7. 1stOpt 非线性差分方程（Difference Equation）求解

差分方程是微分方程的离散化表达形式，是一种递推地定义一个序列的方程式，序列方程中的每一项均为其前一项的函数。

通过四个实际案例，介绍 1stOpt 如何处理不同类型的差分方程。

7.1 案例一

四个变量 x、y、z 和 c，其差分公式如下：

$$\begin{cases} x_{k+1} = x_k - 2 \cdot c_k + h \cdot z_k \cdot sin(y_{k+1} - c_{k+1}) \cdot \left(cos(z_k + x_{k+1})\right)^2 \\ y_{k+1} = y_k - \frac{h \cdot \left(sin(x_{k+1}) - sin(x_k)\right)}{2} \\ z_{k+1} = -z_k + 2 \cdot y_k - h \cdot sin(x_k) \\ c_{k+1} = -c_k + \frac{h \cdot (z_k - z_{k+1})}{2} \end{cases} \tag{7-1}$$

其中：n=1000, h=0.02, k=[1:1:n]; x, y, z 和 c 的起始值：x1=0, y1=0.1, z1=0, c1=0.2;

代码 7-1

```
Constant n=1000, h=0.02;
ChartType = 3;
LoopConstant x1=[0,x2(n)], y1=[0.1,y2(n)], z1=[0,z2(n)],c1=[0.2,c2(n)];
PlotLoopData x2[x],z2;
Function
x2=x1-2*c1+h*z1*sin(y2-c2)*sqr(cos(z1+x2));
y2=y1-h*(sin(x2)-sin(x1))/2;
z2=-z1+2*y1-h*sin(x1);
c2=-c1+h*(z1-z2)/2;
```

上诉代码中，x1、y1、z1 和 c1 视为每一步循环计算的起始值，x2、y2、z2 和 c2 则为本循环对应的结束值，同时自动作为下一步的起始值（自动赋值给 x1、y1、z1 和 c1）。

因为公式本身并不复杂，1000 次循环计算，如果用缺省的算法会耗费较多的时间，在此，算法设定及计算结果如下图：

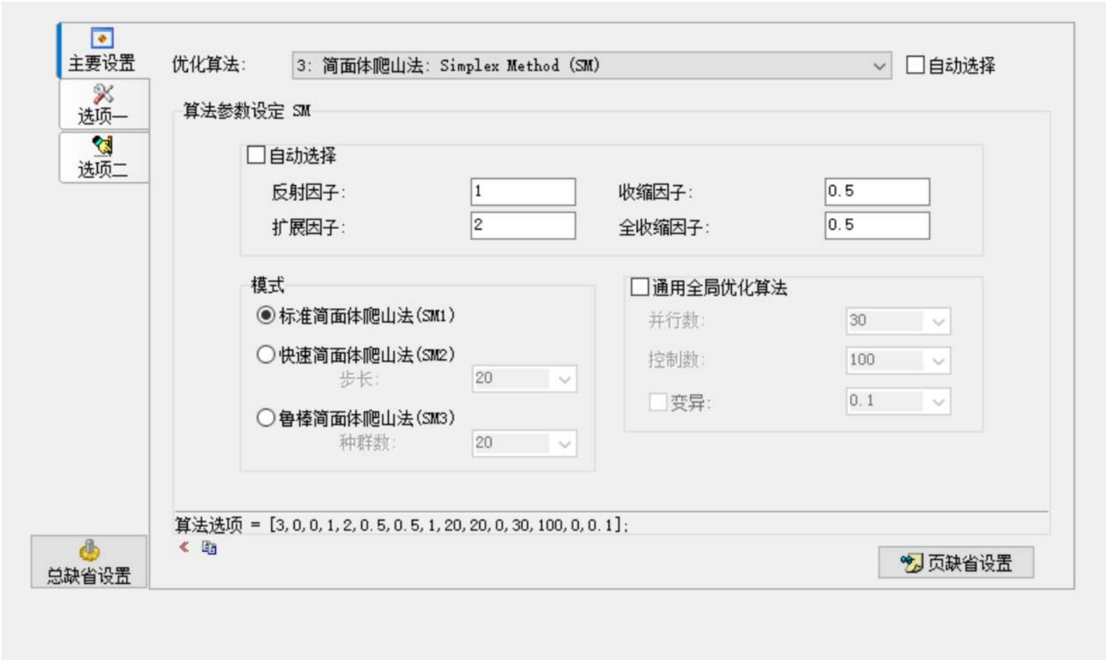

图 7-1. 算法设定

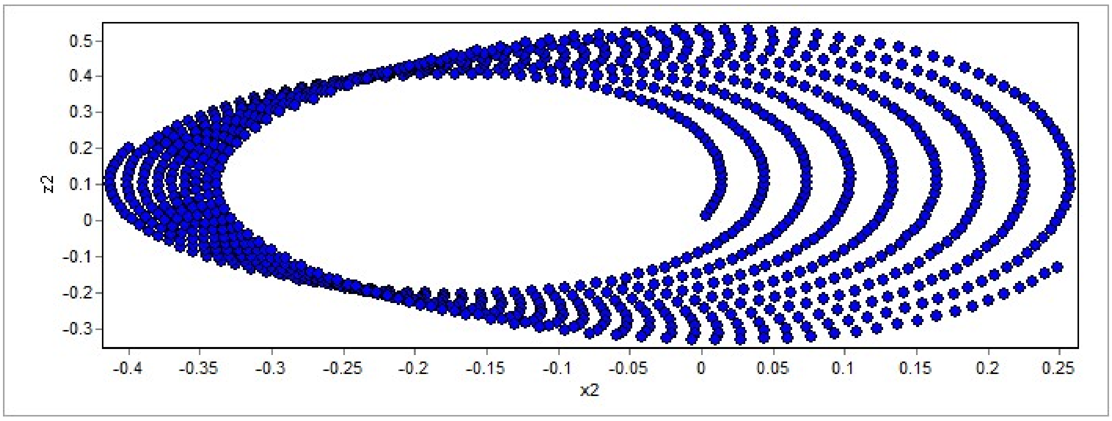

图 7-2. 计算图形一

如果将“PlotLoopData x2[x],z2;”分别改成“PlotLoopData z2[x],cos(y2*c2*x2);”和“PlotLoopData z2[x],cos(y2*c2+x2-z2);”，则可分别得到如下两图：

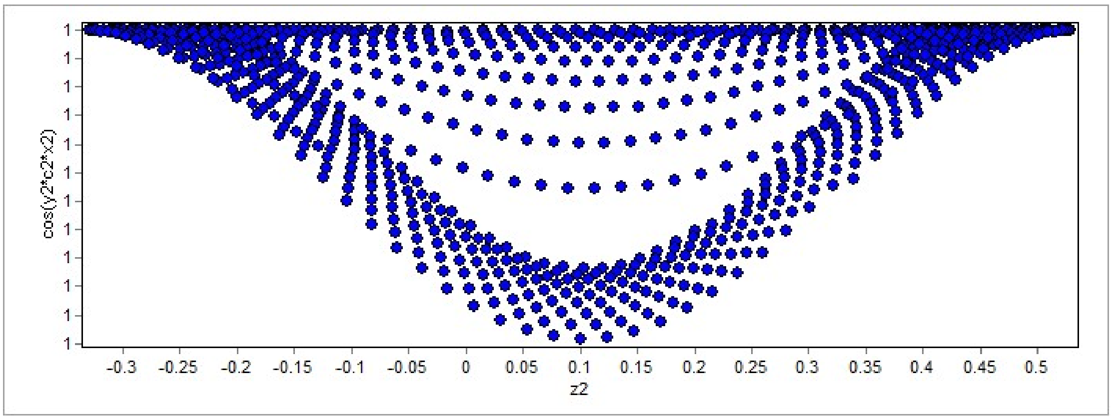

图 7-3. 计算图形二

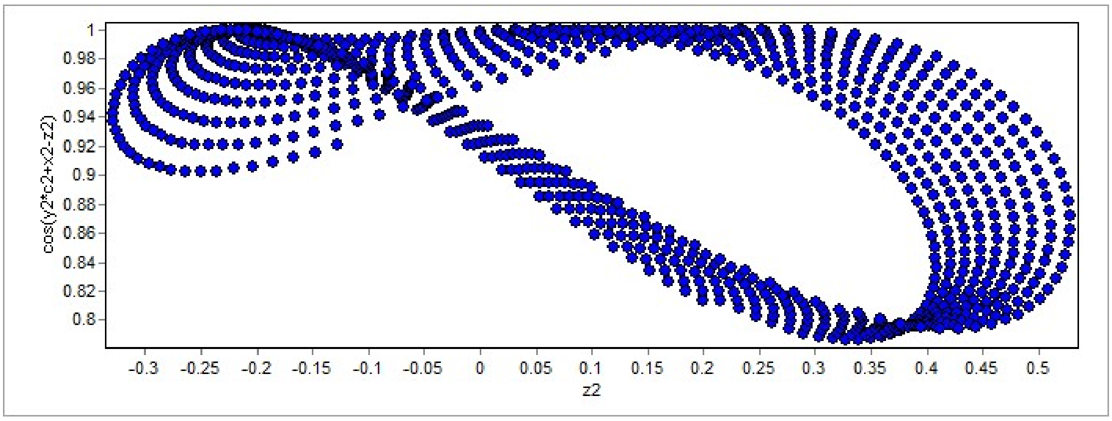

图 7-4. 计算图形三

7.2 案例二

差分方程组如下：

$$\begin{cases} q = q_0 - h \cdot b_0 + \frac{e \cdot (q-q_0)^2}{h} - \frac{h^2 \cdot q_0 \cdot t}{5000} \\ b = b_0 + \frac{h \cdot (q+q_0)}{2} - \frac{e \cdot (q-q_0)^2}{h^2} \end{cases} \tag{7-2}$$

其中：e=0.1, h=0.1, t=[0:1:1000]; q 和 b 的起始值：q0=0.01,b0=0.01；

代码 7-2

```
Constant e=0.1,h=0.1;
ChartType = 1;
LoopConstant t=[0:1:1000],q0=[0.01,q],b0=[0.01,b];
PlotLoopData q0[x],b0;
Function
q=q0-h*b0+e*(q-q0)^2/h-h*h*q0*t/5000;
b=b0+h*(q+q0)/2-e*((q-q0)^2)/(h^2);
```

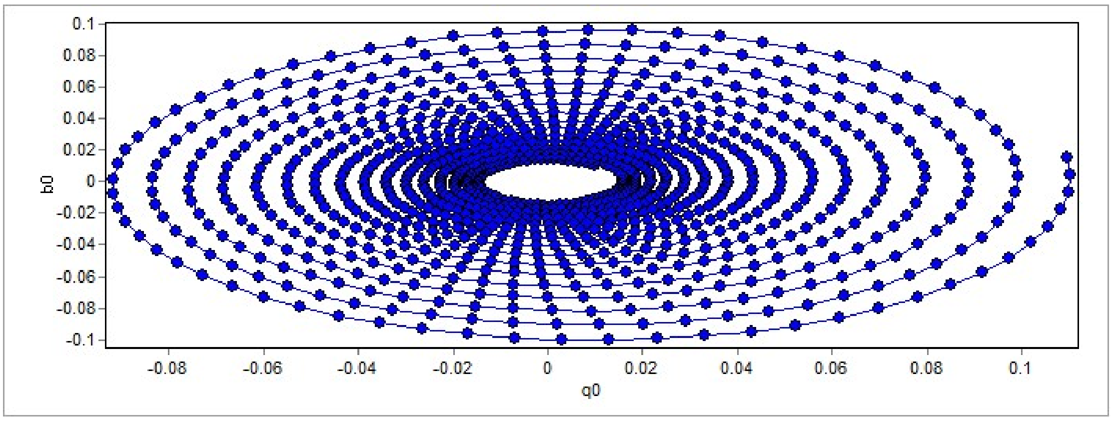

图 7-5. 计算图形四

如果将“PlotLoopData q0[x],b0;”改成“PlotLoopData q0*b0[x],b0;”，可得下图：

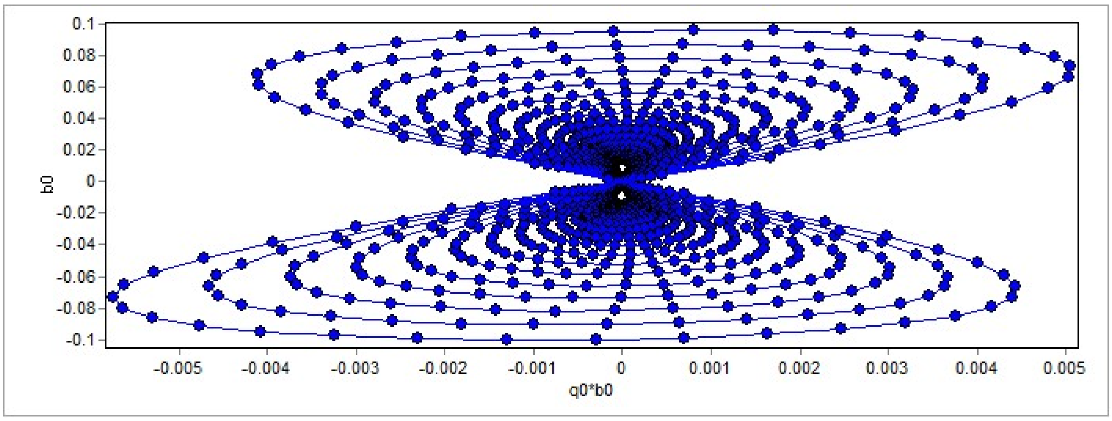

图 7-6. 计算图形五

7.3 案例三

差分方程组如下：

$$\begin{cases} X_{i+1} = a \cdot X_i - c \cdot Y_i + d \\ Y_{i+1} = b \cdot Y_i - c \cdot X_i - d \end{cases} \tag{7-3}$$

其中：n=20, i=1:n-1，Xi, Yi 为未知数，a、b、c、d 已知，分别为 1，1.2，1.5 和 2，且 X(0), Y(n)为已知量，分别为 x[0]=0.1, y[n]=2.5;

与前述案例一和案例二不同的是，两个边界值分别位于起始和终止点，类似于边值微分方程问题。

该问题可视为共有 40 个方程组成的方程组求解问题，因为方程组是线性，因此算法选用线性算法（Algorithm = LP;）

代码 7-3

```
Algorithm = LP;
Constant n=20, a=1,b=1.2,c=1.5,d=2,x[0]=0.1, y[n]=2.5;
Parameter x(n),y(0:n-1);
//PassParameter x(1:n);//PassParameter x(1:20);
Function for(i=0:n-1)(x[i+1]=a*x[i]-c*y[i]+d);
         for(i=0:n-1)(y[i+1]=b*x[i]-c*x[i]+d);
```

结果：

目标函数值:	函数表达式:
-1.83075776760688E-13	1: x1-(1*0.1-1.5*y0+2)-0 = 4.44089209850063E-16
	2: x2-(1*x1-1.5*y1+2) = 0
参数最优解为:	3: x3-(1*x2-1.5*y2+2) = 8.88178419700125E-16
x1: 2.91044704562633	4: x4-(1*x3-1.5*y3+2) = 4.44089209850063E-16
x2: 1.95544704562633	5: x5-(1*x4-1.5*y4+2) = -4.44089209850063E-16
x3: 2.26514821615818	6: x6-(1*x5-1.5*y5+2) = 4.44089209850063E-16
x4: 2.14509938669003	7: x7-(1*x6-1.5*y6+2) = 0
x5: 2.16441608396121	8: x8-(1*x7-1.5*y7+2) = -4.44089209850063E-16

x6: 2.12971080797172	9: x9-(1*x8-1.5*y8+2) = -4.44089209850063E-16
x7: 2.10369804575426	10: x10-(1*x9-1.5*y9+2) = 8.88178419700125E-16
x8: 2.06206790934154	11: x11-(1*x10-1.5*y10+2) = 6.66133814775094E-16
x9: 2.00873202993096	12: x12-(1*x11-1.5*y11+2) = 1.33226762955019E-15
x10: 1.93666258913465	13: x13-(1*x12-1.5*y12+2) = -6.66133814775094E-16
x11: 1.84059200260358	14: x14-(1*x13-1.5*y13+2) = 2.22044604925031E-16
x12: 1.71209016771418	15: x15-(1*x14-1.5*y14+2) = 2.22044604925031E-16
x13: 1.54035656888579	16: x16-(1*x15-1.5*y15+2) = 5.55111512312578E-16
x14: 1.31079714435717	17: x17-(1*x16-1.5*y16+2) = 8.88178419700125E-16
x15: 1.00395760035577	18: x18-(1*x17-1.5*y17+2) = 2.08943973234454E-13
x16: 0.593816315316499	19: x19-(1*x18-1.5*y18+2) = -3.66373598126302E-13
x17: 0.0455972354765977	20: x20-(1*x19-1.5*y19+2) = 4.44089209850063E-16
x18: -0.687185422630769	21: y1-1.97 = 0
x19: -1.66666666666667	22: y2-(1.2*x1-1.5*x1+2) = 2.22044604925031E-16
x20: -2.97590010685046	23: y3-(1.2*x2-1.5*x2+2) = -2.22044604925031E-16
y0: -0.540298030417553	24: y4-(1.2*x3-1.5*x3+2) = 2.22044604925031E-16
y1: 1.97	25: y5-(1.2*x4-1.5*x4+2) = 2.22044604925031E-16
y2: 1.1268658863121	26: y6-(1.2*x5-1.5*x5+2) = 0
y3: 1.4133658863121	27: y7-(1.2*x6-1.5*x6+2) = 0
y4: 1.32045553515255	28: y8-(1.2*x7-1.5*x7+2) = -2.22044604925031E-16
y5: 1.35647018399299	29: y9-(1.2*x8-1.5*x8+2) = 2.22044604925031E-16
y6: 1.35067517481164	30: y10-(1.2*x9-1.5*x9+2) = 2.22044604925031E-16
y7: 1.36108675760848	31: y11-(1.2*x10-1.5*x10+2) = 4.44089209850063E-16
y8: 1.36889058627372	32: y12-(1.2*x11-1.5*x11+2) = 0
y9: 1.38137962719754	33: y13-(1.2*x12-1.5*x12+2) = 4.44089209850063E-16
y10: 1.39738039102071	34: y14-(1.2*x13-1.5*x13+2) = 0
y11: 1.4190012232596	35: y15-(1.2*x14-1.5*x14+2) = 4.44089209850063E-16
y12: 1.44782239921892	36: y16-(1.2*x15-1.5*x15+2) = 2.22044604925031E-16
y13: 1.48637294968575	37: y17-(1.2*x16-1.5*x16+2) = 2.22044604925031E-16
y14: 1.53789302933426	38: y18-(1.2*x17-1.5*x17+2) = -2.22044604925031E-16
y15: 1.60676085669285	39: y19-(1.2*x18-1.5*x18+2) = -3.33066907387547E-14
y16: 1.69881271989327	40: 2.5-(1.2*x19-1.5*x19+2) = 0
y17: 1.82185510540505	
y18: 1.98632082935702	
y19: 2.2061556267892	

7.4 案例四

差分方程组如下：

$$\begin{cases} X_{i+1} = \frac{X_i}{1+R_1\cdot Y_{i+1}+R_2} \\ Y_{i+1} = \frac{Y_i}{1+R_1\cdot X_{i+1}-R_2} \end{cases} \tag{7-4}$$

其中：n=20, i=1:n-1，Xi, Yi 为未知数，R1=0.5, R2=2.3, x[1]=0.5, y[n+1]=0.38;

该方程组与案例三类似，唯一不同的是该差分方程组是非线性的，不能使用线性算法，在此使用缺省的通用全局优化算法，此外注意，下列求解代码将除法

形式的方程改为等效的乘法形式，这种变换有利于获得稳定最优解。

代码 7-4

```
Constant n=20, R1=0.5, R2=2.3, x[1]=0.5, y[n+1]=0.38;
Function For(i=1:n)(x[i+1]*(1+R1*Y[i+1]+R2)=x[i]);
         For(i=1:n)(y[i+1]*(1+R1*X[i+1]-R2)=y[i]);
```

结果：

目标函数值(最小):	函数表达式
3.87522316243264E-28	1: x2*(1+0.5*y2+2.3)-0.5 = -3.33066907387547E-16
参数最优解为:	2: x3*(1+0.5*y3+2.3)-x2 = 1.33226762955019E-15
x2: 0.291336104777615	3: x4*(1+0.5*y4+2.3)-x3 = 4.6074255521944E-15
x3: 0.0640432757786634	4: x5*(1+0.5*y5+2.3)-x4 = -2.22044604925031E-15
x4: 1.8349075467124	5: x6*(1+0.5*y6+2.3)-x5 = -3.44169137633799E-15
x5: 0.298946392518715	6: x7*(1+0.5*y7+2.3)-x6 = 9.43689570931383E-16
x6: 0.486941510273067	7: x8*(1+0.5*y8+2.3)-x7 = 2.44249065417534E-15
x7: 0.0895132409411884	8: x9*(1+0.5*y9+2.3)-x8 = -8.88178419700125E-16
x8: 1.27490860905192	9: x10*(1+0.5*y10+2.3)-x9 = 8.88178419700125E-16
x9: 0.212295463392206	10: x11*(1+0.5*y11+2.3)-x10 = -3.33066907387547E-16
x10: 0.820993126894028	11: x12*(1+0.5*y12+2.3)-x11 = -7.21644966006352E-16
x11: 0.14216637370406	12: x13*(1+0.5*y13+2.3)-x12 = 2.4980018054066E-16
x12: 0.10822980459143	13: x14*(1+0.5*y14+2.3)-x13 = 6.55725473919233E-16
x13: 0.0223556039195754	14: x15*(1+0.5*y15+2.3)-x14 = 8.93382590128056E-16
x14: 0.0105973554816186	15: x16*(1+0.5*y16+2.3)-x15 = 1.52308721190764E-15
x15: 0.00251323450699076	16: x17*(1+0.5*y17+2.3)-x16 = 8.57387078001537E-16
x16: 0.000968623748804805	17: x18*(1+0.5*y18+2.3)-x17 = 9.8841291054641E-16
x17: 0.000252072888951193	18: x19*(1+0.5*y19+2.3)-x18 = 8.29495977114347E-16
x18: 8.74471069328123E-5	19: x20*(1+0.5*y20+2.3)-x19 = 8.99748889759619E-16
x19: 2.41493295640813E-5	20: x21*3.49-x20 = 6.02926522185084E-16
x20: 7.91003205431514E-6	21: y2*(1+0.5*x2-2.3)-y1 = -4.44089209850063E-16
x21: 2.26648482948942E-6	22: y3*(1+0.5*x3-2.3)-y2 = -8.88178419700125E-16
y1: 3.65639074006565	23: y4*(1+0.5*x4-2.3)-y3 = 0
y2: -3.16753837371675	24: y5*(1+0.5*x5-2.3)-y4 = 1.77635683940025E-15
y3: 2.49810128340368	25: y6*(1+0.5*x6-2.3)-y5 = -8.88178419700125E-16
y4: -6.53019454751994	26: y7*(1+0.5*x7-2.3)-y6 = -8.88178419700125E-16
y5: 5.67582999916967	27: y8*(1+0.5*x8-2.3)-y7 = 8.88178419700125E-16
y6: -5.37214660811699	28: y9*(1+0.5*x9-2.3)-y8 = 3.5527136788005E-15
y7: 4.2797649409878	29: y10*(1+0.5*x10-2.3)-y9 = 0
y8: -6.45957700762919	30: y11*(1+0.5*x11-2.3)-y10 = 8.88178419700125E-16
y9: 5.41070045191294	31: y12*(1+0.5*x12-2.3)-y11 = -8.88178419700125E-16
y10: -6.08283254405463	32: y13*(1+0.5*x13-2.3)-y12 = -4.44089209850063E-16
y11: 4.94975125978866	33: y14*(1+0.5*x14-2.3)-y13 = 8.88178419700125E-16
y12: -3.97287941633564	34: y15*(1+0.5*x15-2.3)-y14 = -2.22044604925031E-15
y13: 3.08256594460959	35: y16*(1+0.5*x16-2.3)-y15 = 4.21884749357559E-15
y14: -2.38090893367643	36: y17*(1+0.5*x17-2.3)-y16 = -4.44089209850063E-15
y15: 1.83324047329614	37: y18*(1+0.5*x18-2.3)-y17 = 5.10702591327572E-15
y16: -1.41071053627708	38: y19*(1+0.5*x19-2.3)-y18 = -6.55031584528842E-15
y17: 1.08526716884027	39: y20*(1+0.5*x20-2.3)-y19 = 8.65973959207622E-15
y18: -0.834848978003234	40: 0.38*(1+0.5*x21-2.3)-y20 = -1.11022302462516E-14

y19: 0.64219748640201 y20: -0.493999569367871	

7.5 小结

1stOpt 可以很容易处理求解不同类型的差分方程或方程组，代码编写中请特别注意求解代码中“LoopConstant”及“For()”两个关键字的用法。