1stOpt 8.0 主要新增功能

1stOpt 8.0 版 2018 年 8 月 25 日正式发布，主要新增功能及改进如下，10 月 1 日开始符合免费升级或打算升级的用户可申请办理。

1. 新的界面

在保留传承已有用户界面的基础上，增加了新的用户界面。新用户界面每个代码本都拥有自己的结果页面，代码本表格从“暗处”走向“明处”，更加方便数据处理及结果展示。

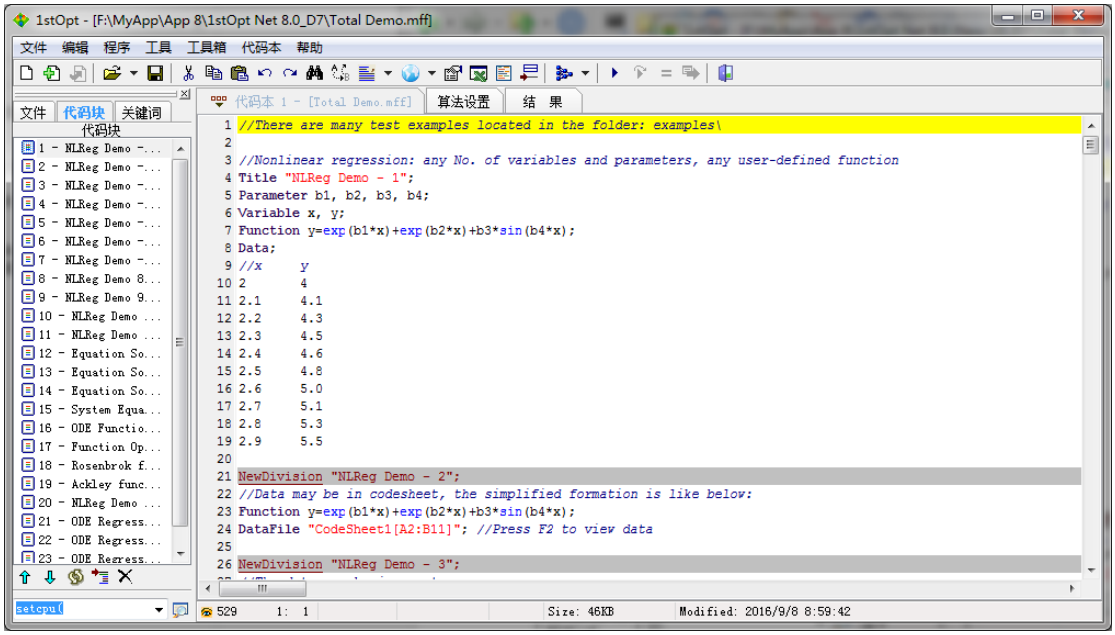

图 1、传统界面

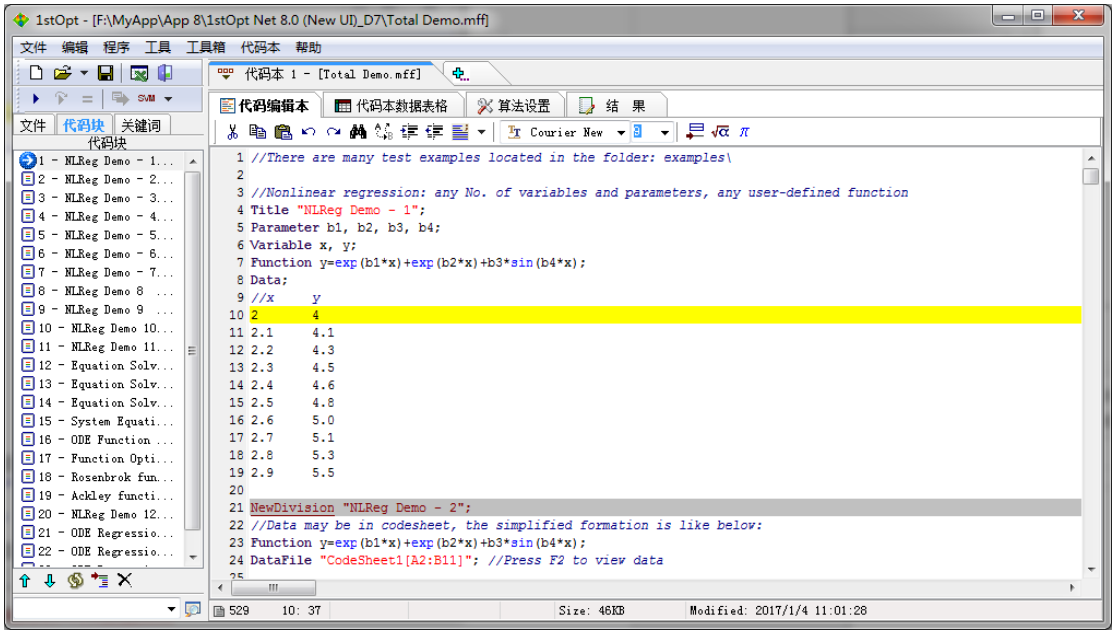

图 2、新界面

2. 计算结果报表的改进

计算结果报表展示改进。

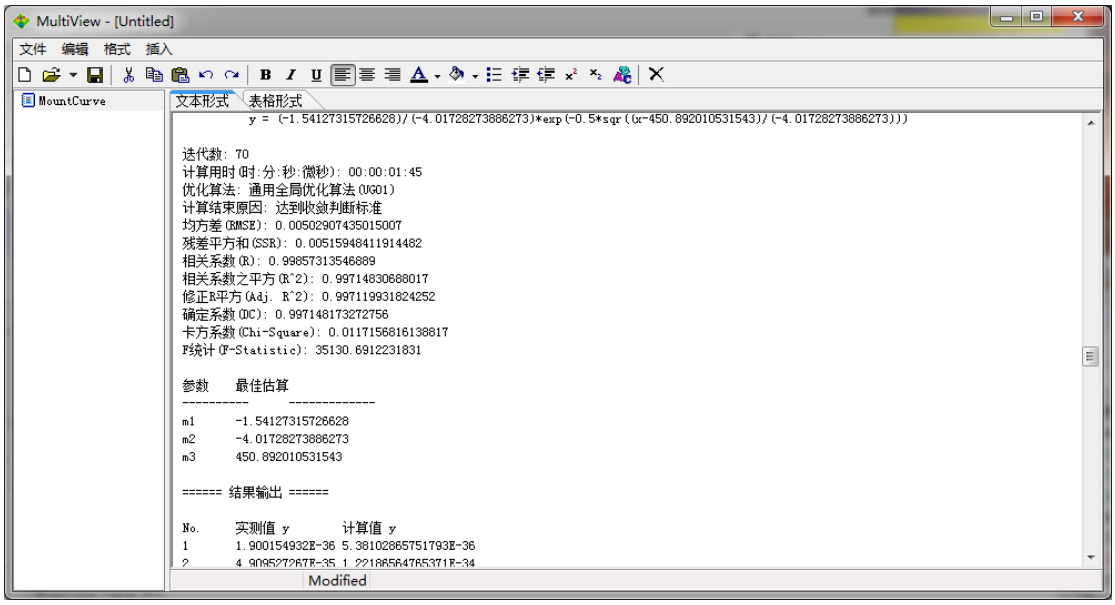

图 3、结果展示一

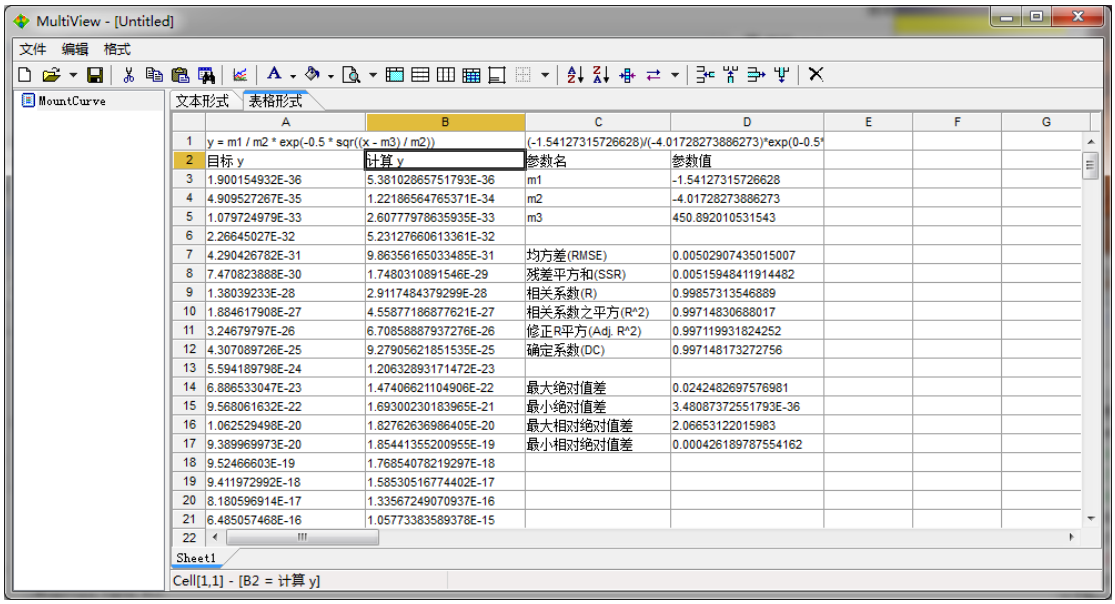

图 4、结果展示二

3. 内部直接支持 Python 高级语言

用 Python 语言编写复杂模型

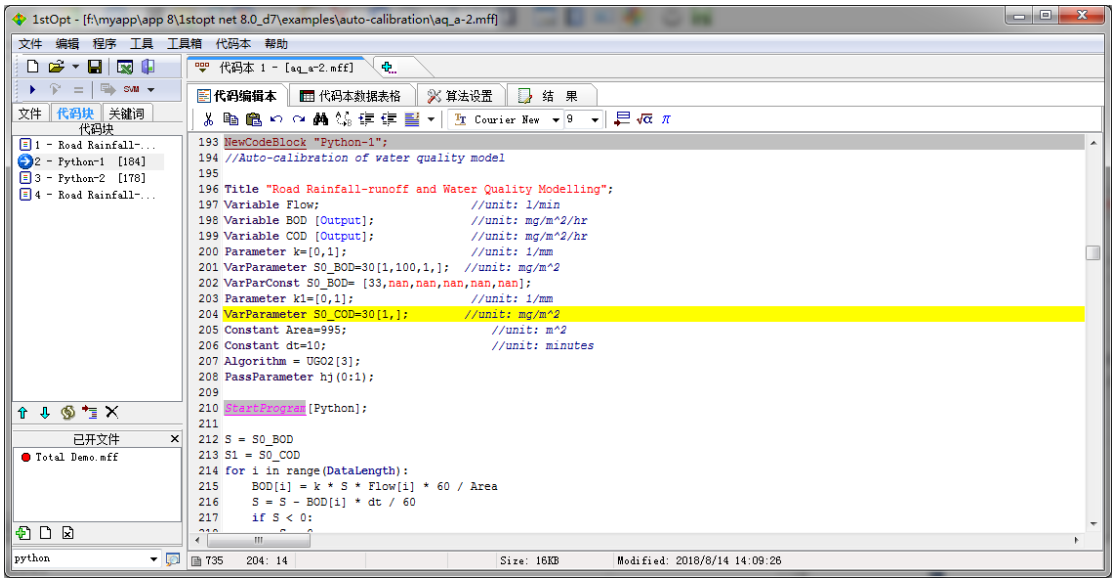

图 5、直接使用 Python 语言

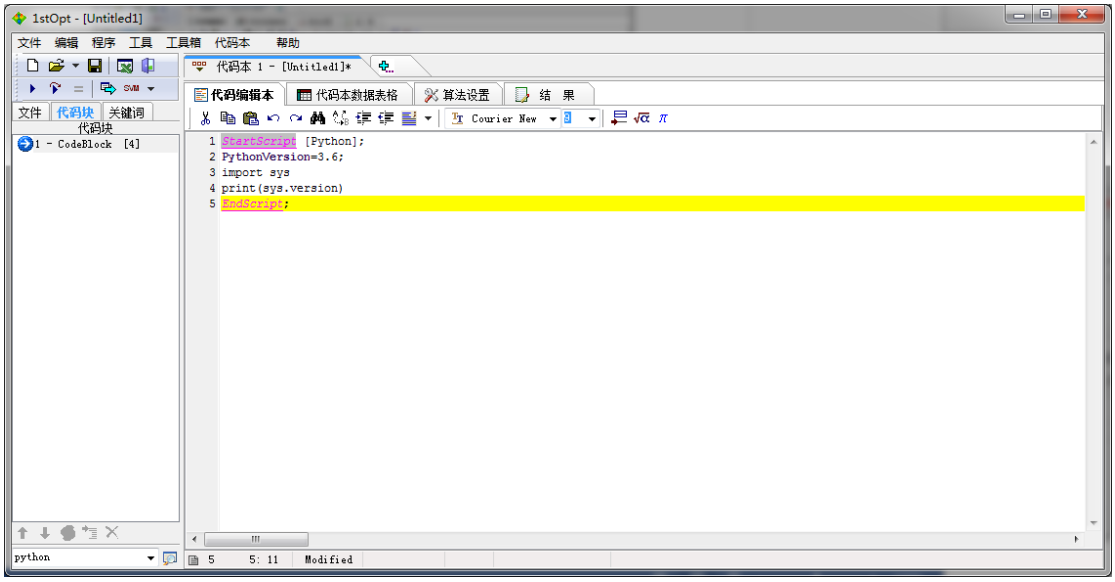

图 6、直接运行 Python 脚本语言

4. 公式自动搜索（快速拟合）的改进，可处理任意多自变量数目

公式自动搜索匹配功能能够对任意数量多的自变量进行自动搜索匹配计算（7.0 及以前版本最多两个自变量）。

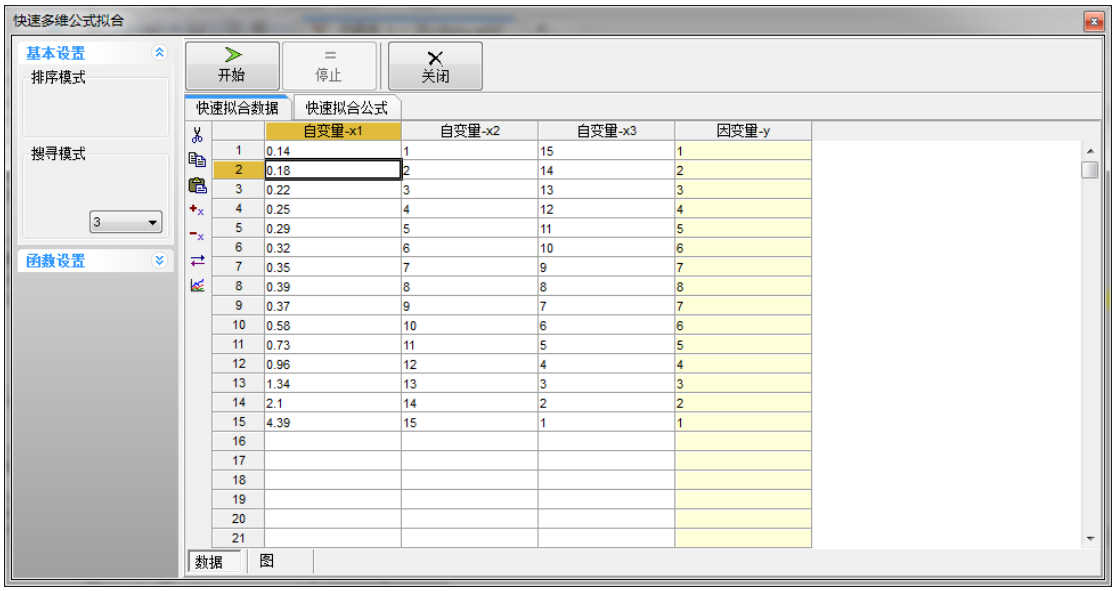

图 6、三自变量公式自动匹配搜索数据界面

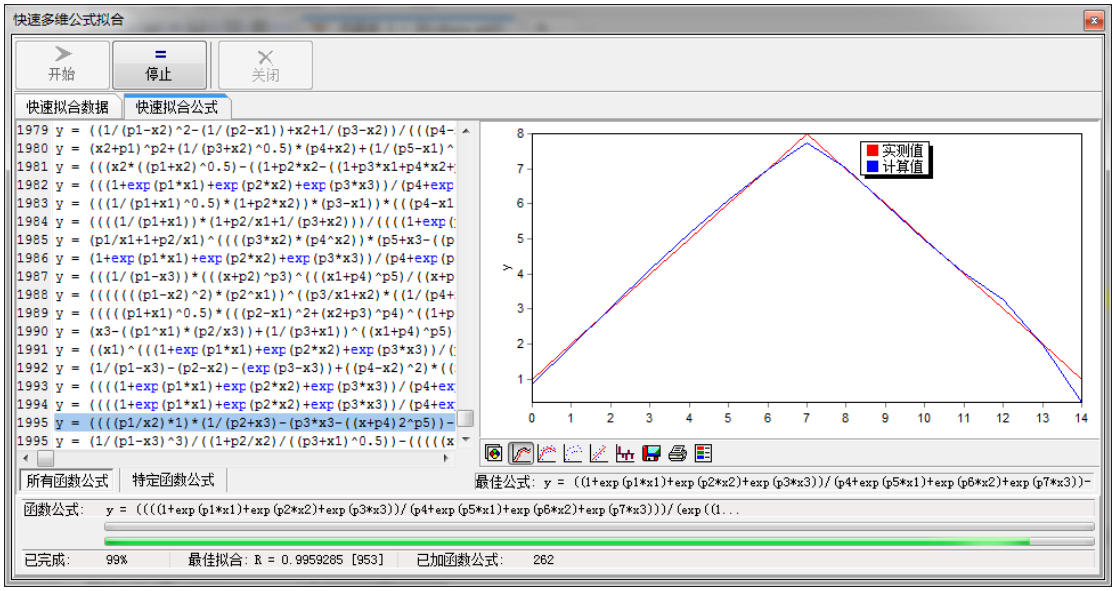

图 7、三自变量公式自动匹配搜索计算结果

5. 微分方程与代数方程混合拟合

可以实现微分方程与代数方程的混合拟合计算。

已知二阶微分方程及代数方程如下：

$$\begin{cases} \dfrac{dy^2}{dt} = a \cdot y^2 \\ z = a \cdot ln(t) + \dfrac{b}{\exp(y)} \end{cases}$$

数据：

t	1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3
y	0.909,0.706,0.363,0.079,-0.226,-0.474,-0.765,-1.183,-1.436,-1.573,-1.811,-1.937,-1.869,-2.013,-1.679,-1.668,-1.465,-1.146,-0.791,-0.586,-0.282
z	0.051,0.276,0.435,0.611,0.807,1.119,1.307,1.514,1.673,1.847,2.168,2.389,2.453,2.693,2.648,2.413,2.288,2.271,2.564,2.392,2.197

两个待求参数 a 和 b

代码：

```
Variable t,y,z;
ODEFunction y"=a*y^2;
            z=a*ln(t)+b/exp(y);
Data;
//t    y      z
1      0.9090.051
1.1    0.7060.276
1.2    0.3630.435
1.3    0.0790.611
1.4    -0.226      0.807
1.5    -0.474      1.119
1.6    -0.765      1.307
1.7    -1.183      1.514
```

1.8	-1.436	1.673
1.9	-1.573	1.847
2	-1.811	2.168
2.1	-1.937	2.389
2.2	-1.869	2.453
2.3	-2.013	2.693
2.4	-1.679	2.648
2.5	-1.668	2.413
2.6	-1.465	2.288
2.7	-1.146	2.271
2.8	-0.791	2.564
2.9	-0.586	2.392
3	-0.282	2.197

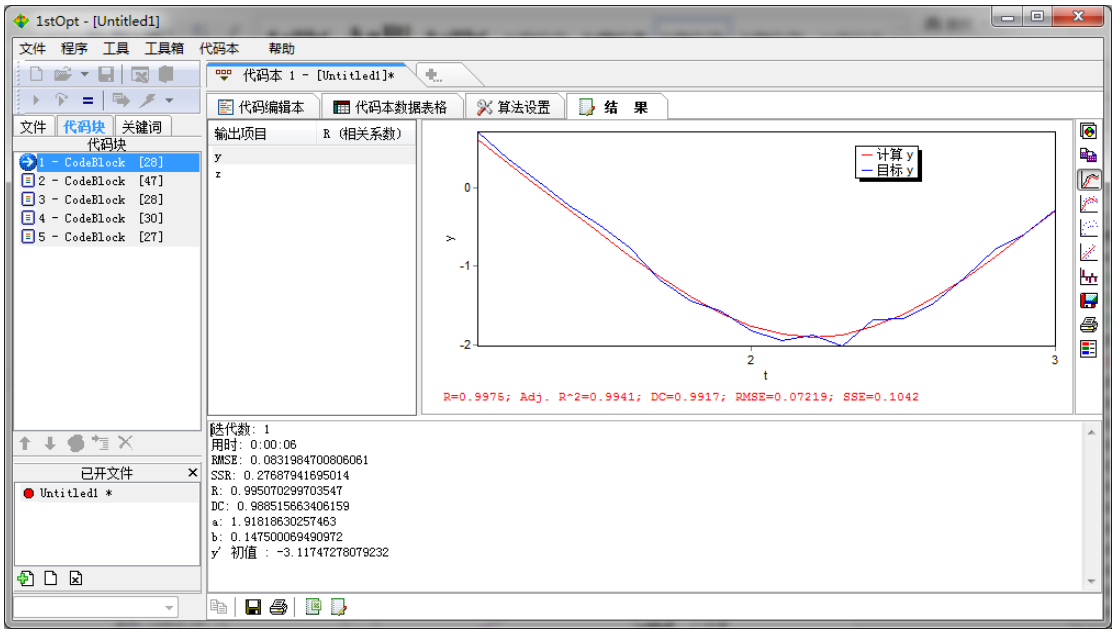

图 8、微分与代数方程混合拟合计算结果一

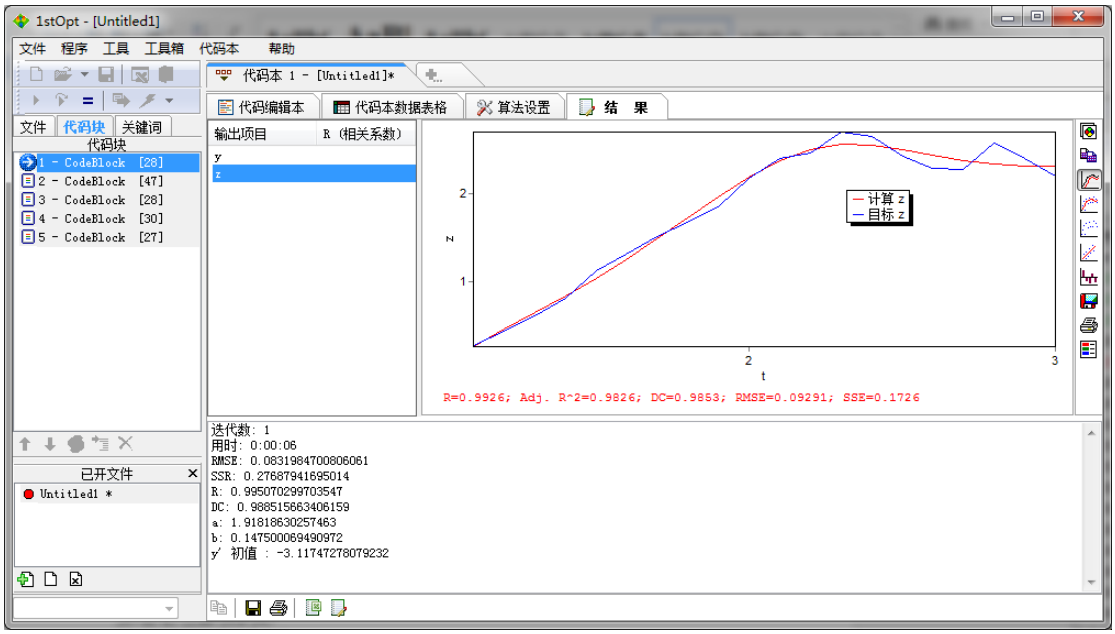

图 9、微分与代数方程混合拟合计算结果二

6. 微分方程拟合可以设定未知微分方程初值的范围

微分方程拟合时可以设定变量初始值（未知参数）的范围，如下定义变量 y2 与 y3 的初始值分别为参数 p2 和 p3，并设定各自范围：

InitialODEValue y2=p1=[1.8*10^4,1.4*10^7], y3=p2=[1.8*10^3,1.8*10^5];

完整问题求解代码：

```
Constant    u=3.9139*10^(-5);
Parameter    k1= [0.0001,50];
Parameter    k2= [0.0001,1];
Parameter    k3= [0.0001,1];
Parameter    k4= [0.1,1];
Parameter    k5= [0.01,1];
Parameter    k6= [0.1,1];
EnhancedBound = 1;
InitialODEValue t=1, y1=1.4*10^8,y2=p1=[1.8*10^4,1.4*10^7], y3=p2=[1.8*10^3,1.8*10^5], y4=1212,y5=0;
Variable t,y4;
ODEFunction
y1'=u*(y1+y2+y3+y4+y5)-k1*(y3+y4)*y1/(y1+y2+y3+y4+y5)+k2*y5-u*y1;
y2'=k1*(y3+y4)*y1/(y1+y2+y3+y4+y5)-k3*y2-u*y2;
y3'= k3*(1-k5)*y2-k4*y3-u*y3;
y4'= k3*k5*y2-k6*y4-u*y4;
y5'= k4*y3+k6*y4-k2*y5-u*y5;
Data;
t=31,59,90,120,151,181,212,243,273,304,334,365;
y4=37567,23862,77756,248609,354347,343100,261263,119096,101654, 87612,79591,60879;
```

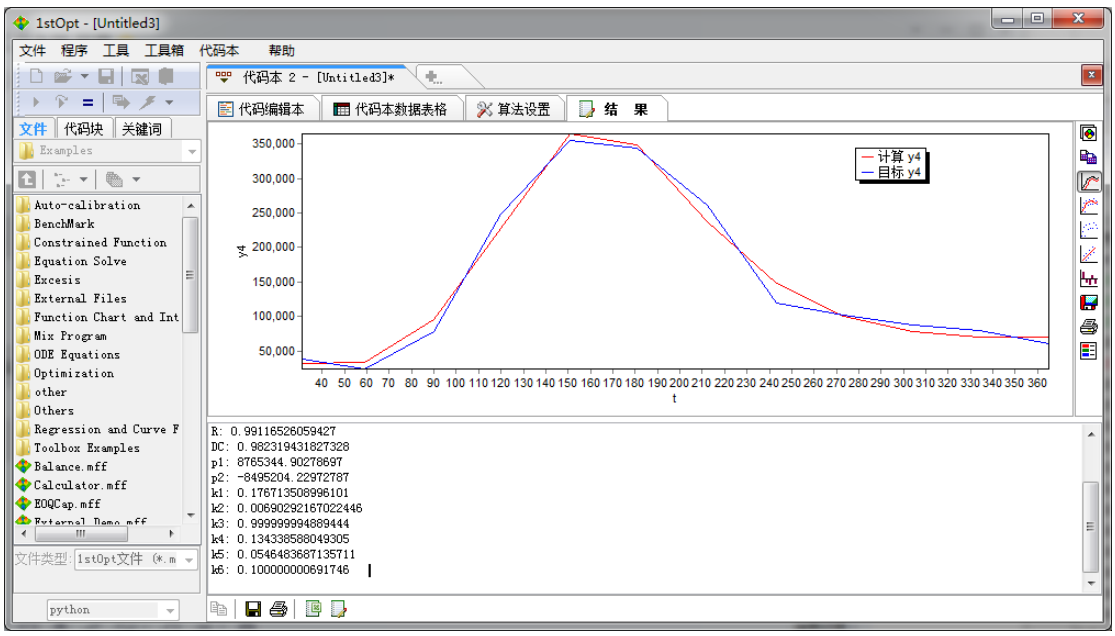

图 10、未知变量初值设定范围的微分方程拟合

7. 微分方程拟合预测时光滑图形功能（预测功能改进）

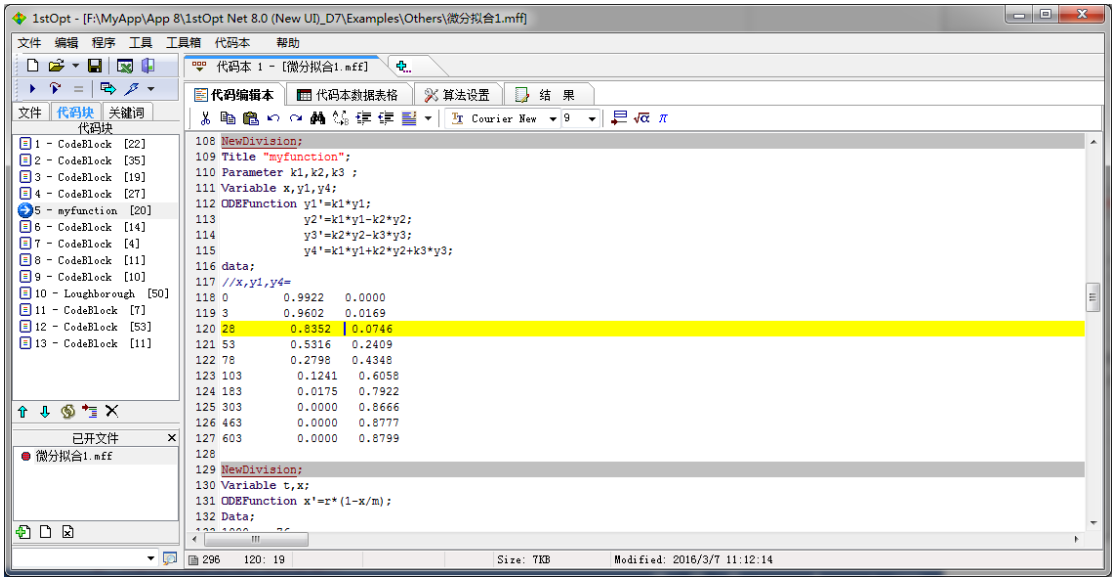

图 11、微分方程拟合

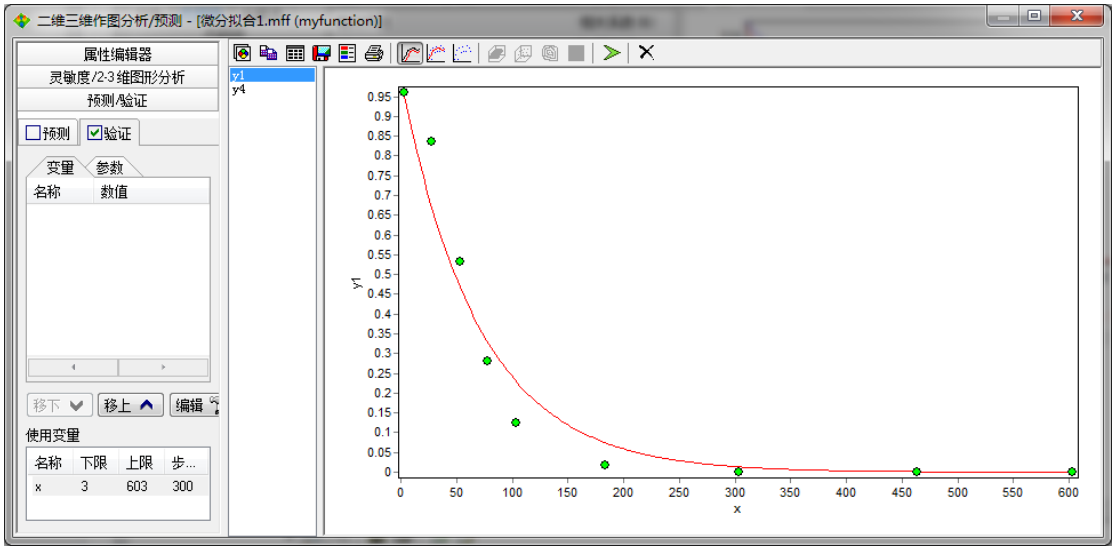

图 12、微分方程预测

8. 多层自回归网络工具箱

创新的多层自回归网络模型工具箱（Multi-Layer Fitting Network， MFN）可广泛用于多变量/多因子拟合及分类建模，与支持向量机及神经网络相比，其最大特点就是处理海量数据时速度超快，为前两者的 10 至 100 倍。

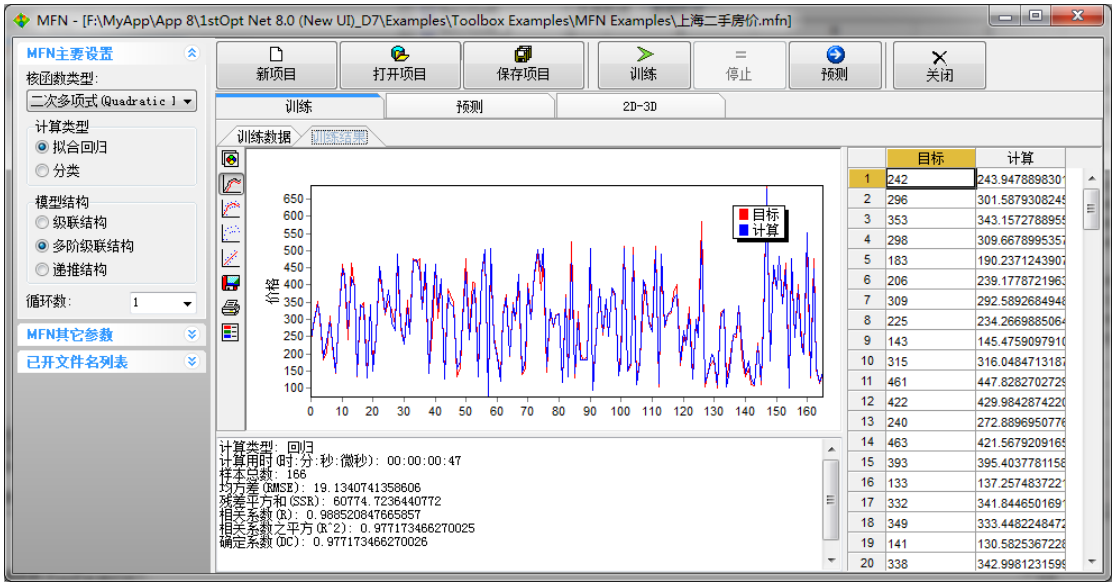

图 13、MFN 多元拟合建模计算

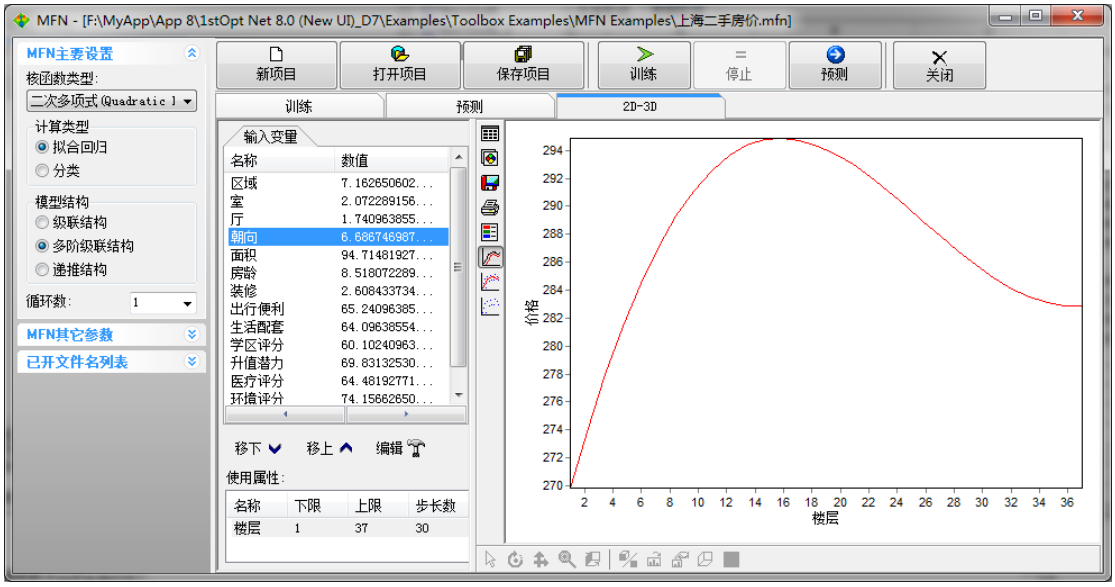

图 14、MFN 预测分析计算

9. 自动取不同起点和终点数据拟合

“RegStartP”与“RegEndP”拟合关键字可指定拟合数据的起始和终止点，进行批量不同数据的计算。下面代码数据共有 141 组，通过“RegStartP=[1:5:10];”与“RegEndP=[130,136,141];”，将分别自动取数据区间 “1 至 130”，“5 至 136”和“10 至 141”进行计算

```
RegStartP=[1:5:10];
RegEndP=[130,136,141];
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;
1428,2857,5714,8571,11429,14286,17143,20000,22857,25714,28571,31429,34286,37143,40000,42857,45714,48
```

571,51429,54286,57143,60000,62857,65714,68571,71429,74286,77143,80000,82857,85714,88571,91429,94286,97143,100000,102857,105714,108571,111429,114286,117143,120000,122857,125714,128571,131429,134286,137143,140000,142857,145714,148571,151429,154286,157143,160000,162857,165714,168571,171429,174286,177143,180000,182857,185714,188571,191429,194286,197143,200000,202857,205714,208571,211429,214286,217143,220000,222857,225714,228571,231429,234286,237143,240000,242857,245714,248571,251429,254286,257143,260000,262857,265714,268571,271429,274286,277143,280000,282857,285714,288571,291429,294286,297143,300000,302857,305714,308571,311429,314286,317143,320000,322857,325714,328571,331429,334286,337143,340000,342857,345714,348571,351429,354286,357143,360000,362857,365714,368571,371429,374286,377143,380000,382857,385714,388571,391429,394286,397143,400000;

0.0159,0.0294,0.0540,0.0832,0.1769,0.3010,0.4822,0.8909,1.4673,2.6091,3.9487,6.1861,10.2854,14.1971,20.8154,27.9622,34.1893,50.9944,52.8896,74.7148,78.2739,94.8341,97.1913,106.9074,110.5764,99.7094,95.2368,100.9800,95.3051,84.7236,72.5028,63.7613,53.8688,45.3433,47.6868,39.2611,41.2603,38.4826,41.0614,42.8718,52.4247,54.6309,58.5655,57.6910,62.2335,63.6598,74.5593,76.3657,76.8343,76.4166,73.9337,83.6326,79.5735,77.3333,82.7579,74.8436,70.2986,69.1072,62.9654,63.9911,63.3608,54.3341,54.7983,54.1602,50.5028,46.6765,37.3148,35.4733,31.4011,31.6141,29.3181,29.2092,25.6456,23.5434,21.6426,20.5469,21.0379,23.0230,23.3786,22.6132,28.5862,29.2285,34.6253,40.5241,43.5328,44.4078,55.3422,63.8790,70.3837,64.7925,84.1060,85.2346,96.3957,101.5065,101.5635,125.4770,123.0295,116.8548,132.5786,146.0718,152.3447,149.5277,149.2917,167.8167,140.3957,146.2560,140.2877,150.3306,151.0844,135.0494,124.1520,128.2794,121.8187,127.7168,109.3237,98.8645,101.0413,95.4619,82.9761,108.2798,173.0225,125.9043,55.3894,48.4749,40.5538,32.5134,30.7203,26.0722,21.6562,19.8353,16.4291,14.3602,11.6957,8.8610,7.2047,6.0873,5.1435,4.1757,3.0479,2.7644,2.1079;

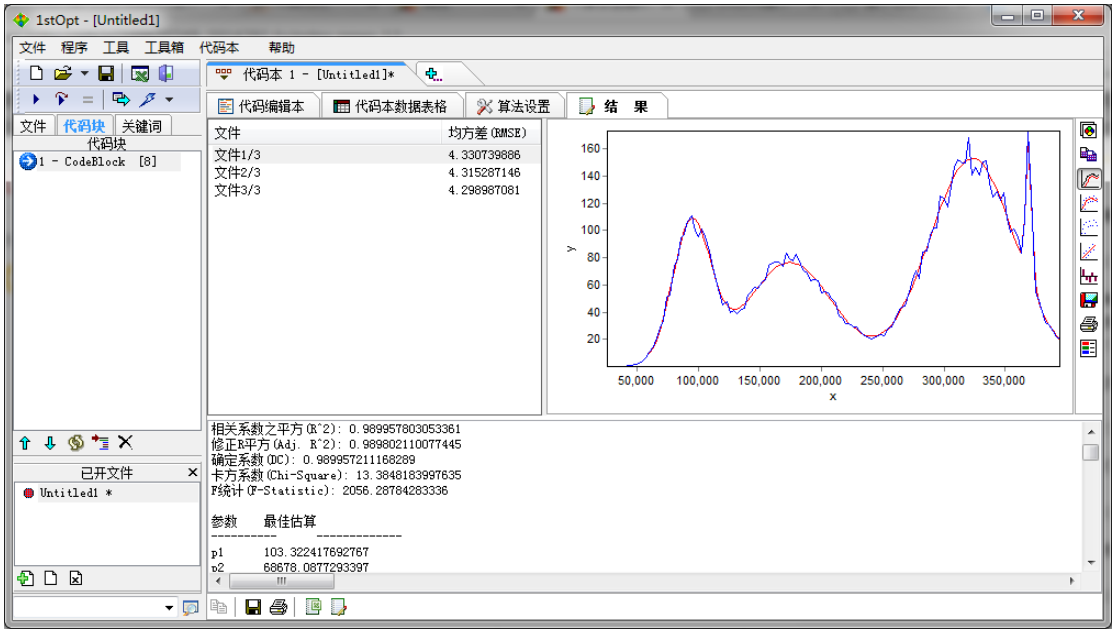

图 15、自动数据起始-终止选取设定拟合计算一

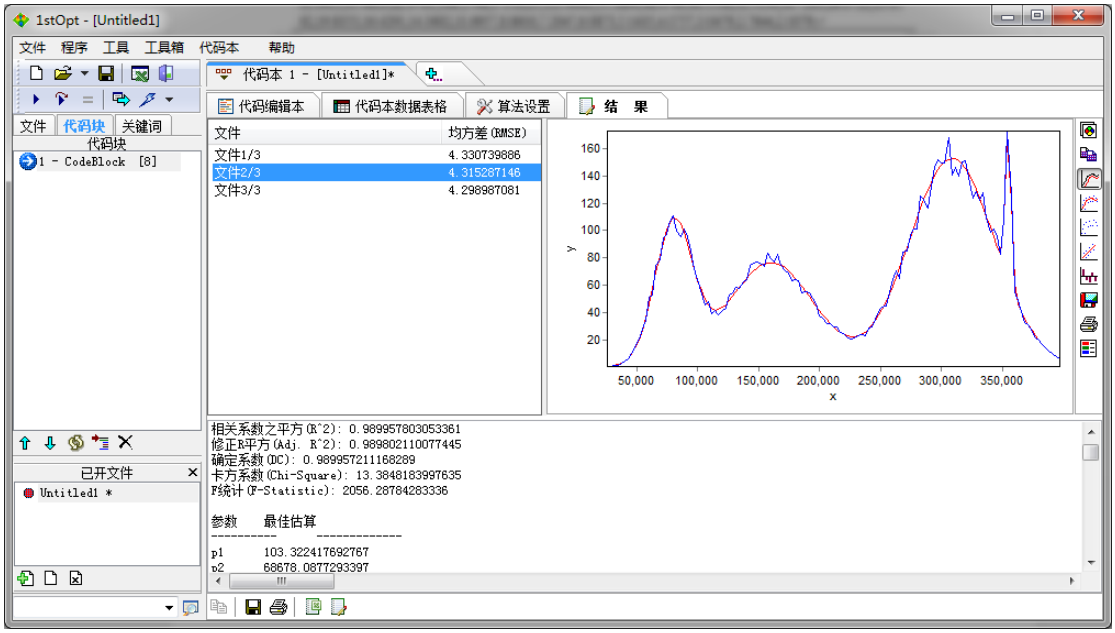

图 16、自动数据起始-终止选取设定拟合计算二

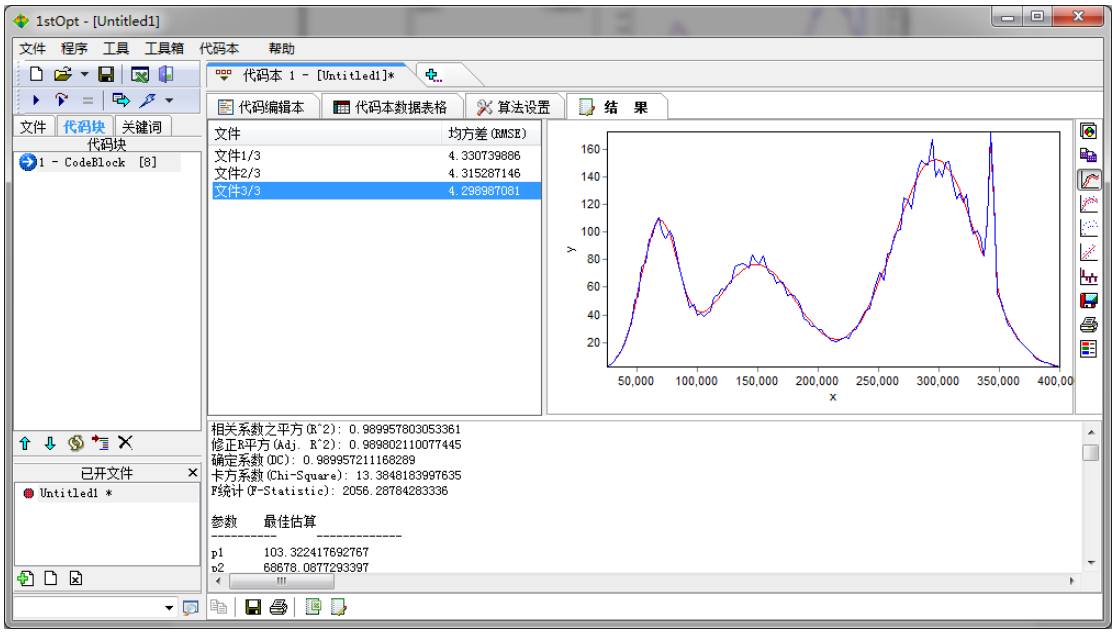

图 17、自动数据起始-终止选取设定拟合计算三

10. TSP 工具箱

旅行商（TSP）问题求解工具箱，简单、快捷、高效。

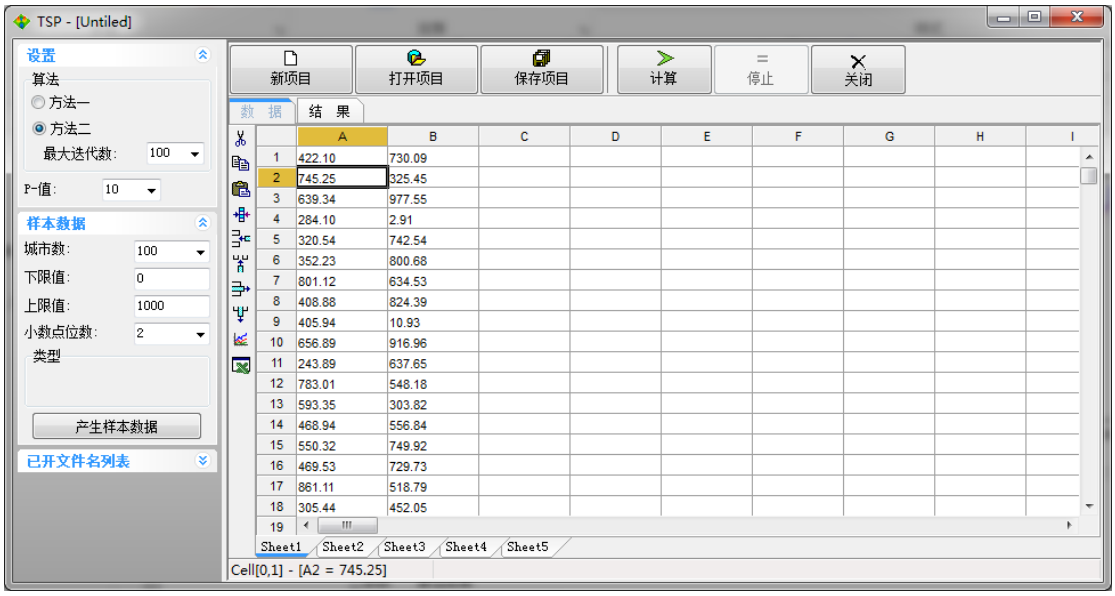

图 18、TSP 城市位置坐标输入界面

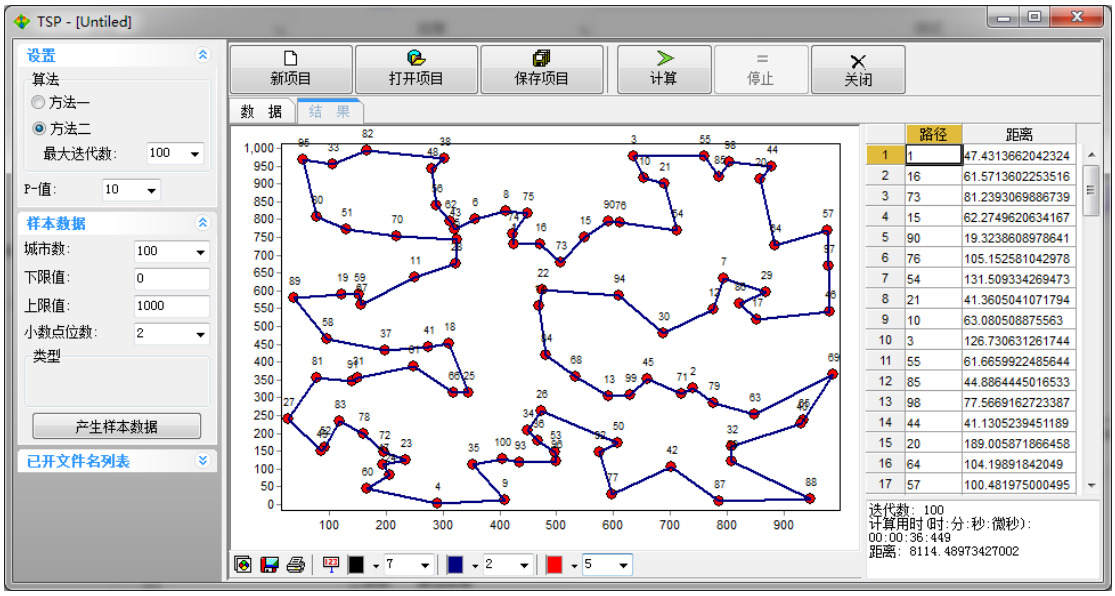

图 18、TSP 最优路径计算结果

11. SETUP 工具箱

有序排列计划工具箱主要用于解决如工厂工序排列问题。

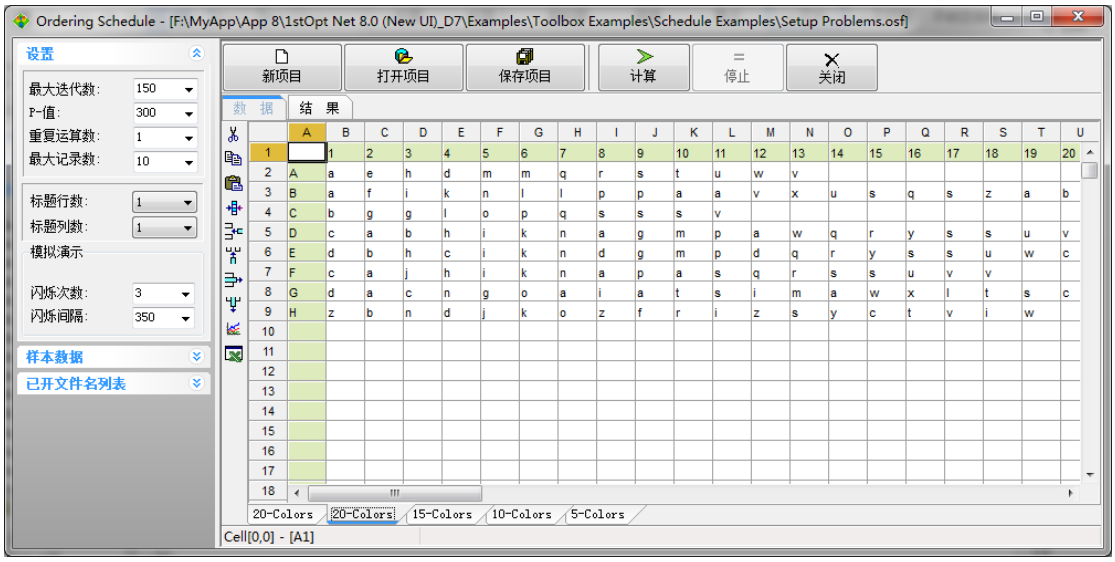

图 19、SETUP 工具箱数据录入界面

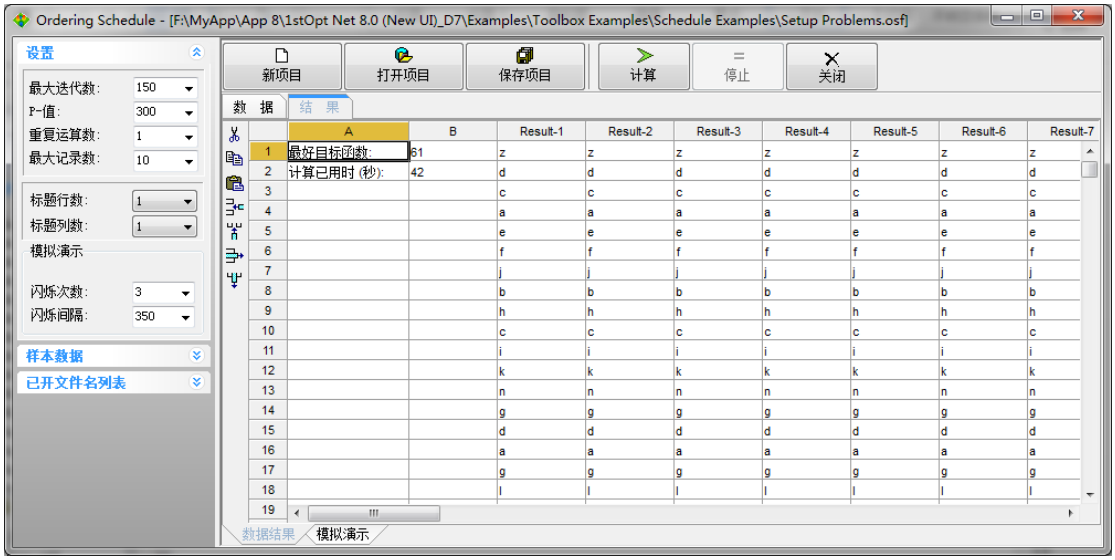

图 20、SETUP 工具箱计算结果界面

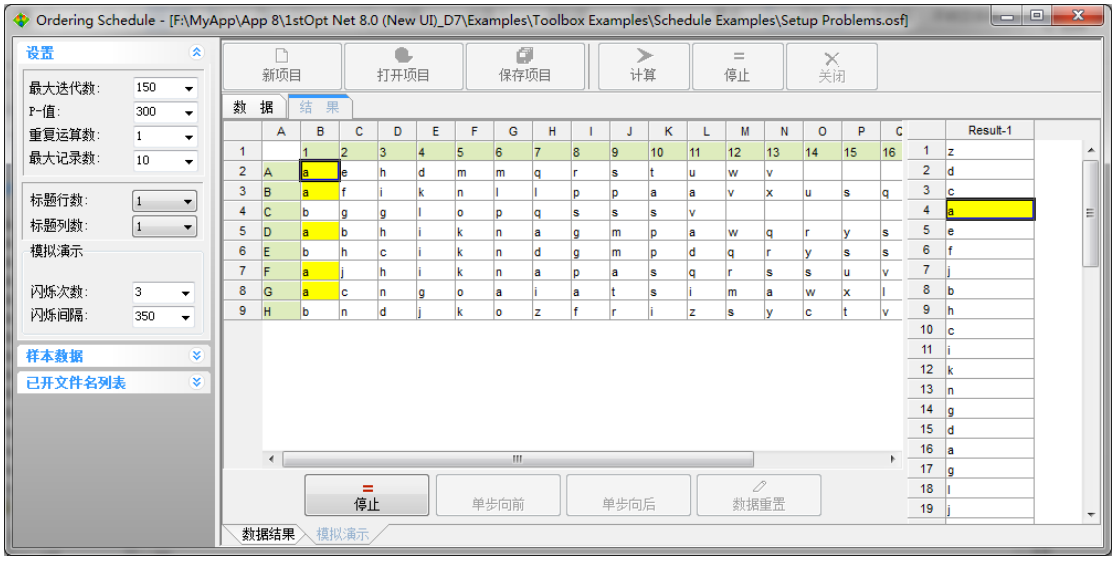

图 21、SETUP 工具箱模拟演示界面

12. 线性算法（LP）直接求解线性方程组

优化形式	Algorithm = LP;
	Constant n=1000;
	MinFunction 0;
	4*x1+x2=1;
	For(i=1:n-2)(x[i]+4*x[i+1]+x[i+2]=i+1);
	x[n-1]+4*x[n]=n;

直接解方程形式	Algorithm = LP;
	Constant n=1000;
	Function 4*x1+x2=1;
	For(i=1:n-2)(x[i]+4*x[i+1]+x[i+2]=i+1);
	x[n-1]+4*x[n]=n;

13. 文本文件读取（行数据）

数据以行形式保存/读取。以 10 项目 0-1 装包问题为例，文本数据存储格式如下（文件名：kp_100_2.txt）

```
10
986 589 848 314 255 905 435 607 979 749
675 233 95 576 495 625 509 544 904 548
2557
```

代码

```
Constant nbItems = "kp_10_2.txt[1]";
Constant weights(nbItems) = "kp_10_2.txt[2]";
Constant prices(nbItems) = "kp_10_2.txt[3]";
Constant knapsackBound = "kp_10_2.txt[4]";
BinParameter x(nbItems);
Algorithm = LP;
MaxFunction Sum(i=1:nbItems)(prices[i]*x[i]);
        Sum(i=1:nbItems)(weights[i]*x[i]) <= knapsackBound;
```

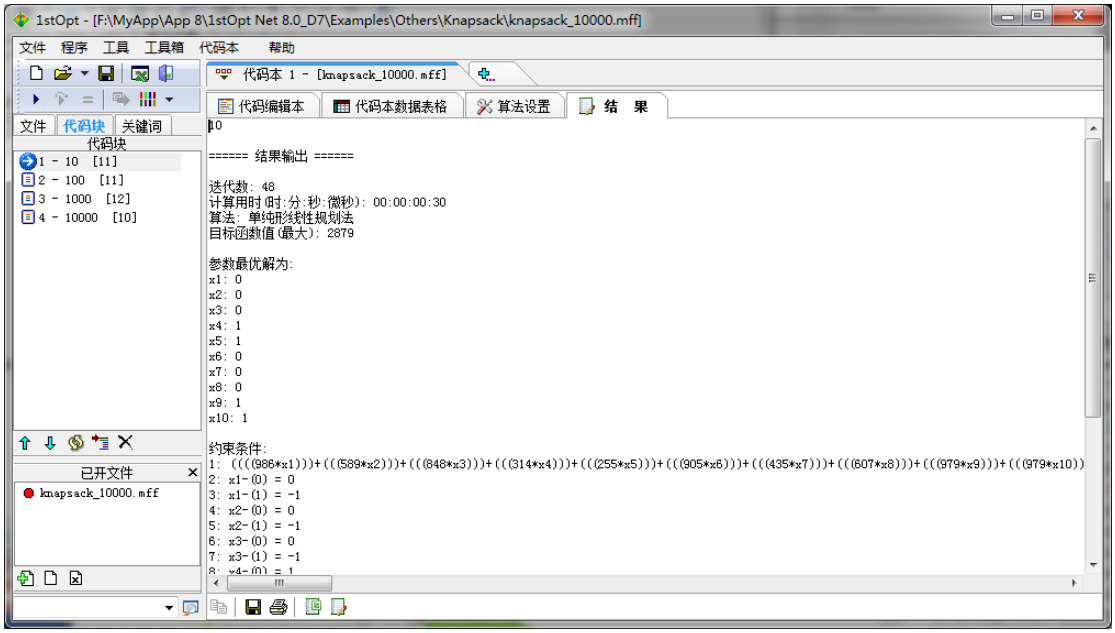

图 22、0-1 装包问题行数据读取计算演示

14. “VarParConst”处理变参数和变常数混合问题

“VarConstant”和“VarParameter”分别用于定义“变常数”和“变参数”，新增加的“VarParConst”则用于定义前面两者的混合状态。如下列，S0_BOD 即有已知数也有未知数，未知数用“NAN”表示。

```
//Auto-calibration of water quality model
Title "Road Rainfall-runoff and Water Quality Modelling";
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=30[1,100,1,];   //unit: mg/m^2
VarParConst S0_BOD= [33,NAN,NAN,NAN,NAN,NAN];
Parameter k1=[0,1];                     //unit: 1/mm
VarParameter S0_COD=30[1,];             //unit: mg/m^2
Constant Area=995;                          //unit: m^2
Constant dt=10;                             //unit: minutes

StartProgram;
Procedure MainModel;
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;
//Auto-calibration of water quality model
Data;
0,0,0,6.7,8.6,8,6.3,4.8,4,3.2,3.4,3.6,2.45,1.3,1.25,1.2,1.5,1.8,2.1,1.7,1.3,0.9;
0.000,0.000,0.000,4.040,5.186,4.535,3.647,2.402,2.086,1.737,1.896,2.062,1.426,0.768,0.746,0.724,0.935,1.158,1.
393,1.162,0.915,0.651;
0,0.000,0.000,10.101,11.928,11.095,7.978,5.210,4.462,3.666,3.793,3.908,2.659,1.411,1.432,1.447,1.779,2.098,2.4
06,1.982,1.542,1.085;
Data;
0,0,0,0.8,4,5.7,5.2,4.8,3.3,1.8;
0.000,0.000,0.000,2.894,13.749,19.248,15.365,14.183,9.452,4.993;
0.000,0.000,0.000,5.307,28.945,32.653,25.713,23.156,15.721,8.466;
Data;
0,0,0,0,0.3,0.6,1.8,4.4,7.5,5.35,3.2,2.35,1.5,0.9,0.3;
0.000,0.000,0.000,0.000,1.990,3.075,14.111,14.858,22.161,15.969,9.648,7.015,4.432,2.876,1.031;
0.000,0.000,0.000,0.000,3.437,6.513,17.367,31.839,49.749,37.101,23.156,17.005,10.854,6.513,2.171;
Data;
0,0,0,0.2,0.3,0.6,4,6.7,7.65,8.6,6,3.4,2.45,1.5,3.25,5,3.8,2.6,1.4,1.25,1.1,0.95,0.8,0.65,0.5,0.61667,0.73333,0.85,0.
96667,1.0833,1.2;
0.000,0.000,0.000,0.277,0.271,0.398,2.894,4.444,5.305,6.223,3.781,1.825,1.315,0.805,1.538,2.050,1.596,1.118,0.
616,0.535,0.458,0.384,0.314,0.247,0.184,0.231,0.280,0.331,0.383,0.437,0.492;
0.000,0.000,0.000,0.422,0.579,1.122,6.513,8.888,9.918,10.890,7.055,3.690,2.659,1.628,3.234,4.523,3.437,2.352,1
.266,1.156,1.039,0.917,0.788,0.653,0.513,0.638,0.766,0.897,1.030,1.165,1.303;
Data;
0.000,0.000,0.300,0.700,3.100,17.100,7.500,4.200,0.900,0.600,0.300,0.350,0.400,0.400,0.400,0.378,0.356,0.333,0
.311,0.289,0.267,0.244,0.222,0.200,0.233,0.267,0.300,0.333,0.367,0.400,7.022,13.644,20.267,26.889,33.511,40.1
33,46.756,53.378,60.000;
0.000,0.000,0.416,0.507,4.486,12.374,4.070,4.052,1.248,0.832,0.416,0.485,0.555,0.555,0.555,0.519,0.484,0.449,0
.415,0.381,0.348,0.316,0.284,0.253,0.295,0.338,0.380,0.422,0.464,0.507,8.163,14.444,19.350,22.880,25.035,25.8
15,25.218,23.247,19.899;
0.000,0.000,0.796,1.815,8.786,29.904,12.211,7.851,1.899,1.266,0.633,0.749,0.868,0.868,0.868,0.820,0.772,0.724,
0.675,0.627,0.579,0.531,0.482,0.434,0.492,0.547,0.597,0.643,0.685,0.724,11.442,19.782,25.747,29.330,30.536,29
.363,25.814,19.885,11.578;
Data;
0,3.9,7.3,1.6,0.3,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.2,48.0,13.6,13.65,13.7,13.4,13.1,7.75,2.4,1.2,0.7;
0.0000,3.7628,8.3638,1.9296,0.3618,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,3.
2804,31.8392,15.5819,14.4045,13.2181,13.7367,14.2191,8.4121,2.6050,1.4472,0.5065;
0.0000,6.1146,11.0050,2.7980,0.5608,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,5
.2101,75.2563,18.8623,19.7548,20.6533,21.0090,21.3286,11.6834,3.3286,1.6643,1.0131;
```

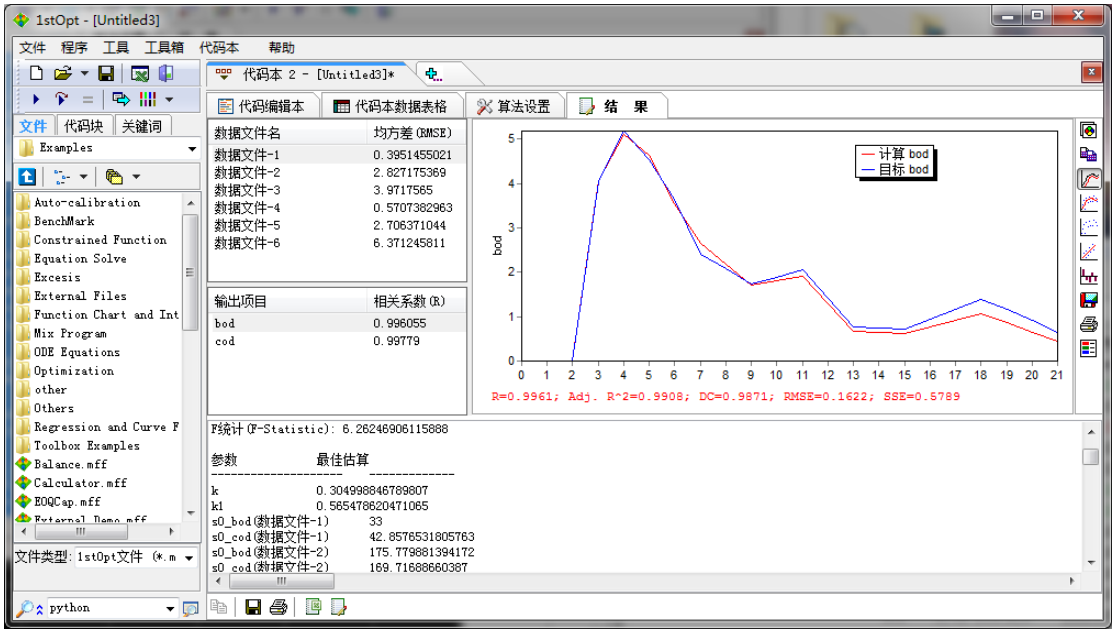

图 23、“VarParConst”范例计算演示

15. “XAxis”“YAxis”自定义横坐标及纵坐标

下面案例代码指定自变量 t2 为横轴，同时对纵轴取平方。

代码

```
Parameters x1,x2,x3;
Variable y, t1, t2, t3;
XAxis = t2; //Variable Name
YAxis = (OutPut)^2;
Function y=x1+t1/(x2*t2+x3*t3);
Data;
//y t1 t2 t3
0.14 1.0 15.0 1.0
0.18 2.0 14.0 2.0
0.22 3.0 13.0 3.0
0.25 4.0 12.0 4.0
0.29 5.0 11.0 5.0
0.32 6.0 10.0 6.0
0.35 7.0 9.0 7.0
0.39 8.0 8.0 8.0
0.37 9.0 7.0 7.0
0.58 10.0 6.0 6.0
0.73 11.0 5.0 5.0
0.96 12.0 4.0 4.0
1.34 13.0 3.0 3.0
2.10 14.0 2.0 2.0
4.39 15.0 1.0 1.0
```

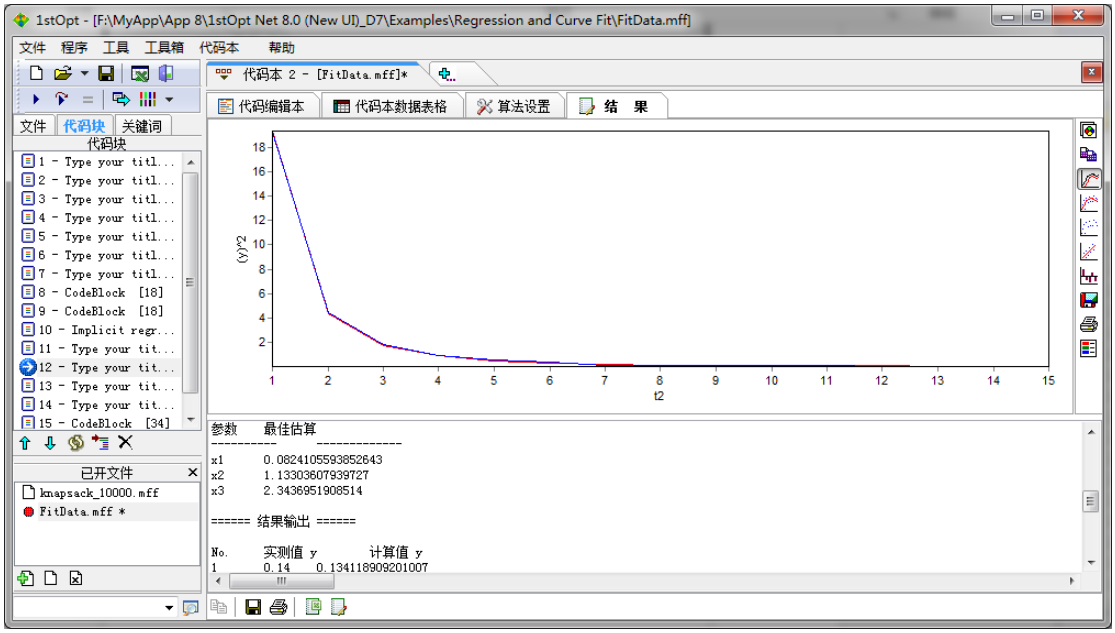

图 25、自定义 X 轴及 Y 轴

16. 打开文件可以选择新开或是在已有代码本

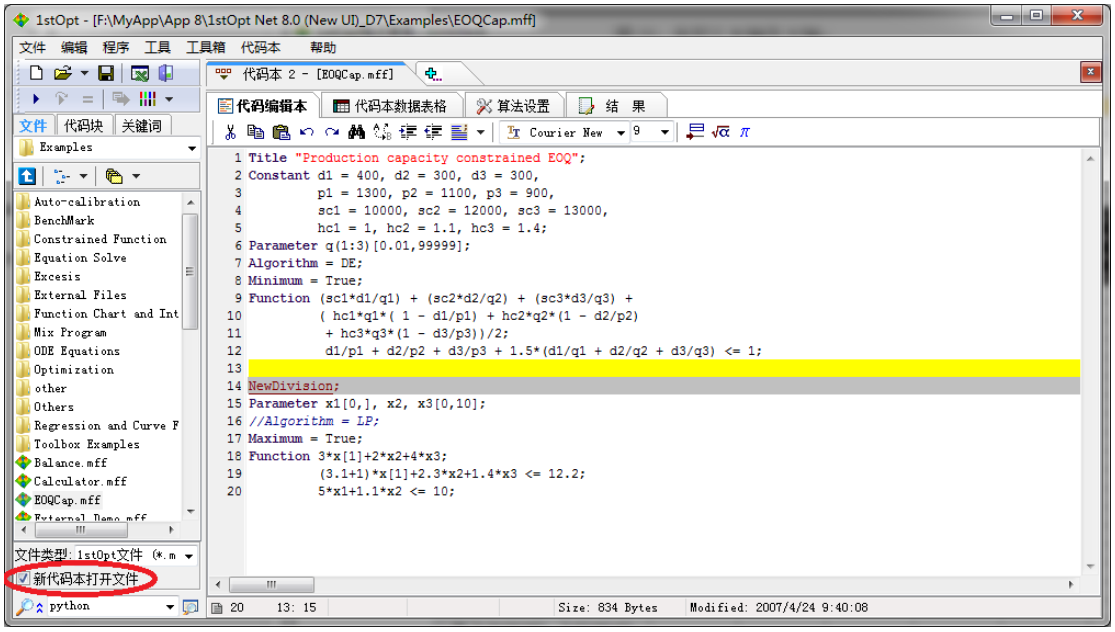

图 26、自定义 X 轴及 Y 轴

17. 经典算法（CLO）参数及结果（RMSE,DC,SSE,R）不能保存的错误
18. 每个代码本可以保存自己独立的所有算法设置
19. 权重数据拟合（指定权重数据）
20. 订正表达式中"^-"引起的错误以及常数"1.856E-3"编程模式下的错误
21. 增加工作目录（WorkPath）
22. 数组字符串定义错误的改进（ConstStr）