基于时间序列分析预测的黄河水沙监测模型

摘　要

在对黄河进行水文研究时，常通过探究水沙关系分析水沙通量的各种特性和变化规律，从而指导水资源分配和调水调沙、防洪减灾等措施。本文在对各附件数据进行预处理后，通过回归分析、小波分析、时间序列分析方法建立水沙关系模型和水沙通量模型，利用时间序列预测模型预测水沙通量的变化趋势，优化水文站采样监测方案，分析调水调沙效果。本文主要借助 Python、Excel、Matlab 等软件进行相关计算。

对于问题一，用“均值替补法”补充缺失数据后，分别研究含沙量与时间、水位、水流量的关系，并进行模型展示。针对含沙量-水位关系和含沙量-水流量关系分别进行**回归分析**，运用最小二乘法进行**分段拟合**，利用 Matlab 中多项式拟合得到分段函数，同时进行拟合度检验。最后建立年总排沙量、年总水流量模型，估算了近 6 年数据，又运用优化的**数值积分法**对上述模型进行了检验。

日期	2016	2017	2018	2019	2020	2021
年总排沙量(亿吨)	0.182	0.191	2.918	3.051	3.510	2.239
年总水流量(亿 m^3)	143.809	153.353	388.973	387.239	433.910	472.626

对于问题二，按月汇总附件 1 数据，将每月水沙通量的特性、变化规律通过**月流量均值**和**月排沙量均值**两要素描述。突变性利用“**累积距平法**”得到累积距平曲线，找到突变月份，利用**“M-K 突变检验”**，判断两要素突变时间；季节性研究通过**季节分解**原理得两要素的时间序列分解图，由此分析季节性变化规律；周期性研究运用**小波分析法**得到该水文站 6 年来每月“小波方差分析图”与“小波变换系数实数等值线图”，获得两要素各自的变化周期。根据各特性进一步得到水沙通量的变化规律。

对于问题三，针对水沙通量趋势的预测，建立 **SARIMA 预测模型**。将 2016-2020 年月流量数据作为训练集，预测 2021 年月水沙通量变化趋势。2021 年月水沙通量作为测试集，将模型预测结果与实际数据进行拟合，观察结果，将参数进行调整，找到最优参数，建立准确 **SARIMA** 预测模型，以此预测未来两年水沙通量变化趋势。根据未来两年趋势走向图，判断出未来两年水沙通量的周期性与突变性，制定最优采样监测方案。

对于问题四，从水沙通量变化、河底高程变化，水情监测数据变化三方面分析调水调沙实际效果。结合问题二水沙通量变化，调水调沙起到一定作用；结合水位、河底高程与起点距离的关系，分析了水面宽度与水深的变化；采用“**断面测量法**”计算了水道断面面积，采用“**流速面积法**”计算了水道平均流速，计算结果展示出面积增大、流速略增的趋势。得出若不调水调沙，河底高程会升高的结论。

关键词：回归分析，数值积分法，累积距平法，M-K 突变检验，小波分析，SARIMA 模型，流速面积法

一、问题背景与问题重述

1.1 问题背景

黄河是中华民族的母亲河。研究黄河水沙通量的变化规律对沿黄流域的环境治理、气候变化和人民生活的影响，以及对优化黄河流域水资源分配、协调人地关系、调水调沙、防洪减灾等方面都具有重要的理论指导意义。

1.2 已知条件

为深入研究黄河水沙通量的变化规律，选取位于小浪底水文站下游黄河某水文站的数据为研究依据。

1. 该水文站近 6 年的水位、水流量与含沙量的实际监测数据（附件 1）；
2. 该水文站近 6 年黄河断面的测量数据（附件 2）；
3. 该水文站部分监测点的相关数据（附件 3）。

1.3 问题重述

问题 1 研究该水文站黄河水的含沙量与时间、水位、水流量的关系，并估算近 6 年该水文站的年总水流量和年总排沙量。

问题 2 分析近 6 年该水文站水沙通量的突变性、季节性和周期性等特性，研究水沙通量的变化规律。

问题 3 根据该水文站水沙通量的变化规律，预测分析该水文站未来两年水沙通量的变化趋势，并为该水文站制订未来两年最优的采样监测方案（采样监测次数和具体时间等），使其既能及时掌握水沙通量的动态变化情况，又能最大程度地减少监测成本资源。

问题 4 根据该水文站的水沙通量和河底高程的变化情况，分析每年 6-7 月小浪底水文站进行“调水调沙”的实际效果。如果不进行“调水调沙”，10 年以后该水文站的河底高程会如何？

二、模型假设

1. 假设题目所给的数据真实可靠；

2. 假设水文站监测仪器因素导致的数据误差属于不可消除误差，忽略不计；

3. 假设所监测的黄河水只含水和沙，不考虑石子等其他物质的影响；

4. 假设选取的小浪底水文站下游的黄河水文站数据具有一定的代表性；

5. 假设黄河水中沙的密度为$2.5\times10^3 kg/m^3$；

6. 假设不考虑太阳蒸发对黄河水位的影响；

7. 假设不考虑地理因素及自然灾害的影响；

8. 假设“水位”和“河底高程”均以“1985 国家高程基准”（海拔 72.36 米）为基准面；

9. 假设附件中的“起点距离”以河岸边某定点作为起点。

三、符号说明

符号	符号含义
$Q_{沙}$	含沙量
t	时间
Z	水位
$Q_{水}$	水流量
Q	流量
$Q_{年沙}$	年总排沙量
$Q_{年水}$	年总水流量
$\overline{Q}_{月}$	月流量均值
$\overline{Q}_{月沙}$	月含沙量均值
$\overline{Q}_{月排}$	月排沙量均值
W_{CAj}	第 j 年的累计距平值
D_i	第i个起点距离
a_n	第i个起点距离
S_i	第i个梯形的面积
h_{i-1}	第i个梯形的上底，即第 i-1 个测深读数
h_i	第i个梯形的下底，即第 i 个测深读数
b_{ij}	第 i 次测速中第 j 个小长方形的高度
$\overline{v}$	水速均值
S	水到断面总面积

四、模型的分析、建立与求解

4.1 本文的研究思路

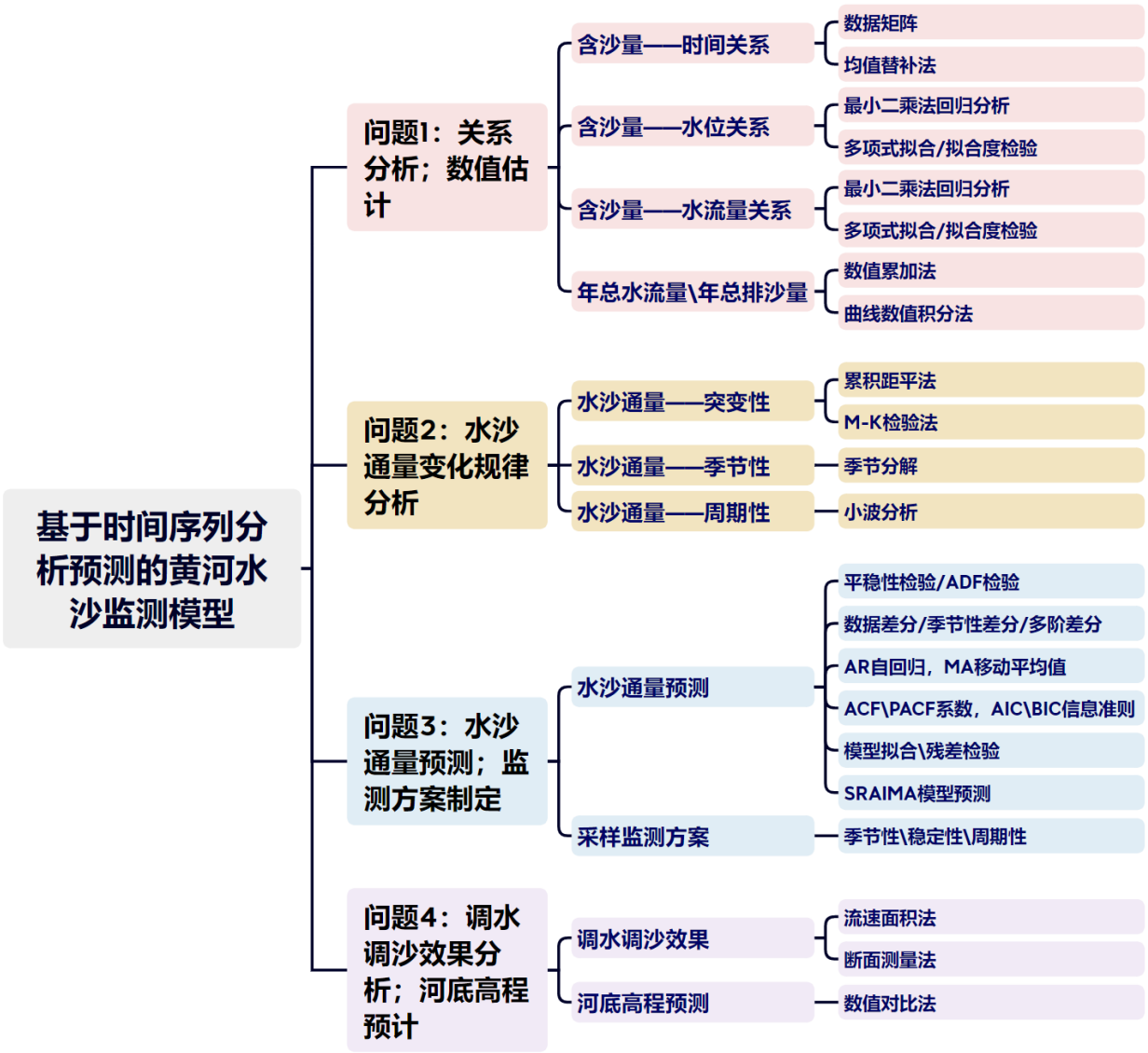

图 1. 本文的研究路径

4.2 问题一

分析该水文站黄河水含沙量的检测数据时，从时间、水位、水流量三方面入手，并对 6 年该水文站的年总水流量和年总排查量进行了合理估算。

4.2.1 数据预处理

(1)对附件各 sheet 进行“首行冻结”操作。

(2)通过“数据筛选”发现，2017、2018、2019、2021 年中有日期缺失情况。因此将缺失日期补全。

(3)观察“附件 1”，6 年中，第i年$(i=1\sim5)$12 月 31 日 24:00 与第i+1 年 1 月 1 日 0:00 的数据完全一致，因而去掉重复值。

4.2.2 问题一的模型建立与求解

1.研究含沙量与时间的关系

考虑到涉及时间的分析，各年数据的数量应具有的一致性，因此剔除掉 2016、2020 这两个闰年的 2 月 29 日数据。

对处理后的各年度数据进行汇总、转置操作，结果如表 1 所示：

表 1　含沙量数据展示汇总表

	1月1日	1月2日	1月3日	1月4日	1月5日	1月6日	1月7日	1月8日	1月9日	1月10日	1月11日	1月12日	1月13日	1月14日	1月15日	1月16日	1月17日	1月18日	…	12月23日	12月24日	12月25日	12月26日	12月27日	12月28日	12月29日	12月30日	12月31日
2016年	0.8	0.84	0.67	0.65	0.69	0.73	0.71	0.92	0.95	0.96	0.87	0.86	0.85	0.88	0.83	0.87	0.89	0.88	…	0.4	0.4	0.4	0.3	0.3	0.3	0.4	0.4	0.4
2017年	0.51	0.58	0.47	0.42	0.42	0.44	0.47	0.53	0.53	0.53	0.48							0.52	…	0.8	0.68	0.76	0.73	0.64	0.68	0.62	0.67	0.79
2018年	0.59	0.64	0.64	0.79	0.68	0.71	1.19	0.97	0.68	1.05	0.71	1.02	0.84	0.86	0.97	0.86	0.8	0.97	…	0.6	0.8	0.9	0.8	0.9	0.7	0.9	0.8	0.7
2019年	0.92	0.89	0.83	0.68	0.68	0.81	0.73	0.73	0.7	0.72	0.83	0.86	0.86	0.92	0.82	0.82	0.9	0.76	…	1.1	0.9	1.1	0.8	1.1	1.6	1.9	1.7	2
2020年	1.53	1.6	1.46	1.03	1.29	0.94	0.88	1.15	0.99	1.21	1.03	0.97	1.02	1	0.97	0.97	1.02	0.98	…		1.13	1.2		0.9	0.75	1.12		2.12
2021年	0.95		1.42					2.01		2.25				1.36				1.02	…			1.8					3.1	2.4
								$Q_{沙t-1}$		$Q_{沙t}$				$Q_{沙t+3}$														

其中，蓝色格子代表该日期的 8 点时段获得了含沙量数据，而空白格子表示该日期未获得数据，因此，采用“**均值替补法**”对空数据进行补充。观察到每年 1 月 1 日的数据都是完整的，则针对某年度，从 1 月 2 日起，逐一扫描每个$Q_{沙}$，若缺少第t日的含沙量$Q_{沙t}$，则向后探测各含沙量数据，直到探测到非空值停止，再考虑前后时点与缺失的第 t 日的间隔不同，使用时点间隔d设置权数，进行加

权计算，给出公式

$$Q_{沙t}=\frac{d}{d+1}Q_{沙t-1}+\frac{1}{d+1}Q_{沙t+d} \qquad \text{公式 1}$$

补充第t日的含沙量，Python 代码见附录。

在对附件 1 数据进行预处理并汇总的基础上，建立含沙量矩阵来描述 6 年中每日含沙量的监测数据：

$$\boldsymbol{Q}_{沙}=\begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1,365} \\ q_{21} & q_{21} & \cdots & q_{2,365} \\ & & \cdots & \\ q_{6,1} & q_{6,2} & \cdots & q_{6,365} \end{pmatrix}_{6\times 365} . \qquad \text{公式 2}$$

$Q_{沙}$ 中的任一元素 Q_{ij}(i=1~6，j=1~365) 表示第i年的第 j 天监测到的含沙量，i、j分别是年、日的逻辑编号。考虑对各年度的同一天的数据进行统计用以分析含沙量与时间的关系，并定义：

$$\bar{q}_j=\frac{1}{6}\sum_{i=1}^{6}q_{ij}, \qquad \text{公式 3}$$

从而获得每日含沙量向量

$$\bar{Q}_{沙}=\left(\bar{q}_1 \quad \bar{q}_2 \quad \ldots \quad \bar{q}_{365}\right)$$

根据以上模型，利用 Excel 软件的绘图功能，描述含沙量与时间的关系如图 2：

图 2　含沙量与时间的关系图

由于同日均值是各年度同一天的概括性度量，较单独的一天数据更方便表示含沙量的变化规律。观察并分析出不同时间段的含沙量特征如下：

(1) 10 月 15 日至来年 6 月 30 日含沙量较低，一般不超过 4kg/m^3，均值为

2.23 kg/ m^3，标准差为 0.96 kg/m^3；

(2) 7 月 1 日至当年 10 月 14 日含沙量较高，一般高于 4kg/m^3，均值为 6.45kg/m^3，标准差为 2.79 kg/m^3。

2.研究含沙量与水位的关系

1）数据准备

对在 4.2.1 中预处理之后的 6 年的所有数据,筛选剔除含沙量缺失的行，保留其余的 2154 行。从中选取“水位”、“含沙量”所在列，作为研究数据。

2）组距分组

上述分析中,“日期”属于离散型数据，故而进行单变量分组。与日期不同，“水位”属于较多的连续型变量，需要把水位值的变化范围分成若干个区间，每个水位值归属于相应的区间，故而进行组距分组。

组距分组时，为了获得较好的组距，采用统计学家 Sturges 提出的**经验公式**确定组数，公式为：

$$K \approx 1+\frac{\ln N}{\ln 2} \qquad \text{公式 3}$$

式中，K 代表组数，N 代表水位值的总数，这里 $N=2154$，$K=12.0728$，通常分组取奇数，因而分 13 组。

利用 Excel 函数“=(max(A)-min(A))/13”确定组距，进而获得各组上限、下限。

3）求解组内含沙量均值

利用 Excel 函数，“=VLOOKUP(B2,$F$3:$G$15,2,TRUE)”模糊查找小于水位值的每组下限,进而将 2154 个水位值归属为 13 组,获得第 k 组内各含沙量数据：

$$\boldsymbol{Q_k}=\left(q_{k1}' \quad q_{k2}' \quad \cdots \quad q_{kn}'\right). \qquad \text{公式 4}$$

式中，$i=1\sim13$，n 表示第 i 组的数据频数。

利用“数据透视表”功能，求得各组均值

$$\overline{q_k}' = \frac{1}{n}\sum_{j=1}^{n} q_{kj}\,. \qquad \text{公式 5}$$

结果如表 2 所示：

表 2 各组水位的含沙量均值

水位分组	组内含沙量均值(kg/m^3)
41.50~42.23组	0.576724638
42.23~42.58组	0.932594595
42.58~42.93组	1.637192727
42.93~43.29组	3.649313889
43.29~43.64组	5.293121387
43.64~43.99组	7.347054264
43.99~44.34组	16.66772277
44.34~44.69组	16.08071429
44.69~45.04组	14.67
45.04~45.40组	12.82211268
45.40~45.75组	10.68325
45.75~46.10组	6.438888889
46.10~47.00组	6.6568

取其变化趋势展示如图 3：

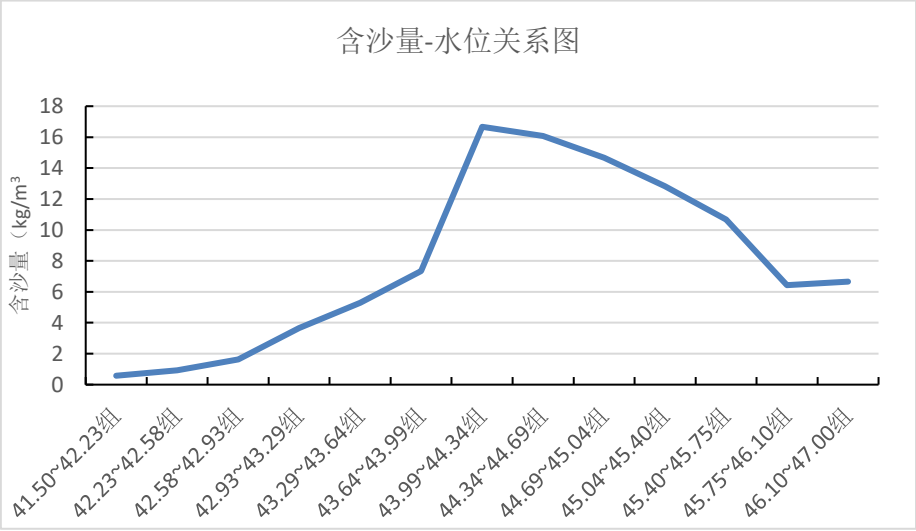

图 3　含沙量与水位的关系图

由上述图、表可以得出，前七组水位数据含沙量的数值是随着水位数值的增加逐渐增长的，特别是水位变化到 43.99-44.34m 时，含沙量数值陡增。而此后，水位再增加时，含沙量的值却开始缓慢下降，显然，变化规律很难拟合成一个函数，该关系曲线更符合分段函数的特点。

4）利用回归分析拟合“含沙量-水位”关系曲线

利用线性**最小二乘法**进行回归分析，借助 Matlab 分别对前后两段曲线进行 *m* **次多项式拟合**，由图中曲线变化趋势可以观察出，前段曲线符合三次函数特点，

后端曲线变化更接近直线。因此两段函数拟合时，选用三次曲线和一次线进行拟合，即多项式函数的次数 m 分别取 3 和 1，得到前后两段函数形成的分段函数表示为：

$$y=\begin{cases}3.73x^3-477.36x^2+20377.60x-289978.65, & 42.08\le x<44.34,\\ -5.46x+258.74, & 44.34\le x\le 46.24.\end{cases}$$

以水位分组后每组数据的均值为横坐标，取每组含沙量的均值为纵坐标，绘制拟合后的“含沙量-水位”分段函数图像见图 4：

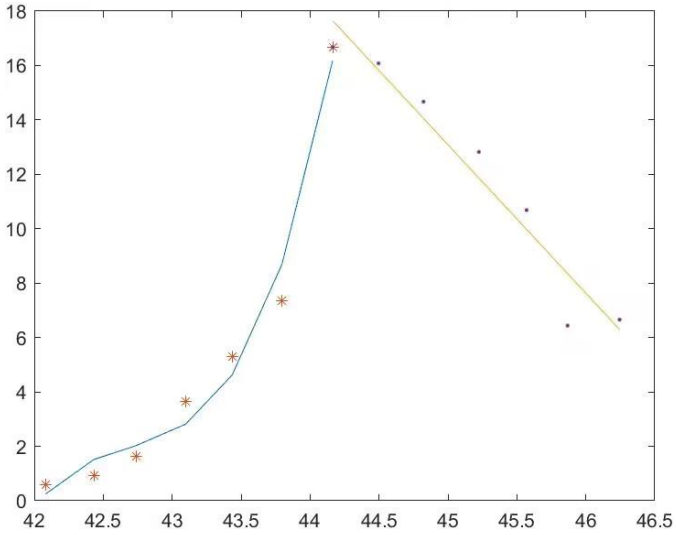

图 4. “含沙量-水位”拟合函数曲线

5）回归模型的**拟合度检验**

利用 Matlab 对两段拟合曲线进行“拟合优度检验”，程序见附件。左边拟合成三次曲线，其 $R^2=0.98$，右边拟合成直线，其 $R^2=0.941$。两段拟合的都非常理想。如果左边用二次曲线去拟合，其 $R^2=0.952$，可见三次曲线拟合效果好。

在第二段函数中含沙量数值随水位数值的增大而减小，这种情况的产生受多因素影响，其中，可能包含该地区进行了“调水调沙”这一因素所产生的作用。

3. 研究含沙量与水流量的关系

1）数据准备

在前期筛选剔除含沙量缺失的行后保留的 2154 行数据基础上，选取“流量”、“含沙量”所在列，作为两列研究对象。而问题一中需要研究“含沙量”与“水流量”的关系，因此给出水流量与流量的转换公式，从而补充附件 1 中含沙量对应的水流量数据作为研究数据。水流量模型：

$$Q_{水}=Q(1-\frac{Q_{沙}}{\rho_{沙}})\text{，（}\rho_{沙}=2.58\times10^{3}\text{kg/m}^{3}\text{）}$$ 公式 6

2）组距分组

“流量”数据的特点与“水位”类似，属于较多的连续型变量，因此同样需要把“水流量”值的变化范围分成若干个区间，每个水流量数值归属于相应的区间，故而进行组距分组。分组方法同“水位”数据分组，利用经验公式确定组数，求得水流量值同样可分为 13 组。再利用 Excel 函数“=(max(A)-min(A))/13”确定组距，进而获得各组上限、下限。

利用“数据透视表”功能，求得各组均值，结果如表 3 所示。

表 3 各组水流量的含沙量均值

水流量分组	组内含沙量均值(kg/m³)
0~548.12组	0.903063325
548.12~917.27组	1.893426439
917.27~1286.42组	4.244539249
1286.42~1655.57组	5.031612903
1655.57~2024.72组	8.182527473
2024.72~2393.87组	17.74513158
2393.87~2763.02组	16.24623656
2763.02~3132.16组	13.46295455
3132.16~3501.31组	12.06622222
3501.31~3870.46组	15.74454545
3870.46~4239.61组	10.02794118
4239.61~4608.76组	7.8375
4608.76~5000组	6.705416667

其变化趋势展示如图 5：

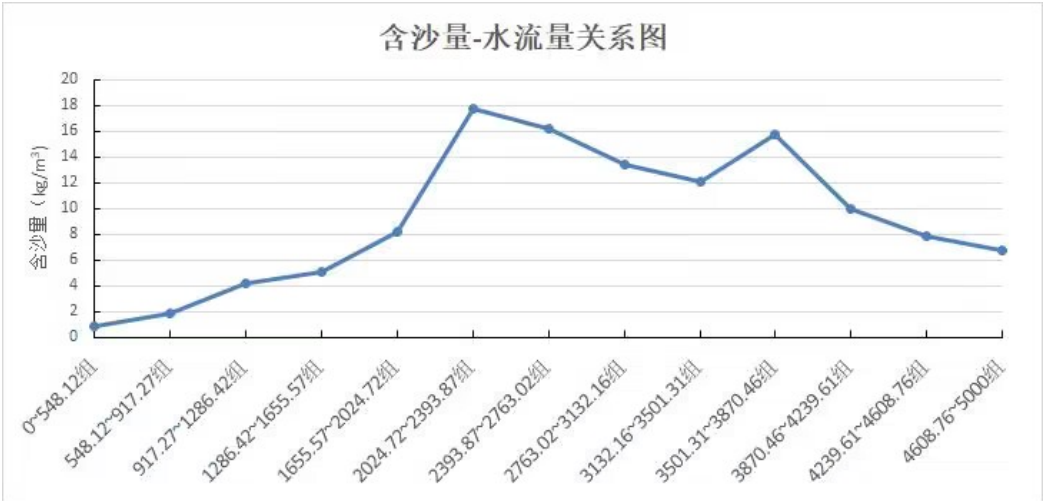

图 5. 含沙量-水流量关系图

同样利用线性**最小二乘法**进行回归分析，借助 Matlab 分别对前、后两段曲线进行 m **次多项式拟合**，多项式函数的次数 m 分别取 2 和 1，得到前后两段函数形成的分段函数表示为：

$$y=\begin{cases}2.37\times10^{-6}x^2+9.81\times10^{-4}x-0.022, & 363.71\le x<2576.96,\\ -0.004x+26.52, & 44.34\le x\le 44816.61.\end{cases}$$

以水流量分组后每组数据的均值为横坐标，取每组含沙量的均值为纵坐标，绘制拟合后的“含沙量-水流量”分段函数图像见图 6：

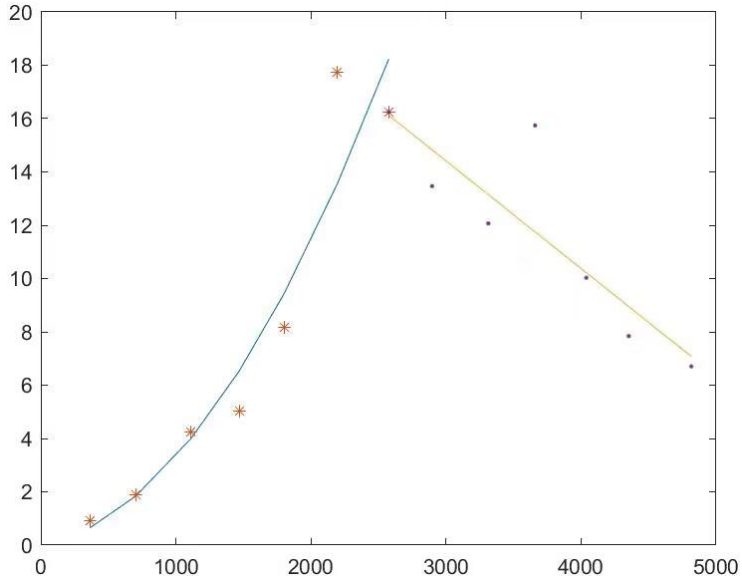

图 6. “含沙量-水流量”拟合函数曲线

利用 Matlab 对两段拟合曲线进行“拟合优度检验”，左边拟合成三次曲线，其 $R^2=0.924$，右边拟合成直线，其 $R^2=0.754$。左侧拟合得比右侧效果更理想。如果左边用二次曲线去拟合，其 $R^2=0.907$，可见三次曲线拟合效果好。

4. 估算近 6 年该水文站年总水流量和年总排沙量

（1）年总排沙量模型的建立与求解

给出第 i 年的年总排沙量的计算模型：

$$Q_{年沙}=10^{-11}\cdot\sum_{j=1}^{365}Q_{沙j}\cdot Q_j\cdot t \qquad \text{公式 7}$$

其中，$Q_{沙j}$ 表示第 j 天 8 点的含沙量，Q_j 表示第 j 天含沙量数据对应的流量，t 表示每天的时长，$t=86400$ 秒。参考水文研究中常用量纲，将年总排沙量的单位取为“**亿吨**”。利用上述模型，计算出近六年的年总排沙量，结果见表 4：

表 4 2016-2021 年的年总排沙量表

日期	2016	2017	2018	2019	2020	2021
年总排沙量(亿吨)	0.182	0.191	2.918	3.051	3.510	2.239

（2）年总水流量模型的建立与求解

年总水流量计算应利用前面给出的“水流量模型$Q_{水}=Q(1-\frac{Q_{沙}}{\rho_{沙}})$”计算求得，经过对该模型的分析，由于$Q_{沙}\ll\rho_{沙}$，因此$1-\frac{Q_{沙}}{\rho_{沙}}\approx1$，即**水流量的数值近似取流量的数值**。所以在估算近 6 年的年总水流量时，直接选取流量数据进行计算。

给出第i年的年总水流量的计算模型：

$$Q_{年水}=10^{-8}\cdot\sum_{j=1}^{n}\triangle t_j\cdot Q \qquad \text{公式 8}$$

式中，$Q_{年水}$为某年总排水量，设相邻两次监测时点之间为一个时段，$\triangle t$为时段长度，n为每年时段总个数。

模型求解时，首先观察到监测时点均为整点，因而使用 Excel 函数“=TEXT(A6,"h")”取出小时数。其次，利用“=IF(TEXT(A6,"h")-TEXT(A5,"h")>0,TEXT(A6,"h")-TEXT(A5,"h"),TEXT(A6,"h")-TEXT(A5,"h")+24)*3600”计算时段长度并将小时转换为秒，统一时间单位。最后，利用“=SUMPRODUCT(B5:B2383,C5:C2383)/10^8”返回年总排水量，并将量纲调整为“亿立方米”。结果见表 5：

表 5 2016-2021 年的年总水流量表

日期	2016	2017	2018	2019	2020	2021
年总水流量(亿 m^3)	143.809	153.353	388.973	387.239	433.910	472.626

（3）年总水流量与年总排沙量的优化模型——**数值积分法**

应用上述两个模型，分别求出了 2016-2021 年的年总水流量和年总排沙量，为了验证模型计算数值的准确性，对模型进行了进一步改进，运用水文研究中流量估计的模型——数值积分法进行估算（程序代码见附录），从而对模型进行了检验和优化，年总排沙量和年总水流量的计算结果如下表 6：

表 6 “数值积分法”估算的年总排沙量和年总水流量表

日期	2016	2017	2018	2019	2020	2021
年总排沙量(亿吨)	0.147	0.215	0.181	4.301	3.510	2.239
年总水流量(亿 m^3)	133.436	161.478	395.784	518.374	449.518	444.308

4.3 问题二

4.3.1 问题二的分析

1.数据预处理

为方便研究水沙通量的突变性、季节性和周期性的特征，我们将附件 1 中的数据进行按月汇总、分析。利用 Excel 处理附件 1 中数据，得到 6 年来月流量和月排沙量的均值$\overline{Q}_{月}$和$\overline{Q}_{月排}$（$\overline{Q}_{月排}=\overline{Q}_{月沙}\cdot\overline{Q}_{月}$）。研究每年中每月的水沙通量，主要通过$\overline{Q}_{月}$和$\overline{Q}_{月排}$这两个要素的变化规律描述水沙通量的各种特性。

2.水沙通量的突变性分析

利用“**累积距平法**”得到累积距平曲线，根据曲线的极值点找到每年中月水沙通量变化两要素各自趋势改变的月份。利用“**M-K 检验法**”对该水文站 6 年来月水沙通量两要素进行突变检验，从而判断水沙通量的两要素各自发生突变的时间，找出其突变性特征。

3.水沙通量的季节性分析

根据每月水沙通量的均值，将 6 年来流量数据和排沙量数据按照季节性分解原理分离成不同成分，其中包括：原始数据均值，长期趋势(Trend)，季节性(Seasonality)和随机残差(Residual)。

4.水沙通量的周期性分析

运用小波分析方法得到该水文站 6 年来每月小波方差分析图与小波变换系数实数等值线图，得到水沙通量的两要素各自的变化周期。

4.3.2 问题二的模型建立与求解

1. 水沙通量突变性模型的建立与求解

运用累积距平法对水沙通量进行突变性分析。**累积距平法**的思路是：在研究

的时段内，计算的累积距平值的数值越大，说明原数据越大于均值，得到的曲线在这一段呈现上升的态势，反之则曲线呈下降的态势。数据的变化特征是通过累积距平曲线的极值体现出来的。对于该水文站 6 年来每月水沙通量值序列，其累积距平值计算公式为：

$$W_{CAj}=\sum_{i=1}^{j}(W_i-\bar{W})\,, \qquad \text{公式 9}$$

其中，W_{CAj} 表示为水沙通量序列 W_i 在第 j 年的累积距平值，W_i 表示为第 i 年数值，n 表示为序列长度，$\bar{W}$ 表示为序列平均值。

Mann-Kendall 法可进一步运用于检验序列突变。首先构建时间序列的秩序列为：

$$S_k=\sum_{i=1}^{k}R_i\,, \qquad \text{公式 10}$$

其中，R_i 表示 $X_i>X_j(1\le j\le i)$ 的累积数。在时间序列独立的假设下，定义统计量：

$$UF_k=\frac{S_k-E(S_k)}{\sqrt{Var(S_k)}},\ k=1,2,\cdots 12\,. \qquad \text{公式 11}$$

其中，$UF_1=0$，$E(S_k)$ 和 $Var(S_k)$ 分别表示序列 S_k 的期望和方差，它们可由下式表示：

$$E(S_k)=\frac{n(n+1)}{4},\quad Var(S_k)=\frac{n(n-1)(2n+5)}{72}\,. \qquad \text{公式 12}$$

将时间序列逆序排列，重复上述过程，同时令 $UB_k=-UF_k$，且 $UB_1=0$。分析统计序列 UB_k 和 UF_k 可以进一步分析序列 x 突变的时间节点，显示突变的区域。若 $UF_k>0$，则表明序列呈上升趋势；反之序列呈下降趋势。如果 UB_k 和 UF_k 曲线出现交点，且交点在两临界直线之间，那么交点对应的时刻为突变开始的时刻。

首先根据 2016-2021 年流量和含沙量数据，整理得每年 12 个月中每月平均流量和含沙量，进而得到每月平均流量和排沙量数据。根据累积距平法将月均流量建立模型，利用 python 绘制首先得到月均流量累积距平曲线，如图 7 所示，通过对曲线图的分析，按照极值点将 6 年来流量变化分为两个阶段，分别是 2016-

2018 年的枯水期和 2018-2021 年的丰水期，分析累积距平图后发现流量变化的极值分别出现在 2018 年间，2019 年间，2021 年间。

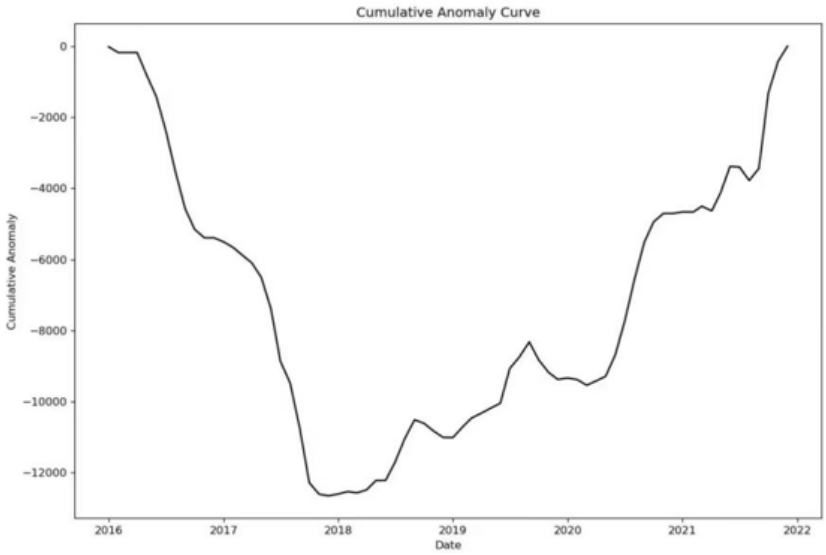

图 7. 2016-2021 年月均流量累积距平曲线

运用 M-K 检验法对水文站月流量进行“**突变性检验**”，得到图 7，为了方便观察规律，将在同一图中分析 2016-2021 年 72 个月流量和排沙量的某一性质。分析图中曲线变化趋势可明显观察到，在 2016 年和 2018-2021 年两阶段 $UF>0$，在此阶段内，该水文站的流量处于增加阶段，在 2016 年 5 月份之前和 2016 年 11 月到 2017 年 5 月间，$UF<0$，说明在这期间水文站的流量呈下降趋势。从图 8 可知在置信区间内月流量统计量曲线 UF、UB 在第 25 个月和第 30 个月间相交，说明在此对应时段内可能存在**突变点**。综合上文所做的累积距平过程图在 2018、2019、2021 年间出现的转折点，确定该水文站的月流量在 2018 年间发生突变。

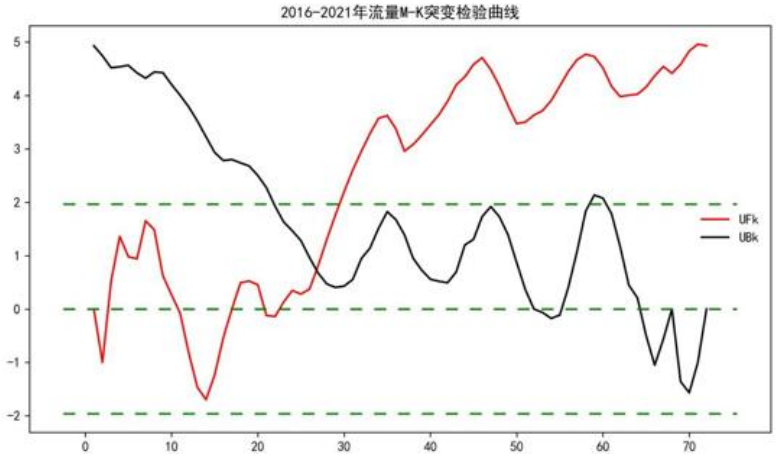

图 8. 2016-2021 年月均流量突变性检验

将 2016-2021 年间月排沙量根据利用累积距平法建立模型，得到图 9 排沙量累积距平曲线图，由于累积距平曲线表现出不稳定波动的趋势，故以起伏变化的最大值点对应时间进行趋势变化的区分，分别是 2016-2018 年中与 2021 年下半年的枯沙期和 2018 年中-2021 年中的丰沙期。分析累积距平曲线（如图 9）后可见月排沙量变化的极值点分别出现在 2018 年间，2019 年间和 2021 年间。

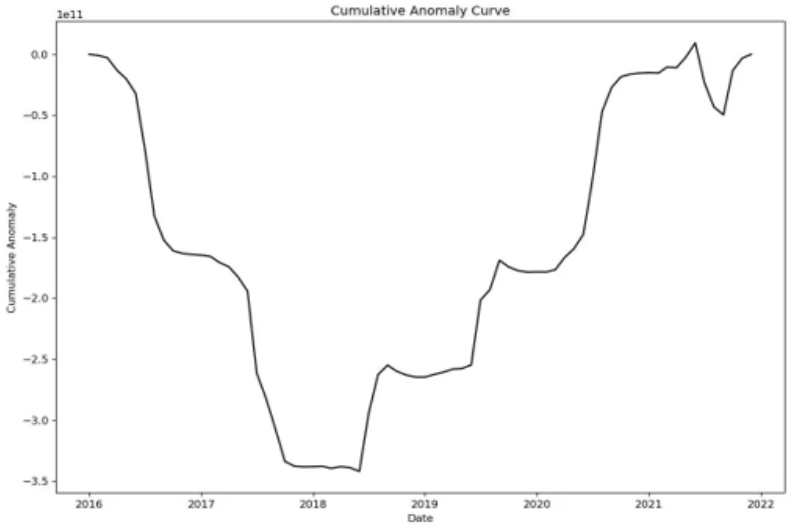

图 9. 2016-2021 年月均排沙量累积距平曲线

为了得到更精确的突变时间，利用 M-K 突变检验方法，分析 6 年来每月平均排沙量，得到图 10，根据曲线变化趋势，观察到在第 22 个月之后$UF>0$，说明排沙量在此期间呈增大趋势，在第 22 个月之前，曲线变化趋势波动明显，但是总体是$UF<0$，说明排沙量减少。*UF*、*UB* 曲线在第 25 个月和第 30 个月之间出现交点，说明突变点发生在此时间段内。与排沙量累积距平图得到的结果综合比较，可以发现，水文站月排沙量在 2018 年间发生突变。

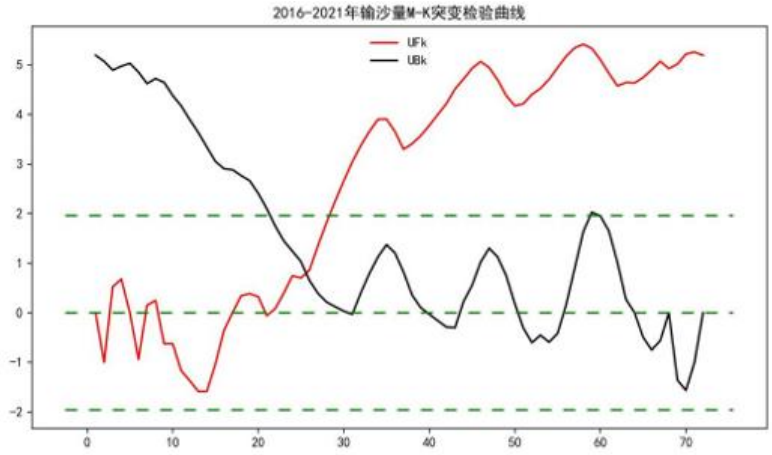

图 10. 2016-2021 年每月平均排沙量突变性检验

该水文站月流量和月排沙量累积曲线表现较不规律的主要原因是由于小浪底水库对下游水沙要进行调控。小浪底水库要有效调节下泄水量，实现黄河下游不断流，并且要在汛前调水调沙，增加对下游河道的冲刷。

2. 水沙通量季节性模型的建立与求解

通过 2016-2021 年每月平均流量数据，通过季节分解原理得到 6 年来关于流量的时间序列分解图，如图 11：

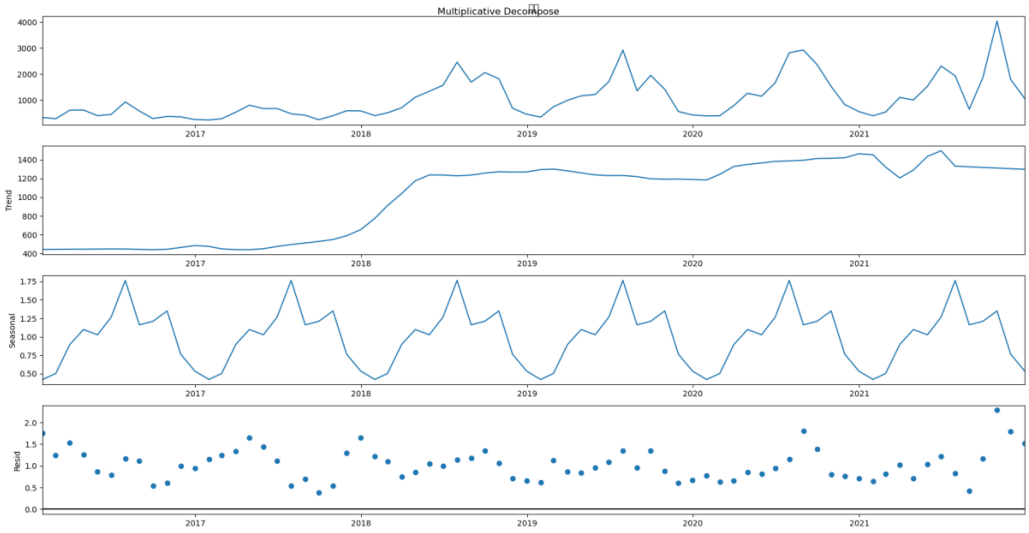

图 11. 2016-2021 年每月平均流量时间序列分解图

根据图中趋势(Trend)曲线，可以观察到水流量在 2018 年呈明显上升趋势，并且在 2018 年 5 月份之后趋于稳定，在 2021 年水流量再次出现小范围波动。从季节性(Seasonality)曲线图可以看出，每年春季流量逐渐增大，夏季流量出现急剧上升趋势，在 6、7 月份达到最大，汛期过后流量急剧减少，秋季流量较平稳，冬季阶段流量骤减。

根据 2016-2021 年每月平均排沙量数据，通过季节分解原理得到 6 年来关于排沙量的时间序列分解图，见图 12，根据图中曲线分别分析趋势和季节性，由趋势(Trend)曲线可以看出排沙量在 2018 年出现急剧上升，自 2018 年 5 月之后趋势趋于平缓，2021 年开始排沙量开始呈下降趋势。根据季节性(Seasonality)曲线图可以看出，每年春季排沙量呈逐步增长趋势，在进入夏季之前出现小范围波动；进入夏季之后排沙量急剧上升在 6，7 月份达到最大值，汛期之后排沙量急剧下降；进入秋季出现小范围上升趋势，但不会超过汛期最大排沙量；冬季排沙量逐步下降。

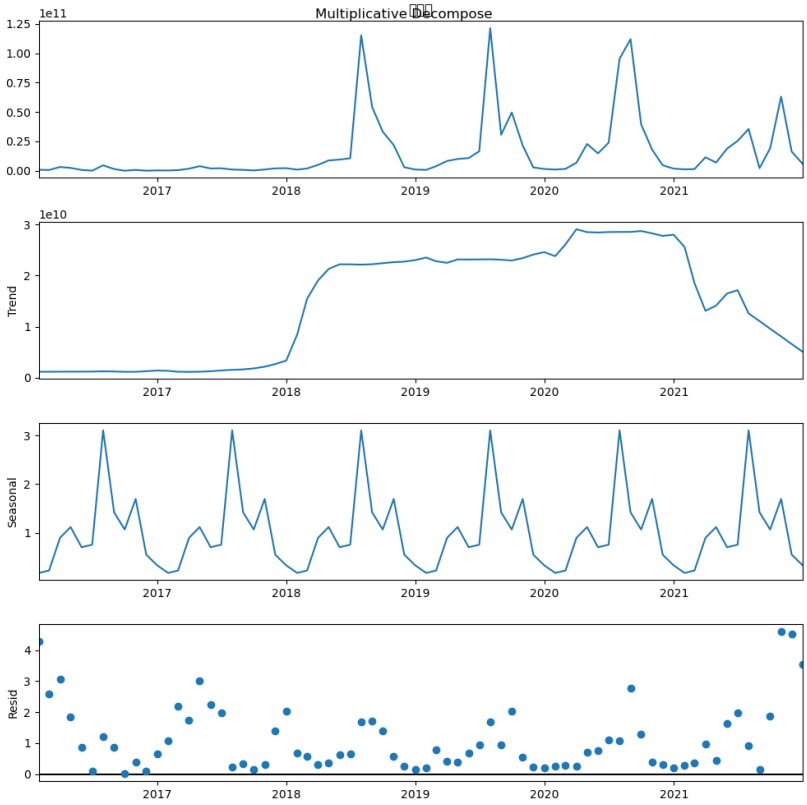

图 12. 2016-2021 年每月平均排沙量时间序列分解图

3. 水沙通量周期性模型的建立与求解

小波分析是通过伸缩、平移等运算对数据逐步地多尺度地细化，进而满足不同特征数据的分析需求，获取不同条件下的变化周期的方法。通过小波变换方程对流量时间序列进行连续小波变换，可以得到小波变换系数，提取小波系数实部，绘制等值线图，从等值线图中识别水文序列在时间上的周期性特征。通过小波方差进行检验可以识别流域流量序列的主要周期，小波方差图上波峰对应尺度为主周期。

选取 2016-2021 年小浪底水库每月流量相关数据，使用 Matlab 编程进行 Morlet 小波变换，得到小波变换系数。通过 origin 绘图工具得到小波变换系数实部等值线图，得到图 13。图中的横坐标对应时间，即所选取的研究时段各个月份，纵坐标对应频率，即对应不同的小波周期。通过图像可以看出该水文站流量值数值较大对应实部值大于零处，这些大于 0 处的部分在图中以实线区反映出来；相反的流量值较小时，是用图中虚线区块显示出来。而在图中的实线区与虚线区相交的小波系数等于零的位置，则代表数据所处年代发生了突变。

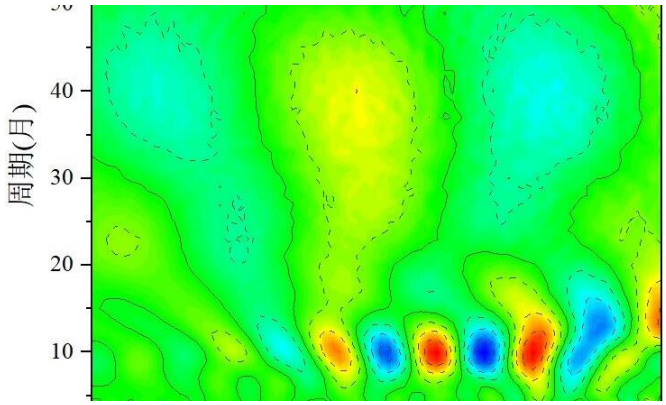

图 13. 2016-2021 年每月平均流量小波系数实部等值线图

根据图 13 观察出，小浪底水文站周期变化主要有 2 种，分别是 3-5 个月、10 个月。所有主要周期变化中以 10 个月为周期尺度的时间序列表现最为稳定，在此周期有明显的明暗交替的丰枯振荡变化。

将 2016-2021 年小浪底水库每月流量进行小波系数方差分析，得到图 14，小波方差图上极值的数值越大，表明该极值点对应的波动越强烈，这也表示其对方差的贡献越大。由图 13 可以观察得到，在 3 个月，10 个月，38-40 个月位置有明显峰值，其峰值与实部等值线图中的周期规律很好对应该水文站序列变化的第一主周期为 10 个月；第二主周期、第三主周期分别为 40 个月和 3 个月。

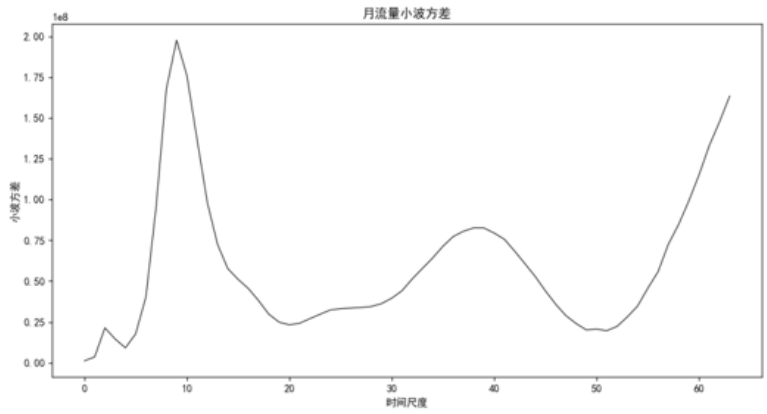

图 14. 2016-2021 年每月平均流量小波系数方差图

图 15 是 2016-2021 年每月排沙量小波系数实部等值线图，根据图中信息可以看出其周期变化主要有 2-3 个月、10 个月。

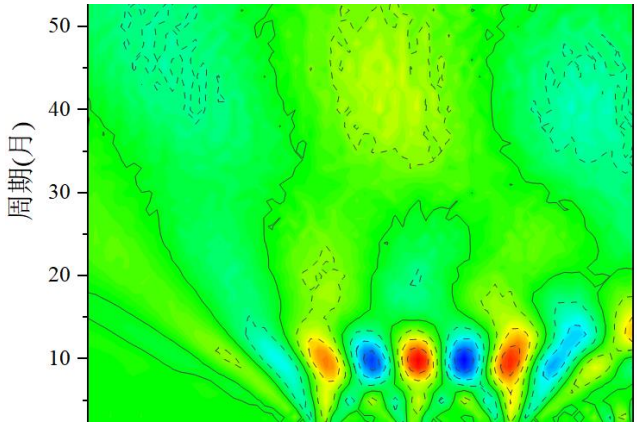

图 15. 2016-2021 年每月平均排沙量小波系数实部等值线图

将 2016-2021 年小浪底水库每月排沙量进行小波系数方差分析，得到图 16 ，根据方差曲线得到，在 4 个月，10 个月，40 个月位置有明显峰值，其峰值与实部等值线图中的周期规律很好对应该水文站序列变化的第一主周期为 10 个月；第二主周期为 40 个月、第三主周期为 3 个月。

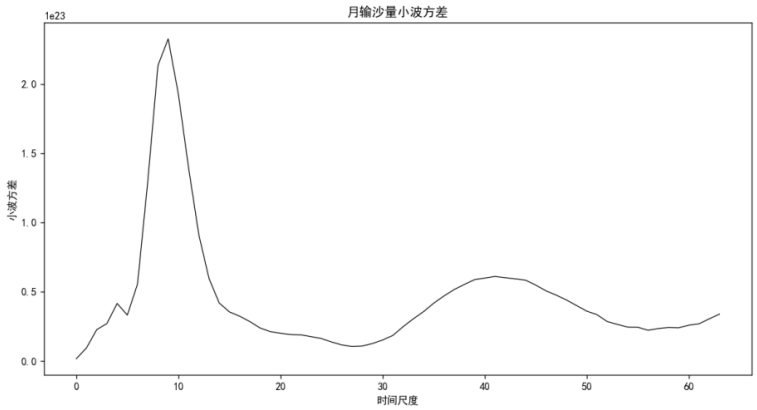

图 16. 2016-2021 年每月平均排沙量小波系数方差图

根据突变性、季节性和周期性综合分析可知，流量在 2018 年急剧增长，这是由于 2018 年开始黄河水量增加，进入丰水期，由于小浪底水文站的调水调沙功能，排沙量在 2018 年同样剧增；根据季节性分析结论，流量和排沙量在夏季最多，汛期时期达到峰值，春季与冬季的水量与排沙量差距不明显，秋季的水量与排沙量变化不明显；通过数据的周期性分析得到，流量与排沙量主要以 10 个月为周期变化，其次以 40 个月，3 个月为周期变化。

4.4 问题三

4.4.1 问题三的分析

问题三是根据问题二水沙通量的变化规律研究预测其变化趋势。ARIMA 时间序列预测模型是水文研究中常用的预测模型，它是将非平稳时间序列转化为平稳时间序列，然后将因变量仅对它的滞后值以及随机误差项的现值和滞后值进行回归所建立的模型。

进行序列预测时，对序列$\{x_t\}$进行小波分解后得到各层细节系数和逼近系数，重构后的细节系数一般可视为平稳过程来处理，通过模型识别可用选用 AR(p)、MA(q)或 ARMA(p,q)等模型进行预测，逼近系数表示序列的趋势或走向，一般为非平稳序列，因此不能直接采用 ARIMA 模型，需要经过季节差分采取 SARIMA 模型。

4.4.2 问题三的模型建立

时间序列预测是数据分析预测中经常遇到的问题，本题主要研究 ARIMA 模型，也是最常用的时间序列模型。本剧第二问的水沙通量变化规律，时间序列具有突变性、周期性、季节性等特征，因此可能无法直接使用 ARIMA 建立模型。

因此建立**季节差分自回归移动平均模型——SARIMA 预测模型**。建模流程如下图 17 所示。

图 17. SARIMA 原理分析图

4.4.2 问题三模型的求解

1.SRAIMA 模型预测：

1） ARIMA 模型更新

对成熟的 ARIMA 进行时间序列预测，借助 MATLAB 程序进行简单分析，得到预测图 18 如下：

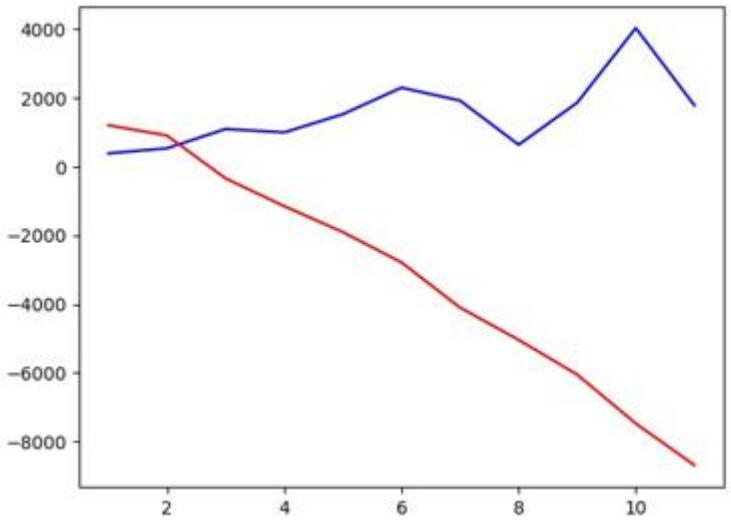

图 18. ARIMA 预测图

分析可得，由于时序数据中存在季节/周期性等的特征，所以 ARIMA 模型无法识别数据到底是哪种季节/周期特征，导致预测输出为一条直线。最后，已知有季节/周期性数据存在，所以就不能再用那个简单的 ARIMA 模型来处理。因此我们使用 SARIMA。

2） 建立水沙通量预测序列数据

根据问题 2 水沙通量特性研究，发现水流量与排沙量都存在突变性、季节性、周期性等规律，且规律基本一致，在本节中我们将水沙通量综合考虑为流量。下面引入 ARIMA 序列预测模型建立对未来水沙通量的趋势预测。

首先选取 2016-2020 年此 5 年的月流量和月排沙量作为原始数据，得到两个连续时间下的 60 组数据集合，将数据按 3 个月为一季节划分尺度，建立季节性自回归移动平均模型。

3） 检查时间序列的平稳性

为进行 SARIMA 序列预测，数据需要检查时间序列的平稳性，获取被观测

系统时间序列数据，对数据绘图，观测是否为平稳时间序列；此时可进行平稳性测试， 我们使用 Augmented Dickey-Fuller 单位根测试测试平稳性。对于平稳的时间序列，由 ADF 测试得到的 p 值必须小于 5%。如果 p 值大于 0.05 或 5%，则可以得出结论：时间序列具有单位根，这意味着它是一个非平稳过程。对于非平稳时间序列要先进行 d 阶差分运算，化为平稳时间序列。对前 60 组数据进行处理得到如下图 19：可以看出序列为非平稳序列。

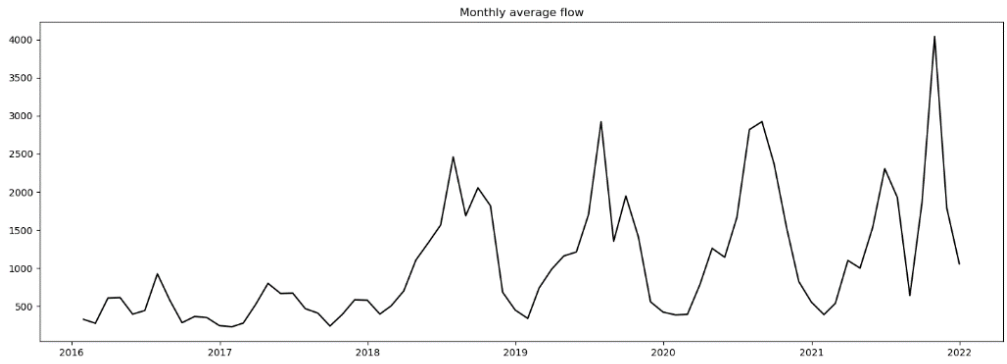

图 19. 原时间序列图

（2）差分操作

针对非平稳时间序列，进行序列的差分操作，差分操作是通过计算当前观测值与前一个观测值之间的差异来消除非平稳性。对于本题的序列，我们需要先消除其季节性，因此进行季节差分处理，得到如下图 20 数据：

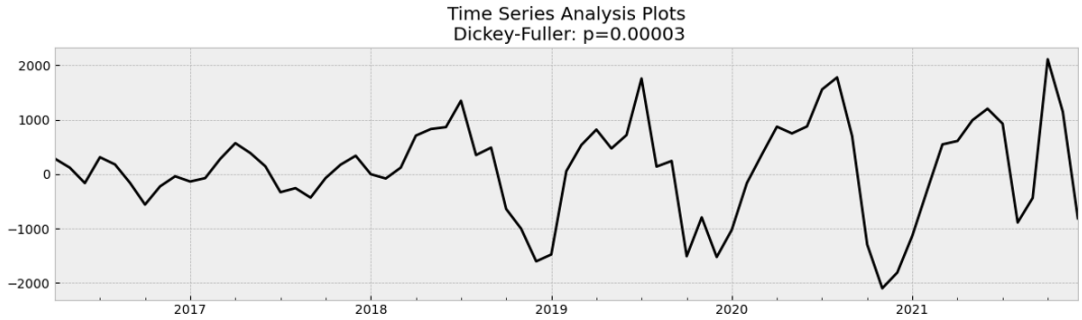

图 20. 季节差分处理图

观察发现序列仍具有一定的不平稳性，因此进行差分处理。此时可以连续多次应用差分方法，产生“一阶差分”,“二阶差分”等，直到得到平稳的差分序列。在我们进行下一步之前，我们应用适当的差分顺序（d）使时间序列平稳。差分公式如下：

一阶差分：$X_T - X_{T-1}$

二阶差分：$\nabla X_T - X_{T-1} = (X_T - X_{T-1}) - (X_{T-1} - X_{T-2})$ 公式 13

K步差分：$X_T - X_{T-K}$

将原始数据进行一阶差分得到下图 21，为实现更优效果，进行二阶差分，如图 21。从图 22 中可以看出，二阶差分之后数据趋势逐渐平稳。

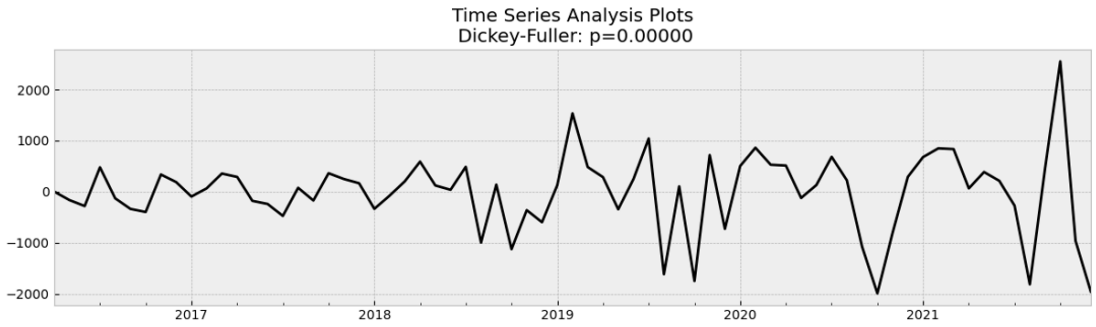

图 21. 一阶差分序列图

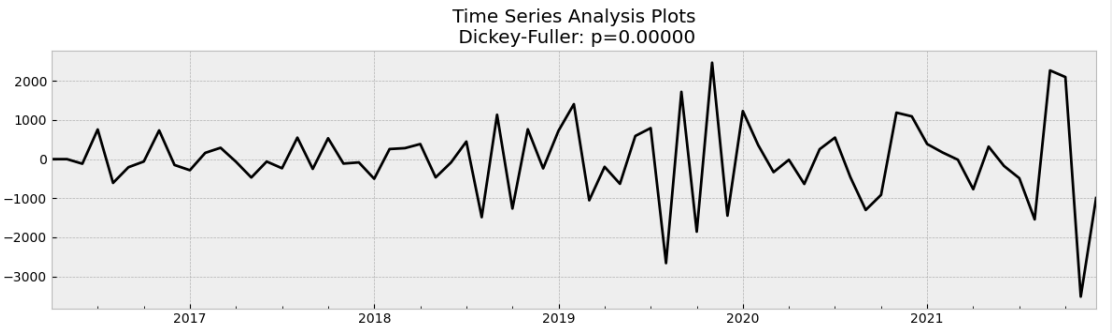

图 22. 二阶差分序列图

（3）模型选择：

经过第二步处理，已经得到平稳时间序列。然后开始对得到的模型进行模型检验。根据差分操作后的时间序列，可以确定 SARIMA 模型的参数。季节性 SARIMA(p,d,q)(P,D,Q,s)模型由 7 个参数表示：自回归阶数(p)、差分阶数(d)、移动平均阶数(q)和季节自回归的阶数(P)、季节差分阶数(D)、季节移动平均阶数(Q)。

一个 p 阶自回归模型可以表示如下：

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t, \quad \text{公式 14}$$

这里的ε_t是白噪声。这相当于将预测变量替换为目标变量的历史值的多元回归。将这个模型称为 AR(p)模型--p 阶自回归模型。可以使用自相关函数（ACF）和偏自相关函数（PACF）的图形，以及信息准则（如 AIC、BIC 等）等指标来选择合适的模型参数。

识别 AR 模型的 p 阶：对于 AR 模型，ACF 将以指数方式衰减，PACF 将用于识别 AR 模型的顺序（p）。如果我们在 PACF 上的滞后 1 处有一个显著峰值，那么我们有一个 1 阶 AR 模型，即 AR（1）。如果我们在 PACF 上有滞后 1,2 和 3 的显著峰值，那么我们有一个 3 阶 AR 模型，即 AR（3）。

识别 MA 模型的 q 阶：对于 MA 模型，PACF 将以指数方式衰减，ACF 图将用于识别 MA 过程的顺序。如果我们在 ACF 上的滞后 1 处有一个显著的峰值，那么我们有一个 1 阶的 MA 模型，即 MA（1）。如果我们在 ACF 上的滞后 1, 2 和 3 处有显著的峰值，那么我们有一个 3 阶的 MA 模型，即 MA（3）。

下图展示了季节差分及二阶差分后的 ACF 及 PACF 图像。

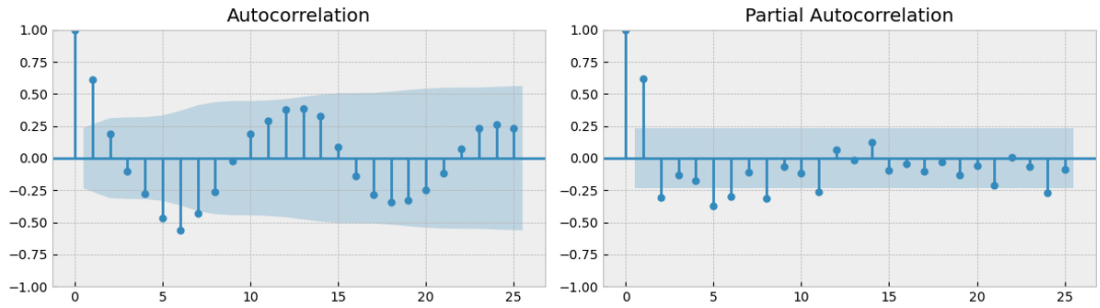

图 23. 一阶差分 ACF、PACF 图

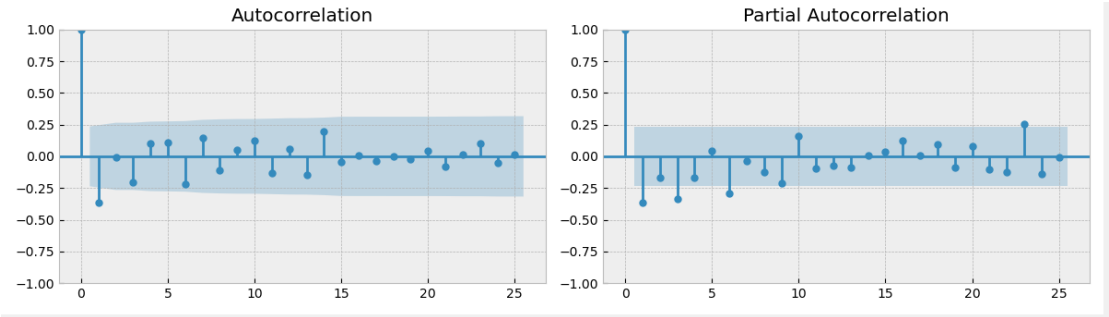

图 24. 二阶差分 ACF、PACF 图

ARIMA 模型结构的初步识别主要依据 ACF(自相关系数)和 PACF(偏自相关系数)的衰减情况。根据模型识别方法，p 阶自回归过程为 ACF 是拖尾的(即呈指数逐步衰减)，而其 PACF 在步距 p 之后截尾；相应地，q 阶滑动平均过程为 ACF 在步距 q 之后截尾，而 PACF 拖尾。

热力图定阶结果如下图 25 所示：

图 25. 热力图

根据 ACF 和 PACF 图定阶，确定出 p=2,q=3。

（4）模型拟合

由以上分析得到：自回归阶数(p)=2；移动平均阶数(q)=3；此外可以确定地是，季节性判定 S=6，差分阶数(d)=2，做了 2 阶差分，季节差分阶数(D)=1，做了季节差分。

得到 SARIMA 模型。为了预测的可靠性，接下来，借助 AIC 信息准则及 BIC 信息准则，利用 SARIMA 模型，带入程序来选择最优的参数。对差分后的时间序列进行模型拟合，算法会估计模型的未知参数，以最小化预测误差。在四个自变量的 81 组候选参数中，得到最优参数组为[2, 3, 0, 1]，即自回归阶数(p)=2，移动平均阶数(q)=3，和季节自回归的阶数(P)=0，季节移动平均阶数(Q)=1。

基于 SARIMA 模型对水沙通量数据进行预测，并将实际数据与预测数据进行拟合。得到图 18，此时贴合度为 90%。贴合度如图 26 所示：

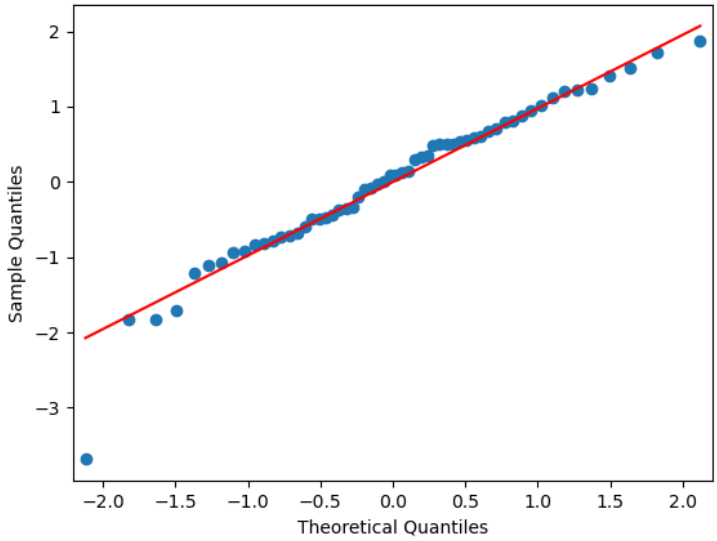

图 26. 贴合度图

（5）模型诊断：

为检验模型的准确性，考虑选取模型残差的白噪声指标对拟合的 SARIMA 模型进行 D-W 检验，以验证模型的适用性，D-W 检验可以检查残差序列是否满足白噪声假设，即不存在自相关性。根据上文得到的结果，我们可以得到 D-W 检验值为 2.020419531977266，已知当 D-W 检验值接近于 2 时，不存在自相关性，所以我们选取的 SARIMA 模型对数据拟合充分。

（6）模型预测

使用训练好的 SARIMA 模型对序列进行内预测，同时为未来的时间序列进行外预测，如下图 27 展示：

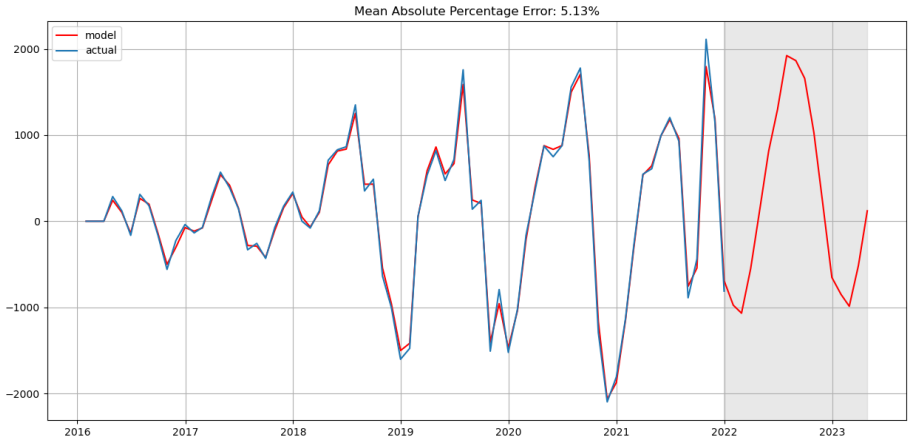

图 27. 序列预测结果图

根据内预测数据我们发现，建立的 SARIMA 模型与有良好的预测性，数据能够接近真实值。此时，外预测图像展示了未来两年的水文站水沙通量情况。从图中可以看出，未来两年水文站水沙通量在仍有一定的季节性、周期性等特征，在未来一段时间内呈现上升趋势，22 年具有极值点，在到达极值点后开始呈下降趋势。

2.制定采样监测方案

根据上述预测，得到未来两年的水沙通量数据，根据图像分析变化趋势，制定有效监测水沙动态的方案。在以下方案中平均采样次数保持在每 3-5 天采样一次。

（1）在秋季时段内，水沙通量具有一定的稳定变化趋势，则只需在该时间段内测得水沙通量数据连续稳定 10 天后可以每 3 天监测一次，在监测到水沙通量数据开始发生突变时，开始采取每天监测一次选取起始点和终止点的数据即可获取在时间趋势内的稳定变化数据情况。

（2）每年进入夏季之前，水沙通量会发生突变，则从 5 月下旬开始每天早上 8:00 和 20:00 一天监测两次，6-7 月每天在早上 8:00、中午 14:00 和晚上 20:00 监测三次。

（3）每年春季和冬季时间，水沙通量具有一定的周期性，在每年春季水沙通量逐步增长，在此期间，采取每天监测一次的采样方案，在 5 月下旬调整至每天监测 2 次。水沙通量在每年冬季水沙通量逐渐减少，在 10 月底开始监测方案由每 3 天采样一次调整到每天监测一次。

4.5 问题四

4.5.1 问题四的分析

由问题一结果看到，2016、2017 两年的总排沙量、总水流量不及 2018-2021 年的 10%，说明 6 年间前期调水调沙能力不如后期。下图展示了 2016 年 6-7 月进行的调水调沙之前、之后的黄河河岸某定点处主河槽的河底高程情况。

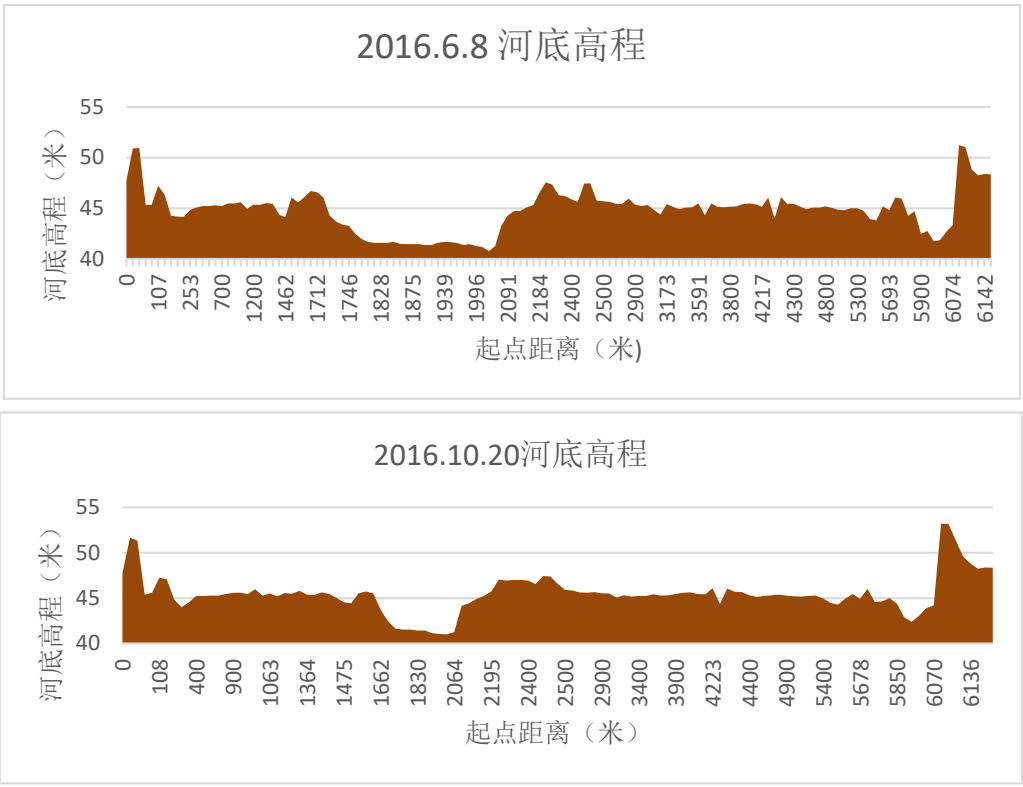

图 28. 2016 年检测日河底高程变化图

借助问题二该水文站月均流量、月均排沙量的计算结果，从主河槽剖面形态变化、平均下切高度两方面入手，分析了附件 2 黄河断面河底高程的变化情况；从测点水面宽度深度、流动力（水流速）、冲击力（水动能）三方面入手，分析附件 3 该水文站部分监测点数据。从而展示“调水调沙”的实际效果，即：小浪底下游河段具有河槽稳定、河床开阔、加深的趋势，黄河下游的“悬河”风险呈现降低态势，过洪能力提高。

4.5.2 问题四的模型建立与求解

1. 从水沙通量变化研究调水调沙效果

每年 6-7 月小浪底水库进行调水调沙，为研究调水调沙效果，借助问题二中计算的该水文站每年每月的月均流量和月均排沙量，提取出每年 5-10 月的对应数据作为水沙通量的主要数据，绘制出图 29 和图 30 观察分析 6-7 月调水调沙前后数据变化。

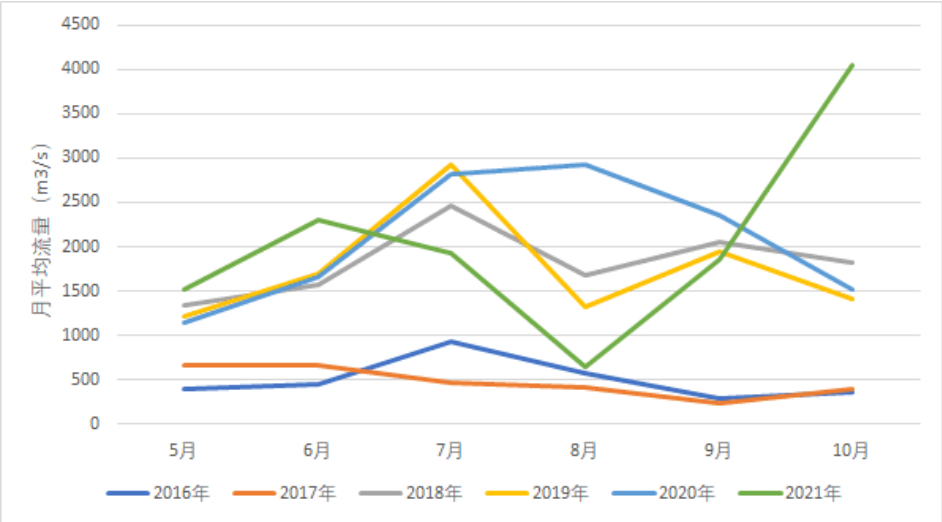

图 29. 六年中 5-10 月的月均流量走势图

观察图 29 可以看出，每年从 6 月到 7 月，月均流量都在增大，一般 7 月达到峰值，主要是因为 7、8 月份往往是汛期。理论上讲，6-7 月调水调沙后河道变宽变深，流量应该增大，但由于汛期降水多，同时上游冲刷来的泥沙多，水沙量的增速大于调水调沙后的有利影响，所以从图像上看，6-7 月份之后，月均流量数值在降低，但过了汛期之后，特别是 8-9 月之后，调水调沙的作用就逐渐显现出来，由于河道畅通，水沙流速增大，月均流量在多数年份有所增加。如果不进行调水调沙，七月达到的峰值会更高，河道快速变窄变浅，不利于排水排沙。

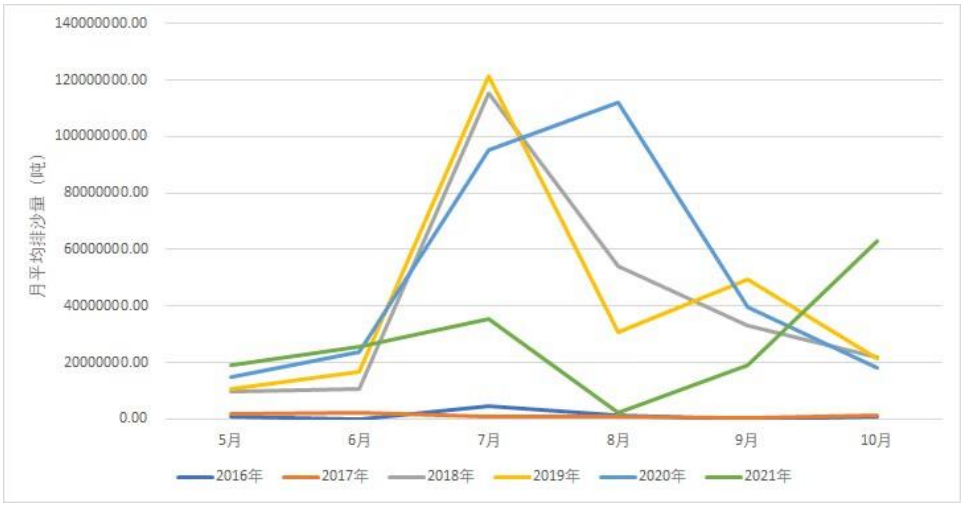

图 30. 六年中 5-10 月的月均排沙量走势图

结合图 29 和图 30 可以看出，月均排沙量的变化与月均流量的变化趋势类

似，每年从 6 月到 7 月，月均排沙量都在增大，主要是七八月汛期的影响。调水调沙后，虽然月均排沙量数值在降低，但过了汛期之后，调水调沙的作用就逐渐显现出来。

问题二研究水沙通量的突变性时，发现 2018 年是水沙通量的突变年，可能是当地政府采取了技术更好的调水调沙方式，因此效果较以往明显。

2. 从河底高程变化研究调水调沙效果

从附件 2 的数据出发，选取 2016 年和 2019 年河底高程与水位和起点距离的数据，分析在这两年调水调沙前后水面宽度与水深的变化。2016 年 6 月 8 日与 2016 年 10 月 20 日河底淤沙的截面图在图 15 中直观地可以看出，经过 6、7 月份调水调沙，河底高程明显降低。为了检验我们的观察，根据附件 1 分析 2016 年 6 月 8 日与 2016 年 10 月 20 日两天的平均水位分别为 42.201m，42.396m，水位轻微增长但变化不明显。我们将河底淤沙建立积分模型求得 6 月 8 日的淤沙截面面积为 276753.8425 m^2，10 月 20 日淤沙截面的面积为 277733.05 m^2，综合分析水位和淤沙截面面积的变化验证了我们的观察。

同样，我们将 2019 年 4 月 13 日和 2019 年 10 月 15 日的淤沙截面图作图如图 16，由于 2019 年 4 月 13 日数据较粗略，只能看出河底高程最低高度无明显变化，所以不能从淤沙截面面积分析变化，故从水深变化以及水面宽度变化来描述结果。从水深角度出发，4 月 13 日平均水位为 43.167m，10 月 15 日平均水位为 43.802m，水位轻微上升；从水面宽度角度出发，在附件 2 中可以看出 4 月 13 日起点距离在 1681m 和 2072m 之间为水面宽度，约为 391m，10 月 15 日起点距离在 1600m 和 2055m 之间为水面宽度，约为 455m，水面宽度明显加宽。

综合以上结果分析得：小浪底水库调水调沙使得水面宽度变宽，故而水沙通量加大，确保了汛期黄河防洪安全；调水调沙前后河底高程无明显变化，实现了水库排沙减淤的调度目标，调水调沙带来的“清淤效果”极为显著，为下游通航提供了最基本的保障。

3. 从“水情监测数据”研究调水调沙效果

观察附件 3 数据中河槽内各处水深不同，且“测深点”、“测速点”间隔出现，考虑以测深垂线为边界，将若干条边界之间的面积视为直角梯形（前后两个三角形视为直角梯形的特例），分别计算出每一部分的面积（中面积），求和即为水道

断面总面积。示意图 31 如下：

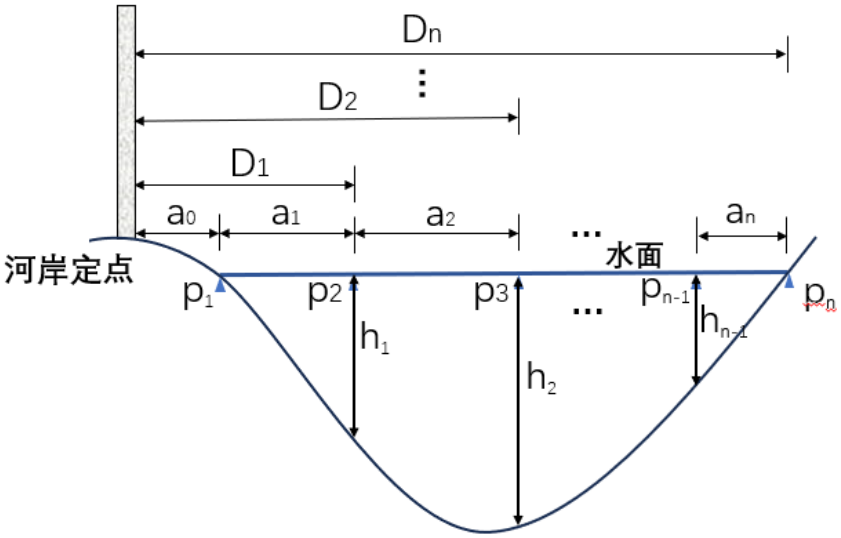

图 31. 求解水道断面面积示意图

记 n 次测深监测中，$P_i(i=1\sim n)$ 为第 i 个测深点，D_i 表示监测点 P_i 起点距离，定义 a_i 为第 i 个直角梯形的高度，公式为：

$$\begin{cases} a_0 = \text{水深为0时的起点距离}; \\ a_1 = D_1 - a_0; \\ a_2 = D_2 - D_1 \\ \cdots\cdots \\ a_n = D_n - D_{n-1} \end{cases} \quad \text{公式 15}$$

定义 h_i 为第 i 个直角梯形的下底，公式为：

$$\begin{cases} h_0 = 0 \\ h_i = \text{第}\, i \,\text{个测深点读数}\,(i=0\sim n) \end{cases} \quad \text{公式 16}$$

定义 S_i 为第 i 个直角梯形的面积，公式为：

$$S_i = \frac{1}{2}\left(h_{i-1} + h_i\right)\cdot a_i \quad \text{公式 17}$$

定义 S 为水道断面总面积，公式为：

$$S = \sum_{i=1}^{n} S_i \quad \text{公式 18}$$

进一步观察附件 3 的水流速数据，发现“测点水速”随起点距离的不同、与

河道中心距离不同、与水面距离不同而发生变化，即水速会受河床地形的影响。由于每个“测速点”在水道断面的坐标位置不同，不能直接代表河道水速。

考虑采用“流速面积法”计算平均水流速度。以各测速点之间的中点为边界，将若干直角梯形进一步划分为若干长方形（小面积），作为该测点的水速的作用区域。示意图 32 如下：

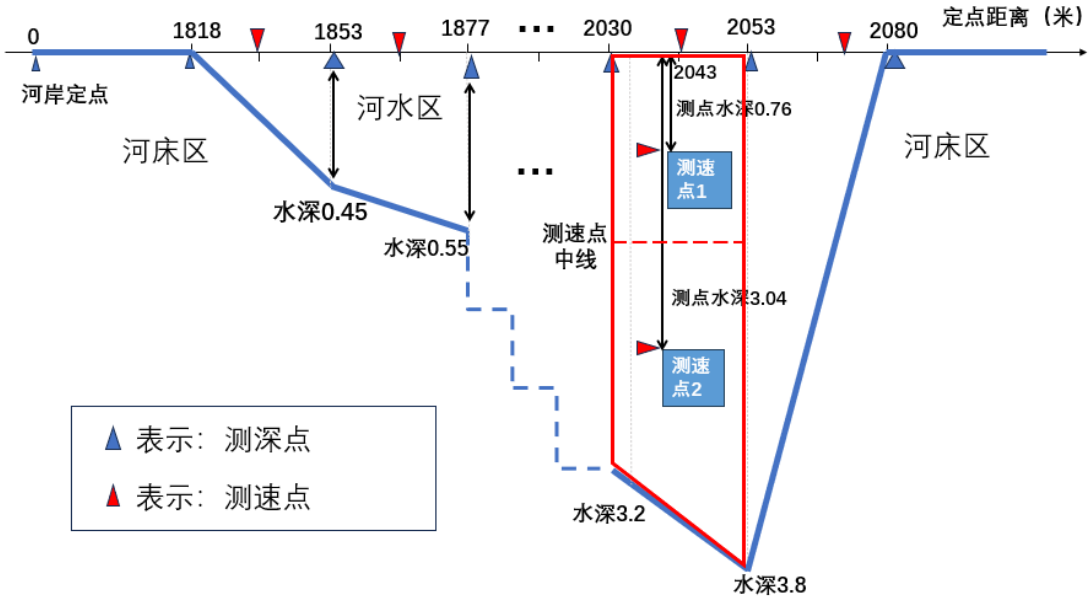

图 32. 求解水道平均速度示意图

记 $m(m>n)$ 次测速中，$Q_i(i=1\sim m)$ 为第 i 次测速，在该次监测中的不同水深处共获得了 k 个水速值和水深值，分别表示为 $v_{ij}(i=1\sim m, j=1\sim k)$、$l_{ij}(i=1\sim m, j=1\sim k)$。定义 b_{ij} 为第 i 次测速中第 j 个小长方形的高度，计算公式为：

$$b_{ij}=\frac{1}{2}(l_{ij}+l_{i,j+1}) \qquad \text{公式 19}$$

第 i 次测速中第 j 个小长方形的宽度即为第 i 次测深中直角梯形的高度，因而该小长方形的面积定义为：

$$S_{ij}=a_{ij}b_{ij} \qquad \text{公式 20}$$

定义 $\bar{v}$ 为水速均值，公式为：

$$\bar{v}=\frac{\sum_{j=1}^{k}\sum_{i=1}^{m}v_{ij}s_{ij}}{S} \qquad \text{公式 21}$$

观察附件 3 发现，每个测深点读数皆为相邻各测速点的水深读数之和，考虑该水文站大概率地采用了“对角拉线法”测水速。依此规律更正了部分数据，使得“测点水深”大于“水深”。

在 Python 下，针对每个监测日的数据，分别计算水道断面总面积、水道平均速度，并在 Excel 下对部分监测日的计算结果进行了检验。部分结果如下表 7：

表 7. 各监测日水道断面面积、平均水速

年 月	2018.4.4	2019．4．17	2020.4.17	2020.7.10	2020.7.11	2021.7.9
面积	572.675	741.6	721.325	1221.6	1240.75	1056.1
水速	0.8172	0.6969	0.8005	1.0119	0.4277	0.973

从以上结果中看出，水道断面具有面积增大的趋势，增强了路经船只的通行能力；同时水速略有上升，为河道的清淤提供了保障。

4. 调水调沙的作用和必要性

水库进行调水调沙是在现代化技术条件下，利用工程设施和调度手段，通过水流的冲击，将水库里的泥沙和河床上的淤沙适时送入大海，从而减少库区和河床的淤积，增大主槽的行洪能力，同时塑造一个稳定的中水河槽，改善滩区的防洪风险。因此，小浪底水库调水调沙的主要是用水转移黄河中的泥沙，冲刷黄河下游的淤泥，防止黄河的河床再次抬升。

如果小浪底水库不进行调水调沙，10 年后不同起点距离的河底高程会逐渐升高，河底高程最低点的增长趋势最明显，势必会造成大面积的灾害，同时影响生态平衡，造成不可挽回的重大损失。

五、模型检验

问题一中，在将含沙量与时间、水位、水流量的关系模型用图、表描述后，用回归分析的方法对各散点图进行了分段曲线拟合，用 Matlab 程序拟合多项式曲线的同时，对各段函数进行了“拟合优度检验”。

经检验，在含沙量与水位的关系的拟合曲线中，根据散点图特点分成两段进行拟合，左边拟合成三次曲线，其 $R^2=0.98$，右边拟合成直线，其 $R^2=0.941$。两段拟合的都非常理想。如果左边用二次曲线去拟合，其 $R^2=0.952$，不如用三次曲线拟合效果好。同样，在含沙量与水流量关系的分段拟合曲线中，左边拟合成

三次曲线,其$R^2=0.924$，右边拟合成直线，其$R^2=0.754$。左侧拟合得比右侧效果更理想。如果左边用二次曲线去拟合，其$R^2=0.907$，仍然不如用三次曲线拟合效果好。两图关系曲线的拟合都通过检验。

随后，又建立了年总水流量和年总排沙量模型，为检验模型估算的数据是否较为准确合理，又选择了更优化的“**数值积分法**”模型同时进行估算，用一种模型的计算数据去检验另一种。结果显示，两种模型的数据比较接近，计算结果准确、合理，通过检验。

问题二、问题三和问题四所建的模型中，在建模求解的过程中都同时包含了检验过程。结果均通过检验。

六、模型评价与推广

6.1 模型的优、缺点评价分析

6.1.1 模型的优点

（1）问题一中含沙量与水位和水流量的关系，画出散点图后都进行了分段拟合，经拟合优度检验，拟合度非常高，说明模型较为准确的反应了含沙量与二者的关系。年总水流量和年总排沙量的估算运用了两种方法分别建模，用一个模型去验证另一个模型，通过验证，说明模型结果准确可靠。

（2）文中建立的“ARIMA 时间序列预测模型”在各类预测方面，具有很好的通用性和推广性。

（3）建立各类水文模型之前，对附件中的数据进行了预处理，补充了缺失数据，校正了个别错误数据，建立了各类数据之间的联系，找出其规律，便于分析解决问题。

（4）文中多个水文模型采用“定量分析与定性分析相结合”,“对已知数据的研究与对未来数据的预测相结合”，使所建模型更有说服力。

（5）本文所有的数据处理均经过精确的分析、比对、效验，具有很强的准确性，真实性；模型求解结果进行了检验，可信度高，可靠性强。

（6）模型求解用到 Phthon、MATLAB、Excel 等多种软件，使求解过程更清晰、专业，运用多种绘图使得数据表达更加清晰明了。

6.1.2 模型的缺点

（1）在对含沙量与水流量的关系图像进行拟合时，分成左、右两端，右端用一次直线拟合，其$R^2=0.754<0.8$，此段拟合效果不够理想。如果将该图像分为三段拟合，可能拟合度会进一步提高。

（2）累积距平法模型在应用时，所求平均值为 6 年来每月平均流量和平均排沙量，对于分析短期突变性的效果不够直观。

6.2 模型的改进与推广

（1）在问题一中，在对含沙量与水流量的图像进行多项式拟合时，分成左、右两端分别拟合，右端用一次直线拟合，通过拟合优度检验得$R^2=0.754$，拟合效果不够理想。如果将该图像分为三段，分别进行拟合，可能拟合度会进一步提高。

（2）问题二在应用累积距平法解决突变性问题时，可以将每年的月均流量和月均排沙量分别作出累积距平曲线，进一步分析每年中发生突变的更具体的时间。

（3）问题二、问题三中建立的 M-K 突变检验模型、小波分析模型、ARIMA 模型近贴近实际，适用于对各领域的问题的分析与预测，有着广泛的通用性和借鉴意义，模型可操作性强，值得推广。

（4）问题四的解决，对政府和相关部门、企业采取一定的应对措施进行防洪减灾，把握问题发展趋势起到了积极的指导作用。

七、参考文献

[1]姜启源，谢金星，叶俊. 数学模型[M]，北京：高等教育出版社，2018.5

[2]韩中庚，周素静. 数学建模实用教程[M]，北京：高等教育出版社，2020.8

[3]司守奎，孙兆亮. 数学建模算法与应用[M]，北京：国防工业出版社，2015.9

[4]卓金武.MATLAB 数学建模方法与实践[M],北京:北京航空航天大学出版社,2011

[5]路贺. 白龙江干流舟曲、武都和碧口水文站水沙特征研究[D].兰州大学,2019

[6]卢晗,王昱,李小宁,周伟. 黑河流域水沙通量多尺度变化特征及影响因素[J],

泥沙研究,2023.3

[7]王光辉,近 60 年黄河干流径流泥沙变异性分析[D]，清华大学,2019

[8]许磊,温雅琴,全栋.黄河干流头道拐水文站水沙关系季节性特征分析[J]，泥沙研究，2023.9

[9]潘彬,黄河水沙变化及其对气候变化和人类活动的响应[D]，山东师范大学,2021.

[10]薛小杰,蒋晓辉,黄强. 小波分析在水文序列趋势分析中的应用[J]，应用科学学报，2002,20

八、附录

一、问题一的程序代码

1. 补充第 t 日的含沙量数据

```
import pandas as pd
def fill_data(list_t):
    for i in range(len(list_t)):
    if list_t[i] == 999:
            for j in range(i, len(list_t)):
                if list_t[j] != 999:
                    t = (list_t[i - 1] + list_t[j]) / 2
                    list_t[i] = t
                    break
    return pd.Series(list_t)
if __name__ == &apos;__main__&apos;:
    f1_path = r&apos;./Data/转置汇总_t.xlsx&apos;
    f1_save = r&apos;./Data/转置汇总_re.xlsx&apos;
    df_ori = pd.read_excel(f1_path)
    df_ori = df_ori.fillna(999)
    df_ori[&apos;2017_t&apos;] = fill_data(df_ori[&apos;2017_t&apos;].values.tolist())
    df_ori[&apos;2018_t&apos;] = fill_data(df_ori[&apos;2018_t&apos;].values.tolist())
    df_ori[&apos;2019_t&apos;] = fill_data(df_ori[&apos;2019_t&apos;].values.tolist())
    df_ori[&apos;2020_t&apos;] = fill_data(df_ori[&apos;2020_t&apos;].values.tolist())
    df_ori[&apos;2021_t&apos;] = fill_data(df_ori[&apos;2021_t&apos;].values.tolist())
    print(df_ori)
    df_ori.to_excel(f1_save, index=False)
```

2. 含沙量与时间关系的画图程序：

```
clc,clear
```

```
clc,clear
data    = xlsread('time-hansha.xlsx', 'sheet1');
time2016=data(:,1);
time2017=data(:,1);
time2018=data(:,1);
time2019=data(:,1);
time2020=data(:,1);
time2021=data(:,1);
sha2016=data(:,2);
sha2017=data(:,3);
sha2018=data(:,4);
sha2019=data(:,5);
sha2020=data(:,6);
sha2021=data(:,7);
sha2016f=sha2016*3600*24;
sha2017f=sha2017*3600*24;
sha2018f=sha2018*3600*24;
sha2019f=sha2019*3600*24;
sha2020f=sha2020*3600*24;
sha2021f=sha2021*3600*24;
subplot(1,2,1),plot(time2016(1:365),sha2016(1:365),'*')
figure
subplot(1,2,1),plot(time2017(1:365),sha2017(1:365),'*')
figure
subplot(1,2,1),plot(time2018(1:365),sha2018(1:365),'*')
figure
subplot(1,2,1),plot(time2019(1:365),sha2019(1:365),'*')
figure
subplot(1,2,1),plot(time2020(1:365),sha2020(1:365),'*')
figure
subplot(1,2,1),plot(time2021(1:365),sha2021(1:365),'*')
```

3. 含沙量与水位的分段拟合及检验程序

```
clc,clear
data    = xlsread('nihejisuan-shuiwei.xlsx', 'sheet1');
wei=data(:,1);
sha2016=data(:,2);
format long e
    nihe1 = polyfit(wei(1:7),sha2016(1:7),2);
    x=wei(1:7);
    y=sha2016(1:7);
    f= polyval(nihe1,x);
    mdl1=fitlm(y,f)
```

```
%R1=1-(sum(f-y).^2/sum((y-mean(y)).^2))
plot(x, f)
hold on
plot(x,y,'*')
xlim([42 46.5]);
ylim([0 18]);

nihe2 = polyfit(wei(8:13),sha2016(8:13),1);
x2=wei(8:13);
f2=sha2016(8:13);
y2 = polyval(nihe2,wei(8:13));
mdl2=fitlm(y2,f2)
rmse2 = sqrt(sum((sha2016(8:13)-y2).^2));

%R2=1-(sum( y2-sha2016(7:13)).^2/sum((sha2016(7:13)-mean(sha2016(7:13))).^2))
plot(wei(8:13), y2)
hold on
plot(wei(8:13),sha2016(8:13),'.')
rmse2;
hold off
```

4. 含沙量与水流量关系的分段曲线拟合及检验程序：

```
clc,clear
data    = xlsread('nihejisuan-shuiliu.xlsx', 'sheet1');
liu=data(:,1);
sha=data(:,2);
plot(liu,sha)
hold off
format long e
nihe1 = polyfit(liu(1:7),sha(1:7),3)
x=liu(1:7);
y=sha(1:7);
yhat1= polyval(nihe1,x);
mdl1=fitlm(y,yhat1)
cha1=sum((sha(1:7)-yhat1.^2));rmse1=sqrt(cha1);
plot(liu(1:7), yhat1)
hold on
plot(liu(1:7),sha(1:7),'*')
rmse1;
nihe2 = polyfit(liu(7:13),sha(7:13),1);
x2=liu(7:13);
y2=sha(7:13);
yhat2 = polyval(nihe2,liu(7:13));
mdl2=fitlm(y2,yhat2)
```

```
rmse2 = sqrt(sum((sha(7:13)-yhat2).^2));
plot(liu(7:13), yhat2)
hold on
plot(liu(7:13),sha(7:13),'.')
rmse2;
hold off
```

5. “数值积分法”年总水流量程序：

```
clc,clear
data    = xlsread('time-shuiliu.xlsx', 'sheet1');
time2016=data(:,1);
time2017=data(:,1);
time2018=data(:,1);
time2019=data(:,1);
time2020=data(:,1);
time2021=data(:,1);
liu2016=data(:,2);
liu2017=data(:,3);
liu2018=data(:,4);
liu2019=data(:,5);
liu2020=data(:,6);
liu2021=data(:,7);
liu2016f=liu2016*3600*24;
liu2017f=liu2017*3600*24;
liu2018f=liu2018*3600*24;
liu2019f=liu2019*3600*24;
liu2020f=liu2020*3600*24;
liu2021f=liu2021*3600*24;
i=1:365;
t=i;
t1=t(1);t2=t(end);
pp2016=csape(t,liu2016f);
pp2017=csape(t,liu2017f);
pp2018=csape(t,liu2018f);
pp2019=csape(t,liu2019f);
pp2020=csape(t,liu2020f);
pp2021=csape(t,liu2021f);
xsh2016=pp2016.coefs
xsh2017=pp2017.coefs
xsh2018=pp2018.coefs
xsh2019=pp2019.coefs
xsh2020=pp2020.coefs
xsh2021=pp2021.coefs
TL2016=quadl(@ (tt)fnval(pp2016,tt),t1,t2)
```

```
TL2017=quadl(@ (tt)fnval(pp2017,tt),t1,t2)
TL2018=quadl(@ (tt)fnval(pp2018,tt),t1,t2)
TL2019=quadl(@ (tt)fnval(pp2019,tt),t1,t2)
TL2020=quadl(@ (tt)fnval(pp2020,tt),t1,t2)
TL2021=quadl(@ (tt)fnval(pp2021,tt),t1,t2)
a = [TL2016,TL2017,TL2018,TL2019,TL2020,TL2021]
```

6. “数值积分法”年总排沙量程序：

```
clc,clear
data    = xlsread('time-hansha.xlsx', 'sheet1');
time2016=data(:,1);
time2017=data(:,1);
time2018=data(:,1);
time2019=data(:,1);
time2020=data(:,1);
time2021=data(:,1);
sha2016=data(:,2);
sha2017=data(:,3);
sha2018=data(:,4);
sha2019=data(:,5);
sha2020=data(:,6);
sha2021=data(:,7);
sha2016f=sha2016/1000;
sha2017f=sha2017/1000;
sha2018f=sha2018/1000;
sha2019f=sha2019/1000;
sha2020f=sha2020/1000;
sha2021f=sha2021/1000;
i=1:365;
t=i;
t1=t(1);t2=t(end);
pp2016=csape(t,sha2016f);
pp2017=csape(t,sha2017f);
pp2018=csape(t,sha2018f);
pp2019=csape(t,sha2019f);
pp2020=csape(t,sha2020f);
pp2021=csape(t,sha2021f);
xsh2016=pp2016.coefs
xsh2017=pp2017.coefs
xsh2018=pp2018.coefs
xsh2019=pp2019.coefs
xsh2020=pp2020.coefs
xsh2021=pp2021.coefs
TL2016=quadl(@ (tt)fnval(pp2016,tt),t1,t2)
```

```
TL2017=quadl(@ (tt)fnval(pp2017,tt),t1,t2)
TL2018=quadl(@ (tt)fnval(pp2018,tt),t1,t2)
TL2019=quadl(@ (tt)fnval(pp2019,tt),t1,t2)
TL2020=quadl(@ (tt)fnval(pp2020,tt),t1,t2)
TL2021=quadl(@ (tt)fnval(pp2021,tt),t1,t2)
a = [TL2016,TL2017,TL2018,TL2019,TL2020,TL2021]
```

二、问题二的程序代码

1. 月流量累积距平曲线代码

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# 读取数据
data = pd.read_excel(r'./Data/月输沙量_1.xlsx')
# 去除重复数据和缺失值
data.drop_duplicates(inplace=True)
data.dropna(inplace=True)
# 将日期转换为时间格式
data['时间'] = pd.to_datetime(data['时间'])

# 计算每日距平流量
daily_mean = data.groupby(data['时间'].dt.dayofyear)['流量'].mean()
data['flow_anomaly'] = data['流量'] - daily_mean[data['时间'].dt.dayofyear].values

# 计算累积距平流量
data['flow_cumsum'] = data['flow_anomaly'].cumsum()

print(data['时间'].values)
print(data['flow_cumsum'])

# 绘制累积距平曲线
plt.figure(figsize=(12,8), dpi=120)
plt.plot(data['时间'].values, data['flow_cumsum'].values.tolist(), color='black')
plt.xlabel('Date')
plt.ylabel('Cumulative Anomaly')
plt.title('Cumulative Anomaly Curve')
plt.show()
```

2. 月排沙量累积距平曲线代码

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
# 读取数据
data = pd.read_excel(r'./Data/月输沙量_1.xlsx')
# 去除重复数据和缺失值
data.drop_duplicates(inplace=True)
data.dropna(inplace=True)
# 将日期转换为时间格式
data['时间'] = pd.to_datetime(data['时间'])

# 计算每日距平输沙量
daily_mean = data.groupby(data['时间'].dt.dayofyear)['输沙量'].mean()
data['flow_anomaly'] = data['输沙量'] - daily_mean[data['时间'].dt.dayofyear].values

# 计算累积距平输沙量
data['flow_cumsum'] = data['flow_anomaly'].cumsum()

print(data['时间'].values)
print(data['flow_cumsum'])

# 绘制累积距平曲线
plt.figure(figsize=(12,8), dpi=120)
plt.plot(data['时间'].values, data['flow_cumsum'].values.tolist(), color='black')
plt.xlabel('Date')
plt.ylabel('Cumulative Anomaly')
plt.title('Cumulative Anomaly Curve')
plt.show()
```

3. 月流量 M-K 突变检验代码

```
import numpy as np
import pandas as pd
from pylab import *

df_ori = pd.read_excel(r'./Data/月流量_1.xlsx')
# df_ori_time = pd.to_datetime(df_ori['时间'].values.tolist())
# df = pd.DataFrame(df_ori['均值'].values.tolist(), index=pd.date_range('2016-01-01',
#
periods=72, freq='M'), columns=['均值'])
df_t = df_ori['流量']
np_t = np.array(df_t.values.tolist())
Data = np_t
# 对时序数据 X ，生成一个序号序列，次数列范围为 2～n。
# i 从 2 开始（即第二个数，其编号为 1，此时 1 处没有必要进行计算，因为
其之前没有数据，所以这里从 2 开始生成）
# 规定第一个结果为 0，因此我们不考虑第一个位置的结果
NUMI = np.arange(1, len(Data))
```

```
# 计算 E
# E = NUMI * (NUMI - 1) / 4
E = (NUMI + 1) * NUMI / 4
# 计算 Var
# VAR = NUMI * (NUMI - 1) * (2 * NUMI + 5) / 72
VAR = (NUMI + 1)  * NUMI * (2 * (NUMI + 1) + 5) / 72
# 1.计算 Ri。即：序列中的某一个值与此值之前的所有值以此相比，结果为大出现的次数。
Ri = [(Data[i] > Data[:i]).sum() for i in NUMI]# 2.计算 Sk。使用 numpy 累计求和函数 cumsum。
Sk = np.cumsum(Ri)# 3.计算 UFk。考虑到 i 从 1 开始，因此把未计算的两个位置填充 0 。
UFk = np.pad((Sk - E) / np.sqrt(VAR), (1,0))
# 思路参考第一步，这里进行简写。
## 对于倒序，由于 Python 支持传入负数表示倒序取值，这里利用此特性直接生成倒序（反向） Bk，不包含最后一个数（编号 -1）。
Bk = np.cumsum([(Data[i] > Data[i:]).sum() for i in -(NUMI+1)])
## 按照 UFk 的计算方法后取负数即为 UBk。由于本身未对 Data 进行倒序，这里计算完成后对数据进行倒序。
UBk = np.pad((-(Bk - E) / np.sqrt(VAR)), (1,0))[::-1]
import matplotlib.pyplot as plt
# 配置参数
PAR = {'font.sans-serif': 'Times New Roman','axes.unicode_minus': False}
plt.rcParams.update(PAR)
plt.figure(figsize = (10, 5.5), dpi = 300)
mpl.rcParams['font.sans-serif'] = ['SimHei']
plt.title('2016-2021 年流量 M-K 突变检验曲线')
plt.plot(range(1 ,len(Data)+1),UFk,label = 'UFk',color = 'r')
plt.plot(range(1 ,len(Data)+1),UBk,label = 'UBk',color = 'black')
# plt.grid(True, linestyle = (0,(6,6)), linewidth = 0.4)## 画出 0.05 置信区间边界
x_lim = plt.xlim()
plt.plot(x_lim,[-1.96,-1.96],linestyle = (0,(6,6)),color = 'g')
plt.plot(x_lim, [0,0],linestyle = (0,(6,6)),color = 'g')
plt.plot(x_lim,[1.96,1.96],linestyle = (0,(6,6)),color = 'g')
plt.legend(frameon = False)
plt.show()
```

4. 月排沙量 M-K 突变检验代码

```
import numpy as np
import pandas as pd
from pylab import *

df_ori = pd.read_excel(r'./Data/月输沙量_1.xlsx')
# df_ori_time = pd.to_datetime(df_ori['时间'].values.tolist())
```

```
# df = pd.DataFrame(df_ori['均值'].values.tolist(), index=pd.date_range('2016-01-01',
#
periods=72, freq='M'), columns=['均值'])
df_t = df_ori['输沙量']
np_t = np.array(df_t.values.tolist())
Data = np_t
# 对时序数据 X ，生成一个序号序列，次数列范围为 2～n。
# i 从 2 开始（即第二个数，其编号为 1，此时 1 处没有必要进行计算，因为其之前没有数据，所以这里从 2 开始生成）
# 规定第一个结果为 0，因此我们不考虑第一个位置的结果
NUMI = np.arange(1, len(Data))
# 计算 E
# E = NUMI * (NUMI - 1) / 4
E = (NUMI + 1) * NUMI / 4
# 计算 Var
# VAR = NUMI * (NUMI - 1) * (2 * NUMI + 5) / 72
VAR = (NUMI + 1)  * NUMI * (2 * (NUMI + 1) + 5) / 72
# 1.计算 Ri。即：序列中的某一个值与此值之前的所有值以此相比，结果为大出现的次数。
Ri = [(Data[i] > Data[:i]).sum() for i in NUMI]# 2.计算 Sk。使用 numpy 累计求和函数 cumsum。
Sk = np.cumsum(Ri)# 3.计算 UFk。考虑到 i 从 1 开始，因此把未计算的两个位置填充 0 。
UFk = np.pad((Sk - E) / np.sqrt(VAR), (1,0))
# 思路参考第一步，这里进行简写。
## 对于倒序，由于 Python 支持传入负数表示倒序取值，这里利用此特性直接生成倒序（反向） Bk，不包含最后一个数（编号 -1）。
Bk = np.cumsum([(Data[i] > Data[i:]).sum() for i in -(NUMI+1)])
## 按照 UFk 的计算方法后取负数即为 UBk。由于本身未对 Data 进行倒序，这里计算完成后对数据进行倒序。
UBk = np.pad((-(Bk - E) / np.sqrt(VAR)), (1,0))[::-1]
import matplotlib.pyplot as plt
# 配置参数
PAR = {'font.sans-serif': 'Times New Roman','axes.unicode_minus': False}
plt.rcParams.update(PAR)
plt.figure(figsize = (10, 5.5), dpi = 300)
mpl.rcParams['font.sans-serif'] = ['SimHei']
plt.title('2016-2021 年输沙量 M-K 突变检验曲线')
plt.plot(range(1 ,len(Data)+1),UFk,label = 'UFk',color = 'r')
plt.plot(range(1 ,len(Data)+1),UBk,label = 'UBk',color = 'black')
# plt.grid(True, linestyle = (0,(6,6)), linewidth = 0.4)## 画出 0.05 置信区间边界
x_lim = plt.xlim()
plt.plot(x_lim,[-1.96,-1.96],linestyle = (0,(6,6)),color = 'g')
plt.plot(x_lim, [0,0],linestyle = (0,(6,6)),color = 'g')
```

```
plt.plot(x_lim,[1.96,1.96],linestyle = (0,(6,6)),color = 'g')
plt.legend(frameon = False)
plt.show()
```

5. 流量季节性分析

```
'''
季节分解法
'''
import pandas as pd
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

def cal_avg(df_ori):
    '''
    计算每月均值
    '''
    list_ori = df_ori.values.tolist()
    print(str(list_ori[0][0]).split(' ')[0])
    list_time = []
    for i in range(len(list_ori)):
        list_time.append(str(list_ori[i][0]).split(' ')[0])

    list_re = []
    sum_num = 0
    list_t = [list_time[0]]
    flag = list_time[0].split('-')[1]
    count_num = 0
    for i in range(len(list_ori)):
        if list_time[i].split('-')[1] == flag:
            count_num = count_num + 1
            sum_num = sum_num + list_ori[i][1]
        else:
            avg_num = sum_num / count_num
            list_t.append(avg_num)
            list_re.append(list_t)
            sum_num = list_ori[i][1]
            flag = list_time[i].split('-')[1]
            list_t = [list_time[i]]
            count_num = 1
    list_t.append(sum_num/count_num)
    list_re.append(list_t)
    df_avg = pd.DataFrame(list_re)
    df_avg.to_excel(r'./Data/时间流量_month.xlsx')
```

```
if __name__ == '__main__':
    df_ori = pd.read_excel(r'./Data/月流量_1.xlsx')
    df_ori_time = pd.to_datetime(df_ori['时间'].values.tolist())
    df = pd.DataFrame(df_ori['流量'].values.tolist(), index=pd.date_range('2016-01-01',
                        periods=72, freq='M'), columns=['流量'])
    print(df)
    # cal_avg(df_ori)
    result_mul = seasonal_decompose(df['流量'],
                model='multiplicative',
                extrapolate_trend='freq')

    plt.rcParams.update({'figure.figsize': (10, 10)})
    result_mul.plot().suptitle('Multiplicative Decompose')
    plt.show()
```

6. 月排沙量季节性分析

```
'''
季节分解法
'''
import pandas as pd
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

def cal_avg(df_ori):
    '''
    计算每月均值
    '''
    list_ori = df_ori.values.tolist()
    print(str(list_ori[0][0]).split(' ')[0])
    list_time = []
    for i in range(len(list_ori)):
        list_time.append(str(list_ori[i][0]).split(' ')[0])

    list_re = []
    sum_num = 0
    list_t = [list_time[0]]
    flag = list_time[0].split('-')[1]
    count_num = 0
    for i in range(len(list_ori)):
        if list_time[i].split('-')[1] == flag:
            count_num = count_num + 1
            sum_num = sum_num + list_ori[i][1]
        else:
            avg_num = sum_num / count_num
```

```
                list_t.append(avg_num)
                list_re.append(list_t)
                sum_num = list_ori[i][1]
                flag = list_time[i].split('-')[1]
                list_t = [list_time[i]]
                count_num = 1
        list_t.append(sum_num/count_num)
        list_re.append(list_t)
        df_avg = pd.DataFrame(list_re)
        df_avg.to_excel(r'./Data/时间流量_month.xlsx')

if __name__ == '__main__':
        df_ori = pd.read_excel(r'./Data/月输沙量_1.xlsx')
        df_ori_time = pd.to_datetime(df_ori['时间'].values.tolist())
        df = pd.DataFrame(df_ori['输沙量'].values.tolist(), index=pd.date_range('2016-
01-01',
                                periods=72, freq='M'), columns=['输沙量'])
        print(df)
        # cal_avg(df_ori)
        result_mul = seasonal_decompose(df['输沙量'],
                        model='multiplicative',
                        extrapolate_trend='freq')

        plt.rcParams.update({'figure.figsize': (10, 10)})
        result_mul.plot().suptitle('Multiplicative Decompose')
        plt.show()
```

三、问题三的程序代码

SRAIMA 模型的建立与预测

```
import warnings                                  # do not disturbe mode
warnings.filterwarnings('ignore')

# Load packages
import numpy as np                               # vectors and matrices
import pandas as pd                              # tables and data manipulations
import matplotlib.pyplot as plt                  # plots
import seaborn as sns                            # more plots

from dateutil.relativedelta import relativedelta # working with dates with style
from scipy.optimize import minimize              # for function minimization

import statsmodels.formula.api as smf            # statistics and econometrics
```

```
import statsmodels.tsa.api as smt
import statsmodels.api as sm
import scipy.stats as scs

from itertools import product                    # some useful functions
from tqdm import tqdm_notebook

# Importing everything from forecasting quality metrics
from sklearn.metrics import r2_score, median_absolute_error, mean_absolute_error
from sklearn.metrics import median_absolute_error, mean_squared_error,
mean_squared_log_error

# MAPE
def mean_absolute_percentage_error(y_true, y_pred):
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

def tsplot(y, lags=None, figsize=(12, 7), style='bmh'):
    """
        Plot time series, its ACF and PACF, calculate Dickey–Fuller test

        y - timeseries
        lags - how many lags to include in ACF, PACF calculation
    """

    if not isinstance(y, pd.Series):
        y = pd.Series(y)

    with plt.style.context(style):
        fig = plt.figure(figsize=figsize)
        layout = (2, 2)
        ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
        acf_ax = plt.subplot2grid(layout, (1, 0))
        pacf_ax = plt.subplot2grid(layout, (1, 1))

        y.plot(ax=ts_ax, color='black')
        p_value = sm.tsa.stattools.adfuller(y)[1]
        ts_ax.set_title('Time Series Analysis Plots\n Dickey-Fuller:
p={0:.5f}'.format(p_value))
        smt.graphics.plot_acf(y, lags=lags, ax=acf_ax)
        smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax)
        plt.tight_layout()
        plt.show()


index = pd.date_range('2016-01', periods=72, freq='M')
```

```
index = list(index)
df = pd.read_excel(r'./Data/时间流量_month.xlsx')
data_list = df['均值'].values.tolist()
dataframe = pd.DataFrame(data_list, index=pd.date_range('2016-01', periods=72,
freq='M'), columns=['Ads'])
# dataframe.to_csv(r'./Data/ARMA_test_old.csv',index=0)


ads = dataframe
print('the data is existting')
plt.figure(figsize=(18, 6))
plt.plot(ads, color='black')
plt.title('Monthly average flow')
plt.show()

ads_diff = ads.Ads - ads.Ads.shift(3) #季节差分
tsplot(ads_diff[3:], lags=25)
ads_diff = ads_diff - ads_diff.shift(1) #一阶差分
ads_diff = ads_diff - ads_diff.shift(1) #二阶差分
ads_diff = ads_diff.fillna(0)
tsplot(ads_diff[3:], lags=25)

# setting initial values and some bounds for them
ps = range(0, 4)
d=2 #除去季节差分外的差分次数
qs = range(0, 4)
Ps = range(0, 4)
D=1
Qs = range(0, 4)
s = 6 # season length is still 24

# creating list with all the possible combinations of parameters
parameters = product(ps, qs, Ps, Qs)
parameters_list = list(parameters)
len(parameters_list)   # 36

def optimizeSARIMA(parameters_list, d, D, s):
    """Return dataframe with parameters and corresponding AIC

        parameters_list - list with (p, q, P, Q) tuples
        d - integration order in ARIMA model
        D - seasonal integration order
        s - length of season
    """
```

```
    results = []
    best_aic = float("inf")

    for param in tqdm_notebook(parameters_list):
        # we need try-except because on some combinations model fails to converge
        try:
            model=sm.tsa.statespace.SARIMAX(ads.Ads, order=(param[0], d,
param[1]),
                                            seasonal_order=(param[2], D, param[3],
s)).fit(disp=-1)
        except:
            continue
        aic = model.aic
        # saving best model, AIC and parameters
        if aic < best_aic:
            best_model = model
            best_aic = aic
            best_param = param
        results.append([param, model.aic])

    result_table = pd.DataFrame(results)
    result_table.columns = ['parameters', 'aic']
    # sorting in ascending order, the lower AIC is - the better
    result_table = result_table.sort_values(by='aic',
ascending=True).reset_index(drop=True)

    return result_table

warnings.filterwarnings("ignore")
result_table = optimizeSARIMA(parameters_list, d, D, s)
print(result_table.parameters[0])

# set the parameters that give the lowest AIC
p, q, P, Q = result_table.parameters[0]

best_model=sm.tsa.statespace.SARIMAX(ads_diff, order=(p, d, q),
                                     seasonal_order=(P, D, Q, s)).fit(disp=-1)

tsplot(best_model.resid[3+1:], lags=25)

def plotSARIMA(series, model, n_steps):
    """Plots model vs predicted values

        series - dataset with timeseries
```

```
        model - fitted SARIMA model
        n_steps - number of steps to predict in the future
    """

    # adding model values
    data = series.copy()
    data.columns = ['actual']
    data['sarima_model'] = model.fittedvalues
    # making a shift on s+d steps, because these values were unobserved by the model
    # due to the differentiating
    data['sarima_model'][:s+d] = np.NaN

    # forecasting on n_steps forward
    forecast = model.predict(start = data.shape[0], end = data.shape[0]+n_steps)
    forecast = data.sarima_model.append(forecast)

    # calculate error, again having shifted on s+d steps from the beginning
    error = mean_absolute_percentage_error(data['actual'][s+d:],
data['sarima_model'][s+d:])

    plt.figure(figsize=(15, 7))
    plt.title("Mean Absolute Percentage Error: {0:.2f}%".format(error))
    plt.plot(forecast, color='r', label="model")
    plt.axvspan(data.index[-1], forecast.index[-1], alpha=0.5, color='lightgrey')
    plt.plot(data.actual, label="actual")
    plt.legend()
    plt.grid(True)
    plt.show()

plotSARIMA(ads, best_model, 12)
```