42. 洪水灾害相关水科学问题的 1stOpt 解决之道

42.1 引言

2024 年 7 月 5 号，洞庭湖湖南华容县团洲垸发生决堤，瞬间引发了全国人民的高度担心和关注，汹涌溃决洪水给当地居民造成了严重生命威胁和巨大经济损失。刚刚经历过的 2023 年“7 •31 北京暴雨“和 2021 年“7 •20 郑州特大暴雨”、记忆深刻的“1998 特大暴雨”…，纵观历史，华夏大地几乎每年都会发生或大或小的暴雨及其引发的洪水灾害。在大自然面前人类无疑还是渺小的，征服自然将自然灾害清零的理想状况既不现实也无可能，但这并不意味着只能采取被动接受自然灾害的“绥靖政策“，而是要在尊重自然、敬畏自然的前提下以科学的手段将自然灾害的损失降到最低，学会与自然和谐相处，这正是涉水科学的重要使命担当和目的之一。在此以水库调洪演算、河道洪水演进和天然河道水面线三个洪水灾害相关的水文水力学计算问题为例，展示 1stOpt 计算平台如何方便、快捷、高效和精准地解决这些与洪水相关的科学计算问题。

42.2 水库调洪演算计算

水库调洪演算是为满足既定的防洪任务和要求而拟定的具体的泄洪规则，其目的是拦蓄洪水，削减洪峰，延长泄洪时间，使下泄流量能安全通过下游河道，是水库运行中经常遇到的问题，科学合理的水库规划调度是降低洪水灾害的有效手段。传统的调洪演算方法如试算法、图解法、辅助线法、简化三角形法等，不仅繁琐低效，精度也差强人意，而 1stOpt 却可以高效高精度实现。

水库调洪演算的基本公式如下：

$$\frac{Q_1+Q_2}{2}-\frac{q_1+q_2}{2}=\frac{V_1-V_2}{\Delta t} \tag{42-1}$$

或

$$V_2=V_1-\frac{(Q_1+Q_2-q_1-q_2)\cdot dt}{2} \tag{42-2}$$

其中：Q1、Q2：t1、t2 时刻水库入流量（m^3/s）；q1、q2：t1、t2 时刻水库出流量（m^3/s）；V1、V2：t1、t2 时刻水库库容（m^3）。

1) 水库相关数据

案例水库的相关公式和数据如下，包括水库入流洪水过程、水位～库容关系、水位～泄流能力关系等。起始时刻（0 时刻）进出水库流量相等均为 $10m^3/s$。

表 42-1 水库入流洪水过程

时间 (hr.)	0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96
流量 (m3/s)	10	60	140	300	710	620	510	390	260	170	120	80	60	40	30	20	15

已知水位（H）～库容（V）、库容（V）～水位（H）和水位(H)～泄流能力(Q)模型关系如下：

$$V\sim H:\quad V=(1.0437+0.0272\cdot(H-74.3302)^{2.4443})\cdot 10^6 \tag{42-3}$$

$$H\sim V:\quad H=\left(\frac{10^6\cdot V-1.0473}{0.0272}\right)^{\frac{1}{2.4443}}+74.3302 \tag{42-4}$$

$$H\sim Q:\quad Q=10+1.5\cdot 45\cdot(H-116)^{1.5} \tag{42-5}$$

其中 V：水库库容（m^3）；Q：水库泄流量（m^3/s）；H：库水位（m）。

2) 调洪演算优化模型之一：传统方法

传统的水库调洪演算方法是基于 t1 时刻的水库水位 H[t1]、入库流量 Q[t1]、出库流量 q[t1]及 t2 时刻的入库流量 Q[t2]，运用调洪演算公式，采用一些简单的一维优化算法如二分法等，迭代求出 t2 时刻的出库流量，然后再更换时间，重复计算出每一时刻的出库流量；1stOpt 中关键字“LoopConstant”在使用中可将上一循环计算的目标函数、参数或传递参数自动作为常数代入下一循环进行计算，该功能可以非常方便进行初始量变化的循环计算，代码中：

LoopConstant Qout1 = [10, Out1(15)], h1 =[116,DH(15)];

表示第一次循环时，初始出流量 Qout 和水位 h1 分别取 10 和 116，下一步循环时，初始出流量和水位将自动取上一循环的计算结果 Out1 和 DH。计算代码及结果如下。

1stOpt 代码

```
Algorithm = UGO1; //设定算法
Constant Dt = 6; //时间间隔
LoopConstant Qin1 = [10,60,140,300,710,620,510,390,260,170,120,80,60,40,30,20], //水库入流
             Qin2 = [60,140,300,710,620,510,390,260,170,120,80,60,40,30,20,15]; //水库出流
LoopConstant Qout1 = [10, Out1(15)], h1 =[116,DH(15)];
PassParameter Out1, DH;
Parameter Qout2 = [0,600]; //求解参数
PlotLoopData Qin1, Qout1, h1[y2];
Minimum;
StartProgram [Pascal];
function H2V(H: double): double; //水位-库容关系
begin
    result := (1.0473+0.0272*power(abs(h-74.3302),2.4443))*1000000;
end;
function V2H(V: double): double; //库容-水位  V:万方
begin
    try
    result := power((abs(V/1000000-1.0473)/0.0272),(1/2.4443))+74.3302;
    except
    end;
end;
function H2Q(H: double): double; //水位-泄流能力
begin
   try
    result := 10+1.6*45*power((H-116),1.5);
   except
   result := 10;
   end;
end;
Procedure MainModel;
var V1, V2, WH, WQ: double;
Begin
    V1 := H2V(H1);
    V2 := V1 + 0.5*(Qin1 + Qin2 - Qout1 - Qout2)*3600*Dt;
    WH := V2H(V2);
    WQ := H2Q(WH);
    Out1 := Qout2;
    DH := WH;
    ObjectiveResult := sqr(WQ - Qout2); //目标函数
End;
EndProgram;
```

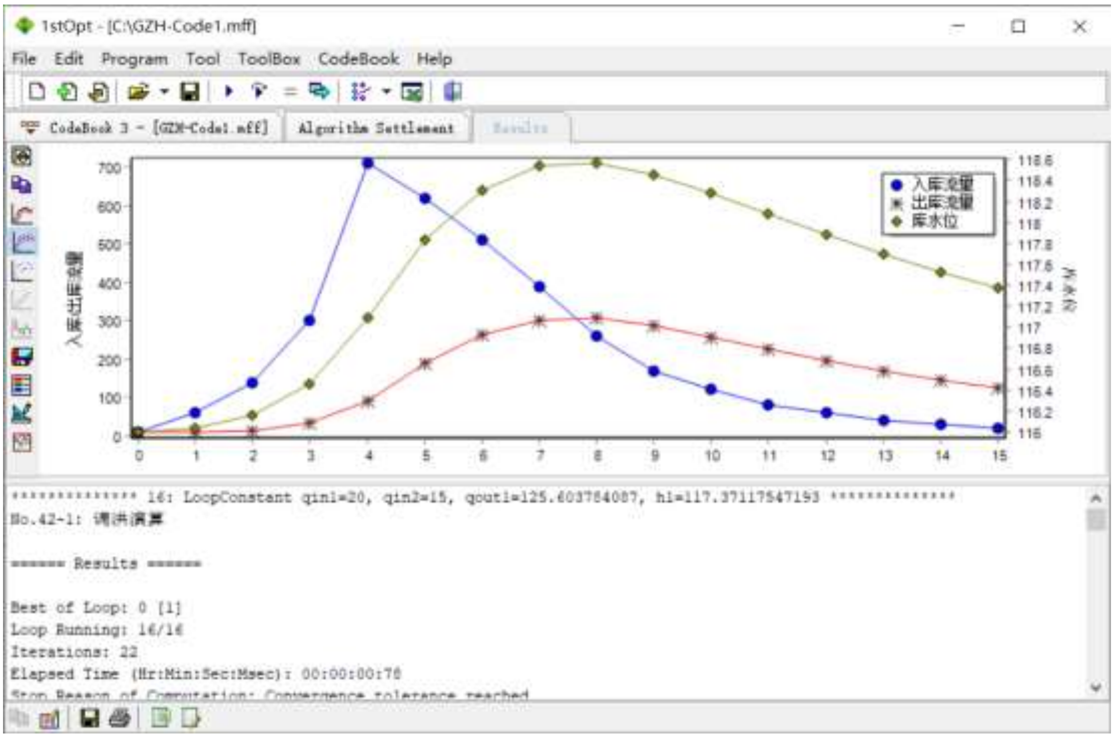

图 42-1 调洪演算结果

3) 调洪演算优化模型之二：全局单一模型法

相比上述传统的水库调洪演算方法，本方法视水库调洪演算为一整体来构筑优化模型，其步骤如下：

- 假设每一时刻待求出库流量为 X_i，$i = 1, 2, 3,\ldots n$-1，即 X 为模型优化参数向量， n 为时段数
- 由公式(42-1)或(42-2)及初始调洪水位 H_0 计算初始水库库容 V_0
- 循环计算：For i = 1 to n-1
    - ◆ 由公式(42-3)计算 i 时刻水库库容 V_i
    - ◆ 由公式(42-4)计算 i 时刻水位 H_i
    - ◆ 由公式(42-5)计算 i 时刻出库流量 $q_i^{'}$
- 优化模型目标函数：

$$Min.\quad \sum_{i=1}^{n-1}(X_i - q_i')^2 \tag{42-6}$$

虽然该模型的求解参数数目为 n – 1，相对于传统方法仅一个待求优化参数有了明显增加，在理论上增加了优化模型的求解难度，但对于 1stOpt 求解却是非常简单和快捷，可一并输出所有时刻的出库流量；同时该模型的一个最大优点是不存在累积误差，可保证调洪演算的精度。结果见表 42-2 和图 42-2.

1stOpt 调洪演算代码

```
Algorithm = UGO1; //设定算法
Constant n = 16, Dt = 6; //常数设定
Constant Qin(0:n-1) = [60,140,300,710,620,510,390,260,170,120,80,60,40,30,20,15]; //入库流量过程
PassParameter PH(0:n-1);
Parameter x(0:n-1) = [10,1000]; //定义待求参数
Plot x, Qin, PH[y2];
Minimum;
```

```
StartProgram [Pascal];
function H2V(H: double): double; //水位 H-库容 V 关系
begin
	result := (1.0473+0.0272*power(abs(h-74.3302),2.4443))*1000000;//方
end;
function V2H(V: double): double; //库容 V-水位 H 关系
begin
	try
	result := power((abs(V/1000000-1.0473)/0.0272),(1/2.4443))+74.3302;
	except
	//result := 118;
	end;
end;
function H2Q(H: double): double; //水位-泄流能力关系
begin
      try
	result := 10+1.6*45*power((H-116),1.5);
      except
      result := 10;
      end;
end;
Procedure MainModel;
var i: integer;
	V0, H0, H1, DH: double;
	WH, WQ: array[0..15] of double;
	DQ, MaxDH, MaxQ: double;
const Q0 = 10;
Begin
	H0 := 116; //初始水位
	V0 := H2V(H0);
	DH := 0;
	for i := 0 to n - 1 do begin
		if i = 0 then
		     V0 := V0 + (Qin[i] + Q0)*3600/2*Dt - (x[i] + Q0)*3600/2*Dt
		else
		     V0 := V0 + (Qin[i] + Qin[i-1])*3600/2*Dt - (x[i] + x[i-1])*3600/2*Dt;
		WH[i] := V2H(V0);
		WQ[i] := H2Q(WH[i]);
		DH := DH + sqr(WQ[i]-x[i]);
		PH[i] := WH[i];
	end;
	ObjectiveResult := DH; //目标函数
End;
EndProgram;
```

结果

Objective Function (Min.): 3.73475944487197E-21	PassParameter:
x0: 10.5075908575273	ph0: 116.036766661484
x1: 14.8746068769543	ph1: 116.166113425309
x2: 32.1083574303257	ph2: 116.455144522556
x3: 92.6041662181436	ph3: 117.095921979214
x4: 189.275447773514	ph4: 117.837069328575
x5: 262.327734222619	ph5: 118.307216387334
x6: 301.085561310216	ph6: 118.537808094053
x7: 306.122449264184	ph7: 118.567000131566
x8: 287.046560613802	ph8: 118.455539046294
x9: 257.961555747635	ph9: 118.280523599088
x10: 226.613403454524	ph10: 118.084020018914
x11: 196.655238195027	ph11: 117.887144361822
x12: 169.708137374585	ph12: 117.700838320735
x13: 145.987606464849	ph13: 117.527957315933
x14: 125.603784086991	ph14: 117.37117547193
x15: 108.195550117205	ph15: 117.229814305956

表 42-2 调洪演算结果

时间 (hr)	出流 (X) (m³/s)	水位 (m)	时间 (hr)	出流 (X) (m³/s)	水位 (m)
0	10	116	54	287.0466	118.4555
6	10.5076	116.0368	60	257.9616	118.2805
12	14.8746	116.1661	66	226.6134	118.0840
18	32.1084	116.4551	72	196.6552	117.8871
24	92.6042	117.0959	78	169.7081	117.7008
30	189.2754	117.8371	84	145.9876	117.5280
36	262.3277	118.3072	90	125.6038	117.3712
42	301.0856	118.5378	96	108.1956	117.2298
48	306.1224	118.5670			

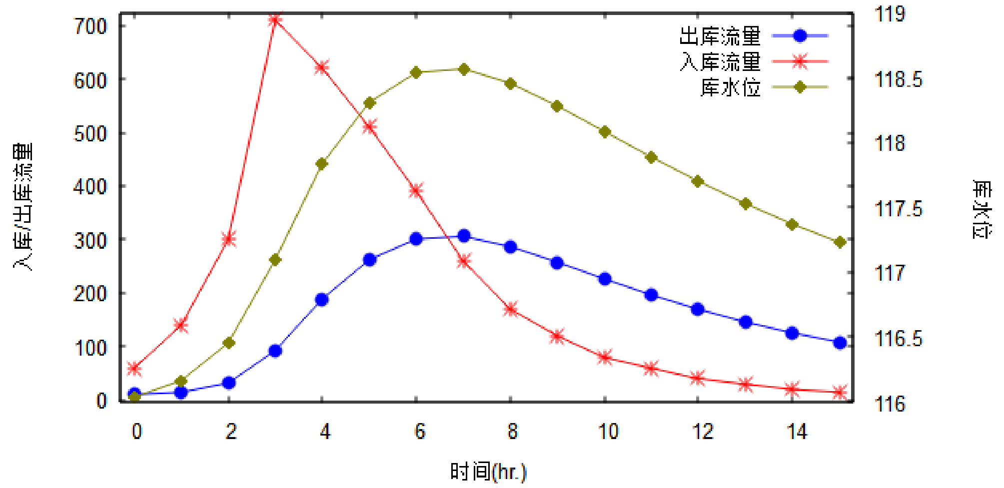

图 42-2 整体调洪演算结果

42.3 河道洪水演进模型

河道洪水演进也是洪水预报的重要手段和方式。马斯京根模型是经典和常用的河道洪水演进模型，作为对比还提出了非线性时间序列河道洪水演进模型（Nonlinear Time Series Model – NTSM）。该案例试验数据如表 42-3 示。

表 42-3 河道入流出流过程

序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
时间 (hr.)	0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120
入流 (m³/s)	22	23	35	71	103	111	109	100	86	71	59	47	39	32	28	24	22	21	20	19	19
出流 (m³/s)	21	22	23	26	34	44	55	66	75	82	85	84	80	73	64	54	44	36	30	25	22

✧ 马斯京根模型

马斯京根模型由于简单、易用和可靠，广泛用于河道洪水演进。非线性马斯京根模型定义如下：

$$S_t = k \cdot (x \cdot I_t + (1 - x) \cdot Q_t)^m \quad (42\text{-}7)$$

或

$$\begin{cases} Q_t = \dfrac{1}{1-x}\cdot\left(\left(\dfrac{S_t}{k}\right)^{\frac{1}{m}} - x\cdot I_t\right) \\ S_{t+1} = S_t + (I_t - Q_t)\cdot\Delta t \end{cases} \tag{42-8}$$

其中 St 和 St+1 是 t 及 t+1 时刻的河道槽蓄量； It 和 Qt 是 t 时刻的入流和出流量；k, x 和 m 为模型参数，x 范围为[0,1]，标准的马斯京根模型中 m 为常数 1，本案例中也设为一待求参数，一般大于 1。

优化模型如公式：

$$Min.\quad \sum_{i=1}^{21}(Q_i - Q_i')^2 \tag{42-9}$$

其中 Q_i 和 $Q_i^{'}$ 是计算和实际的河道出流。

1stOpt 求解代码如下，不用给初值，可非常容易求得参数 *k*, *x* 和 *m*，结果见表 42-4 和图 42-3。

马斯京根模型求解代码

```
Constant QS1 = 22, QE1 = 21; //初始条件
Parameters k, x, m>1;    //定义参数
PassParameter Qcal(21); //计算河道输出流量
Constant Qin = [23,35,71,103,111,109,100,86,71,59,47,39,32,28,24,22,21,20,19,19,18],//入流
            Qout = [22,23,26,34,44,55,66,75,82,85,84,80,73,64,54,44,36,30,25,22,19];    //实际出流
EndRowDataSet;
Plot Qout, Qcal;
StartProgram;
Procedure MainModel;
var i: integer;
    dw0, dQ, temd: double;
begin
    temd := 0; dw0 := k*power(abs(x*QS1+(1-x)*QE1),m);    //计算初始槽蓄量
    for i := 1 to 21 do begin
        dQ := 1/(1-x)*power(abs(dw0/k),1/m)-(x/(1-x))*Qin[i]; //计算出流量
        dw0 := dw0 + (Qin[i] - dQ)*6*3600;
        temd := temd + sqr(Qout[i] - dq);    //计算河道槽蓄量
        Qcal[i] := dq;
    end;
    ObjectiveResult := temd;
end;
EndProgram;
```

✧ 非线性时间系列模型

非线性时间系列模型（Nonlinear Time Series Model – NTSM）是一纯数学模型，不用考虑各参数的物理意义，模型公式如下：

$$Q_{t+1} = C_1\cdot q_t^{k_1} + C_2\cdot q_{t+1}^{k_2} + C_3\cdot Q_t^{k_3} \tag{42-10}$$

其中 Q[t+]1 和 Q[t] 是 t+1 及 t 时刻的河道出流；q[t+1]和 q[t]是 t+1 及 t 时刻的河道入流，c1、c2、c3、k1、k2、k3 为待求优化参数。

优化模型如下式：

$$Min.\quad \sum_{i=1}^{21}(Q_i - Q_i')^2 = \sum_{i=1}^{21}\left(C_1\cdot q_{i-1}^{k_1} + C_2\cdot q_i^{k_2} + C_3\cdot Q_{i-1}^{k_3} - Q_i'\right)^2 \tag{42-11}$$

其中 Q_i 和 $Q_i^{'}$ 分别为计算和实际的河道出流。

上述模型看似非常简单，待求优化参数也仅有 6 个，但其全局最优解却很难被求解获得。一些著名的优化软件包如 Lingo、Mathematica、Matlab 等都很难获得该问题的全局最优解。1stOpt 求解代码如下表，结果见表 42-4 和图 42-3。注意参数 c2 值为-8.707087E-10，虽然很小，但却不能忽视或视为 0，如 c2 值变为-8.707087E-11，目标函数值将巨变为 9335.34285。

从所得计算结果来看，非线性时间序列模型效果优于马斯京根模型，如果仅从目标函数值考虑，精度提高超 30 倍。

非线性时间序列模型求解代码

```
Constant QS1 = 22, QE1 = 21;    //初始条件
Parameter c(3), k(3);           //定义参数
PassParameter Qcal(21);           //计算河道输出流量
Constant Qin(21) = [23,35,71,103,111,109,100,86,71,59,47,39,32,28,24,22,21,20,19,19,18], //实际入流
           Qout(21) = [22,23,26,34,44,55,66,75,82,85,84,80,73,64,54,44,36,30,25,22,19];      //实际出流
Plot Qout, Qcal;
StartProgram [Pascal];
Procedure MainModel;
var i: integer;
    q1,q2,q3,temd, temQ: double;
Begin
    q1 := QS1; q3 := QE1; temd := 0;
    for i := 1 to 21 do begin
        q2 := Qin[i];
        temQ := c1*power(q1,k1)+c2*power(q2,k2)+c3*power(q3,k3);   //替换入流和出流
        q1 := q2; q3 := temQ;
        if q3 < 1 then q3 := 1
        else if q3 > 1200 then q3 := 1200;
        temd := temd + sqr(temQ - Qout[i]);                         //计算出流
        Qcal[i] := temQ;                                             //目标函数
    end;
    ObjectiveResult := temd;
End;
EndProgram;
```

表 42-4 河道洪水演进计算结果

模型	目标函数	参数
马斯京根模型	189.562836	k = 271.252616, x = 0.18521388, m = 2.2961301
非线性时间序列模型	5.6080766	c1 = 1.74484811, c2 = -8.707087E-10, c3 = 0.07907328, k1 = 0.67574882, k2 = 4.91942918, k3 = 1.48200156

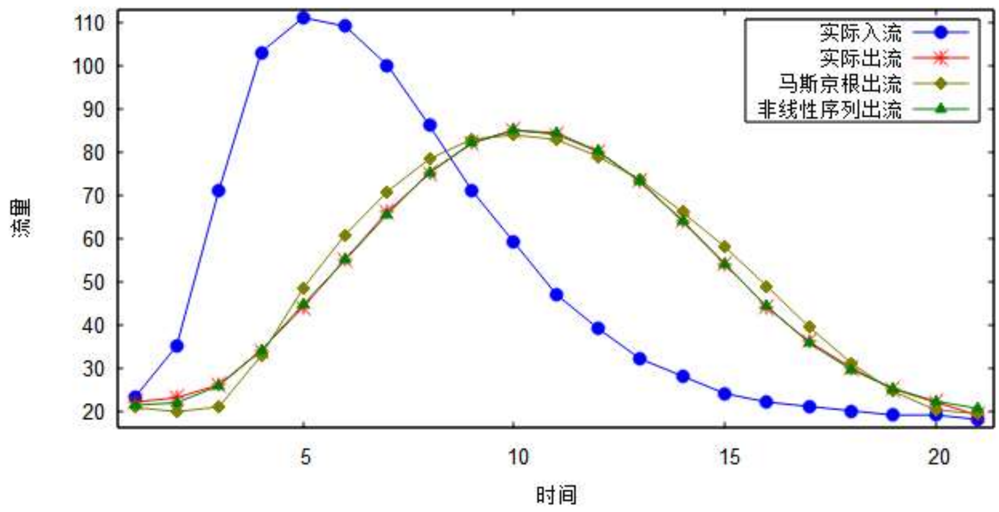

图 42-3 河道洪水演进计算结果

42.4 天然河道水面线计算

计算天然河道水面线的基本公式为伯努利方程：

$$z_u+(\alpha_u+w_u)\cdot\frac{Q^2}{2\cdot g\cdot A_u}=z_d+(\alpha_d+w_d)\cdot\frac{Q^2}{2\cdot g\cdot A_d}+\Delta s\cdot\frac{Q^2}{K^2} \qquad (42\text{-}12)$$

其中 Zu、Zd、au、ad、wu、wd、Au、Ad 分别为上下游两个断面的水位、动能修正系数、局部阻力系数和过水断面面积，Q 为河道流量，g 重力加速度，K 为该河段平均流量模数，其计算公式如下：

$$\frac{1}{K^2}=\frac{1}{2}\cdot\left(\frac{1}{K_u^2}+\frac{1}{K_d^2}\right)=\frac{1}{2}\cdot\left(\frac{n^2}{A_u^2\cdot R_u^{4/3}}+\frac{n^2}{A_d^2\cdot R_d^{4/3}}\right) \qquad (42\text{-}13)$$

其中 n 为河段糙率系数，R 为水力半径，Ku、Kd 分别为上下游断面的流量模数。两式整理可得：

$$\begin{aligned}z_u+(\alpha_u+w_u)\cdot\frac{Q^2}{2\cdot g\cdot A_u}-\frac{\Delta s}{2}\cdot\frac{(n\cdot Q)^2}{A_u^2\cdot R_u^{4/3}}&\\=z_d+(\alpha_d+w_d)\cdot\frac{Q^2}{2\cdot g\cdot A_d}-\frac{\Delta s}{2}\cdot\frac{(n\cdot Q)^2}{A_d^2\cdot R_d^{4/3}}&\end{aligned} \qquad (42\text{-}14)$$

公式左边为上游断面总能量，右端为下游断面总能量。由于断面水利要素 A、R 是水位 Z 的函数，计算过程要根据水位实时动态计算各水利要素，另外也可用拉格朗日二次插值进行快速计算，插值基本公式：

$$y=\frac{y_1\cdot(x-x_2)\cdot(x-x_3)}{(x_1-x_2)\cdot(x_1-x_3)}+\frac{y_2\cdot(x-x_1)\cdot(x-x_3)}{(x_2-x_1)\cdot(x_2-x_3)}+\frac{y_3\cdot(x-x_1)\cdot(x-x_2)}{(x_3-x_1)\cdot(x_3-x_2)} \qquad (42\text{-}15)$$

式中，x_1、x_2、x_3为插值基点，y_1、y_2、y_3为插值基点所对应的函数值。如果将水位 Z 视作 x，各断面水力要素 A、R 作为 y，即可求得个断面水力要素值。

已知某水库河道流量为 650m³/s，其它数据如下表 42-5，试计算库（断面 5）前水位为 477.6 时上游其它四个断面的河道水位。

表 42-5 河道断面水利要素

	水位 Z（m）	过水断面积 A（m2）	水力半径 R（m）	距离 Δs（m）	动能修正系数ξ	局部阻力系数α	糙率系数 n
断面-1（上游）	476	345	0.685	1850	0.025	0.012	0.03
	478	1050	1.346				
	480	1580	1.687				
断面-2	476	320	0.725	1500	0.022	0.018	0.03
	478	820	1.454				
	480	1220	1.890				
断面-3	476	280	0.855	1020	0.020	0.020	0.03
	478	560	1.708				
	480	960	2.564				
断面-4	476	270	1.754	570	0.018	0.022	0.03
	478	440	2.045				
	480	640	2.842				
断面-5（下游）	476	240	1.537	0	0.015	0.023	0.03
	478	380	2.235				
	480	560	2.957				

在 1stOpt 中，可用两种方式进行水面线计算，一种是从下游（断面-5）开始，基于公式(42-14)，求出断面-4 水位，再重复相同步骤，从断面-4 逐步推求上游断面 3、2、1 的水位，该方法使用关键字“LoopConstant”自动循环赋值求解，模型目标函数：

$$Min.\quad (EngL - EngR)^2 \tag{42-15}$$

其中，EngL：上游断面总能量，公式（42-14）左边；EngR：下游断面总能量，公式（42-14）右边。

代码如下

```
Constant Q=650, n=0.03, Hd=477.6, g=9.8;              //初始条件
Constant Z(5,3)=[476,478,480,                          //断面水位
                 476,478,480,
                 476,478,480,
                 476,478,480,
                 476,478,480];
Constant A(5,3)=[345,1050,1580,                        //断面面积
                 320,820,1220,
                 280,560,960,
                 270,440,640,
                 240,380,560];
Constant R(5,3)=[0.685,1.346,1.687,                   //断面水力半径
                 0.725,1.454,1.890,
                 0.855,1.708,2.564,
                 1.754,2.045,2.842,
                 1.537,2.235,2.957];
Constant k=[0.025,0.022,0.020,0.018,0.015],        //能量修正系数
         w=[0.012,0.018,0.02,0.022,0.023],          //局部损失系数
         Ds=[1850,1500,1200,600];                      //断面间距离
LoopConstant p=[4,3,2,1];                              //循环计算断面号
LoopConstant H0=[Hd,H(3)];                             //循环计算初始水位
Parameter H=477[470,490];                              //定义水位参数
PlotLoopData H;                                        //实时作水位图
StartProgram [Pascal];
Function InterP(x,x1,x2,x3,y1,y2,y3: double): double;       //插值子程序
begin
     Result := max(0.001,(y1*(x-x2)*(x-x3)/((x1-x2)*(x1-x3))+
                y2*(x-x1)*(x-x3)/((x2-x1)*(x2-x3))+
                y3*(x-x1)*(x-x2)/((x3-x1)*(x3-x2))));
end;
Procedure MainModel;
var EngL, EngR: double;
    temA, TemR, temH: double;
    i: integer;
Begin
    i := round(p);                                                                  //
循环数号，整数
    temH                                        :=                                 H0;
//下游段水位
    temA := InterP(temH, Z[i+1,1], Z[i+1,2], Z[i+1,3], A[i+1,1], A[i+1,2], A[i+1,3]);      //下游段面积
    temR := InterP(temH, Z[i+1,1], Z[i+1,2], Z[i+1,3], R[i+1,1], R[i+1,2], R[i+1,3]);      //下游段水力半径
    EngR := temH + (k[i+1]+w[i+1])*Q^2/(2*g*temA)+Ds[i]/2*(n*Q)^2/(temA^2*(temR)^(4/3));   //下游段总
能量
    temA := InterP(H, Z[i,1], Z[i,2], Z[i,3], A[i,1], A[i,2], A[i,3]);                     //上游段面积
    temR := InterP(H, Z[i,1], Z[i,2], Z[i,3], R[i,1], R[i,2], R[i,3]);                     //上游段水力半径
    EngL := H + (k[i]+w[i])*Q^2/(2*g*temA)-Ds[i]/2*(n*Q)^2/(temA^2*(temR)^(4/3));          //上游段总
能量
    ObjectiveResult := sqr(EngR-EngL);                                                //目标
函数
End;
EndProgram;
```

结果

```
************** 4: LoopConstant p=1, h0=480.298137353129 **************
No.42.5: 天然河道水面线计算

====== Results ======

Best of Loop: 0 [1]
```

```
Loop Running: 4/4
Iterations: 21
Elapsed Time (Hr:Min:Sec:Msec): 00:00:00:74
Stop Reason of Computation: Convergence tolerance reached
Algorithms: Universal Global Optimization(UGO1)
Objective Function (Min.): 0
h: 480.659967524419

====== Finished ======

************** 3: LoopConstant p=2, h0=479.883534564756 **************
No.42.5: 天然河道水面线计算

====== Results ======

Best of Loop: 0 [1]
Loop Running: 3/4
Iterations: 22
Elapsed Time (Hr:Min:Sec:Msec): 00:00:00:80
Stop Reason of Computation: Convergence tolerance reached
Algorithms: Universal Global Optimization(UGO1)
Objective Function (Min.): 0
h: 480.298137353129

====== Finished ======

************** 2: LoopConstant p=3, h0=478.76007755506 **************
No.42.5: 天然河道水面线计算

====== Results ======

Best of Loop: 0 [1]
Loop Running: 2/4
Iterations: 22
Elapsed Time (Hr:Min:Sec:Msec): 00:00:00:88
Stop Reason of Computation: Convergence tolerance reached
Algorithms: Universal Global Optimization(UGO1)
Objective Function (Min.): 0
h: 479.883534564756

====== Finished ======

************** 1: LoopConstant p=4, h0=477.6 **************
No.42.5: 天然河道水面线计算

====== Results ======

Loop Running: 1/4
Iterations: 21
Elapsed Time (Hr:Min:Sec:Msec): 00:00:00:112
Stop Reason of Computation: Convergence tolerance reached
Algorithms: Universal Global Optimization(UGO1)
Objective Function (Min.): 0
h: 478.76007755506
```

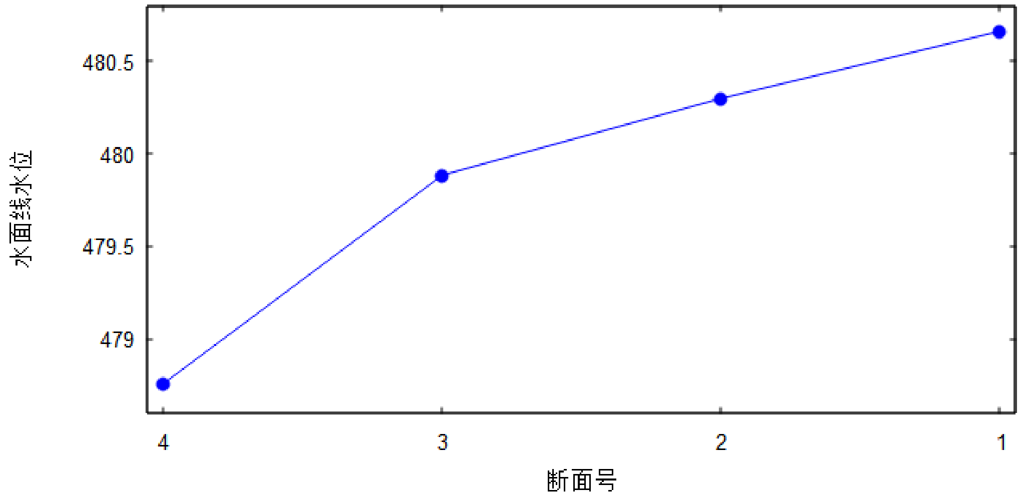

图 42-4 河道水面线计算水位

上面求解方法是从最下游断面开始逐一断面往上游推，下面方法是将每一断面水位设为待求参数 H_i，i=1..4，构筑一整体模型如下：

$$Min.\quad \sum_{i=1}^{4}(Eng_i - Eng_{i+1})^2 \qquad (42\text{-}16)$$

其中，Eng_i：断面 i 的总能量。

代码如下

```
Constant Q=650, n=0.03, Hd=477.6, g=9.8;   //初始条件
Constant Z(5,3)=[476,478,480,                //断面水位
                 476,478,480,
                 476,478,480,
                 476,478,480,
                 476,478,480];
Constant A(5,3)=[345,1050,1580,              //断面面积
                 320,820,1220,
                 280,560,960,
                 270,440,640,
                 240,380,560];
Constant R(5,3)=[0.685,1.346,1.687,          //断面水力半径
                 0.725,1.454,1.890,
                 0.855,1.708,2.564,
                 1.754,2.045,2.842,
                 1.537,2.235,2.957];
Constant k=[0.025,0.022,0.020,0.018,0.015], //能量修正系数
         w=[0.012,0.018,0.02,0.022,0.023],   //局部损失系数
         Ds=[1850,1500,1200,600];            //断面间距离
Parameter H(4)=477[470,490];                 //定义水位参数
PassParameter HH(5);
Plot HH;                                     //实时作水位图
StartProgram;
Function InterP(x,x1,x2,x3,y1,y2,y3: double): double;   //插值子程序
begin
      Result := max(0.001,y1*(x-x2)*(x-x3)/((x1-x2)*(x1-x3))+
                y2*(x-x1)*(x-x3)/((x2-x1)*(x2-x3))+
                y3*(x-x1)*(x-x2)/((x3-x1)*(x3-x2)));
end;
Procedure MainModel;
var EngL, EngR: double;
    temA, TemR: double;
    temD, temH: double;
    i: integer;
Begin
    temD := 0;
```

```
	for i := 4 downto 1 do begin
		if i = 4 then temH := Hd
		else temH := H[i+1];                                                                          //下游段水位
		temA := InterP(temH, Z[i+1,1], Z[i+1,2], Z[i+1,3], A[i+1,1], A[i+1,2], A[i+1,3]);         //下游段面积
		temR := InterP(temH, Z[i+1,1], Z[i+1,2], Z[i+1,3], R[i+1,1], R[i+1,2], R[i+1,3]);         //下游段水力半径
		EngR := temH + (k[i+1]+w[i+1])*Q^2/(2*g*temA)+Ds[i]/2*(n*Q)^2/(temA^2*(temR)^(4/3));    //下游
段总能量
		temH := H[i];
		temA := InterP(temH, Z[i,1], Z[i,2], Z[i,3], A[i,1], A[i,2], A[i,3]);                         //上游段面积
		temR := InterP(temH, Z[i,1], Z[i,2], Z[i,3], R[i,1], R[i,2], R[i,3]);                         //上游段水力半径
		EngL := temH + (k[i]+w[i])*Q^2/(2*g*temA)-Ds[i]/2*(n*Q)^2/(temA^2*(temR)^(4/3));          //上游
段总能量
		temD          :=          temD          +          sqr(EngR-EngL);
//累积目标函数值
		HH[i] := H[i];
	end;
	HH[5] := Hd;
	ObjectiveResult          :=          temD;
//目标函数
End;
EndProgram;
```

结果

Objective Function (Min.): 1.64951446370438E-22 h1: 480.659967524418 h2: 480.29813735313 h3: 479.883534564764 h4: 478.760077555053	PassParameter: hh1: 480.659967524418 hh2: 480.29813735313 hh3: 479.883534564764 hh4: 478.760077555053 hh5: 477.6

两种方法所得结果一样如下表。

表 42-6 水面线计算成果

	断面-1	断面-2	断面-3	断面-4	断面-5
水位（m）	480.660	480.298	479.884	478.760	477.6

42.5 小结

暴雨洪水是自然现象，人类无法避免，但洪水灾害却可以在科学运筹规划指导下将损失将为最低。通过三个洪水灾害相关的水文水力学计算问题，充分展示了 1stOpt 在这一领域的技术优势与应用前景，有望在防洪减灾相关研究中发挥更大作用。