

基于小波变换和背包模型的小浪底水库最优监测方案研究

摘要

本文针对黄河小浪底水库的水沙通量变化趋势及最优监测方案进行了研究。

首先,对附件 1、附件 2、附件 3 的数据进行了整理,使用 python 软件对大量的缺失数据采用平均值进行了填补。

针对问题(1),首先,研究含沙量与时间的关系,得出结论:一是 2016、2017 年的含沙量较小,但从 2018 年开始大幅度增长,含沙量随着年份存在周期性波动。二是每年 2、3 季度为旺季,每年 1、4 季度为淡季。三是建立了含沙量随时间(日)变化的正弦函数。其次,分别建立了含沙量与水位、水流量的双对数回归方程,得出结论:水位每增加 1%,含沙量增加 36.31%;水流量每增加 1%,含沙量增加 1.06%。第三,估算出年均水流量 324 亿 m^3 ,年均总排沙量 2 亿吨。

针对问题(2),把数据整理成以“月”为时间点的时间序列,首先,描述性分析得出结论:各月水流量分化严重,但平均波动幅度较小,排沙量则相反。其次,季节性分析得出结论:水通量和沙通量都在每年 7 月比例最大,在 1 月占比最小;水通量、沙通量每年 6 月-10 月是旺季,每年 1 月-3 月、11 月、12 月是淡季。第三,周期性分析得出结论:水沙通量的周期为 365 天,2018 年有突变现象。第四,使用 M-K 检验法进行突变性分析得出结论:从 $t=30$ 开始出现显著的上升趋势,在 $t=27$ 时出现突变点。再使用 Fisher 最优分割法进行验证,得出的结论与 M-K 法检验法一致。第五,为了详细而准确地寻找水沙通量的周期性和突变点,还使用小波变换分析法得出结论:水通量、沙通量均以 20 月为周期振荡,呈现了 5 次正负循环交替,出现了 8 个突变点。

针对问题(3),首先,使用 R/S 分析法研究水沙通量是否在未来 2 年存在相同趋势、反转趋势或随机趋势。把全时段 2016-2021 年分为 2 个阶段,第一阶段(2016-2017 年)用于检验,第二阶段(2018-2021 年)用于预测,检验结果表明:R/S 分析法具有一定的可靠性,可以用于预测,预测结果表明:水沙通量未来变化将出现反转趋势。其次,分别针对水通量、沙通量建立了傅里叶级数,预测出未来 2 年的水沙通量数据。第三,借用背包问题建立了 0-1 规划模型,获得了未来 2 年最优的采样监测方案。

针对问题(4),首先,使用“微元法”计算平均河底高程,观察其变化情况得出结论:每年“调水调沙”成效不佳,河底高程始终保持在 45m 左右,2019 年成效尤其不佳,下降了 0.23m。其次,考察水通量的变化情况得出结论:每年“调水调沙”成效不佳,水通量呈现逐年下降趋势,2018 年成效尤其不佳,水通量下降了 72%。如果不“调水调沙”,那么水通量逐年上升趋势越来越严重,2022 年水通量上升幅度尤其不佳,上升了 6.4%。第三,考察沙通量的变化情况得出结论:每年“调水调沙”成效不佳,2021 年与 2020 年相比,沙通量下降了 45.4%。第四,使用灰色 GM(1,1) 模型预测了未来十年的河底高程。

本文优点是使用季节指数、双对数回归方程、M-K 检验法、最优分割法、小波变换法、R/S 分析法研究分析季节性、周期性、突变性,能够交叉验证,可靠性强;使用傅里叶级数逼近水沙通量曲线预测精度较高;在建立最优采样监测方案时巧妙转化为背包问题,存在最优解;在计算曲线的平均高度时,使用了“微元法”。不足之处是在预测未来 2 年的水沙通量时,难以给出突变点数值,导致在突变点处误差较大。

本文所使用的模型或方法可以推广到水文资料分析问题中,还可以推广到其它有关季节性、周期性、突变性等问题的分析中。

关键词: 双对数回归; M-K 检验; 小波变换; R/S 分析; 傅里叶级数; 0-1 规划; 微元法



一、问题重述

通过研究黄河水沙通量的变化规律可以对沿黄流域的环境治理、气候变化和人民生活的影 响做出一定判断,以及对优化黄河流域水资源分配、协调人地关系、调水调沙、防洪减灾提供帮 助。对黄河的治理发 展有重要的理论指导意义。

根据附件中的数据,建立数学模型,解决下列问题:

(1) 研究含沙量与时间、水位、水流量的关系,并估算近 6 年的年总水流量和年总排沙量。

(2) 分析水沙通量的突变性、季节性和周期性等特性,研究水沙通量的变化规律。

(3) 根据水沙通量的变化规律,预测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,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。

(4) 分析每年 6-7 月小浪底水库进行“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将会带来什么结果和影响。

二、问题分析

首先,要对原始数据进行整理,主要是对空缺数据的补充。

针对问题(1),该问题分为两个方面,一是研究水位、水流量、时间与含沙量的关系,二是估算近六年 的该水文站的年总流水量和年总排沙量。对于含沙量与时间的关系,可以通过时序图、季节指数、时间趋 势等方法来描述。对于含沙量与水位的关系,可以通过相关分析、回归分析来解决。然后定义年总流水量 和年总排沙量,并计算 6 年的数值结果。

针对问题(2),水通量就是径流量(亿 m^3),沙通量就是排沙量(亿吨),研究它们的变化规律,需要 从统计特征、季节性、周期性和突变性四个方面来入手。季节性可通过月份指数、季度指数来描述。周期 性可通过时序图、建立三角函数来解决。突变性可通过非参数 Mann-Kendall 检验法来解决。

针对问题(3),为预测水沙通量的变化趋势,我们借鉴 R/S 分析法分析水沙通量的分形特征和长期记忆 过程,以判断其未来存在相同趋势、反转趋势或随机趋势。为了制订采样监测方案,使用小波分析法分析信 号不同周期的时间演变规律,以掌握其丰枯旱涝情况,再根据这些情况制订最优的采样监测方案。

针对问题(4),要评估“调水调沙”的实际效果,可以比较“调水调沙”前后河底高程、水沙通量、水 流量等指标的变化。根据起点距的定义,使用“微元法”近似计算平均河底高程。

针对本文建立的模型,还需要进行敏感性分析和稳健性分析。

本文研究路径如图 1 所示。



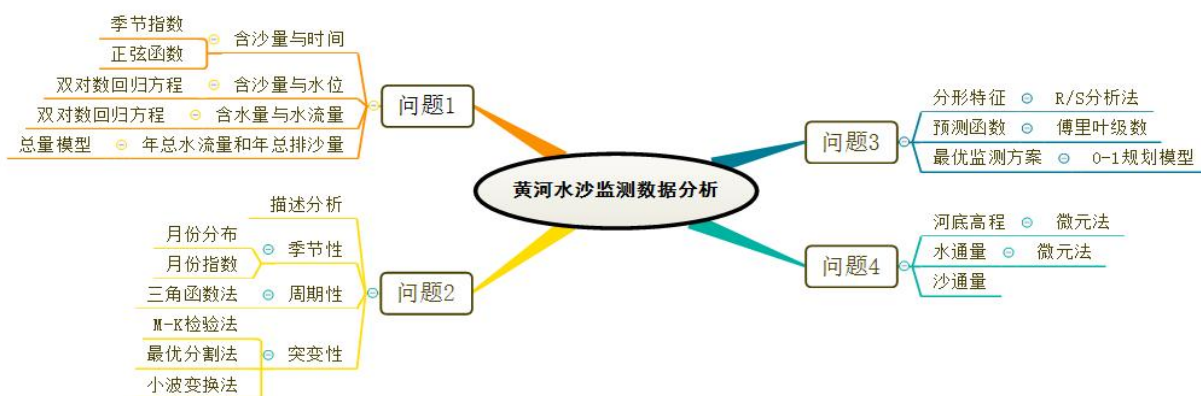


图1 研究路径

三、符号说明

y : 因变量 (含沙量)	x : 自变量 (水位、时间、水流量)
β_1 : 回归系数	β_0 : 回归常数
z : 表示年总水流量 (亿 m ³)	x_1 : 水位
p : 表示水通量 (亿 m ³)	$x_t^{(2)}$: 表示 t 时刻的水流量 (m ³ /s)
q : 表示沙通量 (亿吨)	x_2 : 水流量
$f(t)$: 表示 t 时刻的水沙通量	t : 表示时刻 (日)
x_t : 表示 t 时刻的水沙通量	H : R/S 分析法的统计量
注: 其余符号在文中给出.	

四、模型假设

为了简化问题,作如下假设:

- (1) 附件数据准确无误。
- (2) 水位和河底高程固定均以海拔 72.26 米为基准面。
- (3) $f(t)$ 表示 t 时刻的水沙通量, 假设其满足狄利赫里 (Dirichlet) 条件。
- (4) 水通量、沙通量序列是独立序列。
- (5) 每年 6-7 月份“调水调沙”一次。

五、模型建立与求解

5.0 数据整理

本文提供了以“日”“月”“年”为时间单位的时间序列。首先,检查日期是否连续,如果不连续,就补充完整。其次,由于数据存在缺失,就用邻近值代替。第三,为了排除闰年的影响,一年定为 365 天。附件

1 的数据整理结果（样表）如表 1 所示。

表 1 数据整理

序号	年	月	日	水位(m)	流量(m ³ /s)	含沙量 kg/m ³
1	2016	1	1	42.79	357	0.825
2	2016	1	1	42.8	363	0.825
3	2016	1	1	42.8	363	0.796
4	2016	1	1	42.81	368	0.796
5	2016	1	1	42.84	384	0.796
...						
2338	2016	12	30	42.19	217	0.415

5.1 问题（1）——建立含沙量与时间、水位和水流量的关系

问题分析：含沙量与时间的关系，可以通过时序图、季节指数、时间趋势等方法来描述。含沙量与水位、水流量的关系，可以通过相关性、建立回归方程来描述。

建模思路：首先，研究含沙量与时间的关系，画时序图观察是否存在某种趋势，计算季节指数观察是否存在旺季和淡季。其次，以含沙量为因变量，以水位、水流量为自变量，建立回归方程。

5.1.1 研究含沙量与时间的关系

5.1.1.1 研究含沙量与月份、季度的关系

画出含沙量的时序图，如图 2 所示。从图中可知，有几个离群点，必须剔除。剔除离群点之后，时序图如图 3 所示，从图中可得出以下结论：

- （1）2016、2017 年的含沙量较小，但从 2018 年开始大幅度增长。
- （2）含沙量随着年份存在周期性波动。

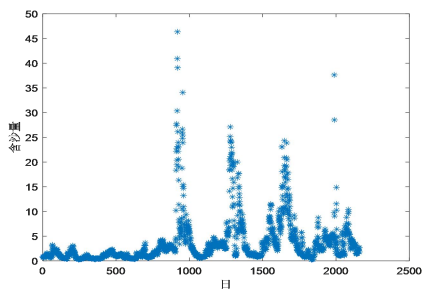


图 2 含沙量时序图

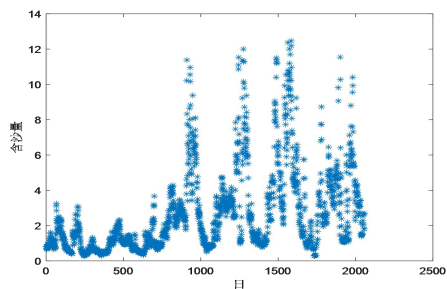


图 3 剔除离群点后的含沙量时序图

分别计算 12 个月的月份指数和 4 个季度的季度指数，如图 4、图 5 所示，从图中可得出以下结论：

- （1）每年 7 月、9 月、10 月为旺季，每年 1 月、2 月、11 月、12 月为淡季。
- （2）每年 2、3 季度为旺季，每年 1、4 季度为淡季。

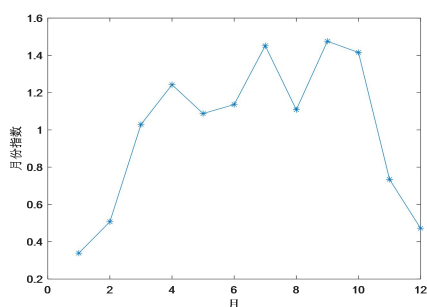


图4 含沙量的月份指数

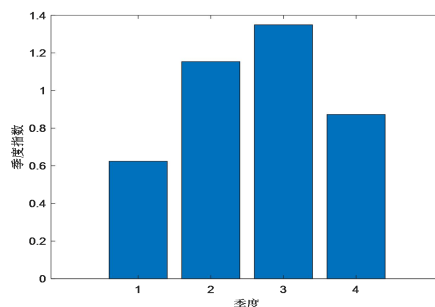


图5 含沙量的季度指数

5.1.1.2 研究含沙量与时刻（日）的关系

设 y 表示含沙量 (kg/m^3), t 表示时刻 (日), y 和 t 的函数关系为

$$y = A \sin(\omega t + \phi) + C \quad (1)$$

由于 y 的周期为 365 天, 故 $T = 365 = \frac{2\pi}{\omega}$, $\omega = \frac{2\pi}{365}$, 于是

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

使用最小二乘法进行分段拟合得,

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

拟合效果如图 6、图 7 所示, 从图中可知, 拟合精度较高。

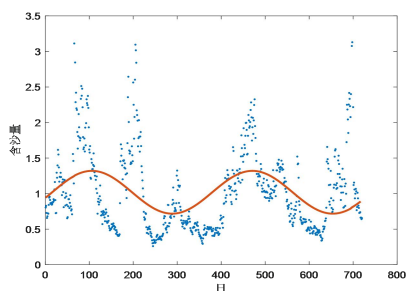


图6 2016-2017 年含沙量函数拟合效果

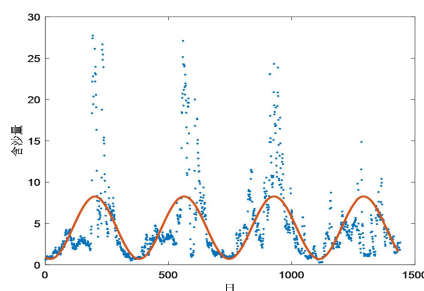


图7 2018-2021 年含沙量函数拟合效果

5.1.2 研究含沙量与水位、水流量的关系

画出水位、水流量、含沙量两两之间的散点图, 如图 6 所示。从图中可得出以下结论:

- (1) 水位与水流量之间是高度的线性正相关关系。

(2) 水位与含沙量之间是线性正相关关系。

(3) 水流量与含沙量之间是线性正相关关系。

基于以上结论,可分别建立含沙量与水位、含沙量与水流量的回归方程,但不能以水位、水流量为自变量建立二元回归模型,这是因为水位与水流量高度相关的缘故。

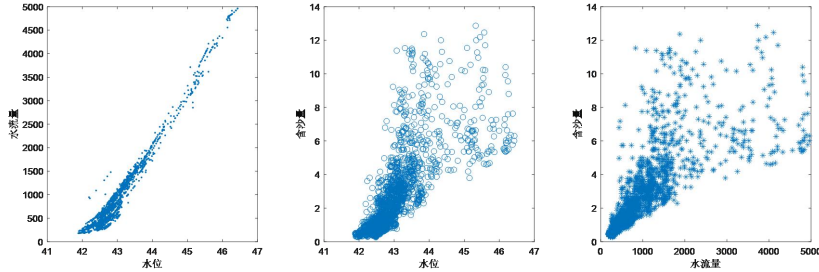


图 8 水位、水流量、含沙量的散点图

5.1.2.1 一元线性回归模型简介

因变量 y 和自变量 x 的一元线性回归模型为^[1]

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (4)$$

其中, β_1 为回归系数, β_0 为回归常数, $\varepsilon \sim N(0, \sigma^2)$, ε_i 相互独立。

估计的回归方程为

$$y = \hat{\beta}_0 + \hat{\beta}_1 x \quad (5)$$

5.1.2.2 含沙量与水位的回归方程

设 y 表示含沙量 (kg/m^3), x_1 表示水位 (m), y 和 x_1 的双对数回归模型为

$$\ln y = \beta_0 + \beta_1 \ln x_1 + \varepsilon \quad (6)$$

使用 MATLAB 软件,给定显著性水平 0.05,使用最小二乘法进行参数估计,结果如表 2 所示。

表 2 参数估计及其检验结果

系数	估计值	95%置信区间下限	95%置信区间上限
β_0	-135.95	-140.98	-130.92
β_1	36.31	34.97	37.64

$$R^2 = 0.5797, \quad p = 0.0000, \quad s_y = 0.31$$

从表 2 可知,2 个回归系数的 95%置信区间均不包含 0,故回归系数检验通过。拟合优度 $R^2=0.5797$, 表明拟合精度尚可; F 检验的相伴概率 $p = 0.0000 < 0.05$,表明含沙量 $\ln y$ 与水位 $\ln x_1$ 的线性关系显著成立。于是有

$$\ln y = -135.95 + 36.31 \ln x_1, \quad x_1 \in (0, +\infty) \quad (7)$$

画出带有正态分布概率曲线的标准化残差直方图,如图 9 所示,从图中可知,残差服从均值为 0 的正态

分布。

以标准化预测值为横坐标,以标准化残差为纵坐标,画散点图,如图 10 所示,从图中可知,残差随机分布在 $\pm 3\sigma$ 以内,且没有明显的趋势,表明残差的标准差是常数,且相互独立。

根据式 (7) 可以得出以下结论:

- (1) 含沙量与水位是正相关关系。
- (2) 水位每增加 1%, 含沙量增加 36.31%, 含沙量对水位的变化非常敏感。

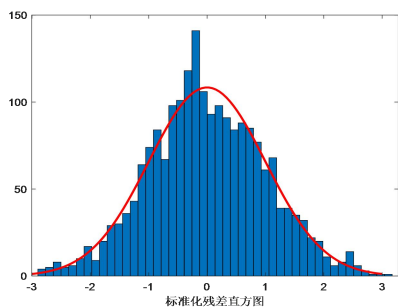


图 9 标准化残差直方图

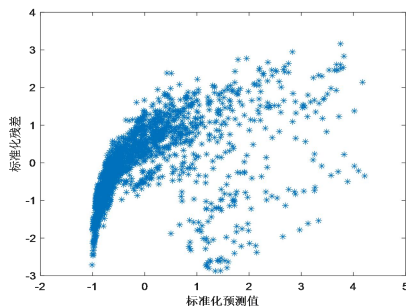


图 10 标准化残差与标准化预测值的散点图

5.1.2.3 含沙量与水流量的回归方程

设 y 表示含沙量 (kg/m^3), x_2 表示水流量 (m^3/s), y 和 x_2 的回归模型为

$$\ln y = \beta_0 + \beta_1 \ln x_1 + \varepsilon \quad (8)$$

使用 MATLAB 软件,给定显著性水平 0.05,使用最小二乘法进行参数估计,结果如表 3 所示。

表 3 参数估计及其检验结果

系数	估计值	95%置信区间下限	95%置信区间上限
β_0	-6.35	-6.50	-6.20
β_1	1.06	1.03	1.08

$$R^2 = 0.8013, \quad p = 0.0000, \quad s_y = 0.1466$$

从表 3 可知,2 个回归系数的 95%置信区间均不包含 0,故回归系数检验通过。拟合优度 $R^2=0.8013$,表明拟合精度较高; F 检验的相伴概率 $p = 0.0000 < 0.05$,表明 $\ln y$ 与 $\ln x_2$ 的线性关系显著成立。于是有

$$\ln y = -6.35 + 1.06 \ln x_2, \quad x_2 \in (0, +\infty) \quad (9)$$

画出带有正态分布概率曲线的标准化残差直方图,如图 11 所示,从图中可知,残差服从均值为 0 的正态分布。

以标准化预测值为横坐标,以标准化残差为纵坐标,画散点图,如图 12 所示,从图中可知,大多数残差随机分布在 $\pm 3\sigma$ 以内,且没有明显的趋势,表明残差的标准差是常数,且相互独立。但存在个别离群点。

根据式 (9) 可以得出以下结论:

- (1) 含沙量与水位是正相关关系。

(2) 水流量每增加 1%, 含沙量增加 1.06%, 含沙量对水位的变化不敏感。

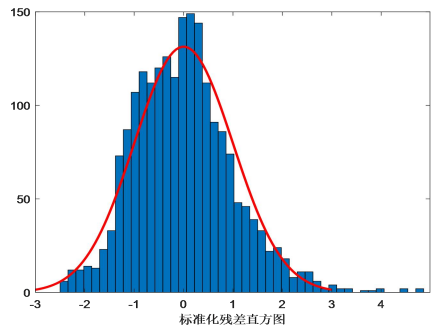


图 11 标准化残差直方图

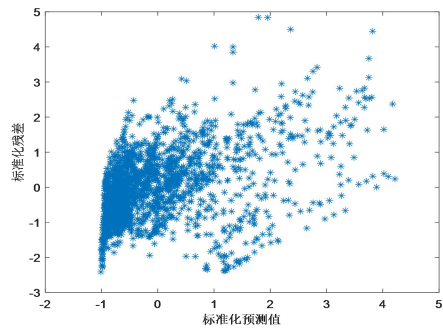


图 12 标准化残差与标准化预测值的散点图

5.1.3 估算年总水流量和年总排沙量

设 $x_t^{(2)}$ 表示某年 t 时刻（日）的水流量（ m^3/s ）， z 表示该年总水流量（亿 m^3 ），则某年总水流量为

$$z = \frac{365 \times 24 \times 3600}{365 \times 1000000000} \sum_{t=1}^{365} x_t^{(2)} = 0.000864 \sum_{t=1}^{365} x_t^{(2)} \quad (10)$$

设 y_t 表示 t 时刻（日）的含沙量（ kg/m^3 ）， u 表示年总含沙量（亿 t），则该年总含沙量为

$$u = \frac{\sum_{t=1}^{365} y_t z_t}{10^{11}} \quad (11)$$

计算结果如表 4 所示。

表 4 年总水流量与年总排沙量

年份	2016	2017	2018	2019	2020	2021	平均
年总流量/亿 m3	141.13	151.31	383.06	381.95	424.22	466.71	324.73
年总含沙量/吨	0.18	0.19	2.91	2.99	3.41	2.24	1.99

根据表 4 画出直方图,如图 13、图 14 所示。

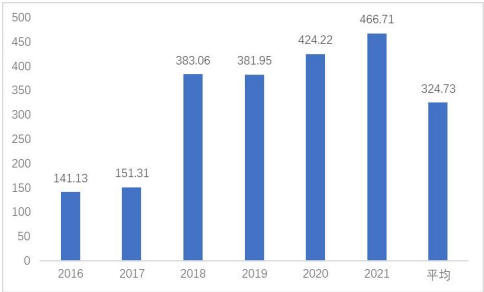


图 13 年总流量

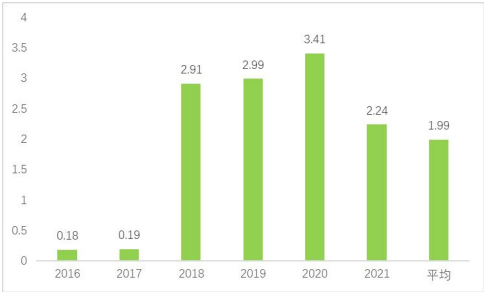


图 14 年总含沙量

5.2 问题（2）——研究水沙通量的变化规律

问题分析：水通量就是径流量（亿 m^3 ），沙通量就是排沙量（亿吨），研究它们的变化规律，需要从统计特征、季节性、周期性和突变性四个方面来入手。

建模思路：首先，研究水沙通量的季节性，包括月份指数、月份分布、统计特征等。其次，研究水沙通量的周期性，包括时序图、即三角函数等。第三，研究水、沙通量的突变点，使用非参数 Mann-Kendall 检验法来解决。

使用 python 软件整理数据，把每年的数据整理成以“月”为时间点的时间序列。

5.2.1 水沙通量的统计特征

对水流量、沙流量进行统计描述，结果如表 5 所示。从表中可以得出以下结论：

- (1) 各月水流量分化严重，最大流量 105.85 亿 m^3 ，最小流量 5.05 亿 m^3 。但平均波动幅度较小。
- (2) 各月排沙量分化不严重，最大流量 1.34 亿吨，最小流量 0 亿吨，但平均波动幅度较大。

表 5 水沙通量的特征

	平均值	标准差	最小值	最大值	极差	变异系数
水流量/亿 m^3	27.10	20.67	5.05	105.85	100.80	0.76
排沙量/亿吨	0.17	0.27	0.00	1.34	1.33	1.66

5.2.2 水沙通量的季节性

5.2.2.1 水沙通量的月份分布

计算水沙流量在各月分布，如表 6 所示。从表中可以得出以下结论：

- (1) 水流量在每年 7 月比例最大，占全年 15.36%。在 1 月占比最小，仅占 2.69%。
- (2) 排沙量在每年 7 月比例最大，占全年 31.17%。在 1 月占比最小，仅占 0.42%。

表 6 水沙通量的月份分布

	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月
水流量	2.69	3.51	6.11	7.79	8.33	11.14	15.36	10.02	11.64	12.81	6.2	4.4
沙流量	0.42	0.94	3.09	4.82	4.67	7.61	31.17	18.37	13.68	11.55	2.56	1.12

5.2.2.2 水沙通量的月份指数

计算水沙通量的月份指数，如表 7 所示。

表 7 水沙通量的月份指数

	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月
水流量	0.322	0.421	0.733	0.935	0.999	1.337	1.844	1.202	1.397	1.537	0.745	0.528
沙流量	0.050	0.112	0.371	0.579	0.560	0.913	3.740	2.205	1.642	1.386	0.307	0.134

根据表 1 画出图形，如图 15、图 16 所示。从图中可得出以下结论：

- (1) 水流量每年 6 月-10 月是旺季，每年 1 月-3 月、11 月、12 月是淡季。
- (2) 沙流量每年 7 月-10 月是旺季，每年 1 月-5 月、11 月、12 月是淡季。

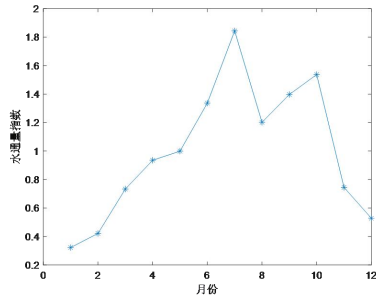


图 15 水通量的月份指数

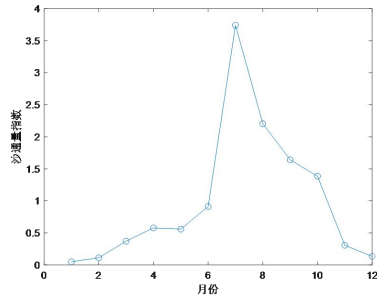


图 16 沙通量的月份指数

5.2.3 水沙通量的周期性

画出水沙通量的时序图,如图 17、图 18 所示。从图中可得出以下结论:

- (1) 水通量具有周期性,周期为 365 天,2018 年有突变现象,且呈上升趋势。
- (2) 沙通量具有周期性,周期为 365 天,2018 年有突变现象,且呈下降趋势。

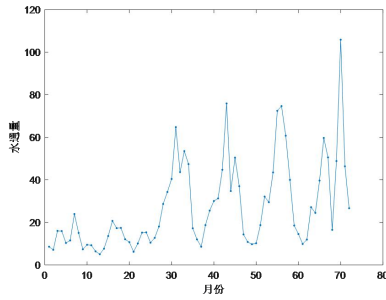


图 17 水通量的时序图

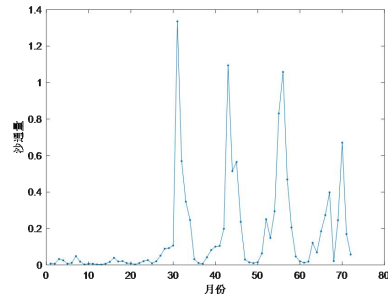


图 18 沙通量的时序图

设 p 表示水通量 (亿 m^3), t 表示时刻 (日), p 和 t 的函数关系为

$$p = A \sin\left(\frac{2\pi}{12}t + \phi\right) + C \quad (12)$$

根据实际情况,可直接从第 2 阶段 (2018-2021 年) 拟合,结果如下:

$$p = 22.76 \sin\left(\frac{2\pi}{12}t - 2.44\right) + 34.54 \quad (13)$$

同理,设 q 表示沙通量 (亿吨), t 表示时刻 (日), q 和 t 的函数关系为

$$q = 0.30 \sin\left(\frac{2\pi}{12}t - 2.43\right) + 0.24 \quad (14)$$

拟合效果检验如图 19、图 20 所示,从图中可知,拟合效果较好。

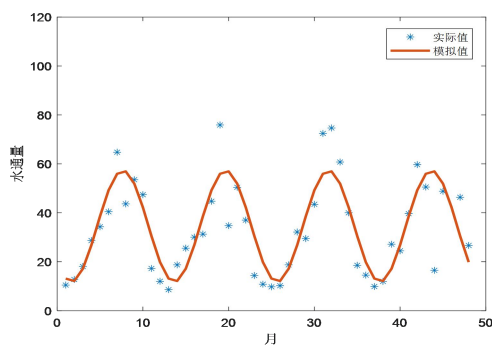


图 19 水通量的时序图

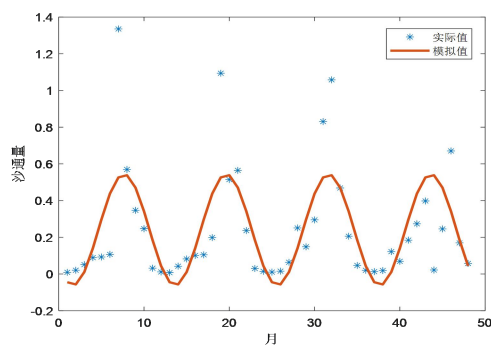


图 20 沙通量的时序图

5.2.4 水沙通量的突变性

5.2.4.1 水沙通量的变化趋势性

为判定水沙通量是否存在显著的上升或下降趋势,使用 M-K 检验法来判定。

M-K 检验法^[2]最初是由曼(H. B. Mann) 和肯德尔(M. G. Kendall) 提出了原理并发展了这一方法,是世界气象组织推荐的用于提取序列变化趋势的有效工具。M-K 检验法不受个别异常值的干扰,能够客观反映时间序列趋势,广泛应用于检验某一自然过程是处于自然波动亦或是存在确定的变化趋势,目前已经被广泛用于气候参数和水文序列的分析中。

M-K 法可以根据输出的两个序列(UF 和 UB) 明确突变的时段和区域。若 UF 值大于 0,则表明序列呈上升趋势,小于 0 则表明呈下降趋势。当它们超过临界置信水平直线时(显著性水平为 0.05 时,置信水平线为 ± 1.96),表明上升或下降趋势显著,超过临界线的范围确定为出现突变的时间区域。如果 UF 和 UB 两条曲线出现交点,且交点在临界线之间,那么交点对应的时刻便是突变开始的时间。

水通量的趋势检验结果如图 21 所示,从图中可得出以下结论:

UF 值从 $t = 23$ 开始大于 0,表明序列呈上升趋势。从 $t = 30$ 开始超过临界线(显著性水平为 0.05),表明上升趋势显著,此后为出现突变的时间区域。 UF 和 UB 两条曲线在 $t = 27$ 时出现交点,且交点在临界线之间,于是此时刻便是突变开始的时刻。

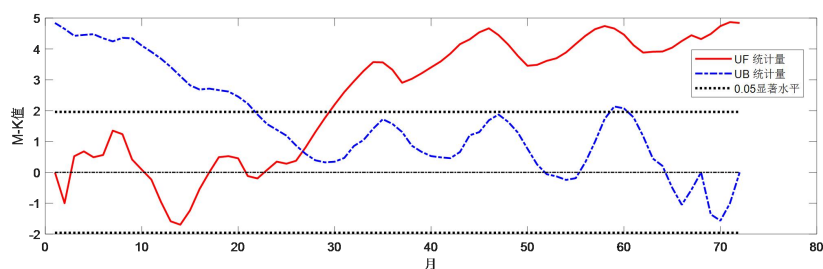


图 21 水通量的趋势检验

沙通量的趋势检验结果如图 22 所示,从图中可得出以下结论:

UF 值从 $t = 23$ 开始大于 0,表明序列呈上升趋势。从 $t = 30$ 开始超过临界线(显著性水平为 0.05 时,置信水平线为 ± 1.96),表明上升趋势显著,此后为出现突变的时间区域。 UF 和 UB 两条曲线在 $t = 27$ 时出现交点,且交点在临界线之间,于是此时刻便是突变开始的时刻。

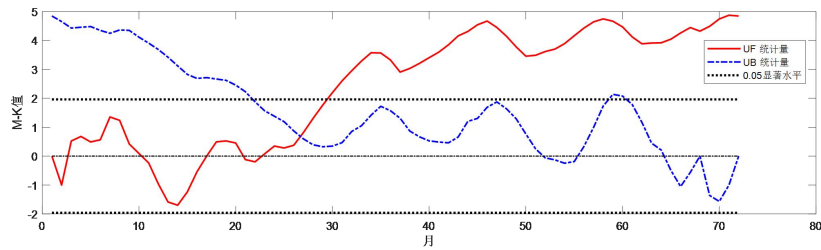


图 22 沙通量的趋势检验

5.2.4.2 水沙通量的突变点

为确定水沙通量在 6 年（72 个月）是否有转折性的变化和显著的转折点（变点），使用 Fisher 最优分割法来判定。

Fisher 最优分割法也称有序聚类分析^[3]，是多元统计分析中针对有序样品的一种统计分类方法。对有序样品分类，实际上就是要将这些样品组成的有序序列进行分段，每个分段为一类，这种分类称为分割。费歇尔(Fisher)曾给出了一个求最优分类的算法，其基本思想是：寻找一个分割使类内样品间的差异最小，而类间样品的差异最大。

使用 Fisher 最优分割法来判定是否存在突变点的方法是：把时间序列 x_t 按照时间点 t 从小到大排序成为一个有序序列，再使用最优分割法进行分类，就可以获得突变点。

水通量的突变点检验结果如图 23 所示，从图中可得出以下结论：在 $t = 27$ 时水流量被分为 2 类，说明此时为突变点，这个结论与 M-K 法检验法一致。

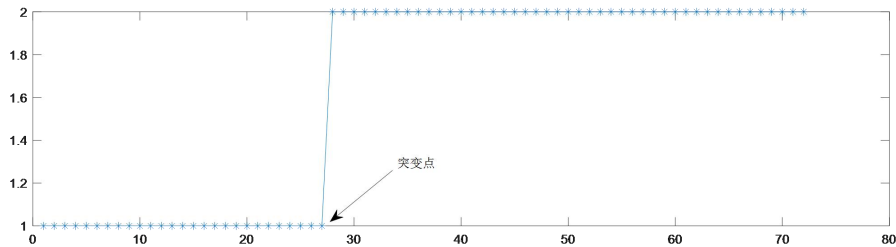


图 23 水通量的突变点

沙通量的突变点检验结果如图 24 所示，从图中可得出以下结论：在 $t = 30$ 时沙流量被分为 2 类，说明此时为突变点，这个结论与 M-K 法检验法一致。

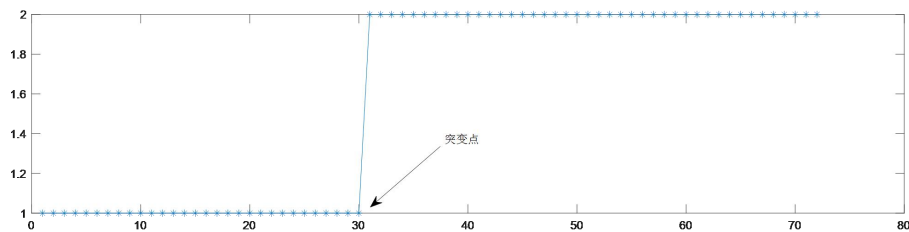


图 24 沙通量的突变点

5.2.5 水沙通量时间尺度的变化特征

为了详细而准确地寻找水沙通量的周期性和突变点，需要使用小波变换分析法^[4]。

小波变换是窗口大小固定但形状可改变、时间窗和频率窗都可改变的时-频局域化分析法，被誉为数学显微镜，它除了可以实现多分辨分析之外，在地球物理资料的处理中还可提取具有物理意义的最缓慢变化部分。小波变换分析在降水和水文资料的研究中得到了广泛的应用。

使用 MATLAB 软件作小波变换，再使用 Origin 软件画图。

画出水通量小波等值线图，如图 25 所示；画出水通量小波方差分析图，如图 26 所示。从图 25 可得出以下结论：

(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。

从图 2 可得出以下结论：水通量的主周期在 20 月附近，次周期在 55 月附近。

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。

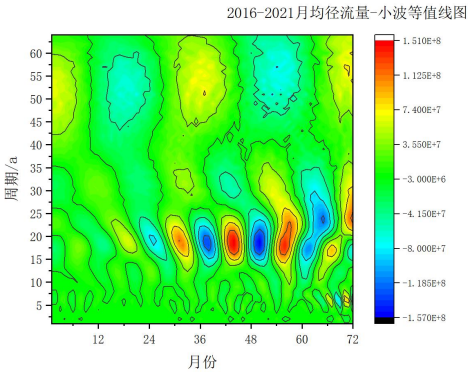


图 25 水通量小波等值线图

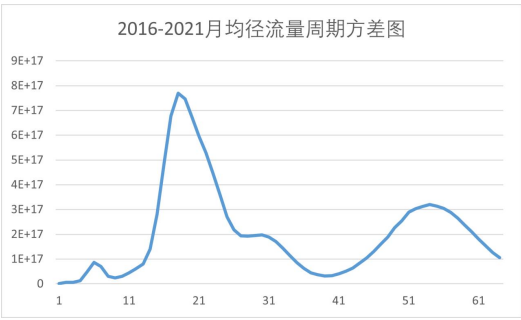


图 26 水通量小波方差分析图

画出差通量小波等值线图，如图 27 所示；画出差通量小波方差分析图，如图 28 所示。从图 27 可得出以下结论：

(1) 沙通量在 30 月以上的的时间尺度上振荡消失。

(2) 沙通量在 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，丰沙期与水通量相同，枯沙期也与水通量相同，突变点也与水通量相同。

从图 4 可得出以下结论：沙通量的主周期在 20 月附近。

综合而言，沙通量以 20 月为周期振荡，且振荡幅度很大，与水通量相同。

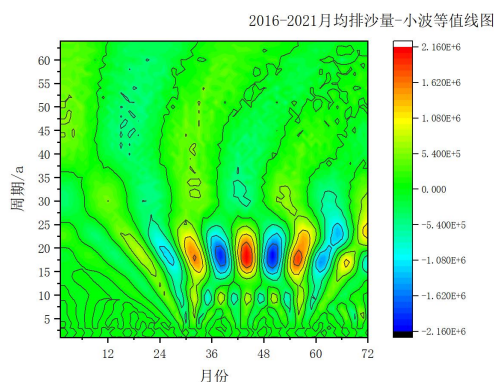


图 27 沙通量小波等值线图



图 28 沙通量小波方差分析图

5.3 问题（3）——预测水沙通量的未来趋势并制订采样监测方案

问题分析：题目要求预测未来 2 年水沙通量的变化趋势,并制订未来 2 年最优的采样监测方案。为预测水沙通量的变化趋势,使用 R/S 分析法分析水沙通量的分形特征和长期记忆过程,以判断其未来是否存在相同趋势、反转趋势或随机趋势。为预测未来 2 年的水沙通量数据,可建立傅里叶级数去逼近。为制订采样监测方案,使用小波分析法分析信号不同周期的时间演变规律,以掌握其丰枯旱涝情况,再根据这些情况把最优方案问题转化为背包问题,从而通过建立 0-1 规划模型来制订最优的采样监测方案。

建模思路：首先,使用 R/S 分析法研究水沙通量是否在未来 2 年存在反转趋势。其次,建立傅里叶级数,预测未来 2 年的水沙通量数据。第三,使用小波分析法掌握其丰枯旱涝情况,再建立 0-1 规划模型,获得最优的采样监测方案。仍然把水沙通量整理成以“月”为时间点的时间序列。

5.3.1 水沙通量的分形特征

为定量描述水沙通量的分形特征和长期记忆过程,探寻未来水沙通量的变化趋势,使用 R/S 分析法^[5]。

R/S 分析法也称为重标极差分析法 (Rescaled Range Analysis), 通常用于分析时间序列的分形特征和长期记忆过程。R / S 分析法属于非参数分析法, 它不必假定潜在的分布是正态高斯分布, 仅仅独立就可以。R/S 分析法被称为最具代表性的分形分析方法之一, 在分形理论中有着重要的应用, 目前已广泛应用于洪水、年径流序列以及气候变化的趋势分析中。

H 为 R/S 分析法的统计量, $0 \leq H \leq 1$ 。 H 能揭示出时间序列中的趋势性成分, 同时可以根据 H 的大小来判断趋势性成分的强弱。当 $H = 0.5$ 时, 说明时间序列为独立同分布的随机序列, 即现在的变化对未来没有影响; 当 $0 \leq H < 0.5$ 时, 表明该过程具有反持续性, 未来变化将与过去总体趋势相反, H 越接近于 0, 反持续性越强; 当 $0.5 \leq H \leq 1$ 时, 时间序列具有长程依赖性, 即未来与过去具有相同的变化趋势, H 越接近于 1, 持续性越强。

根据假设, 水通量、沙通量序列是独立序列。

以全时段 (2016-2021 年, 一共 72 个月) 为例, 水通量、沙通量 $\ln(R/S)$ 与 $\ln(n)$ 的散点图如图 29、图 30 所示, 从图中可知, $\ln(R/S)$ 与 $\ln(n)$ 二者的线性关系成立, 其 Hurst 指数有效。

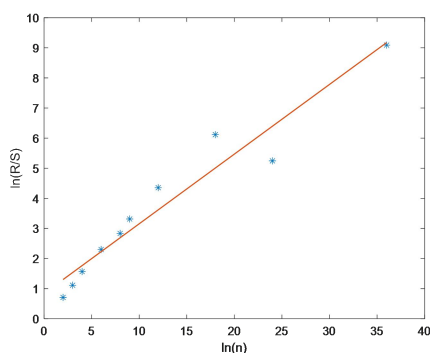


图 29 水通量 $\ln(R/S)$ 与 $\ln(n)$ 的散点图

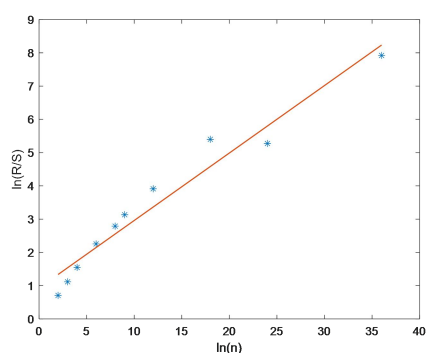


图 30 沙通量 $\ln(R/S)$ 与 $\ln(n)$ 的散点图

把全时段 2016-2021 年（1-72 日）分为 2 个阶段：2016-2017 年（1-24 日）和 2018-2021 年（25-72 日），第一阶段用于检验，第二阶段用于预测，结果如表 8 所示。

表 8 R/S 分析法的结果

阶段	2016-2021 年（1-72 日）		2016-2017 年（1-24 日）		2018-2021 年（25-72 日）	
水沙通量	水通量	沙通量	水通量	沙通量	水通量	沙通量
Hurst 指数	0.2315	0.2029	0.3317	0.2842	0.2261	0.2233
判定结果	趋势反转	趋势反转	趋势反转	趋势反转	趋势反转	趋势反转

从表 1 中可以得出以下结论：

（1）在模型检验阶段，首先，水通量 $H=0.3317 < 0.5$ ，预示水流量未来变化将与过去总体趋势相反，这与实际情况相符。其次，沙通量 $H=0.2842 < 0.5$ ，预示排沙量未来变化将与过去总体趋势相反，这也与实际情况相符。第三，沙通量 $H <$ 水通量 H ，说明沙通量的反转趋势比水通量更强，这也与实际情况相符。以上结果表明，R/S 分析法具有一定的可靠性，可以用于预测。

（2）在模型预测阶段，首先，水通量 $H=0.2261 < 0.5$ ，预示水流量未来变化将与过去总体趋势相反。其次，沙通量 $H=0.2233 < 0.5$ ，预示排沙量未来变化将与过去总体趋势相反。第三，沙通量 $H <$ 水通量 H ，说明沙通量的反转趋势比水通量更强。

5.3.2 水沙通量的预测

5.3.2.1 预测函数的建立

根据前文分析，水通量、沙通量都具有周期性，因此使用傅里叶级数去逼近。

设 $f(t)$ 表示 t 时刻的水沙通量，根据假设，其满足狄利赫里（Dirichlet）条件，那么 $f(t)$ 的傅里叶级数处处收敛，于是

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nt + b_n \sin nt) \quad (15)$$

由于水沙通量的周期为 12，于是水沙通量 $f(t)$ 的函数逼近解析式（取 10 阶）为

$$\begin{aligned}
y &= f(t) \\
&= \frac{a_0}{2} + \left(a_1 \cos \frac{\pi}{6} t + b_1 \sin \frac{\pi}{6} t \right) \\
&\quad + \left(a_2 \cos \frac{2\pi}{6} t + b_2 \sin \frac{2\pi}{6} t \right) + \dots + \left(a_{10} \cos \frac{10\pi}{6} t + b_{10} \sin \frac{10\pi}{6} t \right)
\end{aligned} \tag{16}$$

误差检验公式为

$$e = \sqrt{\frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2} \tag{17}$$

其中， \hat{y}_t, y_t 分别表示模拟值和实际值， n 为时间点个数。

5.3.2.2 预测函数的拟合和应用

取 2018-2020 年（36 个月）的水沙通量数据作拟合，再对 2021 年（12 个月）的水沙通量数据进行预测，接着作预测误差估计，最后对今后 2 年（2022-2023 年 24 个月）的水沙通量数据作预测。

对水通量、沙通量函数进行拟合，傅里叶系数的估计结果如表 9 所示。

表 9 水沙通量函数的傅里叶系数

序号	水通量	沙通量	序号	水通量	沙通量
a0	33.07	0.26	—	—	—
a1	-18.26	-0.25	a6	-1.06	-0.04
b1	-16.53	-0.26	b6	1.00	1.00
a2	-2.48	-0.02	a7	2.17	0.04
b2	2.21	1.09	b7	3.35	1.02
a3	-0.37	0.04	a8	-0.15	-0.04
b3	-0.03	0.95	b8	-0.67	0.96
a4	-0.15	-0.04	a9	-0.37	0.04
b4	4.97	1.04	b9	2.03	1.05
a5	1.62	0.04	a10	-1.55	-0.02
b5	0.77	0.98	b10	-1.45	0.91

拟合效果如图 31、图 32 所示，从图中可知，拟合效果较好。

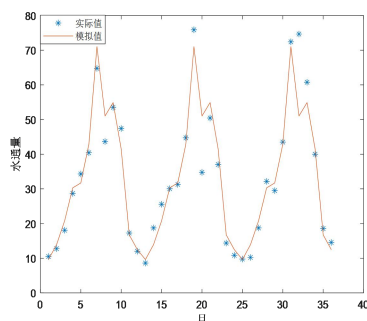


图 31 水通量的拟合效果比较

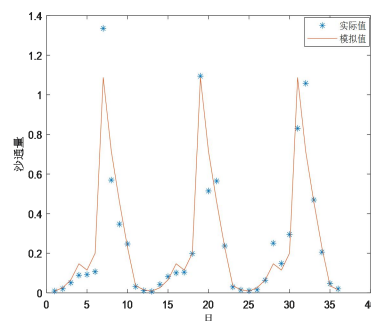


图 32 沙通量的拟合效果比较

预测误差估计结果如表 10 所示，从表中可知，预测误差较小。

表 10 预测误差的估计

	模拟误差	预测误差
水通量	0.1011	0.3205
沙通量	5.7116	24.6572

为了比较水通量、沙通量的预测效果，画出全部 6 年数据与未来 2 年数据的时序图，如图 33、图 34 所示，从图中可知，未来 2 年的预测值也具有相同的周期性，说明预测效果较好。

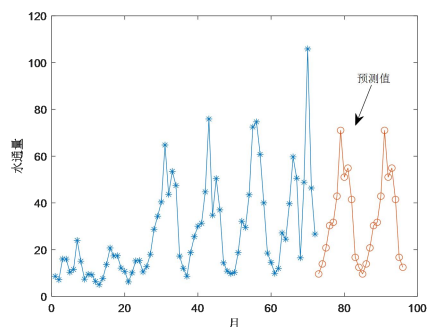


图 33 水通量过去值与未来值的比较

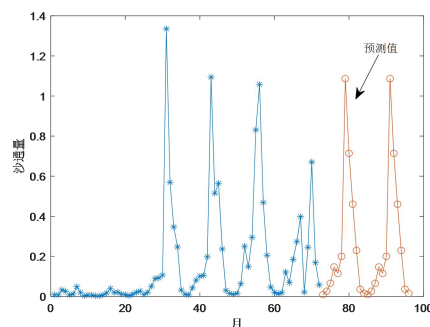


图 34 沙通量过去值与未来值的比较

5.3.3 制订水沙通量的最优采样监测方案

为制定最优的采样监测方案，基本思路是：在水沙通量变化平缓的时间点减少采样次数，相反，在水沙通量变化急剧的时间点增加采样次数。

5.3.3.1 未来 2 年水沙通量时间尺度的变化特征

画出未来 2 年水通量、沙通量小波等值线图，如图 35、图 36 所示，从图中可得出以下结论：

- (1) 水通量在未来 2 年（73~96 月）振荡减弱，只呈现了 1 次正负循环交替，振幅较平缓，有 3 个突变点，因此可减少监测次数。
- (2) 沙通量在未来 2 年（73~96 月）的变化趋势与水通量相同，也可减少监测次数。

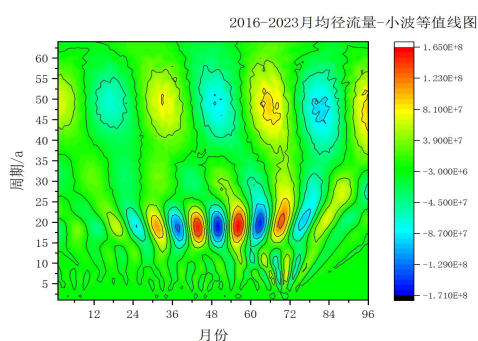


图 35 水通量小波等值线图

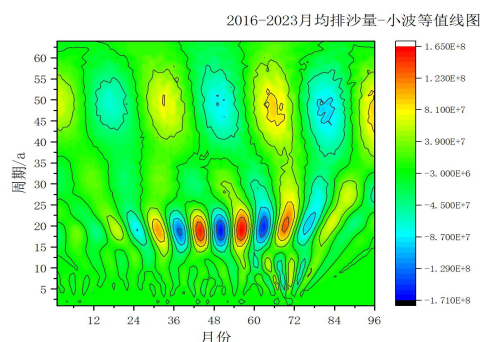


图 36 沙通量小波等值线图

5.3.3.2 未来 2 年水沙通量的采样监测方案

问题分析：基于对水沙通量的小波分析，要制订监测方案，所追求的目标是既要节约监测成本（监测次数），又要及时掌握水沙通量的动态变化。该问题可以转化为背包问题来解决，把水通量的动态变化信

息看作价值，把监测看作物品（有成本），在价值一定的情况下，如何监测才能使得成本最小。

建模思路：以时间点“月”为例，首先，对历年每月的水沙通量所携带的信息进行度量，并汇总成月均信息量。其次，对未来 2 年每月的水沙通量所携带的信息进行度量。第三，以月均信息量为约束条件，以总监测次数最小为目标，建立 0-1 规划模型。

下面开始建模。以时间点“月”为例（模型可以推广到周、日、小时等时间单位），以年为方案制订的周期。

设 x_t 表示 t 时刻的水沙通量（水通量或沙通量）， $t=1,2,\dots,12$ ，令

$$\begin{cases} y_{t+1} = |x_{t+1} - x_t|, & t=1,2,\dots,11 \\ y_1 = y_{12} \end{cases} \quad (18)$$

y_t 表示 t 时刻水沙通量所携带的动态信息，可以提前进行估计， $y_t \geq 0$ 。

设 \bar{y} 表示水沙通量所携带的动态信息的月平均值，可以用历史数据进行估计。

设 $z_t = 1$ 表示 t 时刻监测， $z_t = 0$ 表示 t 时刻不监测，则目标函数为

$$f = \sum_{t=1}^{12} z_t \quad (19)$$

目标函数求最小值，即

$$\min f \quad (20)$$

约束条件为

$$\frac{1}{12} \sum_{t=1}^{12} z_t y_t \geq \bar{y} \quad (21)$$

汇总得，

$$\begin{aligned} \min \quad & f = \sum_{t=1}^{12} z_t \\ \text{s.t.} \quad & \begin{cases} \frac{1}{12} \sum_{t=1}^{12} z_t y_t \geq \bar{y}, \\ z_t \in \{0,1\} \end{cases} \end{aligned} \quad (22)$$

下面开始参数估计。根据前面的分析，水通量与水通量的变化趋势相同，故仅制订水通量的监测方案即可。根据 2018~2021 年每月的水通量可以估计出 $\bar{y} = 10.8$ （亿 m^3 ）。再根据 2022 年、2023 年的水通量数据 x_t 计算出 y_t ，如表 11 所示。从表中可知，2022、2023 年的 y_t 是相同的，这是由于水通量周期性所致。

使用 LINGO 软件求解，计算结果如表 11 所示。

表 11 参数估计和最优解

月份	1	2	3	4	5	6	7	8	9	10	11	12
2022 年 差分/亿 m ³	4.2851	4.2851	6.8839	9.508	1.4074	11.1839	28.1436	20.0018	3.8734	13.4233	24.7519	4.2575
2022 年 最优解	1	1	1	1	0	1	1	1	1	1	1	1
2023 年 差分/亿 m ³	4.2851	4.2851	6.8839	9.508	1.4074	11.1839	28.1436	20.0018	3.8734	13.4233	24.7519	4.2575
2023 年 最优解	1	1	1	1	0	1	1	1	1	1	1	1

从表 11 可知，2022 年、2023 年均 5 月不监测，其余月份都要监测，监测次数减少了 1 次。

结果检验：由于 5 月份差分值最小（1.41），表明 5 月份与 4 月份的水通量变化不大，故可以不监测。

5.4 问题（4）——分析“调水调沙”的实际效果

问题分析：为了评估“调水调沙”的实际效果，可以比较“调水调沙”前后河底高程、水沙通量、水流量等指标的变化。

建模思路：首先，使用“微元法”计算不同时间点的平均河底高程，并进行比较。其次，使用“微元法”计算平均水深，在此基础上计算不同时间点的水通量，并进行比较。第三，计算不同时间点的沙通量，并进行比较。

根据假设，每年 6-7 月份“调水调沙”一次。

5.4.1 “调水调沙”对河底高程的影响

定义 1 河道断面左侧为起点桩位置，断面上测点距起点桩的距离即为起点距。

河道断面如图 37 所示。为了计算平均河底高程，需要使用“微元法”。

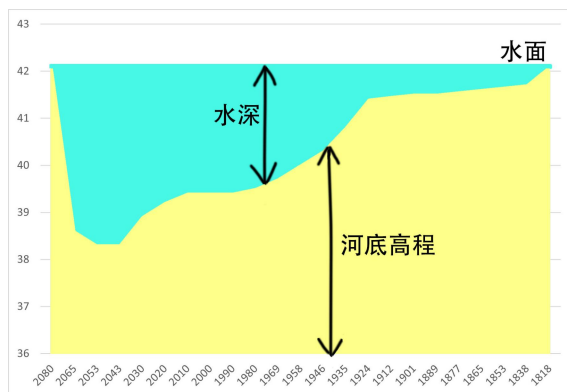


图 37 河道断面示意图

根据起点距的定义，使用“微元法”近似计算平均河底高程，步骤如下：

第 1 步，把数据点 (x_i, y_i) 按照起点距 x_i 从小到大排序， $i = 1, 2, \dots, n$ 。

第 2 步，计算相邻两个起点距的长度： $\Delta x_i = x_i - x_{i-1}$ ， $i = 2, 3, \dots, n$ 。

第 3 步，计算相邻两个河底高程的平均高度： $\bar{y}_i = 0.5(y_i + y_{i-1})$ ， $i = 2, 3, \dots, n$ 。

第 4 步，计算面积微元： $dS = \Delta x_i \bar{y}_i$ ， $i = 2, 3, \dots, n$ 。

第 5 步，计算总面积： $S = \sum_{i=2}^n \Delta x_i \bar{y}_i$ 。

第 6 步，计算平均河底高程： $h = \frac{S}{a}$ ，其中 a 为起点桩与终点桩之间的距离。

根据附件 2 的数据，计算结果如图 38 所示。从图中可得出以下结论：

(1) 每年的“调水调沙”成效显著，河底高程始终保持在 45m 左右，如果不调沙的话，必将出现上升现象。

(2) 2019 年“调水调沙”前后成效尤其显著，从 45m 下降到 44.77m，下降了 0.23m。

5.4.2 “调水调沙”对水通量的影响

根据水通量的计算公式，计算水通量（亿 m^3 ），结果如图 39、图 40 所示。从图 39 中可得出以下结论：

(1) 每年的“调水调沙”效果不佳，水通量呈现逐年下降趋势。

(2) 2018 年“调水调沙”前后成效尤其不佳，水通量从 9.46 亿 m^3 下降 5.49 亿 m^3 。下降了 72%。

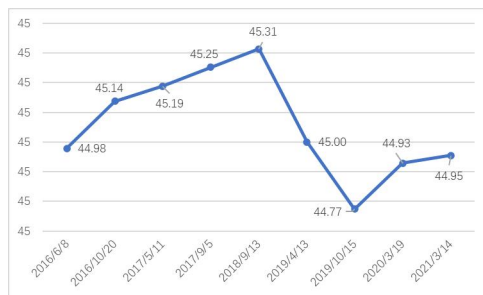


图 38 河底高程的变化情况

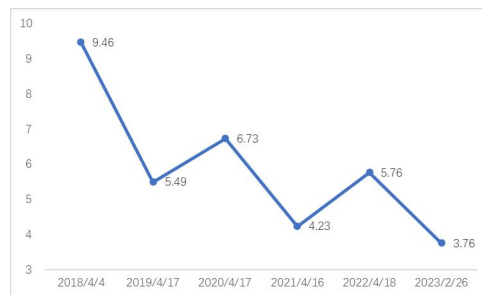


图 39 水通量的变化情况

从图 39 中可得出以下结论：

(1) 如果不“调水调沙”，那么水通量逐年下降趋势越来越严重。

(2) 2022 年水通量上升幅度尤其显著，上升了 6.4%，调沙效果尤其不好。

5.4.3 “调水调沙”对沙通量的影响

根据沙通量的计算公式，计算沙通量（亿吨），结果如图 40 所示。从图中可得出以下结论：

(1) “调水调沙”成效不佳，2021 年与 2020 年相比，沙通量下降了 45.4%。

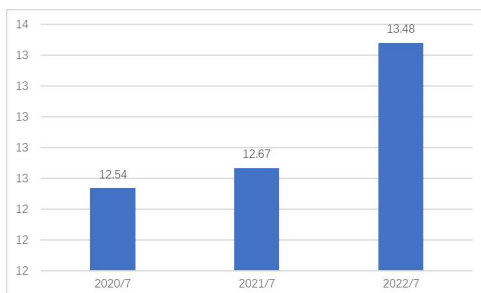


图 40 日均水通量的变化情况

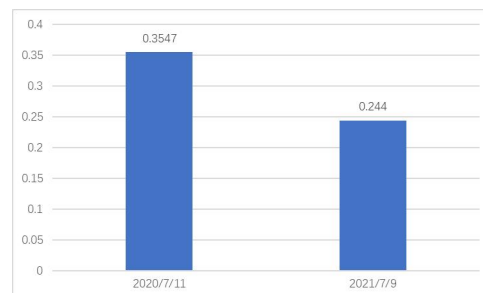


图 40 日均水通量的变化情况

5.4.4 未来 10 年河底高程的预测

根据附件 2 的数据，把 9 个时间点的平均河底高程估算出来，为了排除每年 6、7 月份“调水调沙”的影响，把本年度排沙后的河底高程减去上年度排沙前的河底高程，形成新的时间序列，具体情况如表 12

所示。

表 12 新的时间序列产生方式

时间点	1	2	3	4
两个时间点数据相减	2017/5/11-2016/10/20	2019/4/13-2018/9/13	2020/3/19-2019/10/15	2021/3/14-2020/3/19

新的时间序列如表 13 所示。

表 13 新的时间序列

时间点	1	2	3	4
上年度数据	45.14	45.31	44.77	44.93
本年度数据	45.19	45.00	44.93	44.95
数据相减	0.05	-0.31	0.15	0.03

由于新的时间序列只有 4 个时间点，故选择灰色模型 GM(1,1) 进行预测。

设 $\{z_t\}$ 为时间序列， z_t 表示 t 时刻河底高程 1 年的增量， $z_t \in (-\infty, +\infty)$ ， $t=1,2,3,4$ ，令

$$u_t = z_t + z_0, \quad t=1,2,3,4 \quad (23)$$

则 $u_t > 0$ ，这里取 $z_0 = 1$ ，于是建立 GM(1,1) 模型对未来 m 个时间点的数据进行预测，得到预测时间序列 $\{\hat{u}_t\}$ ， $t=1,2,\dots,4+m$ ，再还原得

$$\hat{z}_t = \hat{u}_t - z_0, \quad t=1,2,\dots,4+m \quad (24)$$

则 \hat{z}_t 表示河底高程 1 年的增量值，再还原为 t 时刻河底高程。

使用 MATLAB 编程计算，结果如表 14 所示。

表 14 模拟数据与预测数据

时间	1	2	3	4	5	6	7	8	9	10	11	12	13	14
数据	45.19	45.12	44.72	45.04	45.34	45.88	46.67	47.78	49.26	51.16	53.58	56.58	60.28	64.79

根据表 14，画出河底高程的演变趋势，如图 41 所示。从表 14 和图 41 可知，未来如果不进行“调水调沙”，那么河底高程将持续升高，10 年后将达到 64.79m。

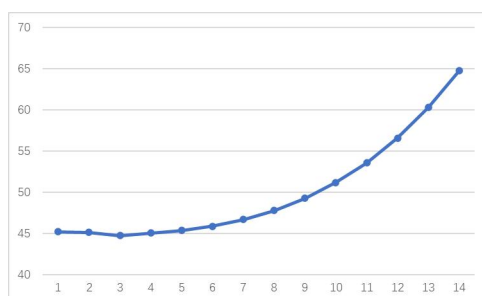


图 41 河底高程未来 10 年的变化情况

六、模型的评价和推广

6.1 模型优点

(1) 在分析含沙量与时间的关系时，建立了季节指数，得到了时间上的分布规律；在分析含沙量与

水位、水流量的关系时，建立了双对数回归方程，得到了相关性和弹性。

(2) 在分析水沙通量的季节性时，仍然建立了季节指数；在分析周期性时，建立了三角函数；在分析突变性时，分别使用 M-K 检验法和 Fisher 最优分割法进行交叉验证，可靠性强；为了详细寻找周期性和突变点，还使用了小波变换法，进一步验证了结论的正确性。

(3) 在分析水沙通量的趋势性时，使用 R/S 分析法；在预测未来 2 年的水沙通量时，建立了傅里叶级数，预测精度较高；在建立最优采样监测方案时巧妙转化为背包问题，存在最优解。

(4) 在计算曲线的平均高度时，使用了“微元法”。

6.2 模型缺点

(1) 在预测未来 2 年的水沙通量时，难以给出突变点的数值，导致在突变点的预测值偏低。

6.3 模型推广

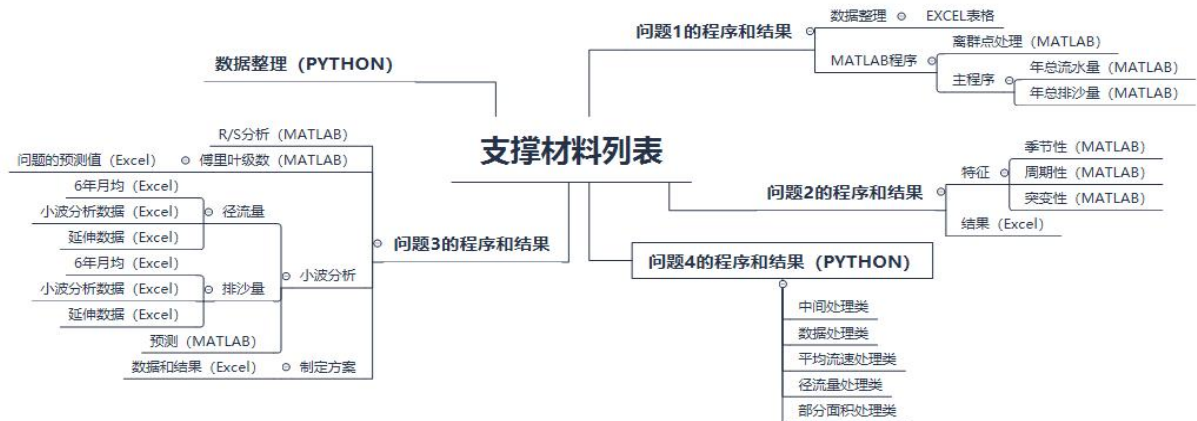
本文所使用的模型或方法可以推广到水文资料分析问题中，还可以推广到其它有关季节性、周期性、突变性等问题的分析中。

七、参考文献

- [1] 郑锦秀. 一元线性回归方程在大电流分流器测量中的应用[J]. 计测技术, 2009, (5): 17-19.
- [2] 靳明. 基于 M-K 检验方法的沈阳地区典型水文站近 53 年洪水要素演变分析[J]. 地下水, 2019, (2): 119-161.
- [3] 刘克琳等. "Fisher 最优分割法在汛期分期中的应用." 水利水电科技进展, 2017, (6): 19-23.
- [4] 王海龙. 基于小波分析的汀江水沙通量变化规律研究[J]. 泥沙研究, 2012.
- [5] 方国华. 线性回归法和 R/S 分析法在南平市年平均气温变化趋势分析中的应用[C]//中国水文科技新进展——2012 中国水文学学术讨论会论文集. 2012.
- [6] 王天华王平洋范明天. 用 0-1 规划求解馈线自动化规划问题[J]. 中国电机工程学报, 2000, 20(5): 54-58.
- [7] 王晓佳. 基于数据分析的预测理论与方法研究. Diss. 合肥工业大学, 2012.

附录

1. 支撑材料列表



2. 支撑材料内容

2.0 数据整理 (Python 软件)

```
from 格式转换类 import wk_format

for year in range(2016, 2022):

    wk_format(year)

from openpyxl import Workbook, load_workbook

def wk_format(year):

    new_wk: Workbook = Workbook()

    new_sheet = new_wk.active

    old_wk: Workbook = load_workbook("待整理数据表格.xlsx", data_only=True)

    for row in old_wk.get_sheet_by_name(str(year)):

        cell_list = []

        for cell in row:

            cell_list.append(cell.value)

        new_sheet.append(cell_list)

    new_wk.save(f"{year}.xlsx")

from pyecharts.charts import Bar

from pyecharts.options import *

from 统计工具类 import *

bar = Bar()

x = ['2016', '2017', '2018', '2019', '2020', '2021']

a, b = sum_utils()

y = a

for i in range(len(y)):

    y[i] = y[i] / 100000000

bar.add_xaxis(x)

bar.add_yaxis("年总水流量(亿立方米)", y, label_opts=LabelOpts(is_show=True, position="top"))

# 全局设置

bar.set_global_opts(
```

```

        legend_opts=LegendOpts(is_show=True, pos_top='2%'),
        title_opts=TitleOpts(title=f"2016-2021 各年总水流量", pos_left='center', pos_bottom='2%'))
bar.render("年总水流量图.html")

from pyecharts.charts import Bar
from pyecharts.options import *
from 测试代码.年总量计算类.统计工具类 import sum_utils
bar = Bar()
x = ['2016', '2017', '2018', '2019', '2020', '2021']
a, b = sum_utils()
y = b
bar.add_xaxis(x)
bar.add_yaxis("年总输沙量(亿吨)", y, label_opts=LabelOpts(is_show=True, position="top"))
for i in range(len(y)):
    y[i] = y[i] / 100000000
bar.set_global_opts(
    legend_opts=LegendOpts(is_show=True, pos_top='2%'),
    title_opts=TitleOpts(title=f"2016-2021 各年总输沙量", pos_left='center', pos_bottom='2%'))
bar.render("年总输沙量图.html")

import decimal
from math import floor
from openpyxl import Workbook, load_workbook

def get_sum(year):
    wk: Workbook = load_workbook(f'{year}年日平均计算.xlsx')
    sheet = wk.active
    sum1 = decimal.Decimal('0.0000')
    sum2 = decimal.Decimal('0.0000')
    for cell in sheet['G2:G361']:
        sum1 += decimal.Decimal(cell[0].value)
    for cell in sheet['H2:H361']:
        sum2 += decimal.Decimal(cell[0].value)
    return floor(sum1), floor(sum2)
def sum_utils():
    sum_list1 = []
    sum_list2 = []
    for year in range(2016, 2022):
        value1, value2 = get_sum(year)
        sum_list1.append(value1)
        sum_list2.append(value2)
    return sum_list1, sum_list2

from 日平均计算类 import day_start
from 月平均计算类 import month_start
from 折线图处理类 import line_start

```

```

for year in range(2016, 2022):
    day_start(f'{year}.xlsx')
    month_start(year)
    line_start(year)

from openpyxl import Workbook, load_workbook
import decimal

def day_start(file):
    wk = load_workbook(file, data_only=True)
    sheet = wk.active
    prev_d = 1
    prev_m = 1
    count = decimal.Decimal('0.0')
    water_level_sum = decimal.Decimal('0.00')
    flow_sum = decimal.Decimal('0.00')
    sand_content_sum = decimal.Decimal('0.00')
    new_wk = Workbook()
    new_sheet = new_wk.active
    new_sheet.append(['年', '月', '日', '平均水位', '平均流量', '平均含沙量', '径流量(立方米)', '输沙量(吨)'])
    flag = True
    for row in sheet:
        if flag:
            flag = False
            continue
        year = row[1].value
        month = row[2].value
        day = row[3].value
        water_level = row[4].value
        flow = row[5].value
        sand_content = row[6].value
        water_level = decimal.Decimal(water_level).quantize(decimal.Decimal('0.000'))
        flow = decimal.Decimal(flow).quantize(decimal.Decimal('0.000'))
        sand_content = decimal.Decimal(sand_content).quantize(decimal.Decimal('0.000'))
        if prev_d != day:
            water_level_sum = water_level_sum / count
            flow_sum = flow_sum / count
            sand_content_sum = sand_content_sum / count
            runoff_sum = flow_sum * 3600 * 24
            sediment_load_sum = sand_content_sum * flow_sum * 3600 * 24 / 1000
            new_sheet.append(
                [year, prev_m, prev_d, water_level_sum, flow_sum, sand_content_sum, runoff_sum, sediment_load_sum])
            prev_d = day
            prev_m = month
            water_level_sum = decimal.Decimal(water_level)
            flow_sum = decimal.Decimal(flow)
            sand_content_sum = decimal.Decimal(sand_content)
            count = decimal.Decimal('1.0')

```

```

        else:
            water_level_sum += decimal.Decimal(water_level)
            flow_sum += decimal.Decimal(flow)
            sand_content_sum += decimal.Decimal(sand_content)
            count += 1
        water_level_sum = water_level_sum / count
        flow_sum = flow_sum / count
        sand_content_sum = sand_content_sum / count
        runoff_sum = water_level_sum * 3600 * 24
        sediment_load_sum = sand_content_sum * flow_sum * 3600 * 24 / 1000

    new_sheet.append([year, month, day, water_level_sum, flow_sum, sand_content_sum, runoff_sum, sediment_load_sum])
    new_wk.save(f'{year}年日平均计算.xlsx')

from openpyxl import Workbook, load_workbook
from pycharts.charts import Line
import decimal
def month_start(year):
    wk: Workbook = load_workbook(f'{year}年日平均计算.xlsx')
    prev_d = 1
    prev_m = 1
    count = decimal.Decimal('0.0')
    new_wk = Workbook()
    sheet = wk.active
    new_sheet = new_wk.active
    new_sheet.append(['年', '月', '日', '平均径流量(立方米)', '平均输沙量(吨)'])
    runoff_sum = decimal.Decimal('0.0000')
    sediment_load_sum = decimal.Decimal('0.0000')
    flag = True
    for row in sheet:
        if flag:
            flag = False
            continue
        year = row[0].value
        month = row[1].value
        day = row[2].value
        runoff = decimal.Decimal(row[6].value)
        sediment_load = decimal.Decimal(row[7].value)
        if prev_m != month:
            runoff_sum = runoff_sum / count
            sediment_load_sum = sediment_load_sum / count
            new_sheet.append([year, prev_m, prev_d, runoff_sum, sediment_load_sum])
            prev_m = month
            prev_d = day
            runoff_sum = runoff
            sediment_load_sum = sediment_load
            count = decimal.Decimal('1.0')

```



```

        else:
            runoff_sum += runoff
            sediment_load_sum += sediment_load
            count += 1
runoff_sum = runoff_sum / count
sediment_load_sum = sediment_load_sum / count
new_sheet.append([year, month, day, runoff_sum, sediment_load_sum])
new_wk.save(f'{year}年月平均计算.xlsx')

from openpyxl import Workbook, load_workbook
from openpyxl.chart import LineChart, Reference

def line_start(year):

    wk: Workbook = load_workbook(f'{year}年月平均计算.xlsx')
    sheet = wk.active

    new_wk = Workbook()
    new_sheet = new_wk.active

    data_list = []
    for row in sheet:
        year = row[0].value
        month = row[1].value
        day = row[2].value
        runoff = row[3].value
        sediment_load = row[4].value
        data_list.append([year, month, day, runoff, sediment_load])
        new_sheet.append(data_list[-1])

    c1 = LineChart()
    c1.title = "月平均径流量图"
    c1.x_axis.title = "月份"
    c1.y_axis.title = "月平均径流量"
    data_c1 = Reference(sheet, min_col=4, min_row=1, max_col=4, max_row=13)
    c1.add_data(data_c1, titles_from_data=True)

    c2 = LineChart()
    c2.title = "月平均输沙量图"
    c2.x_axis.title = "月份"
    c2.y_axis.title = "月平均输沙量"
    data_c2 = Reference(sheet, min_col=5, min_row=1, max_col=5, max_row=13)
    c2.add_data(data_c2, titles_from_data=True)

    new_sheet.add_chart(c1, "G2")
    new_sheet.add_chart(c2, "P2")
    new_wk.save(f'{year}水流通量折线图.xlsx')

```

2.1 问题1的程序和结果

2.1.1 数据整理 (Excel)

2.1.2 程序 (MATLAB软件)

```
% 5.0 数据整理

clc,clear all

load data      % a1,a2,a3,a4,a5,a6=矩阵, 若干行*5 (5列: 年-月-日-水位-流量-含沙量)

%%

a{1}=a1;
a{2}=a2;
a{3}=a3;
a{4}=a4;
a{5}=a5;
a{6}=a6;

save data2 a

% 5.1.1 研究含沙量与时间的关系

clc,clear all

load data2      % a{}=数组, 6个矩阵, 若干行*5 (5列: 年-月-日-水位-流量-含沙量)

%% 按日汇总: 含沙量、水位、水流量

for i=1:6
    b=a{i};%年数据
    b(:,1)=[];
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1)=[];
            b5=mean(b3,1);
            c(k,:,j,i)=b5;% 数组, 30*3*12*6
        end
    end
end

%% 含沙量与水位、水流量

d=[];

for i=1:6
    for j=1:12
        c2=c(:, :, j, i);
        d=[d;c2];% 时间序列矩阵, 2160*3
    end
end
```

```

%% 画时序图，
d2=d(:,3);%含沙量
plot(d2,'*');
k1=1.5;
k2=3;
y=liqundian3(d2,k1,k2);%识别离群点
i=find(y==2);
d2(i)=[];
plot(d2,'*');

%% 月份指数
d3=[];
for i=1:6
    for j=1:12
        c2=c(:,j,i);
        n=size(c2,1);
        c3=[j*ones(n,1),c2];
        d3=[d3;c3];% 时间序列矩阵，2160*4(月份-水位-水流量-含沙量)
    end
end
d5=d3(:,[1,end]);
%% 计算月份指数
y=liqundian3(d5(:,2),k1,k2);%识别离群点
i=find(y==2);
d5(i,:)=[];
for i=1:12
    k=find(d5(:,1)==i);
    d6=d5(k,2);
    d7(i)=mean(d6);
end
d8=d7/mean(d7);%月份指数
plot(d8,'*-')

%% 计算季度指数
k=find(d5(:,1)>=1 & d5(:,1)<=3) ;
d6=d5(k,2);
d9(1)=mean(d6);

k=find(d5(:,1)>=4 & d5(:,1)<=6) ;
d6=d5(k,2);
d9(2)=mean(d6);

k=find(d5(:,1)>=7 & d5(:,1)<=9) ;
d6=d5(k,2);
d9(3)=mean(d6);

```

```

k=find(d5(:,1)>=10 & d5(:,1)<=12) ;
d6=d5(k,2);
d9(4)=mean(d6);

d10=d9/mean(d9);%月份指数
bar(d10);

% 5.1.2研究含沙量与水位、水流量的关系
clc,clear all
load data2      % a{}=数组，6个矩阵，若干行*5（5列：年-月-日-水位-流量-含沙量）

%% 按日汇总：含沙量、水位、水流量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1)=[];
            b5=mean(b3,1);
            c(k,:,j,i)=b5;% 数组，30*3*12*6
        end
    end
end

%% 含沙量与水位、水流量
d=[];
for i=1:6
    for j=1:12
        c2=c(:,:,j,i);
        d=[d;c2];% 时间序列矩阵，2160*3
    end
end

%% 剔除离群点
d2=d(:,3);%含沙量
plot(d2, '*');
k1=1.5;
k2=3;
y=liqundian3(d2,k1,k2);%识别离群点
i=find(y==2);
d(i,:)=[];

```

```

d2=d(:,2);%检验，是否存在离群点
plot(d2, '*');

%% 画散点图
subplot(1,3,1); plot(d(:,1),d(:,2), '. ');xlabel('水位');ylabel('水流量');
subplot(1,3,2); plot(d(:,1),d(:,3), 'o');xlabel('水位');ylabel('含沙量');
subplot(1,3,3); plot(d(:,2),d(:,3), '*');xlabel('水流量');ylabel('含沙量');
%% 建立回归方程
y=d(:,3);%含沙量
x=d(:,1); % 水位或水流量，分别对应2个回归方程
n=length(x);
[b, bint, r,rint,stats]=regress(log(y),[ones(n,1) log(x)],0.05);
z=[b bint],
stats,

%% 残差检验
r2=(r-mean(r))/sqrt(stats(4));
figure, histfit(r2);
y2=(y-mean(y))/std(y);
figure, plot(y2,r2, '*');
% 5.1.3估算年总水流量和年总排沙量
clc,clear all
load data2 % a{}=数组，6个矩阵，若干行*5（5列：年-月-日-水位-流量-含沙量）

%% 按日汇总：含沙量、水位、水流量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1)=[];
            b5=mean(b3,1);
            c(k,:,j,i)=b5;% 数组，30*3*12*6
        end
    end
end

%% 含沙量与水位、水流量
for i=1:6
    c2=c(:, :, :, i);
    c3=mean(mean(c2,1),3);

```

```

d=c3(:,2);

d2=365*24*3600/10^8*d ;

d3(i)=d2;    %%年总水流量, 亿立方米

end

d4=mean(d3);

d5=[d3,d4];

%% 含沙量与水位、水流量

d9=0;

for i=1:6

    c2=c(:, :, :, i);

    for j=1:12

        c3=c2(:, :, j);

        for k=1:30

            c4=c3(k, :);

            d8=c4(2)*c4(3)*24*3600/1000;%吨

            d9=d9+d8;

        end

    end

    d10(i)=d9/10^8;

end

d11=mean(d10);

d12=[d10,d11];

%% 输出

e=[d5;d12]

%% 剔除离群点

d2=d(:,3);%含沙量

plot(d2, '*');

k1=1.5;

k2=3;

y=liqundian3(d2,k1,k2);%识别离群点

i=find(y==2);

d(i,:)=[];

d2=d(:,2);%检验, 是否存在离群点

plot(d2, '*');

%% 年总水流量

d3=d(:,2);

d4=mean(d3)*365*24*3600/100000000    %亿立方米

%% 年总排沙量

d5=d(:,3);

d6=mean(d4)*d4/1000    %吨

```



```

% 5.1.1研究含沙量与时间的关系

clc,clear all

load data2      % a{}=数组，6个矩阵，若干行*5（5列：年-月-日-水位-流量-含沙量）

%% 按日汇总：含沙量、水位、水流量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1)=[];
            b5=mean(b3,1);
            c(k,:,j,i)=b5;% 数组，30*3*12*6
        end
    end
end

%%
d=[];
for i=1:2 % 拟合三角函数（2016-2017年）
    for j=1:12
        c2=c(:, :, j, i);
        d=[d;c2];% 时间序列矩阵，若干日*3
    end
end

n=size(d,1);
d2=[1:n]',d(:,end)];% 矩阵，若干行*2（2列：日序号-含沙量）

%% 剔除离群点
d3=sortrows(d2,2);
d4=flipud(d3);
d4(1:3,:)=[];%有3个离群点
d5=sortrows(d4,1);      %剔除离群点以后的矩阵，若干行*2（2列：日序号-含沙量）
d6=d5(:,2);%检验，是否存在离群点
figure, plot(d6,'*');

%% 拟合三角函数（2016-2017年）
t=d5(:,1);
y=d5(:,2);
f=inline('p(1)*sin(2*pi/365*t+p(2))+p(3)','p','t');
p=lsqcurvefit(f,[0.3,-0.3,1],t,y)

```

```

%% 拟合检验

n=size(d5,1);
t2=1:n;
f2=p(1)*sin(2*pi/365*t+p(2))+p(3);
figure, plot(t,y,'.',t2,f2,'-');

% 5.1.1 研究含沙量与时间的关系

clc,clear all

load data2 % a{}=数组, 6个矩阵, 若干行*5 (5列: 年-月-日-水位-流量-含沙量)

%% 按日汇总: 含沙量、水位、水流量

for i=1:6
    b=a{i};%年数据
    b(:,1)=[];
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1)=[];
            b5=mean(b3,1);
            c(k,:,j,i)=b5;% 数组, 30*3*12*6
        end
    end
end

end

%%
d=[];
for i=3:6 % 拟合三角函数 (2018-2021年)
    for j=1:12
        c2=c(:, :, j, i);
        d=[d;c2];% 时间序列矩阵, 若干日*3
    end
end

end

n=size(d,1);
d2=[1:n]',d(:,end)];% 矩阵, 若干行*2 (2列: 日序号-含沙量)

%% 剔除离群点
d3=sortrows(d2,2);
d4=flipud(d3);
d4(1:7,:)=[];%有7个离群点
d5=sortrows(d4,1); %剔除离群点以后的矩阵, 若干行*2 (2列: 日序号-含沙量)
d6=d5(:,2);%检验, 是否存在离群点
figure, plot(d6,'*');

```

```

%% 拟合三角函数（2016-2017年）

t=d5(:,1);
y=d5(:,2);

f=inline('p(1)*sin(2*pi/365*t+p(2))+p(3)','p','t');
p=lsqcurvefit(f,[4,-2,5],t,y)

%% 拟合检验

n=size(d5,1);

t2=1:n;

f2=p(1)*sin(2*pi/365*t+p(2))+p(3);

figure, plot(t,y,'.',t2,f2,'-');

```

2.2 问题2的程序和结果

2.2.1 季节性（MATLAB）

```

% 5.2.1 水、沙通量的季节性

clc,clear all

load data2 % a{}=数组, 6个矩阵, 若干行*6 (6列: 年-月-日-水位-流量-含沙量)

%% 按月汇总, 水通量, 沙通量

for i=1:6
    b=a{i};%年数据
    b(:,1)=[];% 若干行*5 (5列: 月-日-水位-流量-含沙量)
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];% 若干行*4 (4列: 日-水位-流量-含沙量)
        c2=0;
        d2=0;
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1:2)=[];% 若干行*2 (2列: 流量-含沙量)
            b5=mean(b3,1);% 1*2 (2列: 流量-含沙量), 日均数据
            % 计算水通量
            c=b5(1)*24*3600/10^8;%亿立方米
            c2=c2+c;
            % 计算沙通量
            d=b5(2)*24*3600/1000/10^8;%亿吨
            d2=d2+d;
        end
        c3(j,:)= [j,c2];% 数组, 12*2列 (月-水通量)
        d3(j,:)= [j,d2];% 数组, 12*2列 (月-沙通量)
    end
    c4(:, :, i)=c3;%亿立方米, 12*2*6, 水通量
    d4(:, :, i)=d3;%亿吨, 12*2*6, 沙通量
end

%% 计算月份指数(水通量)

```

```

c5=mean(c4,3);
c6=c5(:,2);
c7=c6/mean(c6);% 水通量的月份指数
subplot(1,2,1); plot(c7,'*-');      xlabel('月份');ylabel('水通量指数');
c8=c7/sum(c7)

%% 计算月份指数(沙通量)
d5=mean(d4,3);
d6=d5(:,2);
d7=d6/mean(d6);% 水通量的月份指数
subplot(1,2,2); plot(d7,'o-');      xlabel('月份');ylabel('沙通量指数');
d8=d7/sum(d7)

%% 输出：季节指数
e=[c7,d7]
e2=[c8,d8]

% 5.2.1水、沙通量的季节性
clc,clear all
load data2      % a{}=数组，6个矩阵，若干行*6（6列：年-月-日-水位-流量-含沙量）

%% 按月汇总，水通量，沙通量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];% 若干行*5（5列：月-日-水位-流量-含沙量）
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];% 若干行*4（4列：日-水位-流量-含沙量）
        c2=0;
        d2=0;
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1:2)=[];% 若干行*2（2列：流量-含沙量）
            b5=mean(b3,1);% 1*2（2列：流量-含沙量），日均数据
            % 计算水通量
            c=b5(1)*24*3600/10^8;%亿立方米
            c2=c2+c;
            % 计算沙通量
            d=b5(2)*b5(1)*24*3600/1000/10^8;%亿吨
            d2=d2+d;
        end
        c3(j,:)=[j,c2];% 数组，12*2列（月-水通量）
        d3(j,:)=[j,d2];% 数组，12*2列（月-沙通量）
    end
    c4(:, :, i)=c3;%亿立方米，12*2*6，水通量

```

```

        d4(:, :, i)=d3;%亿吨, 12*2*6, 沙通量
    end

%% 描述分析
c6=[];    d6=[];
for i=1:6
    c5=c4(:, :, i);
    c5=c5(:, 2);
    c6=[c6; c5];

    %
    d5=d4(:, :, i);
    d5=d5(:, 2);
    d6=[d6; d5];
end

c7=[mean(c6), std(c6), min(c6), max(c6), range(c6), std(c6)/mean(c6)];
d7=[mean(d6), std(d6), min(d6), max(d6), range(d6), std(d6)/mean(d6)];

%% 输出
e=[c7; d7]

```

2.2.2 周期性 (MATLAB)

```

% 5.2.2水、沙通量的周期性
clc, clear all

load data2      % a{}=数组, 6个矩阵, 若干行*6 (6列: 年-月-日-水位-流量-含沙量)

%% 按月汇总, 水通量, 沙通量
for i=1:6
    b=a{i};%年数据
    b(:, 1)=[];% 若干行*5 (5列: 月-日-水位-流量-含沙量)
    for j=1:12
        p=find(b(:, 1)==j);
        b2=b(p, :);%月数据
        b2(:, 1)=[];% 若干行*4 (4列: 日-水位-流量-含沙量)
        c2=0;
        d2=0;
        for k=1:30
            q=find(b2(:, 1)==k);
            b3=b2(q, :);%日数据
            b3(:, 1:2)=[];% 若干行*2 (2列: 流量-含沙量)
            b5=mean(b3, 1);% 1*2 (2列: 流量-含沙量), 日均数据

            % 计算水通量
            c=b5(1)*24*3600/10^8;%亿立方米
            c2=c2+c;

            % 计算沙通量
            d=b5(2)*b5(1)*24*3600/1000/10^8;%亿吨
            d2=d2+d;
        end
    end
end

```

```

        c3(j,:)= [j,c2];% 数组，12*2列（月-水通量）
        d3(j,:)= [j,d2];% 数组，12*2列（月-沙通量）
    end

    c4(:, :, i)=c3;%亿立方米，12*2*6，水通量
    d4(:, :, i)=d3;%亿吨，12*2*6，沙通量
end

%% 周期性(水通量)
c6=[];
d6=[];
for i=1:6
    c5=c4(:, :, i);
    c5=c5(:, 2);
    c6=[c6;c5];
    %
    d5=d4(:, :, i);
    d5=d5(:, 2);
    d6=[d6;d5];
end

subplot(1,2,1); plot(c6,'.-'); xlabel('月份');ylabel('水通量');
subplot(1,2,2); plot(d6,'.-'); xlabel('月份');ylabel('沙通量');

% 5.2.2水、沙通量的周期性
clc,clear all
load data2 % a{}=数组，6个矩阵，若干行*6（6列：年-月-日-水位-流量-含沙量）

%% 按月汇总，水通量，沙通量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];% 若干行*5（5列：月-日-水位-流量-含沙量）
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];% 若干行*4（4列：日-水位-流量-含沙量）
        c2=0;
        d2=0;
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1:2)=[];% 若干行*2（2列：流量-含沙量）
            b5=mean(b3,1);% 1*2（2列：流量-含沙量），日均数据
            % 计算水通量
            c=b5(1)*24*3600/10^8;%亿立方米
            c2=c2+c;
            % 计算沙通量
            d=b5(2)*b5(1)*24*3600/1000/10^8;%亿吨
            d2=d2+d;

```

```

end

c3(j,:)= [j,c2];% 数组, 12*2列 (月-水通量)

d3(j,:)= [j,d2];% 数组, 12*2列 (月-沙通量)

end

c4(:, :, i)=c3;%亿立方米, 12*2*6, 水通量

d4(:, :, i)=d3;%亿吨, 12*2*6, 沙通量

end

%% 周期性(水通量、沙通量)

c6=[];

d6=[];

for i=3:6 % (2018-2021年)

    c5=c4(:, :, i);

    c5=c5(:, 2);

    c6=[c6;c5];

    %

    d5=d4(:, :, i);

    d5=d5(:, 2);

    d6=[d6;d5];

end

n=length(c6);

t=1:n;

%% 剔除离群点

e=[t', c6]; % (水通量、或者沙通量)

t=e(:, 1);

y=e(:, 2);

plot(t, y, 'b*');

f=inline('p(1)*sin(2*pi/12*t+p(2))+p(3)', 'p', 't');

p=lsqcurvefit(f, [22, -2, 34], t, y)

%% 拟合检验

f2=p(1)*sin(2*pi/12*t+p(2))+p(3);

plot(t, y, 'b*', t, f2, 'r-');

```

2.2.3 突变性 (MATLAB)

```

% 5.2.4.1 水沙通量的变化趋势性

clc, clear all

load data2 % a{}=数组, 6个矩阵, 若干行*6 (6列: 年-月-日-水位-流量-含沙量)

%% 按月汇总, 水通量, 沙通量

for i=1:6

    b=a{i};%年数据

    b(:, 1)=[];% 若干行*5 (5列: 月-日-水位-流量-含沙量)

    for j=1:12

        p=find(b(:, 1)==j);

        b2=b(p, :);%月数据

        b2(:, 1)=[];% 若干行*4 (4列: 日-水位-流量-含沙量)

```

```

c2=0;
d2=0;
for k=1:30
    q=find(b2(:,1)==k);
    b3=b2(q,:);%日数据
    b3(:,1:2)=[]; % 若干行*2 (2列: 流量-含沙量)
    b5=mean(b3,1);% 1*2 (2列: 流量-含沙量), 日均数据
    % 计算水通量
    c=b5(1)*24*3600/10^8;%亿立方米
    c2=c2+c;
    % 计算沙通量
    d=b5(2)*b5(1)*24*3600/1000/10^8;%亿吨
    d2=d2+d;
end
c3(j,:)=[j,c2];% 数组, 12*2列 (月-水通量)
d3(j,:)=[j,d2];% 数组, 12*2列 (月-沙通量)
end
c4(:, :, i)=c3;%亿立方米, 12*2*6, 水通量
d4(:, :, i)=d3;%亿吨, 12*2*6, 沙通量
end

%% 趋势性
c6=[];
d6=[];
for i=1:6
    c5=c4(:, :, i);
    c5=c5(:, 2);
    c6=[c6;c5];
    %
    d5=d4(:, :, i);
    d5=d5(:, 2);
    d6=[d6;d5];
end
n=length(c6); % 水通量, 或沙通量, 在这里输入变量!!!!!!!!!!!!!!
t=[1:n]';
e=[t, c6];
y3=MKtest(e)
% 5.2.4.2水沙通量的突变点
clc, clear all
load data2 % a{}=数组, 6个矩阵, 若干行*6 (6列: 年-月-日-水位-流量-含沙量)

%% 按月汇总, 水通量, 沙通量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];% 若干行*5 (5列: 月-日-水位-流量-含沙量)
    for j=1:12

```



```

p=find(b(:,1)==j);
b2=b(p,:);%月数据
b2(:,1)=[];% 若干行*4 (4列: 日-水位-流量-含沙量)
c2=0;
d2=0;
for k=1:30
    q=find(b2(:,1)==k);
    b3=b2(q,:);%日数据
    b3(:,1:2)=[];% 若干行*2 (2列: 流量-含沙量)
    b5=mean(b3,1);% 1*2 (2列: 流量-含沙量), 日均数据
    % 计算水通量
    c=b5(1)*24*3600/10^8;%亿立方米
    c2=c2+c;
    % 计算沙通量
    d=b5(2)*b5(1)*24*3600/1000/10^8;%亿吨
    d2=d2+d;
end
c3(j,:)=[j,c2];% 数组, 12*2列 (月-水通量)
d3(j,:)=[j,d2];% 数组, 12*2列 (月-沙通量)
end
c4(:,:,i)=c3;%亿立方米, 12*2*6, 水通量
d4(:,:,i)=d3;%亿吨, 12*2*6, 沙通量
end

%% 趋势性
c6=[];
d6=[];
for i=1:6
    c5=c4(:,:,i);
    c5=c5(:,2);
    c6=[c6;c5];
    %
    d5=d4(:,:,i);
    d5=d5(:,2);
    d6=[d6;d5];
end
n=length(c6); % 水通量, 或沙通量, 在这里输入变量!!!!!!!!!!!!!!
t=[1:n]';

%% 最优分割法
e=c6; % 水通量, 或沙通量, 在这里输入变量!!!!!!!!!!!!!!
e=e';
m=2; % 类别个数
y=fenlei(e,m);
e2=[t,y']
plot(t,y,'*-')

```

```

%分类函数

function y=fenlei(a,m)

% a是一个行向量（从小到大排序或者从大到小排序），
% m=6是预先估计的分类个数，然后根据输出图形的结果再调整
% y=行向量，每个样品的类别

%%

[x,y,z]=sunshi(a); % z是分点矩阵n*n(例如26*26)

n=size(z,1);
a=zeros(1,n);
y=z(n,m);
a(1,y:n)=m;
t=y-1;
for j=1:(m-1)
    i=m-j;
    if i>1
        y=z(t,i);
        a(1,y:t)=i;
        t=y-1;
    else
        a(1,1:t)=1;
    end
end
y=a;

% 最优分割法—计算直径的函数

function y=zhijing(a)

% a是一个行向量（本文中是相对误差）

%%

b=size(a);
e=zeros(b(2));
for i=1:(b(2)-1)
    for j=(i+1):b(2)
        c=a(1,i:j);
        f=size(c);
        d=mean(c);
        for k=1:f(2)
            e(i,j)=e(i,j)+(c(1,k)-d)^2;
        end
    end
end
y=e;

function [x,y,z]=sunshi(a)

% a是一个行向量（从小到大排序）
% x=直径D(i,j)矩阵，是b阶方阵
% y=损失函数值L[b*(1,k)]矩阵，是b阶方阵
% z=分类位置点j 矩阵，是b阶方阵

```

```

%%
d=zhijing(a);% d是直径矩阵，zhijing() 是自编的求直径的函数
b=length(a);
c=zeros(b);
e=zeros(b);
f=zeros(1,b);
k=2;
for i=(k+1):b
    for j=k:i
        f(1,j)=d(1,j-1)+d(j,i) ;
    end
    g=nonzeros(f);
    c(i,k)=min(g);
    for j=1:b
        if f(1,j)==min(g)
            e(i,k)= j;
        end
    end
end
h=zeros(1,b);
for k=3:b-1
    for i=(k+1):b
        for j=k:i
            h(1,j)=c(j-1,k-1)+d(j,i) ;
        end
        g=nonzeros(h);
        c(i,k)= min(g);
        for j=1:b
            if h(1,j)==min(g)
                e(i,k)= j;
            end
        end
    end
end
x=d; %输出直径D(i, j)矩阵, 是b阶方阵
y=c; %输出损失函数值L[b*(1, k)]矩阵，是b阶方阵
z=e; %输出分类位置点j 矩阵，是b阶方阵
x1=2:b-1;
y1=c(b, 2:b-1);
plot(x1,y1, '-*') %画出损失函数关于分类数的变化图像
% M-K检验法(Mann-Kendall检验法)
% clc,clear all
% load data % a=矩阵，若干行*2列（时间点-平均温度序列）
function y3=MKtest(a)
%%
y=a(:,2); %平均温度序列

```

```

Sk=zeros(size(y));      %定义累计量序列Sk，长度=y，初始值=0，Sk(1)=0
UFk=zeros(size(y));     %定义统计量UFk，长度=y，初始值=0，UFk(1)=0
s=0;                    %定义Sk序列的元素s
for i = 2:length(y)
    for j=1:i
        if y(i) > y(j)
            s=s+1;
        else
            s=s+0;
        end
    end
    Sk(i)=s;
    Var=i*(i-1)*(2*i+5)/72;%Sk(i) 方差，见式(3)
    E=i*(i-1)/4;%Sk(i)的均值，见式(3)
    UFk(i)=(Sk(i)-E)/sqrt(Var);%正序列UF值，见式(2)
end

%%
Sk2=zeros(size(y));%定义逆序累计量序列Sk2，长度=y，初始值=0，Sk(2)=0
UBk=zeros(size(y));%定义逆序统计量UBk，长度=y，初始值=0，UBk(1)=0
s=0;
y2=flipud(y);%按时间序列逆转平均温度序列
for i=2:length(y2)
    for j=1:i
        if y2(i)>y2(j)
            s=s+1;
        else
            s=s+0;
        end
    end
    Sk2(i)=s;
    E=i*(i-1)/4; %均值
    Var=i*(i-1)*(2*i+5)/72; %方差
    UBk(i)=0-(Sk2(i)-E)/sqrt(Var);
end
UBk2=flipud(UBk);%逆序列UB值

%%
x=a(:,1);      %年份序列
n=length(x);   %年份序列的长度
figure,        %画图
set(gcf,'unit','centimeters','position',[3 5 12 6]) % 设置图形的位置及大小
set(gca,'Position',[0.15 0.18 0.8 0.78]); % 设置图片比例大小
plot(x,UFk,'r-','linewidth',1.5);          %画UF线
hold on
plot(x,UBk2,'b-','linewidth',1.5);          %画UB线
plot(x,1.96*ones(n,1),'k:','linewidth',2);
plot(x,-1.96*ones(n,1),'k:','linewidth',2);

```

```

plot(x,0*ones(n,1),'k-','linewidth',1) ; %画0水平线
xlabel('月');
ylabel('M-K值');
legend('UF 统计量','UB 统计量','0.05显著水平') ; % 设置图例

%%

y3=[x,UFk,UBk2];

```

2.3 问题3的结果

2.3.1R/S 分析

```

% 5.2.2水、沙通量的周期性

clc,clear all

load data2 % a{}=数组, 6个矩阵, 若干行*6 (6列: 年-月-日-水位-流量-含沙量)

%% 按月汇总, 水通量, 沙通量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];% 若干行*5 (5列: 月-日-水位-流量-含沙量)
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];% 若干行*4 (4列: 日-水位-流量-含沙量)
        c2=0;
        d2=0;
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1:2)=[];% 若干行*2 (2列: 流量-含沙量)
            b5=mean(b3,1);% 1*2 (2列: 流量-含沙量), 日均数据
            % 计算水通量
            c=b5(1)*24*3600/10^8;%亿立方米
            c2=c2+c;
            % 计算沙通量
            d=b5(2)*b5(1)*24*3600/1000/10^8;%亿吨
            d2=d2+d;
        end
        c3(j,:)=[j,c2];% 数组, 12*2列 (月-水通量)
        d3(j,:)=[j,d2];% 数组, 12*2列 (月-沙通量)
    end
    c4(:, :, i)=c3;%亿立方米, 12*2*6, 水通量
    d4(:, :, i)=d3;%亿吨, 12*2*6, 沙通量
end

%% 建立时间序列
c6=[];
d6=[];
for i=1:6
    c5=c4(:, :, i);

```

```

c5=c5(:,2);
c6=[c6;c5];
%
d5=d4(:, :, i);
d5=d5(:,2);
d6=[d6;d5];
end

%% 分形特征
x=d6(25:end);
y3=RSzhishu2(x)
% R/S分析法, 也称为重标极差分析法
function y=RSzhishu(x,n)
% x=列向量, m*1,
% n=把x分成p个区间, 每个区间有n个元素, n*p=m
% y=R/S值, 1*1
% n=12;
m=size(x);
p=m/n;
c=[];
while length(x)>0
    c=[c,x(1:n)];
    x(1:n)=[];
end
for i=1:p
    c2=c(:,i);
    c3=mean(c2);
    c4=c2-c3;
    c5=cumsum(c4);
    c6=max(c5)-min(c5);
    c7=std(c2);
    c8(i)=(c6-c7)^2; % 有的文献这样定义
    c8(i)=c6/c7; % 有的文献这样定义
end
y=mean(c8);
% R/S分析法, 也称为重标极差分析法
function y3=RSzhishu2(x)
m=length(x);
n0=find(rem(m./(1:m),1)==0);
n0(1)=[];
n0(end)=[];
k=length(n0);
for i=1:k
    n=n0(i);
    y=RSzhishu(x,n);
    y2(i,:)= [n,y];

```

```

end

%% 拟合多项式系数
p=polyfit(y2(:,1),y2(:,2),1); % 多项式系数按照降幂排列，最后1个是常数项系数
y3=polyval(p,y2(:,1)); % 求多项式的值
plot(y2(:,1),y2(:,2),'*',y2(:,1),y3,'-');
y3=p(1);

```

2.3.2 傅里叶级数

```

% 5.2.2水、沙通量的周期性
clc,clear all

load data2 % a{}=数组，6个矩阵，若干行*6（6列：年-月-日-水位-流量-含沙量）

%% 按月汇总，水通量，沙通量
for i=1:6
    b=a{i};%年数据
    b(:,1)=[];% 若干行*5（5列：月-日-水位-流量-含沙量）
    for j=1:12
        p=find(b(:,1)==j);
        b2=b(p,:);%月数据
        b2(:,1)=[];% 若干行*4（4列：日-水位-流量-含沙量）
        c2=0;
        d2=0;
        for k=1:30
            q=find(b2(:,1)==k);
            b3=b2(q,:);%日数据
            b3(:,1:2)=[];% 若干行*2（2列：流量-含沙量）
            b5=mean(b3,1);% 1*2（2列：流量-含沙量），日均数据
            % 计算水通量
            c=b5(1)*24*3600/10^8;%亿立方米
            c2=c2+c;
            % 计算沙通量
            d=b5(2)*b5(1)*24*3600/1000/10^8;%亿吨
            d2=d2+d;
        end
        c3(j,:)=[j,c2];% 数组，12*2列（月-水通量）
        d3(j,:)=[j,d2];% 数组，12*2列（月-沙通量）
    end
    c4(:, :, i)=c3;%亿立方米，12*2*6，水通量
    d4(:, :, i)=d3;%亿吨，12*2*6，沙通量
end

%% 建立时间序列
c6=[];
d6=[];
for i=1:6
    c5=c4(:, :, i);
    c5=c5(:, 2);

```

```

c6=[c6;c5];

%
d5=d4(:, :, i);
d5=d5(:, 2);
d6=[d6;d5];

end

%% 拟合傅里叶级数
y=d6(25:60);    % (水通量、或者沙通量)
y3=d6(61:72);   % 留出最后1年用于预测误差的估计
t=1:length(y);
t=t';
plot(t, y, '*-');
f=inline(' p(1)+(p(2)*cos(pi/6*t)+p(3)*sin(pi/6*t))+(p(4)*cos(pi/6*2*t)+p(5)*sin(pi/6*2*t))+(p(6)*cos(pi/6*3*t)+p(7)*sin(pi/6*3*t))+(p(8)*cos(pi/6*4*t)+p(9)*sin(pi/6*4*t))+(p(10)*cos(pi/6*5*t)+p(11)*sin(pi/6*5*t))+(p(12)*cos(pi/6*6*t)+p(13)*sin(pi/6*6*t))+(p(14)*cos(pi/6*7*t)+p(15)*sin(pi/6*7*t))+(p(16)*cos(pi/6*8*t)+p(17)*sin(pi/6*8*t))+(p(18)*cos(pi/6*9*t)+p(19)*sin(pi/6*9*t))+(p(20)*cos(pi/6*10*t)+p(21)*sin(pi/6*10*t))', 'p', 't');
p=lsqcurvefit(f, ones(1, 21), t, y);
p=p';
%% 拟合检验
f2=p(1)+(p(2)*cos(pi/6*t)+p(3)*sin(pi/6*t))+(p(4)*cos(pi/6*2*t)+p(5)*sin(pi/6*2*t))+(p(6)*cos(pi/6*3*t)+p(7)*sin(pi/6*3*t))+(p(8)*cos(pi/6*4*t)+p(9)*sin(pi/6*4*t))+(p(10)*cos(pi/6*5*t)+p(11)*sin(pi/6*5*t))+(p(12)*cos(pi/6*6*t)+p(13)*sin(pi/6*6*t))+(p(14)*cos(pi/6*7*t)+p(15)*sin(pi/6*7*t))+(p(16)*cos(pi/6*8*t)+p(17)*sin(pi/6*8*t))+(p(18)*cos(pi/6*9*t)+p(19)*sin(pi/6*9*t))+(p(20)*cos(pi/6*10*t)+p(21)*sin(pi/6*10*t));
plot(t, y, '*, t, f2, '-');
e=sqrt(mean((f2-y).^2))

%% 预测误差估计
t2=37:48;
t2=t2';
f3=p(1)+(p(2)*cos(pi/6*t2)+p(3)*sin(pi/6*t2))+(p(4)*cos(pi/6*2*t2)+p(5)*sin(pi/6*2*t2))+(p(6)*cos(pi/6*3*t2)+p(7)*sin(pi/6*3*t2))+(p(8)*cos(pi/6*4*t2)+p(9)*sin(pi/6*4*t2))+(p(10)*cos(pi/6*5*t2)+p(11)*sin(pi/6*5*t2))+(p(12)*cos(pi/6*6*t2)+p(13)*sin(pi/6*6*t2))+(p(14)*cos(pi/6*7*t2)+p(15)*sin(pi/6*7*t2))+(p(16)*cos(pi/6*8*t2)+p(17)*sin(pi/6*8*t2))+(p(18)*cos(pi/6*9*t2)+p(19)*sin(pi/6*9*t2))+(p(20)*cos(pi/6*10*t2)+p(21)*sin(pi/6*10*t2));
e2=sqrt(mean((f3-y3).^2))

%% 今后2年的预测
t2=49:72;
t2=t2';
f3=p(1)+(p(2)*cos(pi/6*t2)+p(3)*sin(pi/6*t2))+(p(4)*cos(pi/6*2*t2)+p(5)*sin(pi/6*2*t2))+(p(6)*cos(pi/6*3*t2)+p(7)*sin(pi/6*3*t2))+(p(8)*cos(pi/6*4*t2)+p(9)*sin(pi/6*4*t2))+(p(10)*cos(pi/6*5*t2)+p(11)*sin(pi/6*5*t2))+(p(12)*cos(pi/6*6*t2)+p(13)*sin(pi/6*6*t2))+(p(14)*cos(pi/6*7*t2)+p(15)*sin(pi/6*7*t2))+(p(16)*cos(pi/6*8*t2)+p(17)*sin(pi/6*8*t2))+(p(18)*cos(pi/6*9*t2)+p(19)*sin(pi/6*9*t2))+(p(20)*cos(pi/6*10*t2)+p(21)*sin(pi/6*10*t2));

%% 把8年数据放在一起检验
t=1:length(d6);

```



```

t=t';
t2=73:96;
plot(t, d6, '*-', t2, f3, 'o-'); % (水通量、或者沙通量)

```

2.3.4 小波分析

```

waveletAnalyzer
>> shibu=real(coefs);
>> mo=abs(coefs);
>> mofang=mo.^2;
>> fangcha=sum(mofang, 2);

```

2.3.5 制定方案

```

!一般模型;
MODEL:
sets:
yue/1..12/:y,z;
endsets
data:
y=@file(y2023.txt);
enddata
@for(yue(i): @bin(z(i)));
@sum(yue(i): z(i)*y(i))/12>=10.8;
min=@sum(yue(i): z(i));
END

```

2.4 问题4的程序和结果

2.4.1 中间处理类1

```

from openpyxl import Workbook, load_workbook
def flow2_1():
    wk: Workbook = load_workbook('处理数据2.xlsx', data_only=True)
    sheet = wk.active
    new_wk = Workbook()
    new_sheet = new_wk.active
    new_sheet.append(['年', '月', '日', '起点距离', '水位', '水深', '测点水深', '测点水流速', '测点含沙量'])
    for row in sheet:
        if str(row[0].value) == '日期':
            continue
        string: str = str(row[0].value)
        sp = string.split(' ')[0].split('-')
        cell_list = [int(sp[0]), int(sp[1]), int(sp[2]), row[1].value, row[2].value, row[3].value, row[4].value,
                    row[5].value, row[6].value]
        new_sheet.append(cell_list)
    new_wk.save('中间数据1.xlsx')

```

中间处理类2

```

from openpyxl import Workbook, load_workbook
def flow2_2():
    wk: Workbook = load_workbook('中间数据1.xlsx', data_only=True)
    sheet = wk.active
    new_wk = Workbook()

```

```

new_sheet = new_wk.active
new_sheet.append(
    ['年', '月', '日', '起点距离', '水位', '水深', '测点水流速', '测点含沙量', '计数'])
flag = True
row_list = []
for row in sheet:
    if flag:
        flag = False
        continue
    year = int(row[0].value)
    month = int(row[1].value)
    day = int(row[2].value)
    s = int(row[3].value)
    w = float(row[4].value)
    h = float(row[5].value)
    v = float(row[7].value)
    try:
        sand = float(row[8].value)
    except:
        sand = None
    row_list.append((year, month, day, s, w, h, v, sand))
count = 1
for i in range(len(row_list) - 1):
    year = row_list[i][0]
    month = row_list[i][1]
    day = row_list[i][2]
    s = row_list[i][3]
    w = row_list[i][4]
    h = row_list[i][5]
    v = row_list[i][6]
    sand = row_list[i][7]
    new_sheet.append([year, month, day, s, w, h, v, sand, count])
    if row_list[i][3] != row_list[i + 1][3]:
        count += 1
new_wk.save("中间数据2.xlsx")

```

2.4.2 数据整理主方法

```

from 数据处理类 import flow1
from 中间处理类1 import flow2_1
from 中间处理类2 import flow2_2
from 平均流速处理类 import flow3
from 部分面积处理类 import flow4
from 径流量处理类 import flow5

flow1()
flow2_1()
flow2_2()

```

```

flow3()

flow4()

flow5()

```

2.4.3 数据处理类

```

from openpyxl import Workbook, load_workbook

def flow1():

    wk: Workbook = load_workbook('附件3.xlsx', data_only=True)

    sheet = wk.active

    new_wk = Workbook()

    new_sheet = new_wk.active

    new_sheet.append(

        ['日期', '起点距离(m)', '水位(m)', '水深(m)', '测点水深(m)', '测点水流速(m/s)', '测点含沙量(kg/m3)'])

    # 1. 水深 = 0 去除

    for row in sheet:

        try:

            if float(row[3].value) > 0:

                cell_list = []

                for cell in row:

                    cell_list.append(cell.value)

                new_sheet.append(cell_list)

            except:

                continue

    new_wk.save("处理数据1.xlsx")

    # 2. 测点水深 > 水深 / 水位 去除

    wk: Workbook = load_workbook('处理数据1.xlsx', data_only=True)

    sheet = wk.active

    new_wk = Workbook()

    new_sheet = new_wk.active

    new_sheet.append(

        ['日期', '起点距离(m)', '水位(m)', '水深(m)', '测点水深(m)', '测点水流速(m/s)', '测点含沙量(kg/m3)'])

    for row in sheet:

        try:

            if row[4].value == None or row[4].value > row[3].value:

                continue

            except:

                continue

            cell_list = []

            for cell in row:

                cell_list.append(cell.value)

            new_sheet.append(cell_list)

    new_wk.save("处理数据2.xlsx")

```

2.4.4 平均流速处理类

```

from openpyxl import Workbook, load_workbook

def flow3():

    wk: Workbook = load_workbook('中间数据2.xlsx', data_only=True)

    sheet = wk.active

```

```

new_wk = Workbook()
new_sheet = new_wk.active
new_sheet.append(['年', '月', '日', '起点距离', '水位', '水深', '平均水流速', '测点含沙量'])
count_list = []
vc_list = []
sc_list = []
def water_v(n, vc):
    list1 = []
    for i in vc:
        if int(i[1]) == n:
            list1.append(float(i[0]))
    return list1
def sand_sum(n, sc):
    list1 = []
    for i in sc:
        if int(i[1]) == n:
            list1.append(float(i[0]))
    return list1
flag = True
for row in sheet:
    if flag:
        flag = False
        continue
    v = float(row[6].value)
    try:
        s = float(row[7].value)
    except:
        s = 0
    count = int(row[8].value)
    count_list.append(count)
    vc_list.append((v, count))
    sc_list.append((s, count))
i = 0
flag = True
for row in sheet:
    if flag:
        flag = False
        continue
    year = int(row[0].value)
    month = int(row[1].value)
    day = int(row[2].value)
    s = int(row[3].value)
    w = float(row[4].value)
    h = float(row[5].value)
    v = float(row[6].value)
    n = int(row[8].value)

```

```

cell_list = []
if i != 0:
    i -= 1
    continue
if count_list.count(n) == 1:
    cell_list = [year, month, day, s, w, h, v, None]
    i = 0
elif count_list.count(n) == 2:
    v_list = water_v(n, vc_list)
    a = v_list[0]
    b = v_list[1]
    sum = (a + b) / 2
    cell_list = [year, month, day, s, w, h, sum, None]
    i = 1
elif count_list.count(n) == 5:
    v_list = water_v(n, vc_list)
    a = v_list[0]
    b = v_list[1]
    c = v_list[2]
    d = v_list[3]
    e = v_list[4]
    sum = (a + 3 * b + 3 * c + 2 * d + e) / 10
    sand_list = sand_sum(n, sc_list)
    sand = 0.0
    for sha in sand_list:
        sand += sha
    sand = sand / 5
    if sand == 0:
        sand = None
    cell_list = [year, month, day, s, w, h, sum, sand]
    i = 4
new_sheet.append(cell_list)
new_wk.save('平均流速.xlsx')

```

2.4.5 径流量处理类

```

from openpyxl import Workbook, load_workbook

def flow5():
    wk: Workbook = load_workbook('部分面积.xlsx', data_only=True)
    sheet = wk.active
    new_wk = Workbook()
    new_sheet = new_wk.active
    new_sheet.append(
        ['年', '月', '日', '起点距离', '水位', '水深', '平均水流速', '截面面积', '截面水流量', '径流量', '测点含沙
量',
        '截面输沙量'])
    row_list = []
    flag = True

```

```

for row in sheet:
    if flag:
        flag = False
        continue
    year = int(row[0].value)
    month = int(row[1].value)
    day = int(row[2].value)
    s = int(row[3].value)
    w = float(row[4].value)
    h = float(row[5].value)
    v = float(row[6].value)
    try:
        m = float(row[7].value)
        js = float(row[8].value)
    except:
        m = None
        js = None
    try:
        sand = float(row[9].value)
    except:
        sand = None
    row_list.append((year, month, day, s, w, h, v, sand, m, js))
for i in range(0, len(row_list)):
    if flag:
        flag = False
        continue
    year = row_list[i][0]
    month = row_list[i][1]
    day = row_list[i][2]
    s = row_list[i][3]
    w = row_list[i][4]
    h = row_list[i][5]
    v = row_list[i][6]
    try:
        m = row_list[i][8]
        js = row_list[i][9]
    except:
        m = None
        js = None
    try:
        sand = row_list[i][7]
    except:
        sand = None
    if m != None:
        cell_list = [year, month, day, s, w, h, v, m, js, js * 24 * 3600]
    else:

```

```

        cell_list = [year, month, day, s, w, h, v, m, js, None]

    if sand:

        cell_list.append(sand)

        if row_list[i + 2][7]:

            pj_js = (float(row_list[i][9]) + float(row_list[i + 1][9])) / 2

            pj_sand = (float(row_list[i][7]) + float(row_list[i + 2][7])) / 2

            cell_list.append(pj_js * pj_sand * 24 * 3600)

        else:

            cell_list.append(sand)

            cell_list.append(sand)

        new_sheet.append(cell_list)

    new_wk.save('径流量.xlsx')

```

2.4.6 部分面积处理类

```

from openpyxl import Workbook, load_workbook

def flow4():

    wk: Workbook = load_workbook('平均流速.xlsx', data_only=True)

    sheet = wk.active

    new_wk = Workbook()

    new_sheet = new_wk.active

    new_sheet.append(
        ['年', '月', '日', '起点距离', '水位', '水深', '平均水流速', '截面面积', '截面水流量', '测点含沙量'])

    s_list = []

    h_list = []

    v_list = []

    row_list = []

    flag = True

    for row in sheet:

        if flag:

            flag = False

            continue

        year = int(row[0].value)

        month = int(row[1].value)

        day = int(row[2].value)

        s = int(row[3].value)

        w = float(row[4].value)

        h = float(row[5].value)

        v = float(row[6].value)

        try:

            sand = float(row[7].value)

        except:

            sand = None

        row_list.append((year, month, day, s, w, h, v, sand))

    index = 0

    flag = True

    for i in range(0, len(row_list)):

        if flag:

```

```

        flag = False

        continue

    year = row_list[i - 1][0]
    month = row_list[i - 1][1]
    day = row_list[i - 1][2]
    s = row_list[i - 1][3]
    w = row_list[i - 1][4]
    h = row_list[i - 1][5]
    v = row_list[i - 1][6]

    try:
        sand = row_list[i - 1][7]
    except:
        sand = None

    if i == len(row_list) - 1:
        cell_list = [year, month, day, s, w, h, v, None, None, sand]
    elif row_list[i][2] == row_list[i - 1][2]:
        sub = abs(row_list[i - 1][3] - row_list[i][3])
        hc = abs(row_list[i - 1][5] + row_list[i][5]) / 2
        vc = (row_list[i - 1][6] + row_list[i][6]) / 2
        cell_list = [year, month, day, s, w, h, v, sub * hc, sub * hc * vc, sand]
    else:
        cell_list = [year, month, day, s, w, h, v, None, None, sand]

    new_sheet.append(cell_list)

new_wk.save('部分面积.xlsx')

```