32. 1stOpt 扩散模型（Diffusion Model）参数优化求解

32.1 问题的提出

已知某扩散模型如下：

$$q(t) = q_e \cdot \left(1 - \sum_{i=1}^{n} \left(\frac{0.66 \cdot \exp\left(-D \cdot p_i^2 \cdot t\right)}{9.9 + 0.01 \cdot p_i^{0.2}}\right)\right) \tag{32-1}$$

其中q_e、D和n为待求参数（n为整数），p_i为满足下列方程的有序系列的非零非负根：

$$\tan(p_i) = \frac{3 \cdot p_i}{3 + 0.1 \cdot p_i^2} \tag{32-2}$$

公式32-1中，t为自变量，q为因变量，具体数值如下表。

表 32-1 已知数据：

t	0,5,10,15,20,30,45,60,75,90,105,120
q(t)	0,3.87,4.62,4.98,5.21,5.40,5.45,5.50,5.51,5.52,5.54,5.53

32.2 问题分析

该问题初看就是一简单的非线性拟合问题，t和q分别为自变量和因变量，q_e、D和n为三个待求未知参数，但稍加仔细分析就可以看出难点：参数n为一未知的整数，不同的n值还需要求出从1到n对应的有序系列p值，该系列p值又只能通过方程32-2数值解求得，因此解决该问题的第一步就是要按顺序（1至n）求出对应的p值，有了系列p值，才可对公式32-1进行拟合求解。

32.3 问题求解

方程32-2含有周期性三角函数，意味着多解，如何求出有序p系列值？首先对方程32-2作图，以了解该方程解的周期性分布变化情况，下面代码展示p从0.1至20间的根分布情况：

代码-1：图解根系列分布之一

```
ChartType = 2;
Variable p=[0.1:0.01:20],y;
PlotFunction y= tan(p)-3*p/(3+0.1*p^2);
```

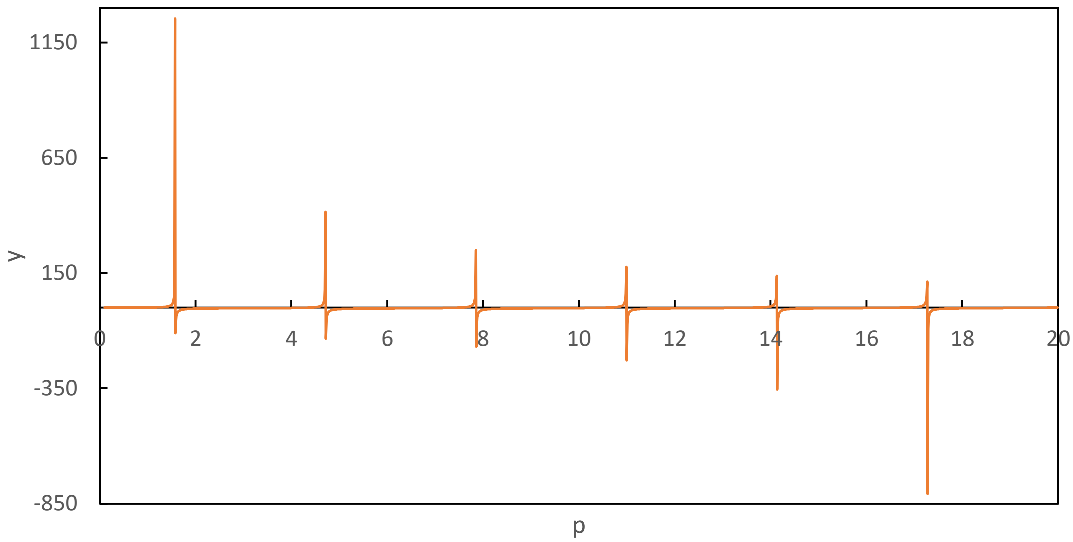

图32-1 根系列分布图之一

因为纵坐标最小最大值均距零点过远，导致由图32-1不易看清p系列根分布情况，将前述代码稍加修改如下，大于或小于5的部分均不予显示，从而可以更清晰展示根分布情况，如图32-2示。

代码-2：图解根系列分布之二

```
ChartType = 2;
ConstStr f=tan(p)-3*p/(3+0.1*p^2);
Variable p=[0.1:0.01:20],y;
PlotFunction y= if(f>5,5,if(f<-5,-5,f));
```

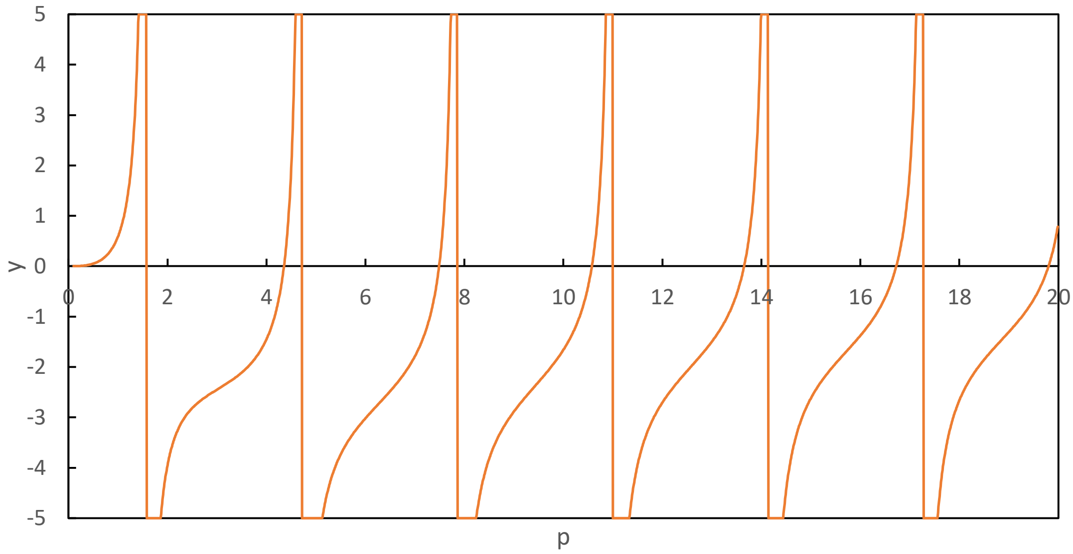

图32-2 根系列分布图之二

由图32-2可以看出，从2开始，大概每隔3.1就会有一个根存在。如何按要求有序求出p系列根值？在此使用关键字“LoopConstant”定义两个循环常数m和v，分布对应每次求解的上下范围区间，如下代码，可以自动一次性求出p在范围[2,104]内的所有33个根，如表32-2示。

代码-3 自动求指定区间根

```
LoopConstant m=[2:3.1:101];
LoopConstant v=[5.1:3.1:104];
EnhancedBound = 1;
Parameter p=[m,v];
Function tan(p)=3*p/(3+0.1*p^2);
```

表32-2 前33组p系列根

No.	p	No.	p
1	4.35378367052702	18	57.0291615831323
2	7.48808641354398	19	60.1496225761141
3	10.5751624468289	20	63.2717206388126
4	13.6511427167985	21	66.3952889411584
5	16.7257079845738	22	69.520180089537
6	19.8025583223923	23	72.6462638614857
7	22.8832678483174	24	75.7734251293724
8	25.9684490866884	25	78.9015619953762
9	29.0582237629262	26	82.0305841384253
10	32.1524537034099	27	85.1604113617219
11	35.2508675693038	28	88.2909723238722
12	38.3531340595535	29	91.4222034345828
13	41.4589045596549	30	94.5540478954731
14	44.5678374146667	31	97.6864548680625
15	47.6796109564739	32	100.819378752088
16	50.7939296792504	33	103.952778559431
17	53.9105263205756		

知道了对应拟合公式32-1中的系列p值，假设n为小于等于33的整数，则拟合计算就可以容易进行了，有一点需要注意的是因为n为未知整数，拟合公式的展开形式随着n值的不同而动态变化，因此无法在快捷模式下实现，只能在编程模式下完成。下面分别给出Pascal和Fortran编程模式实现代码，二者均可得到相同的结果。

代码-4 Pascal编程模式代码

```
Constant
p(1:33)=[4.35378367052701,7.48808641354398,10.5751624468592,13.6511427168261,16.7257079845738,19.8025583223684,22.8832678483399,25.9684490866884,29.0582237629463,32.1524537034099,35.2508675693218,38.3531340595364,41.4589045596711,44.5678374146667,47.6796109564883,50.7939296792504,53.9105263205883,57.0291615831204,60.1496225761141,63.2717206388126,66.3952889411584,69.520180089537,72.6462638614857,75.7734251293724,78.9015619953762,82.0305841384314,85.1604113617219,88.2909723238772,91.4222034345828,94.5540478954731,97.6864548680592,100.819378752091,103.952778559431];
IntParameter n=[2,33];
Parameter qe,D;
Variable t,q;
StartProgram [Pascal];
Procedure MainModel;
var i, j: integer;
      temd: double;
Begin
      for i := 0 to DataLength - 1 do begin
            temd := 0;
            for j := 1 to n do begin
                  temd := temd + (0.66*exp(-D*p[j]^2*t[i]))/(9.9+0.01*p[j]^0.2);
            end;
            q[i] := qe*(1-temd);
      end;
End;
```

```
EndProgram;
Data;
t= 0 5 10 15 20 30 45 60 75 90 105 120;
qt= 0 3.87 4.62 4.98 5.21 5.40 5.45 5.50 5.51 5.52 5.54 5.53;
```

代码-5 Fortran编程模式代码

```
Constant
p(1:33)=[4.35378367052701,7.48808641354398,10.5751624468592,13.6511427168261,16.7257079845738,19.8025583223684,22.8832678483399,25.9684490866884,29.0582237629463,32.1524537034099,35.2508675693218,38.3531340595364,41.4589045596711,44.5678374146667,47.6796109564883,50.7939296792504,53.9105263205883,57.0291615831204,60.1496225761141,63.2717206388126,66.3952889411584,69.520180089537,72.6462638614857,75.7734251293724,78.9015619953762,82.0305841384314,85.1604113617219,88.2909723238772,91.4222034345828,94.5540478954731,97.6864548680592,100.819378752091,103.952778559431];
IntParameter n=[2,33];
Parameter qe,D;
Variable t,q;
StartProgram [Fortran];
Subroutine MainModel
integer i, j
real(8) temd
      do i = 0, DataLength - 1
            temd = 0
            do j = 1, n
                  temd = temd + (0.66*exp(-D*p(j)^2*t(i)))/(9.9+0.01*p(j)^0.2)
            end do
            q(i) = qe*(1-temd)
      end do
End Subroutine
EndProgram;
Data;
t= 0 5 10 15 20 30 45 60 75 90 105 120;
qt= 0 3.87 4.62 4.98 5.21 5.40 5.45 5.50 5.51 5.52 5.54 5.53;
```

结果：

```
Sum Squared Error (SSE): 0.101453491052375
Root of Mean Square Error (RMSE): 0.0919481244381741
Correlation Coef. (R): 0.99815678524769
R-Square: 0.996316967936003
Adjusted R-Square: 0.995498516366226
Determination Coef. (DC): 0.996310096716061
F-Statistic: 1208.97656161048

Parameter  Best Estimate
---------  -------------
n       15
qe             5.76656124275871
d       0.000546668972855795
```

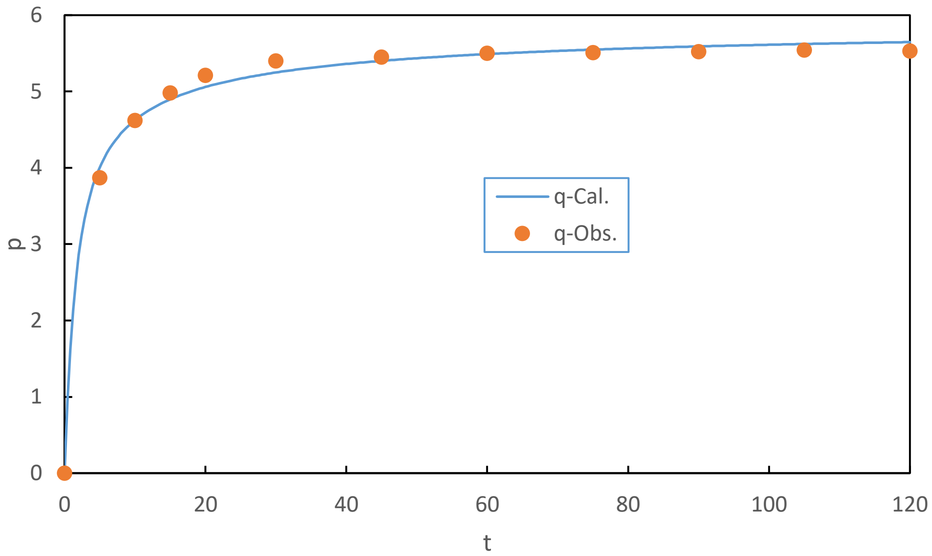

图32-3 拟合计算对比图

前面计算过程及结果是按假设n小于等于33进行的，虽然计算结果n为15，说明假设是正确的，但如果计算结果n等于上限值33，则表明上限值33的假设是不合适的，有可能是50，100或更高，此时该如何处理？一种方法是扩大p值求解范围，如将代码-3中的循环常数语句”m=[2:3.1:101];”和“v=[5.1:3.1:104]”分别改为” m=[2:3.1:301];”和“v=[5.1:3.1:304]”，这样可以求得更多p系列值；另外还有种简单的近似方法，即基于表32-2计算结果，可以建立一个序列号n与对应的p值关系，如图32-4示，p与n的关系几近线性直线关系，在此选择线性直线函数，二次多项式函数及通过1stOpt公式自动搜索匹配计算得到的非线性公式，如表32-3.

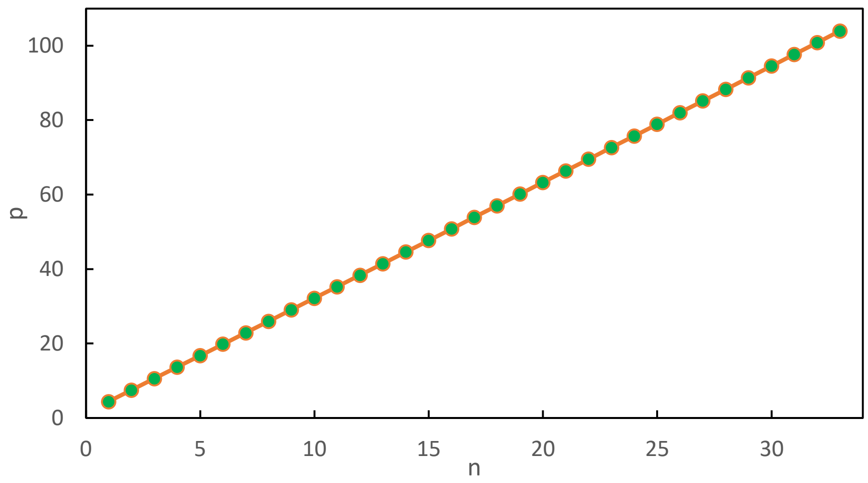

图32-4 p系列解与序列号关系图

表 32-3 p 系列值近似计算公式

序号	公式类别	公式形式
1	线性公式	$y = p_1 + p_2 \cdot x$
2	二次多项式公式	$p_1 + p_2 \cdot x + p_3 \cdot x^2$
3	非线性公式	$y = p_1 + p_2 \cdot x + p_3 \cdot x^2 + p_4 \cdot x^3 + \frac{p_5}{x^{p_6}} + p_7 \cdot \exp(p_8 \cdot x)$

线性及二次多项式的拟合计算非常简单，非线性公式的计算代码见下，三种模型的对比结果见表32-4.

代码-6 非线性拟合公式计算

```
Function y=p1+p2*x+p3*x^2+p4*x^3+p5/x^p6+p7*exp(p8*x);
Data;
x=[1:1:33];
y=[4.35378367052701,7.48808641354398,10.5751624468592,13.6511427168261,16.7257079845738,19.802
5583223684,22.8832678483399,25.9684490866884,29.0582237629463,32.1524537034099,35.25086756932
18,38.3531340595364,41.4589045596711,44.5678374146667,47.6796109564883,50.7939296792504,53.910
5263205883,57.0291615831204,60.1496225761141,63.2717206388126,66.3952889411584,69.52018008953
7,72.6462638614857,75.7734251293724,78.9015619953762,82.0305841384314,85.1604113617219,88.2909
723238772,91.4222034345828,94.5540478954731,97.6864548680592,100.819378752091,103.95277855943
1];
```

表32-4 三种模型计算结果

<table>
<tr><th>模型</th><th>结果</th></tr>
<tr><td>线性</td><td>Sum Squared Error (SSE): 0.250273722191455
Root of Mean Square Error (RMSE): 0.0870864638598896
Correlation Coef. (R): 0.999995683397244
R-Square: 0.999991366813121

Parameter Best Estimate
--------- -------------
b0 1.0922568764445
b1 3.11271519740032</td></tr>
<tr><td>多项式</td><td>Sum Squared Error (SSE): 0.0108575546483993
Root of Mean Square Error (RMSE): 0.0181388204557863
Correlation Coef. (R): 0.999999812734421
R-Square: 0.999999625468877

Parameter Best Estimate
--------- -------------
b0 1.3008607718853
b1 3.07695452961046
b2 0.00105178434676002</td></tr>
<tr><td>非线性</td><td>Sum Squared Error (SSE): 1.94771212059894E-7
Root of Mean Square Error (RMSE): 7.6825503195286E-5
Correlation Coef. (R): 0.999999999996641
R-Square: 0.999999999993281

Parameter Best Estimate
--------- -------------
p1 -4.1009742742895
p2 3.12749071784384
p3 -0.000122689644322487
p4 2.8765722643023E-6
p5 3.96337937292846
p6 -0.0576037524256348
p7 1.53820681929552
p8 -0.120175937532826</td></tr>
</table>

基于上述计算结果，参考代码-4，可进行近似拟合计算，下面仅给出第三种非线性模型的计算代码，全部计算对比结果见表32-5，由该表对比结果看，采用非线性模式计算的近似结果与精确解最为接近，说明在特定情况下可以考虑采用这种近似的计算方式。

代码-7 非线性模型近似计算

```
Constant p(8)=[-4.1017248276084,3.12749314377722,-0.000122737568113045,2.87696411865041E-
6,3.96411097483594,-0.0575926782013512,1.5382236033313,-0.120174704088587];
```

```
ConstStr f(k)=p1+p2*k+p3*k^2+p4*k^3+p5/k^p6+p7*exp(p8*k);
IntParameter n=[2,100];
Parameter qe,D;
XAxis = t;
Variable t,q;
StartProgram [Pascal];
Procedure MainModel;
var i, j: integer;
    temd, tq: double;
Begin
    for i := 0 to DataLength - 1 do begin
        temd := 0;
        for j := 1 to n do begin
            tq := f(j);
            temd := temd + (0.66*exp(-D*tq^2*t[i]))/(9.9+0.01*tq^0.2);
        end;
        q[i] := qe*(1-temd);
    end;
End;
EndProgram;
Data;
t= 0 5 10 15 20 30 45 60 75 90 105 120;
qt= 0 3.87 4.62 4.98 5.21 5.40 5.45 5.50 5.51 5.52 5.54 5.53;
```

表 32-5 近似计算结果对比

计算方式	计算结果
精确解	Sum Squared Error (SSE): 0.101453491052375 Root of Mean Square Error (RMSE): 0.0919481244381741 Correlation Coef. (R): 0.99815678524769 R-Square: 0.996316967936003 Parameter Best Estimate --------- ------------- n 15 qe 5.76656124275871 d 0.000546668972855795
线性	Sum Squared Error (SSE): 0.105917258057039 Root of Mean Square Error (RMSE): 0.0939491254425497 Correlation Coef. (R): 0.998075161563187 R-Square: 0.996154028129382 Parameter Best Estimate --------- ------------- n 15 qe 5.78175649701827 d 0.000546814659710458
多项式	Sum Squared Error (SSE): 0.103004531093148 Root of Mean Square Error (RMSE): 0.0926483185191667 Correlation Coef. (R): 0.998128558277179 R-Square: 0.99626061884848 Parameter Best Estimate --------- ------------- n 15 qe 5.76511224323693 d 0.000548624457556943
非线性	Sum Squared Error (SSE): 0.101452445242983 Root of Mean Square Error (RMSE): 0.0919476505241721 Correlation Coef. (R): 0.998156804302708 R-Square: 0.996317005975794 Parameter Best Estimate --------- ------------- n 15 qe 5.76656294399161 d 0.000546667311814592

32.4 小结

通过实际案例，演示了1stOpt如何进行自动循环求解特定范围的方程解以及如何采取近似计算方法处理整数参数带来的模型动态变化问题，对相关扩散模型或类似问题的有效解决提供有益的参考和帮助。