

# “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的潮汐调和常数，利用多项式拟合方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，使用克里金插值方法制作南海的各主要分潮的同潮图，同时做了结果验证以及验证分析。确定出多项式拟合中利用的特定函数的最佳次数，并且论证了最佳次数对制作同潮图的启示以及地统计学方法的应用。

问题一中，针对采用调和分析方法计算各分潮的调和常数  $H$  和  $g$  这一问题，首先对高度计数据进行预处理，读取高度计点文件，对位置点按顺序进行了编号 1-4520，根据主要分潮的周期及在 T/P 采样的混淆周期计算所得主要分潮基本的分辨时间，由于实现四个主要分潮信号可靠分离至少需要 2.97 年的数据，将观测序列长度小于 110 的点剔除掉；按照卫星运行规律，由于 T/P 卫星自西向东围绕地球运行，我们将由南向北运行为上升轨道，由北向南运行为下降轨道，在南海海区自西向东对 9 条上升轨道、11 条下降轨道进行编号，通过计算各分潮的角速率  $\sigma$  和分潮的初位相  $V_0$  计算出分潮幅角  $V$  然后计算出各分潮的交点因子  $f$  与交点订正角  $\mu$ ，最后用最小二乘法计算各分潮的调和常数  $H$  和  $g$ 。针对调和常数验证与分析评价问题，为了比较模型自身精度，选取上行轨道和下降轨道的交点，计算差值，又以  $M_2$  分潮为例进行交叉点验证，验证结果为  $M_2$  分潮调和与分析结果偏差很小。同符合验证条件的验潮站对比验证，从验潮站验证结果来看，振幅最大差值 0.095 米，最小差值 0.004 米，平均 RMS 为 0.054 米，符合程度较高；迟角最大差值 216.01 度，最小差值 3.39 度，平均 RMS 为 134 度，偏差较大。通过分析，在计算过程中，初始时刻决定了初始相位，影响较大；而  $f$  和  $\mu$  变化较为缓慢，以 18.6 年为一周期，在计算过程中影响较小。对于迟角，有三种迟角，地方迟角，区时迟角，专用区时迟角，取决于分潮初相位时间系统的选取。模型计算结果振幅相差较小，而迟角偏差较大，说明在计算过程中对于时间的判断和计算存在问题。根据参考资料，受海底地形影响的潮汐地摩擦系数对于高度计的调和与分析结果也有影响，具体表现为在一定深度的范围内，调和与分析结果与验潮站相符较好，南海岛礁众多，海底地形复杂，调和与分析计算过程中未考虑摩擦系数影响，也是造成迟角偏差大的原因。

问题二中，针对第一小问，首先，要根据问题一计算得到的各主要分潮的潮汐调和常数沿轨道作图，以  $M_2$  分潮为例，对  $M_2$  分潮沿 up06 轨道的潮汐调和常数  $H$  和  $g$  进行了作图，展示了内潮对正压潮调制的锯齿状细结构。针对第二小问，对各分潮调和常数  $H$  和  $g$  建立了多项式拟合数学模型，以  $M_2$  分潮为例，利用多项式对  $M_2$  的潮汐调和常数  $H$  和  $g$  进行了 3 ~ 20 次多项式拟合，展示了不同次数的潮汐调和常数  $H$  和  $g$  拟合结果，从拟合结果的均方根差可以看出拟合次数越高拟合效果未必更好。从整体来看，14、15、16、17、18、19、20 次多项式拟合的

---

振幅均方根差优于其他次数。对于迟角  $g$ ，16 次多项式拟合得到的均方根差最小，20 次多项式拟合结果的均方根差比 11-18 都大。同时又以  $M_2$  内潮为例以上升轨 up06 号轨道为例，采用了多项式拟合方法进行了不同次数的拟合，沿上升 up06 号轨道分别进行 11、13、16 次多项式拟合得到的振幅平均绝对均差分别为 0.06、0.06、0.05m，16 次多项式拟合均方根差最小。5 次多项式的拟合结果同调和分析结果有较大偏差；11 次多项式拟合结果在某些位置同样存在偏离现象，因此拟合结果不能代表正压潮值；而 16 次多项式拟合结果与海表面  $M_2$  分潮振幅虽然在个别位置有较大偏差，但整体效果较好能准确地表示正压潮值，从而保证内潮提取结果的准确性。最终提取出内潮海表面信号。

问题三中，首先，建立了克里金插值的数学模型。然后利用克里金插值的方法制作出了各主要分潮的同潮图。然后，通过对比，调和分析结果整体分布同论文及验潮站结果相似，表明了克里金插值方法具有一定的优势。最后，同时对于不同水深区域，验证了在近岸区域验潮站的插值结果更加准确，在深海中高度计调和分析插值结果细节表现较好。

问题四中，首先，针对各分潮调和常数  $H$  和  $g$  建立了多项式拟合数学模型。然后，选用 3-20 次多项式对  $M_2$  的潮汐调和常数  $H$  和  $g$  进行了拟合，确定了最佳次数为 16 次。最后，论证了在多项式拟合过程中，拟合次数过高会导致过拟合的情况，选择合适的次数后，可以利用多项式空间插值对区域内任意一点插值，有助于第 3 问同潮图的绘制。最后讨论了克里金插值方法的应用及最佳函数选取。

**关键词：**调和分析；多项式拟合；内潮提取；克里金插值；同潮图；

## 1.问题重述

### 1.1 问题背景

潮汐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。

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（ $N_2$ 、 $K_2$ 、 $P_2$ 、 $Q_1$ ）和长周期分潮（ $S_a$ 、 $S_{sa}$ 、 $S_m$ 、 $S_f$ ），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。

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。为提高其它分潮及长周期分潮的潮汐调和常数获取的精度，需要解决提取的几个问题。

验潮站资料给出了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），World seas 位置由图 1 中的蓝点表示，East China Seas 位置由图 1 中的红点表示。

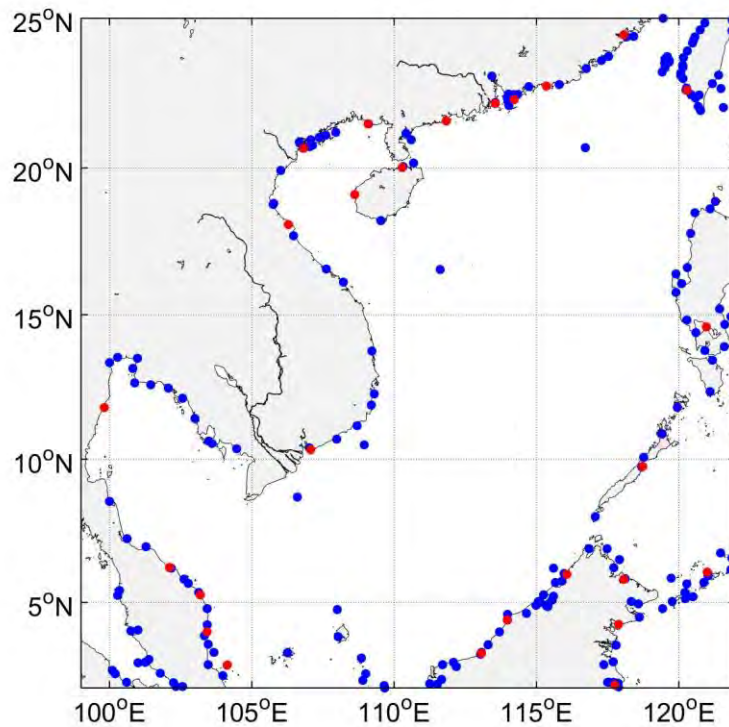


图 1 验潮站资料的分布图

数据包括 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。

1	23.500000	213.740000	8.533000	100.000000
2	65.500000	183.040000	13.350000	100.000000
3	17.000000	37.260000	2.617000	100.100000
4	17.000000	30.160000	2.500000	100.200000
5	67.100000	183.040000	13.533000	100.283000
6	21.600000	0.520000	5.250000	100.283000
7	17.700000	359.720000	5.417000	100.350000
8	18.000000	45.570000	2.200000	100.600000
9	5.500000	285.040000	7.233000	100.617000
10	20.600000	13.320000	4.000000	100.750000
11	69.500000	182.540000	13.150000	100.817000
12	59.400000	175.640000	12.650000	100.883000
13	59.800000	191.040000	13.500000	100.983000
14	24.400000	29.040000	4.033000	101.000000
15	14.000000	33.040000	2.883000	101.000000
16	12.800000	35.080000	2.900000	101.250000
17	12.200000	333.040000	6.950000	101.283000
18	18.400000	29.940000	3.000000	101.383000
19	62.500000	175.040000	12.583000	101.450000
20	5.300000	96.540000	2.517000	101.783000
21	47.400000	169.240000	12.467000	102.067000
22	29.000000	351.520000	6.200000	102.167000
23	6.400000	144.040000	2.183000	102.250000
24	11.900000	170.040000	2.050000	102.333000
25	42.400000	164.640000	12.117000	102.567000
26	13.700000	164.040000	2.050000	102.567000
27	41.500000	359.080000	5.817000	102.617000
28	39.300000	358.080000	5.667000	102.767000
29	37.400000	160.800000	11.417000	103.000000
30	50.300000	2.080000	5.350000	103.133000
31	51.100000	18.040000	3.833000	103.333000
32	56.100000	14.080000	4.217000	103.433000
33	57.300000	3.520000	4.783000	103.433000
34	44.800000	25.520000	2.817000	103.467000
35	50.600000	19.520000	3.533000	103.467000

图 2 潮汐调和常数

其中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)。振幅迟角的求取采用了格林尼治时间。

资料 2 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 4520 个位置数据点, 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。

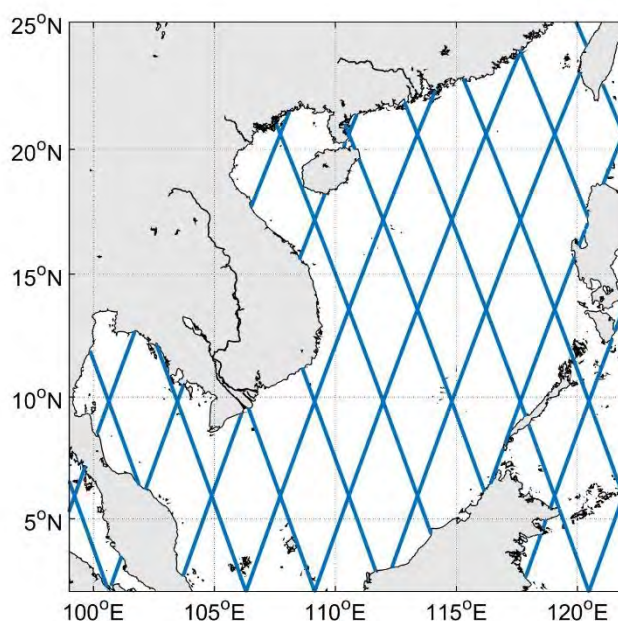


图 3 南海 TOPEX/POSEIDON 高度计资料的星下轨迹

1	100.768700	2.347300	15
2	82379328.260000	2.218400	
3	83236040.150000	1.592400	
4	98656846.520000	1.372700	
5	103797116.880000	2.169400	
6	108937388.570000	1.036500	
7	120074639.210000	1.323600	
8	157769954.740000	1.344600	
9	162910225.290000	1.970300	
10	163766937.620000	2.188200	
11	174047480.140000	1.640100	
12	194608565.170000	1.605700	
13	195465276.290000	1.213800	
14	253721675.280000	1.751800	
15	260575370.010000	1.079400	
16	522729167.430000	1.565000	
17	100.786600	2.397400	107
18	72955502.350000	1.464100	
19	73812213.290000	-0.546200	
20	77239059.980000	1.716400	
21	78095770.950000	1.355200	
22	78952481.700000	-0.764600	
23	81522618.020000	0.604400	
24	82379329.280000	2.215000	
25	83236041.170000	1.562800	
26	87519598.510000	2.297700	
27	96943426.700000	-0.806500	
28	97800137.550000	0.924200	
29	98656847.540000	1.486500	
30	99513560.520000	0.470300	
31	101226983.890000	-1.597800	
32	102083695.390000	-0.829100	
33	103797117.900000	2.199600	
34	105510544.310000	-1.047800	
35	106367255.930000	-1.654000	

图4 南海海面高度异常数据

每个位置点有若干行数据，第一行的三个数据分别代表该位置点的经度(度)，该位置点的纬度(度)，该位置点的观测数量(个)。其余行每行有两个数据，分别表示该位置点观测的时刻(秒)，对应的水位值(米)。每个地点之间没有空行隔断。观测的时刻以2000年1月1日0点0分0秒(北京时间)开始计算，即2000年1月1日0点0分0秒为0，1999年12月31日23点59分59秒为-1。

资料三给出了地形数据，地形数据来自ETOP5，全球的分辨率为5'×5'，图5的区域是2°~25°N，99°~122°E。

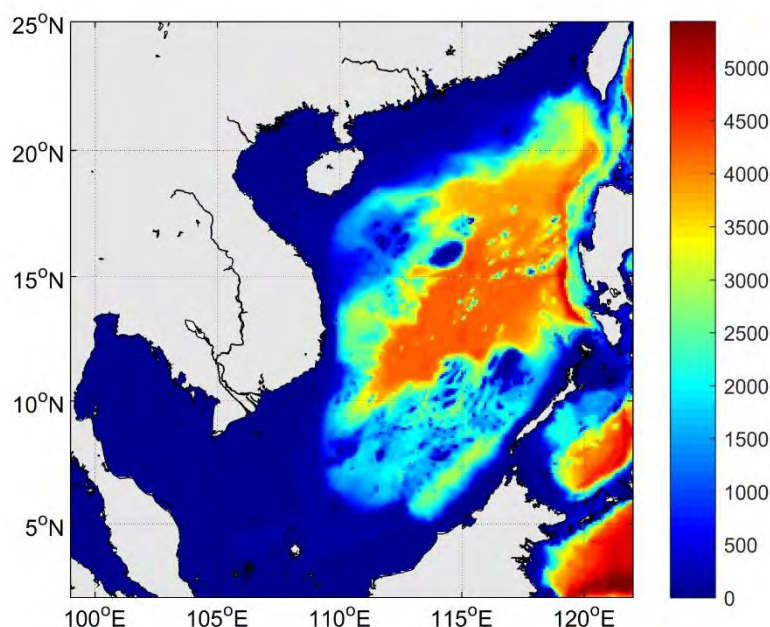


图5 南海地形图



17			NaN	NaN	NaN	NaN	NaN	NaN	1.0000000e+00	1.0000000e+00	1.9000000e+00
18			NaN	NaN	NaN	NaN	NaN	NaN	9.0000000e+00	1.9000000e+01	2.0000000e+01
19			NaN	-0.0000000e+00	-0.0000000e+00	1.9000000e+01	1.8000000e+01	1.9000000e+01	2.0000000e+01	2.0000000e+01	2.9000000e+01
20			NaN	1.9000000e+01	1.8000000e+01	1.9000000e+01	1.8000000e+01	2.0000000e+01	2.0000000e+01	2.5000000e+01	3.9000000e+01
21	1.0000000e+00	1.9000000e+01	1.9000000e+01	2.3000000e+01	2.1000000e+01	1.8000000e+01	2.1000000e+01	3.9000000e+01	3.9000000e+01	4.0000000e+01	4.0000000e+01
22	1.5000000e+01	1.9000000e+01	2.8000000e+01	3.1000000e+01	3.0000000e+01	3.0000000e+01	3.3000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01
23	2.0000000e+01	3.0000000e+01	3.8000000e+01	3.8000000e+01	3.9000000e+01	3.9000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01
24	2.9000000e+01	3.8000000e+01	3.8000000e+01	3.7000000e+01	3.1000000e+01	1.9000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01	4.0000000e+01
25	3.8000000e+01	4.0000000e+01	3.9000000e+01	3.5000000e+01	3.5000000e+01	3.4000000e+01	4.1000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01
26	3.9000000e+01	4.1000000e+01	4.1000000e+01	3.6000000e+01	4.0000000e+01	4.0000000e+01	4.1000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01
27	4.2000000e+01	4.7000000e+01	4.9000000e+01	4.2000000e+01	3.9000000e+01	5.9000000e+01	5.7000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01
28	4.9000000e+01	5.4000000e+01	5.6000000e+01	5.3000000e+01	5.9000000e+01	5.9000000e+01	6.3000000e+01	5.8000000e+01	5.8000000e+01	5.8000000e+01	5.8000000e+01
29	5.9000000e+01	5.9000000e+01	6.2000000e+01	6.5000000e+01	6.0000000e+01	7.8000000e+01	6.5000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01
30	6.0000000e+01	6.3000000e+01	6.0000000e+01	6.9000000e+01	7.8000000e+01	7.8000000e+01	6.0000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01
31	6.0000000e+01	6.0000000e+01	7.8000000e+01	7.8000000e+01	7.8000000e+01	6.0000000e+01	6.0000000e+01	5.3000000e+01	5.3000000e+01	5.3000000e+01	5.3000000e+01
32	7.0000000e+01	7.0000000e+01	7.9000000e+01	7.9000000e+01	7.1000000e+01	6.0000000e+01	5.9000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01
33	7.9000000e+01	7.9000000e+01	7.9000000e+01	7.8000000e+01	7.1000000e+01	6.4000000e+01	6.1000000e+01	6.4000000e+01	6.4000000e+01	6.4000000e+01	6.4000000e+01
34	7.9000000e+01	7.9000000e+01	7.9000000e+01	7.7000000e+01	7.5000000e+01	7.1000000e+01	6.9000000e+01	7.0000000e+01	7.0000000e+01	7.0000000e+01	7.0000000e+01
35	7.9000000e+01	8.0000000e+01	7.9000000e+01	7.8000000e+01	7.8000000e+01	7.8000000e+01	7.7000000e+01	7.4000000e+01	7.4000000e+01	7.4000000e+01	7.4000000e+01
36	8.0000000e+01	7.9000000e+01	7.9000000e+01	7.9000000e+01	8.0000000e+01	8.0000000e+01	7.9000000e+01	7.8000000e+01	7.8000000e+01	7.8000000e+01	7.8000000e+01
37	8.0000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	7.7000000e+01	7.7000000e+01	7.7000000e+01	7.7000000e+01
38	8.0000000e+01	7.9000000e+01	7.9000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	7.8000000e+01	7.5000000e+01	7.5000000e+01	7.5000000e+01	7.5000000e+01
39	8.0000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	7.8000000e+01	7.4000000e+01	6.9000000e+01	6.9000000e+01	6.9000000e+01	6.9000000e+01
40	8.2000000e+01	8.0000000e+01	8.0000000e+01	8.0000000e+01	7.8000000e+01	7.5000000e+01	6.9000000e+01	6.1000000e+01	6.1000000e+01	6.1000000e+01	6.1000000e+01
41	8.1000000e+01	8.8000000e+01	8.0000000e+01	8.2000000e+01	7.9000000e+01	7.8000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01
42	9.8000000e+01	9.9000000e+01	9.8000000e+01	8.0000000e+01	7.9000000e+01	7.8000000e+01	7.8000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01
43	9.9000000e+01	1.2500000e+02	9.2000000e+01	8.6000000e+01	7.9000000e+01	7.8000000e+01	6.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01
44	9.9000000e+01	9.9000000e+01	8.9000000e+01	8.4000000e+01	7.5000000e+01	6.0000000e+01	6.0000000e+01	4.1000000e+01	4.1000000e+01	4.1000000e+01	4.1000000e+01
45	9.9000000e+01	9.9000000e+01	8.7000000e+01	8.0000000e+01	7.1000000e+01	6.1000000e+01	5.6000000e+01	4.8000000e+01	4.8000000e+01	4.8000000e+01	4.8000000e+01
46	8.7000000e+01	8.0000000e+01	8.5000000e+01	7.9000000e+01	7.4000000e+01	6.8000000e+01	6.2000000e+01	5.7000000e+01	5.7000000e+01	5.7000000e+01	5.7000000e+01
47	8.5000000e+01	9.2000000e+01	8.2000000e+01	8.0000000e+01	7.9000000e+01	6.7000000e+01	6.2000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01	6.0000000e+01
48	9.9000000e+01	8.6000000e+01	8.0000000e+01	7.9000000e+01	7.3000000e+01	5.8000000e+01	6.2000000e+01	6.3000000e+01	6.3000000e+01	6.3000000e+01	6.3000000e+01
49	9.3000000e+01	8.0000000e+01	8.0000000e+01	7.9000000e+01	7.3000000e+01	6.4000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01	5.9000000e+01
50	9.8000000e+01	8.0000000e+01	7.9000000e+01	7.9000000e+01	6.0000000e+01	5.9000000e+01	5.9000000e+01	5.0000000e+01	5.0000000e+01	5.0000000e+01	5.0000000e+01

图6 水深数据图

其中水深数据范围为 99E-122E，2N-25N，空间分辨率为 5 分，深度数据单位为米。

## 1.2 需要解决的问题

### 问题一：潮汐调和常数的提取

从问题描述中可知，需要首先读取卫星高度计数据，得到南海海区各轨道观测点的海表面高度异常数据的时间序列，对观测位置点和上升轨道和下降轨道编号编号，数据预处理，剔除掉观测次数在 110 次以下的观测点；然后计算各分潮幅角，利用最小二乘法计算各分潮的调和常数  $H$  和  $g$ ；最后需要进行调和常数验证与分析评价。

### 问题二：正压潮和内潮的分离

通过问题一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，需要 (1) 沿轨道作图，展示潮汐调和常数在沿轨道方向，在空间上有细结构 (2) 对各分潮调和常数  $H$  和  $g$  建立多项式拟合数学模型 (3) 利用多项式对  $M_2$  的潮汐调和常数  $H$  和  $g$  进行拟合 (4) 确定最佳次数，实现正压潮和内潮的分离，提取内潮。

### 问题三：插值出各主要分潮的同潮图

根据题目讲述，需要首先选择适合离散点数据的插值方法；然后建立该插值方法的数学模型；然后再利用该插值方法制作出各主要分潮的同潮图；最后对同潮图的评价以及评价结果的分析，利用潮汐验潮点的调和常数  $H$  和  $g$  给予评价或检验，并给出评价结果的分析或评价。

### 问题四：确定特定函数的最佳次数

根据题目讲述，需要解决 (1) 在沿轨道的潮汐调和常数插值的过程中，利用的特定的函数进行插值，确定出利用的该特定函数的最佳次数？最佳次数是否为最高次数？ (2) 上述结论对第 3 问的启示或帮助。

---

## 2.模型基础假设

1. 假设各种海平面修正包含在余水位  $S_0$  中；
2. 假设实际水位是由主要分潮 ( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 调和叠加的结果；
3. 假设地理事物、属性空间分布上互为相关，存在集聚、随机、规则分布。
4. 假设分潮振幅虽与地点有关，也取决于海洋本身的几何形状及动力学性质
5. 假设在后续处理过程中本文所求分潮调和常数为正确结果。

## 3.符号说明

符号	符号说明
$J$	分潮个数
$M_2$	太阴主要半日分潮
$S_2$	太阳主要半日分潮
$K_1$	太阴太阳合成全日分潮
$O_1$	太阴主要全日分潮
$h$	任一点潮位
$S_0$	余水位
$f_j$	交点因子
$u_i$	交点订正角
$h_j$	调和常数（振幅）
$g_j$	调和常数（迟角）
$\sigma_i$	分潮角速度
$v_i$	分潮初相位
$\hat{h}$	计算的潮位

## 4. 问题一 潮汐调和常数的提取

### 4.1 问题 1.1 潮汐调和常数的提取

#### 4.1.1 问题分析

从问题描述中可知,需要利用给的高度计数据以及各种参数数值提取各分潮的调和常数 $H$ 和 $g$ 。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。

第一步:数据预处理。首先读取高度计点文件,按顺序编号;然后将点可视化后,按照卫星运行规律,确定上升轨道和下降轨道;最后对上升轨道和下降轨道编号,并确定轨道的初始点号和结束点号。

第二步:计算分潮幅角及各参数。

第三步:利用最小二乘法计算各分潮的调和常数 $H$ 和 $g$ 。

第四步:调和常数评价。

#### 4.1.2 问题求解

依照上述分析,将问题分为四步求解:

##### (1) 数据预处理

首先读取高度计点文件,对 4520 个位置点按顺序编号为 1-4520,可视化显示为图 4-1;根据各分潮分离的最小周期选取了观测次数在 110 次以上的观测位置点,剔除掉了 211 个观测位置点,然后对剔除后的点可视化显示为图 4-2。

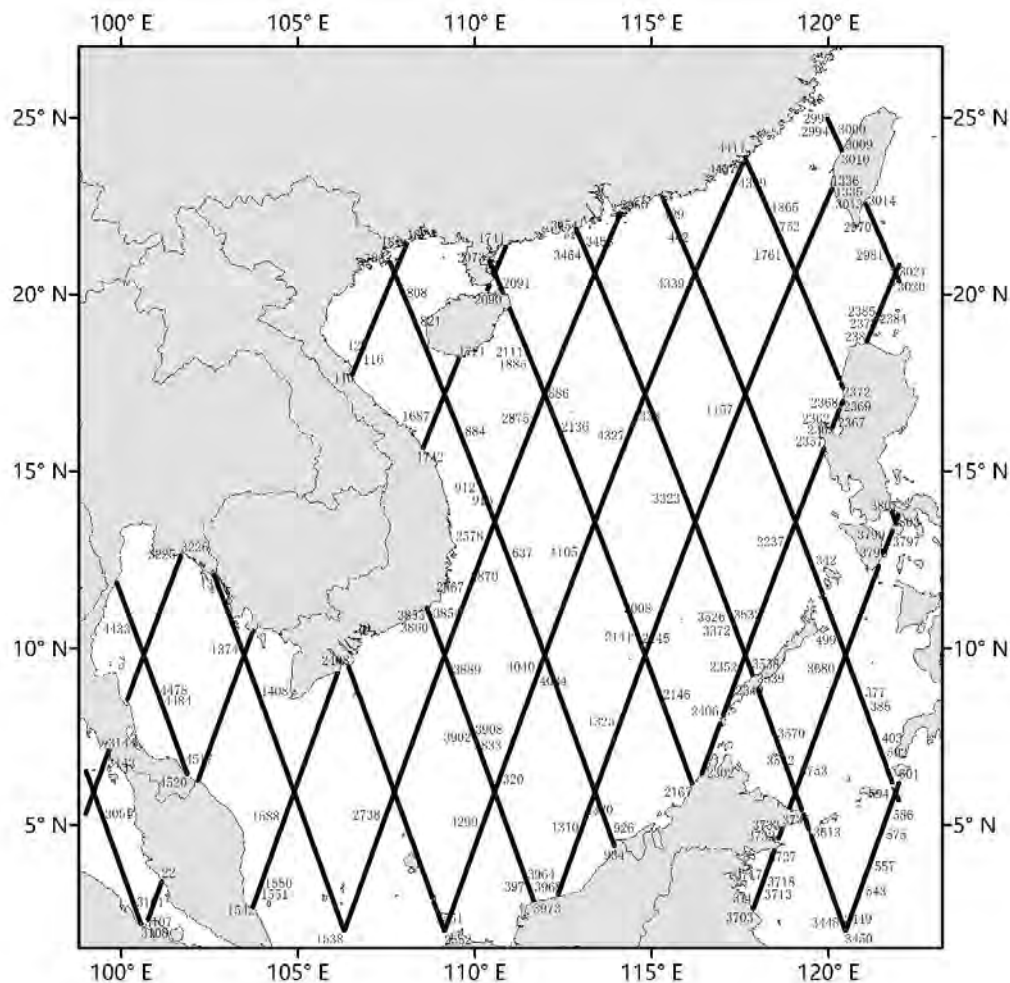


图 4-1 点编号图



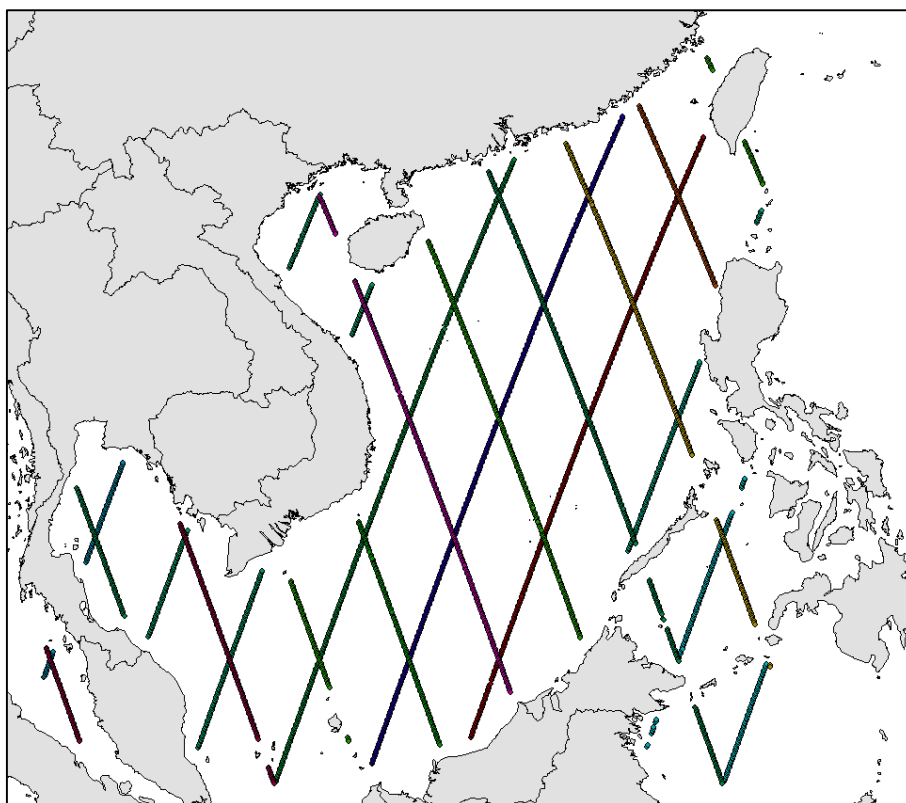


图 4-2 剔除后的观测位置点

然后按照卫星运行规律，确定上升轨道和下降轨道，在南海海区自西向东确定 9 条上升轨道、11 条下降轨道，如图 4-3。

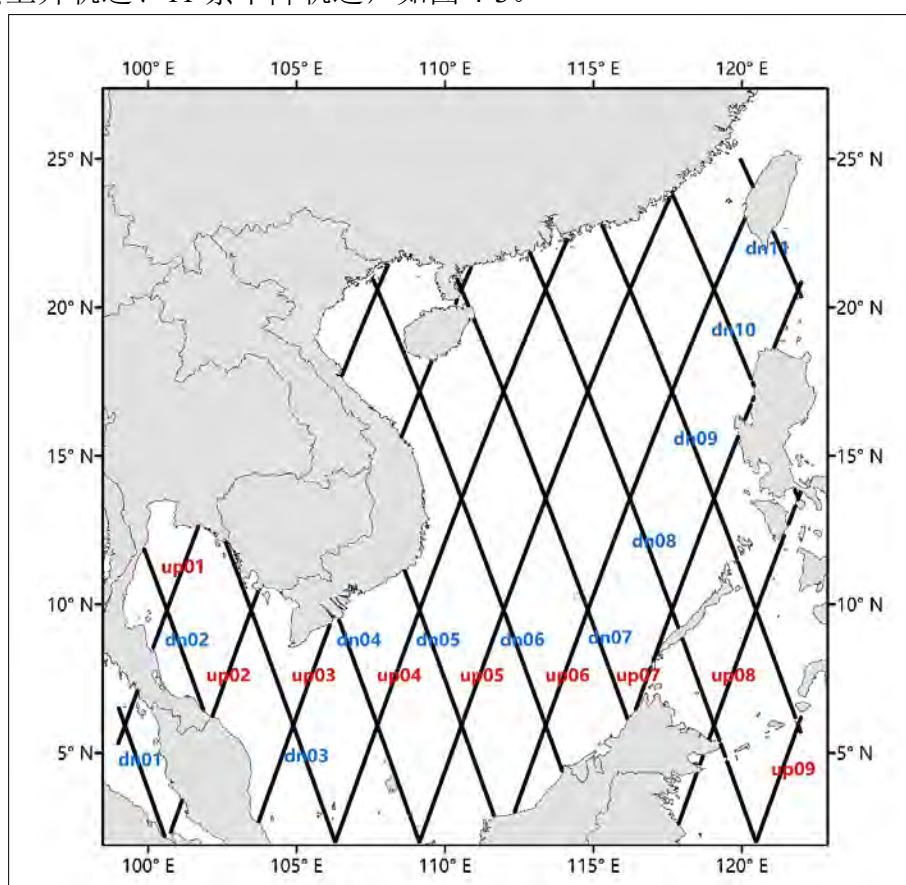


图 4-3 轨道编号图

最后对上升轨道和下降轨道进行编号，并确定轨道的初始点号和结束点号，如表 4-1。

表 4-1 轨道划分点标号

轨道划分点标号	
上升轨道	下降轨道
up01.(3109 3226)	dn01.(3022 3108)
up02.(1 182)	dn02.(4412 4520)
up03.(1539 1742)	dn03.(1337 1538)
up04.(2553 2960)	dn04.(2407 2552)
up05.(3974 4411)	dn05.(3808 3973)
up06.(935 1336)	dn06.(602 934)
up07.(2168 2406)	dn07.(1879 2167)
up08.(3614 3807)	dn08.(3227 3613)
up09.(519 601)	dn09.(183 518)
	dn010.(1743 1878)
	dn011.(2961 3021)

## (2) 计算分潮幅角 $V$

Doodson 用幅角数代表分潮，Doodson 分潮幅角  $V$  的计算过程<sup>[1]</sup>如下：

六个天文变量都随时间线性变化，

$$\begin{cases} \tau = \sigma_{\tau}t + \tau_0 & s = \sigma_s t + s_0 \\ h = \sigma_h t + h_0 & p = \sigma_p t + p_0 \\ N = \sigma_N t + N_0 & p' = \sigma_{p'} t + p'_0 \end{cases} \quad (4-1)$$

其中， $\tau$  为平太阳时角； $s$  为平太阴黄经； $h$  为平太阳黄经； $N$  为升交点平均黄经； $p$  为近地点平均黄经； $p'$  为近日点平均黄经； $s_0$ 、 $h_0$ 、 $p_0$ 、 $N_0$ 、 $p'_0$  是平太阳时  $t = 0$  时刻的平太阴、平太阳、近地点、升交点和近日点的平均黄经。

分潮幅角  $V$  可以写成：

$$\begin{aligned} V &= \mu_1 \tau + \mu_2 s + \mu_3 h + \mu_4 p + \mu_5 N + \mu_6 p' \\ &= (\mu_1 \sigma_{\tau} + \mu_2 \sigma_s + \mu_3 \sigma_h + \mu_4 \sigma_p + \mu_5 \sigma_N + \mu_6 \sigma_{p'})t + \\ &\quad (\mu_1 \tau_0 + \mu_2 s_0 + \mu_3 h_0 + \mu_4 p_0 + \mu_5 N_0 + \mu_6 p'_0) \\ &= \sigma t + V_0 \end{aligned} \quad (4-2)$$

其中  $\sigma = \mu_1 \sigma_{\tau} + \mu_2 \sigma_s + \mu_3 \sigma_h + \mu_4 \sigma_p + \mu_5 \sigma_N + \mu_6 \sigma_{p'}$  为分潮的角速率， $V_0 = \mu_1 \tau_0 + \mu_2 s_0 + \mu_3 h_0 + \mu_4 p_0 + \mu_5 N_0 + \mu_6 p'_0$  为分潮的初位相。

式中  $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6$  为 Doodson 数，具体数据值见表 1，其余参数具体参数值为：

$$\left\{ \begin{array}{l} \tau = 14.49205211 \\ s = 0.54901653 \\ h' = 0.04106864 \\ p = 0.00464183 \\ N' = 0.00220641 \\ p' = 0.00000196 \end{array} \right. \quad (\text{单位: 度/平太阳时})$$

$$\left\{ \begin{array}{l} s_0 = 277.02 + 129.3848(Y - 1900) + 13.1764(n + i + \frac{t}{24}) \\ h'_0 = 280.19 - 0.2387(Y - 1900) + 0.9857(n + i + \frac{t}{24}) \\ p_0 = 334.39 + 40.6625(Y - 1900) + 0.1114(n + i + \frac{t}{24}) \\ N'_0 = 100.84 + 19.3282(Y - 1900) + 0.0530(n + i + \frac{t}{24}) \\ p'_0 = 281.22 + 0.0172(Y - 1900) + 0.00005(n + i + \frac{t}{24}) \\ \tau_0 = 15t - s + h' \end{array} \right. \quad (4-3)$$

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。

表 4-2 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						$\mu_0$	分潮角速度 单位: 度/平太 阳时	交点因子与订正角	
	$\mu_1$	$\mu_2$	$\mu_3$	$\mu_4$	$\mu_5$	$\mu_6$			$f$	$u$
$S_a$	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

续表 4-2

$K_1$	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	$M_2^3$	$3M_2$

由上述过程，即可计算出 Doodson 分潮幅角  $V$ 。

## (2) 计算各分潮的交点因子 $f_j$ 与交点订正角 $u_j$

分潮的系数决定着分潮的振幅，每个分潮系数在 18.61 年的平均值，应与该期间分潮的实际平均振幅  $H$  成正比。分潮某时刻的系数也应与对应时刻的实际振幅  $R$  成正比。分潮某时刻的系数与 18.61 年的平均值之比便叫做交点因子  $f$ ，近似等于分潮某时刻的实际振幅与分析资料时段内的平均振幅之比

$$\frac{R}{H} = \frac{G_2}{[G_2 \cos u]_0} = f \quad (4-4)$$

具体对于单纯分潮  $M_2$  等分潮的交点因子  $f$  可写成：

$$f = \frac{\cos^4 \frac{I}{2}}{0.9154} \quad (4-5)$$

由  $\frac{R}{H} = \frac{G_2}{[G_2 \cos u]_0} = f$  是式可以看出，分潮的表达式可以写成

$$\begin{cases} R \cos(V + u) \\ fH \cos(V + u) \end{cases} \quad (4-6)$$

三者叠加得：

$$\begin{aligned} & \cos 2\tau - 0.03733 \cos(2\tau + N) + 0.00052 \cos(2\tau + 2N) \\ &= \cos 2\tau - 0.03733 \cos 2\tau \cos N + 0.03733 \sin 2\tau \sin N \\ & \quad + 0.00052 \cos 2\tau \cos 2N - 0.00052 \sin 2\tau \sin 2N \end{aligned} \quad (4-7)$$

令  $f \cos(2\tau + N)$  等于上式，对比  $\cos 2\tau$ ， $\sin 2\tau$  的系数，得：

$$\begin{cases} f \cos u = 1 - 0.03733 \cos N + 0.00052 \cos 2N \\ f \sin u = -0.03733 \sin N + 0.00052 \sin 2N \end{cases} \quad (4-8)$$

从而求出交点因子  $f$  和交点订正角  $u$  的大小。

11 个基本分潮  $M_m$ 、 $M_f$ 、 $O_1$ 、 $K_1$ 、 $J_1$ 、 $OO_1$ 、 $M_2$ 、 $K_2$ 、 $M_3$ 、 $M_1$ 、 $L_2$  的  $f$ 、 $u$  是给定的，如下表，其他各主要分潮的  $f$ 、 $u$  可依据 11 个基本分潮的  $f$ 、 $u$  导出。

表 4-3 基本分潮的交点因子  $f$  和交点订正角  $u$  值表

	$f$	$u$
$M_m$	$1.0000-0.1300\cos N+0.0013\cos 2N$	0
$M_f$	$1.0429+0.4135\cos N-0.0040\cos 2N$	$-23^\circ.74\sin N+2^\circ.68\sin 2N-0^\circ.38\sin 3N$
$O_1$	$1.0089+0.1871\cos N-0.0147\cos 2N+0.0014\cos 3N$	$10^\circ.80\sin N-1^\circ.34\sin 2N+0^\circ.19\sin 3N$
$K_1$	$1.0060+0.1150\cos N-0.0088\cos 2N+0.0006\cos 3N$	$-8^\circ.86\sin N+0^\circ.68\sin 2N-0^\circ.07\sin 3N$
$J_1$	$1.0129+0.1676\cos N-0.0170\cos 2N+0.0016\cos 3N$	$-12^\circ.94\sin N+1^\circ.34\sin 2N-0^\circ.19\sin 3N$
$OO_1$	$1.1027+0.6504\cos N+0.0317\cos 2N-0.0014\cos 3N$	$-36^\circ.68\sin N+4^\circ.02\sin 2N-0^\circ.57\sin 3N$
$M_2$	$1.0004-0.0373\cos N+0.0003\cos 2N$	$-2^\circ.14\sin N$
$K_2$	$1.0241+0.2863\cos N+0.0083\cos 2N-0.0015\cos 3N$	$-17^\circ.74\sin N+0^\circ.68\sin 2N-0^\circ.04\sin 3N$
$M_3$	$1+1.5(f-1)=-0.5+1.5f$	$1.5u$
$M_1$	$f\cos u=2\cos p+0.4\cos(p-N)$	$f\sin u=\sin p+0.2\sin(p-N)$
$L_2$	$f\cos u=1.0000-0.2505\cos 2p-0.1103\cos(2p-N)-0.0156\cos(2p-2N)-0.0366\cos N+0.0047\cos(2p+N)$	$f\sin u=-0.2505\sin 2p-0.1103\sin(2p-N)-0.0156\sin(2p-2N)-0.0366\sin N+0.0047\sin(2p+N)$

### (3) 利用最小二乘法计算各分潮的调和常数 $H$ 和 $g$

有前面分析可知实际水位可以看作是许多调和分潮迭加的结果，在实际进行潮汐调和与分析时，只能选取其中有限个比较主要的分潮，假设我们选取了  $J$  个分潮，对某一确定的分潮可将潮位的表达式写为<sup>[7]</sup>：

$$\hat{h}_j = f_j H_j \cos(\sigma_j t + V_{0j} + u_j - g_j) \quad (4-9)$$

可化为如下形式：

$$\hat{h}_j = f_j \cos(\sigma_j t + V_{0j} + u_j) H_j \cos g_j + f_j \sin(\sigma_j t + V_{0j} + u_j) H_j \sin g_j \quad (4-10)$$

其中， $f_j$ 、 $\sigma_j$ 、 $t$ 、 $u_j$ 、 $V_{0j}$  均为已知或可通过简单计算得出。对我们选取的  $J$  个分潮则可将潮位的表达式写为：

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

本文我们选用了最小二乘法获取各分潮的调和常数，上述取得计算的潮位为  $\hat{h}$  用它来逼近实测的潮位  $h$  按最小二乘法原理，必须使以下  $\Delta$  值最小：

$$\Delta = \hat{h} - h \quad (4-11)$$

那么如果在  $n$  个时刻  $t = t_1, t_2, \dots, t_n$ ，有  $n$  个潮高观测值  $h = h_1, h_2, \dots, h_n$ ，就可

以建立如下由  $n$  个方程构成的方程组：

$$\begin{cases} S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_1 + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_1 + V_0 + u_j) = h_1 \\ S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_2 + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_2 + V_0 + u_j) = h_2 \\ \vdots \\ S_0 + \sum_{j=1}^J x_j f_j \cos(\sigma_j t_n + V_0 + u_j) + \sum_{j=1}^J y_j f_j \sin(\sigma_j t_n + V_0 + u_j) = h_n \end{cases} \quad (4-12)$$

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：

$$\begin{cases} S_0 + a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m = h_1 \\ S_0 + a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m = h_2 \\ \vdots \\ S_0 + a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m = h_n \end{cases} \quad (4-13)$$

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，依然用我们选取的最小二乘法来求解。

与最小二乘法上述相同，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：

$$\Delta = \sum_{i=1}^n (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{im}x_m + S_0 - h_i)^2 \quad (4-14)$$

根据多元函数微分的理论，这要求  $\Delta$  相对于各未知数的偏导数均为 0，即：

$$\frac{\partial \Delta}{\partial S_0} = \frac{\partial \Delta}{\partial x_1} = \frac{\partial \Delta}{\partial x_2} = \cdots = \frac{\partial \Delta}{\partial x_m} = 0 \quad (4-15)$$

$$\text{将 } \Delta = \sum_{i=1}^n (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{im}x_m + S_0 - h_i)^2 \text{ 代 } \frac{\partial \Delta}{\partial S_0} = \frac{\partial \Delta}{\partial x_1} = \frac{\partial \Delta}{\partial x_2} = \cdots = \frac{\partial \Delta}{\partial x_m} = 0,$$

可得：

$$\begin{cases} \sum_{i=1}^n a_{i1} (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{im}x_m - h_i) = 0 \\ \sum_{i=1}^n a_{i2} (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{im}x_m - h_i) = 0 \\ \cdots \cdots \cdots \\ \sum_{i=1}^n a_{im} (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{im}x_m - h_i) = 0 \end{cases} \quad (4-16)$$

将矛盾方程转化为矩阵形式为：



[illegible]

其中:

$$\left\{ \begin{aligned} c_{11} &= \sum_{i=1}^n a_{i1} a_{i1} \mathcal{C}_{12} = \sum_{i=1}^n a_{i1} a_{i2}, \cdots, c_{1m} = \sum_{i=1}^n a_{i1} a_{im} \\ c_{21} &= \sum_{i=1}^n a_{i2} a_{i1} \mathcal{C}_{12} = \sum_{i=1}^n a_{i2} a_{i2}, \cdots, c_{2m} = \sum_{i=1}^n a_{i2} a_{im} \\ &\dots\dots\dots \\ c_{m1} &= \sum_{i=1}^n a_{im} a_{i1} \mathcal{C}_{m2} = \sum_{i=1}^n a_{im} a_{i2}, \cdots, c_{mm} = \sum_{i=1}^n a_{im} a_{im} \end{aligned} \right. \quad (4-18)$$

$$\left\{ \begin{array}{l} f_1 = \sum_{i=1}^n a_{i1} h_i \\ f_2 = \sum_{i=1}^n a_{i2} h_i \\ \dots\dots\dots \\ f_m = \sum_{i=1}^n a_{im} h_i \end{array} \right. \quad (4-19)$$

按线性方程组的求解方法, 则对称系数矩阵  $A$  为:

$$A = \begin{bmatrix} c_{11}, c_{12}, \cdots c_{1m} \\ c_{21}, c_{22}, \cdots c_{2m} \\ \cdots \cdots \cdots \cdots \cdots \cdots \\ c_{m1}, c_{m2}, \cdots c_{mm} \end{bmatrix} \quad (4-20)$$

变量矩阵为:  $X = [x_1, x_2 \cdots x_m]^T$ , 常数矩阵为:  $f = [f_1, f_2 \cdots f_m]^T$  则有:

$$AX + A^T X = 2A_X \quad , \text{ 最小二乘的目标为:}$$

$$\min (\|Ax - f\|_2)^2 \quad (4-21)$$

问题转化为多变量的二次优化问题，很容易想到的是对变量进行求导，代入展开为[8]:

$$\begin{aligned} \left( \|A_X - f\|_2 \right)^2 &= (A_X - f)^T (A_X - f) \\ &= X^T A^T A_X - f^T A_X - X^T A^T f + f^T f \end{aligned} \quad (4-22)$$

因此有:

$$\frac{\partial (\|A_X - f\|_2)^2}{\partial x} = 2A^T A_X - 2A^T b \quad (4-23)$$

最后得到:

$$X = (A^T A)^{-1} A^T b \quad (4-24)$$

可解得  $x_1, x_2, \dots, x_m$  的值 又由公式:

$$x_i = H_i \cos g_i + H_j \sin g_i \quad (4-25)$$

进而可计算出各分潮的调和常数  $H$  和  $g$ 。

#### (4) 调和常数验证与分析评价

为了检验上述调和分析的结果 本文在轨道交点处将上行轨和下行轨的调和常数做了比较。

##### 1.高度计资料序列选取。

从参考资料得知<sup>[2]</sup>，卫星高度计数据存在高度计取样过程中的高频混淆，因此我们难以用常规的潮汐分析技术分离高度计资料中的潮汐分量，

按照 Rayleigh 准则，两个频率的潮汐信号实现可靠分离所需要的时间  $T$  应满足：

$$\left| \frac{1}{T_i} - \frac{1}{T_j} \right| T \geq 1 \quad (4-26)$$

即：

$$T \geq \left| \frac{T_i T_j}{T_j - T_i} \right| \quad (4-27)$$

其中  $T_i$  和  $T_j$  为所比较的 2 个分潮的周期。由此根据表 4-4 中数值计算得出8个主要分潮基本的分辨时间<sup>[3]</sup>，见表4-5。

表 4-4 主要分潮的周期及在 T/P 采样的混淆周期

分潮	分潮周期(h)	混淆周期(d)
K <sub>1</sub>	23.934470	173.2
O <sub>1</sub>	25.819342	45.7
P <sub>1</sub>	24.065890	88.9
Q <sub>1</sub>	26.868357	69.3
M <sub>2</sub>	12.420601	62.1
S <sub>2</sub>	12.000000	58.7
N <sub>2</sub>	12.658348	49.5
K <sub>2</sub>	11.967235	86.6

表 4-5 主要分潮基本的分辨时间

	K <sub>1</sub>	O <sub>1</sub>	P <sub>1</sub>	Q <sub>1</sub>	M <sub>2</sub>	S <sub>2</sub>	N <sub>2</sub>	K <sub>2</sub>
K <sub>1</sub>	0	0.17	0.50	0.32	0.27	0.24	0.19	0.47
O <sub>1</sub>		0	0.26	0.37	0.47	0.56	1.63	0.27
P <sub>1</sub>			0	0.86	0.56	0.47	0.31	9.18
Q <sub>1</sub>				0	1.63	1.05	0.47	0.95
M <sub>2</sub>					0	2.97	0.67	0.60
S <sub>2</sub>						0	0.86	0.50
N <sub>2</sub>							0	0.32
K <sub>2</sub>								0

根据问题 1 中要求，计算  $K_1$ 、 $M_2$ 、 $O_1$ 、 $S_2$  四个主要分潮，从表 2 中可知，实现四个主要分潮信号可靠分离至少需要 2.97 年的数据，因此从所给的星下点观测序列数据至少要有约 110 天的观测数据，故将观测序列长度小于 110 的点剔除。

## 2.轨道交点调和常数的比较。

T/P 卫星自西向东围绕地球运行，我们将由南向北运行为上升轨道，由北向南运行为下降轨道。为了比较模型自身精度，选取上行轨道和下降轨道的交点，计算差值，进行比较。

交叉点位置计算步骤：

1.选择一对上升和下降轨道上的四个点，如图 4-4。

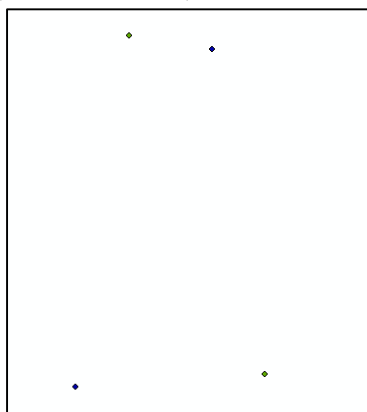


图 4-4 四点图

2.分别对同一个轨道两个点的经纬度进行插值，近似得到两条线段，如图 4-5。

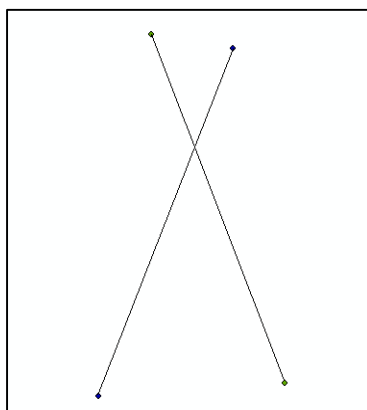


图 4-5 插值线段图

3.线段交点即为交叉点位置，如图 4-6。

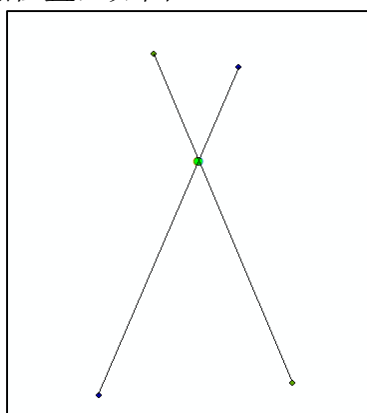


图 4-6 交叉位置图

4.以上工作在 ARCGIS 中利用点集转线工具配合创建要素工具提取交点即可实现。

共提取 24 个交叉点，编号 X1~X24，所提取的交点如图 4-7 所示。

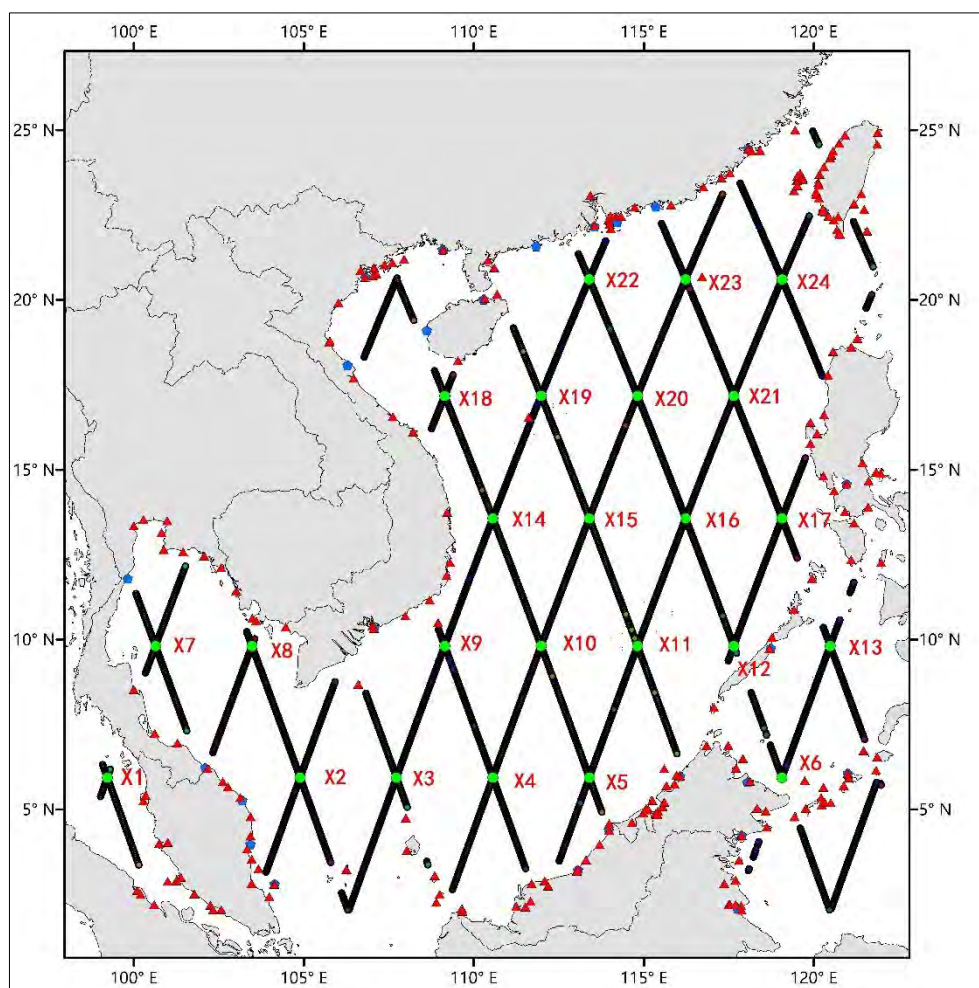


图 4-7 提取的交点图

图中蓝色点为 World sea 验潮站，红色点为 East China sea 验潮站，绿色点为交叉点。

交叉点经纬度信息如表 4-6:

表 4-6 交叉点经纬度

点位	经度	纬度
X1	99.2222977	5.9373498
X2	100.6389999	9.8116798
X3	103.4749985	9.8114901
X4	110.561	5.93782
X5	107.7269974	5.9377899
X6	109.1439972	9.81213
X7	110.5619965	13.5712996
X8	109.1449966	17.1763992
X9	111.9789963	17.1779003
X10	114.8130035	17.1776009
X11	113.3960037	20.6079998
X12	116.2310028	20.6082001
X13	117.6490021	17.1751995
X14	119.0670013	20.6063004
X15	119.0660019	13.5705004

续表 4-6

X16	120.4830017	9.8134604
X17	119.0650024	5.9380598
X18	110.560997	5.93782
X19	113.3970032	5.9370098
X20	111.9789963	9.8137598
X21	114.814003	9.8108702
X22	117.6480026	9.8121405
X23	113.3960037	13.5712996
X24	116.2320023	13.5682001

根据轨道上交叉点相邻点对进行插值得到交叉点位调和常数,以中间四个点的  $M_2$  分潮为例进行交叉点验证,结果如表 4-7 所示。

表 4-7  $M_2$  分潮交叉点验证结果

点号	位置		上升轨		下降轨		$\Delta H$	$\Delta g$
	Lon	Lat	H	g	H	g		
X9	111.9789	17.1779	0.161	341.1	0.178	342.61	0.017	1.51
X10	114.8130	17.1776	0.1825	320.29	0.182	320.31	0.0005	0.02
X14	119.0670	20.6063	0.1785	311.695	0.1795	310.82	0.001	0.875
X15	119.0660	13.5705	0.1545	306.455	0.165	303.685	0.010	2.77
RMS	-	-	-	-	-	-	0.017	2.22

看出在四个点上的  $M_2$  分潮调和结果偏差很小。同理我们可以验证其他点位。

同验潮站对比验证：验潮站选取，根据参考资料<sup>[2]</sup>，高度计资料调和分析的结果和验潮站数据在水深 200 米至 500 米的海区内符合较好，因此利用附件资料中的水深图作图，如图 4-8 所示，图中蓝色为 200 至 500 米水深范围。根据水深、验潮站的分布和卫星星下点轨迹的空间关系，选择星下点水深在 200 米至 500 米范围内，距离星下点 1/3 度内的验潮站进行验证。

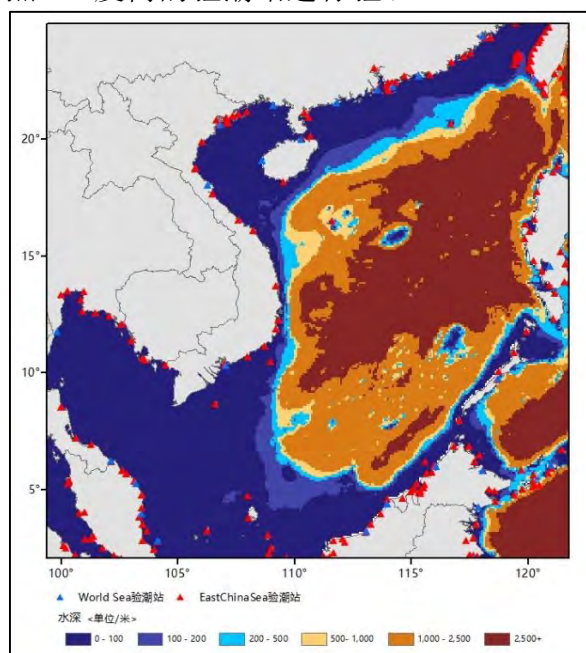


图 4-8 验潮站分布及南海海域地形

共选取 11 个符合条件的验潮站，验证结果如下：  
表 4-8 K1 分潮验潮站验证结果

K1	验潮站		星下点		距离	实测值		计算值		差值	
序号	Lon	Lat	Lon	Lat	Km	H(m)	g(°)	H(m)	g(°)	ΔH(m)	Δg(°)
1	120.417	17.783	120.2647	17.7182	17.67	0.191	316.9	0.216	323.18	-0.025	-6.28
2	121.1	18.617	121.1108	18.7199	11.446	0.122	291.2	0.135	286.42	-0.013	4.78
3	111.617	16.55	111.7067	16.4985	11.14	0.268	305.44	0.255	311.92	0.013	-6.48
4	111.617	16.55	111.7265	16.5481	11.6	0.268	305.44	0.25	312.57	0.018	-7.13
5	120.267	22.617	119.9336	22.6108	34.2	0.156	295.36	0.216	285.65	-0.06	9.71
6	121.283	18.867	109.7478	11.4302	26	0.109	276	0.157	282.72	-0.048	-6.72
7	113.983	4.583	121.1515	18.8187	14.854	0.35	319	0.356	310.83	-0.006	8.17
8	121.467	6.733	113.7284	5.0191	54	0.267	306.9	0.347	322.99	-0.08	-16.09
9	119.433	4.783	121.5296	6.961	26.14	0.152	282.9	0.2	304.23	-0.048	-21.33
10	117.7	2.917	119.3893	5.0402	28.85	0.19	273	0.184	89.45	0.006	183.55
RMS										0.04	58.963

表 4-9 M2 分潮验潮站验证结果

M2	验潮站		星下点		距离	实测值		计算值		差值	
序号	Lon	Lat	Lon	Lat	Km	H(m)	g(°)	H(m)	g(°)	ΔH(m)	Δg(°)
1	120.417	17.783	120.2647	17.7182	17.67	0.077	228	0.061	54.11	0.016	173.89
2	121.1	18.617	121.1108	18.7199	11.446	0.19	195.7	0.212	5.89	-0.022	189.81
3	111.617	16.55	111.7067	16.4985	11.14	0.168	296.88	0.16	286.03	0.008	10.85
4	111.617	16.55	111.7265	16.5481	11.6	0.168	296.88	0.141	292.22	0.027	4.66
5	120.267	22.617	119.9336	22.6108	34	0.154	236.12	0.129	47.14	0.025	188.98
6	109.2	11.883	109.7478	11.4302	26	0.245	177.3	0.222	6.71	0.023	170.59
7	121.283	18.867	121.1515	18.8187	14.854	0.17	335	0.188	323.57	-0.018	11.43
8	113.983	4.583	113.7284	5.0191	54	0.196	242.9	0.285	301	-0.089	-58.1
9	121.467	6.733	121.5296	6.961	26.14	0.362	147	0.267	346.03	0.095	-199.03
10	119.433	4.783	119.3893	5.0402	28.85	0.79	169	0.605	339.88	0.185	-170.88
RMS										0.074	142.667



表 4-10 O1 分潮验潮站验证结果

O1	验潮站		星下点		距离	实测值		计算值		差值	
序号	Lon	Lat	Lon	Lat	Km	H(m)	g(°)	H(m)	g(°)	$\Delta H(m)$	$\Delta g(^{\circ})$
1	120.417	17.783	120.2647	17.7182	17.67	0.185	264.1	0.207	77.66	-0.022	186.44
2	121.1	18.617	121.1108	18.7199	11.446	0.132	248.4	0.12	56.04	0.012	192.36
3	111.617	16.55	111.7067	16.4985	11.14	0.306	304.3	0.222	88.29	0.084	216.01
4	111.617	16.55	111.7265	16.5481	11.6	0.306	304.3	0.211	89.43	0.095	214.87
5	120.267	22.617	119.9336	22.6108	34.2	0.152	248.86	0.172	59.42	-0.02	189.44
6	121.283	18.867	109.7478	11.4302	26	0.133	234.3	0.155	57.42	-0.022	176.88
7	113.983	4.583	121.1515	18.8187	14.854	0.29	251	0.305	87.69	-0.015	163.31
8	121.467	6.733	113.7284	5.0191	54	0.223	265.1	0.302	87.57	-0.079	177.53
9	119.433	4.783	121.5296	6.961	26.14	0.159	260.1	0.226	84.07	-0.067	176.03
10	117.7	2.917	119.3893	5.0402	28.85	0.14	244	0.172	47.41	-0.032	196.59
RMS										0.054	189.628

表 4-11 S2 分潮验潮站验证结果

S2	验潮站		星下点		距离	实测值		计算值		差值	
序号	Lon	Lat	Lon	Lat	Km	H(m)	g(°)	H(m)	g(°)		
1	120.417	17.783	120.2647	17.7182	17.67	0.032	241.2	0.028	58.38	0.004	182.82
2	121.1	18.617	121.1108	18.7199	11.446	0.068	216.8	0.105	27.77	-0.037	189.03
3	111.617	16.55	111.7067	16.4985	11.14	0.077	352.9	0.045	316.36	0.032	36.54
4	111.617	16.55	111.7265	16.5481	11.6	0.077	352.9	0.044	338.66	0.033	14.24
5	120.267	22.617	119.9336	22.6108	34.2	0.063	247.6	0.049	35.5	0.014	212.1
6	121.283	18.867	109.7478	11.4302	26	0.102	205.4	0.112	26.86	-0.01	178.54
7	113.983	4.583	121.1515	18.8187	14.854	0.08	10	0.079	6.61	0.001	3.39
8	121.467	6.733	113.7284	5.0191	54	0.118	280.1	0.149	341.04	-0.031	-60.94
9	119.433	4.783	121.5296	6.961	26.14	0.266	211.1	0.172	33.43	0.094	177.67
10	117.7	2.917	119.3893	5.0402	28.85	0.51	216	0.397	28.65	0.113	187.35
RMS										0.051	147.631

验证评价和分析：从验潮站验证结果来看，振幅最大差值 0.095 米，最小差值 0.004 米，平均 RMS 为 0.054 米，符合程度较高；迟角最大差值 216.01 度，最小差值 3.39 度，平均 RMS 为 134 度，偏差较大。

分析：在计算过程中，初始时刻决定了初始相位，影响较大；而  $f$  和  $u$  变化较为缓慢，以 18.6 年为一周期，在计算过程中影响较小。对于迟角，有三种迟角，地方迟角，区时迟角，专用区时迟角，取决于分潮初相位时间系统的选取。模型计算结果振幅相差较小，而迟角偏差较大，说明在计算过程中对于时间的判断和计算存在问题。

根据参考资料，受海底地形影响的潮汐地摩擦系数对于高度计的调和分析结果也有影响，具体表现为在一定深度的范围内，调和分析结果与验潮站相符较好，南海岛礁众多，海底地形复杂，调和分析计算过程中未考虑摩擦系数影响，也是造成迟角偏差大的原因。

## 5. 问题二

### 5.1 问题分析

在问题一中我们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，并利用潮汐验潮点的调和常数  $H$  和  $g$  给予评价或检验。本题目首先要根据所有星下观测点各主要分潮的潮汐调和常数沿轨道作图，然后进行各主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的内潮提取。

第一步：展示内潮对正压潮的调制的细结构。通过问题一我们已经得到所有星下观测位置点各主要分潮（ $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ）的潮汐调和常数。我们以  $M_2$  为例，对  $M_2$  沿 up06 轨道的潮汐调和常数  $H$  和  $g$  进行作图，展示内潮对正压潮的调制细结构。

第二步：建立多项式拟合数学模型。针对各分潮调和常数  $H$  和  $g$  建立多项式拟合数学模型。

第三步：利用多项式对  $M_2$  的潮汐调和常数  $H$  和  $g$  进行拟合。

第四步：展示不同次数的潮汐调和常数  $H$  和  $g$  拟合结果，并对拟合结果进行分析评价。

第五步：确定最佳次数，提取内潮。

### 5.2 模型建立及问题求解

依照上述分析，将问题分为四步求解：

#### （1）展示内潮对正压潮的调制的细结构

我们以  $M_2$  为例，对  $M_2$  的潮汐调和常数  $H$  和  $g$  沿 up06 轨道进行作图，展示内潮对正压潮的调制细结构， $M_2$  分潮振幅  $H$  曲线如图 5-1， $M_2$  分潮迟角  $g$  曲线如图 5-2，由图我们可以看出分潮振幅  $H$  曲线和分潮迟角  $g$  曲线表现出锯齿状的细结构。

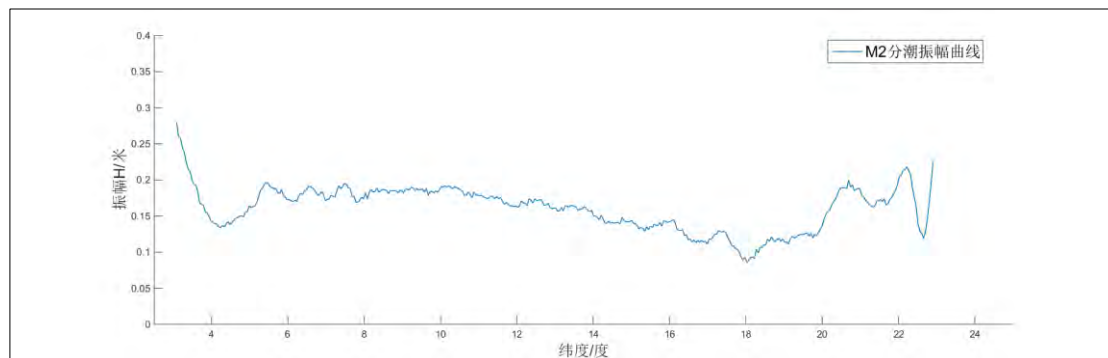


图 5-1  $M_2$  分潮振幅  $H$  曲线图

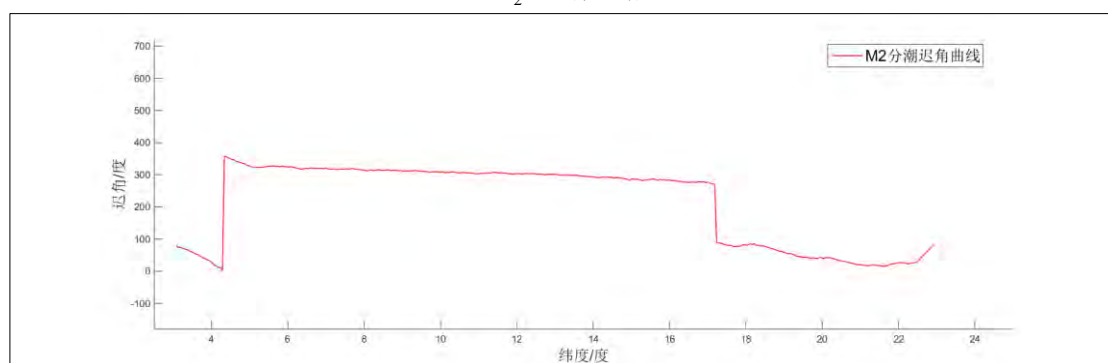


图 5-2  $M_2$  分潮迟角  $g$  曲线图

## (2) 建立多项式拟合数学模型。

本文我们将利用多项式拟合方法拟合问题一中调和分析得出的各分潮调和常数  $H$  和  $g$ ，拟合结果代表正压潮值，通过滤掉正压潮部分来提取内潮海面信号。下面我们将建立多项式拟合数学模型<sup>[4]</sup>。

通过调和分析，得到某一条轨道上的任一观测点（ $x$  和  $y$  分别为点  $P$  的经、纬度）的振幅值  $A$  和迟角值  $G$ 。首先将  $A$ 、 $G$  转化为  $A \cos G$  和  $A \sin G$ ，然后分别对  $A \cos G$ 、 $A \sin G$  沿轨进行  $n$  次多项式拟合（ $n=1,2,3,\dots$ ），最后通过坐标转换得各分潮内潮的海表面振幅和迟角。以拟合  $A \cos G$  为例说明。

设拟合多项式为：

$$\hat{f}(y) = a_0 + a_1 y + a_2 y^2 + \dots + a_n y^n \quad (5-1)$$

构造代价函数：

$$J(a_n) = \sum_{k=1}^N [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - A_k \cos G_k]^2, k=1,2,3,\dots,N \quad (5-2)$$

对式（3）分别求  $a_0, a_1, a_2, \dots, a_n$  的偏导，并使其为 0，

$$\left. \begin{aligned} \sum_{k=1}^N [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - A_k \cos G_k] &= 0; \\ \sum_{k=1}^N y_k [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - A_k \cos G_k] &= 0; \\ \sum_{k=1}^N y_k^2 [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - A_k \cos G_k] &= 0; \\ \dots \\ \sum_{k=1}^N y_k^n [(a_0 + a_1 y_k + a_2 y_k^2 + \dots + a_n y_k^n) - A_k \cos G_k] &= 0; \end{aligned} \right\} \quad (5-3)$$

为便于阐述，引入记号：

$$M_{i,j} = \sum_{k=1}^N y_k^{i+j-2}, b_i = \sum_{k=1}^N y_k^{i-1} A \cos G_k \quad (5-4)$$

其中， $i, j=1,2,\dots,n+1$ 。

式（4）表示为

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

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，并带入式（1）得到正压潮值  $\hat{f}(y)$ ，记  $A_k \cos G_k$  与  $\hat{f}(y)$  之差为：

$$\tilde{f}(y_k) = A_k \cos G_k - \hat{f}(y_k) \quad (5-6)$$

$\tilde{f}(y_k)$  为内潮贡献。同理，对  $A_k \sin G_k$  进行多项式拟合可得

$$\tilde{g}(y_k) = A_k \sin G_k - \hat{g}(y_k) \quad (5-7)$$

通过坐标转换可求得内潮海表面振幅  $\tilde{A}_k$ 、迟角  $\tilde{G}_k$ 。

### (3) 利用多项式对 $M_2$ 的潮汐调和常数 $H$ 和 $g$ 进行拟合。

利用上述第二步的拟合方法,对问题一得到的潮汐调和常数  $H$  和  $g$  沿 9 条上升轨道、11 条下降轨道进行 3 ~ 20 次多项式拟合来提取各分潮内潮海表面信号。

本文我们以  $M_2$  内潮为例 对  $M_2$  的潮汐调和常数  $H$  和  $g$  , 沿 9 条上升轨道、11 条下降轨道采用多项式拟合方法进行不同次数的拟合。

$M_2$  内潮沿 9 条上升轨道、11 条下降轨道进行 3 ~ 20 次多项式拟合得出的振幅  $H$ 、迟角  $g$  的平均绝均差如表 5-1、表 5-1-1 和表 5-1-2, 平均绝均差定义为沿某一轨道进行  $n$  次多项式拟合得到的分潮振幅  $H$  /迟角  $g$  与调和分析得到的分潮振幅  $H$  /迟角差  $g$  的绝对值的均方根。

对沿轨数据按照上述方法进行 3 ~ 20 次多项式拟合, 拟合结果的绝均差的均方根如表 5-1 和的最后一行所示。从表 5-1、表 5-1-1 和表 5-1-2 中可以看出拟合次数越高拟合结果的绝均差不一定越小。从量级上讲, 14、15、16、17、18、19、20 次多项式拟合的振幅绝均差相当, 明显好于其他次数的拟合结果。同样, 对迟角  $g$  的高次多项式拟合得到的绝均差也不一定小, 16 次多项式拟合得到的迟角  $g$  绝均差最小, 反而 20 次多项式拟合结果的绝均差比 11-18 都大。

表 5-1-1 拟合迟角  $g$  的绝均差

轨道	次数								
	3	4	5	6	7	8	9	10	11
up1	131.53	126.70	113.58	117.38	92.97	50.61	64.94	34.76	54.45
up2	187.32	135.48	137.72	136.84	159.92	89.79	97.90	85.69	86.44
up3	87.03	70.45	67.48	77.77	78.37	71.07	88.17	55.30	53.97
up4	99.25	109.27	93.73	87.49	86.31	88.54	81.84	51.09	47.43
up5	130.37	145.98	147.97	113.13	122.49	101.63	101.38	81.89	74.32
up6	67.85	66.71	59.19	25.40	32.18	23.17	23.55	19.10	18.87
up7	76.81	77.32	77.48	77.36	77.33	77.37	77.38	77.96	65.86
up8	6.40	7.10	6.27	4.56	4.82	5.00	3.89	3.63	3.73
up9	1.67	1.65	1.58	1.11	0.90	0.83	0.71	0.66	0.65
dn1	121.67	79.96	57.50	31.82	73.05	63.01	82.02	11.17	73.20
dn2	137.59	75.95	59.47	78.79	91.93	72.82	69.09	87.13	61.60
dn3	167.20	94.59	92.42	80.48	70.78	67.93	52.96	54.48	39.09
dn4	54.70	53.07	25.37	24.37	32.25	24.77	21.04	21.58	7.91
dn5	190.14	132.92	97.69	97.71	104.11	58.00	8.24	28.61	39.98
dn6	102.02	94.12	60.45	58.81	61.83	65.11	63.35	57.37	53.64
dn7	1.83	1.65	1.65	1.54	1.53	1.53	1.43	1.41	1.39
dn8	31.94	7.49	30.29	7.88	6.66	5.72	6.32	5.99	5.10
dn9	34.02	30.23	44.20	39.91	30.48	29.27	25.69	25.43	25.00
dn10	76.57	70.37	73.09	67.95	66.51	77.64	62.32	61.99	62.05
dn11	5.67	48.75	49.07	2.19	2.03	2.03	49.26	1.53	1.79
RMS	104.09	84.22	76.32	70.44	74.08	58.91	59.56	48.57	47.73

表 5-1-2 拟合迟角  $g$  的绝均差

轨道	12	13	14	15	16	17	18	19	20
up1	34.62	34.62	59.82	59.00	58.86	52.66	90.90	79.19	91.05
up2	94.10	62.81	53.11	61.50	61.24	61.28	52.33	21.89	34.58
up3	66.13	69.58	72.50	74.46	67.68	60.30	45.03	51.15	33.30
up4	66.89	46.39	50.53	38.01	30.40	28.52	26.80	41.68	42.46
up5	80.48	79.76	71.61	60.93	62.50	63.62	54.11	56.99	50.39
up6	22.83	13.80	19.25	23.00	22.98	19.55	7.48	23.00	18.93
up7	43.23	43.23	35.31	43.29	43.20	35.28	35.27	43.31	50.00
up8	3.17	3.00	3.08	2.91	2.79	2.82	2.75	2.63	2.66
up9	0.63	0.59	0.50	0.50	0.50	0.50	1.00	0.50	0.50
dn1	39.98	40.05	7.46	8.04	39.48	55.35	66.32	67.57	66.97
dn2	61.58	59.88	71.84	71.35	65.01	62.73	64.47	62.55	64.05
dn3	33.77	41.54	33.94	34.06	29.99	37.53	37.44	32.31	32.33
dn4	21.11	6.86	22.06	19.22	2.97	4.90	2.73	2.22	1.61
dn5	51.15	54.38	39.67	28.42	5.21	61.05	93.79	109.20	92.41
dn6	51.72	50.79	39.07	34.12	33.88	27.55	30.57	30.18	23.24
dn7	1.33	1.33	1.33	1.27	1.25	1.24	1.23	1.23	1.23
dn8	4.79	5.13	4.91	4.19	4.17	4.48	4.31	3.62	3.45
dn9	24.72	19.64	24.81	20.23	20.10	24.63	19.97	19.83	19.86
dn10	57.06	56.85	93.51	67.44	52.16	53.44	61.12	58.19	85.81
dn11	6.44	2.21	48.84	67.86	47.95	1.89	65.74	68.36	47.90
RMS	46.89	42.85	46.20	43.94	40.21	40.66	48.04	48.78	48.26

表 5-2-1 拟合振幅  $H$  的绝均差

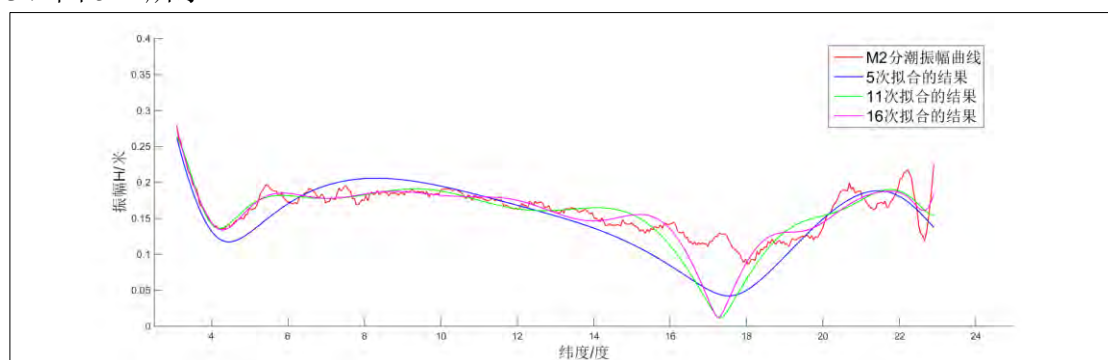
轨道	次数								
	3	4	5	6	7	8	9	10	11
up1	0.07	0.04	0.04	0.04	0.03	0.03	0.02	0.02	0.02
up2	0.06	0.05	0.04	0.03	0.03	0.02	0.02	0.02	0.02
up3	0.19	0.17	0.16	0.20	0.19	0.22	0.22	0.21	0.21
up4	0.10	0.09	0.08	0.06	0.06	0.06	0.06	0.06	0.06
up5	0.10	0.08	0.08	0.08	0.07	0.05	0.05	0.06	0.06
up6	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03
up7	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
up8	0.05	0.05	0.03	0.03	0.03	0.02	0.02	0.02	0.02
up9	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00
dn1	0.14	0.17	0.14	0.15	0.14	0.14	0.13	0.13	0.12
dn2	0.04	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.02
dn3	0.03	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
dn4	0.08	0.08	0.07	0.07	0.06	0.06	0.06	0.05	0.05
dn5	0.08	0.08	0.09	0.09	0.08	0.07	0.06	0.06	0.06
dn6	0.05	0.05	0.05	0.05	0.05	0.04	0.04	0.03	0.03
dn7	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
dn8	0.06	0.06	0.06	0.05	0.05	0.04	0.04	0.04	0.03
dn9	0.05	0.05	0.05	0.04	0.03	0.03	0.03	0.03	0.03
dn10	0.09	0.06	0.05	0.05	0.04	0.03	0.03	0.03	0.03
dn11	0.07	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
RMS	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.06

表 5-2-2 拟合振幅  $H$  的绝均差

轨道	次数								
	12	13	14	15	16	17	18	19	20
up1	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
up2	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
up3	0.18	0.18	0.16	0.16	0.14	0.14	0.14	0.13	0.15
up4	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.04
up5	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05
up6	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
up7	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
up8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
up9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
dn1	0.12	0.12	0.11	0.11	0.11	0.11	0.11	0.10	0.10
dn2	0.02	0.02	0.02	0.02	0.03	0.02	0.02	0.03	0.02
dn3	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
dn4	0.05	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04
dn5	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
dn6	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02
dn7	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
dn8	0.03	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.02
dn9	0.03	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.02
dn10	0.03	0.04	0.04	0.07	0.03	0.06	0.04	0.04	0.06
dn11	0.06	0.01	0.02	0.04	0.03	0.01	0.04	0.05	0.01
RMS	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05

#### (4) 对拟合结果确定最佳次，提取内潮。

本文又以  $M_2$  内潮为例以上升轨 up06 号轨道为例，采用多项式拟合方法进行不同次数的拟合，并选取 5、11、16 次多项式拟合的结果进行对比，以显示不同拟合次数滤波结果的差异。 $M_2$  分潮振幅曲线与  $M_2$  分潮迟角曲线对比结果如图 5-3、图 5-4 所示。

图 5-3  $M_2$  分潮振幅曲线对比图



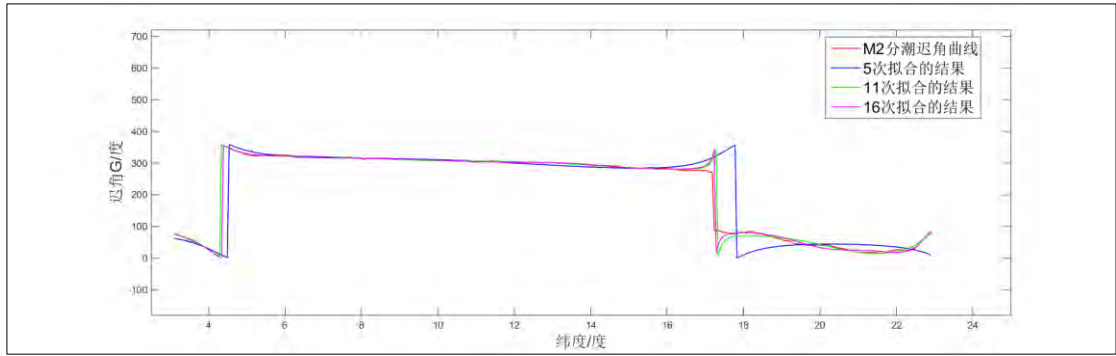


图 5-4  $M_2$  分潮迟角曲线对比图

由表 5-1-1 和表 5-1-2 可知，沿上升 up06 号轨道分别进行 5、11、16 次多项式拟合得到的振幅平均绝均差分别为 0.07、0.06、0.05m，16 次多项式拟合绝均差最小。由图 5-3 可以看出，虽然 5、11、16 次多项式的拟合结果都反映海表面  $M_2$  分潮振幅  $H$  的变化趋势，但不同次数多项式的拟合结果与海表面  $M_2$  分潮振幅  $H$  的符合程度存在明显差异。

分析可知，5 次多项式的拟合结果明显偏离调和分析得到的  $M_2$  分潮振幅值，不能正确地拟合出正压潮值；11 次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地表示正压潮值；而 16 次多项式拟合结果与海表面  $M_2$  分潮振幅的符合程度最高，能准确地表示正压潮值，从而保证内潮提取结果的准确性。结合 5-2-1 和表 5-2-2 与图 5-3 可以看出，多项式拟合结果与海表面  $M_2$  分潮振幅绝均差越小，拟合结果与海表面  $M_2$  分潮振幅的符合程度就越高。表 5-1 与图 5-4 也同样表现这种关系。

提取的 Up6 轨道 M2 分潮部分星下点内潮海表面信号如图 5-5 所示。

1	A	G	lat	lon
2	0.00221	1.71062	3.0864	112.372
3	-0.0057	-2.8881	3.1364	112.39
4	0.00339	0.9486	3.1864	112.408
5	0.00058	-0.3078	3.2364	112.426
6	0.00303	-0.6505	3.2864	112.444
7	-0.0011	0.85331	3.3364	112.462
8	-0.0028	1.19186	3.3864	112.48
9	0.00209	-1.0757	3.4365	112.498
10	-0.0024	0.44928	3.4865	112.516
11	-0.0012	-0.2048	3.5365	112.533
12	0.00476	-1.2512	3.5865	112.551
13	0.0034	1.23928	3.6365	112.569
14	-0.0062	0.13746	3.6865	112.587
15	-0.0032	-0.8857	3.7365	112.605
16	0.00155	-1.3205	3.7865	112.623
17	-0.0021	-0.0795	3.8365	112.641
18	0.00084	0.75007	3.8865	112.659
19	0.00029	2.40546	3.9365	112.677
20	-0.0018	2.25662	3.9865	112.695
21	0.00061	1.08529	4.0365	112.713
22	0.00038	-0.8453	4.0865	112.731
23	0.00252	-1.2997	4.1365	112.749
24	2.24E-05	0.46132	4.1865	112.767

图 5-5 内潮信号

## 6. 问题三

### 6.1 潮汐调和常数的提取

#### 6.1.1 问题分析

在选择插值算法之前 首先要面对的问题就是，什么样的插值结果算是好的？基于不同的准则选择哪种不同的插值算法？通过分析插值问题的本质为：插值后的结果要反映数据集。因此，插值算法是否能反映数据集的特性 是衡量该插值算法是否适用于该数据集的一个衡量标准<sup>[5]</sup>。

对于各分潮的调和常数  $H$  和  $g$  的散乱点数据，克里金 (Kriging) 方法通过引进以距离为自变量的变差函数来计算权值，由于变差函数既可以反映变量的空间结构特性，又可以反应变量的随机分布特性，所以利用克里金方法进行各分潮的调和常数  $H$  和  $g$  这样的散乱点空间数据插值往往可以取得理想的效果。

因此，本文选择使用克里金插值方法制作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。

第一步：克里金插值的数学建模。

第二步：利用克里金插值的方法制作出各主要分潮的同潮图。

第三步：利用潮汐验潮点的调和常数给予检验，并对检验结果进行分析。利用潮汐验潮点的调和常数  $H$  和  $g$  给予评价或检验，并给出评价结果的分析或评价。

#### 6.1.2 问题求解

依照上述分析，将问题分为三步求解：

##### (1) 克里金插值的数学建模。

克里金插值法，又称空间局部估计或空间局部插值法，是地统计学的主要内容之一。克里金插值法在空间相关范围分析的基础上，用相关范围内的采样点来估计待插点属性值。克里金法是建立在变异函数理论及结构分析基础之上的，它在有限区域内对区域化变量的取值进行无偏、最优估计。变异函数和相关分析的结果表明，区域化变量存在空间相关性，其实质是利用区域化变量的原始数据和变异函数的结构特点，对未采样点的区域化变量的取值进行线性无偏、最优估计。与其它的插值方法相比，克里金法的显著特点是使误差的方差最小。克里金插值法主要包括：简单克里金、普通克里金、泛克里金、指示克里金、析取克里金和协同克里金等。普通克里金法是一种较为常用的方法<sup>[11]</sup>。

假设区域化变量  $Z(x)$  满足二阶平稳假设和本征假设，其数学期望为  $m$ ，协方差函数  $c(h)$  及变异函数  $\gamma(h)$  存在。即：

$$\begin{aligned} E[Z(x)] &= m \\ c(h) &= E[Z(x)Z(x+h)] - m^2 \\ \gamma(h) &= 1/2 E[Z(x) - Z(x+h)]^2 \end{aligned} \quad (6-1)$$

设  $Z(x)$  是一个二阶平稳的随机函数，它在  $n$  个位置取样： $Z(x_1), Z(x_2), \dots, Z(x_n)$ ，点  $x_0$  处的估计量为：

$$Z^*(x_0) = \sum_{i=1}^n \lambda_i Z(x_i) \quad (6-2)$$

其中： $\lambda_i$  为权重系数，表示各空间样本点  $x_i$  处的观测值  $Z(x_i)$  对估计值  $Z^*(x)$  的贡献程度。克里金算法的关键就是计算权重系数  $\lambda_i$ ，权重系数计算必须满足两个条件：

(1) 使  $Z^*(x)$  是  $Z(x_i)$  的无偏估计量，即  $E[Z^*(x)] = E[Z(x)]$ 。当  $E[Z^*(x)] = m$

时，即  $E\left[\sum_{i=1}^n \lambda_i Z(x_i)\right] = \sum_{i=1}^n \lambda_i E[Z(x_i)] = m$  时，则有  $\sum_{i=1}^n \lambda_i = 1$ ；

(2) 使估计方差最小，也就是使估计值  $Z^*(x)$  和实际值  $Z(x_i)$  之差的平方和最小。即：

$$\sigma^2 = E[Z(x) - Z^*(x)]^2 = E\left[Z(x) - \sum_{i=1}^n \lambda_i Z(x_i)\right]^2 \quad (6-3)$$

用协方差函数可以表达为：

$$\sigma^2 = c(x, x) + \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j c(x_i, x_j) - 2 \sum_{i=1}^n \lambda_i c(x_i, x) \quad (6-4)$$

要使估计方差最小，根据拉格朗日乘数原理，令：

$$F = \sigma^2 - 2\mu\left(\sum_{i=1}^n \lambda_i - 1\right) \quad (6-5)$$

求  $F$  对  $\lambda_i$  和  $\mu$  的偏导数，并令偏导数为 0，得克里金方程组：

$$\begin{cases} \frac{\partial F}{\partial \lambda_i} = 2 \sum_{j=1}^n \lambda_j c(x_i, x_j) - 2\mu = 0 \\ \frac{\partial F}{\partial \mu} = -2\left(\sum_{i=1}^n \lambda_i - 1\right) = 0 \quad (i = 1, 2, \dots, n) \end{cases} \quad (6-6)$$

整理后得：

$$\begin{cases} \sum_{j=1}^n \lambda_j c(x_i, x_j) - \mu = c(x_i, x) \\ \sum_{i=1}^n \lambda_i = 1 \quad (i = 1, 2, \dots, n) \end{cases} \quad (6-7)$$

解线性方程组 (4)，求出权重系数  $\lambda_i$  和拉格朗日乘数  $\mu$ ，代入公式 (1) 和 (2)，分别求出估计值和估计方差，即：

$$\begin{cases} \sum_{j=1}^n \lambda_j \gamma(x_i, x_j) + \mu = \gamma(x_i, x) \\ \sum_{i=1}^n \lambda_i = 1 \quad (i = 1, 2, \dots, n) \end{cases} \quad (6-8)$$

$$\sigma_k^2 = \sum_{i=1}^n \lambda_i \gamma(x_i, x) - \gamma(x, x) + \mu \quad (6-9)$$

方程组也可以用矩阵表示，令

$$K = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} & 1 \\ c_{21} & c_{22} & \cdots & c_{2n} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}, \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ -\mu \end{bmatrix}, D = \begin{bmatrix} c(x_1, x_1) \\ c(x_2, x_2) \\ \vdots \\ c(x_n, x_n) \\ 1 \end{bmatrix} \quad (6-10)$$

则普通克里金方程组为：

$$K\lambda = D \quad (6-11)$$

解方程组 (7)，可得  $\lambda = K^{-1}D$

其估计方差为：

$$\sigma^2 = c(x, x) - \lambda^T D \quad (6-12)$$

克里金插值的前提是：根据空间场的结构，选择适当的变异函数模型，并求出变异函数，其计算步骤如下：

- (1) 网格化。选择区域的范围及网格的大小。
- (2) 计算被估点（即网格节点）坐标。
- (3) 根据搜索策略（近点距离搜索，方位搜索）选择合适的参估点。
- (4) 根据已经求出的变异函数，求出方程组的系数。

$$K = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} & 1 \\ c_{21} & c_{22} & \cdots & c_{2n} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \quad (6-13)$$

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ -\mu \end{bmatrix}, D = \begin{bmatrix} c(x_1, x_1) \\ c(x_2, x_2) \\ \vdots \\ c(x_n, x_n) \\ 1 \end{bmatrix} \quad (6-14)$$

- (5) 解方程（采用 LU 方法），求权系数  $\lambda_i$ 。

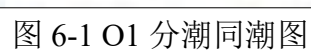
- (6) 用  $Z^*(x) = \sum_{i=1}^n \lambda_i Z(x_i)$  求被估点的值，其中  $n$  是插值点的个数。

- (7) 重复 (2) ~ (6) 步，知道网络节点的值全部求出。

- (8) 输出结果。

## (2) 利用插值的方法制作出各主要分潮的同潮图。

本文用卫星高度计资料计算潮汐调和常数结合验潮站资料，得到各分潮的振幅和迟角并利用克里金插值方法使用球面曲线模型拟合半变异函数插值到规则网格上，提取等值线从而绘制出南海海区域 4 个主要分潮( $M_2$ 、 $S_2$ 、 $K_1$ 、 $O_1$ ) 的同潮图，如图 6-1、6-2、6-3 和 6-4，同潮图中，实线为等迟角线，虚线为等振幅线<sup>[6]</sup>。



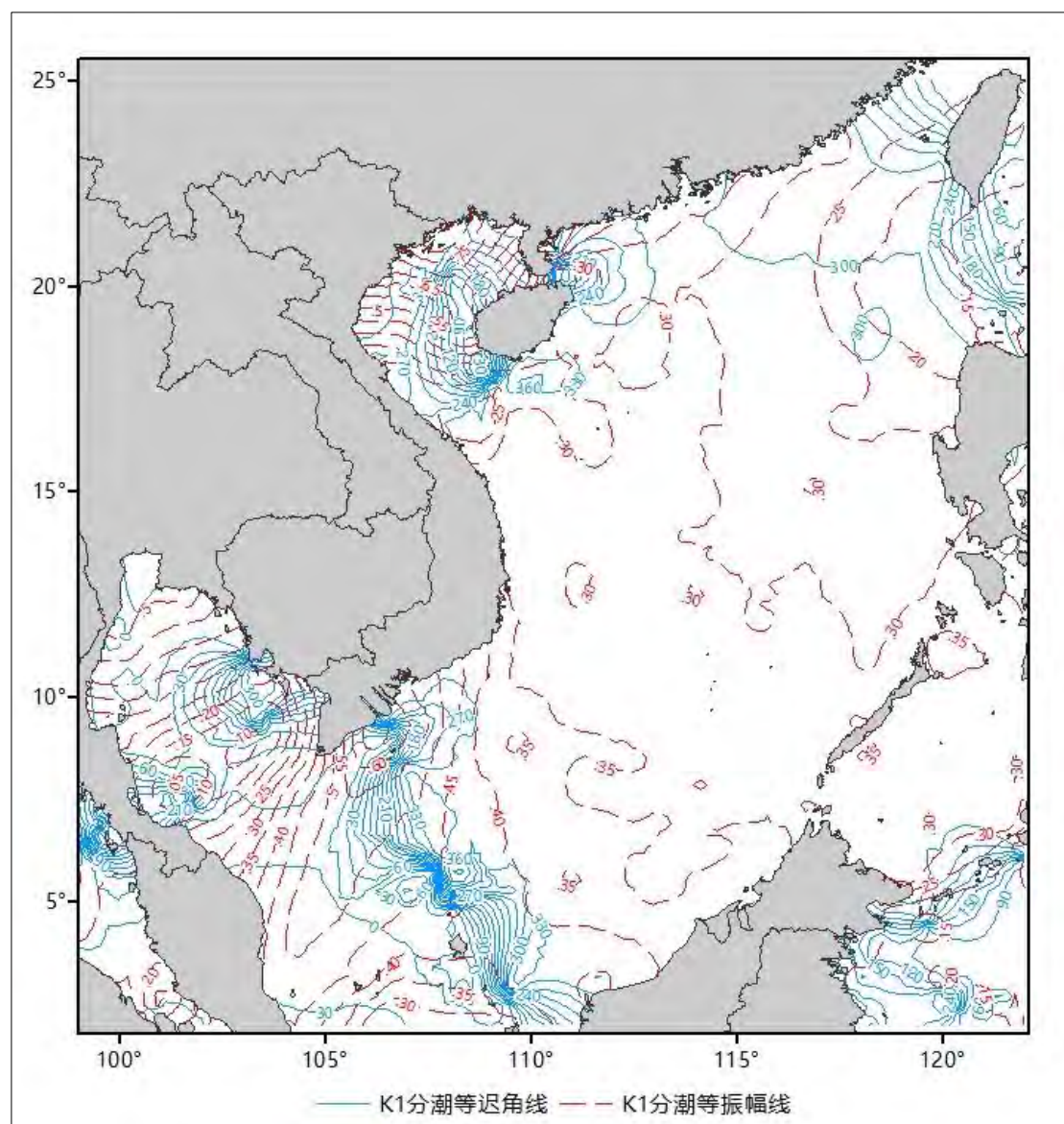


图 6-2 K1 分潮同潮图



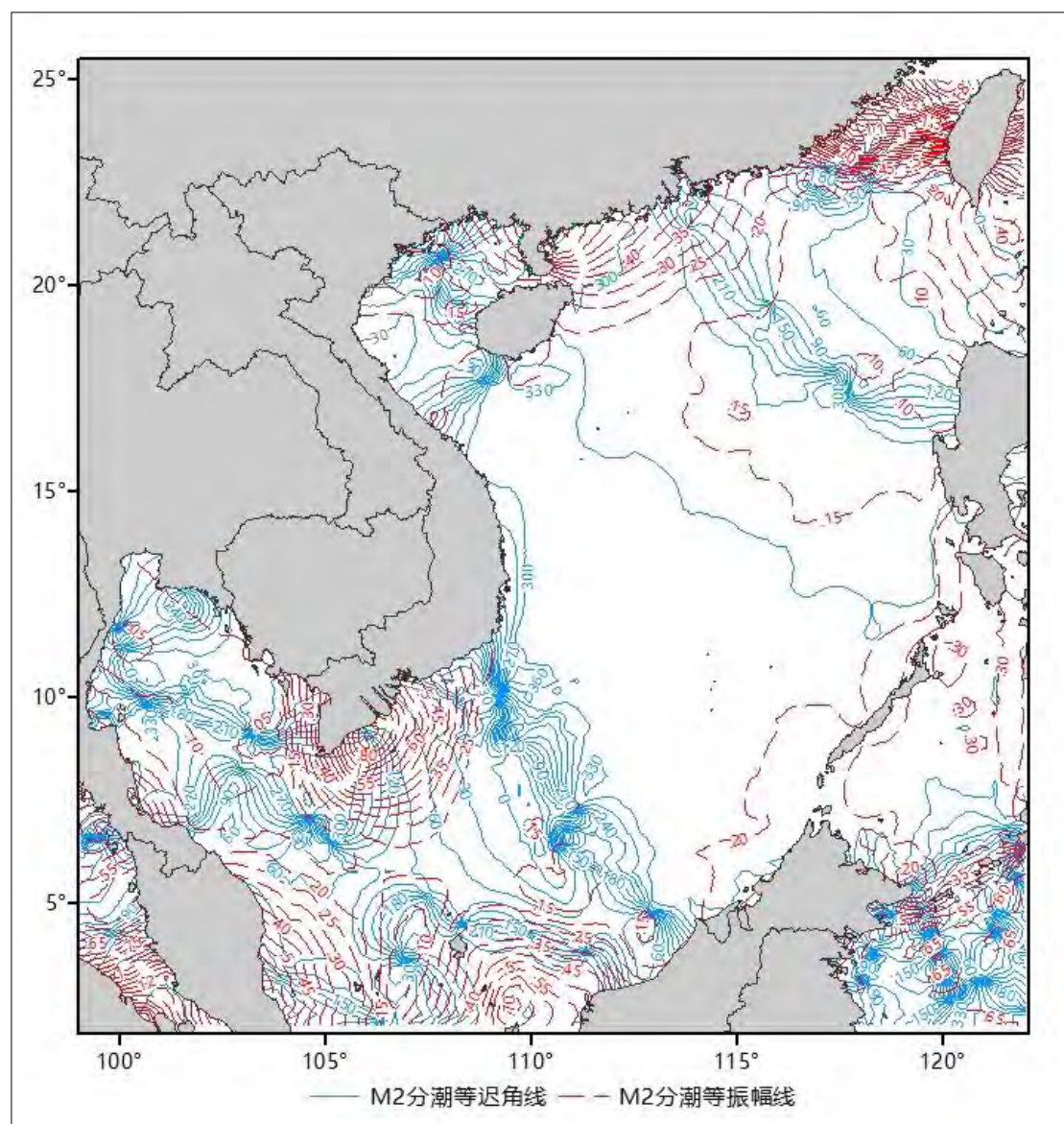


图 6-3 M2 分潮同潮图

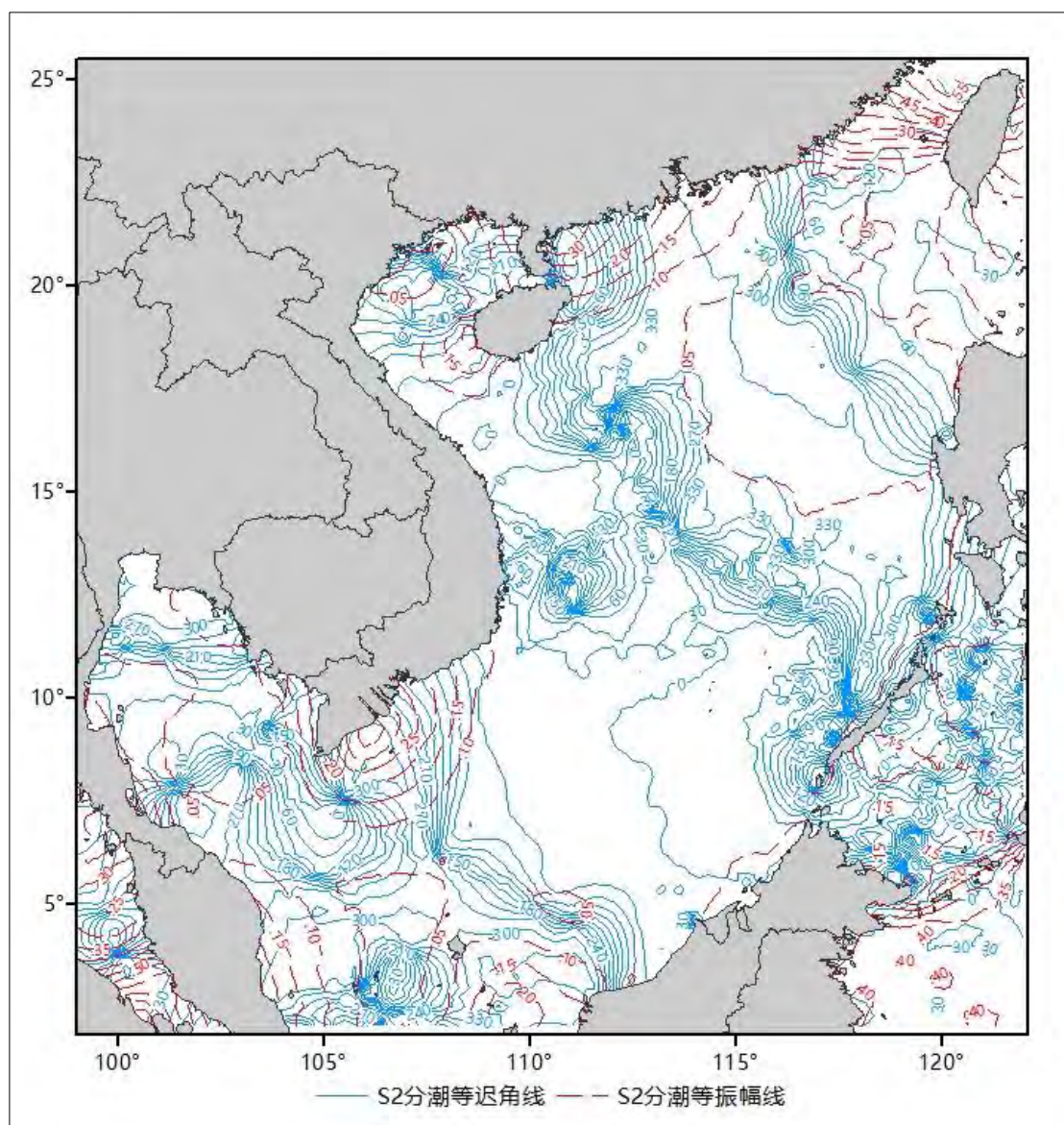
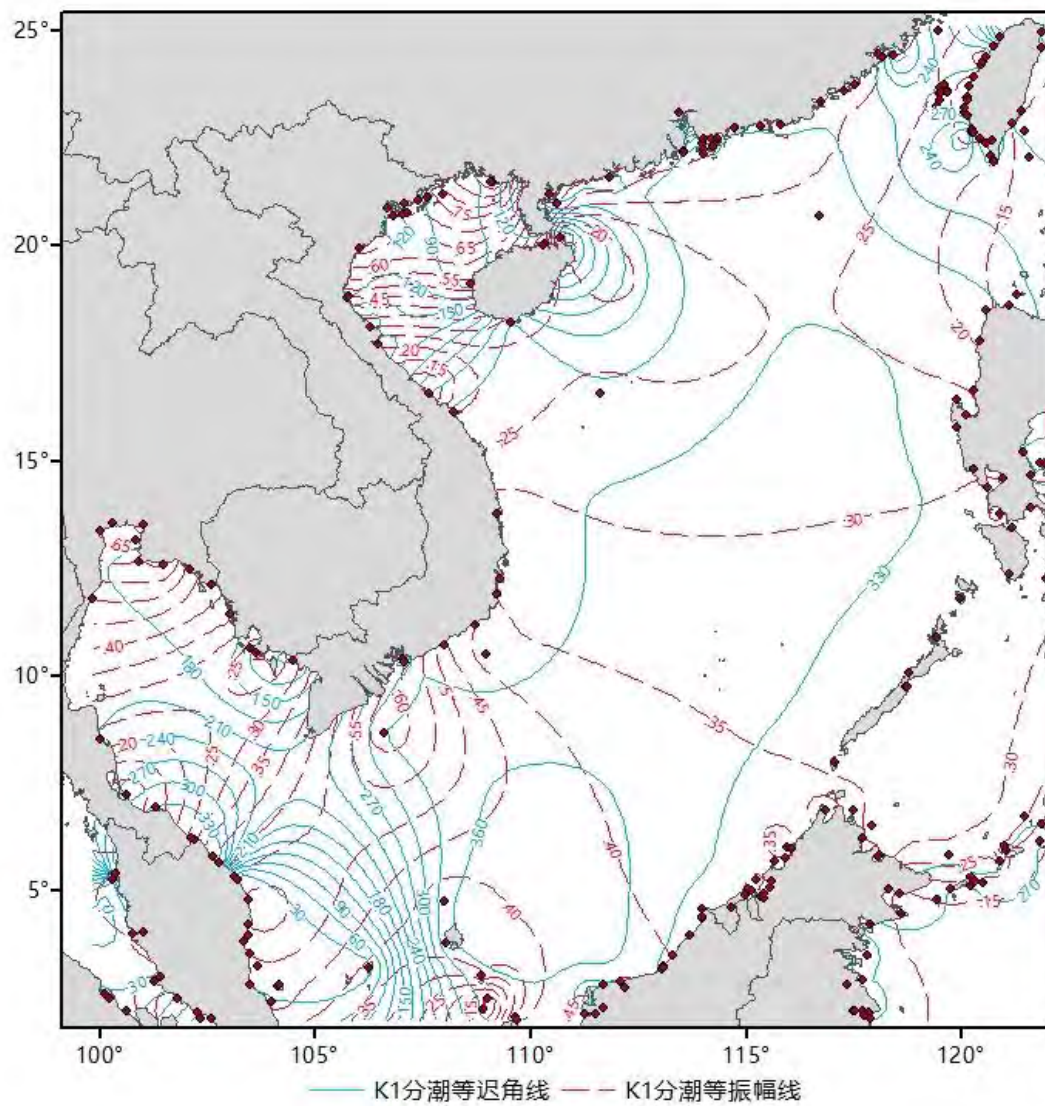


图 6-4 S2 分潮同潮图

(2) 利用潮汐验潮点的调和常数给予检验，并对检验结果进行分析。

利用潮汐验潮点的调和常数制作各分潮的同潮图进行检验，如图 6-5、6-6、6-7 以及 6-8 所示，并对检验结果进行分析。





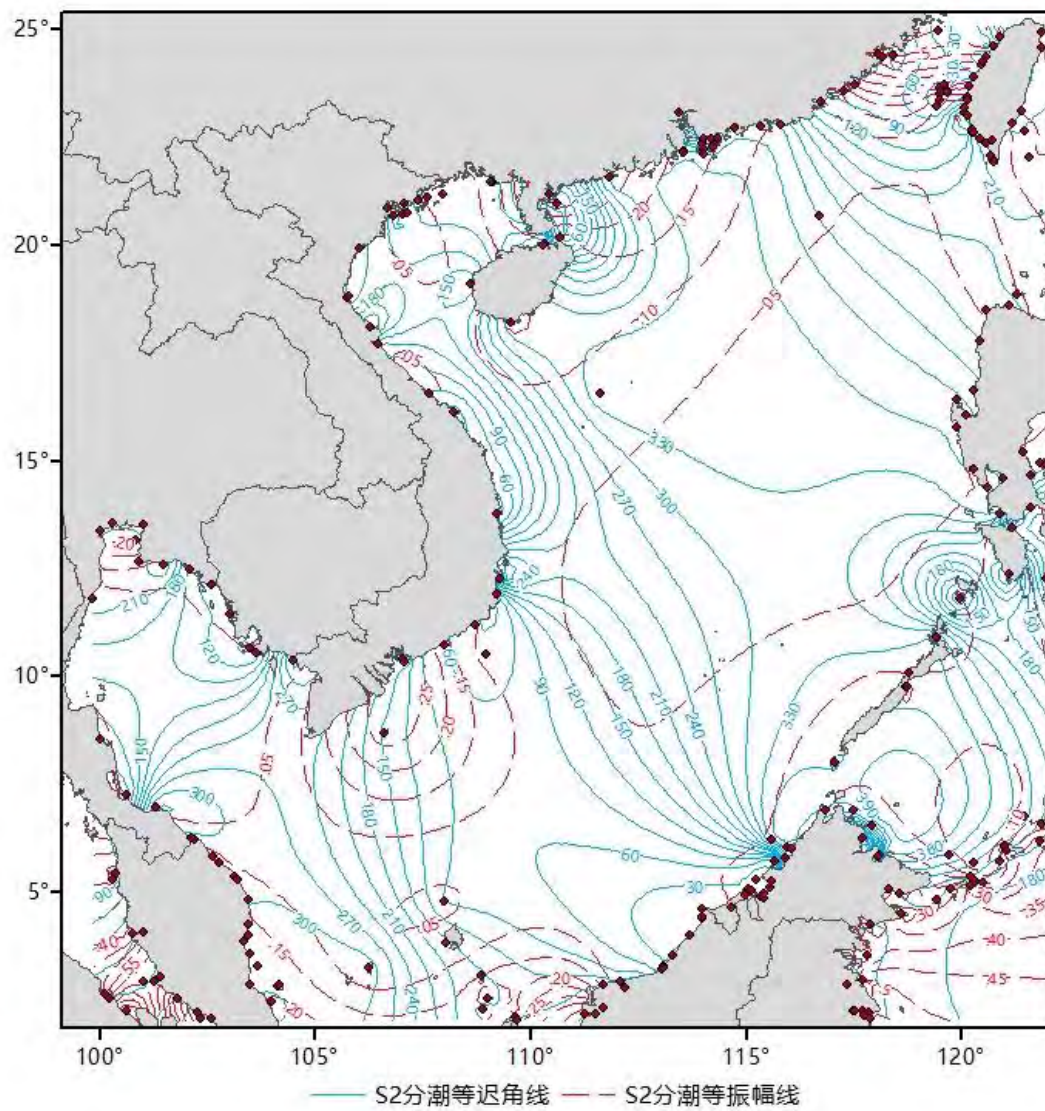


图 6-6 验潮站 S2 分潮同潮图

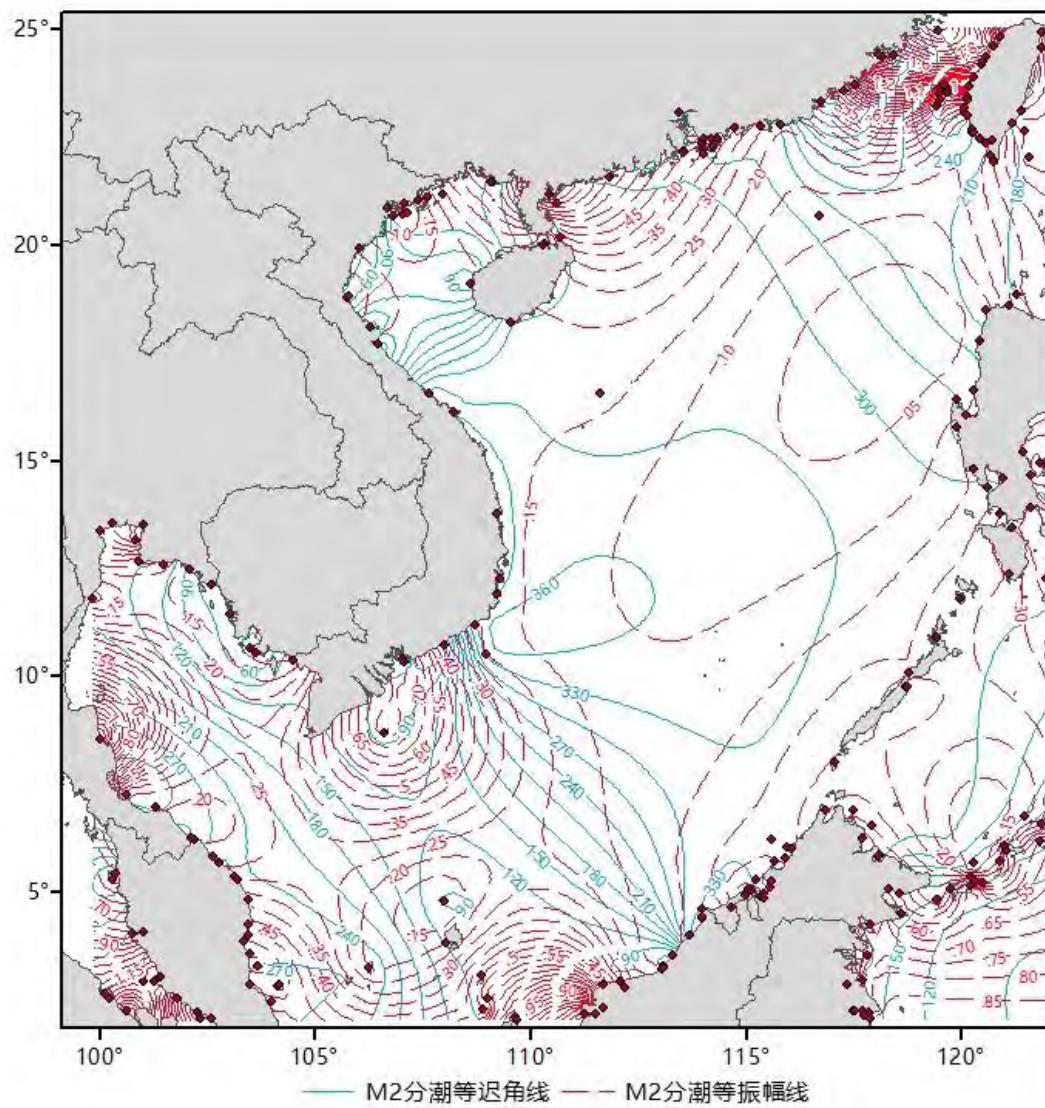


图 6-7 验潮站 M2 分潮同潮图



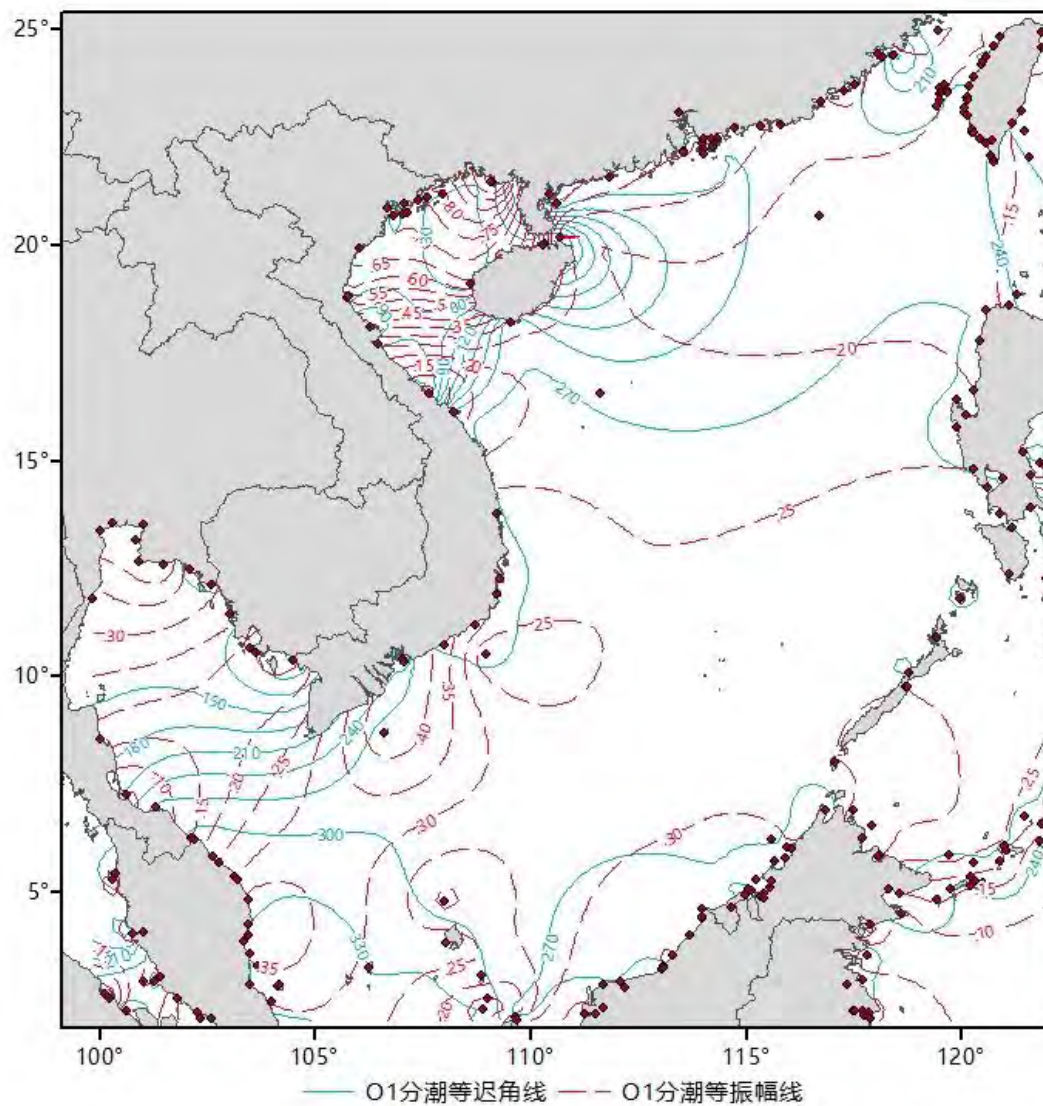


图 6-8 验潮 O1 分潮同潮图

### (3) 同潮图的评价以及评价结果的分析。

把星下点插值结果生成的同潮图和根据验潮站克里金插值生成的同潮图，以及同时选取参考资料<sup>[10]</sup>的 M2、K1 同潮图进行对比。南海 M2 同潮图对比，计算结果、验潮站结果、参考论文结果如图 6-9、6-10 和 6-11；图南海 K1 同潮图对比，计算结果、验潮站结果、参考论文结果如图 6-12、6-13 和 6-14。

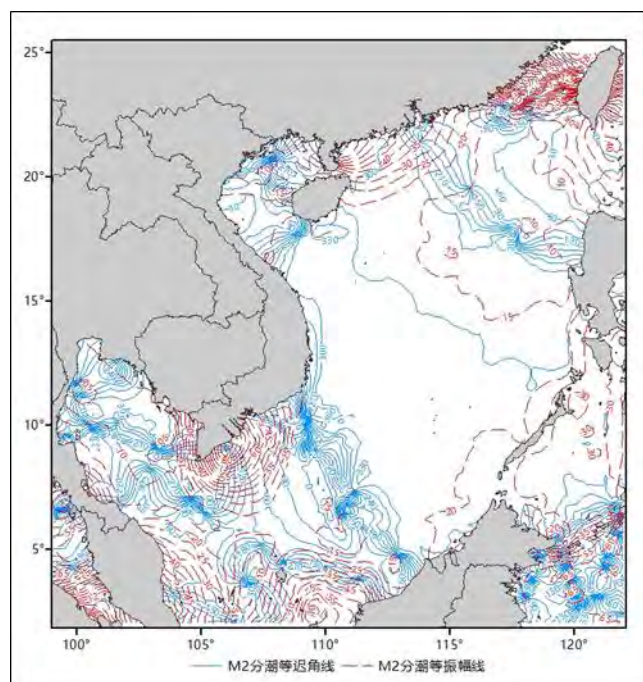


图 6-9 M2 分潮计算结果图

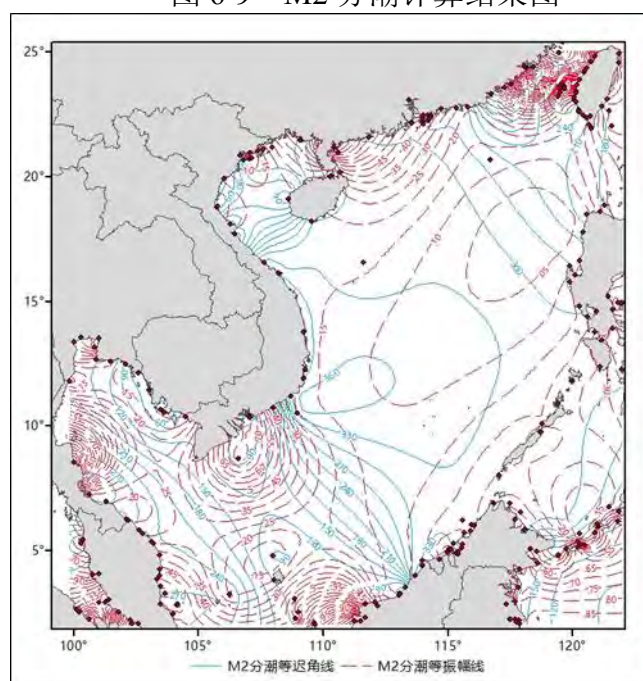


图 6-10 M2 分潮验潮站结果图

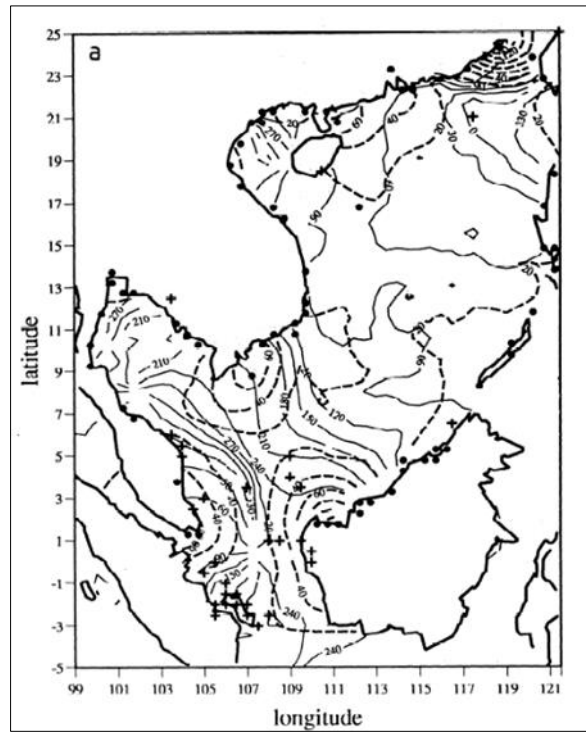


图 6-11 M2 分潮参考论文结果

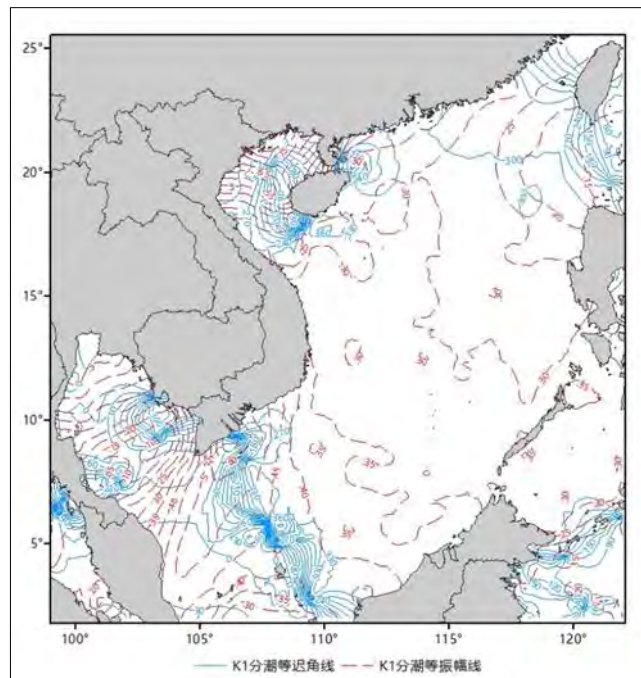


图 6-12 K1 分潮计算结果图



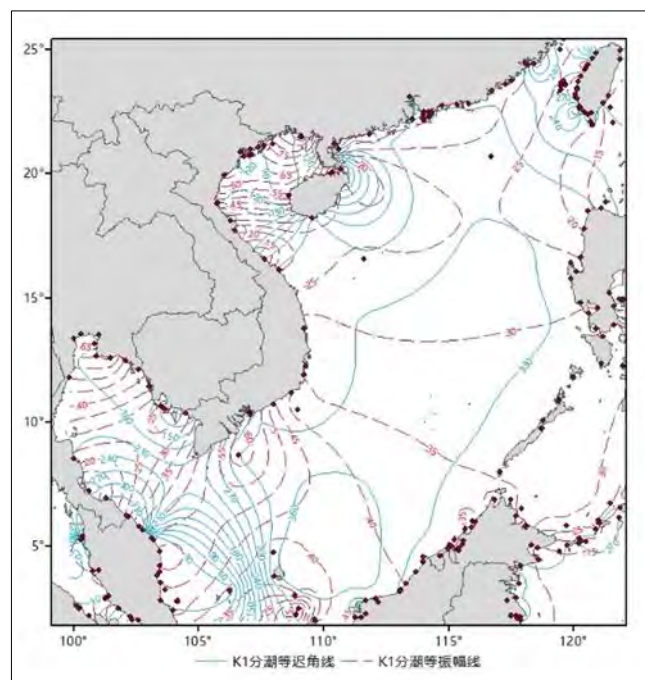


图 6-13 K1 分潮验潮站结果图

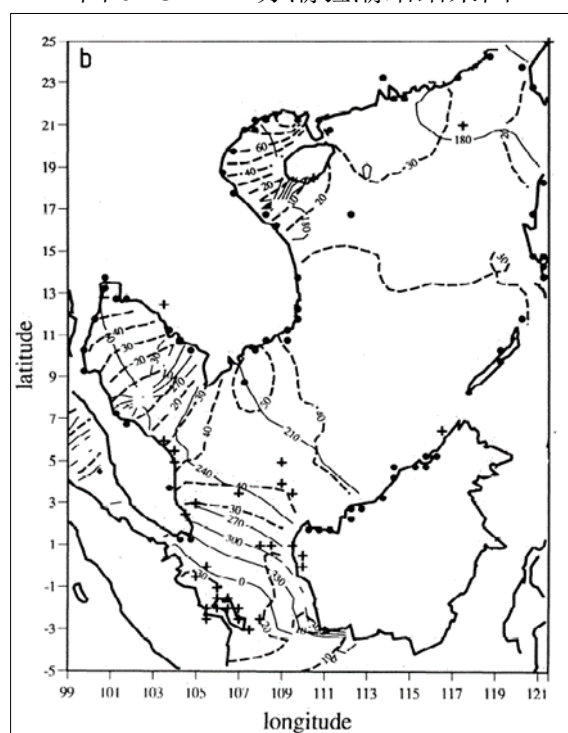


图 6-14 K1 分潮参考论文结果

通过对比，调和和分析结果整体分布同论文及验潮站结果相似，表明克里金插值方法具有一定的优势。同时对于不同水深区域，在近岸区域验潮站的插值结果更加准确，在深海中高度计调和和分析插值结果细节表现较好。

## 7. 问题四

### 7.1 潮汐调和常数的提取

#### 7.1.1 问题分析

详情见问题二分析。

#### 7.1.2 问题求解

依照问题二的分析，将本问题四分为两步求解：

第一步：建立多项式数学模型。针对各分潮调和常数  $H$  和  $g$  建立多项式拟合数学模型。

第二步：利用多项式对  $M_2$  的潮汐调和常数  $H$  和  $g$  进行拟合，确定最佳次数。展示不同次数的潮汐调和常数  $H$  和  $g$  拟合结果，并对拟合结果进行分析，确定最佳次数。

第三步：对问题 3 的启示。

##### (1) 建立多项式数学模型。

详见问题二多项式数学模型。

##### (2) 利用多项式对 $M_2$ 的潮汐调和常数 $H$ 和 $g$ 进行拟合，确定最佳次数。

本题又以  $M_2$  内潮为例以上升轨 up06 号轨道为例，采用 3-20 次多项式拟合方法进行不同次数的拟合，并选对多项式拟合的结果进行对比， $M_2$  分潮振幅与  $M_2$  分潮迟角不同拟合次数的 RMS 对比结果如图 7-1、图 7-2 所示。

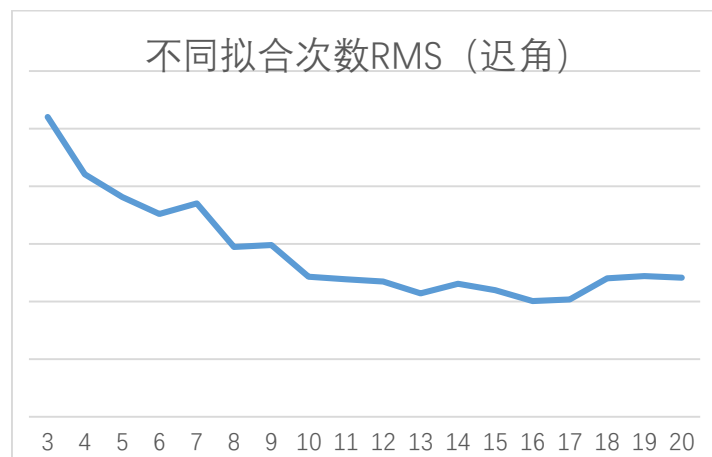


图 7-1  $M_2$  分潮迟角不同拟合次数的 RMS

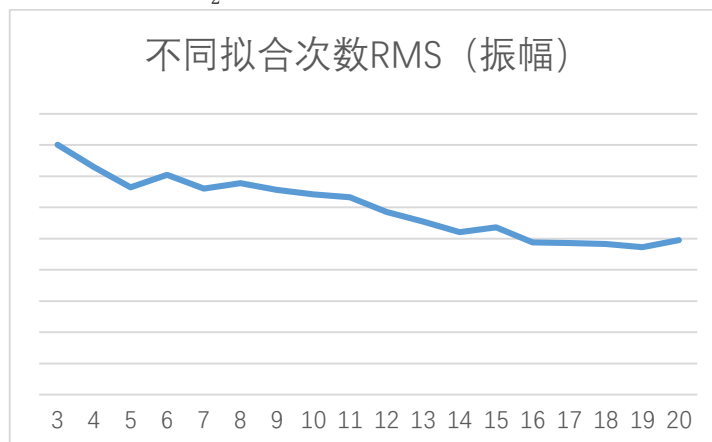


图 7-2  $M_2$  分潮振幅不同拟合次数的 RMS

由上图可以看出, 次数为 16 时  $M_2$  分潮振幅与  $M_2$  分潮迟角不同拟合次数的 RMS 值最小, 即结果最佳。

### (3) 对问题 3 的启示。

在多项式拟合过程中, 拟合次数过高会导致过拟合的情况, 选择合适的次数后, 利用参考文献<sup>[7]</sup>可以对区域内任意一点插值, 有助于第三问同潮图的绘制。

下面讨论本文空间插值方法及最佳函数的选取:

假设分潮调和常数符合在时空间分布上互为相关的条件, 因此可以使用地统计学方法进行插值。本文第三问采用了克里金插值模型即空间自协方差最佳插值法, 其采用了半变异函数来衡量样本数据的空间自相关性, 半变异函数和协方差函数将假设 3 加以量化, 将统计相关性的强度作为距离函数来测量, 可以提供一种最佳线性无偏估计的插值方法。

最佳函数选择:

在差值过程中, 通过计算所有已知点与其不同距离范围内邻居点集的距离离差平方和, 就可以大致利用曲线进行拟合已知点与不同距离范围邻居点的变化程度关系即半变异函数, 常用的曲线拟合函数包括圆曲线拟合、指数拟合、球面拟合、高斯拟合等。

在选择最佳函数时可以将点的利差平方和可视化, 选用最佳函数进行拟合。如图 7-3 所示是利用高斯函数拟合半变异函数:

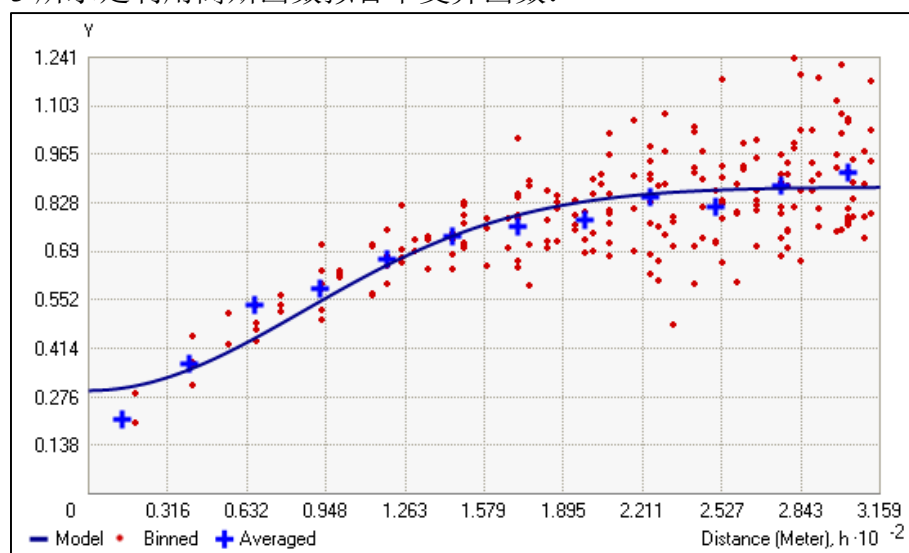


图 7-3 高斯函数拟合半变异函数

在选取最佳函数后, 拟合的结果就是我们插值所需要的块金值、基台值等, 然后即可进行插值。

对于其他插值方法, 我们还可以使用利用模型验证方法选取最佳函数:

交叉验证使用所有数据对趋势和自相关模型进行估计。它会每次移除一个数据位置, 然后预测关联的数据值。例如, 有 10 个数据点。交叉验证会省略一个点, 然后使用剩余的 9 个点计算此位置的值。将省略点位置的预测值与实际值相比较。然后对第二个点重复此过程, 以此类推。交叉验证会对所有点的测量值和预测值进行比较。完成交叉验证后, 得到所有已知点与其预测值之间的偏差, 这个所有点的偏差从某种程度上讲就为我们提供了整个预测方法是否合理的依据。通过比较不同函数的结果偏差大小, 便可以选择出最佳函数, 同时还可剔除偏差值较大的数据点。

---

地统计学方法相对于样条函数、反距离加权插值等空间确定性插值方法，能够从观测数据自身特点出发，考虑空间观测点的整体空间分布情况，更贴合实际。对于分潮调和常数的效果有待进一步验证。

## 8.模型的评价

### 8.1 模型的优点

克里金插值模型，在数据网格化的过程中考虑了空间离散数据的空间相关性，可以使插值结果更科学，更接近于实际情况。

多项式拟合可以拟合各种数据点离散情况，且简单易行。

### 8.1 模型的缺点

克里金插值方法过程复杂，需要选取拟合半变异函数的最佳曲线函数，这个过程需要将数据点可视化。

在多项式拟合过程中，拟合次数过高会导致过拟合的情况。

---

## 9.参考文献

- [1] 陈宗镛, 潮汐学, 北京: 科学出版社, 1980.
- [2] 孙丽艳. 渤黄东海潮汐底摩擦系数的优化研究[D]. 中国海洋大学 2006.
- [3] 李立 吴日升 李燕初 等. TOPEX/POSEIDON 高度计浅海潮汐混淆的初步分析[J]. 海洋学报 1999 21(3): 1 -14.
- [4] 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.
- [5]王靖波 潘懋 张绪定. 基于 Kriging 方法的空间散乱点插值[J]. 计算机辅助设计与图形学学报 1999 11(6):525-529.
- [6] 赵杰, 基于卫星高度计资料提取浙江近海的潮汐信息[J].海洋测绘 1671-3044(2009)03-0074-05.
- [7] 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.
- [8] 方国洪, 郑文振, 陈宗镛, 王骥, 潮汐和潮流的分析与预报, 北京: 海洋出版社, 1986.
- [9] 黄祖珂, 黄磊, 潮汐原理与计算, 青岛, 中国海洋大学出版社, 2005
- [10] Mazzega P Bergé M. Ocean tides in the Asian semiencloded seas from TOPEX/POSEIDON[J]. Journal of Geophysical Research Oceans 1994 99(C12):24867-24881.
- [11] 李莉 胡建平. 克里金插值算法在等高线绘制中的应用[J]. 天津城建大学学报 2008 14(1):68-71.

---

## 10. 附录

### Python 源程序

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
@Author   :   Chunyang Zhu

@Contact  :   zhucy@s.upc.edu.cn

@Software:   PyCharm

@File     :   tidesharmonic.py

@Time     :   2018/9/15 11:18

@Desc     :   求解潮汐调和常数, 输出文件
              格林尼治时间 + 8h = 北京时间
"""

import os          # 导入操作系统功能包
import datetime    # 导入时间包
import numpy as np # 导入 numpy 数值计算包

# 类与函数定义
# 类 高度计点类
class AltimeterPoint:
    time_list = [] # 时间列表 存储时间序列
    height_list = [] # 高度列表 存储高度序列

    HH = []
    gg = []

    # 构造函数 赋值
    def __init__(self, _ID, _lon, _lat, _num, _t_list, _h_list):
        self.ID = _ID
        self.longitude = _lon
        self.latitude = _lat
        self.num_series = _num
        self.time_list = _t_list
        self.height_list = _h_list
        self.h_M2 = -99
        self.g_M2 = -99
        self.h_S2 = -99
        self.g_S2 = -99
```

---

```

        self.h_K1 = -99
        self.g_K1 = -99
        self.h_01 = -99
        self.g_01 = -99

# 类方法 给调和常数赋值
def get_H_g(self, in_H, in_g, in_S):
    self.h_M2 = round(in_H[0, 0], 3)
    self.g_M2 = round(in_g[0, 0], 2)
    self.h_S2 = round(in_H[1, 0], 3)
    self.g_S2 = round(in_g[1, 0], 2)
    self.h_K1 = round(in_H[2, 0], 3)
    self.g_K1 = round(in_g[2, 0], 2)
    self.h_01 = round(in_H[3, 0], 3)
    self.g_01 = round(in_g[3, 0], 2)

# 函数 高度计数据读取
def readfile(path):
    # 记录程序开始时间
    starttime = datetime.datetime.now()

    altimeter_list = [] # 高度计点
    p_list = [] # 经纬度点列表
    t_list = [] # 时间列表
    h_list = [] # 高度列表

    # 打开高度计数据文件
    with open(path) as file:
        # 读取文件中所有行
        all_lines = file.readlines()
        # 遍历每行
        for line in all_lines:
            # 按空格拆分数据 提取经纬度、时间序列数、时间、高度
            line = line.split(' ')
            if len(line) == 3:
                lon = float(line[0])
                lat = float(line[1])
                num = int(line[2])
                p_list.append([lon, lat, num])
            elif len(line) == 2:
                t_list.append(float(line[0]) + 8 * 3600)
                h_list.append(float(line[1]))
        # print(len(p_list))

```

---

```

# 遍历经纬度点列表中每个点
start = 0
_id = 1
for point in p_list:
    end = start + int(point[2])

    # 构造高度计点列表 列表传入 高度计点类
    altimeter_list.append(AltimeterPoint(_id, point[0], point[1],
point[2], _t_list=t_list[start:end],

_h_list=h_list[start:end]))
    _id += 1
    start = end
,,,

# 剔除 T 值小于 0 的数据
for point in altimeter_list:
    t_list = []
    h_list = []
    for time, height in zip(point.time_list, point.height_list):
        if time >= 0:
            t_list.append(time)
            h_list.append(height)
    point.time_list = t_list
    point.height_list = h_list
    point.num_series = len(t_list)
,,,

# 验证数据长度读取是否正确
# for a in altimeter_list:
#     print(a.num_series, len(a.time_list), len(a.height_list))

# 函数结束时间
endtime = datetime.datetime.now()
print(' 高度计数据读取完毕，读取时间：', (endtime -
starttime).seconds, '秒')
return altimeter_list

# 函数 计算分潮初相位 参数 杜德森数，时间列向量
def cal_v0_phase(Doodson, times):
    # 起始时刻
    Y = 2000
    i = int((Y - 1901) / 4)

```



---

```

n = 0
t = 0

# 分潮初相位公式 参考<潮汐学>和附件资料<调和分析方法>
s_0 = (277.02 + 129.3848 * (Y - 1900) + 13.1764 * (n + i + t))
h_0 = (280.19 - 0.2387 * (Y - 1900) + 0.9857 * (n + i + t))
p_0 = (334.39 + 40.6625 * (Y - 1900) + 0.1114 * (n + i + t))
N_0 = (100.84 + 19.3282 * (Y - 1900) + 0.0530 * (n + i + t))
_p_0 = (281.221 + 0.0172 * (Y - 1900) + 0.00005 * (n + i + t))
tau_0 = 15 * t * 24 + h_0 - s_0

v0_const = (tau_0, s_0, h_0, p_0, N_0, _p_0, 90)

_v0 = Doodson[0] * v0_const[0] + Doodson[1] * v0_const[1] +
Doodson[2] * v0_const[2] + \
      Doodson[3] * v0_const[3] + Doodson[4] * v0_const[4] +
Doodson[5] * v0_const[5] + Doodson[6] * 90

return _v0

# 函数 计算交点因子和交点订正角 参数 时间列向量
def cal_f_u(times):
    # 观测起始时刻
    Y = 2000 + np.trunc(times/24/365.2564)
    yu = times/24 % 365.2564 * 365 * 24

    i = np.trunc((Y - 1901) / 4)
    n = 0
    t = 0
    # n = np.trunc(times/24)
    t1 = (times/24 - np.trunc(times/24)) * 24
    # 高度计数据观测记录时刻
    N_0 = (259.157 - 19.3282 * (Y - 1900) - 0.0530 * (n + i + t))

    # 计算升交点黄经 N
    sigma_N = 0.00220641
    N = -sigma_N * yu - N_0
    N *= np.pi / 180

    # 各分潮 f u 计算 参考<潮汐学>
    f_M2 = 1.0004 - 0.0373 * np.cos(N) + 0.0002 * np.cos(2 * N)
    f_S2 = np.ones(N.shape)
    f_K1 = 1.0060 + 0.1150 * np.cos(N) - 0.0088 * np.cos(2 * N) +

```

---

```

0.0016 * np.cos(3 * N)
    f_01 = 1.0089 + 0.1871 * np.cos(N) - 0.0147 * np.cos(2 * N) +
0.0014 * np.cos(3 * N)

    u_M2 = -2.14 * np.sin(N)
    u_S2 = np.zeros(N.shape)
    u_K1 = -8.86 * np.sin(N) + 0.68 * np.sin(2 * N) - 0.07 * np.sin(3
* N)
    u_01 = 10.8 * np.sin(N) - 1.34 * np.sin(2 * N) + 0.19 * np.sin(3
* N)

    f_mat = np.hstack((f_M2, f_S2, f_K1, f_01))
    u_mat = np.hstack((u_M2, u_S2, u_K1, u_01))

    return f_mat, u_mat

# 最小二乘法求解
def cal_least(_A, _h):
    AtA = np.mat(np.dot(_A.T, _A))
    AtA_I = AtA.I
    _X = np.dot(np.dot(AtA_I, _A.T), h)
    return _X

# 函数 求分潮调和常数
def cal_harmonic_const(_X):
    # 最小二乘解 H*cosg
    Xj = _X[0:4, 0]
    # 最小二乘解 H*sing
    Yj = _X[4:8, 0]

    Sc = _X[8, 0]
    Hc = np.sqrt(np.square(Xj) + np.square(Yj))
    gc = (np.arctan(Yj/Xj))

    # 修正迟角 g 的取值范围
    for i_gc in range(0, 4):
        if gc[i_gc] * np.sin(gc[i_gc]) < 0:
            gc[i_gc] = gc[i_gc] + np.pi
        if gc[i_gc] < 0:
            gc[i_gc] = gc[i_gc] + 2 * np.pi

```

---

```

gc = gc * 180/np.pi

# 同潮图修正 360 和 0 的突变
,,,

for i_gc in range(0, 4):
    if 340 < gc[i_gc] <= 360:
        gc[i_gc] -= 360
    ,,,

return Hc, gc, Sc

# 输出调和常数结果
def output_hg(alti_list):
    if not os.path.exists('G:/Desktop/result/TP_HG'):
        os.mkdir('G:/Desktop/result/TP_HG')
    filename = 'G:/Desktop/result/TP_HG/TP_M2.txt'
    with open(filename, 'w') as f:
        for index_num, points in enumerate(alti_list, 1):
            # print('输出 M2, 第', index_num, '个点')
            strings = str(index_num) + " " + str(points.h_M2) + " " +
str(points.g_M2) + " " + str(
                points.latitude) + " " + str(points.longitude) + "\n"
            f.writelines(strings)
    filename = 'G:/Desktop/result/TP_HG/TP_S2.txt'
    with open(filename, 'w') as f:
        for index_num, points in enumerate(alti_list, 1):
            # print('输出 S2, 第', index_num, '个点')
            strings = str(index_num) + " " + str(points.h_S2) + " " +
str(points.g_S2) + " " + str(
                points.latitude) + " " + str(points.longitude) + "\n"
            f.writelines(strings)
    # 输出文件 TP_K1
    filename = 'G:/Desktop/result/TP_HG/TP_K1.txt'
    with open(filename, 'w') as f:
        for index_num, points in enumerate(alti_list, 1):
            # print('输出 K1, 第', index_num, '个点')
            strings = str(index_num) + " " + str(points.h_K1) + " " +
str(points.g_K1) + " " + str(
                points.latitude) + " " + str(points.longitude) + "\n"
            f.writelines(strings)
    # 输出文件 TP_O1
    filename = 'G:/Desktop/result/TP_HG/TP_O1.txt'
    with open(filename, 'w') as f:
        for index_num, points in enumerate(alti_list, 1):

```

---

```

        # print('输出 01, 第', index_num, '个点')
        strings = str(index_num) + " " + str(points.h_01) + " " +
str(points.g_01) + " " + str(
            points.latitude) + " " + str(points.longitude) + "\n"
        f.writelines(strings)

# 输出去除不符值调和常数结果
def output_hg_filter(alti_list):
    if not os.path.exists('G:/Desktop/result/TP_HG_F'):
        os.mkdir('G:/Desktop/result/TP_HG_F')
    filename = 'G:/Desktop/result/TP_HG_F/TP_M2_f.txt'
    with open(filename, 'w') as f:
        for index_num, points in enumerate(alti_list, 1):
            if points.h_M2 == -99:
                continue
            else:
                strings = str(points.h_M2) + " " + str(points.g_M2) +
" " + str(points.latitude) + " " + \
                    str(points.longitude) + "\n"
                f.writelines(strings)
    filename = 'G:/Desktop/result/TP_HG_F/TP_S2_f.txt'
    with open(filename, 'w') as f:
        for index_num, points in enumerate(alti_list, 1):
            if points.h_S2 == -99:
                continue
            else:
                strings = str(points.h_S2) + " " + str(points.g_S2) +
" " + str(points.latitude) + " " + \
                    str(points.longitude) + "\n"
                f.writelines(strings)

# 输出文件 TP_K1
filename = 'G:/Desktop/result/TP_HG_F/TP_K1_f.txt'
with open(filename, 'w') as f:
    for index_num, points in enumerate(alti_list, 1):
        if points.h_K1 == -99:
            continue
        else:
            strings = str(points.h_K1) + " " + str(points.g_K1) +
" " + str(points.latitude) + " " + \
                str(points.longitude) + "\n"
            f.writelines(strings)

# 输出文件 TP_01
filename = 'G:/Desktop/result/TP_HG_F/TP_01_f.txt'

```

---

```

with open(filename, 'w') as f:
    for index_num, points in enumerate(alti_list, 1):
        if points.h_M2 == -99:
            continue
        else:
            strings = str(points.h_01) + " " + str(points.g_01) +
" " + str(points.latitude) + " " + \
                str(points.longitude) + "\n"
            f.writelines(strings)

# 按轨道输出调和常数文件
def output_track(alti_p_list):
    d_tide = ('M2', 'S2', 'K1', 'O1')
    track = ('up1', 'up2', 'up3', 'up4', 'up5', 'up6', 'up7', 'up8',
'up9', 'dn1', 'dn2', 'dn3', 'dn4',
            'dn5', 'dn6', 'dn7', 'dn8', 'dn9', 'dn10', 'dn11')
    track_index = [(3109, 3226), (1, 182), (1539, 1742), (2553, 2960),
(3974, 4411), (935, 1336), (2168, 2406),
                    (3614, 3807), (519, 601), (3022, 3108), (4412,
4520), (1337, 1538), (2407, 2552), (3808, 3973),
                    (602, 934), (1879, 2167), (3227, 3613), (183, 518),
(1743, 1878), (2961, 3021)]

    for track_i, track_n in enumerate(track):
        for _d_tide in d_tide:
            output_path = "G:/Desktop/result/TP_HG_T/TP_" + _d_tide +
"_" + track_n + ".txt"
            if not os.path.exists('G:/Desktop/result/TP_HG_T'):
                os.mkdir('G:/Desktop/result/TP_HG_T')
            a1 = track_index[track_i][0]
            a2 = track_index[track_i][1]
            p_list = alti_p_list[a1:a2]
            if _d_tide == 'M2':
                with open(output_path, 'w') as f_w:
                    for i_num, a_points in enumerate(p_list, 1):
                        str_t = str(a_points.h_M2) + " " +
str(a_points.g_M2) + " " + \
                                str(a_points.latitude) + " " +
str(a_points.longitude) + "\n"
                        f_w.writelines(str_t)
            if _d_tide == 'S2':
                with open(output_path, 'w') as f_w:
                    for i_num, a_points in enumerate(p_list, 1):

```

---

```

        str_t = str(a_points.h_S2) + " " +
str(a_points.g_S2) + " " + \
        str(a_points.latitude) + " " +
str(a_points.longitude) + "\n"
        f_w.writelines(str_t)
    if _d_tide == 'K1':
        with open(output_path, 'w') as f_w:
            for i_num, a_points in enumerate(p_list, 1):
                str_t = str(a_points.h_K1) + " " +
str(a_points.g_K1) + " " + \
                str(a_points.latitude) + " " +
str(a_points.longitude) + "\n"
                f_w.writelines(str_t)
    if _d_tide == '01':
        with open(output_path, 'w') as f_w:
            for i_num, a_points in enumerate(p_list, 1):
                str_t = str(a_points.h_01) + " " +
str(a_points.g_01) + " " + \
                str(a_points.latitude) + " " +
str(a_points.longitude) + "\n"
                f_w.writelines(str_t)

```

# 输出按轨道

```

def output_track_filter(alti_p_list):
    d_tide = ('M2', 'S2', 'K1', '01')
    track = ('up1', 'up2', 'up3', 'up4', 'up5', 'up6', 'up7', 'up8',
'up9', 'dn1', 'dn2', 'dn3', 'dn4',
        'dn5', 'dn6', 'dn7', 'dn8', 'dn9', 'dn10', 'dn11')
    track_index = [(3109, 3226), (1, 182), (1539, 1742), (2553, 2960),
(3974, 4411), (935, 1336), (2168, 2406),
        (3614, 3807), (519, 601), (3022, 3108), (4412,
4520), (1337, 1538), (2407, 2552), (3808, 3973),
        (602, 934), (1879, 2167), (3227, 3613), (183, 518),
(1743, 1878), (2961, 3021)]

    for track_i, track_n in enumerate(track):
        for _d_tide in d_tide:
            output_path = "G:/Desktop/result/TP_HG_T_F/TP_" + _d_tide
+ "_" + track_n + "_f.txt"
            if not os.path.exists('G:/Desktop/result/TP_HG_T_F'):
                os.mkdir('G:/Desktop/result/TP_HG_T_F')
            a1 = track_index[track_i][0]
            a2 = track_index[track_i][1]

```

---

```

p_list = alti_p_list[a1:a2]
if _d_tide == 'M2':
    with open(output_path, 'w') as f_w:
        for i_num, a_points in enumerate(p_list, 1):
            if a_points.h_M2 == -99:
                continue
            else:
                str_t = str(a_points.h_M2) + " " +
str(a_points.g_M2) + " " + \
                                str(a_points.latitude) + " " +
str(a_points.longitude) + "\n"
                f_w.writelines(str_t)

if _d_tide == 'S2':
    with open(output_path, 'w') as f_w:
        for i_num, a_points in enumerate(p_list, 1):
            if a_points.h_S2 == -99:
                continue
            else:
                str_t = str(a_points.h_S2) + " " +
str(a_points.g_S2) + " " + \
                                str(a_points.latitude) + " " +
str(a_points.longitude) + "\n"
                f_w.writelines(str_t)

if _d_tide == 'K1':
    with open(output_path, 'w') as f_w:
        for i_num, a_points in enumerate(p_list, 1):
            if a_points.h_K1 == -99:
                continue
            else:
                str_t = str(a_points.h_K1) + " " +
str(a_points.g_K1) + " " + \
                                str(a_points.latitude) + " " +
str(a_points.longitude) + "\n"
                f_w.writelines(str_t)

if _d_tide == 'O1':
    with open(output_path, 'w') as f_w:
        for i_num, a_points in enumerate(p_list, 1):
            if a_points.h_O1 == -99:
                continue
            else:
                str_t = str(a_points.h_O1) + " " +
str(a_points.g_O1) + " " + \
                                str(a_points.latitude) + " " +

```

---

```

str(a_points.longitude) + "\n"
                                f_w.writelines(str_t)

# 程序主入口
if __name__ == '__main__':
    starttime = datetime.datetime.now()

    # 高度计点数据读取
    altimeterPoint = readfile("G:/TP.txt")

    # Doodson 数 M2 S2 K1 O1
    Doodson_M2 = (2, 0, 0, 0, 0, 0, 0)
    Doodson_S2 = (2, 2, -2, 0, 0, 0, 0)
    Doodson_K1 = (1, 1, 0, 0, 0, 0, 1)
    Doodson_O1 = (1, -1, 0, 0, 0, 0, -1)

    # 分潮角速度 M2 S2 K1 O1
    sigma_velocity_M2 = 28.9841042
    sigma_velocity_S2 = 30.0000000
    sigma_velocity_K1 = 15.0410686
    sigma_velocity_O1 = 13.9430356

    # 分潮角速度列向量
    sigma_j = np.array([[sigma_velocity_M2,    sigma_velocity_S2,
sigma_velocity_K1, sigma_velocity_O1]])

    # print('sigma_j:', sigma_j.shape)

    # 循环求解所有点
    flag_n = 0
    for index_p, alti_point in enumerate(altimeterPoint, 1):
        print('开始计算第', index_p, '个星下点...num_sreies:',
alti_point.num_series)

        if alti_point.num_series <= 110:
            flag_n += 1
            print('剔除')
            continue

        # 高度列向量
        h
        =
np.array(alti_point.height_list).reshape(alti_point.num_series, 1)
        # print('h:', h.shape)

```



---

```

# 时间序列
t_arr =
np.array(alti_point.time_list).reshape(alti_point.num_series, 1)
# 时间 小时
# t_arr = 24 * ((t_arr / (24 * 60 * 60)) - np.trunc(t_arr /
(24 * 60 * 60)))

t_arr2 = t_arr / 3600
# print(t_arr.shape)

# 构造时间序列矩阵
t_arr_tile = np.tile(t_arr2, (1, 4))
# print('t:', t_arr_tile.shape)

# 构造分潮角速度矩阵
sigma_j = np.array([[sigma_velocity_M2, sigma_velocity_S2,
sigma_velocity_K1, sigma_velocity_O1]])
sigma_j = np.tile(sigma_j, (alti_point.num_series, 1))

# 构造分潮初相位矩阵
v0_M2 = cal_v0_phase(Doodson_M2, t_arr2)
v0_S2 = cal_v0_phase(Doodson_S2, t_arr2)
v0_K1 = cal_v0_phase(Doodson_K1, t_arr2)
v0_O1 = cal_v0_phase(Doodson_O1, t_arr2)

v0 = np.hstack((v0_M2, v0_S2, v0_K1, v0_O1))
# print('v0:', v0.shape)

# 交点因子与交点订正角
f_arr, u_arr = cal_f_u(t_arr2)
# print(f_arr.shape, u_arr.shape)

# 构造系数矩阵 A
a1 = f_arr * np.cos((sigma_j * t_arr_tile + v0 + u_arr) * np.pi
/ 180)
a2 = f_arr * np.sin((sigma_j * t_arr_tile + v0 + u_arr) * np.pi
/ 180)
a3 = np.ones((alti_point.num_series, 1))
A = np.hstack((a1, a2, a3))
A = np.mat(A)
# print('A:', A.shape)

# 最小二乘法
X = cal_least(A, h)

```

---

```
# print('最小二乘解:', X)

# 求分潮调和常数 h_M2, g_M2, h_S2, g_S2, h_K1, g_K1, h_O1,
g_O1
H_c, g_c, S_c = cal_harmonic_const(X)

# 将分潮调和常数赋值到类中
alti_point.get_H_g(H_c, g_c, S_c)
print('Done.')
```

```
# 输出文件 TP_hg
# output_hg(altimeterPoint)
output_hg_filter(altimeterPoint)

# 按轨道输出调和常数
# output_track(altimeterPoint)
output_track_filter(altimeterPoint)

# 程序运行时间
deltatime = datetime.datetime.now() - starttime
print('共剔除:', flag_n)
print("Finished...Time used:", deltatime.seconds, 'Seconds')
```