0.192081249	0.285050738	2.103593577	1.765671774
2	2	102.106667	6.226667	0.155686795	0.406612666	0.635891283	0.66795366
3	3	103.186667	5.265	0.328468632	0.291299134	1.773272319	0.9970915
4	4	103.43	3.975	4.028897068	0.953722425	3.514427831	2.463870743
5	5	104.14	2.806667	2.037992154	0.299702585	1.865366519	0.738714192
6	6	106.283333	18.083333	0.924117404	0.075032572	0.720352749	0.554659529
7	7	106.816667	20.666667	1.011303019	3.106175371	3.044551182	2.039717148
8	8	107.066667	10.333333	5.769173942	0.014483178	3.844493629	3.345212991
9	9	108.616667	19.1	0.298344527	0.07577579	2.692209161	0.896957069
10	10	109.083333	21.483333	1.789917679	0.696913069	5.062849895	0.043969888
11	11	110.283333	20.016667	0.146327648	0.29708023	0.638747738	2.449648033
12	12	111.833333	21.583333	4.214813751	0.754506233	1.506216408	1.513536358
13	13	113.066667	3.216667	0.495991079	0.070783674	0.201566614	1.302118492
14	14	113.55	22.166667	1.218932755	1.089170854	2.414107772	0.250963036
15	15	113.971667	4.391667	0.390418235	0.342858772	0.69512048	0.294660543
16	16	114.216667	22.3	2.914048418	0.360824471	2.088981582	0.85066939
17	17	115.35	22.75	0.597259236	0.231669558	1.452188606	1.234783846
18	18	116.066667	5.983333	4.351698381	4.331918363	0.248951638	0.399191407
19	19	117.75	2.1	4.164541935	0.261568334	0.226045507	0.245503407
20	20	117.883333	4.233333	1.158347476	4.11118981	0.76571234	0.579030638
21	21	118.066667	24.45	1.353067468	1.902174691	1.139929483	0.3915822
22	22	118.066667	5.81	0.523620083	1.261791271	0.915595564	1.367352274
23	23	118.733333	9.75	0.656293312	0.254469652	1.415093235	0.228541163
24	24	120.291667	22.61	0.430280496	0.255674446	0.315470192	0.082795373
25	25	120.966667	14.583333	0.067222343	0.648196856	0.604739455	0.070187696

(二) M2、S2、K1、O1 的  $RMS_{\Delta HM2}$ 、 $RMS_{\Delta HS2}$ 、 $RMS_{\Delta HK1}$ 、 $RMS_{\Delta HO1}$ 

利用 MATLAB 可以分别计算得到同一验潮站的四个主要分潮的潮汐调和常数的理论值与实际值之间的振幅差的均方差  $RMS_{\Delta HM2}$ 、 $RMS_{\Delta HS2}$ 、 $RMS_{\Delta HK1}$ 、 $RMS_{\Delta HO1}$ 。计算结果如表 16 所示：

(表 16  $RMS_{\Delta HM2}$ 、 $RMS_{\Delta HS2}$ 、 $RMS_{\Delta HK1}$ 、 $RMS_{\Delta HO1}$ )

$RMS_{\Delta HM2}$	$RMS_{\Delta HS2}$	$RMS_{\Delta HK1}$	$RMS_{\Delta HO1}$
4.321318998	1.135592118	2.8910464996	2.9380151231

(三) M2、S2、K1、O1 的矢量差  $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$ 

利用 MATLAB 可以分别计算得到同一验潮站的四个主要分潮的潮汐调和常数的理论值与实际值之间的振幅差的矢量差  $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$ 。部分计算结果如表 17 所示：

(表 17  $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$  部分数据)

	A	B	C	D	E	F	G
1	序号	验潮站经度	验潮站纬度	$\Delta_{M2}$	$\Delta_{S2}$	$\Delta_{K1}$	$\Delta_{O1}$
2	1	99.816667	11.795	0.154548527	0.305544153	0.241530119	1.250246535
3	2	102.106667	6.226667	0.273649543	0.106380381	0.274342774	0.285775569
4	3	103.186667	5.265	0.019932275	0.70631544	0.775484604	0.946490884
5	4	103.43	3.975	0.403045281	0.190617934	1.169137722	0.302843374
6	5	104.14	2.806667	0.440030149	0.790160066	0.469257353	0.278979381
7	6	106.283333	18.083333	0.139432023	0.771254337	0.782341464	0.061412678
8	7	106.816667	20.666667	0.086813082	0.604771544	1.290180925	1.564812454
9	8	107.066667	10.333333	1.050737505	0.943288617	1.472241783	1.230073781
10	9	108.616667	19.1	0.157517094	0.179082503	2.107862166	1.134920014
11	10	109.083333	21.483333	0.426779271	1.017095717	1.143967208	0.047500925
12	11	110.283333	20.016667	0.045115242	0.29530876	0.245334638	1.838086062
13	12	111.833333	21.583333	0.912118066	1.158996953	0.018868349	0.627634208
14	13	113.066667	3.216667	0.056376426	0.021485303	0.147400456	0.895901395
15	14	113.55	22.166667	0.657038515	0.054829984	1.008272246	0.475544324
16	15	113.971667	4.391667	0.027040429	0.762347388	1.01682301	1.033472617
17	16	114.216667	22.3	0.164179937	0.909822679	1.058964181	0.610249741
18	17	115.35	22.75	0.211736395	0.510876278	0.380975216	0.026291053
19	18	116.066667	5.983333	0.438133682	0.195839794	0.44722531	0.349698034
20	19	117.75	2.1	1.543531346	0.574072101	0.156189199	0.740157276
21	20	117.883333	4.233333	0.965985387	2.242994785	0.130565865	0.439268566
22	21	118.066667	24.45	0.523660696	3.099444566	0.60990287	0.167377205
23	22	118.066667	5.81	0.148759677	0.604406235	0.316350315	0.370287576
24	23	118.733333	9.75	0.35386767	0.341857989	0.824096156	0.461732278
25	24	120.291667	22.61	0.087626459	0.716657446	0.212301176	0.100048007
26	25	120.966667	14.583333	0.2170268	0.250713336	0.620271065	0.442855903

(四)  $M2$ 、 $S2$ 、 $K1$ 、 $O1$  矢量差均方差  $RMS_{\Delta M2}$ 、 $RMS_{\Delta S2}$ 、 $RMS_{\Delta K1}$ 、 $RMS_{\Delta O1}$ 

利用 MATLAB 可以分别计算得到同一验潮站的四个主要分潮的潮汐调和常数的理论值与实际值之间的振幅差的矢量差的均方差  $RMS_{\Delta M2}$ 、 $RMS_{\Delta S2}$ 、 $RMS_{\Delta K1}$ 、 $RMS_{\Delta O1}$ 。部分计算结果如表 18 所示：

(表 18  $RMS_{\Delta M2}$ 、 $RMS_{\Delta S2}$ 、 $RMS_{\Delta K1}$ 、 $RMS_{\Delta O1}$ )

$RMS_{\Delta HM2}$	$RMS_{\Delta HS2}$	$RMS_{\Delta HK1}$	$RMS_{\Delta HO1}$
4.135592118	1.891049466	2.938015152	2.309558244

## (五) 根据计算结果对提取得到的潮汐调和常数进行分析、评价

通过对上文中计算得到的同一验潮站的四个主要分潮  $M2$ 、 $S2$ 、 $K1$ 、 $O1$  理论计算值与实际测量值的振幅差  $\Delta H$ ，潮汐矢量差  $\Delta$ ，振幅差均方差  $RMS_{\Delta H}$  以及矢量差均方差  $RMS_{\Delta}$  分析可得：

所求得的验潮站的潮汐调和常数理论计算值与实际测量值之间的振幅差  $\Delta H_{M2}$ 、 $\Delta H_{S2}$ 、 $\Delta H_{K1}$ 、 $\Delta H_{O1}$  分别对应的最小值、最大值、均方差和平均值如下表 19 所示：

(表 19  $\Delta H_{M2}$ 、 $\Delta H_{S2}$ 、 $\Delta H_{K1}$ 、 $\Delta H_{O1}$  的最小值、最大值、均方差和平均值)

	$\Delta H_{M2}$	$\Delta H_{S2}$	$\Delta H_{K1}$	$\Delta H_{O1}$
min	0.005845	0.000388	2.647172	0.00326
max	4.69895	4.892526	0.01186	3.007717
RMS	1.949907	1.847521	1.613317	1.125755
average	2.081015	1.552348	1.936187	1.090057

所求得的验潮站的潮汐调和常数理论计算值与实践测量值之间的矢量差

$\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$ 分别对应最小值、最大值、均方差和平均值如下表 20 所示：

（表 20  $\Delta_{M2}$ 、 $\Delta_{S2}$ 、 $\Delta_{K1}$ 、 $\Delta_{O1}$ 的最小值、最大值、均方差和平均值）

	$\Delta_{M2}$	$\Delta_{S2}$	$\Delta_{K1}$	$\Delta_{O1}$
<i>min</i>	0.152558	0.055293	0.130577	0.171622
<i>max</i>	6.107241	6.801177	3.481833	5.024363
<i>RMS</i>	2.991068	1.171226	0.944306	1.227163
<i>average</i>	2.641627	1.486639	1.192259	1.548672

由表 19 可以看出，全部验潮站各个分潮的潮汐调和常数振幅差 $\Delta H$ 的最小值为 0.000388，振幅均方差 $RMS_{\Delta H}$ 的最小值为 1.891049466，最大值为 4.135592118。

由表 20 可以看出，全部验潮站各个分潮的潮汐调和常数矢量差 $\Delta$ 的最小值为 0.055293，最大值为 0.171622，振幅均方差 $RMS_{\Delta H}$ 的最小值为 0.944306，最大值为 2.991068。

由此可得，所求得的验潮站的四个主要分潮的潮汐调和常数的理论计算值与验潮站的测量实际数据吻合度较高，误差较小，同时证明验证了本模型的准确性。

## 7. 问题四 插值求出各分潮的同潮图，并利用验潮站检验

### 7.1 问题分析

在求解星下观测点的四个主要分潮的潮汐调和常数沿轨迹分布情况以及南海海域的同潮图的过程中，在通过数据插值的方法求解的过程中，利用了 12 阶次高阶多项式进行拟合，12 阶正压潮插值函数如下所示：

$$\begin{aligned}\hat{f}(y) = & 3.1312 * 10^{-11} y^{12} + 3.7421 * 10^{-9} y^{11} + 2.4765 * 10^{-7} y^{10} \\ & - 1.1531 * 10^{-5} y^9 + 0.0003731 * y^8 - 0.007896 * y^7 + 0.1046 * y^6 \\ & - 0.7344 * y^5 + 0.2700 * y^4 + 41.41 * y^3 - 349.5 * y^2 + 1265 * y^1 - 1788\end{aligned}\quad (7-1)$$

可以通过穷举的方式，将 3-18 阶次的共计 16 个高次多项式分别求出，同时算出各个高次多项式的计算值与实际值之间偏差的均方差 *RMS*，作为衡量标准，从而确定高次多项式拟合效果最佳时对应的最佳次数。

### 7.2 数据处理

在本章节中的数据处理可以直接利用问题 2 中和问题 3 中的已经通过 MATLAB 处理过的数据。

### 7.3 模型准备

在问题二中，本文采取的是 12 阶次的高阶多项式进行的插值计算，为了确定高次多项式的最佳次数，现在以 M2 分潮的调和常数沿轨迹方向正压潮和内潮的分离过程为例，依次求解 M2 分潮的正压潮、内潮的 3-18 次高阶多项式插值的系数，部分求解结果如下表所示：



(表 21 3-18 阶次部分高阶多项式系数)

	A	B	C	D	E	F	G	H
1	次数N		N-0	N-1	N-2	N-3	N-4	N-5
2	3	正压潮	0.0209793	-0.801648	9.9852686	-40.503783		
3		内潮	0.0234402	-0.883175	10.833656	-43.308249		
4	4	正压潮	-1.46E-05	0.0217193	-0.814901	10.0830732	-40.75128	
5		内潮	0.0008094	-0.017716	-0.14604	5.39382147	-29.54245	
6	5	正压潮	0.0008955	-0.044202	1.0916437	-13.062995	75.841961	-172.2927
7		内潮	0.0008432	-0.05276	1.2793825	-14.99474	85.115122	-189.0137
8	6	正压潮	-4.20E-06	0.0010152	-0.053954	1.24341016	-14.3272	81.159756
9		内潮	4.70E-05	-0.002735	0.0563993	-0.4194354	-0.843647	25.589705
10	7	正压潮	2.55E-05	-0.002288	0.0844498	-1.7006618	19.96965	-136.5782
11		内潮	3.75E-05	-0.003287	0.1201354	-2.3686371	27.157898	-180.8774
12	8	正压潮	4.72E-07	-2.25E-05	-0.000197	0.03494467	-0.985698	13.600156
13		内潮	3.18E-06	-0.000285	0.0106523	-0.213133	2.4445187	-15.72188
14	9	正压潮	1.08E-06	-0.000123	0.006081	-0.1718263	3.0475809	-35.14585
15		内潮	1.92E-06	-0.000216	0.0105938	-0.295268	5.1567761	-58.44502
16	10	正压潮	3.78E-08	-3.76E-06	0.0001497	-0.0028017	0.0134217	0.4696979
17		内潮	-3.11E-07	4.17E-05	-0.002452	0.08327086	-1.806763	26.131339
18	11	正压潮	1.70E-10	3.27E-08	-4.67E-06	0.0002298	-0.00587	0.0829508
19		内潮	4.40E-09	-4.09E-07	1.40E-05	-0.0001494	-0.003617	0.1481774
20	12	正压潮	3.13E-11	-3.74E-09	2.47E-07	-1.15E-05	0.0003707	-0.007897
21		内潮	1.71E-10	-1.71E-08	7.80E-07	-2.42E-05	0.0006523	-0.015316
22	13	正压潮	2.68E-11	-2.59E-09	8.71E-08	-5.79E-07	-3.28E-05	0.0005552
23		内潮	5.84E-11	-5.55E-09	1.81E-07	-1.02E-06	-7.04E-05	0.0010496
24	14	正压潮	5.17E-13	-2.31E-11	-9.74E-10	8.08E-08	-1.37E-06	-2.02E-05
25		内潮	1.42E-12	-8.05E-11	-9.04E-10	1.56E-07	-3.09E-06	-3.41E-05
26	15	正压潮	3.34E-14	-2.23E-12	1.78E-11	1.84E-09	-3.69E-08	-6.62E-07
27		内潮	4.43E-14	-2.39E-12	-5.71E-12	1.92E-09	1.34E-08	-1.92E-06
28	16	正压潮	-7.99E-16	7.86E-14	-1.79E-12	-4.60E-11	1.80E-09	3.29E-08
29		内潮	5.02E-16	9.60E-15	-1.99E-12	1.11E-11	1.87E-09	-6.01E-09

## 7.4 模型建立与求解

为评价不同阶次的高阶多项式的效果,本文采取高阶多项式插值函数的计算值与实际测量值之间的偏差的均方差 $RMS$ 为标准,对上述十六个高阶多项式插值性能进行评价,均方差越小说明插值效果越好,此时的高阶多项式对应的次数即为最佳次数。

均方差  $RMS$  的计算公式为:

$$RMS = \sqrt{\sum_{k=1}^N (\hat{f}(y) - f(y))^2} \quad (7-2)$$

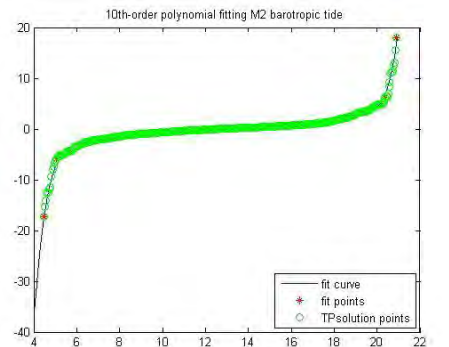
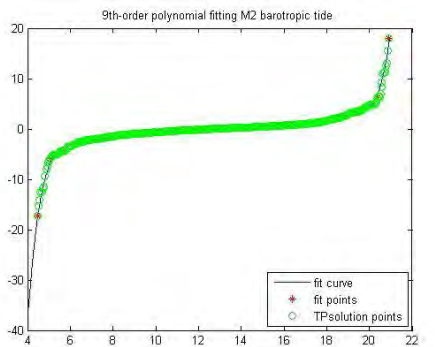
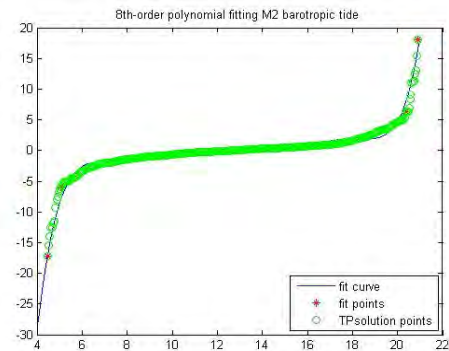
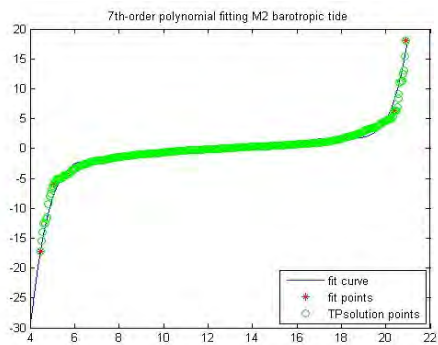
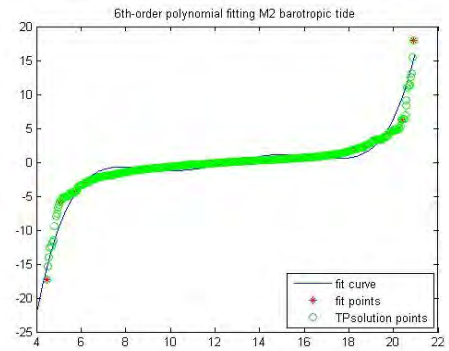
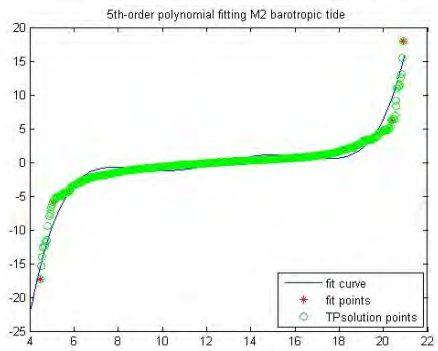
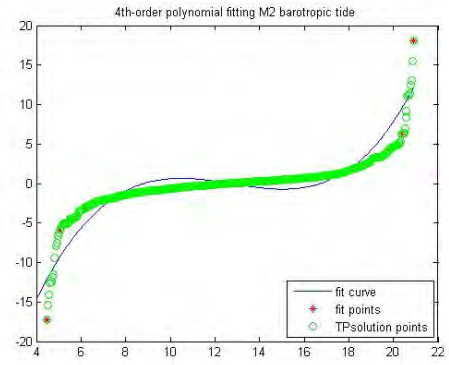
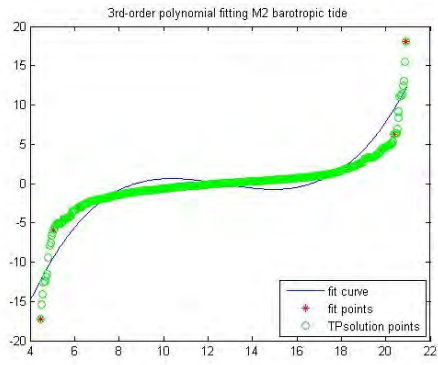
其中 $N$ 为待测点个数。

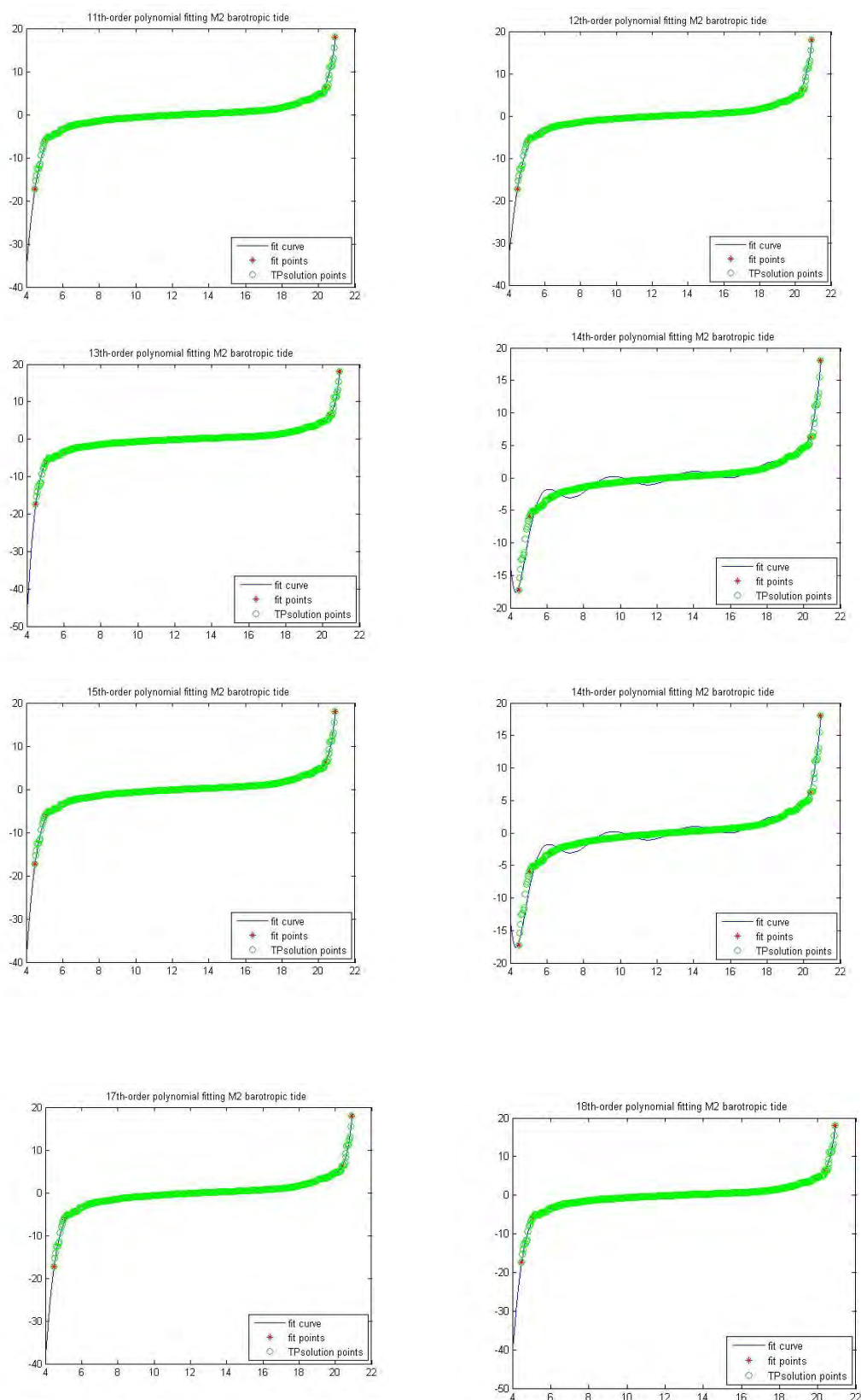
比较各个不同阶次的高阶多项式的均方差 $RMS$ ,最终选取最小值对应的高阶多项式,该多项式对应的次数即为最优次数。

## 7.5 求解结果

以 M2 分潮的正压潮为例,利用 MATLAB 计算并绘制 3-18 阶次高阶多项式插值函数,结果如下图所示:

其中实线为高阶多项式的差值函数曲线,绿色圆圈曲线代表实际的测量点构成的曲线。





(图 42 M2 分潮的正压潮 3-18 阶次高阶多项式插值公式)

利用 MATLAB 计算 M2 正压潮 3-18 阶高阶多项式的理论计算值与实际测量值偏差的均方差，计算结果见下表：

(表 22 M2 正压潮 3-18 阶多项式插值均方差比较)

阶数	3	4	5	6	7	8	9	10
RMS (cm)	2.7966	2.7976	1.9298	1.9299	1.2622	1.2528	0.7370	0.7280
阶数	11	12	13	14	15	16	17	18
RMS (cm)	0.7134	0.7998	0.6285	1.5328	0.4709	0.5652	0.4784	0.5103

通过插值函数图像与 *RMS* 可以看出,低阶插值函数在低纬度与高纬度处插值效果较差,存在较大的误差,15 阶 17 阶插值效果较好。

当高阶次多项式的阶次为 15 次时,均方差最小,多项式插值效果最佳,即 15 次为最佳阶次。

函数的阶数并非越高越好,从插值函数图像可以看出,14 阶多项式插值函数出现了振荡现象,即龙格现象。所谓龙格现象,一般来说,选择插值的节点个数越多,插值函数和被插值函数就会有越多相等的地方.但随着插值节点的个数增加,两个插值节点间插值函数与被插值函数并不一定能够很好地逼近.其次,从舍入误差来看,由于高次插值计算量大,可能会产生更严重的误差积累,因此,稳定性也无法保证.这就是龙格现象.一种解决龙格现象的方法是采用分段低次多项式插值:有分段线性插值和分段三次 Hermite 插值.在每个小区间采用低次插值,则可有效避免龙格现象。

## 8. 模型评价

### 8.1 模型的优点

(1)在解决问题一的过程中,所建立模型是在对初始数据进行清洗筛选,去除测量次数较少的非可信数据点的基础上所建立,排除了偶然因素对模型的扰动,模型可靠性较高。

利用调和分析法求解时,充分考虑格林尼治时间与北京时间之间的转化关系,并编写出一套完整的秒数时间转换系统,移植性较强。

在利用验潮站数据验证所得数据时,充分考虑了地球自身球体形状对两点间距离公式的影响,采用球面上两点之间距离,精度更高。

采用四个主要分潮相应潮汐调和常数的振幅差 $\Delta H$ ,振幅均方差 $RMS_{\Delta H}$ ,矢量差 $\Delta$ 和矢量差均方差 $RMS_{\Delta}$ 等共计四个参数,从不同角度对所得结果进行评价,评价结果可信度较高。

对数据存储格式进行优化,按照以观测点、验潮站经纬度为区分,存储相应数据信息于矩阵之中,方便后续计算调用。

(2)在解决问题二过程中,采用高阶次多项式轨线插值法构建函数,同时将星下观测点四个主要分潮潮汐调和常数沿轨迹变化图转变为沿纬度变化图,更加形象直观。

(3)在解决问题三的过程中,采用最近点跟踪法对海洋网格交叉点进行数据清洗,避免无效点的引入对模型结果造成影响,同时最近点跟踪法求出有效点周围四个邻近点,为后续有效网格点邻近搜索高维插值法进行高精度模型构建奠定基础。

(4)在解决问题四个过程中,引入均方差 $RMS$ 作为高阶次多项式插值效果评价指标,量化角度精确得到最佳阶次。

### 8.2 模型的缺点

(1)本文中的模型在建立时,对一些条件进行了假设,对某些信息考虑不足,与实际情况有些许差距。

(2)本文问题三的模型建立中,由于算法自身局限性导致所需要进行大量数据运算,模型求解对计算机运算能力要求较高。

(3)问题三模型建立中对于边界点处理较为简单,统一视为无效点废弃,对结果有些许影响。

## 9. 参考文献

- [1]. 王斌,张晓爽,吕咸青, etc. 高度计资料提取内潮信号的方法[J]. 解放军理工大学学报(自然科学版), 2015, 16 (03): 266-272.
- [2]. 朱学明. 中国近海潮汐潮流的数值模拟与研究[D]. 中国海洋大学 2009.
- [3]. 徐晓庆,方国洪,王新怡, etc. 渤、黄、东海潮汐的相对导纳及 N2,K2,P1 和 Q1 分潮的经验同潮图[J]. 海洋科学进展, 2011, 29 (3): 293-306.
- [4]. Fan, L. L.,Wang, B.,Lv, X. Q. Cotidal Charts near Hawaii Derived from TOPEX/Poseidon Altimetry Data[J]. Journal of Atmospheric & Oceanic Technology, 2011, 28 (4): 606-614.
- [5]. Camargo, R. D.,Harari, J. Modeling the Paranagua Estuarine Complex, Brazil: tidal circulation and cotidal charts[J]. Brazilian Journal of Oceanography, 2003, 51 (unico): 23-31.
- [6]. Fang, G.,Wang, Y.,Wei, Z., etc. Empirical cotidal charts of the Bohai, Yellow, and East China Seas from 10 years of TOPEX/Poseidon altimetry[J]. Journal of Geophysical Research Oceans, 2004, 109 (C11): -.
- [7]. Larsen, J. C. Cotidal Charts for the Pacific Ocean near Hawaii Using f-Plane Solutions[J]. Journal of Physical Oceanography, 1977, 7 (1): 100-109.
- [8]. Mazzega, P.,Bergé, M. Ocean tides in the Asian semiencloded seas from TOPEX/POSEIDON[M]. 1994; 24867-24882.
- [9]. Wang, Y.,Fang, G.,Wei, Z., etc. Cotidal charts and tidal power input atlases of the global ocean from TOPEX/Poseidon and JASON-1 altimetry[J]. Acta Oceanologica Sinica, 2012, 31 (4): 11-23.
- [10]. Yanagi, T.,Morimoto, A.,Ichikawa, K. Co-tidal and Co-range Charts for the East China Sea and the Yellow Sea Derived from Satellite Altimetric Data[J]. Journal of Oceanography, 1997, 53.
- [11]. 方国洪,曹德明,黄企洲. 南海潮汐潮流的数值模拟[J]. 海洋学报, 1994, 16 (4): 1-12.
- [12]. 陈倩. 浙江近海潮汐潮流的三维数值模拟[D]. 浙江大学 2002.
- [13]. 曹德明,方国洪. 南海北部潮汐潮流的数值模型[J]. 热带海洋学报, 1990, (2): 63-70.
- [14]. 王彪,朱建荣. 基于 FVCOM 模型的珠江河口及其邻近海域的潮汐模拟[J]. 热带海洋学报, 2012, 31 (4): 17-27.

## 10. 附录

#附录 1: 观测点数据整理源代码

#附录 2: 时间转换源代码

#附录 3: 整体计算结果源代码

#附录 1: 观测点数据整理源代码

%% 高度计数据提取

% 格式为 matrix(存储一个点多个时间、水位值), cell(存储不同点的矩阵), 另一个 matrix(存储每个点的经纬度与时间个数)

% 保留测量点的个数 position\_count, 测量点的经纬度 position, 时间水位 waterlevel, 时间 totalseconds, 水位 h\_total

%% 读取 dat 文件

TP = dlmread('资料\高度计\TP.dat');%将空的部分处理为 0, 共后续判断

%% 将文件内容逐个点分离, 并存为 cell, 经纬度存储为 matrix

position = [];waterlevel = {};totalseconds = {};h\_total = {};

line = 1;position\_count = 0;

while line <= size(TP,1)

if TP(line,3) ~= 0

position = [position;TP(line,:)];% 储存经纬度 matrix

if line == 1

startline = line+1;%定义第一个测量点数据的起始行

else

endline = line-1;%定义该测量点数据的终止行

totalseconds{position\_count} = TP(startline:endline,1) + 8\*3600\*ones(endline-startline+1,1);%格林尼治时间转换为北京时间

h\_total{position\_count} = 100\*TP(startline:endline,2);%单位由米转换为厘米

waterlevel{position\_count} = TP(startline:endline,1:2);%将时间和水位赋值, 遇到下一个测量点时给上一个测量点的矩阵赋值

startline = line+1;%重新定义下一个测量点数据的起始行

end

position\_count = position\_count+1;%记录有多少个测量点

end

if line == size(TP,1)

totalseconds{position\_count} = TP(startline:endline,1) + 8\*3600\*ones(endline-startline+1,1);%格林尼治时间转换为北京时间

h\_total{position\_count} = 100\*TP(startline:endline,2);

waterlevel{position\_count} = TP(startline:end,1:2);

```

        end
        line = line + 1;%TP 中的数据逐行扫描
    end
    clear line startline endline TP

%% 数据清洗
% 注意有些测量点处测量的时刻小于 9 个，要把这些点排除掉，数据清洗
index = find(position(:,3)<300);
position(index,:) = [];
position_count = position_count-length(index);
totalseconds(index) = [];
h_total(index) = [];
waterlevel(index) = [];
clear index

#附录 2：时间转换源代码
%% 时间转换中的 Y 计算
% Y = totalseconds/3600/24/365or366 年数
% n = totalseconds/3600/24 剩余天数
% i = Y/4（需要判断，整除 4 或者 400 之类的） 闰年数
% t = (totalseconds - (Y-2000)*365*24*3600 - i*24*3600 -
n*24*3600)/3600; 剩余小时数
function [Y,n,i,t] = timeconvert(totalseconds)

%% 求解总天数和剩余小时数
totaldays = fix(totalseconds/3600/24);
t = totalseconds - totaldays*3600*24;%一天多余的小时数
if t < 0
    t = 3600*24+t;%负数小时数的处理方法
end
t = t/3600;

%% 考虑闰年求年数和到 1900 年的闰年数
% 2000 年之后至今的闰年每 4 年一次，不用考虑整除 100 的特例
year = fix(totaldays/365);
if totaldays >= 0
    runnian = fix((year-1)/4) + 1;
    year = fix((totaldays - runnian)/365);%把包括 2000 年的闰年数
减掉
else
    runnian = fix(year/4);
    year = fix((totaldays - runnian)/365) - 1;%负时间的年数计算
end
Y = 2000+year;%按照北京时间 2000 年 1.1 之后计算

```



```

i = fix((Y-1901)/4);%闰年个数

%% 一年求累积天数
if totaldays > 0
    n = totaldays - fix((year+3)/4) - year*365;
else
    n = totaldays - fix((year+1)/4) - (year+1)*365;
    if mod(Y,4) == 0
        n = 365 + n;%需要考虑余下的小时数占用了时间，每年总天数减
    else
        n = 364 + n;
    end
end

end

#附录 3：整体计算结果源代码
%% 主函数，执行的入口点
clear
tic
%% 读文件
gaoduji_dataprocess;

%% 初始化
const_initial;

%% 每个测量点循环求解
for j = 1:position_count
    [f{j},u{j}] = fusolve(p{j},N_pie{j});
    x{j} = equationsolve(h_total{j},f{j},sigma,v0{j},u{j},t{j});
    [h{j},g{j}] = hgsolve(x{j});
end
clear x f u p N_pie sigma v0 t

fileID = fopen('celldata_h.dat','w');
for i = 1:length(h)
    formatSpec = '%f %f %f %f\n';
    fprintf(fileID,formatSpec,h{i});
end
fclose(fileID);

```

```

fileID = fopen('celldata_g.dat','w');
for i = 1:length(g)
    formatSpec = '%f %f %f %f\n';
    fprintf(fileID,formatSpec,g{i});
end
fclose(fileID);
clear fileID formatSpec

final_result = zeros(position_count,10);
for i = 1:length(h)
    final_result(i,1:2) = position(i,1:2);
    final_result(i,3:6) = h{i};
    final_result(i,7:10) = g{i};
end
clear position_count i j
xlswrite('final_result.xls',final_result)
toc
%% 输出

%% 高度计数据提取
% 格式为 matrix(存储一个点多个时间、水位值), cell(存储不同点的矩阵), 另一个 matrix (存储每个点的经纬度与时间个数)
% 保留测量点的个数 position_count, 测量点的经纬度 position, 时间水位 waterlevel, 时间 totalseconds, 水位 h_total

%% 读取 dat 文件
TP = dlmread('资料\高度计\TP.dat');%将空的部分处理为 0, 共后续判断

%% 将文件内容逐个点分离, 并存为 cell, 经纬度存储为 matrix
position = [];waterlevel = {};totalseconds = {};h_total = {};
line = 1;position_count = 0;
while line <= size(TP,1)
    if TP(line,3) ~= 0
        position = [position;TP(line,:)];% 储存经纬度 matrix
        if line == 1
            startline = line+1;%定义第一个测量点数据的起始行
        else
            endlime = line-1;%定义该测量点数据的终止行
            totalseconds{position_count} = TP(startline:endlime,1)
+ 8*3600*ones(endlime-startline+1,1);%格林尼治时间转换为北京时间
            h_total{position_count} =
100*TP(startline:endlime,2);%单位由米转换为厘米
            waterlevel{position_count} =
TP(startline:endlime,1:2);%将时间和水位赋值, 遇到下一个测量点时给上一

```

个测量点的矩阵赋值

```

        startline = line+1;%重新定义下一个测量点数据的起始行
    end
    position_count = position_count+1;%记录有多少个测量点
end
if line == size(TP,1)
    totalseconds{position_count} = TP(startline:endline,1) +
8*3600*ones(endline-startline+1,1);%格林尼治时间转换为北京时间
    h_total{position_count} = 100*TP(startline:endline,2);
    waterlevel{position_count} = TP(startline:end,1:2);
end
line = line + 1;%TP 中的数据逐行扫描
end
clear line startline endline TP
% raw=cell(line_count,1);
% % frewind(fid);
% line=1;
% while feof(fid)==0
%     raw{line,1}=fgetl(fid);
%     line=line+1;
% end

%% 数据清洗
% 注意有些测量点处测量的时刻小于 9 个，要把这些点排除掉，数据清洗
index = find(position(:,3)<300);
position(index,:) = [];
position_count = position_count-length(index);
totalseconds(index) = [];
h_total(index) = [];
waterlevel(index) = [];
clear index

%% 计算用到的常数初始化
% 包含 7 个 dooson 数，分潮角速度 sigma，
tao1,s1,h_pie1,p1,N_pie1,p_pie1
% 分潮初相位 v0, tao,s,h_pie,p,N_pie,p_pie（与时间有关，写为函数）
% 正交因子 f 与修正角 u（与时间有关，写为函数）

%% 计算 sigma
tao1 = 14.49205211;
s1 = 0.54901653;
h_pie1 = 0.04106864;
p1 = 0.00464183;
N_pie1 = 0.00220641;

```

```

p_piel = 0.00000196;

miuM2 = [2, 0, 0, 0, 0, 0, 0];
miuS2 = [2, 2, -2, 0, 0, 0, 0];
miuK1 = [1, 1, 0, 0, 0, 0, 1];
miuO1 = [1, -1, 0, 0, 0, 0, -1];

sigma
[miuM2(1)*tao1+miuM2(2)*s1+miuM2(3)*h_piel+miuM2(4)*p1+miuM2(5)*N_pie
1+miuM2(6)*p_piel;

miuS2(1)*tao1+miuS2(2)*s1+miuS2(3)*h_piel+miuS2(4)*p1+miuS2(5)*N_piel
+miuS2(6)*p_piel;

miuK1(1)*tao1+miuK1(2)*s1+miuK1(3)*h_piel+miuK1(4)*p1+miuK1(5)*N_piel
+miuK1(6)*p_piel;

miuO1(1)*tao1+miuO1(2)*s1+miuO1(3)*h_piel+miuO1(4)*p1+miuO1(5)*N_piel
+miuO1(6)*p_piel;
];
clear tao1 s1 h_piel p1 N_piel p_piel

%% 计算 vo
v0 = cell(size(totalseconds));
for j=1:length(totalseconds)
    [Y{j}, n{j}, i{j}, t{j}] = timeconvert(totalseconds{j});
    s{j} = 277.02*ones(length(Y{j}), 1) + 129.3848*(Y{j}-
1900*ones(length(Y{j}), 1)) + 13.1764*(n{j}+i{j}+t{j}/24);
    h_pie{j} = 280.19*ones(length(Y{j}), 1) - 0.2387*(Y{j}-
1900*ones(length(Y{j}), 1)) + 0.9857*(n{j}+i{j}+t{j}/24);
    p{j} = 334.39*ones(length(Y{j}), 1) + 40.6625*(Y{j}-
1900*ones(length(Y{j}), 1)) + 0.1114*(n{j}+i{j}+t{j}/24);
    N_pie{j} = 100.84*ones(length(Y{j}), 1) + 19.3282*(Y{j}-
1900*ones(length(Y{j}), 1)) + 0.0530*(n{j}+i{j}+t{j}/24);
    p_pie{j} = 281.22*ones(length(Y{j}), 1) + 0.0172*(Y{j}-
1900*ones(length(Y{j}), 1)) + 0.00005*(n{j}+i{j}+t{j}/24);
    tao{j} = 15*t{j} - s{j} + h_pie{j};
    v01 = miuM2(1)*tao{j} + miuM2(2)*s{j} + miuM2(3)*h_pie{j} +
miuM2(4)*p{j} + miuM2(5)*N_pie{j} + miuM2(6)*p_pie{j} +
miuM2(7)*90*ones(size(tao{j}));
    v02 = miuS2(1)*tao{j} + miuS2(2)*s{j} + miuS2(3)*h_pie{j} +
miuS2(4)*p{j} + miuS2(5)*N_pie{j} + miuS2(6)*p_pie{j} +
miuS2(7)*90*ones(size(tao{j}));
    v03 = miuK1(1)*tao{j} + miuK1(2)*s{j} + miuK1(3)*h_pie{j} +

```

```

miuK1(4)*p{j} + miuK1(5)*N_pie{j} + miuK1(6)*p_pie{j} +
miuK1(7)*90*ones(size(tao{j}));
v04 = miu01(1)*tao{j} + miu01(2)*s{j} + miu01(3)*h_pie{j} +
miu01(4)*p{j} + miu01(5)*N_pie{j} + miu01(6)*p_pie{j} +
miu01(7)*90*ones(size(tao{j}));
v0{j} = [v01, v02, v03, v04];
% v0{j} = [miuM2(1)*tao{j} + miuM2(2)*s{j} + miuM2(3)*h_pie{j}
+ miuM2(4)*p{j} + miuM2(5)*N_pie{j} + miuM2(6)*p_pie{j} +
miuM2(7)*90*ones(size(tao{j})),
% miuS2(1)*tao{j} + miuS2(2)*s{j} + miuS2(3)*h_pie{j} +
miuS2(4)*p{j} + miuS2(5)*N_pie{j} + miuS2(6)*p_pie{j} +
miuS2(7)*90*ones(size(tao{j})),
% miuK1(1)*tao{j} + miuK1(2)*s{j} + miuK1(3)*h_pie{j} +
miuK1(4)*p{j} + miuK1(5)*N_pie{j} + miuK1(6)*p_pie{j} +
miuK1(7)*90*ones(size(tao{j})),
% miu01(1)*tao{j} + miu01(2)*s{j} + miu01(3)*h_pie{j} +
miu01(4)*p{j} + miu01(5)*N_pie{j} + miu01(6)*p_pie{j} +
miu01(7)*90*ones(size(tao{j}))
% ];
end
% s = s(Y,n,i,t);
% h_pie = h_pie(Y,n,i,t);
% p = p(Y,n,i,t);
% N_pie = N_pie(Y,n,i,t);
% p_pie = p_pie(Y,n,i,t);
% tao = tao(t,s,h_pie);
clear v01 v02 v03 v04 j
clear miuM2 miuS2 miuK1 miu01
clear tao s h_pie p_pie
clear Y n i

%% 时间转换中的 Y 计算
% Y = totalseconds/3600/24/365or366 年数
% n = totalseconds/3600/24 剩余天数
% i = Y/4 (需要判断, 整除 4 或者 400 之类的) 闰年数
% t = (totalseconds - (Y-2000)*365*24*3600 - i*24*3600 -
n*24*3600)/3600; 剩余小时数
function [Y,n,i,t] = timeconvert(totalseconds)

%% 求解总天数和剩余小时数
totaldays = fix(totalseconds/3600/24);
t = totalseconds - totaldays*3600*24;%一天多余的小时数
if t < 0
t = 3600*24+t;%负数小时数的处理方法

```

```

end
t = t/3600;

%% 考虑闰年求年数和到 1900 年的闰年数
% 2000 年之后至今的闰年每 4 年一次，不用考虑整除 100 的特例
year = fix(totaldays/365);
if totaldays >= 0
    runnian = fix((year-1)/4) + 1;
    year = fix((totaldays - runnian)/365);%把包括 2000 年的闰年数
减掉
else
    runnian = fix(year/4);
    year = fix((totaldays - runnian)/365) - 1;%负时间的年数计算
end
Y = 2000+year;%按照北京时间 2000 年 1.1 之后计算

i = fix((Y-1901)/4);%闰年个数

%% 一年求累积天数
if totaldays > 0
    n = totaldays - fix((year+3)/4) - year*365;
else
    n = totaldays - fix((year+1)/4) - (year+1)*365;
    if mod(Y,4) == 0
        n = 365 + n;%需要考虑余下的小时数占用了时间，每年总天数减
一
    else
        n = 364 + n;
    end
end
end

end

%% 求解 f, u
%不同潮汐分量计算出不同的 f, u, 潮汐分量编号 1-10 (第一问只用到了 4)
%分潮编号如下: M2, S2, K1, O1 分别为 1-4
%Sa, Ssa, Mm, Msf, Mf, Q1, M1, P1, S1, J1, O01, N2, L2, k2, M4, MS4, M6
function [f,u] = fusolve(p,N_pie)

% totalseconds = 8e7;%随便写的
% [Y,n,i,t] = timeconvert(totalseconds);
% p = 334.39 + 40.6625*(Y-1900) + 0.1114*(n+i+t/24);
% N_pie = 100.84 + 19.3282*(Y-1900) + 0.0530*(n+i+t/24);
f = zeros(length(p),4);u = zeros(length(p),4);

```

```

    fcosuM2 = 0.0005*cosd(-2*N_pie) + 1 + 0.0006*cosd(2*p) +
0.0002*cosd(2*p+N_pie);
    fsinuM2 = 0.0005*sind(-2*N_pie) + 0 + 0.0006*sind(2*p) +
0.0002*sind(2*p+N_pie);
    f(:,1) = sqrt(fcosuM2.^2+fsinuM2.^2);
    % u(:,1) = atand(fsinuM2./fcosuM2);%M2 分潮的交点因子与订正角
角度需要修正
    u(:,1) = arctancorrection(fsinuM2,fcosuM2);

    f(:,2) = 1;
    u(:,2) = 0;%S2 分潮的交点因子与订正角

    fcosuK1 = 0.0002*cosd(-2*p-N_pie) + 0.0001*cosd(-2*N_pie) + 1 +
0.1356*cosd(N_pie);
    fsinuK1 = 0.0002*sind(-2*p-N_pie) + 0.0001*sind(-2*N_pie) + 0 +
0.1356*sind(N_pie);
    f(:,3) = sqrt(fcosuK1.^2+fsinuK1.^2);
    % u(:,3) = atand(fsinuK1./fcosuK1);%K1 分潮的交点因子与订正角
    u(:,3) = arctancorrection(fsinuK1,fcosuK1);

    fcosu01 = 0.1885*cosd(-N_pie) + 1 + 0.0002*cosd(2*p-N_pie);
    fsinu01 = 0.1885*sind(-N_pie) + 0 + 0.0002*sind(2*p-N_pie);
    f(:,4) = sqrt(fcosu01.^2+fsinu01.^2);
    % u(:,4) = atand(fsinu01./fcosu01);%01 分潮的交点因子与订正角
    u(:,4) = arctancorrection(fsinu01,fcosu01);
end

%% 修正角度值，根据 sin 和 cos 值判断象限
function angle = arctancorrection(sina,cosa)

[m,n] = size(sina);
angle = zeros(m,n);
for i = 1:m
    for j = 1:n
        if cosa(i,j) > 0
            if sina(i,j) > 0
                angle(i,j) = atand(sina(i,j)/cosa(i,j));
            else
                angle(i,j) = atand(sina(i,j)/cosa(i,j))+360;
            end
        else
            angle(i,j) = atand(sina(i,j)/cosa(i,j))+180;
        end
    end
end
end

```

```

end

end

%% 方程组求解，输出 x1~x8，只需要写出解一个测量点的方程组求解即可
%%根据 h_total,fj,sigmaj,v0j,uj，列出 n 个方程组（n 代表一个点观测的
时间数）
%求出 a11~anm
%将偏导函数表示出来 delta = sum(),（需要推公式，求偏导）

function x = equationsolve(h_total,f,sigma,v0,u,t)

a = zeros(length(h_total),9);
a(:,1) = f(:,1).*cosd(sigma(1).*t(:)+v0(:,1)+u(:,1));%分潮 1cos
a(:,2) = f(:,2).*cosd(sigma(2).*t(:)+v0(:,2)+u(:,2));%分潮 2cos
a(:,3) = f(:,3).*cosd(sigma(3).*t(:)+v0(:,3)+u(:,3));%分潮 3cos
a(:,4) = f(:,4).*cosd(sigma(4).*t(:)+v0(:,4)+u(:,4));%分潮 4cos

a(:,5) = f(:,1).*sind(sigma(1).*t(:)+v0(:,1)+u(:,1));%分潮 1sin
a(:,6) = f(:,2).*sind(sigma(2).*t(:)+v0(:,2)+u(:,2));%分潮 2sin
a(:,7) = f(:,3).*sind(sigma(3).*t(:)+v0(:,3)+u(:,3));%分潮 3sin
a(:,8) = f(:,4).*sind(sigma(4).*t(:)+v0(:,4)+u(:,4));%分潮 4sin
a(:,9) = ones(length(h_total),1);

B = zeros(9);%B 为法方程的系数矩阵
for i = 1:9
    for j = 1:9
        B(i,j) = 0;
        for k = 1:length(h_total)
            B(i,j) = B(i,j) + a(k,i)*a(k,j);
        end
    end
end

H = zeros(9,1);
for i = 1:length(h_total)
    H = H + a(i,:)'*h_total(i);
end

x = B\H;
end

%% 将 equationsolve 中输出的 x1~x8 转换为 h,g
%x1 = h1*cos(g1),x2 = h1*sin(g1)

```



```

%x3 = h2*cos(g2), x4 = h2*sin(g2)
%x5 = h3*cos(g3), x6 = h3*sin(g3)
%x7 = h4*cos(g4), x8 = h4*sin(g4)

function [h,g] = hgsolve(x)

h(1) = sqrt(x(1)^2+x(5)^2);%M2 振幅
% g(1) = atand(x(5)/x(1));%M2 迟角    cos 小于 0 时要修正角度
g(1) = arctancorrection(x(5),x(1));

h(2) = sqrt(x(2)^2+x(6)^2);%S2 振幅
% g(2) = atand(x(6)/x(2));%S2 迟角
g(2) = arctancorrection(x(6),x(2));

h(3) = sqrt(x(3)^2+x(7)^2);%K1 振幅
% g(3) = atand(x(7)/x(3));%K1 迟角
g(3) = arctancorrection(x(7),x(3));

h(4) = sqrt(x(4)^2+x(8)^2);%O1 振幅
% g(4) = atand(x(8)/x(4));%O1 迟角
g(4) = arctancorrection(x(8),x(4));
end

```