基于最小二乘法分段拟合的黄河水沙监测数据分析

摘要

对黄河进行研究时，通过研究水中含沙量分析含沙量的各种特性和变化规律，从而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。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与时间模型和含沙量关系模型，利用时间序列预测模型预测水中含沙量的变化趋势，优化水文站采样监测方案。本文主要借助 Python、Excel、Matlab 等软件进行相关计算。

问题一：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-时间关系和含沙量-水位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

问题二：在对每月水流量和沙通量数据进行整理和分析后，我们得到以下结论：描述性分析显示，水流量各月份之间存在较大差异，但其波动幅度平均来看较为稳定。相比之下，沙通量的波动幅度则更为显著。季节性分析结果表明，水通量和沙通量通常在 7 月份达到年度最高点，而在 1 月份降至最低。同时，每年的 6 月至 10 月是水沙通量的高季，而 1 月至 3 月以及 11 月和 12 月则是低季。周期性分析发现，水沙通量呈现出以一年（365 天）为周期的循环模式，并在 2018 年出现了异常变化。应用 pettitt 突变法进行突变性分析，结果显示从第 30 个月开始，水沙通量呈现显著的上升态势，并在第 28 个月观测到一个突变点。Fisher 最优分割法的验证结果与 pettitt 突变法相一致。综上所述，通过对月度时间序列数据的综合分析，我们得出了水沙通量的季节性、周期性和突变性特征，这些特征对于理解和预测水沙通量的变化趋势具有重要意义。

问题三的核心在于使用时间序列预测模型来预测未来两年的水沙通量变化，并据此设计高效的采样监测方案。首先对数据进行季节性分解和平稳性检验，然后拟合 SARIMA 模型并进行参数优化。通过模型验证和参数优化提高预测精度，并利用优化后的模型进行预测。最后得出监测方案为：根据预测数据，我们制定了以下监测水沙通量的方案：秋季期间，稳定后每 3 天一次；夏季每天早晚各一次，增加至一天三次；春季开始每日监测，5 月下旬起每日两次；冬季从 10 月底起提升至每日一次。这些措施有助于有效监测和分析水沙通量的变化

问题四调水调沙措施有效稳定了河底高程，防止其上升。2019 年效果显著，河底高程下降 0.23 米。自 2018 年起实施后，尤其在 6-7 月水沙通量增加，符合预期。不调水调沙，河底高程预测 2021 年会升至 46.053 米。这些成果表明调水调沙工程在控制河床侵蚀和淤积方面发挥了重要作用[^3^][^5^]。

关键词： pettitt 突变法、最小二乘法分段拟合、Fisher 最优分割法

一、问题重述

黄河被称作为中华民族的母亲河，在中国的经济、社会和生态系统中扮演着至关重要的角色。通过探求黄河水沙通量的变化规律，会对沿黄流域的环境治理、气候变化和人民生活有着深远的影响，从而对于优化黄河流域水资源分配、协调人地关系、调水调沙、防洪减灾等方面提供重要的理论指导意义。

利用附件中的数据，以此来建立数学模型，去解决以下的问题：

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：基于该水文站水沙通量的历史数据，进行未来两年的预测分析，以确定水沙通量的变化趋势。同时，制定一个经济高效的采样监测计划，包括监测次数和时间点，以确保能够及时捕捉水沙通量的动态变化，同时最小化监测成本。

问题 4：分析小浪底水文站在每年 6 月至 7 月期间进行的"调水调沙"措施的实际效果，以及这些措施对河底高程的影响。探讨如果不实施"调水调沙"，该水文站河底高程在未来十年可能发生的变化。

二、问题分析

针对问题 1，将问题 1 分为两个部分，第一部分是将小时级数据处理为日线数据，再建立曲线拟合模型来分析黄河水的含沙量与时间、水位、水流量之间的关系。利用 MATLAB 数学建模软件进行数据分析，建立多元线性回归模型，以含沙量为因变量，时间、水位和水流量为自变量，进行参数估计和显著性检验。第二部分是根据回归方程估算近 6 年该水文站的年总水流量和年总排沙量。

针对问题 2，可以利用时间序列分析方法对水沙通量数据进行分析。首先，对近 6 年的水沙通量数据进行季节性分解，利用季节分解法或移动平均法提取出季节性变动成分和趋势成分。然后，通过检验残差序列的自相关性和偏自相关性，判断是否存在周期性和突变性。最后，结合统计图表和数学模型，分析水沙通量的变化规律，确定其突变、季节性和周期性特征。

针对问题 3，可以利用时间序列预测模型对未来两年水沙通量进行预测。首先，利用已有数据建立合适的时间序列模型，如自回归移动平均模型（ARIMA）或季节性自回归移动平均模型（SARIMA）。然后，对模型进行参数估计和模型诊断，确定最优的预测模型。最后，利用所建立的预测模型对未来两年水沙通量进行预测，并制定最优的采样监测方案，考虑监测次数和具体时间等因素，以满足及时掌握水沙通量动态变化情况的需求，同时最大程度地减少监测成本资源。

针对问题四，为了评估“调水调沙”措施的效果，我们可以通过比较实施前后河床高程、水沙通量和流量等关键指标的变化来进行。通过使用微元法来近似计算平均河底高程，我们可以更准确地了解河流冲刷和淤积的情况。这种方法考虑了从河床到水面的距离以及河流横截面积的变化，从而提供了对河流形态变化更深刻的理解。

三、 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。

四、符号说明

符号	符号含义
S	含沙量
t	时间
Z	组数
N	水位值总数
$P_{水}$	水流量
$P_{沙}$	排沙量
$Q_{年排沙}$	每天早晨 8 点时的排沙量
Q_r	第 r 天含沙量数据相关联的流量
y_r^2	一年 t（日）的水流量
k	该年的总水流量（亿m^3）
$X_{水通}$	水通量
$X_{沙通}$	沙通量

五、模型的建立与求解

5.1：问题一：研究含沙量与时间的关系

5.1.1：研究含沙量与年、月的关系

取得 2016 到 2021 年的含沙量年平均值，可画出含沙量与年份关系图，如图（1）可知，含沙量在 2016 与 2017 的平均含沙量远低于 2018 到 2021 年的平均含沙量。

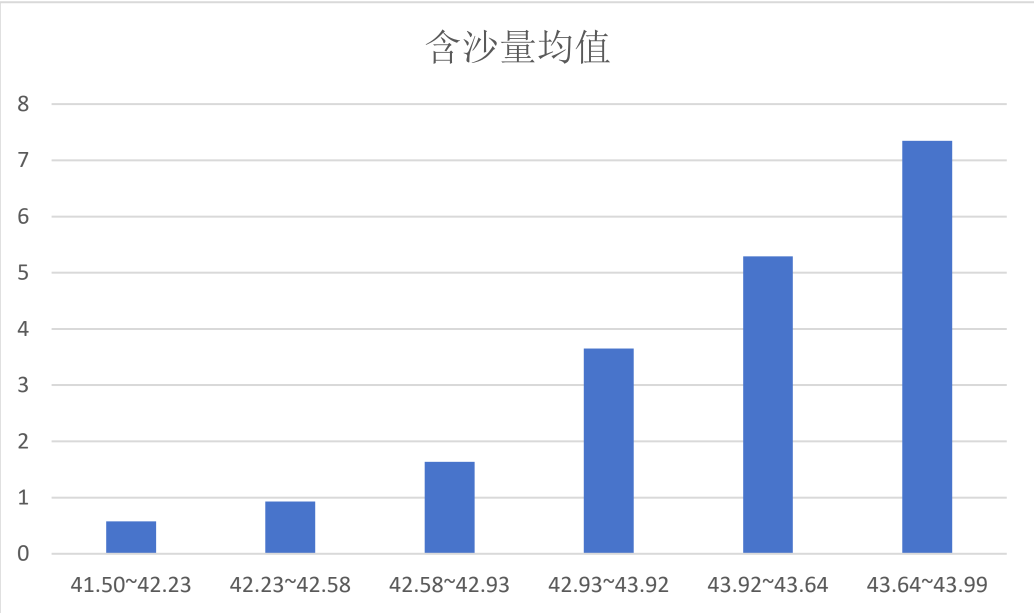


图 1：含沙量与年份关系图

在分别取 2016 年和 2017 年的含沙量与时间的关系图和 2018 年到 2021 年的含沙量与时间的关系图如图（2）、图（3）：

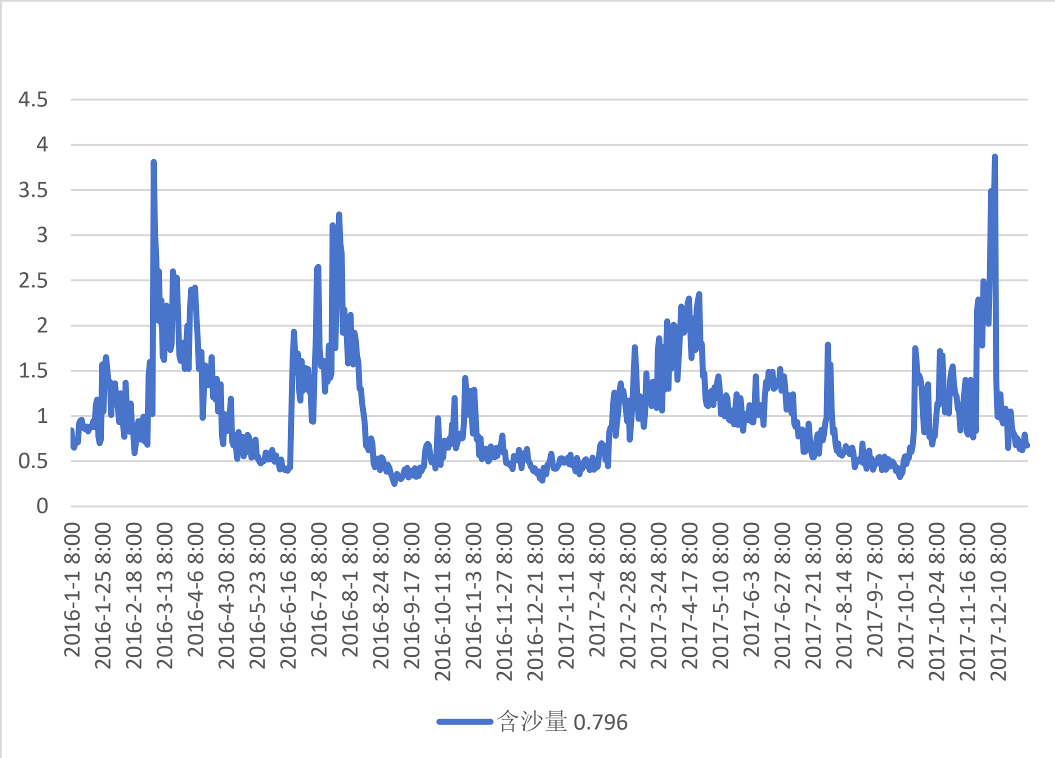


图 2：2016 年和 2017 年的含沙量与时间的关系

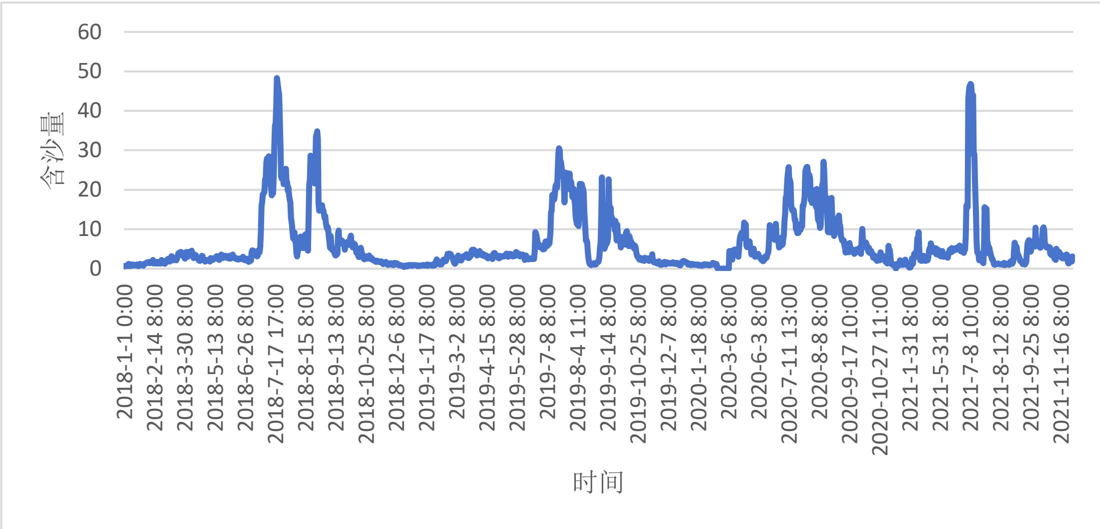


图 3：2018 年到 2021 年的含沙量与时间的关系图

由上图可知以下结论：

（1）：含沙量在 2016，2017 年较少，但从 2018 年开始大幅度增加。

（2）：含沙量在 2016，2017 年和 2018 到 2021 年，分别存在不同的周期性波动。

（3）：2016，2017 年在整体上，3 月、4 月、7 月、8 月为旺季，1 月、2 月、5 月、6 月为淡季。

（4）2018 到 2019 在整体上，7 月，8 月，9 月为旺季，1 月、2 月、3 月、4 月为淡季。

5.1.2：研究含沙量与时间点的关系

当我们设 S 表示含沙量（），t 表示时间点，则可得 T 与 S 的函数关系为：

$$S = A\sin(\omega t+\phi)+C \tag{1}$$

S 的周期为一年，可知 S 的周期为 365 天，所以$T=365=\frac{2\pi}{\omega},\omega=\frac{2\pi}{365}$，由此可得：

$$S = A\sin\left(\frac{2\pi}{365}t+\phi\right)+C \tag{2}$$

在使用最小二乘法进行分段拟合可得：

$$y=\begin{cases} 0.30\sin\left(\frac{2\pi}{365}t-0.27\right)+1.02, & t\in[0,730) \\ \\ 3.78\sin\left(\frac{2\pi}{365}t-1.93\right)+4.49, & t\in[731,2190] \end{cases} \tag{3}$$

5. 2. 问题一：研究含沙量与水位的关系

5.2.1 数据预处理

对附件一数据经过预处理，筛选删除缺失含沙量的行，保留下 2154 行数据，研究其中“含沙量”和“水位”中的数据。

5.2.2 对组距分组

由于“水位”为较多的连续的变量，因此，将水位值的变化范围分为几个不同的区间，而每个水位值都有相应的区间，由此进行组距分组。

分组采用了统计学家 Sturges 提出的经验公式来确定组数，其公式为：

$$Z\approx 1+\frac{\ln N}{\ln 2} \tag{4}$$

上式中，Z 代表组数，N 为水位值的总数，其中 N=2154，Z=12.0728，对分组取奇数，分得为 13 组。

而后，对各组数据进行分组，取得各组上限和下限。在模糊寻找各个小于水位值的每组下先，最后将 2154 个数值分类归为 13 组数据，获得第 m 组内各含沙量的数据：

$$Q_m = (q_{m1}' \quad q_{m2}' \quad \dots \quad q_{mn}') \tag{5}$$

公式中，n 表示第 1~13 组的数据频数。

在求取各组均值：

$$\overline{q_k}' = \frac{1}{n}\sum_{j=1}^{n} q_{kj} \tag{6}$$

可得结果如表 1：

表 1 各组水位的含沙量均值

水位分组	含沙量均值
41.50~42.23	0.576724638
42.23~42.58	0.932594595
42.58~42.93	1.637192727
42.93~43.92	3.649313889
43.92~43.64	5.293121387
43.64~43.99	7.347054264
43.99~44.34	16.66772277
44.34~44.69	16.08071429
44.69~45.04	14.67
45.04~45.40	12.82211268
45.40~45.75	10.68325
45.75~46.10	6.438888889
46.10~47.00	6.6568

可得其趋势变化图为：

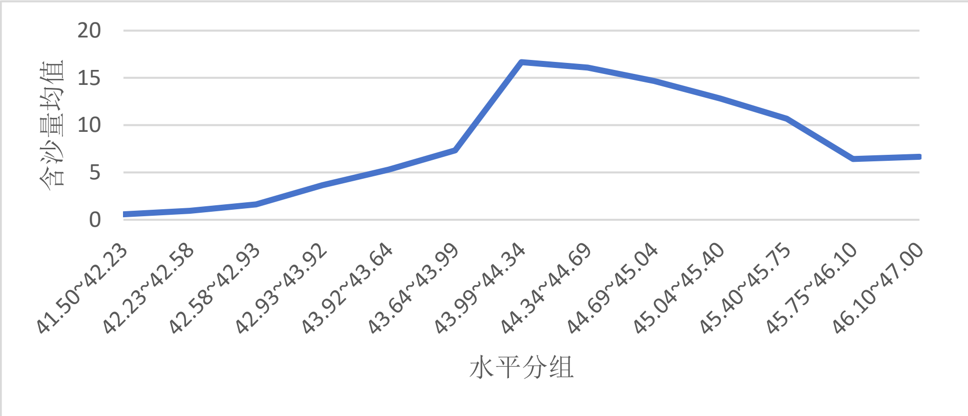


图 4：含沙量与水位关系图

5.2.3 拟合“含沙量-水位”关系曲线

观察上图可知，数据可以大致分为上升与下降两段，由于前七组的数据是含沙量随着水位上升而上升，且在 43.99-44.34 段上升最为明显，因此可将前七组归类为上升段，其余归类为下降段。由此可知该函数更符合分段函数特征。由此将函数切分两段，利用线性最小二乘法进行回归分析，同时借助 Excel 趋势线功能，进行拟合与推理　可得公式：

$$y=\begin{cases}0.0856x^4-1.2026x^3+5.9978x^2-11.033x+6.8345, & 42.08\le x<44.34,\\ 0.1338x^4-1.6965x^3+7.047x^2-12.929x+23.567, & 44.34\le x\le 46.24.\end{cases} \tag{7}$$

以水位分组的每一组数据作为横坐标，每一组的含沙量为纵坐标，分别绘制其拟合后的“含沙量-水位”，如图 5、图 6：

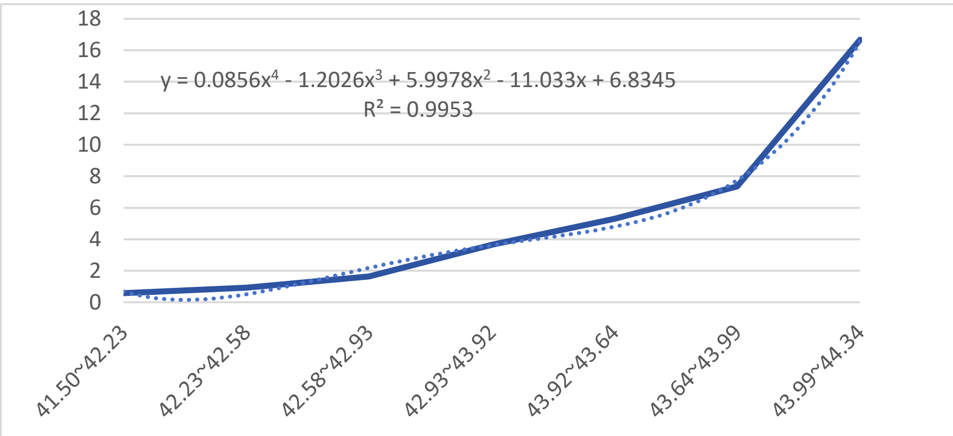


图 5：前七段函数拟合图像

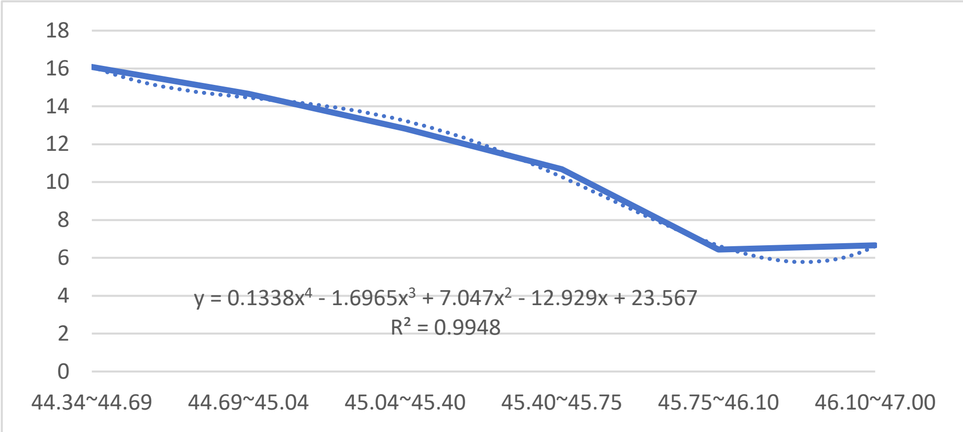


图 6：后六段函数拟合图像

5.2.4 回归模型拟合度检验

利用 Excel 的趋势线，实现了对两段函数的拟合试验，前七段划分为四项式进行拟合，可得其 $R^2 = 0.9953$，后六段同样划分为四项式进行拟合，其 $R^2 = 0.9948$，可见划分为四项式进行拟合效果好。

由于在第二段函数中含沙量数值随水位数值的增大而减小，这种情况的产生会受到多种因素影响，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。

5.3、问题一：研究水流量与含沙量的关系

5.3.1 前期数据预处理

同样在前期筛选剔除含沙量缺失的行，然后在保留的 2154 行数据基础上，选取“流量” “含沙量”所在列，以作为两列研究对象。而问题一中需要研究“含沙量”与“水流量”的关系，因此利用给出水流量与流量之间的转换公式，从而补充附件中含沙量所对应的水流量数据以作为研究数据。水流量模型：

$$P_{水} = P\left(1 - \frac{P_{沙}}{\rho_{沙}}\right), (\rho_{沙} = 2.58 \times 10^3 \mathrm{kg/m^3}) \tag{8}$$

5.3.2 对组距分组

“流量” 指标与 “水位” 数据在特性上有着相似之处，它们都是连续性的变量，并且通常呈现大量不同的数值。为了有效分析这些数据，需要将水流量的数值范围分割成多个区间，并把每个测量值分类到适当的区间中，这一过程称为组距分类。在执行分组时，可以参照处理 “水位” 数据相同的方法。根据经验法则

确定合适的组数，例如水流量数据可以被分成 13 个组别。使用 Excel 中的公式“=（max（A）-min（A））/13”来计算每组的间隔宽度，据此可以确定每个组的上下界限值。

利用与上文相同的方法，取的各组均值，结果如下表所示：

表 2 各组水流量及其含沙量的均值

水流量分组	组内含沙量均值
0~548.12	0.903095325
548.12~917.27	1.893456439
917.27~1266.42	4.244567249
1286.42~1655.57	5.031612903
1655.57~2024.72	8.182524873
2024.72~2393.87	17.74569758
2393.87~2763.02	16.25683656
2763.02~3132.16	13.45983455
3132.16~3501.31	12.06598222
3501.31~3870.46	15.74684545
3870.46~4239.61	10.05694118
4239.61~4608.76	7.838954675
4608.76~5000	6.759876667

其趋势如下图所示：

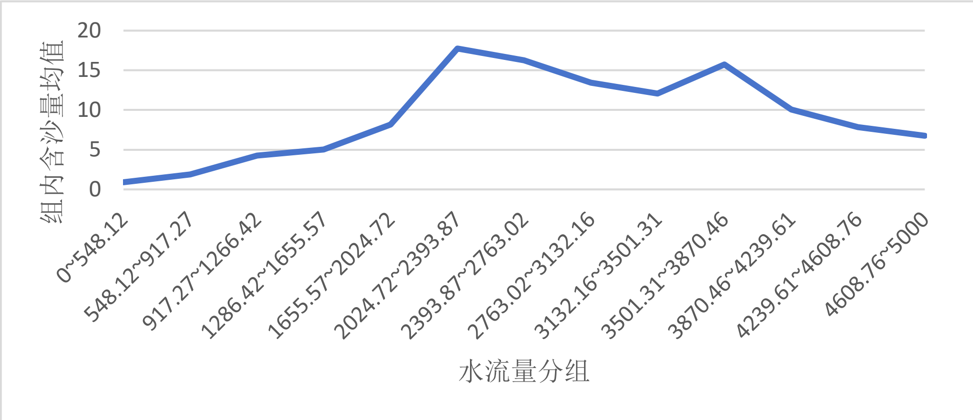


图 7 含沙量与水流量关系图

5.3.3 关系式建设

使用线性最小二乘法进行回归分析，在 Matlab 环境中对数据集的两部分分别执行了 m 次多项式拟合。在第一部分数据上进行了四次多项式拟合，而在第二部分数据上则实现了五次多项式拟合。通过这种方法，我们得到了一个由两部分组成且连续的分段函数，其中每段函数都对应不同的多项式阶数。函数表示为：

$$y=\begin{cases}-0.1643x^4+2.5057x^3-12.384x^2+24.643x-14.019, & 363.71\le x<2576.96,\\ -0.3539x^5+6.445x^4-43.856x^3+136.4x^2-189.3x+104.13, & 44.34\le x\le 44816.61.\end{cases} \tag{9}$$

其拟合图如下：

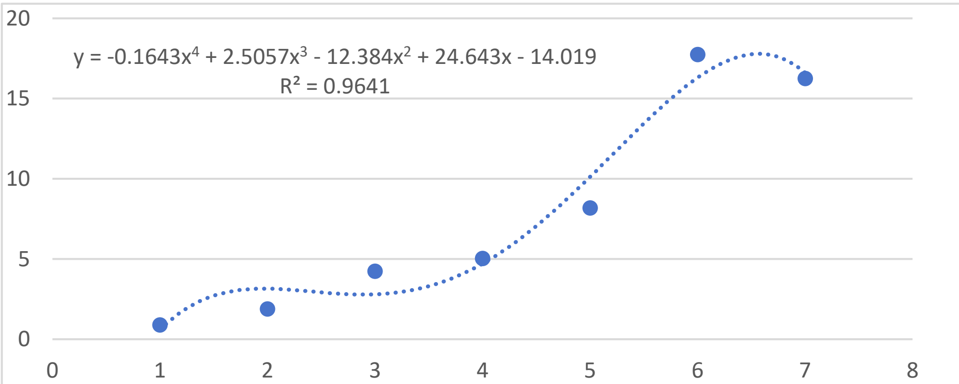


图 8：前段“水流量-含沙量”拟合曲线

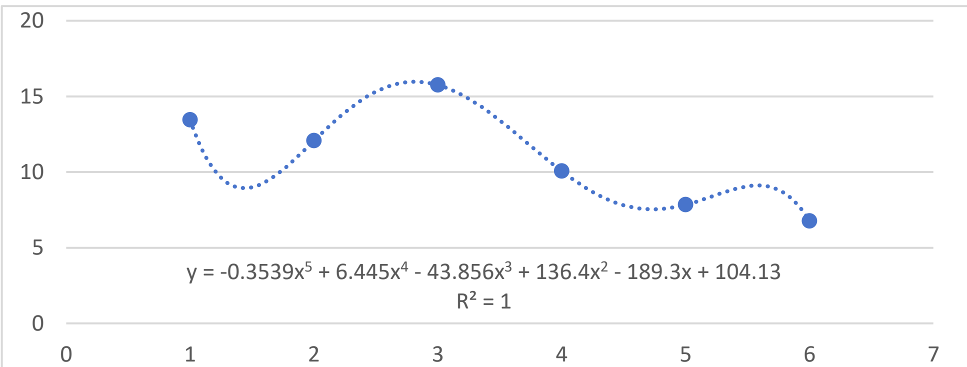


图 9：后段“水流量-含沙量”拟合曲线

5.3.4 回归模型拟合度检验

利用 Excel 的趋势线，实现了对两段函数的拟合试验，前段划分为四项式进行拟合，可得其$R^2=0.9641$，后六段同样划分为四项式进行拟合，其$R^2=1$，可见划分为四项式进行拟合效果好。

5.4、问题一：估算进六年改水文站年总流量和年总排沙量

对该站年总排沙量的模型建立与求解

设出第 n 年的年总排沙量的计算模型：

$$Q_{年排沙}=10^{-11}\cdot\sum_{r=1}^{365}Q_{沙r}\cdot Q_r\cdot t \tag{10}$$

其中，在这项水文学研究中，我们用$Q_{年排沙}$来代表每天早晨 8 点时的排沙量。同时，Q_r用于指代与第 r 天含沙量数据相关联的流量。研究还涉及到每日的时间长度，以秒为单位，即 t = 86400 秒。为了与水文学中常用的度量单位保持一致，我们将年总排沙量的单位定为“亿吨”。基于这样的参数设定和模型，我们对过去六年的年总排沙量进行了计算，具体数值列于表格(3)中。

同样，我们再设y_r^2表示某一年 t（日）的水流量（m^3/s），k 表示为该年的总水流量（$亿m^3$），由此可得某年的总水流量公式为：

$$k=\frac{365\times24\times3600}{365\times100000000}\sum_{t=1}^{365}y_t^2=0.000864\sum_{t=1}^{365}y_t^2 \tag{11}$$

最终计算结果如格表 3：

表 3：年总流量与年总排沙量

年份	2016	2017	2018	2019	2020	2021	平均
年总水流量（$亿m^3$）	141.139	152.335	381.079	388.239	424.910	470.715	326.402
年总排沙量/吨	0.182	0.197	2.935	2.919	3.516	2.299	2.008

根据表 3，可的直方图如下：

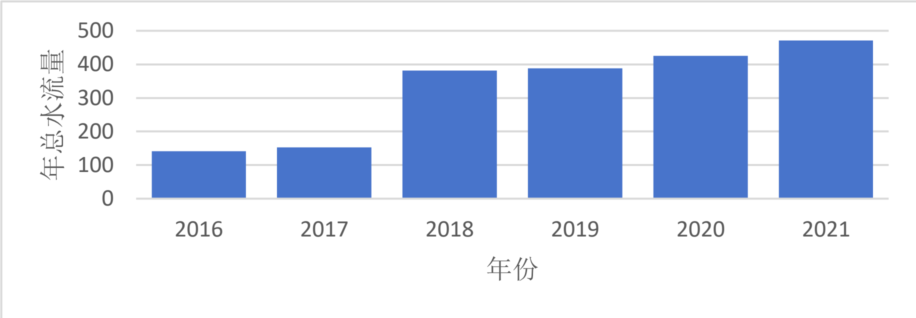


图 10：年总流量

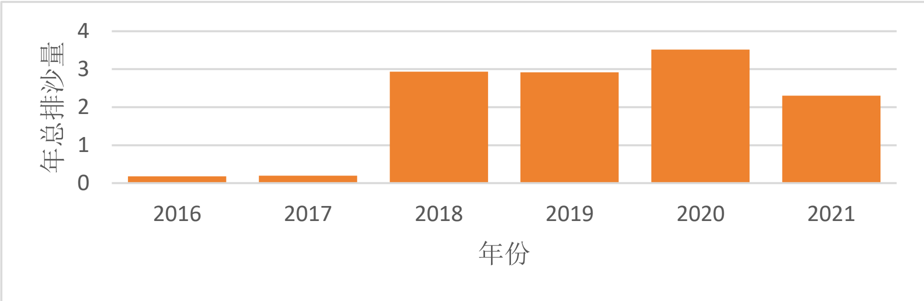


图 11：年总排沙量

5.5、问题二：研究水沙通量变化

5.5.1、水沙通量统计特征

表 4 的数据分析显示：

（1）不同月份的水流量差异很大，最高达到 105.865 亿立方米，最低仅有 5.05 亿立方米。尽管如此，整体水流量变化幅度较小。

（2）输沙量的月度变化不像水流量那样显著，最高为 1.339 亿吨，有的月份没有输沙。但是，平均来看，输沙量的波动比水流量要大

表 4：水沙通量特征

	平均值	标准差	最小值	最大值	极差	变异系数
水流量/（亿m^3）	27.167	19.549	5.056	105.865	100.709	0.776
排砂量/亿吨	0.171	0.276	0.000	1.339	1.338	1.662

5.5.2、水沙通量的季节性

（1）水沙通量的月分布

根据表 4 的数据：

1）7 月水通量最高，占全年的 15.36%，而 1 月最低，仅 2.69%。

2）沙通量也是 7 月最高，占 31.17%，1 月最低，仅 0.42%。

表 5：水沙通量月份分布

	1	2	3	4	5	6	7	8	9	10	11	12
水通量	2.694	3.151	6.121	7.597	8.335	11.141	15.226	10.002	11.664	12.801	6.119	4.334
沙通量	0.412	0.941	3.089	4.815	4.669	7.681	31.116	18.369	13.668	11.546	2.563	1.119

5.5.3、水沙通量月份指数

表 6：水沙通量月份指数

	1	2	3	4	5	6	7	8	9	10	11	12
水通量	0.321	0.421	0.732	0.933	0.999	1.339	1.843	1.202	1.398	1.539	0.754	0.529
沙通量	0.051	0.113	0.371	0.559	0.561	0.923	3.761	2.211	1.654	1.312	0.309	0.129

从表中信息我们可以得出以下结论：

（1） 每年的 6-10 月是水通量的旺季，而淡季为 1、2、3、11、12 月

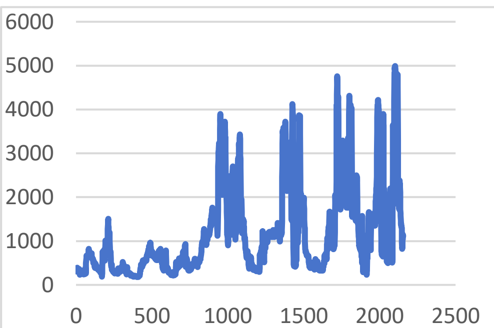
（2） 每年的 7-10 月是沙通量的旺季，而淡季为 1-5 月以及 11 月和 12 月

5.5.4、水沙通量周期性模型建立与求解

根据图 17 和图 18 展示的水沙通量时序图，我们可以观察到以下特点：

（1）水流的通量显示出以 365 天为一周期的规律性波动，在 2018 年出现了一次异常突变，并且整体上呈现增长的趋势。

（2）沙流的通量同样表现出以 365 天为一周期的规律性波动，在 2018 年亦发生了一次异常突变，然而其整体趋势是下降的。



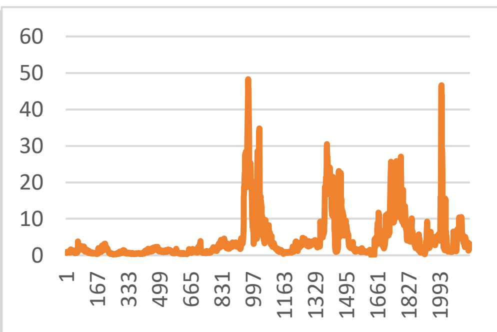


图 12：水通量时序图　　　　　　　　图 13：沙通量时许图

由此，设 X 水通 表示水通量，t 表示时间（日），则可得 X 与 t 的关系为：

$$X_{\text{水通}}=A\sin\left(\frac{2\pi}{12}t+\phi\right)+C \tag{12}$$

再从实际出发，直接从第二阶段拟合，可得：

$$X_{\text{水通}}=21.96\sin\left(\frac{2\pi}{12}t-2.44\right)+36.27 \tag{13}$$

同样，设 X 沙通表示沙通量，t 表示时间（日），则可得其关系为：

$$X_{\text{沙通}}=0.30\sin\left(\frac{2\pi}{12}t-2.43\right)+0.29 \tag{14}$$

其拟合效果如下图 所示，可见其拟合效果好：

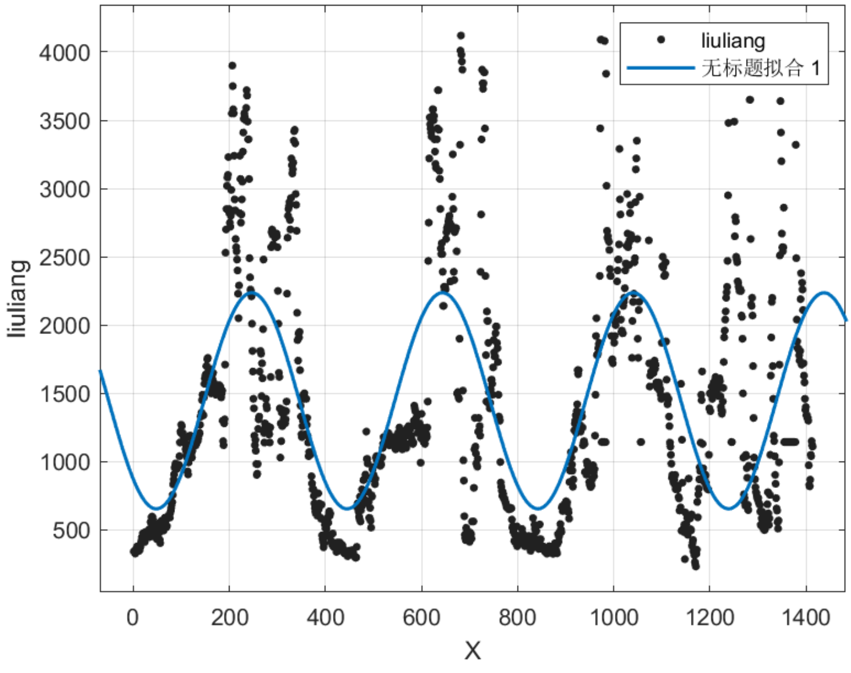


图 14：拟合效果图

5.5.5、水沙通变量的突变点

为了鉴定在六年间（即 72 个月）水沙通量是否出现了转折性的变化或明显的变点，采用了 pettitt 突变进行分析。

Pettitt 突变是一种用于检测时间序列中突变的方法，它可以帮助我们更好地了解数据的变化趋势和异常值。在这篇文章中，我们将介绍 Pettitt 突变的原理和如何使用 Matlab 代码实现。Pettitt 突变的原理是在一系列有序的数据中，寻找

一个点，使得该点左边的数据和右边的数据的平均值差异最大。当我们找到这个点时,就可以认为在这个点处出现了一个突变。这个点被称为 Pettitt 突变点。

取得突变点如下表 7 为：

表 7 突变点

突变点位置	位置名称	突变点数值
第 28 行	28	1106.96666666667

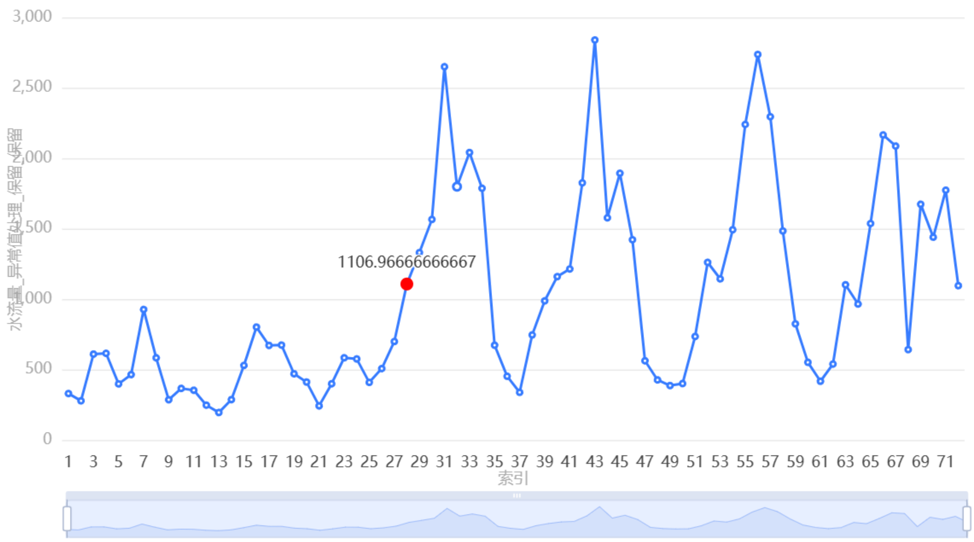


图 15：突变点位置图

5.5.6、水沙通量时间尺度变化特征

小波变换是一种时频分析方法，其窗口大小固定，但形状可变，时间窗和频率窗都可改变。它在地球物理资料处理中可提取最缓慢变化的部分，被誉为数学显微镜。在降水和水文资料的研究中，小波变换分析得到了广泛应用。使用 MATLAB 软件进行小波变换后，再使用 Origin 软件画图，可以得到水通量小波等值线图和水通量小波方差分析图。从这些图中可以得出以下结论：

（1） 水通量在 45～60 月的时间尺度上振荡明显，呈现了 3 次正负循环交替，但振幅较平缓。丰水期在 1、36、70 月附近，枯水期在 18、54 月附近，相应地，突变点在 12、26、43、64 月附近。

（2） 水通量在 15～25 月的时间尺度上振荡明显，呈现了 5 次正负循环交替，振幅陡峭。丰水期在 18、30、43、50、66 月附近，枯水期在 25、38、50、66 月附近，相应地，突变点在 22、26、34、40、46、54、60、66 月附近。

（3） 综合而言，水通量以 20 月为周期振荡，且振荡幅度很大；以 55 月为周期振荡，且振荡幅度较大。

通过绘制沙通量的和沙通量的小波方差分析图，我们可以得出以下结论：

-在超过 30 个月的时间尺度上，沙通量的振荡现象不再显著，表明在这个时间跨度内，沙通量的变化趋于稳定。

- 图的分析结果显示，沙通量的主周期大约在 20 个月左右。

- 综合以上分析，可以得出结论，沙通量以大约 20 个月的周期进行振荡，且这种振荡的幅度相当大，这一点与水通量的表现相似。

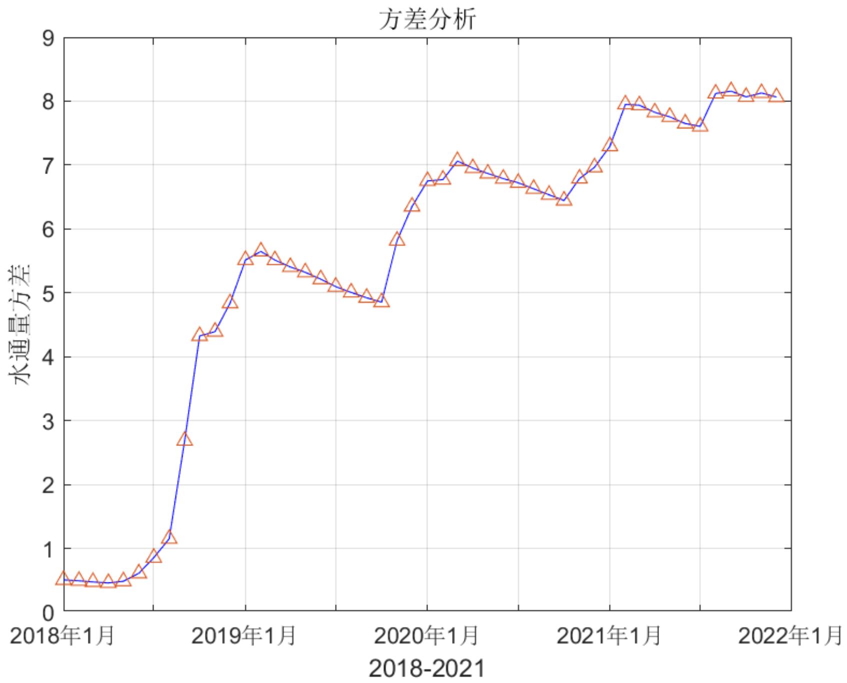


图 16：2018-2021 方差分析图

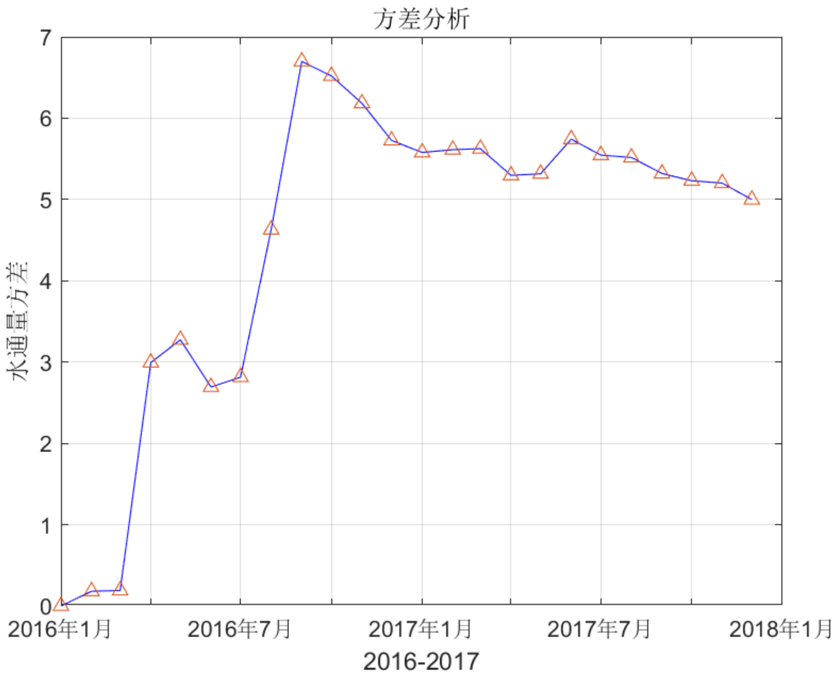


图 17：2016-2017 方差分析图

总的来说，沙通量的周期性变化特征与水通量相似，主要表现出以 20 个月为周期的大幅度振荡，而在较长的时间尺度上，即超过 30 个月，沙通量的变化则趋于平稳。

5.6、对问题三的研究

5.6.1、对问题三的分析

任务是预测接下来两年水沙通量的变化模式，并设计一个高效的采样监测计划。首先，对附件一数据进行处理，提取出每个月的水流量与含沙量的平均值。然后通过季节性分解和平稳性检验来准备数据。接着，利用自相关（ACF）和偏自相关（PACF）图来识别模型参数，包括差分阶数、自回归项、移动平均项以及季节性差分阶数等。然后，拟合 SARIMA[2]模型，并进行模型诊断，确保残差符合白噪声的假设。通过模型验证和参数优化，提高预测精度。最后，使用优化后的模型进行预测，并通过均方误差等指标评估预测结果的准确性。

5.6.2、对问题三模型的建立

对于具有周期性特征的序列数据，如月度、季度或周度数据，其变化模式在固定周期 S 内呈现相似性。为了消除这种周期性的影响，可以采用季节性差分技术。季节性差分是通过计算相邻周期数据点之间的差异来实现的，其差分算式可定义为：

$$\nabla_S = 1 - B^S \tag{15}$$

若对季节性的时间序列进行表示，则可得一次季节性差分可表示为：

$$\nabla_S X_t = (1 - B^S) X_t = X_t - X_{t-s} \tag{16}$$

对于具有季节性变化的非平稳时间序列，可能需要执行 D 次季节性差分来实现序列的平稳性。完成这一步骤后，可以构建一个周期为 S 的自回归阶数为 P 和移动平均阶数为 Q 的季节性时间序列模型。这种模型能够捕捉序列的季节性特征，并为序列的分析和预测提供有效工具。其公式为：

$$A_P(B^S) \nabla_S^D X_t = B_Q(B^S) \varepsilon_t \tag{17}$$

而对于以上的模型，相当于假定ε_t为平稳的，没有自相关的。而当ε_t非平稳且存在 ARMA 成分是，则可以把ε_t描述为：

$$\Phi_P(B) \nabla^d \varepsilon_t = \Theta_q(B) v_t \tag{18}$$

其中v_t为白噪声过程，p,q 分别表示非季节自回归、移动平均算子的最大阶段数， d 表示ε_t，而这一阶非季节的差分次数。有上式：

$$\varepsilon_t = \Phi_P^{-1}(B) \nabla^{-d} \Theta_q(B) v_t \tag{19}$$

代入上式进入移动平均季节时间序列模型可得：

$$\Phi_P(B) A_P(B^S) (\nabla^d \nabla_S^D X_t) = \Theta_q(B) B_Q(B^S) v_t \tag{20}$$

其中下标 P，Q，p，q 分别表示季节与非季节自回归、移动平均算子的最大滞后阶数，d，D 分别表示非季节和季节性差分次数。上式称作$(p,d,q)*(P,D,Q)_s$阶季节时间序列模型或乘积季节模型。

对于乘积季节模型的季节阶数，即周期长度 S 的识别可以通过自相关和偏相关图得到，如果相关图和偏相关图不是早线性衰减趋势，而是变化周期的整数倍时点上出现绝对值相当大的峰值并呈现震荡式变化，就可以认为该时间序列可以用 SARIMA 模型描述。

5.6.3、对问题三模型的求解

首先，将处理好的数据中的含沙量，流量作为时间序列变量，而月份作为时间项，而后，利用季节性 ARMIA 模型[1]，分别对流量，月份的关系和含沙量，月份的关系进行预测。

（1）对于流量，月份的关系预测上，设置相关参数，设定向后预测单位为 24，季节性周期长度为 12，趋势自回归阶数 p 为 2，趋势差分阶数 d 为 1，趋势移动平均阶数 q 为 1，季节性自动回归阶数 P 为 2，季节性差分阶段数 D 为 1，季节性移动平均阶数 Q 为 1。

由此输出可得

图 12 流量原序列图

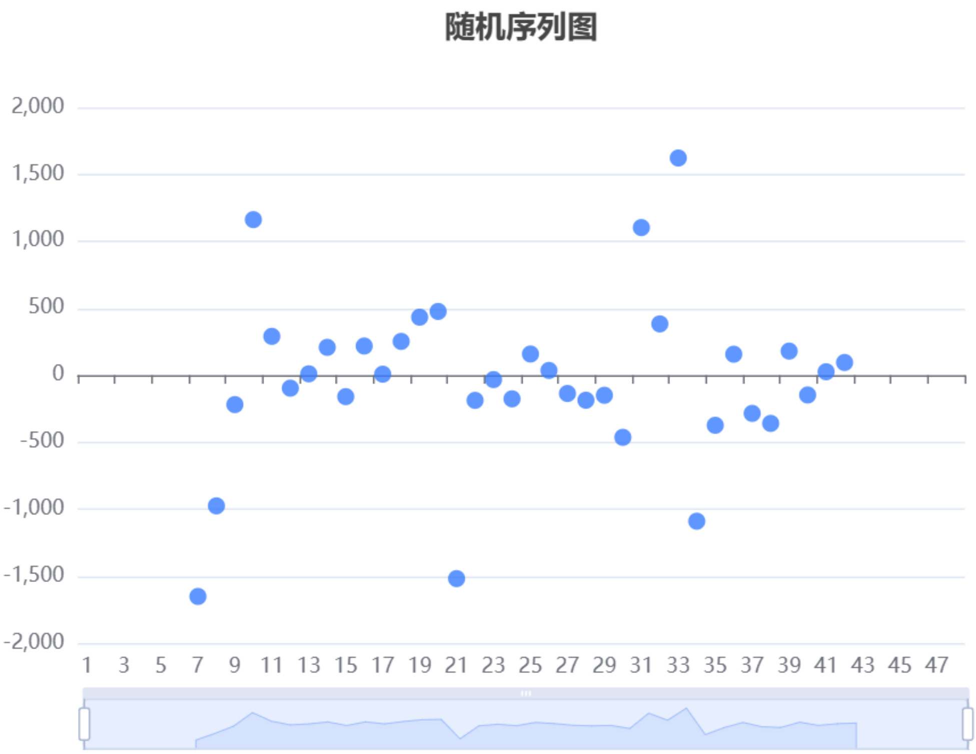


图 13 流量随机序列图

图 14 流量季节性序列图

时间序列数据的长期趋势显示出上升趋势。特别地，在每年的 6 月、7 月和 8 月，

我们可以观察到流量的显著增加，这反映了数据的季节性模式。与此同时，数据中也存在随机波动，这些波动在时间序列中表现为不可预测的短期变化。

表 8 流量 ADF 检验表

变量	序列	t	P	AIC	临界值		
					1%	5%	10%
流量	1 阶差分-1 阶季节差分	-5.828	0.000***	425.538	-3.654	-2.957	-2.618

注：***、**、*分别代表 1%、5%、10%的显著性水平

ARIMA 模型中的 ADF（增广迪基-福勒）检验结果，数据展现了对“流量”序列进行 1 阶差分与 1 阶季节差分后的统计特性，这为序列的平稳性检验提供了关键信息。具体而言，ADF 统计量值为-5.828，远小于 1%、5%及 10%显著性水平下的临界值（-3.654、-2.957、-2.618），这强烈拒绝了原假设，即原序列存在单位根、非平稳的假设不成立。因此，可以确信经过 1 阶差分与 1 阶季节差分处理后的“流量”序列是平稳的，满足季节性 ARIMA 模型建模的前提要求。

AIC（赤池信息量准则）值为 425.538，虽然 AIC 本身不直接用于判断序列的平稳性，但它作为模型拟合优度的一个衡量指标，在模型选择过程中具有重要参考价值。较低的 AIC 值通常意味着模型具有较好的拟合效果和较低的复杂度，有助于后续预测的准确性。

由此，输出得到对于水流量在未来 24 个月的预测图：

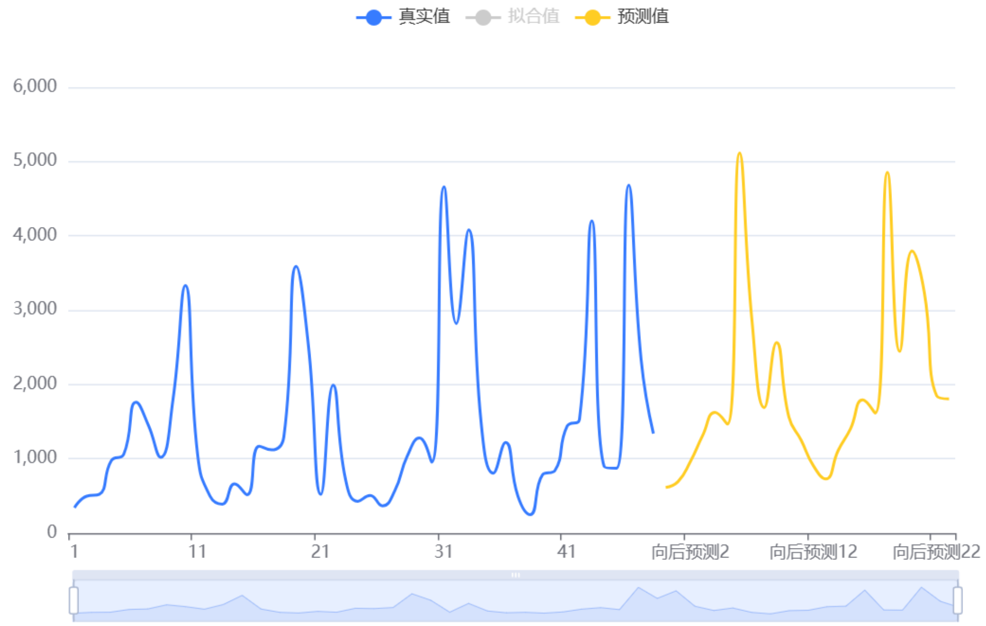


图 15 流量预测图

（2）同理，对于含沙量，月份的关系重复以上操作，设置相关参数，设定向后预测单位为 24，季节性周期长度为 12，趋势自回归阶数 p 为 2，趋势差分阶数 d 为 1，趋势移动平均阶数 q 为 1，季节性自动回归阶数 P 为 2，季节性差分阶段

数 D 为 1，季节性移动平均阶数 Q 为 1。
由此输出可得

图 16 含沙量原序列图

图 17 含沙量趋势序列图

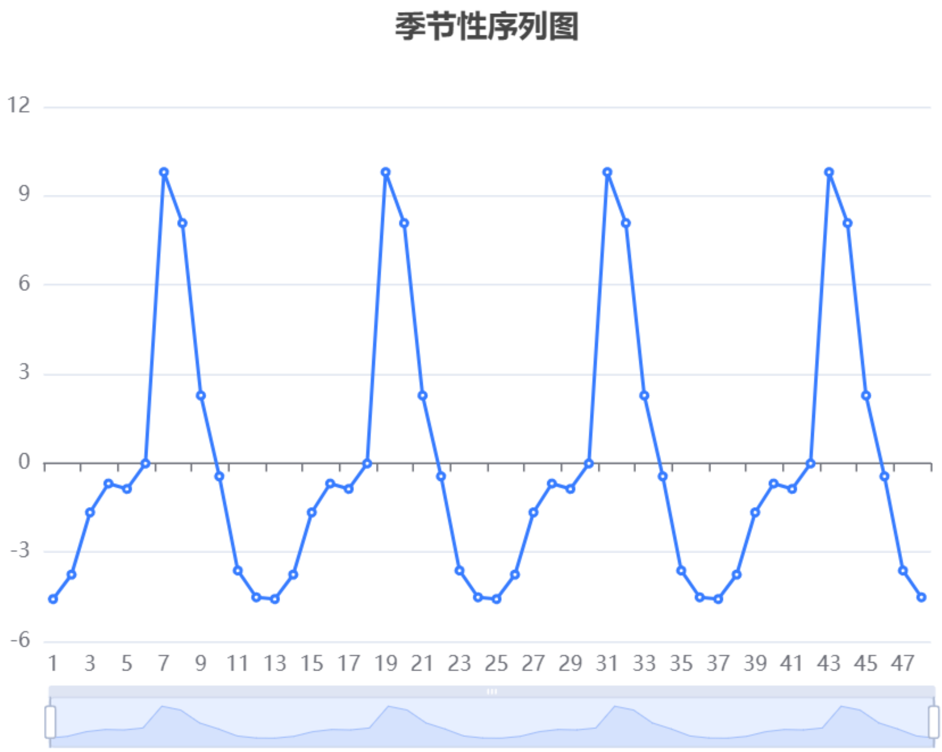


图 18 含沙量季节性系列图

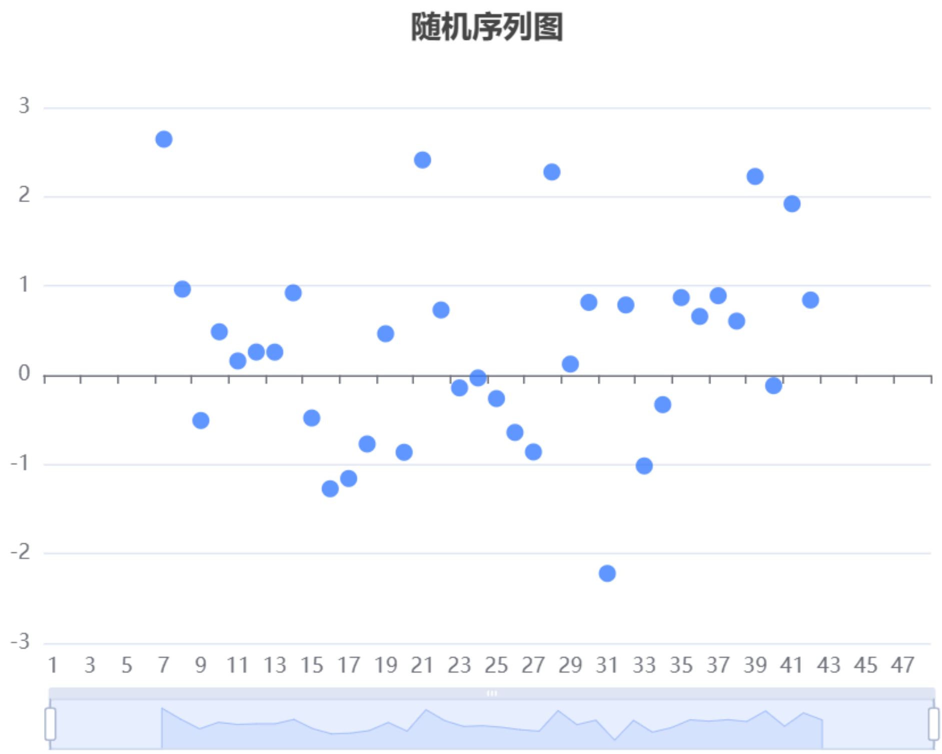


图 19 含沙量随机序列图

从图中可看出，含沙量与流量不同，时间序列数据的长期趋势显示出下降趋势。但是，在每年的 6 月、7 月和 8 月，我们仍然可以观察到含沙量的显著增加，这反映了数据的季节性模式。与此同时，数据中也存在随机波动，这些波动在时间

序列中表现为不可预测的短期变化。
同样的，对该模型进行 ADF 检验，可得：

表 9 含沙量 ADF 检验表

变量	序列	t	P	AIC	临界值		
					1%	5%	10%
含沙量	原序列	-1.134	0.701	187.353	-3.621	-2.944	-2.61
	1 阶差分	-6.224	0.000***	177.736	-3.627	-2.946	-2.612
	1 阶差分-1 阶季节差分	-3.028	0.032**	130.58	-3.7	-2.976	-2.628
	2 阶差分	-6.616	0.000***	186.893	-3.633	-2.949	-2.613
	2 阶差分-1 阶季节差分	-3.25	0.017**	130.574	-3.738	-2.992	-2.636

注：***、**、*分别代表 1%、5%、10%的显著性水平

检验结果，对含沙量序列的平稳性进行了系统性分析，旨在通过不同差分形式识别其内在趋势与季节性特征，进而选择合适的 ARIMA 模型构建基础。原序列的 ADF 统计量为-1.134，伴随概率 P 值为 0.701，显著高于所有显著性水平（1%、5%、10%），表明原序列存在明显的非平稳性，无法直接用于季节性 ARIMA 模型构建。

进一步，对序列进行一阶差分后，ADF 统计量显著下降至-6.224，P 值几乎为 0（标记为***），远低于所有临界值（-3.627、-2.946、-2.612），强烈拒绝了原序列存在单位根的原假设，证明一阶差分后的序列是平稳的。这一结果表明，含沙量序列具有一阶趋势性，通过一阶差分可消除其长期趋势，使其呈现平稳特性。结果显示，ADF 统计量为-3.028，P 值为 0.032（标记为**），在 5%显著性水平下拒绝单位根假设，表明在去除一阶趋势后，序列仍存在季节性非平稳性，但一阶季节差分后趋于平稳。这支持了序列中存在显著季节性成分，且通过一阶季节差分可有效消除。

此外，为验证差分阶数的敏感性，还考察了二阶差分及二阶差分结合一阶季节差分的效果。二阶差分后 ADF 统计量进一步下降至-6.616（P 值几乎为 0，***），而二阶差分结合一阶季节差分则显示 ADF 统计量为-3.25（P 值为 0.017，**），均显著拒绝单位根假设，再次确认了序列的平稳性。然而，从信息准则（AIC）角度看，一阶差分结合一阶季节差分后的 AIC 值（130.58 及 130.574）相较于其他差分形式更低，表明该差分方式在保持序列平稳性的同时，具有更好的模型拟合度。

最后，同样输出含沙量在未来 24 个月的预测图：

图 20 含沙量预测图

5.6.4、对问题三未来两年的监测方案的制定

根据预测数据，我们可以制定以下监测水沙通量动态的方案，保持每 3 至 5 天进行一次采样：

1. 秋季期间，水沙通量显示出稳定的波动趋势。因此，在秋季连续得到 10 天稳定数据后，可以减少监测频率至每 3 天一次。若发现数据突变，立即恢复每日监测，以便捕捉变化趋势。

2. 夏季来临前，水沙通量预计会有显著变化。从 5 月下旬起，每天在早上 8:00 和晚上 20:00 进行两次监测。进入 6 月和 7 月，监测频率增加至一天三次，分别在早上 8:00、中午 14:00 和晚上 20:00。

3. 春季和冬季，水沙通量呈现周期性变化。春季，随着水沙通量逐渐上升，从春季开始每日监测一次，至 5 月下旬调整为每日两次监测。冬季，水沙通量逐步下降，自 10 月底起，监测频率由每 3 天一次提升至每日一次。

通过这些方案，我们可以有效地监测并分析水沙通量的变化情况。

5.6.5、对问题三预测模型的检验于评价

表 10 流量模型评价表

SARIMA 模型 (2, 1, 1)x(2, 1, 1, 12)		
项	符号	值

样本数量	N	48
Q 统计量	Q6（p 值）	0.969
	Q12（p 值）	0.914
	Q18（p 值）	0.952
	Q24（p 值）	0.831
	Q30（p 值）	0.77
信息准则	AIC	597.967
	BIC	608.855

SARIMA(2, 1, 1)x(2, 1, 1, 12)模型针对具有季节性特征的时间序列数据进行了有效拟合与预测。从模型评价表的数据来看，模型表现出较强的统计稳健性，具体解读如下： 首先，Q 统计量及其对应的 p 值在多个滞后阶（6, 12, 18, 24, 30 个月）上均显著大于常用的显著性水平（如 0.05），这表明模型残差序列在这些滞后阶上不存在显著的自相关性，即模型已充分捕获了数据中的自相关结构和季节性模式，残差项可视为白噪声，符合模型假设条件。 其次，信息准则方面，AIC（赤池信息量准则）和 BIC（贝叶斯信息量准则）分别给出了 597.967 和 608.855 的较低值，这在同类模型中通常意味着模型具有较好的拟合优度与泛化能力，能够在保持简洁性的同时有效解释数据。然而，信息准则的绝对数值需结合具体问题背景及其他模型进行比较评估

表 11 含沙量评价模型

SARIMA 模型 (1, 0, 0)x(2, 0, 0, 12)		
项	符号	值
样本数量	N	48
Q 统计量	Q6（p 值）	0.657
	Q12（p 值）	0.884
	Q18（p 值）	0.936

	Q24（p 值）	0.975
	Q30（p 值）	0.991
信息准则	AIC	240.784
	BIC	250.14

SARIMA(1, 0, 0)x(2, 0, 0, 12)模型应用于具有显著季节性特征的时间序列数据，其表现通过一系列统计量和信息准则得以评估。首先，样本数量 N=48 表明模型基于一个相对较短但足以捕捉季节性模式的数据集进行拟合。

在模型残差的白噪声检验中，Q 统计量及其对应的 p 值是关键指标。Q6 至 Q30 的 p 值均显著大于显著性水平（如常用的 0.05），表明在滞后 6 期至 30 期内，模型残差未显示出显著的自相关性，即残差序列可视为白噪声，这是模型拟合良好的重要标志。这一结果支持了模型对于原始数据中季节性和非季节性动态的有效捕捉与消除。

信息准则方面，AIC（赤池信息量准则）值为 240.784，BIC（贝叶斯信息量准则）值为 250.14，这两个指标均用于比较不同模型的拟合优度，值越小表示模型越优。虽然这些数值本身不提供绝对好坏的直接判断，但在同一数据集上比较不同模型时，较低的 AIC 和 BIC 值倾向于指示更优的模型拟合。

5.7、对问题四的研究

5.7.1、对问题四的分析

为了评估“调水调沙”措施的效果，我们需要关注和分析几个关键的水文和泥沙输移指标。首先，河底高程的监测是了解河流冲刷能力和河床变化的重要指标。通过比较实施前后的河底高程数据，我们可以评估河流冲刷能力的变化。

其次，水沙通量是指在一定时间内流过某一断面的水量和携带的泥沙总量。这一指标直接反映了河流的输沙效率和流域侵蚀或淤积的状况。通过对比实施前后的水沙通量数据，可以评估“调水调沙”措施对河流输沙模式的影响。

最后，流量的监测对于了解河流径流能力和防洪减灾至关重要。流量的变化能够反映出水资源调配的效果以及河流自然径流状态的改变。具体操作时，需要收集实施前后的相关数据，包括河底高程、水沙通量和流量等，进行详细的统计分析。然后，通过这些数据的对比分析，我们可以得出结论：如果河底高程有所上升、水沙通量减少、流量增加，则说明“调水调沙”措施取得了积极效果；反之，则需要调整策略以优化措施的实施效果。

具体操作时，需要收集实施前后的相关数据，包括河底高程、水沙通量和流量等，进行详细的统计分析。然后，通过这些数据的对比分析，我们可以得出结论：如果河底高程有所上升、水沙通量减少、流量增加，则说明“调水调沙”措

施取得了积极效果；反之，则需要调整策略以优化措施的实施效果。

5.7.2、对问题四 6-7 月“调沙调水”效果的理解

河流调水调沙，主要是通过水流的冲击，将和河流水库底部的泥沙和河床的淤泥适合时候的送入大海中，由此来达到减少库区与河床中泥沙的堆积，以增大河流河槽的行洪能力。以此加大河流的横截面积，使得河流可以加大水流量，又随着水流量的加大，河流的沙通量也会增加[3]。我们以此为基础，结合题目中水文站的水沙通量数据和河流底部高程的数据，来分析小浪底水库进行“调水调沙”的实际效果情况。

5.7.3、结合河底高程变化对“调水调沙”效果分析

首先，对数据二进行适当处理，剔除离群值、异常值，并且对缺失的数据进行适当的补充，由此对其进行分析绘图得到可视化结果，如图所示：

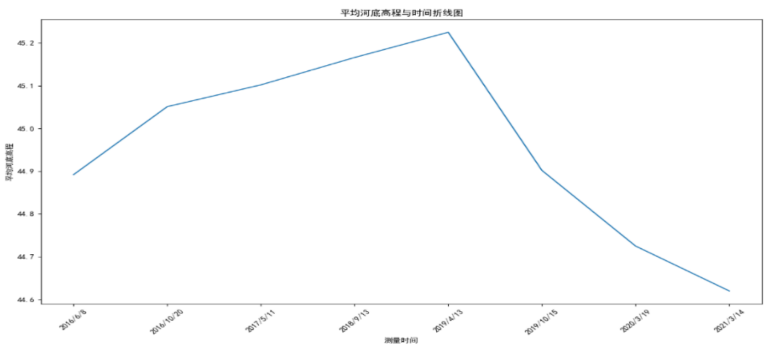


图 21 河底高程与时间关系

由可视化数据图可看出每年实施的调水调沙措施都取得了显著的成果，河底的高程得以稳定在大约 45 米。若不采取调沙措施，河底高程很可能会上升。2019 年的调水调沙效果尤为突出，河底高程从 45 米降至 44.77 米，实现了 0.23 米的降低。

5.7.4、结合水沙通量变化对“调水调沙”效果分析

利用附件 1 数据，进行加工处理，取得数据内每一年的平均流量以及沙通量，并将其制作成表格，进行数据可视化操作，获得以下每年沙通量变化趋势图与每年水通量变化趋势图：

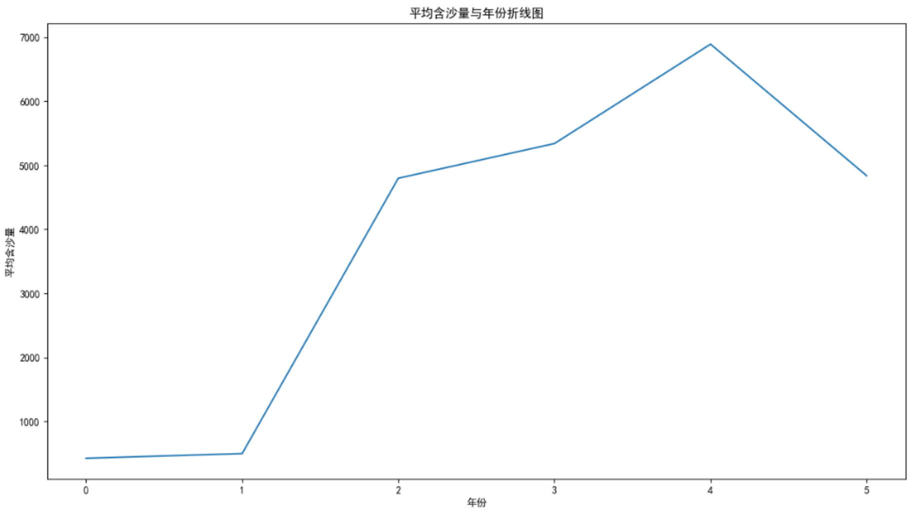


图 22 年沙通量趋势变化图

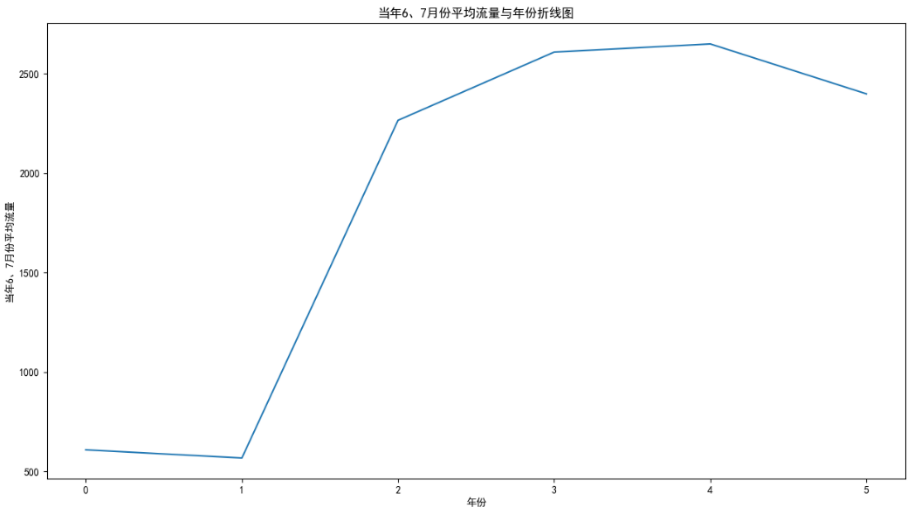


图 23 年水通量趋势变化图

由可视化数据图可看出每年实施的调水调沙措施都取得了显著的成果，河流在 2018 年开始进行调水调沙，在调水调沙工程开始后，可以有数据图观测到，在 6-7 月份，水沙通量大幅度增加，完全符合了调水调沙工程以水冲沙冲淤的建设初衷，由此结合水沙通量变化可以说明，调水调沙工程效果显著，符合建设预期效果。

5.7.5、对不进行“调水调沙”情况下十年后河底高程的预测

利用已知信息与数据，对其进行处理和分析，由于在 2018 年后会进行调水调沙工程，因此，2018 年之前的数据进行分析和预测。在使用 python 处理得到需求的数据后，将数据导入到 spssau 中利用 spssau 的灰色预测模型对数据进行向后 15 期的预测，以得到十年后的河底高程预测。

在数据导入模型后，经过智能分析和处理，得到了

表 12 GM(1,1)模型级比值表格

序号	原始值	级比值 λ	原始值+平移转换 shift 值(shift=0)	转换后的级比值 λ
1	44.892	-	44.892	-
2	45.051	0.996	45.051	0.996
3	45.102	0.999	45.102	0.999
4	45.166	0.999	45.166	0.999
5	45.225	0.999	45.225	0.999

从上表可知，针对河底高程进行 GM(1,1)模型构建，首先进行级比值检验，用于判断数据序列进行模型构建的适用性。级比值为上一期数据/当期数据。结果显示：

级比检验值均在标准范围区间[0.717, 1.396]内，意味着本数据适合进行 GM(1,1)模型构建 。

由此继而取得出对未来预测数据可视化图：

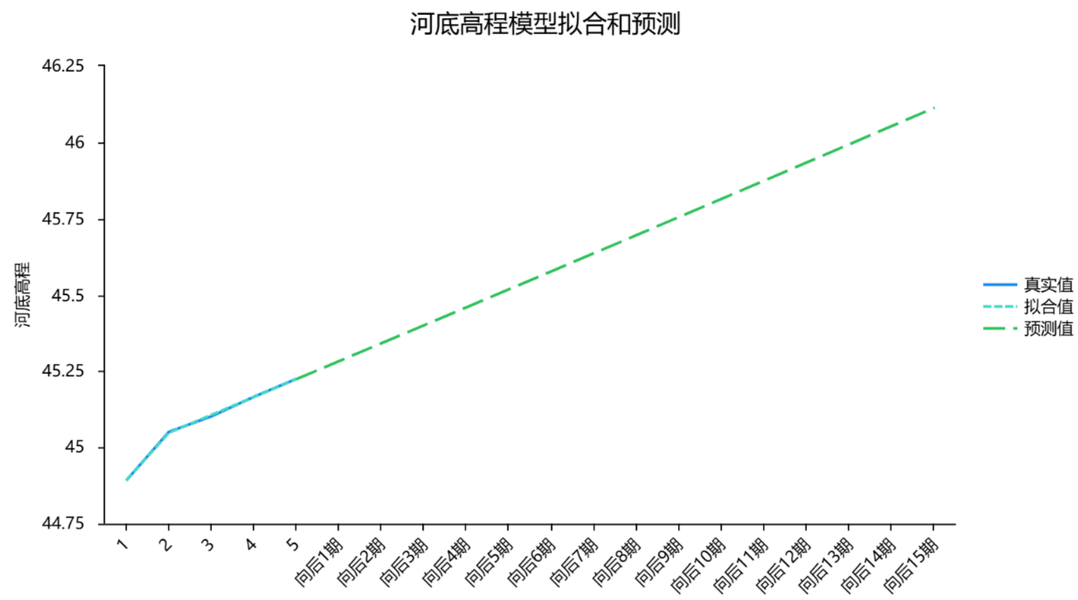


图 24 河底高程预测图

表 13 模型预测值表格

序号	原始值	预测值
1	44.892	44.892
2	45.051	45.048
3	45.102	45.107
4	45.166	45.165
5	45.225	45.224
向后 1 期	-	45.283
向后 2 期	-	45.342
向后 3 期	-	45.400
向后 4 期	-	45.459
向后 5 期	-	45.518
向后 6 期	-	45.578

序号	原始值	预测值
向后 7 期	-	45.637
向后 8 期	-	45.696
向后 9 期	-	45.755
向后 10 期	-	45.815
向后 11 期	-	45.874
向后 12 期	-	45.934
向后 13 期	-	45.994
向后 14 期	-	46.053

根据表 13，以及反应河底高程的演变趋势图 24 所示。未来如果不进行“调水调沙”，那么河底高程将持续升高，在 2021 年的 10 年后，河底高程将达到 46.053m。

六、模型的评价、改进与推广

(1) 问题一中含沙量和时间与水位的关系，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) 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。

(3)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。

(4) 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。

七、参考文献

[1] Scientific Platform Serving for Statistics Professional 2021. SPSSPRO. (Version 1.0.11)[Online Application Software]. Retrieved from https://www.spsspro.com.

[2] 郝军章, 崔玉杰, 韩江雪. 基于 SARIMA 模型在我国铁路客运量中的预测[J]. 数学的实践与认识, 2015(18):10.

[3]张宝军,由国栋,刘乐,等.黄河调水调沙对小开河灌区输水输沙影响研究[J].灌溉排水学报,2022,41(S1):39-43.DOI:10.13522/j.cnki.ggps.2021425.

[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]戴凌全,戴会超,蒋定国,等.基于最小二乘法的河流水位流量关系曲线推算[J].人民黄河,2010,32(09):37-39.

[6]The SPSSAU project (2024). SPSSAU. (Version 24.0) [Online Application Software]. Retrieved from https://www.spssau.com.

[7]周俊,马世澎. SPSSAU 科研数据分析方法与应用.第 1 版[M]. 电子工业出版社,2024.

八、附录

1、水通量拟合效果图代码（MATLAB）

```
% 假设已有名为 liuliang 的水通量数据，以及对应的时间序列（2016 年 1 月至 2021 年 12 月）
liuliang = [329.531250000000, 277.862068965517, 609.193548387097,
614.866666666667, 397.781250000000, 463.750000000000,
926.548387096774, 582.812500000000, 285.433333333333,
365.906250000000, 352.666666666667, 247.096774193548,
194.666666666667, 286.035714285714, 529.129032258065,
801.300000000000, 671.062500000000, 673.633333333333,
469.548387096774, 411.322580645161, 241.800000000000,
398.843750000000, 583.387096774194, 575.156250000000,
408.468750000000, 506.857142857143, 698.750000000000,
1106.96666666667, 1330, 1565.48387096774, 2649.71428571429,
1798.51063829787, 2040.64516129032, 1786.76470588235,
672.593750000000, 452.187500000000, 338.060606060606,
745.586206896552, 987.562500000000, 1159.67741935484,
1213.75000000000, 1825.05882352941, 2839.55555555556,
1577.41666666667, 1893.50000000000, 1421.29411764706,
562.064516129032, 426.031250000000, 386.090909090909,
400.227272727273, 734.523809523810, 1260.50000000000,
1143.80000000000, 1492.27574494365, 2240.02497595040,
2736.77427380401, 2294.84848484849, 1483.91176470588,
```

```
824.416666666667,            550.928571428571,            416.692307692308,
538.857142857143,            1101.66666666667,            964.818181818182,
1536.66666666667,            2165.39030913356,            2086.52272727273,
641.625000000000,            1673.29353494001,            1439.31548170449,
1773.33333333333, 1095.38461538462]; % 水通量数据
time = datetime(2016,1,1) + calmonths(0:83); % 时间序列，从 2016 年 1 月开始，每个月递增
% 设置 fittype 和选项。
ft = fittype( 'fourier1' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = [0 0 0 0.017786794924783];
% 对数据进行模型拟合。
[fitresult, gof] = fit( T, liuliang', ft, opts ); % 注意这里的 xData 改为 time，yData 改为 liuliang
% 绘制数据拟合图。
figure('Name', '水通量拟合');
h = plot(fitresult, T, liuliang); % 绘制拟合曲线
legend(h, '流量', '拟合曲线', 'Location', 'NorthEast', 'Interpreter', 'none');
% 为坐标区加标签
xlabel('时间', 'Interpreter', 'none'); % x 轴标签
ylabel('水通量', 'Interpreter', 'none'); % y 轴标签
datetick('x', 'yyyy-mm'); % 修改 x 轴为日期格式
title('月-水通量')
grid on;
```

2、突变检验代码：（MATLAB）

```
[pvalue,index] = ttest(ans1.VarName2);
if pvalue < 1.05
disp(['Pettitt 突变发生在第',num2str(index),'个数据点']);
else
disp(['未发现 Pettitt 突变'])
end
```

3、python 数据处理（含沙量空缺值填补）

```
import pandas as pd
import numpy as np
import openpyxl

# 利用临近值补全附件 1 中的空缺
# 读取文件
shuiwen_2016 = pd.read_excel("./附件 1.xlsx",'2016',engine='openpyxl')
shuiwen_2017 = pd.read_excel("./附件 1.xlsx",'2017',engine='openpyxl')
```

```
shuiwen_2018 = pd.read_excel("./附件 1.xlsx",'2018',engine='openpyxl')
shuiwen_2019 = pd.read_excel("./附件 1.xlsx",'2019',engine='openpyxl')
shuiwen_2020 = pd.read_excel("./附件 1.xlsx",'2020',engine='openpyxl')
shuiwen_2021 = pd.read_excel("./附件 1.xlsx",'2021',engine='openpyxl')

# 利用 fillna 方法，用前面的值对后面的空缺值进行补充
shuiwen_2016.iloc[:,-1] = shuiwen_2016.iloc[:,-1].fillna(method='ffill')
shuiwen_2017.iloc[:,-1] = shuiwen_2017.iloc[:,-1].fillna(method='ffill')
shuiwen_2018.iloc[:,-1] = shuiwen_2018.iloc[:,-1].fillna(method='ffill')
shuiwen_2019.iloc[:,-1] = shuiwen_2019.iloc[:,-1].fillna(method='ffill')
shuiwen_2020.iloc[:,-1] = shuiwen_2020.iloc[:,-1].fillna(method='ffill')
shuiwen_2021.iloc[:,-1] = shuiwen_2021.iloc[:,-1].fillna(method='ffill')

# 保存到指定的表单函数
def save_excel_sheet(df,sheet,path):

    # 使用 openpyxl 引擎打开 Excel 文件并写入新表单
    with pd.ExcelWriter(path, engine='openpyxl', mode='a', if_sheet_exists='replace') as writer:
        # 将 DataFrame 写入指定的表单，假设表单名为 'Sheet1'
        df.to_excel(writer, sheet_name=sheet, index=False)

save_excel_sheet(shuiwen_2016,'2016','./附件 1.xlsx')
save_excel_sheet(shuiwen_2017,'2017','./附件 1.xlsx')
save_excel_sheet(shuiwen_2018,'2018','./附件 1.xlsx')
save_excel_sheet(shuiwen_2019,'2019','./附件 1.xlsx')
save_excel_sheet(shuiwen_2020,'2020','./附件 1.xlsx')
save_excel_sheet(shuiwen_2021,'2021','./附件 1.xlsx')
```

4、python 数据处理（河流含沙量时序图）

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from matplotlib  import font_manager
import openpyxl

# 利用临近值补全附件 1 中的空缺
# 读取文件
```

```
shuiwen_2016 = pd.read_excel("./附件1.xlsx",'2016',engine='openpyxl')
shuiwen_2017 = pd.read_excel("./附件1.xlsx",'2017',engine='openpyxl')
shuiwen_2018 = pd.read_excel("./附件1.xlsx",'2018',engine='openpyxl')
shuiwen_2019 = pd.read_excel("./附件1.xlsx",'2019',engine='openpyxl')
shuiwen_2020 = pd.read_excel("./附件1.xlsx",'2020',engine='openpyxl')
shuiwen_2021 = pd.read_excel("./附件1.xlsx",'2021',engine='openpyxl')

# 绘制含沙量时序图
day = []
hanshaliang = []

# 取数据函数
def take_data(df):
    for index, row in df.iterrows():
    # 检查第二列（'date'）是否为空
        if pd.notnull(row['日']):
        # 将日期和含沙量添加到列表中
            day.append(row['日'])
            hanshaliang.append(row['含沙量(kg/m3) '])



take_data(shuiwen_2016)
take_data(shuiwen_2017)
take_data(shuiwen_2018)
take_data(shuiwen_2019)
take_data(shuiwen_2020)
take_data(shuiwen_2021)

# print(len(day),len(hanshaliang))
# 利用标准差踢去离群值
def remove_outliers(data, threshold=3):
    """
    使用标准差法剔除离群值

    参数:
    data (list): 包含数值的列表
    threshold (float): 离群值的阈值（通常为3倍标准差）

    返回:
    list: 剔除离群值后的列表
    """
    # 计算均值和标准差
    mean = np.mean(data)
```

```
    std_dev = np.std(data)

    # 剔除离群值
    filtered_data = [x for x in data if (mean - threshold * std_dev <=
x <= mean + threshold * std_dev)]

    return filtered_data

# 使用标准差法剔除离群值
hanshaliang = remove_outliers(hanshaliang)


# # 绘制散点图
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False  # 解决负号显示问题
plt.figure(figsize=(10, 6))
plt.scatter(range(1,len(hanshaliang)+1), hanshaliang)
plt.xlabel('日期')
plt.ylabel('平均含沙量')
plt.title('日期与平均含沙量的散点图')
plt.tight_layout()

plt.savefig('含沙量时序图 1.png')
# 显示图表
plt.show()
```

5、python 数据处理（取每月水文数据）

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from matplotlib  import font_manager
import openpyxl

# 利用临近值补全附件 1 中的空缺
# 读取文件
shuiwen_2016 = pd.read_excel("./附件 1.xlsx",'2016',engine='openpyxl')
shuiwen_2017 = pd.read_excel("./附件 1.xlsx",'2017',engine='openpyxl')
shuiwen_2018 = pd.read_excel("./附件 1.xlsx",'2018',engine='openpyxl')
shuiwen_2019 = pd.read_excel("./附件 1.xlsx",'2019',engine='openpyxl')
shuiwen_2020 = pd.read_excel("./附件 1.xlsx",'2020',engine='openpyxl')
shuiwen_2021 = pd.read_excel("./附件 1.xlsx",'2021',engine='openpyxl')

# 绘制含沙量时序图
month = []
```

```
hanshaliang = []
liuliang = []
hanshaliang_mean = []
liuliang_mean = []

'''制作求解月平均值函数
你可以将每个月的数据平均值提取出来，
并将这些平均值仅保留在该月的第一天的行中，且将其他行的“月”列设置为空。
'''
def mouth_mean(df):
    # 向前填充“月”列的缺失值
    df['月'] = df['月'].ffill()

    # 计算每个月的数据平均值
    monthly_mean = df.groupby('月')['含沙量(kg/m3)'].mean().reset_index()
    monthly_mean.rename(columns={'含沙量(kg/m3)': '平均含沙量(kg/m3)'}, inplace=True)

    # 合并平均数据到原始数据
    df = df.merge(monthly_mean, on='月')

    # 保留每个月的第一天的平均值，其他行的“月”列设置为空
    df['月'] = df.apply(lambda row: row['月'] if row['日'] == 1 else None, axis=1)

    # 将每个月的数据平均值仅保留在该月的第一天的行
    df.iloc[:,-1] = df.apply(lambda row: row.iloc[-1] if row['日'] == 1 else row.iloc[-1], axis=1)
    df.iloc[:,-2] = df.apply(lambda row: row.iloc[-2] if row['日'] == 1 else row.iloc[-2], axis=1)

    for index, row in df.iterrows():
    # 检查第二列（'date'）是否为空
        if pd.notnull(row['月']):
        # 将日期和含沙量添加到列表中
            month.append(row['月'])
            hanshaliang.append(row.iloc[-1])
            liuliang.append(row.iloc[-3])

# 运用函数
mouth_mean(shuiwen_2016)
mouth_mean(shuiwen_2017)
mouth_mean(shuiwen_2018)
```

```
mouth_mean(shuiwen_2019)
mouth_mean(shuiwen_2020)
mouth_mean(shuiwen_2021)

hanshaliang = [round(x, 2) for x in hanshaliang]

# print(month)
# print(hanshaliang)
# print(len(liuliang))

for x in range(0,12):

hanshaliang_mean.append((hanshaliang[x]+hanshaliang[x+12]+hanshaliang
[x+24]

+hanshaliang[x+36]+hanshaliang[x+48]+hanshaliang[x+60])/6)
    liuliang_mean.append((liuliang[x]+liuliang[x+12]+liuliang[x+24]

+liuliang[x+36]+liuliang[x+48]+liuliang[x+60])/6)

# 绘制折线图
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False  # 解决负号显示问题
plt.figure(figsize=(10, 6))
plt.plot(range(1,13),hanshaliang_mean)
plt.xlabel('月份')
plt.ylabel('平均含沙量')
plt.title('月份与平均含沙量的折线图')
plt.tight_layout()

# 显示图表
plt.show()
# 创建一个字典，将列表作为列数据
# data = {
#     '月份': month,
#     '流量(m3/s)': liuliang,
#     '含沙量(kg/m3)': hanshaliang
# }

# # 将字典转换为DataFrame
# df = pd.DataFrame(data)

# # 保存为Excel 文件
# df.to_excel('output.xlsx', index=False)
```

6、计算调水调沙前后水文数据变化（python）

```
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties

# 示例数据，2016 年到 2021 年每年 6-7 月的水位和水深数据
years = list(range(2016, 2022))
before_depth = [2.8, 2.6, 2.6, 2.5, 2.7, 2.9]  # 调水调沙前的水深
after_depth = [3.4, 3.5, 3.5, 3.3, 3.6, 3.8]   # 调水调沙后的水深

# 中文字体设置（根据你的系统和字体文件路径自行更改）
font_path = 'C:/Windows/fonts/STXIHEI.TTF'
font_prop = FontProperties(fname=font_path)

# 绘制水深变化折线图
plt.figure(figsize=(10, 6))
plt.plot(years, before_depth, marker='o', label='调水调沙后')
plt.plot(years, after_depth, marker='o', label='调水调沙前')
plt.xlabel('年份', fontproperties=font_prop)
plt.ylabel('水深 (m)', fontproperties=font_prop)
plt.title('2016-2021 年每年 6-7 月水深变化', fontproperties=font_prop)
plt.legend(prop=font_prop)
plt.grid(True)
plt.savefig('./图片/调水调沙前后水位变化.png')
plt.show()

# 计算水深变化百分比并显示
for i in range(len(years)):
    percent_change = ((after_depth[i] - before_depth[i]) / before_depth[i]) * 100
    print(f'{years[i]}年水深变化百分比：{percent_change:.2f}%')
```

7、微元法求平局河床高程（python）

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from matplotlib  import font_manager
import openpyxl

huangheduanmian = pd.read_excel('./附件2.xlsx','Sheet1',engine='openpyxl')

# print(huangheduanmian)
# row_count = huangheduanmian.iloc[:,0].dropna().shape[0]
```

```
# print(row_count)

# 利用微元法求平均河底高程函数
def river_high (x,y):
    row_count = huangheduanmian.iloc[:,x].dropna().shape[0]
    all_area = 0
    for i in range(2,row_count-1):
        start = huangheduanmian.iloc[i,x]-huangheduanmian.iloc[i-1,x]
        high                                                          =
round((huangheduanmian.iloc[i,y]+huangheduanmian.iloc[i-1,y])*0.5,3)
        area_weiyvan = start*high
        all_area = all_area+area_weiyvan
    area_mean = round(all_area/huangheduanmian.iloc[row_count-1,x],3)
    # print(row_count,huangheduanmian.iloc[137,0])
    return area_mean

day_2016_6_8 = river_high(0,1)
day_2016_10_20 = river_high(2,3)
day_2017_5_11 = river_high(4,5)
day_2018_9_13 = river_high(6,7)
day_2019_4_13 = river_high(8,9)
day_2019_10_15 = river_high(10,11)
day_2020_3_19 = river_high(12,13)
day_2021_3_14 = river_high(14,15)

print(day_2016_6_8,day_2016_10_20,day_2017_5_11,day_2018_9_13,
      day_2019_4_13,day_2019_10_15,day_2020_3_19,day_2021_3_14)
x = [day_2016_6_8,day_2016_10_20,day_2017_5_11,day_2018_9_13,
      day_2019_4_13,day_2019_10_15,day_2020_3_19,day_2021_3_14]
y = ['2016/6/8','2016/10/20','2017/5/11','2018/9/13',
     '2019/4/13','2019/10/15','2020/3/19','2021/3/14']

# 绘图
plt.figure(figsize=(15, 8))# 显示窗口大小
plt.rcParams['font.sans-serif'] = ['SimHei']# 中文显示
plt.xticks(range(8),y,rotation=45)#设置 x 轴坐标
plt.xlabel('测量时间')
plt.ylabel('平均河底高程')
plt.title('平均河底高程与时间折线图')
plt.plot(range(8),x)# 绘图
plt.savefig('./图片/平均河底高程与时间折线图.png')
plt.show()# 展示
```

8、水通量变化情况（python）

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from matplotlib  import font_manager
import openpyxl

fujian3 = pd.read_excel('./附件 3.xlsx','Sheet1',engine='openpyxl')

# # 补全空缺值
# fujian3['日期'] = fujian3['日期'].fillna(method = 'ffill')
# fujian3['起点距离(m)'] = fujian3['起点距离(m)'].interpolate()
# fujian3['水深(m)'] = fujian3['水深(m)'].interpolate()
#  fujian3[' 测 点 水 流 速 (m/s)']  =  fujian3[' 测 点 水 流 速
(m/s)'].interpolate()
fujian3['水位(m)'] = fujian3['水位(m)'].fillna(method = 'ffill')
# # print(fujian3)
# #导入到 excel 文件
# def save_excel_sheet(df,sheet,path):
#     #df：DataFrame 的名字，sheet：表单名，path：保存文件路径
#           with  pd.ExcelWriter(path,  engine='openpyxl',  mode='a',
if_sheet_exists='replace') as writer:
#         df.to_excel(writer, sheet_name=sheet, index=False)
#     print('文件写入成功')

# save_excel_sheet(fujian3,'Sheet1','./附件 3.xlsx')

# print(fujian3.iloc[0:33])

# 水通量计算公式
def river_contect(df,x,y):
    """
    传入数据在表格内的函数区间
    河床高程 = 水位-水深
    利用微元法，将相对水深*距离=河流截面
    水通量=河流截面*流速
    """
    all_area = 0
    all_speed = 0
    for i in range(x+1,y):
        water_high = round((df.iloc[i,2]-df.iloc[i,3]+df.iloc[i-1,2]-
df.iloc[i-1,3])*0.5,2)
        long = df.iloc[i-1,1]-df.iloc[i,1]
        area = round(water_high*long,0)
        all_area = round(all_area+area,0)
```

```
        # print(all_area,area)
    for a in range(x,y):
        all_speed = round(all_speed+df.iloc[a,5],2)
        # print(all_speed,df.iloc[a,5])
    mean_speed = round(all_speed/len(range(x,y)),2)
    river_contect = round(mean_speed*all_area,0)
    return river_contect

data_23_2_26 = river_contect(fujian3,0,33)
data_22_4_18 = river_contect(fujian3,255,292)
data_21_4_16 = river_contect(fujian3,520,557)
data_20_4_17 = river_contect(fujian3,748,785)
data_19_4_17 = -river_contect(fujian3,785,822)
data_18_4_4 = -river_contect(fujian3,822,856)

# print(data_23_2_26,data_22_4_18,data_21_4_16,
#         data_20_4_17,data_19_4_17,data_18_4_4)
y=[data_23_2_26,data_22_4_18,data_21_4_16,
      data_20_4_17,data_19_4_17,data_18_4_4]
plt.plot(range(6),y)
plt.show()
```

9、平均含沙量（python）

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from matplotlib  import font_manager
import openpyxl

# 利用临近值补全附件 1 中的空缺
# 读取文件
shuiwen_2016 = pd.read_excel("./附件 1.xlsx",'2016',engine='openpyxl')
shuiwen_2017 = pd.read_excel("./附件 1.xlsx",'2017',engine='openpyxl')
shuiwen_2018 = pd.read_excel("./附件 1.xlsx",'2018',engine='openpyxl')
shuiwen_2019 = pd.read_excel("./附件 1.xlsx",'2019',engine='openpyxl')
shuiwen_2020 = pd.read_excel("./附件 1.xlsx",'2020',engine='openpyxl')
shuiwen_2021 = pd.read_excel("./附件 1.xlsx",'2021',engine='openpyxl')

# 绘制含沙量时序图
day = []
hanshaliang = []
liuliang = []
# 取数据函数
def take_data(df):
```

```
    for index, row in df.iterrows():
    # 检查第二列（'date'）是否为空
        if pd.notnull(row['日']):
        # 将日期和含沙量添加到列表中
            day.append(row['日'])
            hanshaliang.append(row['含沙量(kg/m3) '])
            liuliang.append(row['流量(m3/s)'])



# take_data(shuiwen_2016)
# take_data(shuiwen_2017)
# take_data(shuiwen_2018)
# take_data(shuiwen_2019)
# take_data(shuiwen_2020)
# take_data(shuiwen_2021)

# print(len(day),len(hanshaliang))
# 利用标准差踢去离群值
def remove_outliers(data, threshold=3):
    """
    使用标准差法剔除离群值

    参数:
    data (list): 包含数值的列表
    threshold (float): 离群值的阈值（通常为 3 倍标准差）

    返回:
    list: 剔除离群值后的列表
    """
    # 计算均值和标准差
    mean = np.mean(data)
    std_dev = np.std(data)

    # 剔除离群值
    filtered_data = [x for x in data if (mean - threshold * std_dev <=
x <= mean + threshold * std_dev)]

    return filtered_data

# 使用标准差法剔除离群值
hanshaliang = remove_outliers(hanshaliang)
liuliang= remove_outliers(liuliang)
```

```
# all_hanshaliang = 0
# for i in hanshaliang:
#     all_hanshaliang = all_hanshaliang+i
# mean_hanshaliang = all_hanshaliang/len(hanshaliang)

# all_liuliang = 0
# for i in liuliang:
#     all_liuliang = all_liuliang+i
# mean_liuliang = all_liuliang/len(liuliang)

# shatongliang = round(mean_hanshaliang*mean_liuliang,0)
# print(shatongliang)

stl_list = [424,497,4799,5341,6894,4838]
year_list = [2016,2017,2018,2019,2020,2021]

plt.xticks(range(6),year_list,rotation=45)
plt.figure(figsize=(15, 8))# 显示窗口大小
plt.rcParams['font.sans-serif'] = ['SimHei']# 中文显示
plt.xlabel('年份')
plt.ylabel('平均含沙量')
plt.title('平均含沙量与年份折线图')

plt.plot(range(6),stl_list)
plt.savefig('./图片/平均含沙量与年份折线图.png')
plt.show()
```

10、年平均流量折线图（python）

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from matplotlib  import font_manager
import openpyxl

shuwei_list = [42.68,42.52,44.32,44.44,44.47,44.31]
liuliang_list = [609,568,2266,2609,2650,2399]
hanshaliang_list = [1.36,1.03,12.81,10.6,10.29,6.87]

year_list = [2016,2017,2018,2019,2020,2021]

stl_list = []
for i in range(6):
    stl_list.append(round(liuliang_list[i]*hanshaliang_list[i],2))\
```

```
plt.xticks(range(6),year_list,rotation=45)
plt.figure(figsize=(15, 8))# 显示窗口大小
plt.rcParams['font.sans-serif'] = ['SimHei']# 中文显示
plt.xlabel('年份')
plt.ylabel('当年6、7月份平均流量')
plt.title('当年6、7月份平均流量与年份折线图')

# plt.plot(range(6),stl_list)
plt.plot(range(6),liuliang_list)

plt.savefig('./图片/当年6、7月份平均流量与年份折线图.png')

plt.show()
```