综合优化软件包1stOpt使用手册

第一篇 1stOpt 简介

1．1：概要

1stOpt 是七维高科有限公司（7D-Soft High Technology Inc.）独立开发，拥有完全自主知识产权的一套数学优化分析综合工具软件包。在非线性回归，曲线拟合，非线性复杂模型参数估算求解，线性/非线性规划等领域傲视群雄，首屈一指，居世界领先地位。除去简单易用的界面，其计算核心是基于七维高科有限公司科研人员十数年的革命性研究成果【通用全局优化算法】(Universal Global Optimization - UGO)，该算法之最大特点是克服了当今世界上在优化计算领域中使用迭代法必须给出合适初始值的难题，即用户勿需给出参数初始值，而由 1stOpt 随机给出，通过其独特的全局优化算法，最终找出最优解。以非线性回归为例，目前世界上在该领域最有名的软件工具包诸如 Matlab, OriginPro, SAS, SPSS, DataFit, GraphPad 等，均需用户提供适当的参数初始值以便计算能够收敛并找到最优解。如果设定的参数初始值不当则计算难以收敛，其结果是无法求得正确结果。而在实际应用当中，对大多数用户来说，给出(猜出)恰当的初始值是件相当困难的事，特别是在参数量较多的情况下，更无异于是场噩梦。而 1stOpt 凭借其超强的寻优，容错能力，在大多数情况下（大于 90%），从任一随机初始值开始，都能求得正确结果。

1．2：国内外类似软件概况

数据综合分析领域，国外软件无疑占绝对统治地位。在非线性曲线拟合，参数优化方面，名声大，应用广的有诸如OriginPro，Matlab，SAS，SPSS，DataFit，GraphPad，TableCurve2D，TableCurve3D 等。无论这些软件界面，历史，名声如何，最常用算法有麦夸特法（Levenberg-Marquardt）或简面体爬山法（Simplex Method）等，均可归属于局部最优法。因而如何有效地确定参数初始值始终是难以克服的瓶颈，由于此，一些实际问题可能永远无法获得正解。国内方面，因无自己独特有效的技术理论与方法，虽有个别分析软件面市，与上述国外类似产品相比，功能相差甚远，即使在国内也无竞争优势，在国际上就更无声影了。而 1stOpt凭借自己革命性的算法理论，在非线性拟合，参数估算等优化领域强于目前世界上任何已知软件包，其英文版已远销美国，德国，法国，英国，芬兰，瑞典，荷兰，南非，澳大利亚，新西兰，土耳其等国。

1. 3: 1stOpt 应用的优化算法

最优化算法包括：

1) Levenberg-Marquardt 法 (LM) + 通用全局优化算法(Universal Global Optimization - UGO)
2) Quasi-Newton 法 (BFGS) + 通用全局优化算法(Universal Global Optimization - UGO)
3) 遗传算法 (Genetic Algorithms - GA)
4) 摸拟退火 (Simulated Annealing - SA)
5) 下山单体法 (Simplex Method - SM) + 通用全局优化算法(Universal Global Optimization - UGO)
6) 离子群法 (Particle Swarm Optimization - PSO)
7) 最大继承法 (Max Inherit Optimization - MIO)
8) 差分进化法 (Differential Evolution - DE)
9) 自组织群移法 (Self-Organizing Migrating Algorithms - SOMA)
10) 共扼梯度法 (Conjugate-Gradient Method - CGM) + 通用全局优化算法(Universal Global Optimization - UGO)
11) 包维尔法 (Powell Optimization - PO) + 通用全局优化算法(Universal Global Optimization - UGO)
12) 禁忌搜索法 (Tabu Search - TS)
13) 单纯线性规划法 (Simplex Linear Program)

1. 4: 1stOpt 应用范围

1) 模型自动优化率定
2) 参数估算
3) 任意模型公式线性，非线性拟合，回归
4) 非线性连立方程组求解
5) 任意维函数，隐函数极值求解
6) 隐函数根求解，作图，求极值
7) 线性，非线性及整数规划
8) 组合优化问题
9) 高级计算器

1. 5: 1stOpt 特长

1) 功能强劲，是目前唯一能以任何初始值而求得美国国家标准与技术研究院(NIST：National Institute of Standards and Technology)非线性回归测试题集最优解的软件包。
2) 可广泛用于水文水资源及其它工程模型优化计算。内镶 VB 及 Pascal 语言，可帮助描述处理复杂模型。
3) 可连接由任何语言（C++, Fortran, Basic, Pascal…）编译而成的外部目标函数动态连接库或命令行可执行文件。
4) 非线性曲线拟合可处理任意类型模型公式，任意多数目的待求参数及变量
5) 模型自动率定时可同时处理多个数据文件
6) 可非常容易处理一些特殊的参数，如降雨径流模型中的流域初期土壤含水量
7) 可同时处理多个输出量
8) 实时显示计算结果
9) 可直接读存 Excel，CSV 等格式文件

10）界面简单友好，使用方便

11）自带有上百个实例，覆盖范围包括几乎所有优化方面。通过不同类型实例，用户可轻松掌握 1stOpt 的用法。

1．6：1stOpt 关键词

表.1：主要关键词

关键词名	意义及示例
Parameter	定义参数 例： 定义 a， b， c， d 四个参数： *Parameter a, b, c, d;* 例： 定义 a1， a2， a3， a4， a5， a6， a7， a8， a9， a10 十个参数： *Parameter a1, a2, a3, a4, a5, a6, a7, a8, a9, a10;* 也可简写为： *Parameter a(1:10);* 例： 定义参数 a，其取值范围在【-1，1】，初始值为 0.5 *Parameter a = 0.5 [-1, 1];* 例： 定义参数 a 为整数，其取值范围在【-100，100】 *Parameter a[-100,100,0];*
Variable	定义变量 例：定义 x， y， z 三个变量： *Variable x, y, z;*
Function	定义函数 例：两变量曲线拟合： *Function y = a + b*exp(c - x);* 例：两变量函数优化： *Function (x+((2-x)*(2+y))^2)*sin(x*y);*
Constant	定义常量 例：两变量曲线拟合： *Function y = a + b*exp(c - x);* 例：两变量函数优化： *Function (x+((2-x)*(2+y))^2)*sin(x*y);*
ConstStr	定义常字符串量 例：两变量曲线拟合： *Function y=a*(c-x)^2+b*exp((c - x)^4);* 可写为： *ConstStr B = （c-x）^2* *Function y = a*B + b*exp(B^2);*
VarConstant	定义变常量，详情见 2.4 节
VarParameter	定义变参数，详情见 2.4 节
Data	定义数据开始
DataFile	定义数据文件
NewDivision	定义新得代码块
StartProgram	编程模式开始
EndProgram	编程模式结束
Maximum	求最大值
Minimum	求最小值
PlotFunction	画函数图
Algorithms	定义优化方法

Exclusive	定义问题为排它问题，如 TSP 问题
StartRange	定义初始值范围
SharedModel	定义共享参数问题
DataSet	定义常数
EndDataSet	结束定义常数
MinFunction	最小值求优
MaxFunction	最大值求优
PlotParaFunction	画参数方程函数图
Title	定义代码块名

1stOpt 还有两个特殊定义符：

求和定义：如 $\sum_{i=1}^{n}(x_i \cdot \sin(x_i + 1))$，在 1stOpt 中表达为：Sum(i=1:n)(x[i]*sin(x[i]+1))

求积定义：如 $\prod_{i=1}^{n}(x_i \cdot \sin(x_i + 1))$，在 1stOpt 中表达为：Prod(i=1:n)(x[i]*sin(x[i]+1))

1. 7: 1stOpt 界面

1. 7. 1: 主界面

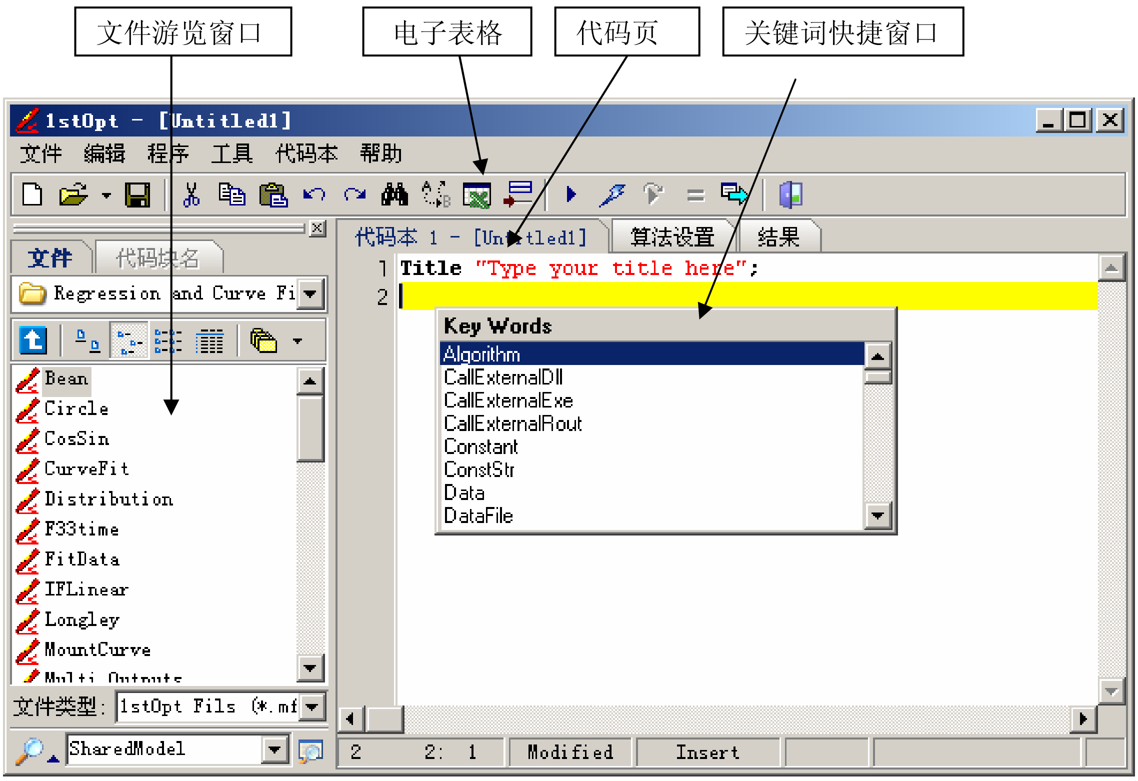


图.1 1stOpt 主画面

关键词快捷窗口由组合键“Ctrl + K”弹出，可帮助用户准确快速输入关键词。在同一代码本中可写多个不同问题的代码，由关键词“NewDivision”来区分。可同时开启多个代码编辑本。同一代码文件中还可加入富文本如图，表，公式等，也可把不同格式的文件添付进来。

1．7．2：数据处理电子表格

1stFit 电子表格类似与 Excel，多表单，支持公式，直接输入输出到 Excel，文本文件(.txt，.csv)，树型表单管理，直观并可分类，可方便用于数据前，后处理。

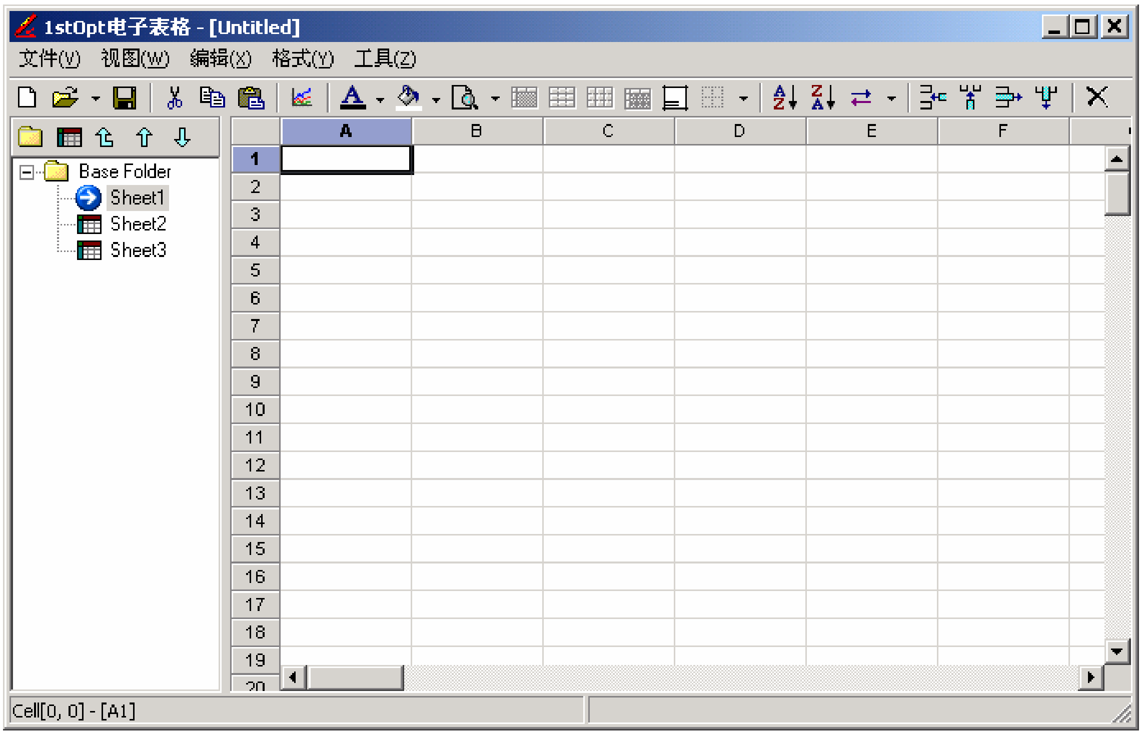


图.2 1stOpt 电子表格

1．7．3：输入代码

每一句代码以‘;’号作结束符。如

Parameter a, b, c, d;

Constant p1 = 1, p2 = 4, p3 = 5;

对曲线拟合，对二维，缺省自变量名为 x，因变量名为 y；对三维或多维，缺省自变量名为 x1，x2，x3…，因变量名为 y。如下两段代码效果等同，右边代码中无需再定义变量和参数，将由 1stOpt 自动识别。

代码 1	代码 2
Variables x, y; **Parameters** a, b, c, d; **Function** y=a-b*exp(-c*x^d); **Data**; 0.05 0.13 0.15 0.13 0.25 0.19 0.35 0.34	**Function** y=a-b*exp(-c*x^d); **Data**; 0.05 0.13 0.15 0.13 0.25 0.19 0.35 0.34

对函数优化，如参数没有范围限制，也可省去参数定义，下列左右两段代码效果等同

代码 1	代码 2
Parameters x, y; **Minimum** = True; **Function** exp(sin(50*x)) +sin(60*exp(y)) + sin(70*sin(x))+sin(sin(80*y))- sin(10*(x+y)) +(x^2+y^2)/4;	**MinFunction** exp(sin(50*x)) +sin(60*exp(y)) sin(70*sin(x))+sin(sin(80*y))- sin(10*(x+y)) +(x^2+y^2)/4;

1．7．4：执行计算

因为 1stOpt 的初始值通常是随机产生，一次计算不成功，并非意味下次亦同样不成功，反之亦然。按快捷键”F9”执行计算，“F10“中止计算。

1．7．5：优化算法设定

在 1stOpt 中，共有 13 种有化算法。不同的问题该选用何种算法？一般而言：

- ✧ 非线性回归，曲线拟合问题：
  1) Levenberg-Marquardt 法 (LM) + 通用全局优化算法(Universal Global Optimization - UGO)
  2) BFGS + 通用全局优化算法(Universal Global Optimization - UGO)
  3) 下山单体法 (Simplex Method - SM) + 通用全局优化算法(Universal Global Optimization - UGO)
  4) 差分进化法
  5) 最大继承法
- ✧ 函数优化，方程求根问题：
  1) 下山单体法 (Simplex Method - SM) + 通用全局优化算法(Universal Global Optimization - UGO)
  2) BFGS + 通用全局优化算法(Universal Global Optimization - UGO)
  3) 差分进化法
  4) 最大继承法
- ✧ 线性规划问题：
  1) 单纯线性规划法 (Simplex Linear Program - SLP)
  2) 下山单体法 (Standard Simplex Method - SM) + 通用全局优化算法(Universal Global Optimization - UGO)
  3) 差分进化法
- ✧ 优化组合问题：
  1) 最大继承法
  2) 禁忌搜索法
  3) 模拟退火
  4) 遗传算法

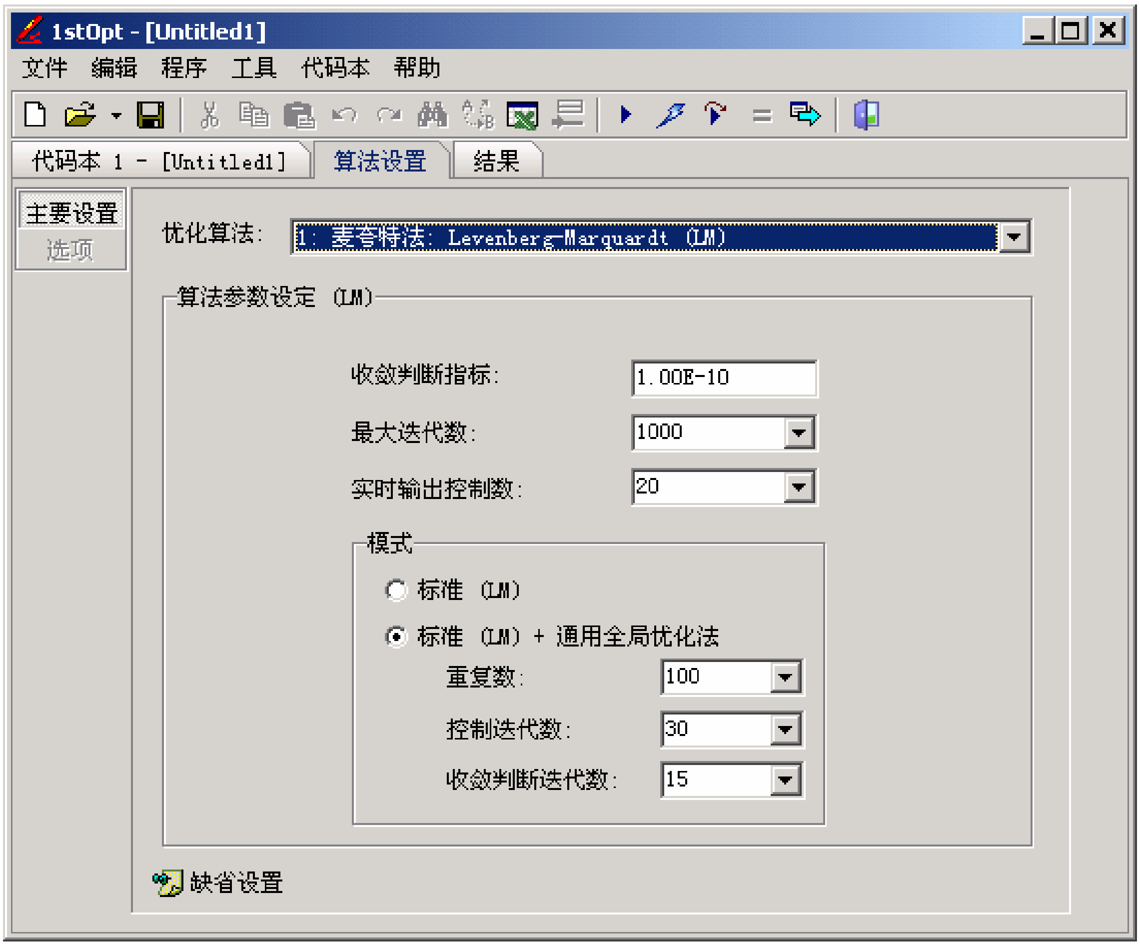


图.3 优化算法设置

对 90%以上的问题，缺省优化设置均可满足要求。对个别较为复杂的问题，可适当改变参数，方法如下表.2：

表.2：优化参数修改方法

算法	参数改变
Levenberg-Marquardt 法 (LM) + 通用全局优化算法 Quasi-Newton 法 (BFGS) + 通用全局优化算法 共扼梯度法 + 通用全局优化算法 包维尔法 + 通用全局优化算法	加大“重复数”，如从缺省值“30”变为“60”。“重复数”越大，成功概率越高，但计算时间相对变长。
最大继承法	增加“种群数”和“局部步长”
差分进化法	启动“局部搜索”，变换进化法选项
下山单体法	变换“模式”选项，增加“多重计算数”
自组织群移法	增加“种群数”，减少“步长”
摸拟退火	加大“内部循环数”
遗传算法	增加“种群数“
禁忌搜索法	
离子群法	增加“种群数“

第二篇 1stOpt 应用

2．1：求任意形式，任意维数，约束或非约束的函数最优值。

1stOpt 即可用于无约束函数求优，也可用于有约束函数求优。约束函数即可是不等式也可是等式。

例 2．1．1：求下列一维函数最小值

$$f = x \cdot \sin(x) + \sin(x) \qquad (1)$$

其中， $x \in [-3\pi, 3\pi]$

1stOpt 代码：

```
Parameter x = [-3*pi,3*pi];
Minimum;
Function x*sin(x)+sin(x);
```

或更为简单形式：

```
Parameter x = [-3*pi,3*pi];
MinFunction x*sin(x)+sin(x);
```

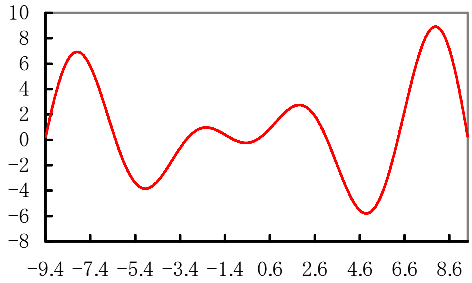


图. 4 一维函数图

结果：f = -5.7976，x = 4.8808

例 2.1.2：求下列多维函数最小值

$$f = \sum_{i=1}^{n-1}\left(3\cdot\left(\cos(2\cdot x_i) + \sin(2\cdot x_{i+1})\right) + \sqrt{x_i^2 + x_{i+1}^2}\right) \qquad (2)$$

其中， $\overline{X} \in [-30, 30]$，n = 20

1stOpt 代码：

```
Constant n = 20;
Parameter x(1:n) = [-30,30];
MinFunction Sum(i = 1:n-1) (3*(Cos(2*x[i]) + Sin(2*x[i+1])) + Sqrt(x[i+1]^2 + x[i]^2));
```

结果：f = -51.7695

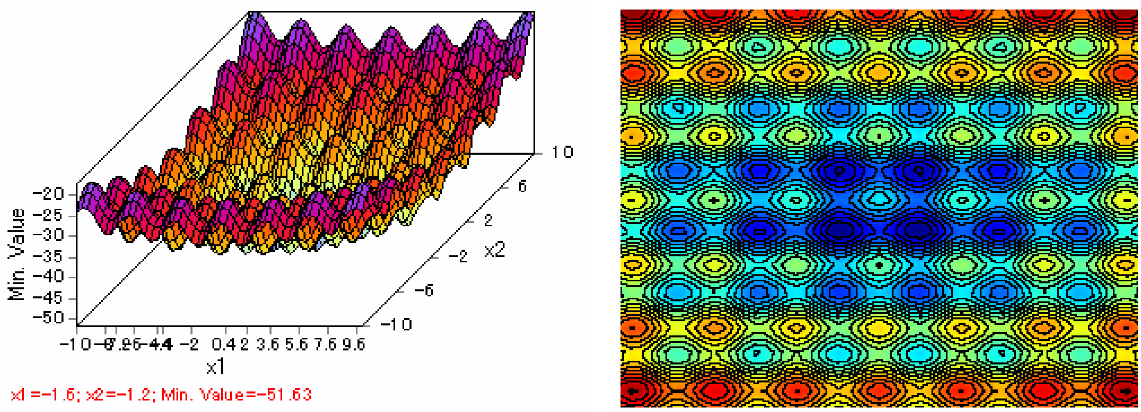


图. 5 多维函数优化三维图

例 2.1.3：求下列隐函数z的最小值

$$z=\sin\left((z\cdot x-0.5)^2+2\cdot x\cdot y^2-\frac{z}{10}\right)\cdot\exp\left(-\left(\left(x-0.5-\exp(-y+z)\right)^2+y^2-\frac{z}{5}+3\right)\right)\quad(3)$$

其中，$x\in[-1,7]$，$y\in[-2,2]$

1stOpt 代码：

```
Parameters x[-1,7], y[-2,2];
Minimum = z;
Function z = sin((z*x-0.5)^2 + x*2*y^2-z/10)*exp(-((x-0.5-exp(-y+z))^2 + y^2-z/5+3));
```

结果：z = 0.02335 (x = 2.898329，y = -0.8573138)

隐函数最优化乃 1stOpt 特色之一。据测试，目前尚无其它软件，如著名的 Lingo/Lindo，能求出此例的正确答案。

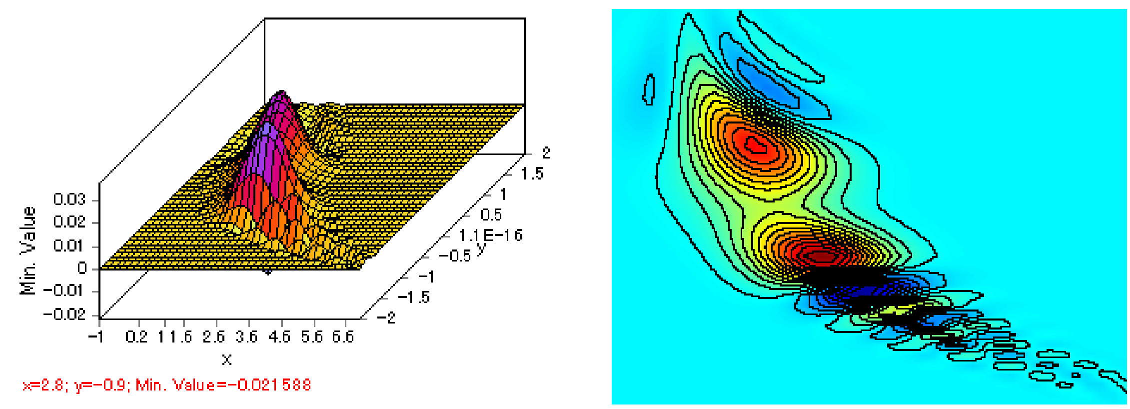


图.6 隐函数优化三维图

2．1．4 针状全局最优的函数

$$f(r)=\frac{\sin(r)}{r}+1\qquad(4)$$

其中：

$$r=\sqrt{(x-50)^2+(y-50)^2}+e$$

$$0\le x,y\le 100$$

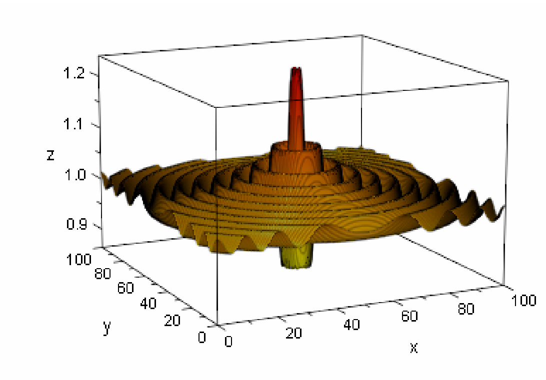


图.7 针状函数三维图

该函数的函数图形如上所示，在(50, 50)处取得全局最大值 1.1512，其第二极大值为 1.12837，它是一个多峰值函数，采用传统优化方法几乎不能找到全局最优点。1stOpt 可轻易求的最优值

1stOpt 代码：

```
Parameter x[0,100], y[0,100];
ConstStr r = sqrt((x-50)^2+(y-50)^2)+exp(1);
MaxFunction Sin(r)/r + 1;
```

2．1．4 线性规划实例

潘森等等将线性规划用于饲料配方工作中的应用，见<<计算机农业应用专刊>>(全国农业计算机应用技术学术交流会(二) 1992 P148-151)。 其目标函数和约束条件方程如下：

Min: 0.44x1+0.94x2+0.88x3+0.48x4+4x5+3.4x6+2.3x7+0.12x8+1.6x9+19x10+25x11

3230x1+2640x2+2500x3+1730x4+2900x5+2230x6+2500x7>2750

8.27x1+43x2+40x3+15.4x4+62x5+50x6+45x7>15

8.27x1+43x2+40x3+15.4x4+62x5+50x6+45x7<16

0.038x1+0.32x2+0.32x3+0.14x4+3.91x5+4.6x6+33.4x8+21x9>2.85

0.038x1+0.32x2+0.32x3+0.14x4+3.91x5+4.6x6+33.4x8+21x9<3

0.058x1+0.15x2+0.14x3+0.32x4+2.9x5+2.15x6+0.14x8+18.5x9>0.5

0.058x1+0.15x2+0.14x3+0.32x4+2.9x5+2.15x6+0.14x8+18.5x9<0.55

0.26x1+2.45x2+2.41x3+0.54x4+4.35x5+3.28x6+2.6x7+99x11>0.8

0.125x1+0.48x2+0.51x3+0.18x4+1.65x5+1.31x6+0.65x7+99x10>0.31

0.298x1+1.08x2+1.4x3+0.58x4+2.21x5+1.74x6+0.83x7+99x10>0.58

0.298x1+1.08x2+1.4x3+0.58x4+2.21x5+1.74x6+0.83x7+99x10<0.63

0.077x1+0.6x2+0.6x3+0.27x4+0.8x5+0.64x6>0.19

x1>0.5, x1<0.66

x2+x3>0.1, x2+x3<0.22

x4>0.04, x4<0.2

x5+x6>0.03, x5+x6<0.07

x7<0.035

x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11=1

1stOpt 代码：

```
Parameter x1[0.5,0.66], x4[0.04,0.2],x7[,0.035];
MinFunction    0.44*x1+0.94*x2+0.88*x3+0.48*x4+4*x5+3.4*x6+2.3*x7+0.12*x8+1.6*x9+19*x10+25*x11;
               3230*x1+2640*x2+2500*x3+1730*x4+2900*x5+2230*x6+2500*x7>2750;
               8.27*x1+43*x2+40*x3+15.4*x4+62*x5+50*x6+45*x7>15;
               8.27*x1+43*x2+40*x3+15.4*x4+62*x5+50*x6+45*x7<16;
               0.038*x1+0.32*x2+0.32*x3+0.14*x4+3.91*x5+4.6*x6+33.4*x8+21*x9>2.85;
               0.038*x1+0.32*x2+0.32*x3+0.14*x4+3.91*x5+4.6*x6+33.4*x8+21*x9<3;
               0.058*x1+0.15*x2+0.14*x3+0.32*x4+2.9*x5+2.15*x6+0.14*x8+18.5*x9>0.5;
               0.058*x1+0.15*x2+0.14*x3+0.32*x4+2.9*x5+2.15*x6+0.14*x8+18.5*x9<0.55;
               0.26*x1+2.45*x2+2.41*x3+0.54*x4+4.35*x5+3.28*x6+2.6*x7+99*x11>0.8;
               0.125*x1+0.48*x2+0.51*x3+0.18*x4+1.65*x5+1.31*x6+0.65*x7+99*x10>0.31;
               0.298*x1+1.08*x2+1.4*x3+0.58*x4+2.21*x5+1.74*x6+0.83*x7+99*x10>0.58;
               0.298*x1+1.08*x2+1.4*x3+0.58*x4+2.21*x5+1.74*x6+0.83*x7+99*x10<0.63;
               0.077*x1+0.6*x2+0.6*x3+0.27*x4+0.8*x5+0.64*x6>0.19;
               x2+x3>0.1;
               x2+x3<0.22;
               x5+x6>0.03;
               x5+x6<0.07;
               x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11=1;
```

原文最小值	DPS最小值	1stOpt 最小值
0.6567	0.6544	0.48133

*1stOpt优化算法：简面体爬山法（标准）+ 通用全局优化法

显而易见，1stOpt 结果最优。

例 2．1．5：线性规划问题

$$\text{Max}\quad 2\cdot x_1+3\cdot x_2+x_3 \tag{5}$$

$$\text{St.}\quad \begin{cases} x_1+3\cdot x_2+x_3\le 15 \\ 2\cdot x_1+3\cdot x_2-x_3\le 18 \\ x_1-x_2+x_3\le 3 \\ x_1,x_2,x_3\ge 0 \end{cases} \tag{6}$$

1stOpt 代码：

```
Parameter x(1:3)[0,];
MaxFunction 2*x1+3*x2+x3;
            x1+3*x2+x3 <= 15;
            2*x1+3*x2-x3 <= 18;
            x1-x2+x3 <= 3;
```

结果：x_1=5，x_2=3，x_3=1，最大值为 20

例 2．1．6：非线性规划问题

$$\text{Max}\quad \frac{(\sin(2\cdot\pi\cdot x_1))^3\cdot\sin(2\cdot\pi\cdot x_2)}{x_1^3\cdot(x_1+x_2)} \tag{7}$$

$$\text{St.}\quad \begin{cases} x_1^2-x_2+1\le 0 \\ 1-x_1+(x_2-4)^2\le 0 \\ 0\le x_1\le 10 \\ 0\le x_2\le 10 \end{cases} \tag{8}$$

1stOpt 代码：

```
Parameters x1[0,10], x2[0,10];
MaxFunction (sin(2*pi*x1))^3*sin(2*pi*x2)/(x1^3*(x1+x2));
            x1^2-x2+1 <= 0;
            1-x1+(x2-4)^2 <= 0;
```

结果：x_1= 1.22797，x_2= 4.24537，最大值为 0.095825

例 2．1．7：非线性混合整数规划问题 1

$$\text{Min}\quad 1.5\cdot(x_1-\sin(x_1-x_2))^2+0.5\cdot x_2^2+x_3^2-x_1\cdot x_2-2\cdot x_1+x_2\cdot x_3 \tag{9}$$

$$\text{St.}\quad \begin{cases} -20 < x_1 < 20 \\ -20 < x_2 < 20 \\ -10 < x_3 < 10 \end{cases} \qquad x_1，\ x_2 \text{为实数，}\ x_3 \text{为整数} \qquad (10)$$

1stOpt 代码：

```
Parameters x1[-20,20],x2[-20,20],x3[-10,10,0];
MinFunction 1.5*(x1-sin(x1-x2))^2+0.5*x2^2+x3^2-x1*x2-2*x1+x2*x3;
```

结果：当x_1= 4.49712，x_2= 9.147501，x_3= -4 时，得最小值为-10.51832

例 2．1．8：非线性整数规划问题 2

$$\text{Min:}\quad -2\cdot x_1 - x_2 - 4\cdot x_3 - 3\cdot x_4 - x_5 \qquad (11)$$

$$\text{St.}\quad \begin{cases} 2\cdot x_2 + x_3 + 4\cdot x_4 + 2\cdot x_5 < 54 \\ 3\cdot x_1 + 4\cdot x_2 + 5\cdot x_3 - x_4 - x_5 < 62 \\ x_1, x_2 \in [0,100]; x_3 \in [3,100], x_4 \in [0,100], x_5 \in [2,100] \end{cases} \qquad (12)$$

其中，x1 至 x5 均为整数。

1stOpt 代码：

```
Parameter x(1:2)[0,100,0], x3[3,100,0], x4[0,100,0], x5[2,100,0];
MinFunction -2*x1-x2-4*x3-3*x4-x5;
            2*x2+x3+4*x4+2*x5<54;
            3*x1+4*x2+5*x3-x4-x5<62;
```

结果有两组：x1, x2, x3, x4, x5 = (15, 0, 6, 11, 2)或 x1, x2, x3, x4, x5 = (19, 0, 4, 10, 5)，最小值为-89

2．1．9 北师大范例

此列来自北京师范大学系统科学系系统分析与集成实验室开发的通用遗传算法平台所付带的实例。

$$\text{Min:}\quad f(x_1, x_2) = (x_1 - 10)^3 + (x_2 - 20)^3 \qquad (13)$$

其中：$x_1 \in [13,100]$，$x_2 \in [0,100]$

约束条件：$(x_1 - 5)\cdot(x_1 - 5) + (x_2 - 5)\cdot(x_2 - 5) - 100 \ge 0$ (14)

1stOpt 代码：

```
Parameter x1[13,100],x2[0,100];
MinFunction (x1-10)^3+(x2-20)^3;
            (x1-5)*(x1-5)+(x2-5)*(x2-5)-100>=0;
```

结果：$f(x_1, x_2) = -7950.96189$，$x_1$ = 13.660254，x_2 = 0；而北师大给出的正解是：$f(x_1, x_2) = -7936$，x_1 = 14，x_2 = 0；显然，1stOpt所的结果优于后者。

2．2：非线性拟合

如第一章前所述，1stOpt的非线性拟合功能强于目前任何已知软件包，如著名的SPSS，SAS，Matlab，Origin，Systat，DataFit等。其最大特点是，在绝大多数情况下，不需要使用者提供(猜测)任何初始值，仅依靠自身的全局搜索能力，从任意随机值出发，既可求得最优解。美国国家标准与技术研究院（NIST：National Institute of Standards and Technology）提供有一套27道非线性拟合测试题，世界上几乎所有著名的数据分析软件包都以能通过该套测试题集为验证标准。经对比测试，1stOpt是目前唯一不依赖使用NIST提供的初始值，而能以任意随机初始值就可求得全部最优解的软件包(如果使用NIST提供的初始值，则更可轻易求得最优解)。由于在实际应用当中，选择确定合理的初始值组是一件非常困难的事，尤其是在参数量比较多的情况下。从此意义而言，1stOpt的实用能力达业界领先水平。

表.3 NIST 测试题结果

序号	测试题名	难度	待定参数量	初始值	1stOpt 使用算法	成功率(%)
1	Misra1a	低	2	1stOpt 默认设置：0-5 间随机值	任一算法	100
2	Chwirut2		3		任一算法	100
3	Chwirut1		3		任一算法	100
4	Lanczos3		6		改进 LM 及 BFGS	100
5	Gauss1		8		改进 LM 及 BFGS	＞ 90
6	Gauss2		8		改进 LM 及 BFGS	＞ 90
7	DanWood		2		任一算法	100
8	Misra1b		2		任一算法	100
9	Kirby2	中	5	1stOpt 默认设置：0-5 间随机值	任一算法	100
10	Hahn1		7		改进 LM，BFGS 及 SM	100
11	Nelson		3		改进 LM，BFGS 及 MIO	100
12	MGH17		5		任一算法	100
13	Lanczos1		6		改进 LM 及 BFGS	100
14	Lanczos2		6		改进 LM 及 BFGS	100
15	Gauss3		8		改进 LM 及 BFGS	＞ 90
16	Misra1c		2		任一算法	100
17	Misra1d		2		任一算法	100
18	Roszman1		4		任一算法	100
19	ENSO		9		任一算法	100
20	MGH09	高	4		任一算法	100
21	Thurber		7		改进 LM，BFGS，SM 及 DE	100
22	BoxBod		2		任一算法	100
23	Rat42		3		任一算法	100
24	MGH10		3		任一算法	100
25	Eckerle4		3		MIO 及改进 SM	100
26	Rat43		4		任一算法	100

27	Bennett5		3		任一算法	>90

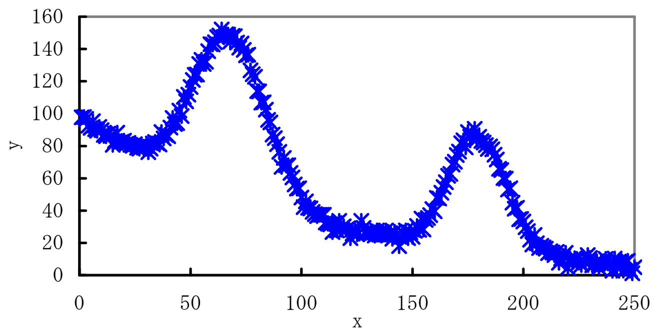
注：SM：下山单体法；MIO：最大继承法；DE：差分进化法；SA：摸拟退火
NIST 网址：http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

例 2．2．1：两变量非线性回归

回归模型公式： $$y = b_1 \cdot \exp(-b_2 \cdot x) + b_3 \cdot \left(\frac{(x-b_4)^2}{b_5^2}\right) + b_6 \cdot \left(\frac{(x-b_7)^2}{b_8^2}\right) \qquad (15)$$

其中：b_1至b_8：待定参数、x：自变量、y：因变量

表.2 原回归数据

X	y	图.8
1.000000	97.62227	
2.000000	97.80724	
3.000000	96.62247	
..	.	
.	.	
248.0000	3.362571	
249.0000	1.182746	
250.0000	4.875359	

1stOpt 代码：

```
Parameter b(1:8);
Variable x, y;
Function y = b1*exp(-b2*x ) + b3*exp(-((x-b4)^2/b5^2) ) + b6*exp( -((x-b7)^2 /b8^2) );
Data;
1.000000     97.62227
2.000000     97.80724
3.000000     96.62247
.
248.0000     3.362571
249.0000     1.182746
```

结果：

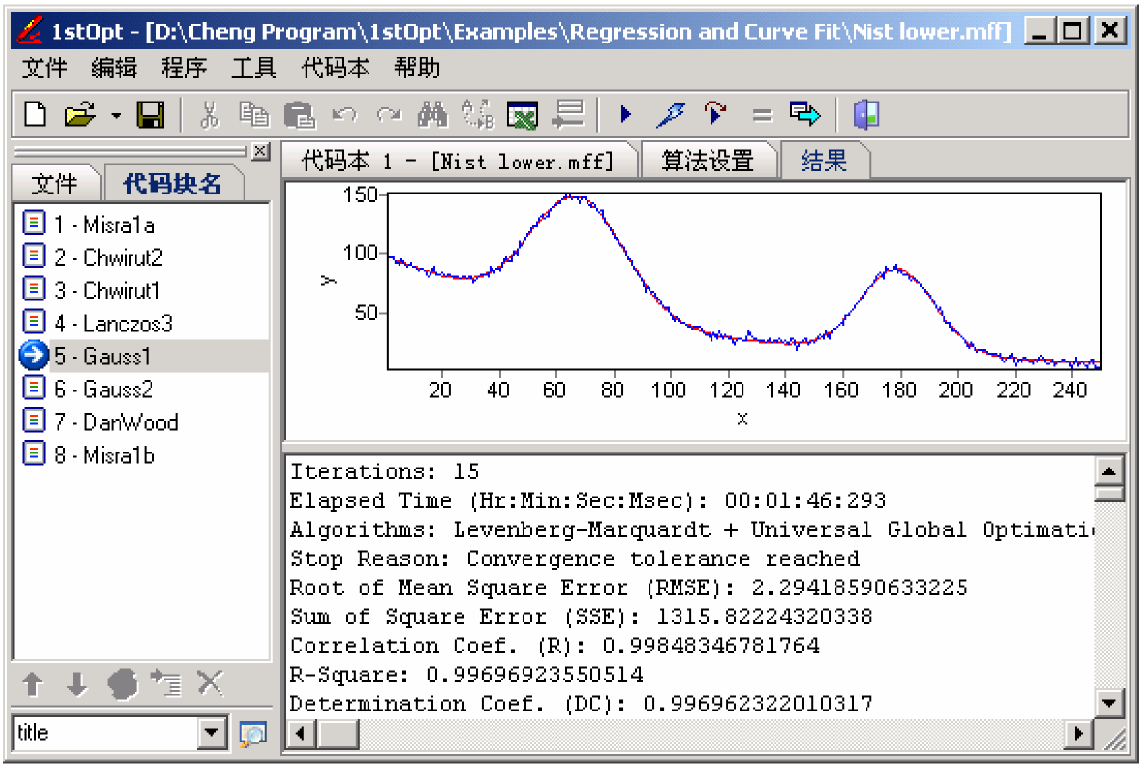
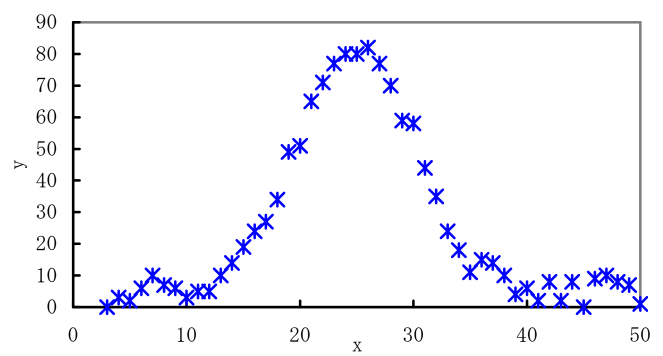


图.9 非线性拟合示图

例 2. 2. 2：两变量非线性回归 － 回归模型公式自动搜索

1stOpt 可自动搜索最匹配的模型公式。对常用的两变量及三变量问题，1stOpt 模型公式库中分别预置有超过 3，7 0 0 和近 1,000 个函数式，且可根据需要自行随意添加修改。而对多变量问题，也可非常容易建立自己的函数库。

表.4 原数据

X	y	图.10
1	3	
2	7	
3	0	
.	.	
..	.	
48	8	
49	7	
50	1	

1stOpt 代码：

```
Variable x, y;
Data;
1         3
2         7
3         0
.
48        8
49        7
50        1
```

结果：

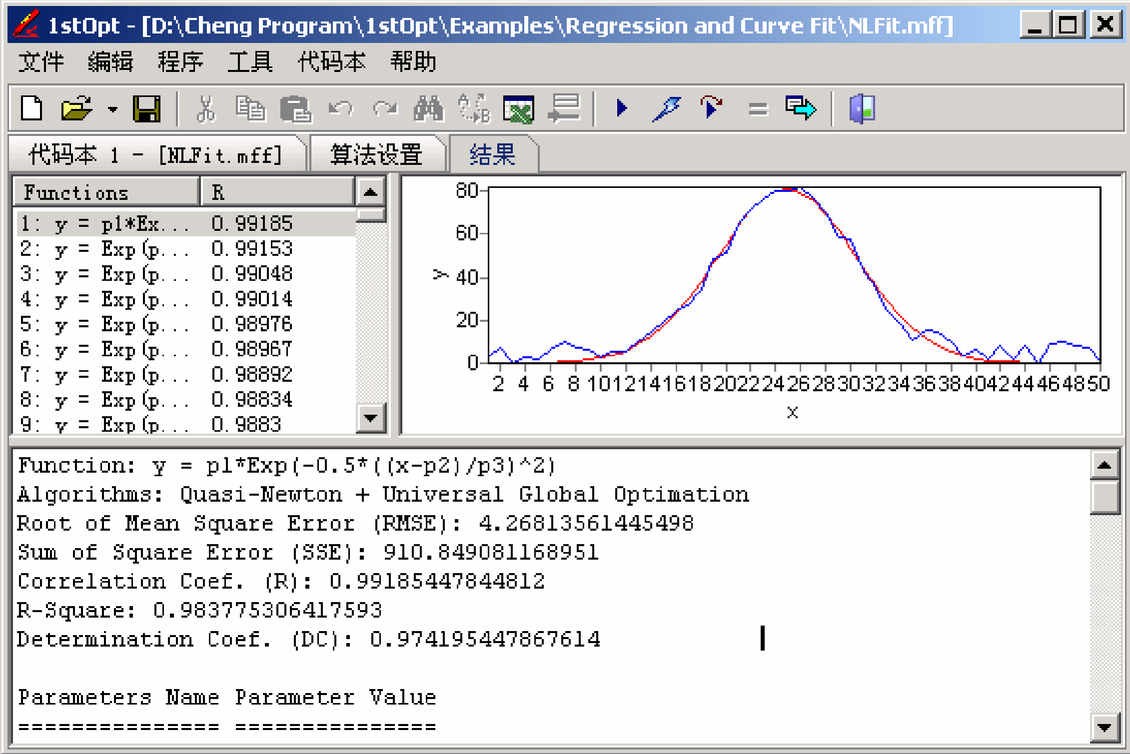


图. 11 非线性拟合示图

例 2. 2. 3：多变量多模型公式非线性回归

此例有两个自变量，两个因变量，两个回归模型共享部分参数。

回归模型 1： $y_1 = a + b\cdot(x_1 + x_2)^c + d\cdot\ln(x_1 + x_2)$ (16)

回归模型 2： $y_2 = a + b\cdot(x_1\cdot x_2)^c$ (17)

1stOpt 代码：

快捷模式：

```
Parameters a, b, c, d;
Variable x1, x2, y1, y2;
SharedModel;
Function y1 = a+b*(x1+x2)^c+d*LN(x1+x2);
         y2 = a+b*(x1*x2)^c;
Data;
1       4.8000    14.59212103     2.25629772
2       5.7246    16.65184769     2.431782548
3       6.3616    18.03897085     2.586695244
.       .         .               .
.       .         .               .
40      12.8970   30.86000633     6.241595159
41      12.9870   31.01610057     6.322904271
42      13.0754   31.16901097     6.403749069
41      12.9870   31.01610057     6.322904271
42      13.0754   31.16901097     6.403749069
```

编程模式：

```
Parameters a, b, c, d;
Variable x1, x2, y1[Output], y2[Output];
StartProgram;
var i:integer;
begin
    for i := 0 to DataLength - 1 do begin
        y1[i] := a+b*(x1[i]+x2[i])^c+d*LN(x1[i]+x2[i]);
        y2[i] := a+b*(x1[i]*x2[i])^c;
    end;
end;
EndProgram;
Data;
1       4.8000    14.59212103      2.25629772
2       5.7246    16.65184769      2.431782548
3       6.3616    18.03897085      2.586695244
.       .         .                .
.       .         .                .
40      12.8970   30.86000633      6.241595159
41      12.9870   31.01610057      6.322904271
42      13.0754   31.16901097      6.403749069
```

结果：

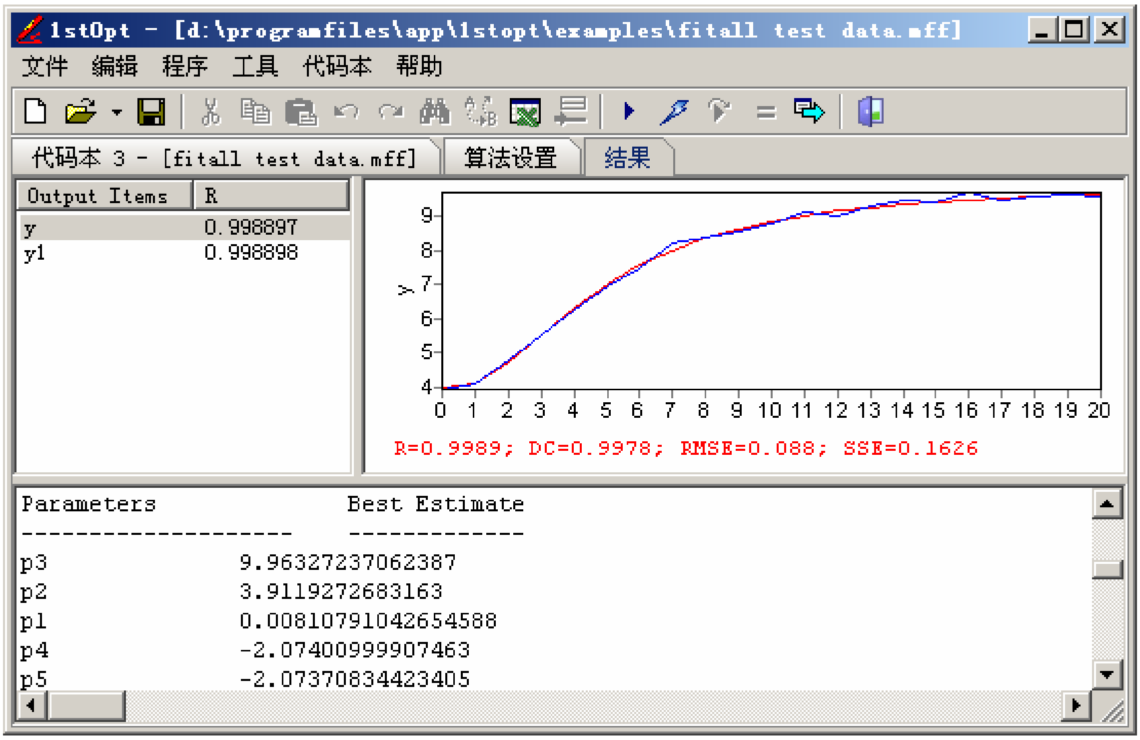


图.12 非线性拟合示图

例 2. 2. 4：隐函数非线性回归

在有些情况下，非线性回归的模型式为隐函数，如下式：

$$
\begin{aligned}
&-(x-p_4)\cdot\sin(p_3)+(y-p_5)\cdot\cos(p_3)+p_5 = \\
&p_1+p_2\cdot\sin(p_3)\cdot\ln\big((x-p_4)*\cos(p_3)+(y-p_5)\cdot\sin(p_3)+p_4\big)
\end{aligned} \tag{18}
$$

其中，x 为自变量，y 为因变量。对于此类问题，1stOpt 可轻易处理。

1stOpt 代码：

```
Variable x, y;
Function -(x-p4)*sin(p3)+(y-p5)*cos(p3)+p5 =
          p1+p2*sin(p3)*ln((x-p4)*cos(p3)+(y-p5)*sin(p3)+p4);
Data;
//x       y
0.0000    262.5079
1.0000    262.8300
2.0000    263.1452
3.0000    263.4532
```

结果：

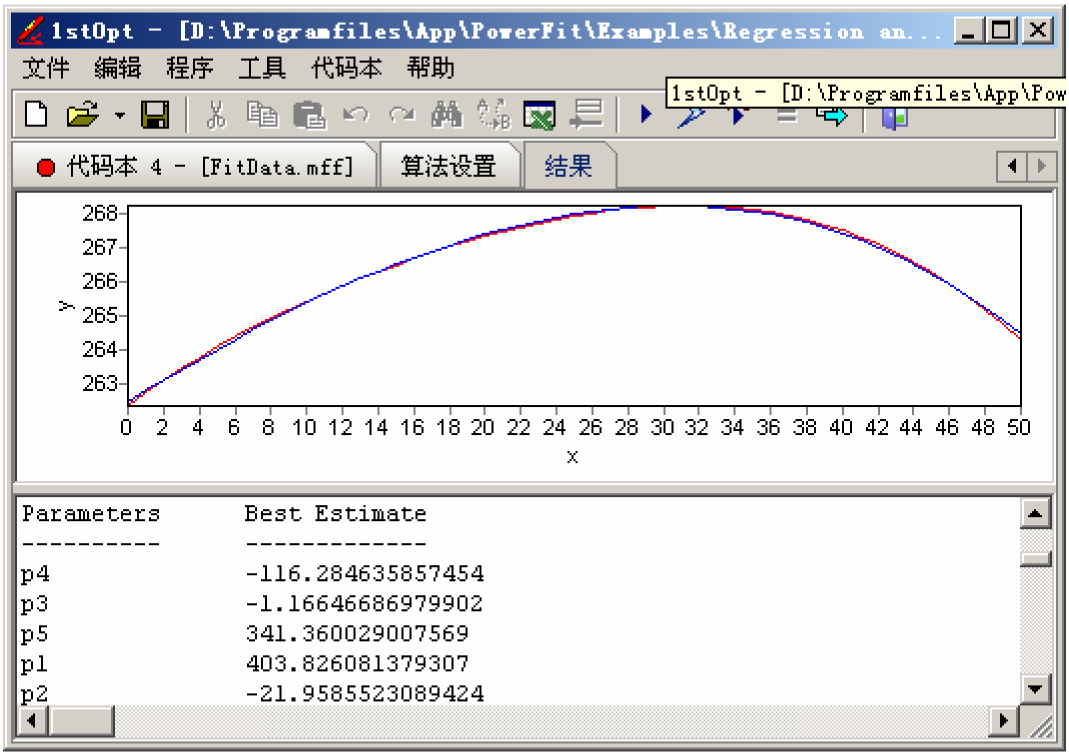


图. 13 非线性拟合示图

例 2．2．5：渐开线渐隐函数非线性回归

已知渐开线参数方程及数据如下：

$$\begin{cases} x = R\cdot\left[\cos(t)+(t-B)\cdot\sin(t)\right]+x_0 \\ y = R\cdot\left[\sin(t)-(t-B)\cdot\cos(t)\right]+y_0 \end{cases} \qquad (19)$$

其中，t是自变量，R ，B，x0，y0 为待求渐开线方程参数。

表 5. 数据

序号	X	Y
1	15.5910	86.2142
2	24.6601	83.7114
3	33.3732	80.2618
4	49.3445	70.7216
5	62.7992	58.0891
6	68.3962	50.8058
7	73.1584	42.9946

8	79.9955	26.1718
9	83.0531	8.4225

此题难点是不知到对应于 x 及 y 的自变量 t 值。运用 1stOpt，此题有两种解法：

✧ 方法 1：将未知 t 变量值当作参数来求，即有 9 个 t 变量参数及 R ，B，x0，y0 共 13 个参数。优化目标函数是最小化下列函数：

$$\min: f=\sum_{i=1}^{9}(y_i-y_i^{'})^2+\sum_{i=1}^{9}(x_i-x_i^{'})^2$$
$$=\sum_{i=1}^{9}(y_i-(R\cdot[\sin(t)-(t-B)\cdot\cos(t)]+y_0\ ))^2+\qquad(20)$$
$$\sum_{i=1}^{9}(x_i-(R\cdot[\cos(t)+(t-B)\cdot\sin(t)]+x0))^2$$

1stOpt 代码 ：

```
Parameter R ,B,x0,y0;
Parameter t(1:9);
DataSet;
        X, Y =
            15.5910     86.2142
            24.6601     83.7114
            33.3732     80.2618
            49.3445     70.7216
            62.7992     58.0891
            68.3962     50.8058
            73.1584     42.9946
            79.9955     26.1718
            83.0531     8.4225
EndDataSet;
MinFunction Sum(i=1:9)((-X[i]+R*(COS(t[i])+(t[i]-B)*SIN(t[i]))+X0)^2)+
            Sum(i=1:9)((-Y[i]+R*(SIN(t[i])-(t[i]-B)*COS(t[i]))+Y0)^2);
```

结果：R = 3.1995227, B = -24.438500, x0 = -0.000329680, y0 = 0.000374680
t1 = 2.926154, t2 = 2.8184350, t3 = 2.710714, t4 = 2.495275, t5 = 2.2798359
t6 = 2.1721167, t7 = 2.064396, t8 = 1.848957, t9 = 1.633520
该种解法虽能得到正解，但当数据较多时，如 500 组数据，即有 500 个 t 值要被当作参数来求，如此多的参数量将会大大增加求解的难度，况且中间变量 t 值也并非我们想获得的值。

✧ 方法 2：将公式进行变换，消除 t 变量

由公式

$$\begin{cases}x=R\cdot\left[\cos(t)+(t-B)\cdot\sin(t)\right]+x_0\\ y=R\cdot\left[\sin(t)-(t-B)\cdot\cos(t)\right]+y_0\end{cases}\qquad(21)$$

变换得

$$\frac{(x-x_0)^2}{R^2}+\frac{(y-y_0)^2}{R^2}=1+(t-B)^2\qquad(22)$$

即

$$t=\sqrt{\frac{(x-x_0)^2}{R^2}+\frac{(y-y_0)^2}{R^2}-1}+B\qquad(23)$$

目标函数如方法 1

$$f=\sum_{i=1}^{9}(y_i-(R\cdot[\sin(t)-(t-B)\cdot\cos(t)]+y_0))^2+\sum_{i=1}^{9}(x_i-(R\cdot[\cos(t)+(t-B)\cdot\sin(t)]+x0))^2 \qquad (24)$$

再将上述 t 代入目标函数用于消除 t。主意关键词 ConstStr。

1stOpt 代码：

```
ConstStr t = B+Sqrt(1/R^2*((X[i]-X0)^2+(Y[i]-Y0)^2)-1);
Parameter R ,B,X0,Y0;
DataSet;
          X, Y =
            15.5910      86.2142
            24.6601      83.7114
            33.3732      80.2618
            49.3445      70.7216
            62.7992      58.0891
            68.3962      50.8058
            73.1584      42.9946
            79.9955      26.1718
            83.0531      8.4225
EndDataSet;
MinFunction Sum(i=1:9)((-X[i]+R*(COS(t)+(t-B)*SIN(t))+X0)^2)+
            Sum(i=1:9)((-Y[i]+R*(SIN(t)-(t-B)*COS(t))+Y0)^2);
```

结果：R = 3.199541, B = 0.694399, x0 = -0.0003171, y0 = 0.00035955

例 2．2．6：有约束的非线性回归

回归公式及数据如下：

公式

$$y=\frac{a+b\cdot x}{1+c\cdot x+d\cdot x^2} \qquad (25)$$

数据：

X	1	2	8	12	17	21	24
Y	1	2	3	6	6	4	4

与一般拟合问题不同之处是，该例拟合后的曲线必须通过点 1 和 3，即点(x_1, y_1)=（1，1），(x_2, y_2)=（8，3）；此约束条件即：

$$y_1=\frac{a+b\cdot x_1}{1+c\cdot x_1+d\cdot {x_1}^2} \quad 及 \quad y_2=\frac{a+b\cdot x_2}{1+c\cdot x_2+d\cdot {x_2}^2} \qquad (26)$$

1stOpt 代码：

```
Constant x1=1,  y1=1， x2=8, y2=3;
DataSet;
           x, y=
            1           1
            2           2
            8           3
            12          6
            17          6
            21          4
            24          4
EndDataSet;
MinFunction Sum(i=1:7)((y[i]-(a+b*x[i])/(1+c*x[i]+d*x[i]^2))^2);
            y1=(a+b*x1)/(1+c*x1+d*x1^2);
            y2=(a+b*x2)/(1+c*x2+d*x2^2);
```

结果：a = 0.861917，b = 0.035839, c = -0.105828, d = 0.0035860

如果无约束条件，本例答案为：a = 1.011742, b = 0.056511,　c = -0.103843, d: = 0.003781868

例 2．2．7：峰值拟合

拟合模型公式如下：

$$y = p_1 \cdot \exp\left(-2.77 \cdot \left(\frac{x - p_2}{p_3}\right)^2\right) + p_1 \cdot \exp\left(-2.77 \cdot \left(\frac{x - 5 \cdot p_2}{p_4}\right)^2\right)$$
$$+ a_0 \cdot \exp\left(-0.5 \cdot \left(\frac{x - a_1}{a_2}\right)^2\right) + 2 \cdot a_0 \cdot \exp\left(-0.5 \cdot \left(\frac{x - 2 \cdot a_1}{a_2}\right)^2\right) \tag{27}$$

其中，p_1 至 p_4，a_0 至 a_2 为待求参数

表.6：峰值拟合数据

序号	X 值	Y 值	序号	X 值	Y 值	序号	X 值	Y 值	序号	X 值	Y 值
1	0.0000	0.0169	37	10.2857	41.2603	73	20.5714	25.6456	109	30.8571	151.0844
2	0.2857	0.0294	38	10.5714	38.4826	74	20.8571	23.5434	110	31.1429	135.0494
3	0.5714	0.0540	39	10.8571	41.0614	75	21.1429	21.6426	111	31.4286	124.1520
4	0.8571	0.0832	40	11.1429	42.8718	76	21.4286	20.5469	112	31.7143	128.2794
5	1.1429	0.1769	41	11.4286	52.4247	77	21.7143	21.0379	113	32.0000	121.8187
6	1.4286	0.3010	42	11.7143	54.6309	78	22.0000	23.0230	114	32.2857	127.7168
7	1.7143	0.4822	43	12.0000	58.5655	79	22.2857	23.3786	115	32.5714	109.3237
8	2.0000	0.8909	44	12.2857	57.6910	80	22.5714	22.6132	116	32.8571	98.8645
9	2.2857	1.4673	45	12.5714	62.2335	81	22.8571	28.5862	117	33.1429	101.0413
10	2.5714	2.6091	46	12.8571	63.6598	82	23.1429	29.2285	118	33.4286	95.4619
11	2.8571	3.9487	47	13.1429	74.5593	83	23.4286	34.6253	119	33.7143	82.9761
12	3.1429	6.1861	48	13.4286	76.3657	84	23.7143	40.5241	120	34.0000	108.2798
13	3.4286	10.2854	49	13.7143	76.8343	85	24.0000	43.5328	121	34.2857	173.0225
14	3.7143	14.1971	50	14.0000	76.4166	86	24.2857	44.4078	122	34.5714	125.9043
15	4.0000	20.8154	51	14.2857	73.9337	87	24.5714	55.3422	123	34.8571	55.3894
16	4.2857	27.9622	52	14.5714	83.6326	88	24.8571	63.8790	124	35.1429	48.4749
17	4.5714	34.1893	53	14.8571	79.5735	89	25.1429	70.3837	125	35.4286	40.5538
18	4.8571	50.9944	54	15.1429	77.3333	90	25.4286	64.7925	126	35.7143	32.5134
19	5.1429	52.8896	55	15.4286	82.7579	91	25.7143	84.1060	127	36.0000	30.7203
20	5.4286	74.7148	56	15.7143	74.8436	92	26.0000	85.2346	128	36.2857	26.0722
21	5.7143	78.2739	57	16.0000	70.2986	93	26.2857	96.3957	129	36.5714	21.6562
22	6.0000	94.8341	58	16.2857	69.1072	94	26.5714	101.5065	130	36.8571	19.8353
23	6.2857	97.1913	59	16.5714	62.9654	95	26.8571	101.5635	131	37.1429	16.4291
24	6.5714	106.9074	60	16.8571	63.9911	96	27.1429	125.4770	132	37.4286	14.3602
25	6.8571	110.5764	61	17.1429	63.3608	97	27.4286	123.0295	133	37.7143	11.6957
26	7.1429	99.7094	62	17.4286	54.3341	98	27.7143	116.8548	134	38.0000	8.8610
27	7.4286	95.2368	63	17.7143	54.7983	99	28.0000	132.5786	135	38.2857	7.2047
28	7.7143	100.9800	64	18.0000	54.1602	100	28.2857	146.0718	136	38.5714	6.0873
29	8.0000	95.3051	65	18.2857	50.5028	101	28.5714	152.3447	137	38.8571	5.1435
30	8.2857	84.7236	66	18.5714	46.6765	102	28.8571	149.5277	138	39.1429	4.1757
31	8.5714	72.5028	67	18.8571	37.3148	103	29.1429	149.2917	139	39.4286	3.0479
32	8.8571	63.7613	68	19.1429	35.4733	104	29.4286	167.8167	140	39.7143	2.7644
33	9.1429	53.8688	69	19.4286	31.4011	105	29.7143	140.3957	141	40.0000	2.1079
34	9.4286	45.3433	70	19.7143	31.6141	106	30.0000	146.2560			
35	9.7143	47.6868	71	20.0000	29.3181	107	30.2857	140.2877			
36	10.0000	39.2611	72	20.2857	29.2092	108	30.5714	150.3306			

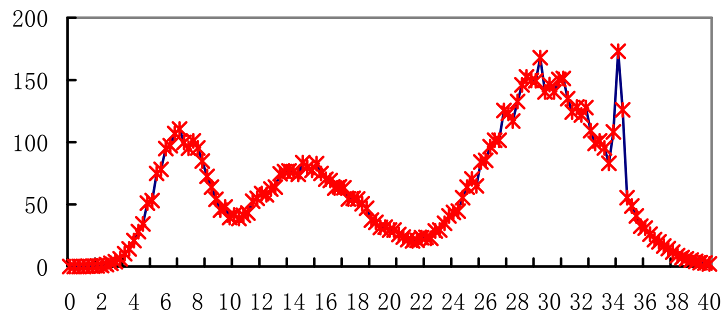


图.14: 多峰曲线拟合

如上图示，共有 4 个峰，最后一个峰仅有三个数据点。对任何其它拟合软件来说，求得正解都将很困难，因为很难给出合适的参数初始值。而 1stOpt 则无需给出初始值，将自动求出最优解，成功率高于 90%。

1stOpt 代码：

```
Function y = p1*exp(-2.77*((x-p2)/p3)^2)+p1*exp(-2.77*((x-5*p2)/p4)^2)+
              a0*exp(-0.5*((x-a1)/a2)^2)+2*a0*exp(-0.5*((x-2*a1)/a2)^2);
Data;
//x        y
0.0000     0.0169
0.2857     0.0294
0.5714     0.0540
.
.
.
39.4286    3.0479
39.7143    2.7644
40.0000    2.1079
```

2．3：任意形式，线性，非线性方程或方程组求解

例 2．3．1：求下列方程组的根：

$$\begin{cases} (x-0.3)^{y^z}+\dfrac{x}{y\cdot z}-x\cdot y\cdot \sin(z)+(x+y-z)^{\cos(x-1)}=1 \\ (y-0.2)^{z^x}+\dfrac{y}{z\cdot x}-y\cdot z\cdot \sin(x)+(y+z-x)^{\cos(y-2)}=2 \\ (z-0.1)^{x^y}+\dfrac{z}{x\cdot y}-z\cdot x\cdot \sin(y)+(z+x-y)^{\cos(z-3)}=3 \end{cases} \quad (28)$$

1stOpt 代码：

```
Parameter x, y, z;
Function  (x-0.3)^y^z+x/y/z-x*y*sin(z)+(x+y-z)^cos(x-1) = 1;
          (y-0.2)^z^x+y/z/x-y*z*sin(x)+(y+z-x)^cos(y-2) = 2;
          (z-0.1)^x^y+z/x/y-z*x*sin(y)+(z+x-y)^cos(z-3) = 3;
```

结果： x ＝0.79390634413219， y ＝ 0.902585377949916， z ＝ 1.21622367662841

例 2．3．2：求下列方程组的根：

$$\begin{cases}\exp(0.1\cdot x_1)-\exp(0.1\cdot x_2)-x_3\left(\exp(-0.1)-\exp(-10\cdot 0.1)\right)=0\\ \exp(0.2\cdot x_1)-\exp(0.2\cdot x_2)-x_3\left(\exp(-0.2)-\exp(-10\cdot 0.2)\right)=0\\ \exp(0.3\cdot x_1)-\exp(0.3\cdot x_2)-x_3\left(\exp(-0.3)-\exp(-10\cdot 0.3)\right)=0\end{cases} \qquad (29)$$

其中，$x_1 \in [-100, 100]$，$x_2 \in [-100, 100]$，$x_3 \in [0.1, 100]$

1stOpt 代码：

```
Parameter x1[-100,100], x2[-100,100], x3[0.1,100];
Function   exp(-0.1*x1)-exp(0.1*x2)-x3*(exp(-0.1)-exp(-10*0.1))=0;
           exp(-0.2*x1)-exp(0.2*x2)-x3*(exp(-0.2)-exp(-10*0.2))=0;
           exp(-0.3*x1)-exp(0.3*x2)-x3*(exp(-0.3)-exp(-10*0.3))=0;
```

结果：x_1 = 1，x_2 = -10，x_3 = 1

2．4：模型参数优化率定

1stOpt 内镶 Pascal/VB 语言，因此可处理任何复杂的模型参数优化问题，而无需外部编译器。尤其是通过引入两个关键词”VapParameter”和”VarConstant”，更可轻易处理一些特殊的参数，常量及多数据文件同时优化问题。

例２．４．１：ＴＡＮＫ模型参数优化率定:

ＴＡＮＫ模型因其结构简单，易懂，应用效果佳而成为水文分析中经常使用的降雨径流模型。图.12是三段联立ＴＡＮＫ模型。降雨为输入，流量Ｑ为输出：

$$Q = Q_1 + Q_3 + Q_5 \qquad (30)$$

使用参数：

- 初期贮留：L_1，L_2，L_3（总数 3），单位：mm
- 贮留能力：S_1, S_2, S_3（总数 3），单位：mm
- 流量系数：K_1, K_2, K_3, K_4, K_5（总数 5）

如果同时使用 8 次降雨径流过程进行参数率定，则初期贮留参数量 L 将变为：３ｘ８＝24

最后待定参数总量为：２４＋３＋５＝３２

在下面代码中，特殊参数”初期贮留”用关键词”VapParameter”来进行定义。

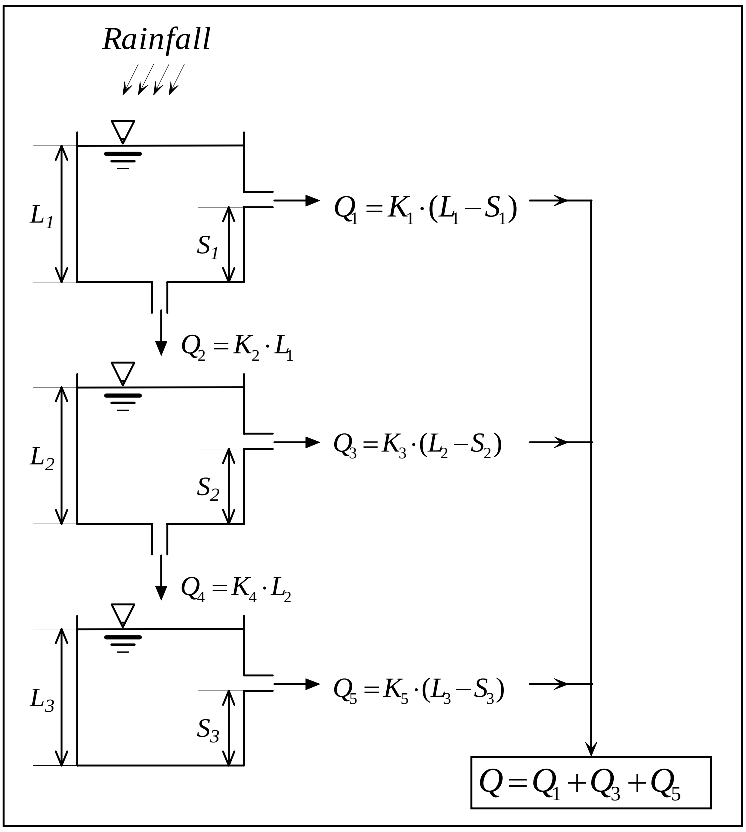


图.15 三层ＴＡＮＫ模型

1stOpt 代码：

```
Variable Rainfall;                          //unit: mm
Variable Runoff;                            //unit: m^3/s
Constant DT = 1;                            //time interval, unit: hr
Constant n = 3;                             //number of rainfall process
Parameter Area [0,3];                       //catchment area, unit: km^2
Parameter K(1:5) [0.0001,1];                //discharge coef.
Parameter S(1:3) [1,100];                   //storage coef.
VarParameter L(1:3) = 3 [0, 40];            //initial storage
StartProgram;
var
   i : integer;
   Q1, Q2, Q3, Q4, Q5: Double;
   SS1, SS2, SS3: Double;
begin
     SS1 := L1;
     SS2 := L2;
     SS3 := L3;
     for i := 0 to DataLength-1 do begin
         //tank 1
         if SS1 > S1 then
            Q1 := K1 * (SS1 - S1)
         else
            Q1 := 0;
         Q2 := K2*SS1;
         SS1 := SS1 + (Rainfall[i] - Q1 - Q2) * DT;
         if SS1 < 0 then SS1 := 0;
         //tank 2
         if SS2 > S2 then
            Q3 := K3*(SS2-S2)
         else
            Q3 := 0;
         Q4 := K4 * SS2;
         SS2 := SS2 + (Q2 - Q4 - Q5) * DT;
         if SS2 < 0 then SS2 := 0;
         //tank 3
         if SS3 > S3 then
            Q5 := K5 * (SS3 - S3)
         else
            Q5 := 0;
         SS3 := SS3 + (Q4 - Q5) * DT;
         if SS3 < 0 then SS3 := 0;
         // change unit from mm -> m^3/s
         Runoff[i] := 3600*(Q1+Q3+Q5)*Area*10/(DT*36);
     end;
end;
EndProgram;

DataFile C:¥Applications¥1stOpt¥yangui_1.csv;
DataFile C:¥Applications¥1stOpt¥yangui_3.csv;
DataFile C:¥Applications¥1stOpt¥yangui_4.csv;
DataFile C:\Applications\1stOpt\yangui_6.csv;
DataFile C:\Applications\1stOpt\yangui_7.csv;
DataFile C:\Applications\1stOpt\yangui_10.csv;
DataFile C:\Applications\1stOpt\yangui_11.csv;
DataFile C:\Applications\1stOpt\yangui_12.csv;
```

结果：平均相关系数 R = 0.9887

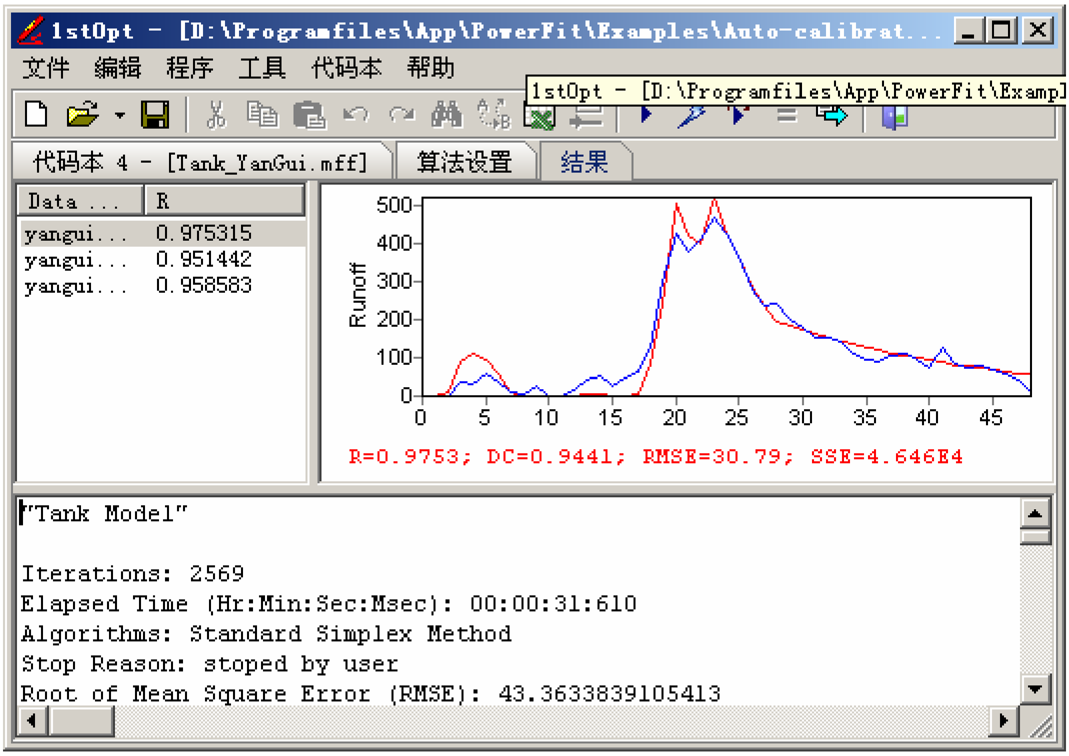


图. 16 TANK 模型优化结果示图

例 2. 4. 2：路面排水水质模型参数优化率定

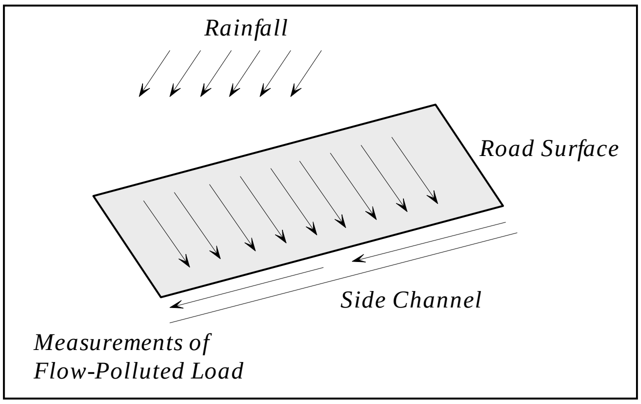


図.17 路面排水试验概要图

调查路面概要如右图.17 示，面积为 995m²，降雨为人工降雨。观测项目为BOD，COD，SS及流量。时间间隔为 10 分钟。共有 6 次实测资料。

路面排水水质模型如下列公式，式(31)为运动式，式(32)为连续式：

$$\begin{cases} L_t = k \cdot S_t \cdot Q_t & (31) \\ S_{t+1} = S_t - \int L_t dt & (32) \end{cases}$$

其中，L_t： t时刻污浊负荷流出量 (g/m²/hr)

k： 污浊负荷量流出系数 (1/mm)

Q_t： t时刻雨水流出强度 (mm/hr)

S_t， S_{t+1}： t と $t+1$ 时刻路面堆积负荷残存量 (g/m²)

使用参数：

本事例中 BOD 与 COD 同时进行优化率定，6 次实测过程数据同时使用，参数如下：

- 初期BOD污浊负荷量：S_{BOD}， 总数 6，单位：g/m²
- 初期COD污浊负荷量：S_{cod}， 总数 6，单位：g/m²
- BOD及COD污浊负荷量流出系数： K_{bod}， K_{cod}， 总数 2，单位：1/mm

最终参数总数：6 + 6 + 2 = 14

1stOpt 代码：

```
Variable Flow;                          //unit: l/min
Variable BOD [Output];                  //unit: mg/m^2/hr
Variable COD [Output];                  //unit: mg/m^2/hr
Parameter k [0,1];                      //unit: 1/mm
VarParameter S0_BOD[1,100,1,]=30;       //unit: mg/m^2
Parameter k1 [0,1];                     //unit: 1/mm
VarParameter S0_COD[1,]=30;             //unit: mg/m^2
Constant Area=995;                      //unit: m^2
Constant dt=10;                         //unit: minutes
StartProgram;
var
    i: integer;
    S, S1: Double;
Begin
    S := S0_BOD;
    S1 := S0_COD;
    for i := 0 to DataLength - 1 do begin
        BOD[i] := k * S * Flow[i] * 60 / Area;
        S := S - BOD[i] * dt / 60;
        if S < 0 then S := 0;
        COD[i] := k1 * S1 * Flow[i] * 60 / Area;
        S1 := S1-COD[i] * dt / 60;
        if S1 < 0 then S1 := 0;
    end;
End;
EndProgram;
```

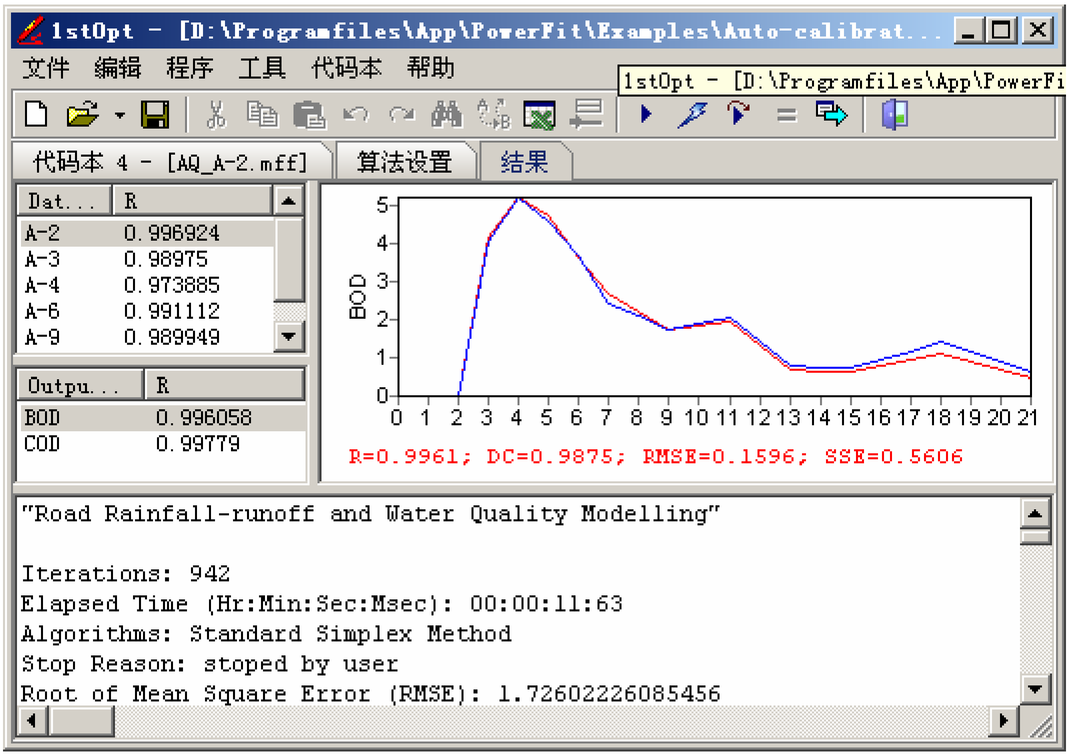


图. 18. 水质模型优化结果示图

例 2．4．3：时系列模型参数优化率定

此案例研究的是大肠杆菌(z)与降雨量(x)及流量(y)间的关系。表.7 是三次实测过程数据，共有 9 次类似数据。因为大肠杆菌不仅与降雨量及流量有关，还受其初始值影响，采用的数学模型公式如下式，其中，p_1至p_7为待定参数，t表示时间。

$$z_t = \frac{p_1 + p_2 \cdot x_t + p_3 \cdot y_t + p_4 \cdot z_{t-1}}{1 + p_5 \cdot x_t + p_6 \cdot y_t + p_7 \cdot z_{t-1}} \qquad (33)$$

表.7 原始数据

过程 1			过程 2			过程 3		
降雨 x	径流 y	大肠杆菌 z	降雨 x	径流 y	大肠杆菌 z	降雨 x	径流 y	大肠杆菌 z
1	1.25	22469.71	1	1.38	21146.99	3	1.23	43453.54
1	1.5	18596.52	1.5	1.62	18290.44	4.5	2.17	28745.68
0.5	1.6	15013.49	1	1.84	14610.85	5.5	3.88	16267.61
1.5	1.44	13300.36	1.5	1.91	12073.21	0.5	5.9	9466.555
1	1.67	10110.15	1	2.02	9667.794	0.5	4.81	8041.477
0.5	1.72	8769.884	1.5	1.99	8021.34	1	3.07	6047.688
1.5	1.58	8531.308	2	2.03	6525.932	0.5	1.99	4509.077
1	1.78	7133.543	2	2.28	4982.795	1.5	1.36	3817.827
1.5	1.82	6892.542	1.5	2.62	3926.618	2.5	1.46	2888.571
1.5	2.07	6322.324	1.5	2.7	3331.667	0	2.11	2008.761
1.5	2.27	5914.068	2.5	2.57	2879.199	0	1.89	2293.048
1	2.4	5581.956	1.5	2.75	2286.745	0	1.35	2710.212
1.5	2.24	5645.432	2.5	2.81	2042.801			
1	2.25	5240.338	1	2.99	1717.063			
1.5	2.09	5288.167	1	2.77	1639.511			
2	2.15	4877.341	1	2.31	1639.707			
0.5	2.46	4216.897	1	1.92	1637.137			
1	2.11	4574.577	0	1.69	1665.447			
0.5	1.86	4426.319	0	1.31	1936.61			
0	1.48	4620.076						
0	1.06	5297.758						

在此例中，每一数据文件的第一行数值被当作初始值，使用关键词”VarConstant”用以描述各次过程的初始大肠杆菌量。1stOpt 代码：

```
Parameter p(1:7);
Variable x, y, z;
VarConstant z0 = [27177.83, 25288.04, 7751.078, 11028.05, 10725.02, 34615.46,
                  22479.53, 18309.19, 44856.28];
StartProgram;
var i        : integer;
    Temz     : Double;
begin
    Temz := z0;
    for i := 0 to Datalength - 1 do begin
        z[i] := (p1+p2*x[i]+p3*y[i]+p4*Temz)/(1+p5*x[i]+p6*y[i]+p7*Temz);
        Temz := z[i];
    end;
end;
EndProgram;
//x          y         z
Data "2000-03-29";   //file 1
//1          1.0100    27177.83
1            1.2500    22469.71
1            1.5000    18596.52
```

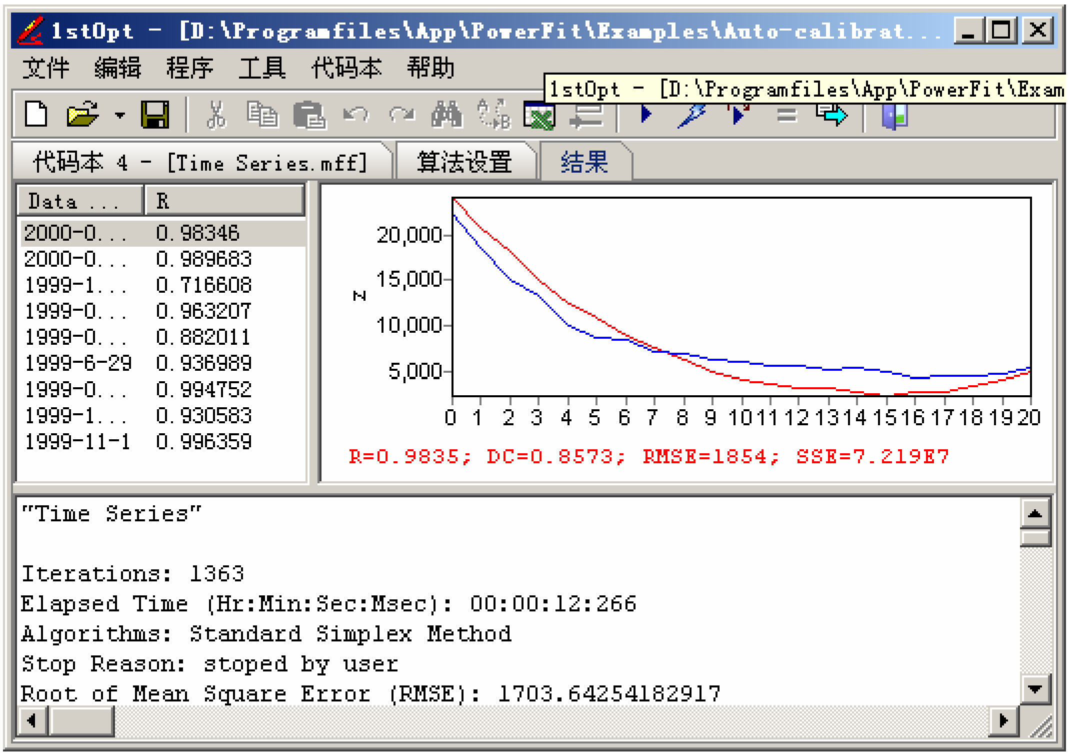


图.19 时系列模型优化结果示图

<u>例 2. 4. 4：降雨水位模型参数优化</u>

已知某观测点两时段t_1及t_2总降雨量及由其引起的水位增量数据如下表.8，同时已知t_1及t_2内每分钟的降雨量如表.9。假如已知每分钟降雨量与每分钟水位增量的模型为指数关系如式（34），试确定参数a及b。

$$h = a \cdot R^b \qquad (34)$$

其中，h每分钟水位增量，R每分钟降雨量。

表.8 降雨量及水位增量

时段	时间间隔（hr.）	降雨量（mm）	水位增量（m）
t_1	1	71.5	30.9524
t_2	1	22.5	0.2547

表.9 每分钟降雨量 R

t_1 时段	5, 0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 1, 0.5, 1, 5, 1, 1, 1, 1, 1, 1.5, 1, 1, 1.5, 2, 1.5, 1, 2, 1.5, 2, 1, 1.5, 2, 2, 2, 2, 2.5, 2, 2.5, 2, 1.5, 2.5, 2, 1.5, 2, 2, 1, 1, 1, 1.5, 1, 1, 1.5, 0.5, 1, 1
t_2 时段	5, 0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0.5, 0.5, 0, 5, 0, 5, 0, 5, 0, 0.5, 0.5, 0, 0.5, 0.5, 5, 0.5, 0, 0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0.5, 0.5, 0, 0.5, 0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0.5, 0, 0.5, 0.5, 0, 0.5

此例求最优解的标准是通过（34）式所算出的t_1及t_2时段每分钟降雨量所引起的水位增量之和应分别等于t_1及t_2时段 1 小时间观测到的水位增量，如图.20 示。目标函数如下：

$$\begin{cases} t_1\text{时段} & \sum_{i=0}^{59} a \cdot R_{1,i}{}^{b} - 30.9524 = 0 \quad (35) \\ t_2\text{时段} & \sum_{i=0}^{59} a \cdot R_{2,i}{}^{b} - 0.2547 = 0 \quad (36) \end{cases}$$

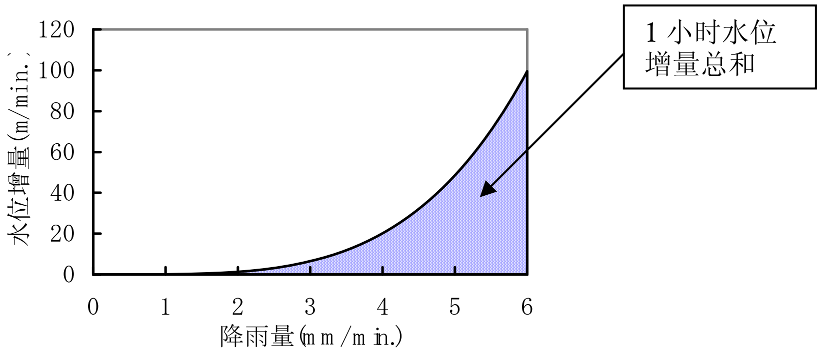


图.20 每分钟降雨水位关系示图

问题现转换为求特殊联立方程组（35）和（36）的根：a 和 b

1stOpt 代码：

```
Constant Rain1(0:59)=[0.5,0,0.5,0.5,0.5,0.5,0.5,0.5,0,0.5,0.5,0.5,1,0.5,0.5, 1,0.5,1,0.5,
                     1,1,1,1,1.5,1,1,1.5,2,1.5,1,2,1.5,2,1,1.5,2,2,2,2,2.5,2,2.5,2,
                     1.5,2.5,2,1.5,2,2,1,1,1,1.5,1,1,1.5,0.5,1,1];
Constant Rain2(0:59)=[0.5,0,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0,0.5,0.5,0,0.5,0,0.5,
                     0,0.5,0,0.5,0.5,0,0.5,0.5,0.5,0.5,0,0,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0,0.5,
                     0.5,0,0.5,0,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0,0.5,0,0.5,0.5,0,0.5];
Parameter a, b;
Function    Sum(i = 0:59)(a*Rain1[i]^b)-30.9524 = 0;
            Sum(i = 0:59)(a*Rain2[i]^b)-0.2547 = 0;
```

结果：a = 0.086476, b = 3.93344。

例 2．4．5：时系列自回归模型

已知某河段流量过程如下表.10 及图.21

表.10 河段流量过程

时段 t	流量 Q	时段 t	流量 Q
1	642	17	3200
2	570	18	3440
3	523	19	2970
4	512	20	2430
5	547	21	2020
6	579	22	1860
7	796	23	1740
8	1020	24	1660
9	1230	25	1620
10	1240	26	1730
11	1110	27	1790
12	954	28	1600
13	850	29	1400
14	1070	30	1190
15	1470	31	948
16	2360		

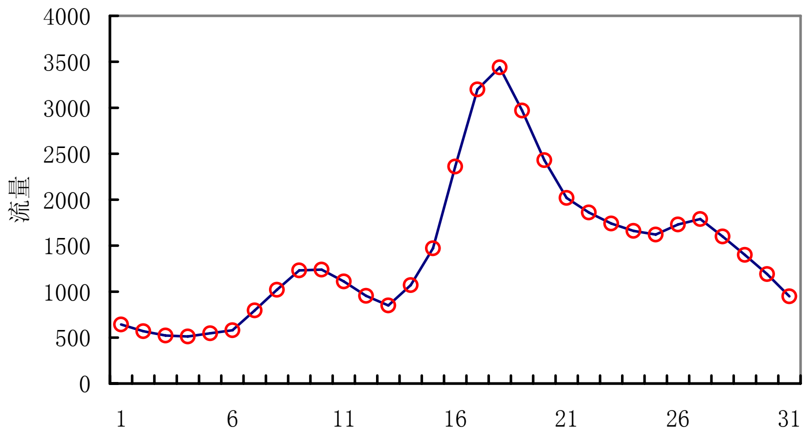


图.21 实测流量过程

假设流量 Q 有如下线性自回归关系式：

$$Q_t = p_1 \cdot Q_{t-1} + p_2 \cdot Q_{t-2} + p_3 \cdot Q_{t-3} + p_4 \qquad (37)$$

其中，p_1至p_4为待求参数；Q_t，Q_{t-1}，Q_{t-2}，Q_{t-3}分别为 t，$t-1$，$t-2$，$t-3$ 时刻流量。

此例的目标函数是最小化计算与实测流量间的残差，即：

$$\text{Min} \sum_{i=1}^{n} \left(ObsQ_i - CalQ_i \right)^2 \qquad (38)$$

其中，ObsQ，CalQ 分别是观测和计算的流量值；n = 31 - 3 = 28；

1stOpt 代码 ：

```
Constant n=27;
Parameters p(1:4);
Constant Q(1:3) = [642,570,523];
Constant ObsQ(0:n) = [512,547,579,796,1020,1230,1240,1110,954,850,1070,1470,2360, 3200, 3440,
                      2970,2430,2020,1860,1740,1660,1620,1730,1790,1600,1400,1190,948];
StartProgram;
var i: integer;
    temSSE, CalQ: double;
    tq1, tq2, tq3: double;
begin
     temSSE := 0;
     tq1 :=Q1;
     tq2 := Q2;
     tq3 := Q3;
     for i := 0 to n do begin
         CalQ := p1*tq1+p2*tq2+p3*tq3+p4;
         if CalQ < 0 then CalQ := 0;
         tq1 := tq2;
         tq2 := tq3;
         tq3 := CalQ;
         temSSE := temSSE + sqr(CalQ-ObsQ[i]);
     end;
     FunctionResult := sqrt(temSSE/28);
end;
EndProgram;
```

结果：p1 = 1.1578229980, p2 = -3.25408590, p3 = 3.1013431829, p4 = -5.271458666

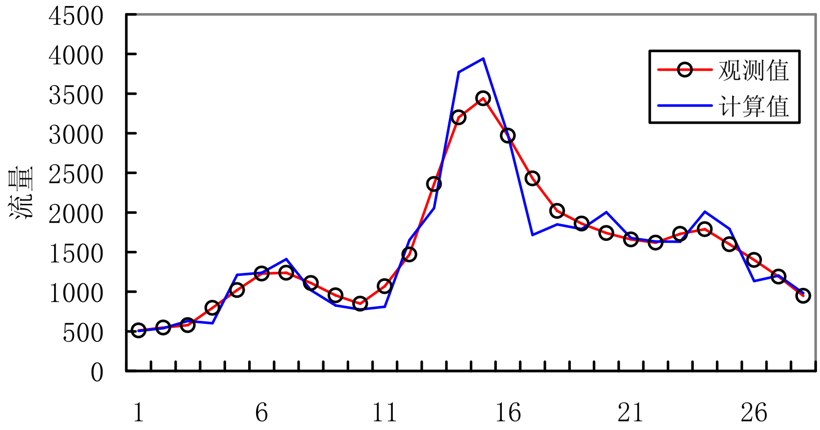


图.22 实测与模拟流量过程

2．5：组合优化问题

1stOpt 亦可用与解决组合优化问题。自行开发的最大继承法（MIO）在解决该类问题时，比其它诸如遗传算法，模拟退火及禁忌算法等表现更优。

<u>例 2．5．1：ＴＳＰ问题</u>

ＴＳＰ问题是非常著名的组合优化问题：有Ｎ个城市，从某一城市出发，每个城市访问一次，最后回到起始城市，试求最短距离的访问路线。下面以某地１５个城市为例。

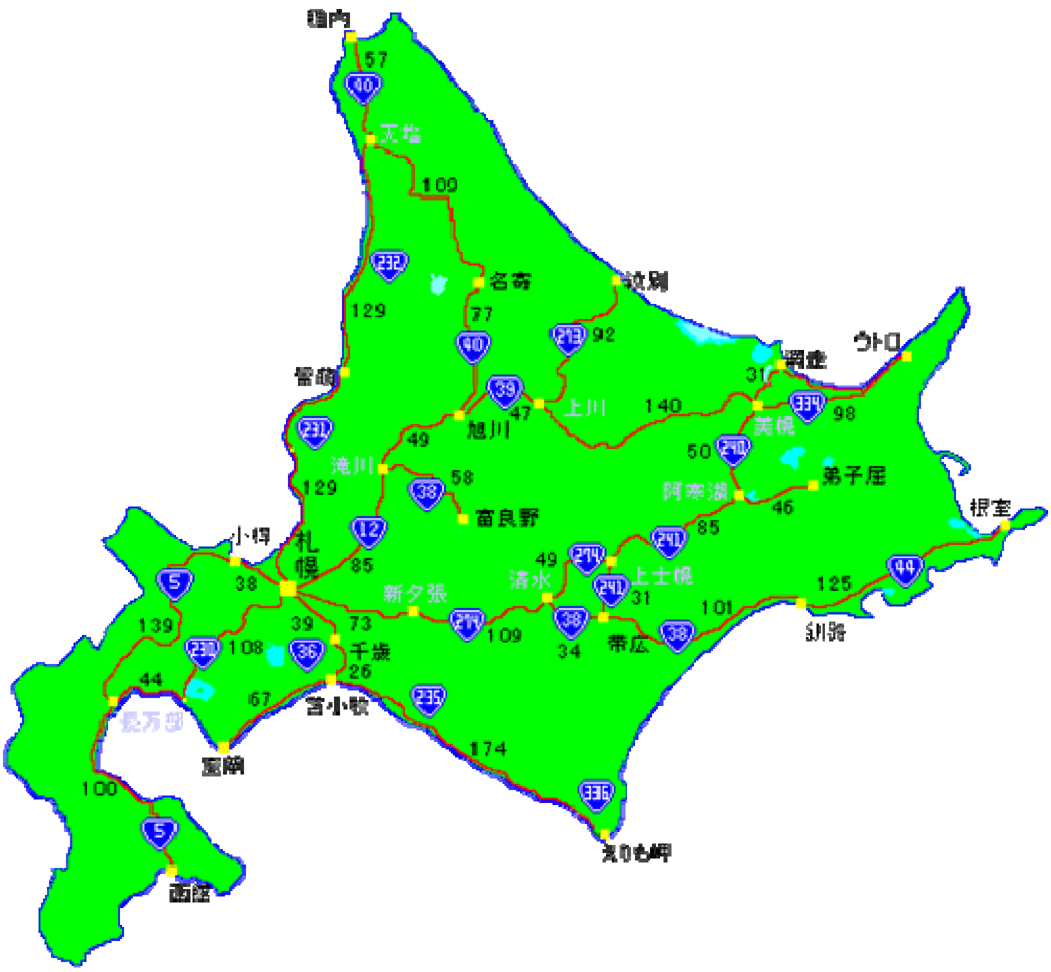


图.23 １５个城市地图

表.11　１５个城市距离表：

	札幌	涵館	旭川	弟子屈	苫小牧	千歳	釧路	稚内	帯広	ウトロ	小樽	富良野	根室	網走	留萌
札幌	0	261	134	364	66	39	328	315	164	418	38	143	453	349	129
涵館	261	0	395	625	255	251	591	587	425	679	244	420	715	615	389
旭川	134	395	0	242	200	173	289	243	177	281	157	57	386	215	78
弟子屈	364	625	242	0	356	360	72	404	165	107	402	285	116	77	320
苫小牧	66	255	200	356	0	25	304	381	191	463	104	168	429	390	201
千歳	39	251	173	360	25	0	308	365	195	467	77	166	432	394	168
釧路	328	591	289	72	304	308	0	479	114	182	366	233	125	152	367
稚内	315	587	243	404	381	365	479	0	420	410	330	300	520	327	182
帯広	164	425	177	165	191	195	114	420	0	272	202	120	239	199	255
ウトロ	418	679	281	107	463	467	182	410	272	0	438	338	165	83	359
小樽	38	244	157	402	104	77	366	330	202	438	0	214	491	387	148
富良野	143	420	57	285	168	166	233	300	120	338	214	0	357	272	131
根室	453	715	386	116	429	432	125	520	239	165	491	357	0	193	464
網走	349	615	215	77	390	394	152	327	199	83	387	272	193	0	293
留萌	129	389	78	320	201	168	367	182	255	359	148	131	464	293	0

1stOpt 代码：

```
Constant n = 15;
Constant Distance(0:n-1, 0:n-1) = [0,261,134,364,66,39,328,315,164,418,38,143,453,349,129,
                                  261,0,395,625,255,251,591,587,425,679,244,420,715,615,389,
                                  134,395,0,242,200,173,289,243,177,281,157,57,386,215,78,
                                  364,625,242,0,356,360,72,404,165,107,402,285,116,77,320,
                                  66,255,200,356,0,25,304,381,191,463,104,168,429,390,201,
                                  39,251,173,360,25,0,308,365,195,467,77,166,432,394,168,
                                  328,591,289,72,304,308,0,479,114,182,366,233,125,152,367,
                                  315,587,243,404,381,365,479,0,420,410,330,300,520,327,182,
                                  164,425,177,165,191,195,114,420,0,272,202,120,239,199,255,
                                  418,679,281,107,463,467,182,410,272,0,438,338,165,83,359,
                                  38,244,157,402,104,77,366,330,202,438,0,214,491,387,148,
                                  143,420,57,285,168,166,233,300,120,338,214,0,357,272,131,
                                  453,715,386,116,429,432,125,520,239,165,491,357,0,193,464,
                                  349,615,215,77,390,394,152,327,199,83,387,272,193,0,293,
                                  129,389,78,320,201,168,367,182,255,359,148,131,464,293,0];
Parameters Cities(0:n-1)[0,n-1];
Exclusive = True;
Minimum = True;
StartProgram;
Var TemSum : Double;
    i : integer;
Begin
    TemSum := 0;
    for i := 0 to n-2 do
        TemSum := TemSum + Distance[Cities[i], Cities[i+1]];
    FunctionResult := TemSum + Distance[Cities[n-1], Cities[0]];
End;
EndProgram;
```

结果：最短距离为２０７６ＫＭ,访问路径为：

札幌 ⇒ 旭川 ⇒ 富良野 ⇒ 帯広 ⇒ 釧路 ⇒ 根室 ⇒ 弟子屈 ⇒ ウトロ ⇒ 網走 ⇒ 稚内 ⇒ 留萌 ⇒ 小樽 ⇒ 涵館 ⇒ 苫小牧 ⇒ 千歳 ⇒ 札幌

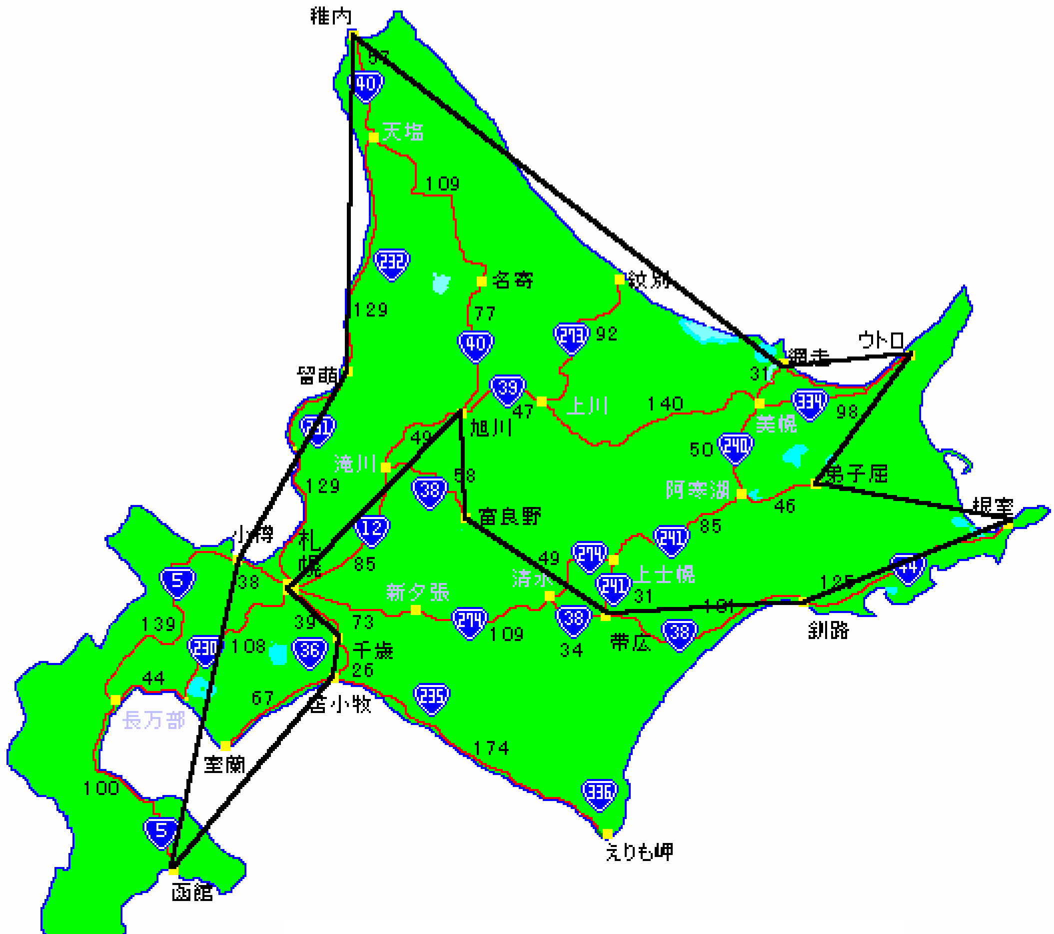


图.24　1 5 个城市最短路径地图

表.12 魔法矩阵问题

	X_1	X_2	X_3	X_4	34
	X_5	X_6	X_7	X_8	34
	X_9	X_{10}	X_{11}	X_{12}	34
	X_{13}	X_{14}	X_{15}	X_{16}	34
34	34	34	34	34	34

例 2．5．2：魔法矩阵问题

将 1 至 16 个数排列成 4 行 4 列矩阵如表.12，每个数只用一次，如何排列使其每行，每列及两对角 4 数之和均分别等于 34？也即求X_1至X_{16}的值.

1stOpt 代码：

```
Constant S = 34;
Parameters x(1:16)[1,16,0];
Exclusive = true;
Function x1 + x2 + x3 + x4        = S;
         x5 + x6 + x7 + x8        = S;
         x9 + x10 + x11 + x12     = S;
         x13 + x14 + x15 + x16 = S;
         x1 + x5 + x9 + x13       = S;
         x2 + x6 + x10 + x14      = S;
         x3 + x7 + x11 + x15      = S;
         x4 + x8 + x12 + x16      = S;
         x1 + x6 + x11 + x16      = S;
         x4 + x7 + x10 + x13      = S;
```

表.13 魔法矩阵结果

	2	14	15	3	34
	7	11	10	6	34
	9	5	8	12	34
	16	4	1	13	34
34	34	34	34	34	34

计算结果如下：$x_1=2$，$x_2=14$，$x_3=15$，$x_4=3$，$x_5=7$，$x_6=11$，$x_7=10$，$x_8=6$，$x_9=9$，$x_{10}=5$，$x_{11}=8$，$x_{12}=12$，$x_{13}=16$，$x_{14}=4$，$x_{15}=1$，$x_{16}=13$

例 2．5．3：最短路径问题

图.23 表示从起点A到终点E之间各点的距离。求 A 到E的最短路径。

此问题也乃经典的组合优化问题。不同与传统的用动态线性规划方法求解，1stOpt 使用全局优化来寻优。

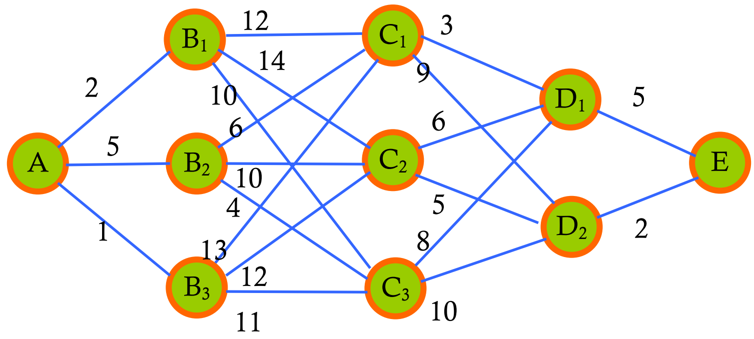


图. 25 最短路径示意图

表. 13 城市距离表（无联结城市间的距离用虚拟值 1000 表示）：

	序号	0	1	2	3	4	5	6	7	8	9
序号		A	B1	B2	B3	C1	C2	C3	D1	D2	E
0	A	0	2	5	1	1000	1000	1000	1000	1000	1000
1	B1	2	0	1000	1000	12	14	10	1000	1000	1000
2	B2	5	1000	0	1000	6	10	4	1000	1000	1000
3	B3	1	1000	1000	0	13	12	11	1000	1000	1000
4	C1	1000	12	6	13	0	1000	1000	3	9	1000
5	C2	1000	14	10	12	1000	0	1000	6	5	1000
6	C3	1000	10	4	11	1000	1000	0	8	10	1000
7	D1	1000	1000	1000	1000	3	6	8	0	1000	5
8	D2	1000	1000	1000	1000	9	5	10	1000	0	2
9	E	1000	1000	1000	1000	1000	1000	1000	5	2	0

1stOpt 代码

```
Constant n = 8;
Parameters Cities(1:n)[1,n,0];
Constant Dis(0:n+1,0:n+1) = [0,2,5,1,1000,1000,1000,1000,1000,1000,
                             2,0,1000,1000,12,14,10,1000,1000,1000,
                             5,1000,0,1000,6,10,4,1000,1000,1000,
                             1,1000,1000,0,13,12,11,1000,1000,1000,
                             1000,12,6,13,0,1000,1000,3,9,1000,
                             1000,14,10,12,1000,0,1000,6,5,1000,
                             1000,10,4,11,1000,1000,0,8,10,1000,
                             1000,1000,1000,1000,3,6,8,0,1000,5,
                             1000,1000,1000,1000,9,5,10,1000,0,2,
                             1000,1000,1000,1000,1000,1000,1000,5,2,0];
Minimum = True;
StartProgram;
var i : integer;
    temD : Double;
begin
    temD := 0;
    for i := 1 to n - 1 do
        temD := temD + Dis[Cities[i],Cities[i+1]];
    FunctionResult := temD + Dis[0,Cities[1]] + Dis[Cities[n],n+1];
end;
EndProgram;
```

结果：最短路径值为 19，路线图为：0 -> 2 -> 4 -> 7 -> 9 也既：A -> B2 -> C1 -> D1 ->E

2．6：二维/三维分析及预测验证

在优化计算完成后，可进行二维/三维分析及数据预测验证。如下图.24，二维/三维分析主要图形化显示目标函数值与一个或两个参数变量之间的关系。预测验证则是曲线拟合计算结束后，用于计算未知点，如由未知自变量值计算因变量值等，如图.25

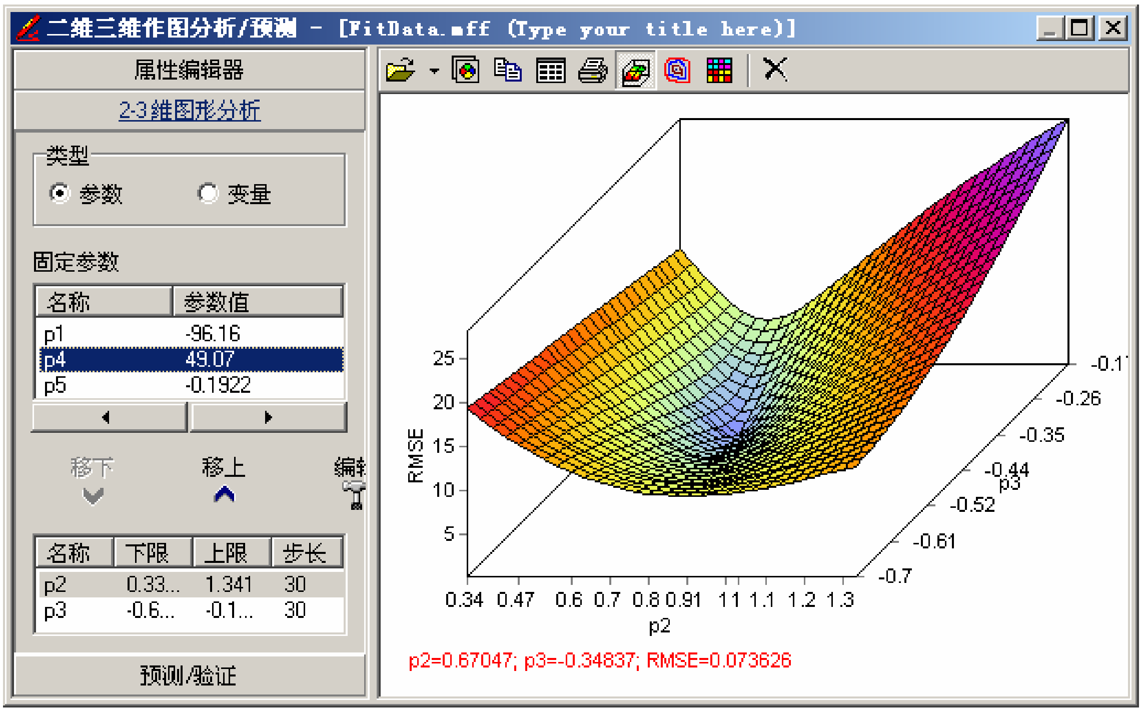


图. 26 二维/三维分析

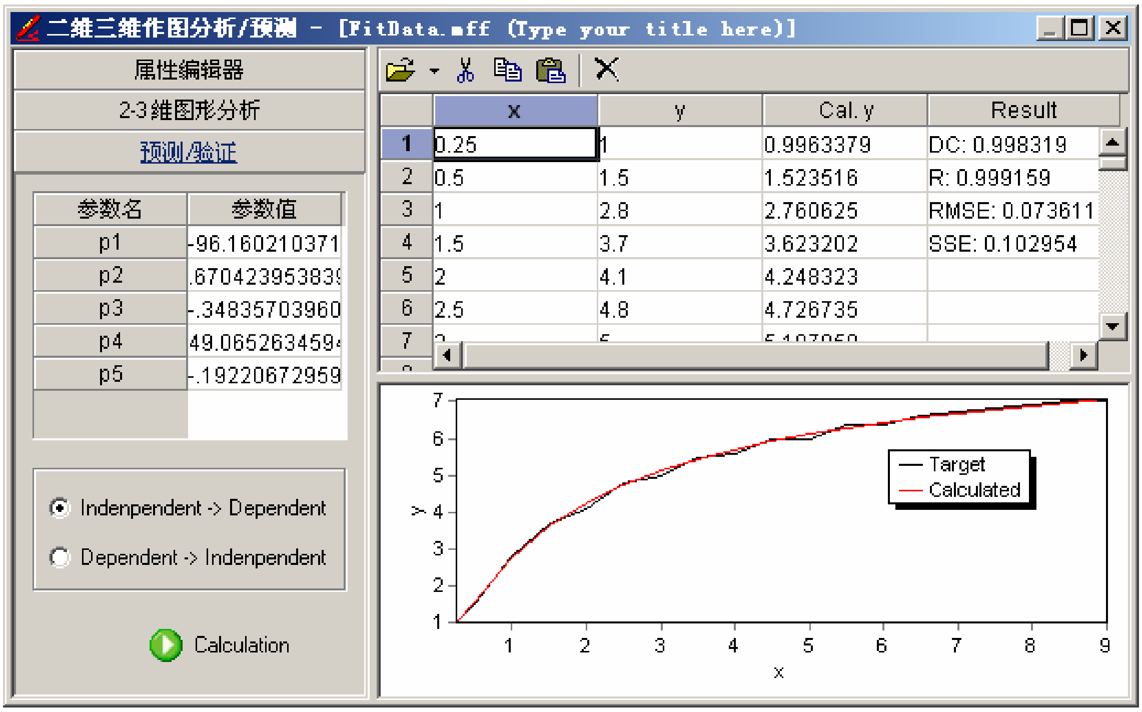


图. 27 验证预测

2. 7：其它应用

很多问题都可转化为优化问题，1stOpt 因此还可用于解决许多其它问题。

例 2. 7. 1：隐函数作图

二维：x vs. y

$$\ln(3.5\cdot x^{y})+x-y^{x}-(\sin(y-x))^{2}+0.6-\frac{(x+y)^{(0.1*x)}}{x}=0 \qquad (39)$$

其中， $x\in[1.5,10]$

三维：z vs. x，y

$$\ln(z)+\sin(x+y-z)^{2}=(x-10-z)^{3}+\cos(z)*(y-100)^{2} \qquad (40)$$

其中， $x\in[9,13], y\in[97,103]$

1stOpt 代码(二维)：

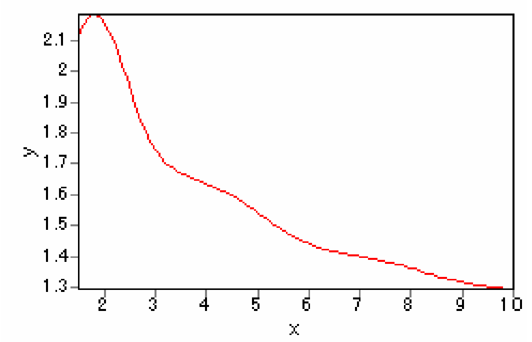
```
Parameters x[1.5,10], y;
StepX = 100;
PlotFunction ln(3.5*x^y)+x-y^x-(sin(y-x))^2+0.6-(x+y)^(0.1*x)/x   =   0;
```

1stOpt 代码(三维)

```
Parameters x[9,13], y[97,103], z;
StepX = 30;
StepY = 30;
PlotFunction ln(z)+sin(x+y-z)^2 = (x-10-z)^3 +cos(z)*(y-100)^2;
```

结果：

二维



三维

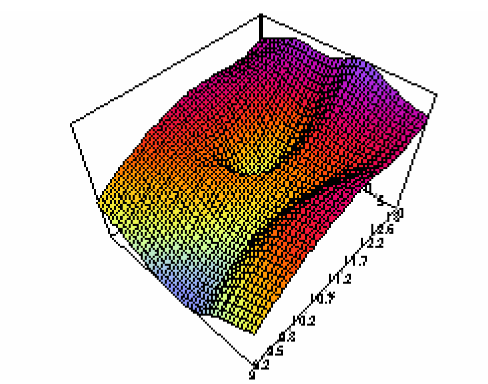


图.28 隐函数作图结果

例 2. 7. 2：作为高级计算器使用

可直观用于计算任意表达式之值。支持特殊计算符号如求和（Σ），求积（Π），积分（°），同时也支持一些特殊函数如伽玛函数（Gamma Function），贝塞尔函数等（Bessel Function）。

试求下列各式之值

1： $$f_1 = 5 \cdot \sin(\pi \cdot 6)^2 + \exp\left(6.45 + \ln(2.14)\right) \tag{41}$$

2： $$f_2 = \int_0^{\pi} \left(x^{\sin(Gamma(x))} + \left(abs\left(\sin(x)\right)\right)^x\right)dx + \sum_{i=1}^{10}\left(\prod_{j=1}^{i}\left(0.1 \cdot \left(\ln(i \cdot j)\right)^2\right)\right) \tag{42}$$

3： $$f_3 = \sqrt{\max(f_1^{0.1}, f_2)} \tag{43}$$

4： $$f_4 = \sum_{i=1}^{3}\sum_{j=1}^{i}\sum_{m=1}^{4}\prod_{n=1}^{m}\prod_{p=1}^{10}\left(f_3^{\frac{i+j+m+n+p}{1000}}\right) \tag{44}$$

1stOpt 代码：

```
f1 = 5*sin(pi*6)^2+exp(6.45+ln(2.14));
f2 = int(x^(sin(Gamma(x)))+(abs(sin(x)))^x, x=0:pi)+sum(i=1:10)(prod(j=1:i)(0.1*ln(i*j)^2));
f3 = sqrt(max(f1^0.1, f2));
f4 = Sum(i=1:3)(Sum(j=1:i)(Sum(m=1:4)(Prod(n=1:m)(Prod(p=1:3)(f3^((i+j+m+n+p)/1000))))));
```

结果：f_1 = 1353.9829，f_2 = 48.669818，f_3 = 6.97637，f4 = 28.2972