



“华为杯”第十五届中国研究生 数学建模竞赛

学 校 西安电子科技大学

参赛队号 18107010021

队员姓名	1.杨 琦
	2.李镇宁
	3.艾 壮

“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：

本文根据卫星高度计海面高度异常资料建立了提取潮汐调和常数的 j,v 模型，运用最小二乘和多项式拟合的方法求出了南海各主要分潮的调和常数，讨论并解决了相关问题。

针对问题一，构建了 j,v 模型进行潮汐的调和分析。首先对海面高度异常值数据进行了筛选处理：为避免频率混叠效应，选取从 2003 年初到 2016 年末共计 14 年的采样数据；为提高分析结果的有效性，选取观测次数大于 300、水深大于 200 米的数据用于数值模拟工作。接着构建了根据海面高度异常值来提取星下观测点主要分潮的潮汐调和常数的 j,v 模型：首先计算分潮角速度及初相位，然后计算各分潮的交点因子及交点订正角，最后通过最小二乘法提取出四个主要分潮（ M_2 、 S_2 、 K_1 、 O_1 ）的潮汐调和常数。为验证结果的可靠性，将调和分析结果同 12 个验潮站的调和常数进行了比较，得到四个分潮的单分潮预报误差（RMS）均小于 1.5cm，最大值为 M_2 潮的 1.41cm，最小值为 O_1 潮的 0.49cm；各分潮振幅差的平均值 ΔH 均小于 2.95cm，最大值和最小值分别为 M_2 潮的 2.95cm 和 S_2 潮的 1.56cm；各分潮迟角差的平均值均小于 6.53° ，最大值和最小值分别为 M_2 潮的 6.53° 和 S_2 潮的 2.02° ，表明所构建的 j,v 模型调和分析结果可靠。

针对问题二，构建了正压潮与内潮分离模型。首先选取上行/下行两条交叉轨道，然后将振幅 H 、迟角 G 转化为 $H \cos G$ 、 $H \sin G$ ，分别对 $H \cos G$ 、 $H \sin G$ 沿轨道进行多项式拟合，最后通过坐标转换得到轨道上任意点的调和常数，进而实现将正压潮与内潮的分离。通过对比多项式拟合与调和分析结果示表明：对于部分分潮（如 M_2 、 S_2 分潮）的振幅拟合结果的大致趋势走向可与调和分析结果保持一致，但对于某些位置会有一定幅度的波动。考虑到轨道所跨越的经度较小（仅有 4° ），分辨率较高，得到的结果是可以接受的。

针对问题三，为得到整个南海的调和常数，首先选取一个了初始的“中心点”，以该点为中心作小正方形范围内的搜索，对该范围内所有满足要求的星下观测点进行调和分析，得到该范围内所有点的调和常数并取其平均值，记录该“中心点”

的经纬度和平均调和常数。当该区域结束后移动“中心点”到下一个位置，重复该操作直到遍历完所有的“中心点”。根据所得的调和常数拟合出南海各主要分潮的同潮图，分析可知南海潮型以全日潮 K_1 和 O_1 为主，四个分潮在深海区传播速度较快，等值线稀疏。相对于全日潮，南海大部分海区的半日潮影响较小，半日潮 M_2 在台湾海峡及海南岛附近有较大振幅。 S_2 分潮在湄公河入海口处振幅较大，超过 30cm，在南海整体振幅较小。这与多数文献中南海潮波主要由太平洋由吕宋海峡传入南海向西南方向前进，部分半日潮向台湾海峡前进与其它潮波系统发生作用的结论基本一致。

针对问题四，根据问题二所构建的模型对各主要分潮的 $H \cos G$ 、 $H \sin G$ 沿轨进行 3~11 次多项式拟合。对比可知 3~11 次的多项式拟合结果均可反映出海表面 M_2 、 S_2 、 K_1 、 O_1 各分潮的振幅变化趋势，但不同次数多项式拟合结果与海表面各分潮振幅及迟角的拟合程度存在明显差异。以 M_2 分潮为例，3 次多项式的拟合结果明显偏离调和与分析得到的振幅值，不能正确地拟合出正压潮值；采用 7、9 次多项式拟合的振幅绝对均差相当，优于其他次数的拟合效果，采用 7 次多项式拟合的迟角平均误差最小。综合考虑振幅与迟角的平均误差，得出 7 次为最佳的多项式拟合次数。

关键词：T/P 卫星高度计 潮汐调和常数 j, v 模型 最小二乘法 多项式拟合

目 录

一、问题重述	5
1.1 潮汐潮流现象的研究意义	5
1.2 潮汐潮流数值模拟所面临的问题	5
1.3 资料描述	5
1.3.1 地形数据	5
1.3.2 验潮站资料	6
1.3.3 TOPEX/POSEIDON 卫星高度计简介	7
1.3.4 南海高度计资料	7
1.4 要解决的问题	7
二、模型假设	8
三、符号说明	8
四、问题分析	9
4.1 数据处理	9
4.1.1 对时间系统的处理	9
4.1.1 对采样时间的处理	9
4.1.2 对观测点的处理	9
4.1.3 对水深数据的处理	10
4.2 问题一的模型建立与求解	10
4.2.1 问题一的描述与分析	10
4.2.2 问题一的模型建立	11
4.2.3 问题一的模型求解与结果检验分析	15
4.3 问题二的模型建立与求解	18
4.3.1 问题二的描述与分析	18
4.3.2 问题二的模型建立	19
4.3.3 问题二的模型求解	20

4.3.4 问题二模型结果检验分析.....	21
4.4 问题三的模型建立与求解.....	23
4.4.1 问题三的描述与分析.....	23
4.4.2 问题三的建模与求解.....	24
4.4.3 问题三结果评价分析.....	25
4.5 问题四的模型建立与求解.....	26
4.5.1 问题四的描述与分析.....	26
4.5.2 问题四的求解与启示.....	26
五、模型的评价与推广.....	27
5.1 模型的优点.....	27
5.2 模型的缺点.....	28
5.3 模型的推广.....	28
六、参考文献.....	28
七、附录.....	28
7.1 问题一的源代码.....	28
7.2 问题二的源代码.....	38
7.3 问题三的源代码.....	40
7.4 问题四的源代码.....	43

一、问题重述

1.1 潮汐潮流现象的研究意义

海洋潮汐是在天体引潮力作用下形成的长周期波动现象，在水平方向上表现为潮流的涨落，在铅直方向上则表现为潮位的升降。潮汐潮流运动是海洋中的基本运动之一，它是动力海洋学研究的重要组成部分，对它的研究直接影响着波浪、风暴潮、环流、水团等其他海洋现象的研究，在大陆架浅海海洋中，对潮汐潮流的研究更具重要性。

海岸附近和河口区域是人类进行生产活动十分频繁的地带，而这个地带的潮汐现象非常显著，它直接或间接地影响着人们的生产和生活。潮汐潮流工作的开展和研究，可为国防建设、交通运输、海洋资源开发、能源利用、环境保护、海港建设和海岸防护提供资料。例如，沿海地区的海滩围垦、农田排灌，水产的捕捞和养殖，制盐，海港的选址及建设，以至于潮能发电等活动，无不与潮汐潮流现象有着密切的关系。

1.2 潮汐潮流数值模拟所面临的问题

区域海洋潮汐的数值模拟需要提供开边界的水位调和常数，而开边界的水位调和常数，或者来源于观测、或者来源于全球海洋潮汐的数值模拟；而全球海洋潮汐的数值模拟，相当耗费资源。虽然目前有国外学者或研究机构，能够提供区域海洋潮汐的调和常数，但实质上的评价结果难以令人满意。

从区域海洋潮汐的数值模拟的现状来讲，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可以得到令人满意的结果，但其它分潮（ N_2 、 K_2 、 P_1 、 Q_1 等）的单一分潮的数值模拟与同化，结果却差强人意；这意味着其它分潮的数值模拟，只有与四个主要分潮同时进行数值模拟，才能得到可以接受的结果。从具体操作来讲，其它分潮由于相对较弱，导致模拟结果的精度难以提高。

长周期分潮（ S_a 、 S_{sa} 、 M_m 、 M_f ）的获取，目前已有基于全球长周期分潮数值模拟手段的报道，但其面临的困境，与其它较弱分潮面临的困境没有差别。

从各分潮的调和常数获取的发展史来说，通过对已有观测结果进行插值曾经是首选，但发展过程中逐渐被数值模拟方法所取代。高度计资料的出现，引发部分学者开展了插值方法的研究，并取得了一些值得一提的结果，尽管被所谓的主流方式淹没，但也难掩其光芒所在。鉴于目前已有高度计资料作为支持，其它分潮及长周期分潮的调和常数获取的插值方法研究大有可为。

1.3 资料描述

1.3.1 地形数据

地形数据来自 ETOP5，全球的分辨率为 $5' \times 5'$ ，图 1.1 的区域是 $2^\circ \sim 25^\circ \text{N}$ ， $99^\circ \sim 122^\circ \text{E}$ 。

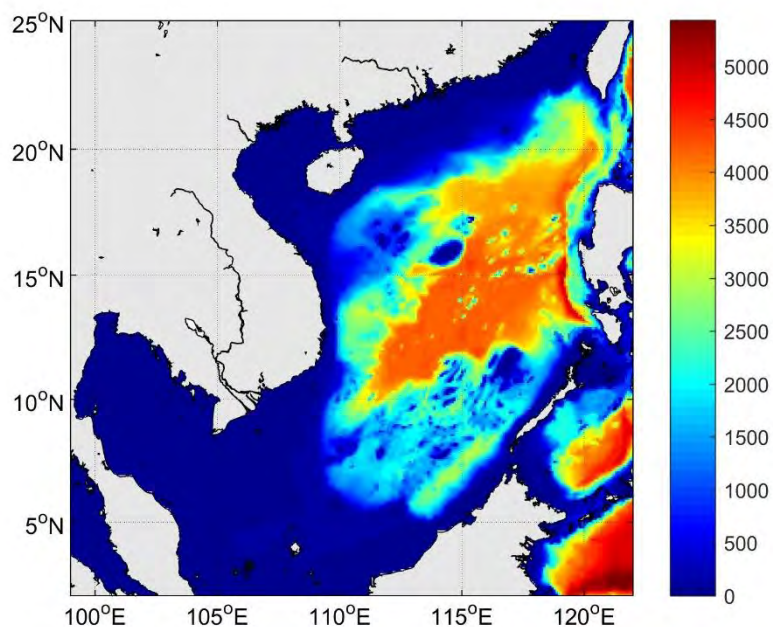


图 1.1 南海地形图

1.3.2 验潮站资料

中国近海及周边海域 770 个验潮点的资料, 和 56 个验潮点的资料 (是国际上公开的长期验潮站数据分析得到的调和常数), 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。

图 1.2 显示了上述资料点所在的位置, 从图中可以看出上述验潮点主要分布在近岸或岛屿附近。

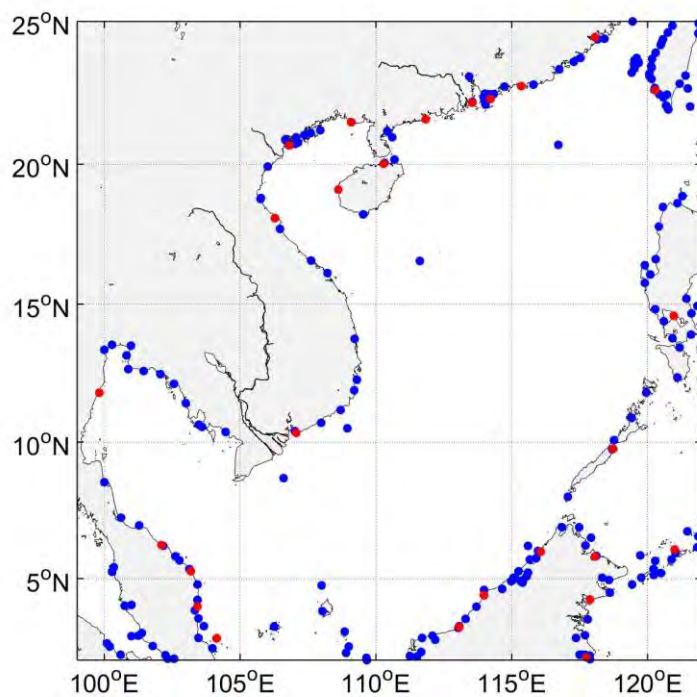


图 1.2 验潮站资料的分布图

1.3.3 TOPEX/POSEIDON 卫星高度计简介

卫星高度计是一种向卫星下方海洋发射脉冲的雷达,通过测量脉冲经海面反射之后的往返时间,获得卫星距海面的高度。主要用途:利用所得到的海面动力高度同化反演海洋重力场、流场、潮、大地水准面、海洋重力异常;根据回波强度获取风速资料;根据回波波形前沿斜率获取海面有效波高。

TOPEX/POSEIDON 卫星是由美国国家航空航天局和法国空间局联合于 1992 年 8 月 10 日发射的,是世界上第一颗专门为研究世界大洋环流而设计的高度计卫星。其轨道高度达 1336km,倾角为 66° ,覆盖面大,保证了资料的连续性。轨道的交点周期(绕地球一圈的时间)为 6745.8s,轨道运行 127 圈以后精确重复,轨道重复周期为 9.9156 天。相邻最近的轨道之间在赤道上的间隔为 $360^\circ/127=2.835^\circ$ 。卫星在一个周期内的每一圈分为上行轨和下行轨两条轨道,一个完整的周期内共有 254 条轨道,沿轨道的两个相邻的星下观测点的距离 5.75km。高度计系统的定规精度和测高精度较以前有显著提高,其测量精度约为 5cm,是目前观测海面高度精度最高的卫星。

当然,本文只是涉及到 TOPEX/POSEIDON 卫星高度计资料与潮汐相关的研究,即海面高度异常产品。

1.3.4 南海高度计资料

图 1.3 的给出了 $2^\circ \sim 25^\circ \text{N}$, $99^\circ \sim 122^\circ \text{E}$, TOPEX/POSEIDON 卫星高度计星下观测点所在的轨道。一共有超过 4000 个数据点,每个点都对应一个海面高度异常的时间序列,从 1992 年到 2017 年,时间跨度为 25 年。

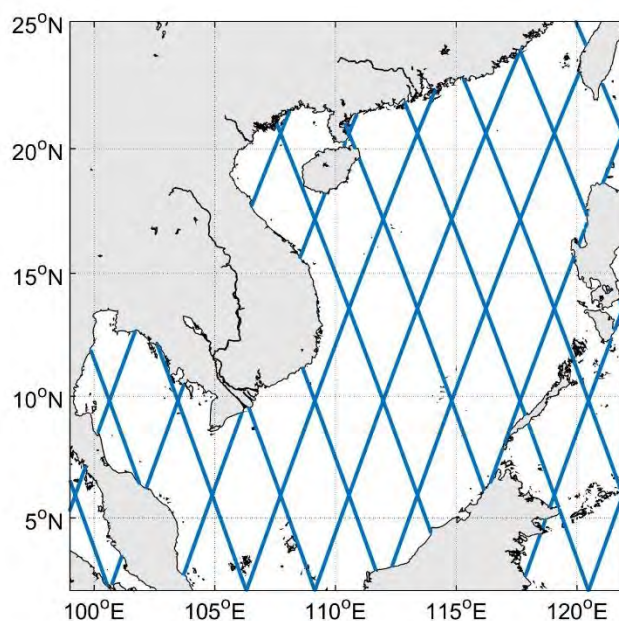


图 1.3 南海 TOPEX/POSEIDON 高度计资料的星下轨迹

1.4 要解决的问题

根据以上介绍及提供的数据文件及表格,请你们团队研究下列问题:

1. 根据沿轨道的星下观测点的海面高度异常值,提取所有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,注意能有效提取那些分潮的潮汐调和常数取决于相应的资料长度;对提取的潮汐调和常数,应利用潮汐验潮点的调和常数给予评价或检验,并给出评价结果的分析或评价。

2. 得到所有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, 沿轨道作图后, 可发现潮汐调和常数在沿轨道方向, 在空间有细结构, 而此细结构是内潮对正压潮的调制; 请设法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。
 3. 设计数据插值或拟合方法给出南海的各主要分潮的同潮图, 并利用潮汐验潮点的调和常数给予评价或检验, 并给出评价结果的分析或评价。
- 如果你们还有时间和兴趣, 还可考虑下列:
4. 如果在对沿轨道的潮汐调和常数分离、插值或拟合的过程中, 利用了特定的函数进行拟合, 是否能够确定出需利用的特定函数的最佳 (高) 次数? 上述结论是否对第 3 问有启示或帮助。

二、模型假设

假设一: 针对本文官方所给的数据均为真实有效的数据。

假设二: 经过误差修正后, 对于实际水位中调和分潮叠加部分, 仅为海平面的时间变化部分, 主要成分是海洋潮汐。

假设三: 对于水深大于 200 米区域的高度计资料都符合精度要求。

三、符号说明

符号	意义
M_2 、 S_2 、 N_2 、 K_2	半日分潮符号
K_1 、 O_1 、 P_1 、 Q_1	全日分潮符号
M_m 、 M_f 、 S_a 、 S_{sa}	长周期分潮符号
H	振幅
G	迟角
h	潮位/潮高
S_0	余水位
f_j	交点因子
u_j	交点订正角
h_j 、 g_j	分潮调和常数 (振幅和迟角)
$\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$	Doodson 数
Y, M, D, t	年, 月, 日, 时刻
S	平太阳黄经
h'	平太阴黄经
p	近地点黄经
N'	升交点平均黄经
P'	近日点平均黄经
σ	分潮的角速度
ν_0	天文初相角
J	分潮个数
ΔH	振幅差
Δ	调和常数的矢量差

四、问题分析

4.1 数据处理

4.1.1 对时间系统的处理

根据选取的时间系统，分潮相角可与地点发生联系或无关，相应的，迟角也根据时间系统的不同具有不同的意义，对应的有地方迟角(k)、区时专用迟角(g)、世界时迟角(G)之分。由于我国采用东八区时间系统，所以首先将格林尼治时间转换成东八区时间，相应的分潮迟角的变换关系为：

$$\begin{cases} g = k - \mu_1 + \delta N \\ G = g - \delta N \end{cases} \quad \text{公式 1}$$

式中， N 为区时号； μ_1 为分潮的第一个 Doodson 数； σ 为分潮角频率。

4.1.1 对采样时间的处理

由于 T/P 卫星高度计轨道的重复周期是 9.9156d，在此期间卫星绕地球旋转 127 圈[1]，根据 Nyquist 采样定理，采样的时间间隔大于信号的半周期时会产生频率混叠效应，在频位上形成虚假谱峰，造成高低频分量间的混乱。区分 M_2 和 S_2 分潮需要 2.97 年的数据，该时间是 4 个主要分潮中分辨两个分潮所需的最长时间，因此要获取 M_2 、 S_2 、 K_1 、 O_1 的稳定的调和数据至少需要 2.98 年的采样时间。为保证调和分析结果的可靠性，选取 14 年（2003 年初-2016 年末）的采样数据用于模拟工作

4.1.2 对观测点的处理

考虑到观测次数过少，会降低调和分析结果的可靠性，在调和分析的过程中选取观测次数在 300 次以上的观测点，图 4.1 为选取的观测次数大于 300 次时的 T/P 高度计资料的星下轨迹。

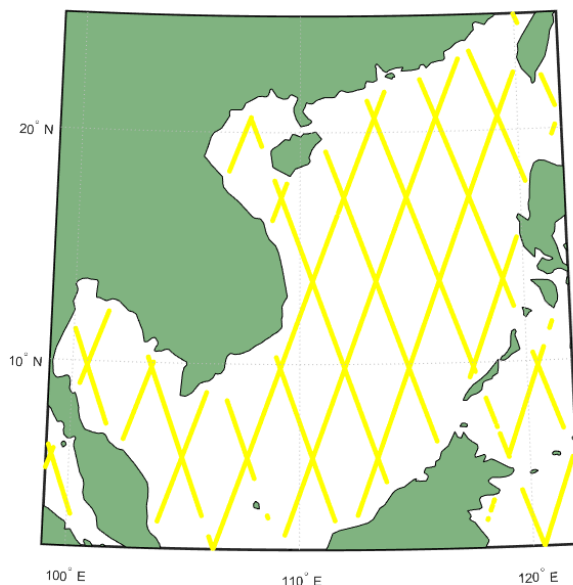


图 4.1 观测次数大于 300 次时的 T/P 高度计资料的星下轨迹

4.1.3 对水深数据的处理

由于高度计资料在浅水区精确度比较低,因此选取水深大于 200 米的数据用于数值模拟工作。图 4.2 的左右两张图分别为由原始水深数据以及筛选后水深大于 200m[2]得到的南海 3D 地形图,图 4.3 则为平面示意图。

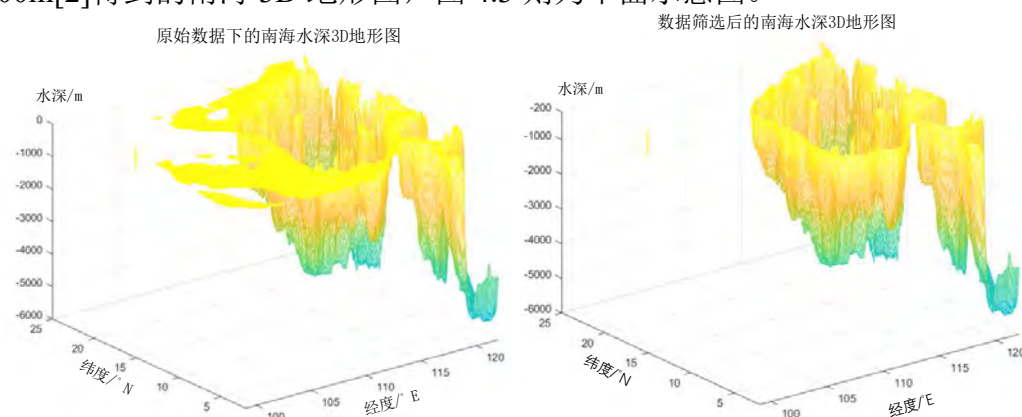


图 4.2 原始数据与筛选后的南海水深 3D 地形对比示意图

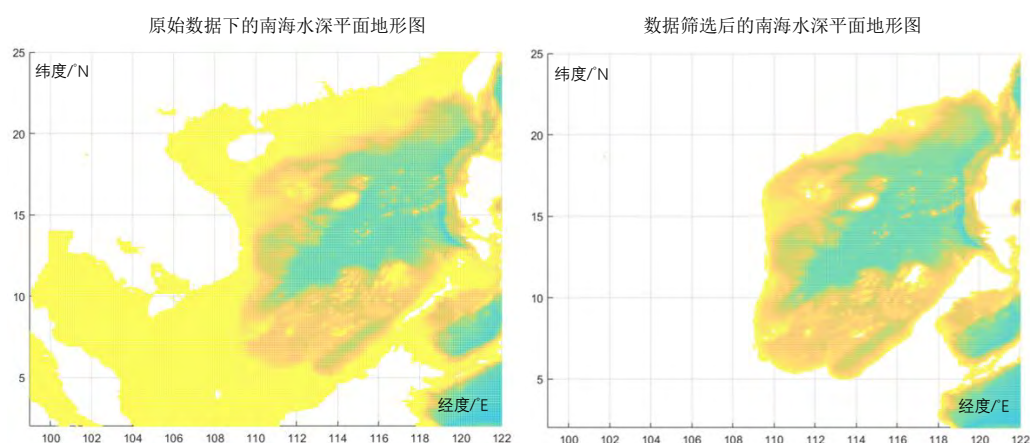


图 4.3 原始数据与筛选后的南海水深平面地形对比示意图

4.2 问题一的模型建立与求解

4.2.1 问题一的描述与分析

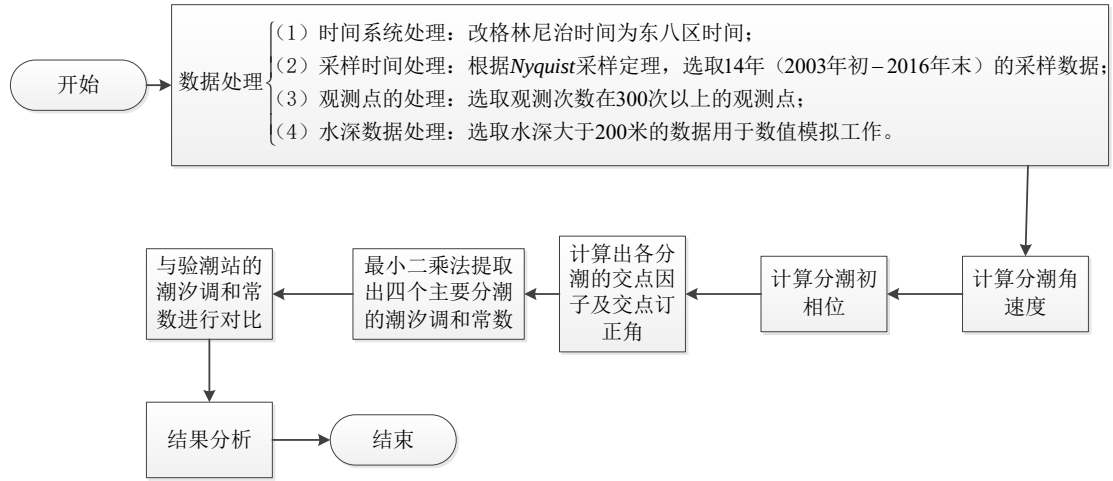
(1) 问题一的描述

根据沿轨道的星下观测点的海面高度异常值,提取所有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,注意能有效提取那些分潮的潮汐调和常数取决于相应的资料长度;对提取的潮汐调和常数,应利用潮汐验潮点的调和常数给予评价或检验,并给出评价结果的分析或评价。

(2) 问题一的分析

根据题中所给的观测点海面高度异常值数据,构建 j,v 模型进行潮汐的调和与分析。首先应计算出分潮角速度及初相位,其次计算出各分潮的交点因子及交点订正角,最后通过最小二乘法提取出四个主要分潮 (M_2 、 S_2 、 K_1 、 O_1) 的潮汐调和常数。

问题一分析的流程图如下:



4.2.2 问题一的模型建立

我们选取 J ($J=4$) 个分潮, 对于任一点的潮位表达式为:

$$h = S_0 + \sum_{j=1}^J f_j h_j \cos(v_j + u_j - g_j) = S_0 + \sum_{j=1}^J f_j h_j \cos(\sigma_j t + v_{0j} + u_j - g_j) \quad \text{公式 2}$$

其中, S_0 为余水位, f_j 为交点因子, u_j 为交点订正角, h_j, g_j 为分潮的调和常数 (振幅和迟角)。

(1) 计算分潮角速度

$$\sigma = \mu_1 \dot{\tau} + \mu_2 \dot{s} + \mu_3 \dot{h}' + \mu_4 \dot{p} + \mu_5 \dot{N}' + \mu_6 \dot{p}' \quad \text{公式 3}$$

其中: σ 为分潮的角速度, $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$ 为 Doodson 数,

$$\begin{cases} \dot{\tau} = 14.49205211 \\ \dot{s} = 0.54901653 \\ \dot{h}' = 0.04106864 \\ \dot{p} = 0.00464183 \\ \dot{N}' = 0.00220641 \\ \dot{p}' = 0.00000196 \end{cases} \quad (\text{单位: 度/平太阳时}) \quad \text{公式 4}$$

(2) 计算分潮初相位

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:

$$v_0 = \mu_1 \tau + \mu_2 s + \mu_3 h' + \mu_4 p + \mu_5 N' + \mu_6 p' + \mu_0 90 \quad \text{公式 5}$$

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

$$\left\{ \begin{array}{l} s = 277.02 + 129.3848(Y - 1900) + 13.1764(n + i + \frac{t}{24}) \\ h' = 280.19 - 0.2387(Y - 1900) + 0.9857(n + i + \frac{t}{24}) \\ p = 334.39 + 40.6625(Y - 1900) + 0.1114(n + i + \frac{t}{24}) \\ N' = 100.84 + 19.3282(Y - 1900) + 0.0530(n + i + \frac{t}{24}) \\ p' = 281.22 + 0.0172(Y - 1900) + 0.00005(n + i + \frac{t}{24}) \\ \tau = 15t - s + h' \end{array} \right. \quad \text{公式 6}$$

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)。

表 4.1 部分分潮的 Doodson 数、分潮角速度和交点因子与订正角

分潮符号	Doodson 数							分潮角速度 单位: 度/平太阳时	交点因子与订正角	
	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_0		f	u
S_a	0	0	1	0	0	0	0	0.0410686	1	0
S_{sa}	0	0	2	0	0	0	0	0.0821373	1	0
M_m	0	1	0	-1	0	0	0	0.5443747	M_m	M_m
MS_f	0	2	-2	0	0	0	0	1.0158958	M_2	$-M_2$
M_f	0	2	0	0	0	0	0	1.0980331	M_f	M_f
Q_1	1	-2	0	1	0	0	-1	13.3986609	O_1	O_1
O_1	1	-1	0	0	0	0	-1	13.9430356	O_1	O_1
M_1	1	0	0	0	0	0	1	14.4920521	M_1	M_1
P_1	1	1	-2	0	0	0	-1	14.9589314	P_1	P_1
S_1	1	1	-1	0	0	0	2	15.0000000	1	0
K_1	1	1	0	0	0	0	1	15.0410686	K_1	K_1
J_1	1	2	0	-1	0	0	1	15.5854434	J_1	J_1
OO_1	1	3	0	0	0	0	1	16.1391017	OO_1	OO_1
N_2	2	-1	0	1	0	0	0	28.4397295	M_2	M_2
M_2	2	0	0	0	0	0	0	28.9841042	M_2	M_2
L_2	2	1	0	-1	0	0	2	29.5284789	L_2	L_2

S_2	2	2	-2	0	0	0	0	30.0000000	1	0
k_2	2	2	0	0	0	0	0	30.0821373	k_2	k_2
M_4	4	0	0	0	0	0	0	57.9682085	M_2^2	$2M_2$
MS_4	4	2	-2	0	0	0	0	58.9841043	M_2	M_2
M_6	6	0	0	0	0	0	0	86.9523127	M_2^3	$3M_2$

表中交点因子及交点订正角的含义说明：例如，表中 M_6 分潮的交点因子是分潮 M_2 的交点因子的三次方， M_6 分潮的交点订正角是 M_2 分潮的交点订正角的三倍。

(3) 计算 f_j 、 u_j

由于 f_j 和 u_j 随时间变化非常缓慢，一般情况下取资料序列的中间时刻计算。各分潮的 f_j 、 u_j 的具体计算公式如下：

$$\begin{cases} f \cos u = \sum_{m=1}^M \rho_m \cos(\Delta\mu_4^m p + \Delta\mu_5^m N') \\ f \sin u = \sum_{m=1}^M \rho_m \sin(\Delta\mu_4^m p + \Delta\mu_5^m N') \end{cases} \quad \text{公式 7}$$

ρ_m 、 $\Delta\mu_4^m$ 、 $\Delta\mu_5^m$ 和 Doodson 数见表 4.2。

表 4.2 ρ_m 、 $\Delta\mu_4^m$ 、 $\Delta\mu_5^m$ 列表

分潮符号	Doodson 数							分潮角速度 单位：度/平太阳时	交点因子与订正角	
	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_0		f	u
S_a	0	0	1	0	0	0	0	0.0410686	1	0
S_{sa}	0	0	2	0	0	0	0	0.0821373	1	0
M_m	0	1	0	-1	0	0	0	0.5443747	M_m	M_m
MS_f	0	2	-2	0	0	0	0	1.0158958	M_2	$-M_2$
M_f	0	2	0	0	0	0	0	1.0980331	M_f	M_f
Q_1	1	-2	0	1	0	0	-1	13.3986609	O_1	O_1
O_1	1	-1	0	0	0	0	-1	13.9430356	O_1	O_1
M_1	1	0	0	0	0	0	1	14.4920521	M_1	M_1
P_1	1	1	-2	0	0	0	-1	14.9589314	P_1	P_1
S_1	1	1	-1	0	0	0	2	15.0000000	1	0
K_1	1	1	0	0	0	0	1	15.0410686	K_1	K_1
J_1	1	2	0	-1	0	0	1	15.5854434	J_1	J_1

OO_1	1	3	0	0	0	0	1	16.1391017	OO_1	OO_1
N_2	2	-1	0	1	0	0	0	28.4397295	M_2	M_2
M_2	2	0	0	0	0	0	0	28.9841042	M_2	M_2
L_2	2	1	0	-1	0	0	2	29.5284789	L_2	L_2
S_2	2	2	-2	0	0	0	0	30.0000000	1	0
k_2	2	2	0	0	0	0	0	30.0821373	k_2	k_2
M_4	4	0	0	0	0	0	0	57.9682085	M_2^2	$2M_2$
MS_4	4	2	-2	0	0	0	0	58.9841043	M_2	M_2
M_6	6	0	0	0	0	0	0	86.9523127	M_2^3	$3M_2$

对于 M_m 、 Mf 、 O_1 、 P_1 、 K_1 、 J_1 、 OO_1 、 M_2 、 L_2 、 k_2 分潮的 f 和 u 依照上式计算，其他分潮由这些分潮组合计算，但 M_1 分潮的 f 和 u 由以下公式计算得出：

$$f \cos u = -0.008 \cos(-p - 2N') + 0.094 \cos(-p - N') + 0.510 \cos p - 0.041 \cos(p - N') + 1.418 \cos p + 0.284 \cos(p + N') - 0.008 \cos(p + 2N') \quad \text{公式 8}$$

$$f \sin u = -0.008 \sin(-p - 2N') + 0.094 \sin(-p - N') - 0.510 \sin p - 0.041 \sin(p - N') + 1.418 \sin p + 0.284 \sin(p + N') - 0.008 \sin(p + 2N') \quad \text{公式 9}$$

(4) 最小二乘法提取分潮调和常数

在进行潮汐调和分析时，对某一确定的分潮：

$$f_j H_j \cos(\sigma_j t + V_{0j} + u_j - g_j) \quad \text{公式 10}$$

可化为如下形式：

$$f_j \cos(\sigma_j t + V_{0j} + u_j) H_j \cos g_j + f_j \sin(\sigma_j t + V_{0j} + u_j) H_j \sin g_j \quad \text{公式 11}$$

对应 J 个分潮，则有：

$$h = S_0 + \sum_{j=1}^J f_j \cos(\sigma_j t + V_{0j} + u_j) H_j \cos g_j + \sum_{j=1}^J f_j \sin(\sigma_j t + V_{0j} + u_j) H_j \sin g_j \quad \text{公式 12}$$

如果在 n 个时刻 $t = t_1, t_2, \dots, t_n$ ，有 n 个潮高观测值 $h = h_1, h_2, \dots, h_n$ ，那么，就可以建立如下由 n 个方程构成的方程组：

$$\begin{cases} S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_1 + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_1 + V_0 + u_j) = h_1 \\ S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_2 + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_2 + V_0 + u_j) = h_2 \\ \vdots \\ S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_n + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_n + V_0 + u_j) = h_n \end{cases} \quad \text{公式 13}$$

方程组中 x_j 对应 $H_j \cos g_j$ ， y_j 对应 $H_j \sin g_j$ ，它们和 S_0 共同构成了方程组中的全部未知量。潮汐调和分析的目的正是求出 x_j 与 y_j ，从而求出各个分潮的

调和常数 H 和 g 。

为了表示的方便，将方程组写成如下形式：

$$\begin{cases} S_0 + a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m = h_1 \\ S_0 + a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m = h_2 \\ \vdots \\ S_0 + a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m = h_n \end{cases} \quad \text{公式 14}$$

其中 $m=2J$ ，即所选分潮数的两倍， S_0 与 x_j 为待求解的未知数。

为了尽量减小噪声 r 对分析结果的影响，使调和常数尽可能接近真值，在实际潮汐分析中，总是希望使用更多的观测数据。因此，方程的数量 n 一般远大于未知数的数量 $m+1$ 。对于这样的矛盾方程组，采用最小二乘法来求解。

最小二乘法[5]的思想是，寻求一组解，使拟合值与实际值之差的平方和，即下式的值达到最小。

$$\Delta = \sum_{i=1}^n (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{im}x_m + S_0 - h_i)^2 \quad \text{公式 15}$$

根据多元函数微分的理论，这要求 Δ 相对于各未知数的偏导数均为 0，即：

$$\frac{\partial \Delta}{\partial S_0} = \frac{\partial \Delta}{\partial x_1} = \frac{\partial \Delta}{\partial x_2} = \cdots = \frac{\partial \Delta}{\partial x_m} = 0 \quad \text{公式 16}$$

从而得到关于分潮调和常数的线性方程组，进而可提取出分潮的调和常数。

4.2.3 问题一的模型求解与结果检验分析

(1) 确定 12 个验潮站的位置

由于研究范围为 $2^\circ \sim 25^\circ N$ ， $99^\circ \sim 122^\circ E$ ，首先应确定所选取对比验潮站的位置，如图 4.4 所示，选取中国南海海域周边距离高度计星下点不大于 120km[3] 的 12 个长期验潮站作为对比，所选出的验潮站用三角形表示。

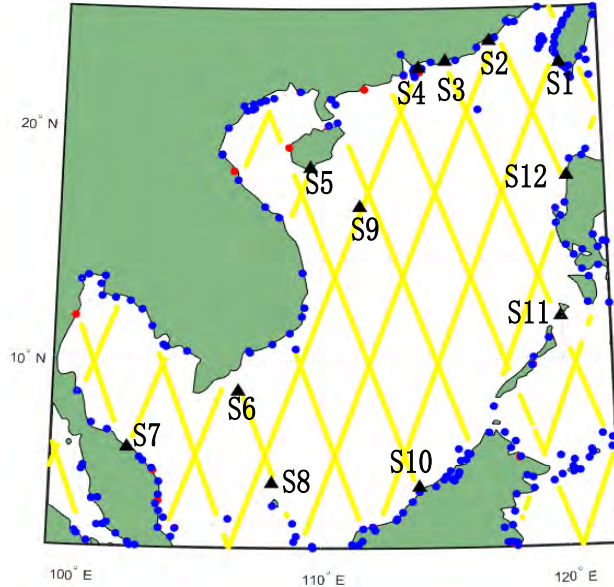


图 4.4 南海 T/P 高度计资料下的星下轨迹及所选验潮站示意图

如图所示，黄线为高度计资料的星下轨迹，蓝点与红点分别为中国近海周边及国际公开的验潮点位置。S1-S12 为选取的验潮点位置，其中 S3、S5、S7 为国际公开的长期验潮站。

(2) 确定 12 个验潮站处潮汐验潮点的调和常数

为方便对比提取的潮汐调与潮汐验潮点的调和常数，首先列出验潮站部分验潮点的调和常数，如表 4.3 所示

表 4.3 12 个验潮站处潮汐验潮点的调和常数

验潮站	位置 (° E/° N)	M ₂		S ₂		K ₁		O ₁	
		H	G	H	G	H	G	H	G
S1	120.29/22.61	0.174	219.43	0.068	351.21	0.170	166.11	0.161	240.08
S2	117.28/23.60	0.753	142.34	0.180	200.40	0.320	171.38	0.293	247.51
S3	115.35/22.75	0.277	255.26	0.110	277.56	0.326	176.45	0.262	138.12
S4	114.18/22.47	0.370	269.85	0.150	57	0.360	182.67	0.280	254
S5	109.53/18.22	0.203	303.4	0.168	95.2	0.285	189.27	0.287	271.5
S6	106.60/8/68	0.797	208.78	0.284	261.8	0.637	332.7	0.446	176.35
S7	102.11/6.23	0.171	254.80	0.081	52.66	0.255	231.55	0.130	188.61
S8	108.00/4.75	0.090	217.36	0.04	206	0.36	228.01	0.18	170.99
S9	111.62/16.55	0.168	69.36	0.077	112.9	0.268	305.44	0.234	149.49
S10	113.98/4.58	0.17	107.48	0.08	130	0.35	198.67	0.3	157.45
S11	119.95/11.8	0.237	303.43	0.101	124.1	0.298	198.0715	0.298	267.6
S12	120.42/17.78	0.077	228.36	0.032	241.2	0.191	196.5715	0.185	264.1

(3) 筛选出 12 个对应位置处的星下观测点的潮汐调和常数

在列出潮汐验潮点的调和常数之后，应对应位置处的星下观测点的潮汐调和常数筛选出来以方便对比，筛选出的调和常数如表 4.2 所示，其中在

表 4.4 所晒选的 12 个对应位置处的星下观测点的潮汐调和常数 (cm,°)

验潮站	序号	M ₂		S ₂		K ₁		O ₁	
		H	G	H	G	H	G	H	G
S1	1	0.2178	165.87	0.0835	347.12	0.1955	193.56	0.1626	214.59
	2	0.2236	164.47	0.0861	343.28	0.1957	192.02	0.1563	214.30
	3	0.2234	187.70	0.0903	347.15	0.1898	244.81	0.1564	284.91
	4	0.2276	256.95	0.0887	344.16	0.1926	205.18	0.1564	194.68
	5	0.2122	181.03	0.0901	344.01	0.1872	202.15	0.1563	319.76
	6	0.1938	181.64	0.0888	345.90	0.1918	199.98	0.1544	319.63
S2	1	0.9498	216.92	0.2431	179.60	0.3063	282.56	0.2207	114.42
	2	0.8573	193.82	0.1967	174.31	0.2946	149.87	0.2113	329.34
	3	0.8195	293.31	0.1946	175.00	0.2906	150.42	0.2115	330.98
	4	0.7797	192.54	0.1850	173.00	0.2916	152.60	0.2112	331.37
	5	0.7351	210.46	0.1704	172.27	0.2889	164.90	0.2112	336.43
	6	0.7017	191.89	0.1559	172.00	0.2905	208.47	0.2138	274.19
S3	1	0.1599	241.44	0.1104	243.81	0.2837	219.24	0.2120	251.26
	2	0.2032	228.48	0.1233	251.33	0.2888	224.18	0.2137	124.65
	3	0.1855	252.12	0.1126	231.80	0.2874	204.52	0.2162	271.53
	4	0.2032	128.48	0.1233	251.33	0.2888	224.18	0.2137	124.65
	5	0.2541	141.36	0.1425	258.63	0.2907	209.65	0.2181	145.25
	6	0.2858	259.86	0.1549	266.29	0.2920	218.52	0.2182	252.19
S4	1	0.2110	232.94	0.1950	36.48	0.3174	212.03	0.2403	123.78
	2	0.2142	149.39	0.1932	38.13	0.3201	222.86	0.2391	231.22
	3	0.2070	278.53	0.1203	35.95	0.3156	203.12	0.2323	117.46
	4	0.2069	161.93	0.1857	36.86	0.3127	193.07	0.2363	271.86
	5	0.3048	195.71	0.1329	15.30	0.2831	205.64	0.2092	271.71
	6	0.3212	215.95	0.1209	26.04	0.2809	220.46	0.2160	251.63

S5	1	0.1937	252.23	0.1669	118.09	0.2502	283.64	0.2869	176.75
	2	0.1887	135.41	0.1651	116.92	0.2454	273.63	0.2905	70.94
	3	0.1986	267.63	0.1663	119.77	0.2502	294.49	0.2977	283.07
	4	0.1980	273.35	0.1671	118.81	0.2513	278.86	0.1966	303.50
	5	0.1983	155.50	0.1675	116.17	0.2618	272.50	0.2122	198.31
	6	0.2017	356.63	0.1671	125.16	0.2639	271.19	0.2138	198.80
S6	1	0.6107	109.57	0.2242	244.00	0.5782	231.06	0.3704	189.65
	2	0.5908	312.40	0.2159	243.68	0.5747	324.18	0.3700	190.51
	3	0.5815	201.71	0.2163	242.40	0.5653	300.40	0.3670	104.05
	4	0.5565	328.84	0.2049	242.05	0.5662	279.72	0.3597	251.11
	5	0.6092	188.25	0.2878	240.44	0.5589	333.35	0.3600	321.36
	6	0.5419	310.30	0.2820	241.63	0.5490	323.86	0.3567	188.84
S7	1	0.1580	148.21	0.0744	46.38	0.2472	266.89	0.1220	304.33
	2	0.1619	263.81	0.0669	47.92	0.2421	276.16	0.1202	50.94
	3	0.1606	141.66	0.0652	49.82	0.2370	281.21	0.1167	282.47
	4	0.1548	126.49	0.0698	46.23	0.2361	273.21	0.1079	175.32
	5	0.1530	280.87	0.0647	50.68	0.2285	234.03	0.1039	114.65
	6	0.1513	270.75	0.0576	53.00	0.2265	262.10	0.0988	70.74
	7	0.1433	265.10	0.0635	42.59	0.2205	279.00	0.1015	50.62
S8	1	0.0786	209.98	0.0268	250.30	0.4449	288.53	0.1757	140.28
	2	0.0729	236.41	0.0274	244.98	0.4425	284.29	0.2720	268.07
	3	0.0827	104.10	0.0297	249.03	0.4489	307.14	0.1752	118.35
	4	0.0869	249.31	0.0256	247.88	0.4483	296.53	0.1761	270.07
	5	0.0907	206.42	0.0223	239.04	0.4493	287.20	0.2723	138.39
	6	0.0932	334.84	0.0249	246.76	0.4487	268.02	0.1739	283.99
	7	0.0913	223.34	0.0248	241.26	0.4436	242.37	0.2728	197.14
S9	1	0.1767	47.31	0.0576	105.15	0.2936	242.45	0.2413	97.03
	2	0.1794	174.22	0.0563	105.35	0.2930	222.84	0.2437	244.46
	3	0.1804	174.51	0.0568	103.13	0.2947	223.82	0.2421	144.79
	4	0.1733	48.24	0.0486	103.19	0.2903	303.09	0.2449	41.80
	5	0.1799	173.32	0.0504	101.23	0.3091	281.05	0.2326	187.34
	6	0.1659	40.09	0.0464	96.30	0.2653	319.72	0.2281	24.01
	7	0.1648	145.69	0.0504	91.80	0.2616	343.58	0.2218	110.97
S10	1	0.1867	77.62	0.0793	124.18	0.3729	174.49	0.2824	174.45
	2	0.1830	259.20	0.0979	128.00	0.3702	78.29	0.2761	189.03
	3	0.1526	126.56	0.0723	127.47	0.3652	182.26	0.2842	147.81
	4	0.1602	69.80	0.0697	125.97	0.3586	222.27	0.2878	113.11
	5	0.1609	180.41	0.0771	126.71	0.3570	231.02	0.2866	219.02
S11	1	0.2342	194.28	0.0865	83.80	0.3085	225.16	0.2323	268.41
	2	0.1947	308.27	0.0661	88.59	0.2995	250.16	0.2363	354.04
	3	0.2100	179.91	0.0925	85.59	0.3060	270.58	0.2392	207.35
	4	0.1916	294.51	0.0617	85.47	0.3017	280.80	0.2376	313.64
	5	0.2253	252.28	0.0920	92.44	0.3044	290.42	0.2371	59.70
	6	0.1909	295.44	0.0617	88.51	0.2987	281.46	0.2349	312.91
	7	0.2276	191.61	0.0898	91.03	0.3229	185.39	0.2282	302.32
S12	1	0.0950	199.54	0.0189	347.65	0.2082	231.68	0.1563	304.56
	2	0.0940	106.10	0.0453	358.47	0.2068	89.86	0.1535	353.27
	3	0.0827	196.30	0.0434	357.83	0.2020	106.55	0.1626	335.03
	4	0.0907	222.57	0.0417	280.39	0.2037	88.39	0.1639	120.71
	5	0.0833	103.78	0.0449	354.60	0.2010	91.91	0.1581	353.24
	6	0.0840	244.44	0.0399	251.93	0.2088	141.34	0.1640	180.05
	7	0.0757	106.80	0.0335	357.50	0.2174	161.68	0.1620	34.33

(4) 调和与分析结果与 12 个验潮站调和常数的对比

为了验证调和与分析结果的可靠性，我们将上述调和与分析结果与 12 个验潮站

的调和常数进行比较。用 ΔH [4] 表示交点处的振幅差, Δ 表示上升轨与下降轨在交点处调和常数的矢量差, RMS 表示单分潮预报误差 [2], 表达式为:

$$\Delta = [(H_0 \cos g_0 - H_m \cos g_m)^2 + (H_0 \sin g_0 - H_m \sin g_m)^2]^{\frac{1}{2}} \quad \text{公式 17}$$

$$RMS = \left\{ \frac{1}{N} \sum_{m=1}^N \frac{1}{2} [(H_0 \cos G_0 - H_m \cos G_m)^2 + (H_0 \sin G_0 - H_m \sin G_m)^2] \right\}^{\frac{1}{2}} \quad \text{公式 18}$$

H_0 、 H_m , G_0 、 G_m 分别表示验潮站结果及本文结果所对应的振幅与迟角。

表 4.5 星下观测点与潮汐验潮点各主要分潮的调和常数振幅与矢量差的对比
单位 ($\Delta H/\Delta G/\Delta/RMS - m^\circ/m/cm$)

	M2			S2			K1			O1		
Station	ΔH	Δg	Δ	ΔH	Δg	Δ	ΔH	Δg	Δ	ΔH	Δg	Δ
S1	0.0426	-2.983	0.0902	0.0196	-0.594	0.0257	0.0221	4.018	0.1520	-0.0040	1.790	0.3134
S2	0.0542	7.414	0.1528	0.0110	-2.604	0.1311	-0.0263	1.342	0.3411	-0.0797	3.861	0.4243
S3	-0.0622	-4.664	0.1518	0.0178	-2.704	0.2262	-0.0377	4.026	0.3505	-0.0476	5.679	0.0316
S4	-0.1258	-6.411	0.2193	0.0080	-2.554	0.0558	-0.0550	2.686	0.3239	-0.0511	-4.272	0.2725
S5	-0.0065	-6.327	0.4233	-0.0013	2.395	0.0665	-0.0312	8.978	0.5531	-0.0374	-6.627	0.4499
S6	-0.2152	3.306	1.3701	-0.0455	-1.943	0.0782	-0.0716	-3.394	0.4215	-0.0820	3.123	0.3737
S7	-0.0169	-4.096	0.0887	-0.0157	-0.458	0.0081	-0.0218	3.596	0.1517	-0.0205	-3.875	0.0521
S8	-0.0048	6.12	0.0128	-0.0141	3.960	0.0296	0.0866	5.400	0.8539	0.0368	3.133	0.2590
S9	0.0063	4.541	0.1767	-0.0247	-1.202	0.0695	0.0188	-2.879	0.1652	0.0023	-2.801	0.4342
S10	-0.0014	2.301	0.0627	-0.0011	-5.79	0.1398	-0.0013	0.29	0.1256	-0.0248	8.03	0.0426
S11	-0.0264	-5.825	0.4732	-0.0224	-3.618	0.1892	0.0080	5.678	0.3226	-0.0629	-7.83	0.0776
S12	0.0095	-5.986	0.0573	0.0062	8.857	0.0628	0.0158	-6.637	0.0571	-0.0249	-2.393	0.1868
mean	0.0276	6.538	0.2731	0.0156	2.022	0.0192	0.0233	4.055	0.3182	0.0295	3.262	0.2431
RMS		1.4147			0.7382			0.8767			0.4983	

注: 平均值 mean 一行的数据为各列数据的绝对值的平均数据。

由表 4.5 我们可以看到四个分潮的预报中误差 RMS 均小于 1.5cm, 最大值和最小值分别为 M_2 潮的 1.4147cm, O_1 潮的 0.4983cm; ΔH 平均小于 2.95cm, ΔG 平均小于 45° 。上述一系列比较证实本文调和分析的可靠性。

4.3 问题二的模型建立与求解

4.3.1 问题二的描述与分析

(1) 问题二的描述

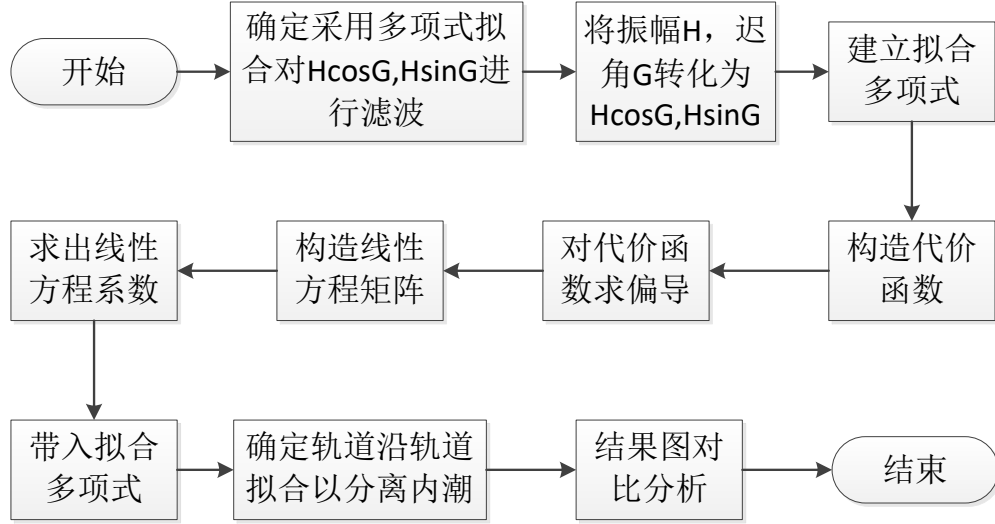
得到所有星下观测点各主要分潮 (M_2 、 S_2 、 K_1 、 O_1) 的潮汐调和常数, 沿轨道作图后, 可发现潮汐调和常数在沿轨道方向, 在空间有细结构, 而此细结构是内潮对正压潮的调制; 请设法对沿轨道的各分潮的潮汐调和常数进行正压潮和内潮的分离。

(2) 问题二的分析

在问题一所得到的结果中, 既含有表面潮 (正压潮) 的信息, 同时又包含同频率斜压潮 (内潮) 的成分。由于同频率的表面潮的波长比内潮的波长要大的多, 内潮的波长通常为几十公里左右, 很少有超过 400 公里的, 把波长大于 400 公里的部分滤除, 滤除的是真正的正压部分。多项式拟合法[8]是用 1 个多项式去拟合 1 组数据, 拟合的结果不会受到内潮波长变化的影响。前人通过多项式拟合以及平均滑动的方法进行滤波, 将正压潮与内潮分离。已经证实对 $H \cos G$ 、 $H \sin G$ 进行滤波的结果要好于对 H 、 G 进行滤波的结果, (其中 H 、 G

分别代表海表面振幅与迟角)。基于 Cummins[6] 以前的工作, 本文分别对 $H \cos G$ 、 $H \sin G$ 沿 T/P 轨进行多项式拟合[7], 通过该调和和分析得到研究区域内轨道上任一观测点 $P(x, y)$ (x, y 分别为 p 点的经纬度) 的振幅值 H 和迟角值 G 。大致步骤为首先将振幅 H 、迟角 G 转化为 $H \cos G$ 、 $H \sin G$; 然后分别对 $H \cos G$ 、 $H \sin G$ 沿轨进行多项式拟合; 最后通过坐标转换得到轨道上任意点的调和常数, 实现将正压潮与内潮分离。

问题二分析的流程图如下:



4.3.2 问题二的模型建立

设拟合多项式可表示为:

$$\hat{f}(y) = a_0 + a_1 y + a_2 y^2 + \dots + a_n y^n \quad \text{公式 19}$$

y 为轨道上任意点的纬度值, 构造代价函数:

$$J(a_n) = \sum_{k=1}^N [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - H_k \cos G_k]^2 \quad \text{公式 20}$$

其中 y_k ($k=1, 2, 3, \dots, N$) 代表轨道上观测点的纬度值, 上式分别对 $a_0, a_1, a_2, \dots, a_n$ 求偏导[8], 并使其为零, 可得:

$$\begin{cases} \sum_{k=1}^N [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - H_k \cos G_k] = 0 \\ \sum_{k=1}^N y_k [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - H_k \cos G_k] = 0 \\ \sum_{k=1}^N y_k^2 [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - H_k \cos G_k] = 0 \\ \dots\dots\dots \\ \sum_{k=1}^N y_k^n [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - H_k \cos G_k] = 0 \end{cases} \quad \text{公式 19}$$

为便于阐述, 引入记号:

$$M_{i,j} = \sum_{k=1}^N y_k^{i+j-2}, b = \sum_{k=1}^N y_k^{i-1} H_k \cos G_k \quad \text{公式 20}$$

其中, $i=1,2,3,\dots,n+1; j=1,2,3,\dots,n+1$ 。

表示为线性方程矩阵可得:

$$\begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} & \dots & M_{1,n+1} \\ M_{2,1} & M_{2,2} & M_{2,3} & \dots & M_{2,n+1} \\ M_{3,1} & M_{3,2} & M_{3,3} & \dots & M_{3,n+1} \\ \dots & \dots & \dots & \dots & \dots \\ M_{n+1,1} & M_{n+1,2} & M_{n+1,3} & \dots & M_{n+1,n+1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_{n+1} \end{bmatrix} \quad \text{公式 21}$$

通过求解上述线性方程组可得系数 $a_0, a_1, a_2, \dots, a_n$ 的值, 带入拟合多项式中, 可得:

$$\hat{f}(y) = a_0 + a_1 y + a_2 y^2 + \dots + a_n y^n \quad \text{公式 22}$$

同理, 对 $H \sin G$ 进行多项式拟合可得:

$$\hat{g}(y) = a_0 + a_1 y + a_2 y^2 + \dots + a_n y^n \quad \text{公式 23}$$

4.3.3 问题二的模型求解

在得到所有星下观测点各主要分潮的潮汐调和常数前提下, 沿轨道作图之前首先选取两条轨道, 这里选取两条交叉的上行轨与下行轨 (轨道 1、轨道 2) 进行示意, 选取的两条轨道在星轨中的位置如图 4.5 所示。

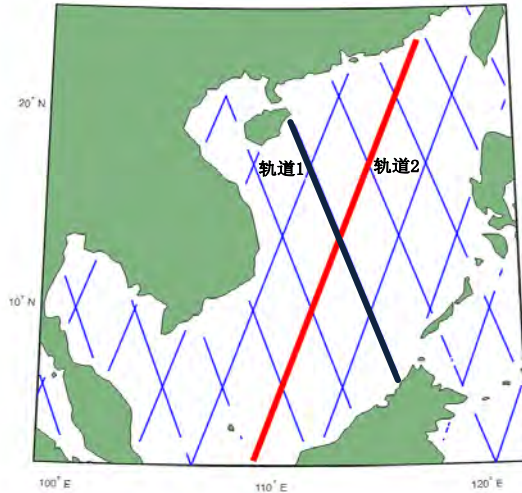


图 4.5 选取的上行轨、下行轨示意图

由于轨道在纬度上变化太小, 迟角和幅度在分辨率高的情况下波动很大, 当我们选取 4 次以上的多项式插值时插值矩阵接近奇异值, 在这里我们选取 3 次插值拟合。依照上述方法我们对 M_2 、 S_2 、 K_1 、 O_1 各分潮的 $H \cos G$ 、 $H \sin G$ 分别沿区域内的两条轨道进行 3 次多项式拟合, 以 M_2 潮为例, 得到轨道 1、2 各分潮的拟合多项式的系数如表 4.6 所示。

表 4.6 轨道 1、2 各分潮的拟合多项式系数

Hcos(G)	a0	a1	a2	a3	a4	a5
轨道 1	-142127.98	1994.99	13.56	-0.1839	-0.0014	8.56E-06
轨道 2	471839.81	-6875.22	-43.08	0.6392	0.0046	-3.12E-05
Hsin(G)	a0	a1	a2	a3	a4	a5
轨道 1	116995.06	-1601.21	-11.4543	0.1466	0.0011	-6.59E-06
轨道 2	131583.450	-1822.65	-12.6559	0.1665	0.0013	-7.66E-06

4.3.4 问题二模型结果检验分析

(1) 对沿轨道 1 的数据进行拟合

为验证所构建的多项式拟合模型的有效性，图 4.6 给出了对于轨道 1 的 M_2 分潮 3 次多项式拟合结果与 $H \cos G$ 、 $H \sin G$ 的对比图。

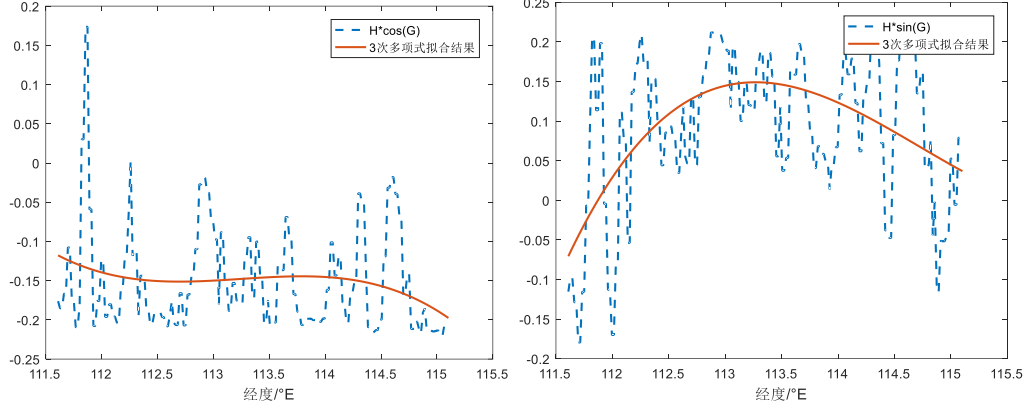


图 4.6 M_2 分潮拟合结果与 $H \cos G$ 、 $H \sin G$ 的对比图

由图 4.6 可知由于轨道在纬度上变化太小，在分辨率高的情况下拟合有一定程度的波动，但是所拟合曲线的大致趋势是吻合的。

以下为沿轨道 1 进行 3 次多项式拟合来提取 M_2 、 S_2 、 K_1 、 O_1 分潮海表面信号，并给出了拟合结果与海表面分潮振幅及迟角的对比，如图 4.7、4.8、4.9、4.10 所示。

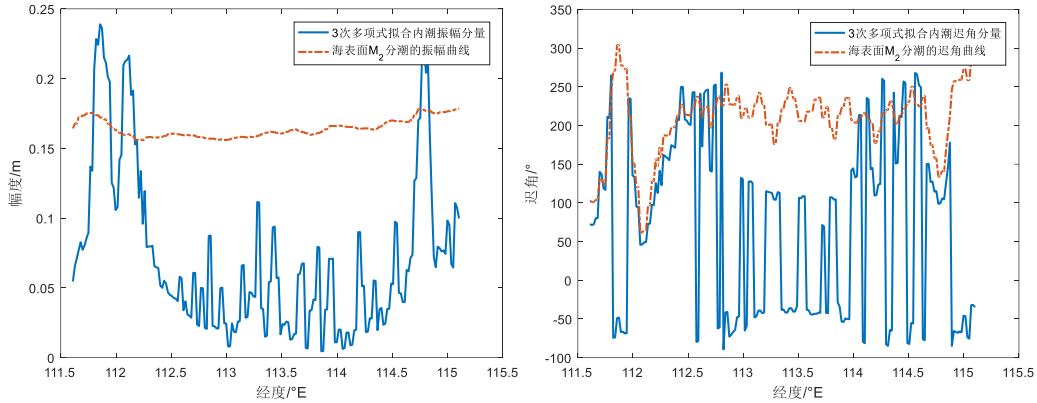


图 4.7 海表面 M_2 分潮与拟合结果的振幅及迟角对比图

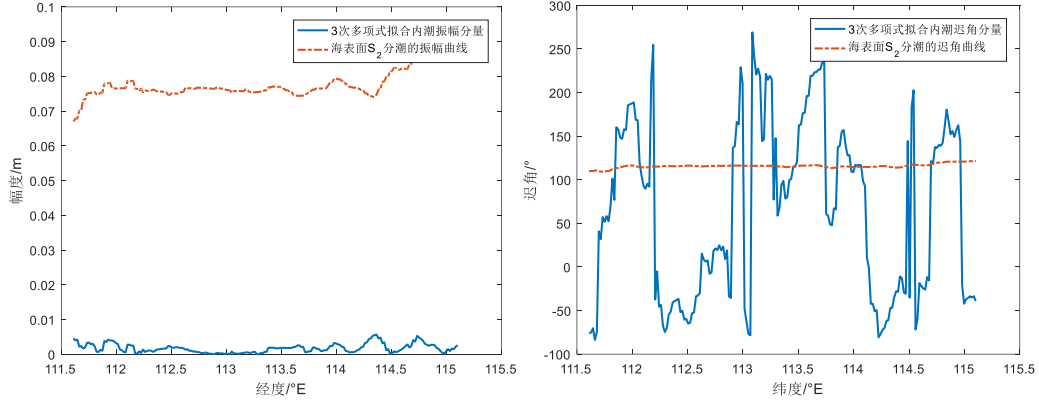


图 4.8 海表面 S_2 分潮与拟合结果的振幅及迟角对比图

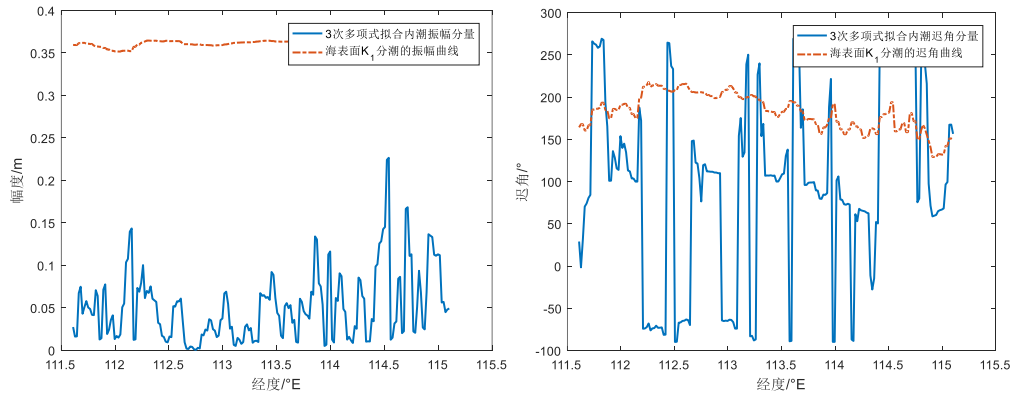


图 4.9 海表面 K_1 分潮与拟合结果的振幅及迟角对比图

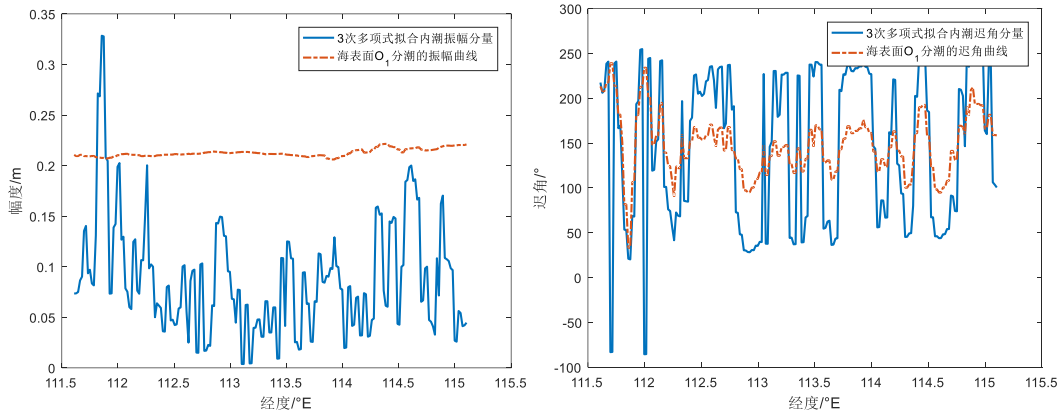


图 4.10 海表面 O_1 分潮与拟合结果的振幅及迟角对比图

由图 4.7、4.8、4.9、4.10 可知对沿轨道 1 的数据进行拟合分析时，对 M_2 、 S_2 、 K_1 、 O_1 分潮振幅拟合效果并不十分明显，这是由于轨道 1 的斜率模值较大，轨道较为陡峭，在对该轨道进行拟合分析时，所跨越的经度较小（仅有 4° ），分辨率较高，波动较大；此外，在经度跨度较小的情况下容易出现观测点缺失的情况，因此造成拟合效果并非十分显著，但是对于几个分潮的迟角拟合效果可靠性较高。

(2) 对轨道 2 的数据进行拟合

以下为沿轨道 2 进行 3 次多项式拟合来提取 M_2 、 S_2 、 K_1 、 O_1 分潮海表面信号，并给出了拟合结果与海表面分潮振幅及迟角的对比，如图 4.11、4.12、4.13、4.14 所示。

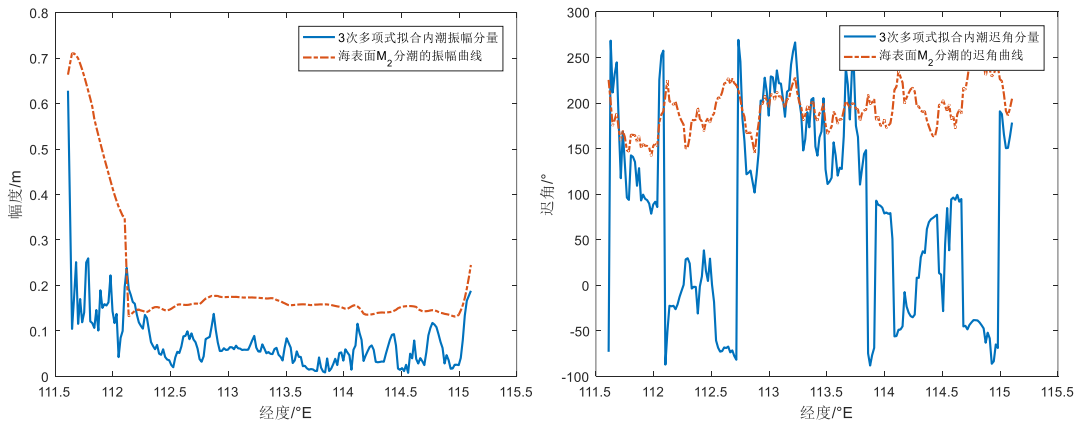


图 4.11 海表面 M_2 分潮与拟合结果的振幅及迟角对比图

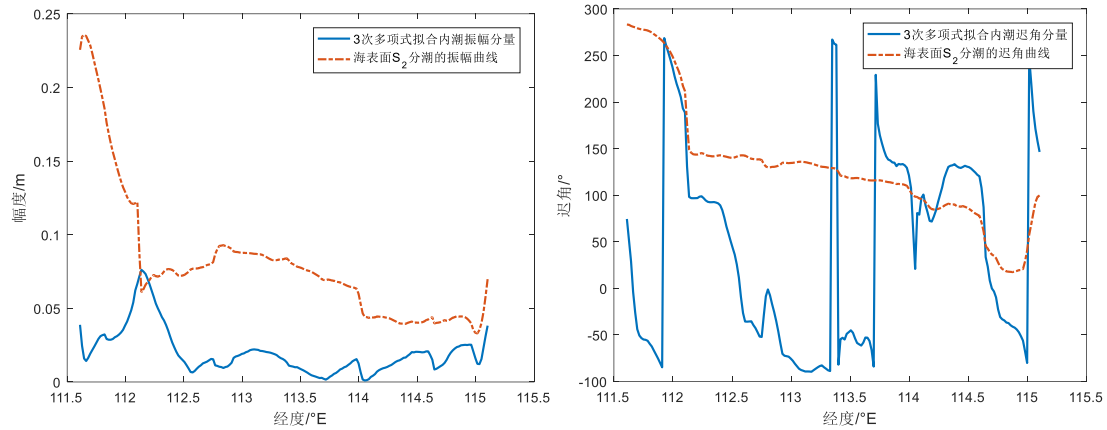


图 4.12 海表面 S_2 分潮与拟合结果的振幅及迟角对比图

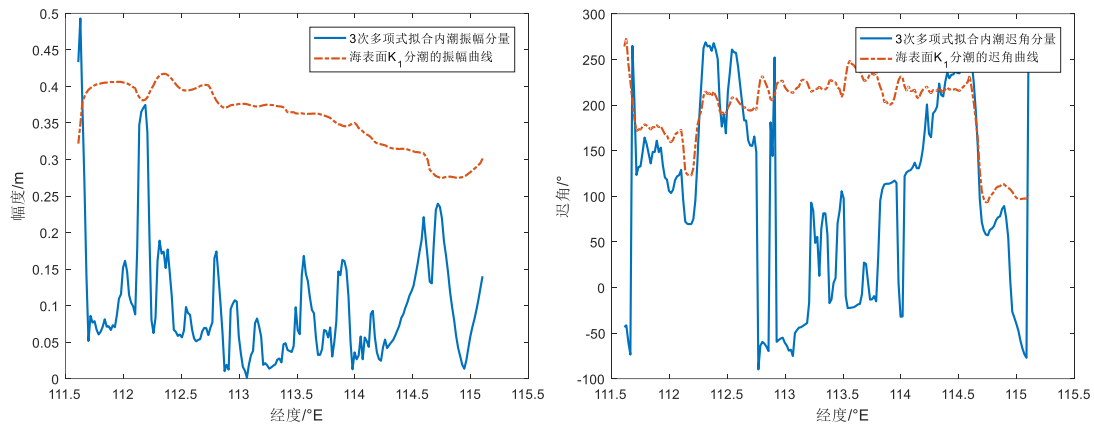


图 4.13 海表面 K_1 分潮与拟合结果的振幅及迟角对比图

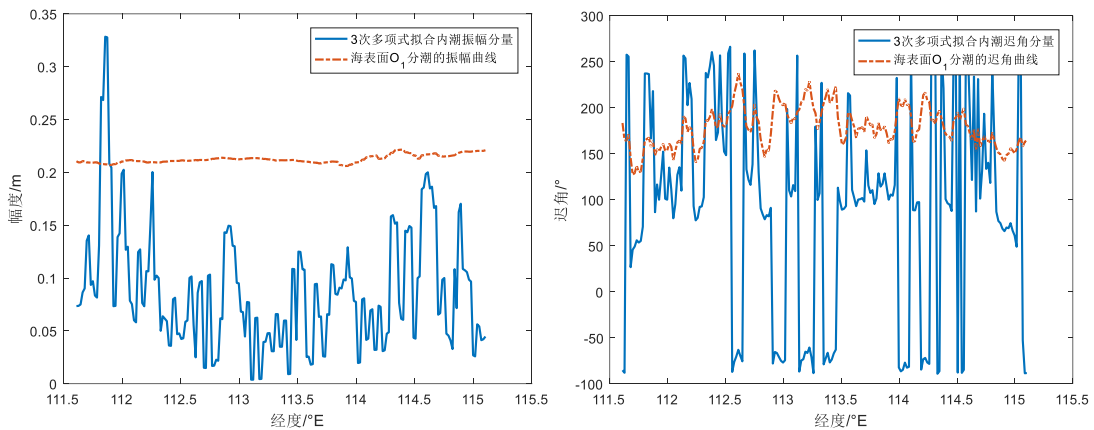


图 4.14 海表面 O_1 分潮与拟合结果的振幅及迟角对比图

由图 4.11、4.12、4.13、4.14 可知对沿轨道 2 的数据进行拟合分析时，对各分潮的拟合效果除了 K_1 、 O_1 分潮的振幅稍有偏差，对于其他分潮的振幅及迟角的拟合重合度较高。

综合两条轨道的拟合结果可知，所构建的正压潮与内潮分离模型，拟合效果具有较高的可靠性。

4.4 问题三的模型建立与求解

4.4.1 问题三的描述与分析

(1) 问题三的描述

设计数据插值或拟合方法给出南海的各主要分潮的同潮图，并利用潮汐验潮点的调和常数给予评价或检验，并给出评价结果的分析或评价。

(2) 问题三的分析

在问题一中通过 j,v 模型, 利用卫星高度计资料获得 T/P 卫星的星点下 4 个主要分潮的调和常数, 通过插值可得出南海区域 M_2 、 S_2 、 K_1 、 O_1 分潮的同潮图。

4.4.2 问题三的建模与求解

针对此问题我们需要对整个南海所有星下点的数据进行各个调和常数的提取, 我们采取的方案是以正方形小方块的形式对南海图进行遍历分析:

首先, 我们选取一个起始的“中心点”, 以这个点为中心作一个正方形范围的搜索。我们遍历在这个范围内的所有满足要求的星下观测点 (观测次数大于 300), 然后逐点进行调和常数, 得到的调和常数取所有点的平均值, 记录此中心点的经纬度和平均调和常数。当一个点结束后, 我们移动“中心点”到下一个位置, 重复此操作, 如下图所示, 直到遍历完所有的点。

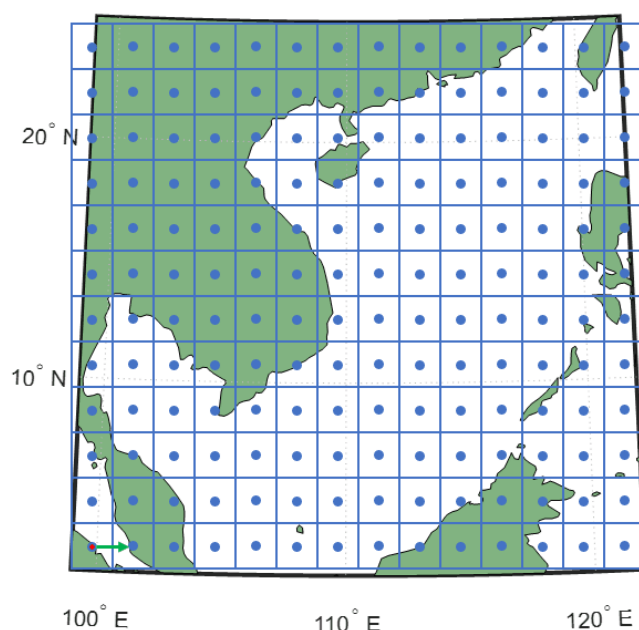
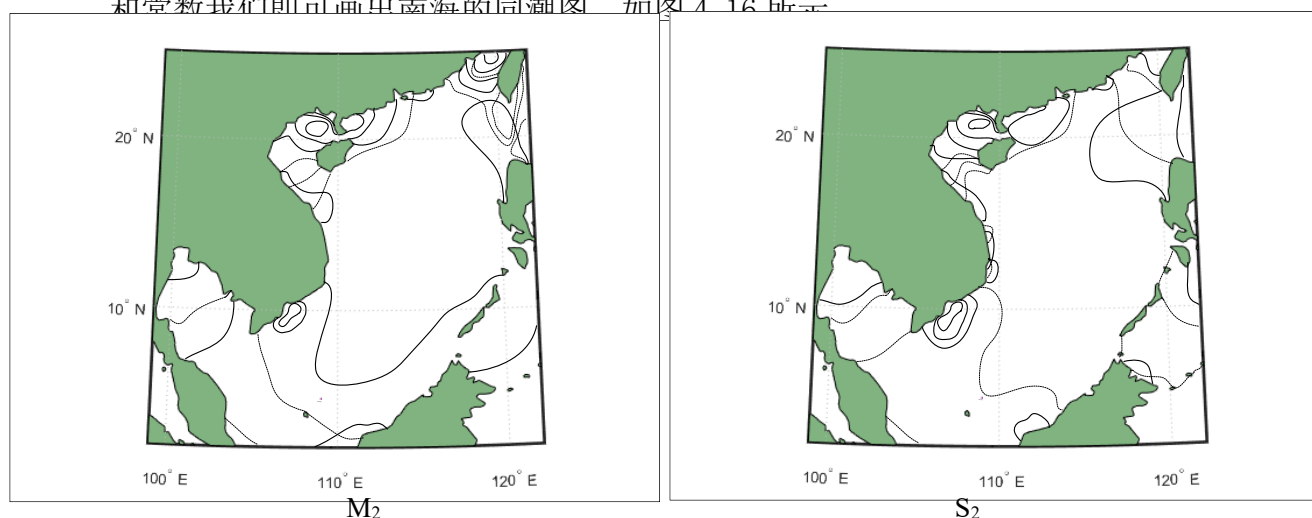


图 4.15 遍历南海分布图

当遍历完成所有的“中心点”后, 即可得到了整个南海的调和常数, 利用调和常数我们即可画出南海的同潮图。如图 4.16 所示。



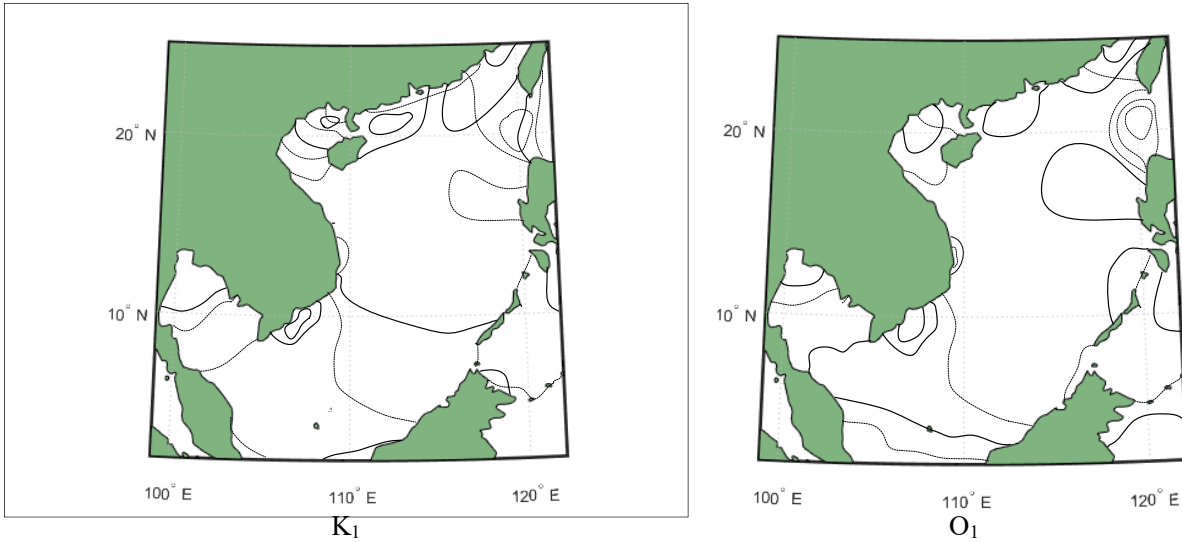


图 4.16 南海各主要分潮同潮图

4.4.3 问题三结果评价分析

(1) 与验潮站资料的比较

本文利用差值拟合的方法求得了卫星轨道间任一点的分潮，为了验证该方法提取宽阔海域信息的可靠性，我们选取问题一中 12 个长期验潮站的分潮结果，将模型结果与验潮站结果四个主要分潮进行比较，比较结果见表 4.7。

表 4.7 利用拟合得到的分潮的调和常数与验潮站资料的比较($\Delta H(m)$, $\Delta G(^{\circ})$)

Station	$^{\circ}E/^{\circ}N$	M_2		S_2		K_1		O_1	
		ΔH	ΔG	ΔH	ΔG	ΔH	ΔG	ΔH	ΔG
S1	120.29/22.61	0.013	3	0.012	9	0.067	12	0.020	1
S2	117.28/23.60	0.022	2	0.053	7	0.012	7	0.089	10
S3	115.35/22.75	0.011	6	0.016	2	0.038	9	0.047	9
S4	114.18/22.47	0.032	6	0.039	4	0.033	4	0.025	5
S5	109.53/18.22	0.019	9	0.075	3	0.032	6	0.045	1
S6	106.60/8/68	0.023	10	0.044	7	0.020	6	0.033	3
S7	102.11/6.23	0.053	5	0.031	4	0.051	3	0.017	9
S8	108.00/4.75	0.034	3	0.021	9	0.013	9	0.024	4
S9	111.62/16.55	0.013	2	0.011	2	0.046	0	0.049	8
S10	113.98/4.58	0.048	3	0.048	4	0.018	11	0.032	3
S11	119.95/11.8	0.036	8	0.034	5	0.023	0	0.014	9
S12	120.42/17.78	0.019	5	0.029	1	0.058	2	0.025	8

从表 4.7 可以看出本文所提出的模型计算结果与验潮站实测结果差别较小，进一步分析可知， K_1 分潮振幅平均误差为 3.42cm，迟角绝对平均误差为 5.75° ，其向量均方根误差为 5.71cm； O_1 分潮振幅平均误差为 3.5cm，迟角绝对平均误差为 5.83° ，其向量均方根误差为 4.6cm； M_2 分潮振幅平均误差为 2.6cm，迟角绝对平均误差为 5.16° ，其向量均方根误差为 3.2cm； S_2 分潮振幅平均误差为 3.4cm，迟角绝对平均误差为 4.75° ，其向量均方根误差为 4.61cm。由于验潮站多在近海区域，近海区域地形变化较为复杂，模型计算结果和验潮计结果存在一定差异。经过以上比较，向量均方根误差均小于 6cm，最大振幅差与迟角差均在可接受范围，模型得出的调和常数与验潮站资料显示出较好的一致性,从而表明本文拟合方法具有一定的可行性。

(2) 南海海潮模型的评价

中国南海是热带大陆边缘海之一，海岸线曲折，岛屿众多，导致其潮型分布较为复杂。针对问题三，本文利用南海约 14 年的 T/P 测高数据，基于调和分析

法，并考虑了不同海域卫星测高数据精度的差异，采用多项式拟合的方法对轨道间的点进行调和分析，构建了南海潮汐模型。利用该潮汐模型分析南海主要分潮同潮图。如下图所示。

由图 4.16 可知，南海潮型以全日潮 K_1 和 O_1 为主，四个分潮在深海区传播速度较快，等值线稀疏。相对于全日潮，南海大部分海区的半日潮影响较小，半日潮 M_2 在台湾海峡及海南岛附近有较大振幅。 S_2 分潮在湄公河入海口处振幅较大，超过 30cm，在南海整体振幅较小。这与多数文献中南海潮波主要由太平洋由吕宋海峡传入南海向西南方向前进，部分半日潮向台湾海峡前进与其他潮波系统发生作用的结论基本一致。

4.5 问题四的模型建立与求解

4.5.1 问题四的描述与分析

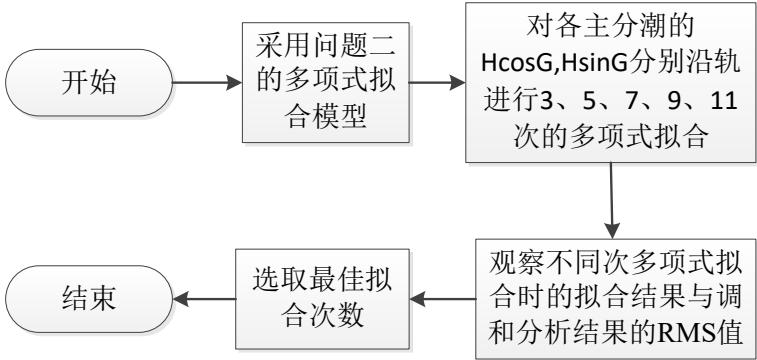
(1) 问题四的描述

如果在对沿轨道的潮汐调和常数分离、插值或拟合的过程中，利用了特定的函数进行拟合，是否能够确定出需利用的特定函数的最佳（高）次数？上述结论是否对第 3 问有启示或帮助。

(2) 问题四的分析

对于问题四，可以在问题二的基础上进行分析，对 M_2 、 S_2 、 K_1 、 O_1 各分潮的 $H \cos G$ 、 $H \sin G$ 分别沿轨进行 3、5、7、9、11 次的多项式拟合，观察不同次多项式拟合时的拟合结果与调和分析结果的 RMS 值，看其是否有最佳值。

问题四分析的流程图如下：



4.5.2 问题四的求解与启示

依照问题二中的方法，我们对 M_2 、 S_2 、 K_1 、 O_1 的 $H \cos G$ 、 $H \sin G$ 分别沿区域内每一条轨迹进行 3、5、7、9、11 次多项式拟合，下表给出了各分潮沿轨道 2 进行 3、5、7、9、11 次多项式拟合结果与调和分析结果的平均误差值。

表 4.7 对 M_2 分潮不同次多项式拟合结果与调和分析结果的平均误差表

M_2	3 阶	5 阶	7 阶	9 阶	11 阶
幅度 (m)	0.1006	0.1006	0.0997	0.1010	0.1020
迟角 (°)	113.7639	113.7610	101.3320	110.8436	104.9140

表 4.8 对 S_2 分潮不同次多项式拟合结果与调和分析结果的平均误差表

S_2	3 阶	5 阶	7 阶	9 阶	11 阶
幅度 (m)	0.1012	0.1033	0.0893	0.1132	0.1102
迟角 (°)	112.7334	109.6100	91.3410	103.6381	101.0094

表 4.9 对 K_1 分潮不同次多项式拟合结果与调和分析结果的平均误差表

K1	3 阶	5 阶	7 阶	9 阶	11 阶
幅度 (m)	0.1022	0.1007	0.0627	0.1023	0.1020
迟角 (°)	103.3123	111.6100	100.3124	110.8077	104.7205

表 4.10 对 O_1 分潮不同次多项式拟合结果与调和分析结果的平均误差表

O1	3 阶	5 阶	7 阶	9 阶	11 阶
幅度 (m)	0.1103	0.1023	0.0633	0.1089	0.1220
迟角 (°)	104.8476	103.6661	99.2310	113.6536	106.9740

由于各分潮沿所有轨道进行 3、5、7、9、11 多项式拟合结果对比数据较大，这里仅沿轨道 2 做出了拟合结果与调和分析结果的平均误差表格。分析可知，虽然 3-11 次的多项式拟合结果均可反映出海表面 M_2 、 S_2 、 N_2 、 K_2 各分潮的振幅变化趋势，但不同次数多项式拟合结果与海表面各分潮振幅及迟角的符合程度存在明显差异。3 次多项式的拟合结果明显偏离调和分析得到的振幅值，不能正确地拟合出正压潮值；5 次的多项式拟合结果在某些位置同样存在偏离现象，即拟合结果不能准确的表示正压潮值。随着拟合次数的增加，拟合效果趋于好转，但由上表可知，并不是拟合次数越高拟合效果就越好。采用 9、11 次多项式拟合的振幅平均误差甚至上升，另一方面，采用 7 次多项式拟合的迟角平均误差最小。综合考虑振幅与迟角的平均误差，得到 7 次为最佳的多项式拟合次数。这对第二问的多项式拟合模型有了一定的指导意义。

五、模型的评价与推广

5.1 模型的优点

问题一根据海面高度异常值构建了 j,v 模型来提取星下观测点主要分潮的潮汐调和常数。首先为保证调和分析结果的可靠性，根据 Nyquist 采样定理确定采用时间，并由观测次数的大小来选取观测点，大大提高了调和分析结果的准确性及提取效率。其次通过计算分潮角速度及初相位、各分潮的交点因子及交点订正角，并通过最小二乘法提取出四个主要分潮 (M_2 、 S_2 、 K_1 、 O_1) 的潮汐调和常数。最后通过与 12 个验潮站的调和常数进行了比较，证实了该模型的有效性与可靠性，且该模型思路清晰，运算简便。

问题二构建了正压潮与内潮分离模型。首先将振幅 H 、迟角 G 转化为 $H \cos G$ 、 $H \sin G$ ；然后分别对 $H \cos G$ 、 $H \sin G$ 沿轨进行多项式拟合；最后通过坐标转换得到轨道上任意点的调和常数，进而将正压潮与内潮进行分离。通过对比各主分潮的多项式拟合结果与调和分析结果的振幅与迟角，证明了所建模型的可靠性。

问题三我们提出了一种新的方法提取整个南海的调和常数，首先选取一个初始的“中心点”，以该点为中心作小正方形范围内的搜索，对该范围内所有满足要求的星下观测点进行调和分析，得到该范围内所有点的调和常数并取其平均值，记录该“中心点”的经纬度和平均调和常数。当该区域结束后移动“中心点”到下一个位置，重复该操作直到遍历完所有的“中心点”。根据所得的调和常数做出了南海各主要分潮的同潮图，这种方法做出来的同潮图与预期一致。

问题四在问题二的基础上，进一步考量了拟合过程中所需特定函数的最佳次

数。对 M_2 、 S_2 、 K_1 、 O_1 各分潮的 $H \cos G$ 、 $H \sin G$ 分别沿轨进行 3-15 次的多项式拟合, 观察不同次多项式拟合时的拟合结果与调和和分析结果的绝均差以确定最佳拟合次数, 并对问题二有一定的指导作用, 在此验证了所提出模型的有效性。

5.2 模型的缺点

在本文所有涉及到的模型中不能分析卫星的所有数据是否都为有效, 在某个区域内时间上跨度太大的数据所计算的调和常数会和时间跨度短的调和常数一起平均掉, 进而导致精度上会存在误差。

5.3 模型的推广

所构建的根据高度计海面异常高度资料来获取潮汐调和常数的模型不仅适用于中国南海潮汐, 对于其他海域也具有适用性。此外本文仅仅分析了四个主要半日分潮 M_2 、 S_2 、 K_1 、 O_1 的调和常数及其应用, 对于其它半/全日分潮及长周期分潮同样适用。

六、参考文献

- [1] ZHAO Yunxia, WEI Zexun, Wang Xinyi,等. The South China Sea tides analysis based on TOPEX/Poseidon altimetry 利用 T/P 卫星高度计资料调和和分析南海潮汐信息[J]. 海洋科学, 2012, 36(5):10-17.
- [2] 孙丽艳: 渤黄东海潮汐底摩擦系数的优化研究 [硕士学位论文]. 青岛: 中国海洋大学海洋环境学院, 2006.
- [3] 付延光, 周兴华, 许军,等. 利用 TOPEX/Poseidon 和 Jason-1 高度计数据提取中国南海潮信息[J]. 武汉大学学报(信息科学版), 2018(6).
- [4] 范丽丽: 风暴潮数值同化研究和高度计资料拟合方法研究 [硕士学位论文]. 青岛: 中国海洋大学海洋环境学院, 2011.
- [5] 陈宗镛, 潮汐学, 北京: 科学出版社, 138-144,1980.
- [6] Cummins P F, Cherniawsky J Y, Foreman M G G. North Pacific internal tides from the Aleutian Ridge: Altimeter observations and modeling[J]. Journal of Marine Research, 2001, 59(2):167-191.
- [7] 王斌, 张晓爽, 吕咸青,等. 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(3):266-272.
- [8] Fan, L.L., Wang B., and Lv X.Q., 2011, Cotidal Charts near Hawaii Derived from TOPEX/Poseidon Altimetry Data, J Atmos Ocean Technol, 28, 606-614.

七、附录

7.1 问题一的源代码

HG_calculator.m

```
clear,clc
%% 数据格式
% 在某一点的观测值 Time/h ele/m
%          32      2
%          33      3
%          ...      ...
```

```

data = load('obs_data.txt');
time = data(:,1)/60/60;
ele = data(:,2);
N = length(ele);
% Start time: 2000/01/01 00:00:00
year = 2000;
month = 1;
day = 1;
%%% 计算振幅和相位
% mu0,mu1,mu2,mu3,mu4,mu5,mu6
mu = [0., 2., 0., 0., 0., 0., 0.; %M2
      0., 2., 2., -2., 0., 0., 0.; %S2
      1., 1., 1., 0., 0., 0., 0.; %K1
      -1., 1., -1., 0., 0., 0., 0.]; %O1
% sigma1 为 M2 分潮, sigma2 为 S2 分潮, sigma3 为 K1 分潮, sigma4 为 O1 分潮
% sigma 为角速度, 单位化为弧度/小时
sigma = ones(1,4);
for i = 1:4
    sigma(i) = 14.49205211*mu(i,2) + 0.54901653*mu(i,3) + ...
              0.04106864*mu(i,4) + 0.00464183*mu(i,5) + 0.00220641*mu(i,6) +
              0.00000196*mu(i,7);
    sigma(i) = sigma(i)*pi/180.;
end
% G 为最小二乘法左手矩阵 最后一列的值为 1 对应 S0
G(:,9) = ones(N, 1);
for i = 1 : 4
    G(:,2*i-1) = cos(sigma(i) * time);% 此时的 time 是一个累积量 单位是小时
    G(:,2*i) = sin(sigma(i) * time);
end
% 最小二乘法求解出 xi 和 yi; z 列向量保存的值分别是 x1,y1,x2,y2...,S0
z = G\ele;
% 此时 theta = g - voj -uj; R = fj*Hj
% xj = fj*Hj*cos(g - voj -uj)
% yj = fj*Hj*sin(g - voj -uj)
% R 为振幅 theta 为相位角 一共计算了 8 个分潮值
R = zeros(1,4);
theta = zeros(1,4);
for i = 1:4
    R(i) = sqrt(z(2*i-1)^2+z(2*i)^2);
    if(z(2*i-1) >= 0)
        theta(i) = asin(z(2*i)/R(i));
    else
        theta(i) = pi - asin(z(2*i)/R(i));
    end
end
end
%%% 计算调和常数 H 和 g
midtime = median(time); % median(M):返回 M 的从大到小排列的中间值
midday = fix(midtime / 24); % 中间小时数转换成累积的天数
midhour = mod(midtime,24); % 中间小时数转换成当天具体的 24 小时时刻值

% 计算 1900 年至 2000 年之间的闰年数
ir = 0;
for i = 1900:year
    if(mod(i,4)==0 && mod(i,100)~=0 || mod(i,400)==0)
        ir = ir+1;
    end
end

```

```

end

% 计算 2000/01/01 00 时起至数据起始时刻累计的日期数
m = [0,31,28,31,30,31,30,31,31,30,31,30,31]; % 每个月的天数
dd = -1;
if(mod(year,4)==0 && mod(year,100)~=0 || mod(year,400)==0) % 判断是否为闰年
    m(3) = 29; % 闰年二月天数为 29
end
for i = 1: month
    dd = dd + m(i);
end
dd = dd + day; % 在此题目中 dd = 0

% 计算 2000/01/01 00 时至中间时刻累计的日期数 %
dd = dd + midday;
nd = dd;

% 计算中间时刻的各个天文元素
t = 0; %t 通常设为 0
s = 277.02 + 129.3848*(year -1900.) + 13.1764*(nd +ir +t/24);
h1 = 280.19 - 0.2387*(year -1900.) + 0.9857*(nd +ir +t/24);
p = 334.39 + 40.6625*(year -1900.) + 0.1114*(nd +ir +t/24);
n1 = 100.84 + 19.3282*(year -1900.) + 0.0530*(nd +ir +t/24);
p1 = 281.22 + 0.0172*(year -1900.) + 0.00005*(nd +ir +t/24);
tau = 15*t -s +h1;

% Doodson 数差值
dmu4 = [-2.,-2.,-2.,0.,0.,0.,0.,0.,2.,2.,2.,2.];
dmu5 = [-1.,0.,1.,-2.,-1.,0.,1.,2.,3.,-1.,0.,1.,2.];

% 计算引潮力系数比 rho
rho = [0.,0.,0.,0.0005,-0.0373,1.,0.,0.,0.,0.,0.0006,0.0002,0.; %M2
        0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.; %S2
        0.,0.,0.,0.0005,-0.0373,1.,0.,0.,0.,0.,0.0006,0.0002,0.; %N2
        0.,0.,0.,0.,-0.0128,1.,0.298,0.0324,0.,0.,0.,0.; %K2
        0.0002,0.,0.,0.0001,-0.0198,1.,0.1356,-0.0029,0.,0.,0.,0.; %K1
        0.,0.,0.,-0.0058,0.1885,1.,0.,0.,0.,0.0002,-0.0064,-0.0010,0.; %O1
        0.,0.,0.,0.0008,-0.0112,1.,0.,0.,0.,0.,-0.0015,-0.0003,0.; %P1
        0.,0.,0.,-0.0058,0.1885,1.,0.,0.,0.,0.0002,-0.0064,-0.0010,0.]; %Q1

% 计算交点因子和交点订正角
for i = 1:4
    f(i) = 0.; %各个分潮的交点因子
    u(i) = 0.; %各个分潮的交点订正角
    fc(i) = 0.; %fcosu
    fs(i) = 0.; %fsinu
end
for i = 1:4
    for j = 1:13
        fc(i)=fc(i)+rho(i,j)*cos((dmu4(j)*p +dmu5(j)*n1)*pi/180.);% 注意转换成弧度计算
        fs(i)=fs(i)+rho(i,j)*sin((dmu4(j)*p +dmu5(j)*n1)*pi/180.);
    end
    f(i) = sqrt(fc(i)^2.+fs(i)^2.);
    if(fc(i) >= 0.)
        u(i) = asin(fs(i)/f(i));
    else

```

```

        u(i) = pi - asin(fs(i)/f(i));
    end
end
f(2) = 1.; % S2 为常数不用计算
u(2) = 0.;

% 计算初始位相(取中间时刻, t=1 时刻的天文元素)
for i = 1:4
    v0(i) = mu(i,1)*90 + mu(i,2)*tau + mu(i,3)*s + mu(i,4)*h1 + mu(i,5)*p + mu(i,6)*n1
    + mu(i,7)*p1;
    v0(i) = v0(i)*pi/180.;
end

% 利用交点因子、交点订正角和初始位相计算调和常数
H = R ./ f;
g = theta + v0 + u;
g = g * 180/pi;
g = mod(g, 360); % g 对 360 取余

```

sec_to_year_month_date.m

```

function [real_year, real_month, real_date,...
    real_hour, real_min, real_sec] = sec_to_year_month_date( account_sec )
%account_sec = year_table(1996 - 1991,2) + (31 + 29)*24*60*60 + 23*60*60 -1;% 1 day =
86400 sec
leap_year_month = [31,29,31,30,31,30,31,31,30,31,30,31];
comm_year_month = [31,28,31,30,31,30,31,31,30,31,30,31];

year_table=[
    1992    -252460800    -220838401
    1993    -220838400    -189302401
    1994    -189302400    -157766401
    1995    -157766400    -126230401
    1996    -126230400    -94608001
    1997    -94608000     -63072001
    1998    -63072000     -31536001
    1999    -31536000      -1
    2000         0          31622399
    2001     31622400      63158399
    2002     63158400      94694399
    2003     94694400     126230399
    2004    126230400     157852799
    2005    157852800     189388799
    2006    189388800     220924799
    2007    220924800     252460799
    2008    252460800     284083199
    2009    284083200     315619199
    2010    315619200     347155199
    2011 347155200      378691199
    2012    378691200     410313599
    2013    410313600     441849599
    2014    441849600     473385599
    2015    473385600     504921599
    2016    504921600     536543999
    2017    536544000     568079999
];
if(account_sec>0)

```

```

        for i=1:18
            if(account_sec>=year_table(i+8,2)&& account_sec<=year_table(i+8,3))
                real_year = year_table(i+8,1);
                break
            end
        end
    else
        for i=1:8
            if(account_sec>=year_table(i,2) && account_sec<=year_table(i,3))
                real_year = year_table(i,1);
                break
            end
        end
    end
    end
    %%% month
    sec_in_year = account_sec - year_table(real_year-1991,2);
    if(mod(real_year,4)==0)
        for j=1:12
            if((sec_in_year - leap_year_month(j)*24*60*60)>=0)
                sec_in_year = sec_in_year - leap_year_month(j)*24*60*60;
            else
                real_month = j;break
            end
        end
    else
        for j=1:12
            if((sec_in_year - comm_year_month(j)*24*60*60)>=0)
                sec_in_year = sec_in_year - comm_year_month(j)*24*60*60;
            else
                real_month = j;break
            end
        end
    end
    end
    %%% date hour min sec
    real_date = floor(sec_in_year/(24*60*60)) + 1;
    real_hour = floor(mod(sec_in_year/60/60,24));
    real_min = floor(mod(sec_in_year/60,60));
    real_sec = mod(sec_in_year,60);
    DISPDATE = [num2str(real_year),',',num2str(real_month),',',num2str(real_date)...
        ', ',num2str(real_hour),':',num2str(real_min),':',num2str(real_sec)];
    disp(DISPDATE)

```

pre_trea_data.m

```

clear;clc;
load('dataTP.mat');
data_length= length(TP);
temp=1;
for i=1:data_length
    if(TP(i,3)>=300)
        index(temp)=i;% index(i)表示第 i 个大于 300 的 TP 索引
        temp= temp+1;
    end
end
for i=1:temp-1
    posi(i,1) = TP(index(i),1);% N
    posi(i,2) = TP(index(i),2);% E

```

```

        num_data(i) = TP(index(i),3);
    end
    temp2=1;
    % 16.067000,120.100000
    for i=1:temp-1
        if(posi(i,2)>22 && posi(i,2)<23)
            if(posi(i,1) >119 && posi(i,1)<120.2) % E N
                record(temp2)=i;
                temp2 = temp2+1;
            end
        end
    end
    disp('record = ');
    disp(length(record))
    %% 手动提取观测点的数值
    idxi = 1;%重复 1: record 的长度
    temp1 = 0;
    obs_point(1,:) = TP(index(record(idxi)),:);
    for j=1:obs_point(1,3)
        [real_year, real_month, real_date,...
        real_hour, real_min, real_sec] = sec_to_year_month_date( TP(index(record(idxi))+j,1) );
        if(real_year > 2002)
            temp1 = temp1 + 1;
            obs_point(temp1,1) = real_year;
            obs_point(temp1,2) = real_month;
            obs_point(temp1,3) = real_date;
            obs_point(temp1,4) = TP(index(record(idxi))+j,1);
            obs_point(temp1,5) = TP(index(record(idxi))+j,2);
        end
    end
end

```

map_nanhai.m

```

clear;clc;
load('dataTP.mat');
[data_length,tem] = size(TP);
temp=1;
for i=1:data_length
    if(TP(i,3)>=200)
        index(temp)=i;
        temp= temp+1;
    end
end
for i=1:temp-1
    posi(i,1) = TP(index(i),1);
    posi(i,2) = TP(index(i),2);
    num_data = TP(index(i),3);
end
%% 在地图上标点
ax = worldmap([2 25],[99 122]);
land = shaperead('landareas', 'UseGeoCoords', true);
geoshow(ax, land, 'FaceColor', [0.5 0.7 0.5])
lakes = shaperead('worldlakes', 'UseGeoCoords', true);
geoshow(lakes, 'FaceColor', 'blue');
labelLat = posi(:,2);
labelLon = posi(:,1);
geoshow(labelLat, labelLon,...

```

```

        'DisplayType', 'point',...
        'Marker', 'o',...
        'MarkerEdgeColor', 'r',...
        'MarkerFaceColor', 'r',...
        'MarkerSize', 3);
world_seas = [11.795000    99.816667
6.226667 102.106667
5.265000 103.186667
3.975000 103.430000
2.806667 104.140000
18.083333    106.283333
20.666667    106.816667
10.333333    107.066667
19.100000    108.616667
21.483333    109.083333
20.016667    110.283333
21.583333    111.833333
3.216667 113.066667
22.166667    113.550000
4.391667 113.971667
22.300000    114.216667
22.750000    115.350000
5.983333 116.066667
2.100000 117.750000
4.233333 117.883333
24.450000    118.066667
5.810000 118.066667
9.750000 118.733333
22.610000    120.291667
14.583333    120.966667
6.066667 121.000000
];
geoshow(world_seas(:,1), world_seas(:,2),...
        'DisplayType', 'point',...
        'Marker', 'o',...
        'MarkerEdgeColor', 'b',...
        'MarkerFaceColor', 'b',...
        'MarkerSize', 5);
china_seas = [8.533000 100.000000
13.350000    100.000000
2.617000 100.100000
2.500000 100.200000
13.533000    100.283000
5.250000 100.283000
5.417000 100.350000
2.200000 100.600000
7.233000 100.617000
4.000000 100.750000
13.150000    100.817000
12.650000    100.883000
13.500000    100.983000
4.033000 101.000000
2.883000 101.000000
2.900000 101.250000
6.950000 101.283000
3.000000 101.383000
12.583000    101.450000

```

2.517000 101.783000
 12.467000 102.067000
 6.200000 102.167000
 2.183000 102.250000
 2.050000 102.333000
 12.117000 102.567000
 2.050000 102.567000
 5.817000 102.617000
 5.667000 102.767000
 11.417000 103.000000
 5.350000 103.133000
 3.833000 103.333000
 4.217000 103.433000
 4.783000 103.433000
 2.817000 103.467000
 3.533000 103.467000
 10.633000 103.483000
 10.550000 103.600000
 3.250000 103.667000
 2.433000 103.983000
 2.800000 104.133000
 10.367000 104.467000
 18.767000 105.750000
 18.800000 105.767000
 19.917000 106.017000
 3.233000 106.250000
 17.700000 106.467000
 8.683000 106.600000
 20.867000 106.667000
 20.667000 106.800000
 10.400000 107.017000
 20.717000 107.033000
 10.333000 107.067000
 20.950000 107.067000
 20.767000 107.133000
 21.033000 107.367000
 21.100000 107.600000
 16.567000 107.617000
 21.200000 107.950000
 10.700000 107.983000
 4.750000 108.000000
 3.800000 108.033000
 16.117000 108.217000
 11.167000 108.700000
 3.050000 108.850000
 2.267000 108.900000
 10.500000 108.950000
 2.500000 109.000000
 21.483000 109.100000
 11.883000 109.200000
 13.750000 109.217000
 12.267000 109.300000
 18.217000 109.533000
 2.083000 109.650000
 2.000000 109.667000
 20.050000 110.317000
 21.167000 110.417000

20.950000	110.600000
20.167000	110.683000
2.150000	111.250000
2.133000	111.517000
16.550000	111.617000
2.300000	111.667000
2.817000	111.700000
2.900000	112.083000
2.750000	112.183000
3.183000	113.033000
3.250000	113.067000
3.517000	113.300000
23.083000	113.433000
22.200000	113.550000
22.200000	113.550000
3.967000	113.700000
22.367000	113.967000
4.383000	113.983000
4.583000	113.983000
22.233000	114.000000
22.483000	114.017000
22.100000	114.033000
22.467000	114.183000
22.467000	114.333000
4.617000	114.650000
22.733000	114.733000
4.900000	114.983000
5.033000	115.067000
5.000000	115.133000
5.267000	115.250000
4.883000	115.350000
4.850000	115.400000
4.950000	115.417000
5.083000	115.550000
5.217000	115.600000
6.200000	115.600000
5.700000	115.667000
22.800000	115.800000
5.750000	115.900000
6.017000	115.967000
5.983000	116.067000
20.683000	116.717000
23.333000	116.750000
6.883000	116.850000
8.000000	117.067000
23.600000	117.283000
2.817000	117.367000
6.883000	117.483000
2.200000	117.500000
2.200000	117.533000
23.733000	117.533000
2.917000	117.700000
2.200000	117.700000
6.217000	117.717000
3.500000	117.800000
2.167000	117.867000
4.233000	117.883000

2.033000 117.883000
 6.500000 117.917000
 24.450000 118.067000
 5.833000 118.117000
 24.383000 118.167000
 5.033000 118.333000
 24.400000 118.417000
 4.950000 118.583000
 4.483000 118.617000
 9.733000 118.700000
 10.067000 118.767000
 23.217000 119.417000
 10.883000 119.417000
 4.783000 119.433000
 25.000000 119.450000
 23.517000 119.500000
 23.650000 119.517000
 23.367000 119.517000
 23.550000 119.550000
 23.550000 119.550000
 23.733000 119.600000
 23.600000 119.667000
 23.550000 119.667000
 5.850000 119.733000
 5.033000 119.767000
 16.400000 119.900000
 15.767000 119.900000
 11.800000 119.950000
 23.100000 120.067000
 23.183000 120.067000
 16.067000 120.100000
 23.450000 120.133000
 23.383000 120.150000
 23.000000 120.150000
 23.700000 120.167000
 5.350000 120.217000
 5.133000 120.233000
 22.617000 120.250000
 22.617000 120.267000
 22.683000 120.267000
 14.817000 120.283000
 5.650000 120.283000
 16.617000 120.300000
 23.917000 120.300000
 5.217000 120.317000
 17.783000 120.417000
 22.467000 120.433000
 24.183000 120.483000
 5.200000 120.500000
 24.267000 120.517000
 18.483000 120.567000
 24.383000 120.567000
 22.350000 120.583000
 14.383000 120.600000
 22.067000 120.683000
 22.450000 120.717000
 21.967000 120.750000

```

24.617000    120.750000
21.933000    120.767000
5.700000 120.883000
13.767000    120.917000
24.850000    120.917000
14.583000    120.967000
6.067000 121.000000
5.933000 121.033000
12.350000    121.100000
18.617000    121.100000
22.833000    121.183000
13.433000    121.183000
18.867000    121.283000
23.117000    121.400000
15.200000    121.433000
6.733000 121.467000
22.667000    121.483000
22.033000    121.567000
13.900000    121.600000
14.667000    121.617000
6.150000 121.833000
14.933000    121.833000
24.583000    121.867000
6.550000 121.867000
24.933000    121.883000
14.900000    121.967000
12.267000    121.983000
];
geoshow(china_seas(:,1), china_seas(:,2),...
        'DisplayType', 'point',...
        'Marker', 'o',...
        'MarkerEdgeColor', 'y',...
        'MarkerFaceColor', 'y',...
        'MarkerSize', 5);

```

7.2 问题二的源代码

Track_HG_calculator.m

```

% 计算轨道的 H 和 g
clear;clc;
load('dataTP.mat');%E 纬度 lat N 经度 lon
[data_length,tem] = size(TP);
temp=1;
for i=1:data_length
    if(TP(i,3)>=0)
        index(temp)=i;
        temp= temp+1;
    end
end
posi = zeros(length(index),2);
num_data = zeros(length(index),1);
for i=1:temp-1
    posi(i,1) = TP(index(i),1);
    posi(i,2) = TP(index(i),2);
    num_data(i) = TP(index(i),3);

```

```

end
Elat = 113;
M2_mean_H = zeros(50,1);M2_mean_g = zeros(50,1);
S2_mean_H = zeros(50,1);S2_mean_g = zeros(50,1);
K1_mean_H = zeros(50,1);K1_mean_g = zeros(50,1);
O1_mean_H = zeros(50,1);O1_mean_g = zeros(50,1);
record = 0;
for kk = 1:200 % Elat from 110 to 115
    [dE,dN] = xtoy(Elat);
    temp2 = 1;
    for i=1:temp2-1
        if(posi(i,2) >= dN(1) && posi(i,2) <= dN(2))
            if(posi(i,1) >= dE(1) && posi(i,1) <= dE(2)) % E N 如果有符合数据的点
                record(temp2)=index(i);% 记录 i 的具体对应 TP 的位置
                temp2 = temp2+1;
            end
        end
    end
    if(temp2 == 1)% temp2 == 1 代表没有找到数据
        Elat = Elat + 0.1;
        continue
    end
    Nrecord = length(record);
    M2_H = zeros(Nrecord,1);M2_g = zeros(Nrecord,1);
    S2_H = zeros(Nrecord,1);S2_g = zeros(Nrecord,1);
    K1_H = zeros(Nrecord,1);K1_g = zeros(Nrecord,1);
    O1_H = zeros(Nrecord,1);O1_g = zeros(Nrecord,1);
    for i = 1:Nrecord
        temp1 = 0;
        num_TPdata = TP(record(i),3);
        obs_point = zeros(num_TPdata,2);
        for j=1:num_TPdata
            [real_year, real_month, real_date,...
             real_hour, real_min, real_sec] = sec_to_year_month_date(TP(record(i)+j,1));
            if(real_year > 2002)
                temp1 = temp1 + 1;
                obs_point(temp1,1) = TP(record(i)+j,1);
                obs_point(temp1,2) = TP(record(i)+j,2);
            end
        end
        end
        % 将 obs_point 传进函数里面去 返回 H,g
        [H,g] = funHG( obs_point );
        M2_H(i) = H(1);M2_g(i) = g(1);
        S2_H(i) = H(2);S2_g(i) = g(2);
        K1_H(i) = H(3);K1_g(i) = g(3);
        O1_H(i) = H(4);O1_g(i) = g(4);
    end
    M2_mean_H(kk) = mean(M2_H);M2_mean_g(kk) = mean(M2_g);
    S2_mean_H(kk) = mean(S2_H);S2_mean_g(kk) = mean(S2_g);
    K1_mean_H(kk) = mean(K1_H);K1_mean_g(kk) = mean(K1_g);
    O1_mean_H(kk) = mean(O1_H);O1_mean_g(kk) = mean(O1_g);
    Elat = Elat + 0.01;
end
save D:\HGdata\M2_mean_H.txt -ascii M2_mean_H
save D:\HGdata\M2_mean_g.txt -ascii M2_mean_g
save D:\HGdata\S2_mean_H.txt -ascii S2_mean_H
save D:\HGdata\S2_mean_g.txt -ascii S2_mean_g

```

```

save D:\HGdata\K1_mean_H.txt -ascii K1_mean_H
save D:\HGdata\K1_mean_g.txt -ascii K1_mean_g
save D:\HGdata\O1_mean_H.txt -ascii O1_mean_H
save D:\HGdata\O1_mean_g.txt -ascii O1_mean_g

```

子函数:

```

function [X,Y] = xtoy(x)
%y1 = -2.5914*x + 307.3062; %轨道 1 的表达式
y2 = 2.5875*x - 280.5173; %轨道 2 的表达式
X = [x-0.3 x+0.3];
Y = [y2-0.3 y2+0.3];

```

7.3 问题三的源代码

遍历地图上所有的数据点并做调和与分析

```

clear;clc
clear;clc;
load('dataTP.mat');%E 纬度 lat N 经度 lon
[data_length,tem] = size(TP);
temp=1;
for i=1:data_length
    if(TP(i,3)>=100)
        index(temp)=i;
        temp= temp+1;
    end
end
posi = zeros(length(index),2);
num_data = zeros(length(index),1);
for i=1:temp-1
    posi(i,1) = TP(index(i),1);
    posi(i,2) = TP(index(i),2);
    num_data(i) = TP(index(i),3);
end
num_grid = 46;
M2_mean_H = zeros(num_grid,num_grid);M2_mean_g = zeros(num_grid,num_grid);
S2_mean_H = zeros(num_grid,num_grid);S2_mean_g = zeros(num_grid,num_grid);
K1_mean_H = zeros(num_grid,num_grid);K1_mean_g = zeros(num_grid,num_grid);
O1_mean_H = zeros(num_grid,num_grid);O1_mean_g = zeros(num_grid,num_grid);
record = 0;
center_E = 99.25; center_N = 2.25;
temE = 0;temN = 0;
for iN = 1:46
    center_E = 99.25;
    for jE = 1:46
        dE = [center_E - 0.25, center_E + 0.25];
        dN = [center_N - 0.25, center_N + 0.25];
        temp2 = 1;
        for i=1:temp-1
            if(posi(i,2) >= dN(1) && posi(i,2) <= dN(2))
                if(posi(i,1) >= dE(1) && posi(i,1) <= dE(2)) % E N 如果有符合数据的点
                    record(temp2)=index(i);% 记录 i 的具体对应 TP 的位置
                    temp2 = temp2+1;
                end
            end
        end
    end
end

```

```

if(temp2 == 1)% temp2 == 1 代表没有找到数据
    center_E = center_E + 0.5;
    continue
end
Nrecord = length(record);
M2_H = zeros(Nrecord,1);M2_g = zeros(Nrecord,1);
S2_H = zeros(Nrecord,1);S2_g = zeros(Nrecord,1);
K1_H = zeros(Nrecord,1);K1_g = zeros(Nrecord,1);
O1_H = zeros(Nrecord,1);O1_g = zeros(Nrecord,1);
for i = 1:Nrecord
    temp1 = 0;
    num_TPdata = TP(record(i),3);
    obs_point = zeros(num_TPdata,2);
    for j=1:num_TPdata
        [real_year, real_month, real_date,...
         real_hour, real_min, real_sec] =
sec_to_year_month_date(TP(record(i)+j,1));
        if(real_year > 2002)
            temp1 = temp1 + 1;
            obs_point(temp1,1) = TP(record(i)+j,1);
            obs_point(temp1,2) = TP(record(i)+j,2);
        end
    end
    % 将 obs_point 传进函数里面去 返回 H,g
    [H,g] = funHG( obs_point );
    M2_H(i) = H(1);M2_g(i) = g(1);
    S2_H(i) = H(2);S2_g(i) = g(2);
    K1_H(i) = H(3);K1_g(i) = g(3);
    O1_H(i) = H(4);O1_g(i) = g(4);
end
M2_mean_H(iN,jE) = mean(M2_H);M2_mean_g(iN,jE) = mean(M2_g);
S2_mean_H(iN,jE) = mean(S2_H);S2_mean_g(iN,jE) = mean(S2_g);
K1_mean_H(iN,jE) = mean(K1_H);K1_mean_g(iN,jE) = mean(K1_g);
O1_mean_H(iN,jE) = mean(O1_H);O1_mean_g(iN,jE) = mean(O1_g);
center_E = center_E + 0.5;
end
center_N = center_N + 0.5;
end
save D:\HGdata\M2_mean_H.txt -ascii M2_mean_H
save D:\HGdata\M2_mean_g.txt -ascii M2_mean_g
save D:\HGdata\S2_mean_H.txt -ascii S2_mean_H
save D:\HGdata\S2_mean_g.txt -ascii S2_mean_g
save D:\HGdata\K1_mean_H.txt -ascii K1_mean_H
save D:\HGdata\K1_mean_g.txt -ascii K1_mean_g
save D:\HGdata\O1_mean_H.txt -ascii O1_mean_H
save D:\HGdata\O1_mean_g.txt -ascii O1_mean_g

```

插值多项式拟合

```

clear;clc;
load M2_mean_H.txt
load M2_mean_g.txt
load S2_mean_H.txt
load S2_mean_g.txt
load K1_mean_H.txt
load K1_mean_g.txt
load O1_mean_H.txt

```

```

load O1_mean_g.txt
Amp = M2_mean_H;
Phi = M2_mean_g/180*pi;
EL = 111.61;ER = 115.12;
N = 200;
y = EL:(ER-EL)/N:ER-(ER-EL)/N;

pl_order = 7;
M = zeros(pl_order+1);
b = zeros(pl_order+1,1);c = zeros(pl_order+1,1);
for i=1:pl_order+1
    for j=1:pl_order+1
        fsum = 0;
        for k=1:N
            fsum = fsum + y(k)^(i+j-2);
        end
        M(i,j) = fsum;
    end
    fsumb = 0;fsumc = 0;
    for k=1:N
        fsumb = fsumb + y(k)^(i-1)*Amp(k)*cos(Phi(k));
        fsumc = fsumc + y(k)^(i-1)*Amp(k)*sin(Phi(k));
    end
    b(i) = fsumb;
    c(i) = fsumc;
end
a1 = M\b;
a2 = M\c;
sumf = zeros(N,1);
sumg = zeros(N,1);
interf = zeros(N,1);
interg = zeros(N,1);
interA = zeros(N,1);
interG = zeros(N,1);
for i=1:N
    for j = 1:pl_order
        sumf(i) = sumf(i) + a1(j+1)*y(i)^j;
        sumg(i) = sumg(i) + a2(j+1)*y(i)^j;
    end
    sumf(i) = sumf(i) + a1(1);
    sumg(i) = sumg(i) + a2(1);
end
for i=1:N
    interf(i)= Amp(i)*cos(Phi(i)) - sumf(i);
    interg(i)= Amp(i)*sin(Phi(i)) - sumg(i);
    interA(i)= sqrt(interf(i)^2 + interg(i)^2);
    if(interf(i) >= 0)
        interG(i) = asin(interg(i)/interA(i));
    else
        interG(i) = pi - asin(interg(i)/interA(i));
    end
end
figure(1)
A1 = interA; B1 = Amp;
plot(y,interA,'-','LineWidth',1.5);
hold on
plot(y,Amp,'-','LineWidth',1.5)

```

```

xlabel('经度/°E')
ylabel('幅度/m')
legend('3 次多项式拟合内潮振幅分量','海表面 M_2 分潮的振幅曲线')

figure(2)
A2=interG/pi*180;B2=Phi/pi*180;
plot(y,A2,'-', 'LineWidth',1.5);
hold on
plot(y,B2,'-', 'LineWidth',1.5)
xlabel('经度/°E')
ylabel('迟角/°')
legend('3 次多项式拟合内潮迟角分量','海表面 M_2 分潮的迟角曲线')

```

7.4 问题四的源代码

由于问题四是对问题二多项式拟合次数的最佳值确定，因此问题四的源代码与问题二相同，这里不再赘述。