# 地质信息处理技术与应用

董大忠　胡素云　关春林　编著

石油工业出版社
Petroleum Industry Press

# 地质信息处理技术与应用

董大忠　胡素云　关春林　编著

石 油 工 业 出 版 社

## 内 容 提 要

本书介绍了30多种用于石油地质信息的处理方法，由时间序列分析法、回归预测及决策模型法、数理统计概算法和图形信息处理法四部分组成。对于每个算法都给出了相应的实际算例。

本书主要适用于油田地质信息处理人员，以及各院校相关专业人士。

图书在版编目（CIP）数据

地质信息处理技术与应用/董大忠，胡素云，关春林编著.
北京：石油工业出版社，2008.1
ISBN 978-7-5021-6356-3

Ⅰ.地
Ⅱ.①董… ②胡… ③关…
Ⅲ.石油天然气地质-信息处理系统-研究
Ⅳ.P618.130.2

中国版本图书馆CIP数据核字（2007）第181442号

出版发行：石油工业出版社
（北京安外安华里2区1号　100011）
网　址：www.petropub.com.cn
编辑部：(010)64523589　发行部：(010)64210392
经　销：全国新华书店
印　刷：保定彩虹印刷有限公司

2008年1月第1版　2008年1月第1次印刷
787×1092毫米　开本：1/16　印张：15
字数：382千字　印数：1—1000册

定价：45.00元
（如出现印装质量问题，我社发行部负责调换）

# 前　　言

预测或决策方法，在采矿业的勘探和开发中是经常使用的重要计算方法。在石油与天然气勘探开发的生产、科研和管理中，对油气资源量或储量的预测、勘探目标决策和经济指标分析等，都要用到有关的预测和决策理论与技术。

预测和决策算法多种多样，对不同的信息采集条件、不同的时间阶段、不同的目的要求，可选择不同的处理方法。例如，要从机理和成因规律来预测某区域的油气资源量，此时应选择盆地模拟法，由此需要提供与“五史”有关信息。再如，根据历年某探区的资源发现量的记录，来预测今后的发现量，这只需用时间序列法或用回归模型法进行外延预测。

对预测来说，实际上是先进行历史拟合、后进行预测（有时是这两种方式交替进行）；也就是说对当前使用的预测模型，应进行历史检验。这样的计算过程是合理的，预测的结果也是可信的（如油田开发数值模拟软件即是如此）。同样，使用时间序列法、回归模型法和数理统计中的概算法，都含有历史拟合的过程。因为测量值（历史记录）与预测值总是有误差的，找出使得误差总和达到最小的相关参数，组成新的预测式，进行下步预测。再一种办法，是把各项的误差看成一个新的序列 $e_i = y_i - f_i\ (i=1, \cdots, k)$，对序列$\{e_i\}$进行自相关分析，通过自相关系数计算出统计量，查 $\chi^2$ 分布数值表，利用判别式进行显著性检验，以此判断预测结果的可信度。本书中给出的时序量预测法和回归模型预测法，都对预测误差序列$\{e_i\}$进行了统计检验。

本书介绍的 30 多种用于石油地质信息处理方法，由时间序列法、回归模型预测法及决策模型法、数理统计概算法和图形信息处理法 4 部分组成。这些是笔者在 20 世纪末的近 10 年中，在科研项目中与有关其他专业人员的多年合作基础上完成的。书中每个算法都给出了结合实际的算例，对算法也进行了简要介绍，并有意避开较深奥的理论问题。同时也指出各种方法应用环境和它们的优缺点，以利于对不同的实际资料，选取有效的处理方法。

笔者认为图形处理技术也是预测或决策工作中的有效方法。图形处理技术不仅仅是图形的构建、图形的绘制和图形的变换等问题，因为图形是信息最丰富的数据集合，类同于数据一样，可对它进行加减乘除计算。所以本书在最后部分，简单地介绍了图形的生成、图与数的加乘计算、图与图的加乘叠合计算、图形的填充或图形的挖补计算等。例如，图和图的乘法计算可预测某区域的油气资源量，用图形的叠合法可对勘探风险进行决策，用图形的挖补功能有利于区带评价研究。

书中介绍的所有方法都给出了用 C++算法语言编写的源程序，并且每个算例都有中文提示和中文输出显示。

在本书的编写过程中，得到了中国石油勘探开发研究院资源规划所的领导和专家的热情指导和帮助，特别是书中的插图全部由规划所唐惠同志绘制，笔者在此表示真诚的感谢！

因笔者水平有限，书中难免有不妥和不足之处，请读者不吝赐教。

# 目　　录

# §1 时序量预测法

时序量预测法，即是时间序列预测法。这种预测方法只关心原始数据的表现形式，不考虑这些数据在系统内部是如何生成的。所以为了观察数据的形态，先画出图形，根据图的形态选取针对性的预测方法。这里给出 7 种方法：平滑指数法、趋势分析法、加权系数法、自回归系数法、残差辨识法、灰色建模法和维纳预测法。在预测过程中，根据序列历史数据（时序量）确定它是否为平稳的还是非平稳的，判别方法是由“自相关系数”来确定。

对原始数据组成的序列，经过预测后得到新的序列，同时也得到误差序列。如何判断预测模型（递推公式）是否可用于预测，同样要求给出误差序列的自相关系数，再根据自相关系数计算出一个统计量 $Q$，用它与“$\chi^2$ 数值表”中相应的数值进行比较，由此实现对误差序列的随机性检验。根据检验结果决定是否对序列的未来（下一步）进行预测。

这种处理方法是比较合理的，即先进行前 $N$ 项的预测（按递推公式计算），这个过程称为历史拟合。再通过随机性检验，观察它们是否基本遵循给定的模型轨迹。如果是，可进行后面（第 $N+1$ 项）的预测，预测的结果是有效的。如果序列不符合模型规律，说明已知的前 $N$ 项数据与 $N$ 个预测结果存在着显著性差异。所以对后面的预测结果只能作为参考。

## 1.1 平滑指数法

### 1.1.1 功能

对平稳的时间序列 $\{x_i \mid i=1,2,\cdots,n\}$ 进行外推预测。即由已知的 $n$ 个 $x_i$ 值利用指数平滑公式计算 $x_{n+1}$ 的近似值。

其用途：对等时间的平稳的观测数据进行外推预测。即由已知的观测序列$\{x_i,i=1,2,\cdots,n\}$的 $n$ 个值求出 $f_{n+1} \approx x_{n+1}$，同时求出它们的误差 $e_i = f_i - x_i$。

在讨论序列的平稳性之前先定义一个统计量 $B=1.96/\sqrt{n}$。其中，$n$ 是序列的长度（即观测值的个数）。求出时间序列的前 $k$ 个自相关系数 $\{r_i, i=1,2,\cdots,k\}$（一般 $k$ 取为 $n/2$ 的整数，记为 $k=[n/2]$，自相关系数的计算参看附录 1），如果 $k$ 个自相关系数全部落在区间 $(-B,B)$ 之内，此时就可得出以下结论。

有 95%的置信程度认为这些自相关系数 $r_i$ 与零没有显著性差异，此时可断定这个时间序列是一个随机时间序列。

所谓一个时间序列是平稳的，这是指此序列的 $k$ 个自相关系数 $r_1,r_2,\cdots,r_k$ 只有第一个 $r_1$ 落在区间（$-B,B$）之外，与零有显著性差异，其余的都落在区间之内，与零没有显著性差异（图 1-1-1），则序列$\{x_i\}(i=1,2,\cdots,n)$就称为平稳时间序列。

例如时间序列 $x=\{6，9，7，5，4，7，9，6，7，8，6，4，2，3，5，3，4，7，4，5\}$ 长度 $n=20$，$k=9$，其 $B=0.438269$。

其自相关系数为：0.442932，0.047298，0.139326，0.101268，−0.031521，−0.207672，−0.076951，0.144496，−0.079019。

将自相关系数和 $B$ 值作图（图 1-1-1），可见只有第一个自相关系数落在区间$(-B,B)$之外，由此表明序列$\{x\}$为平稳时间序列。

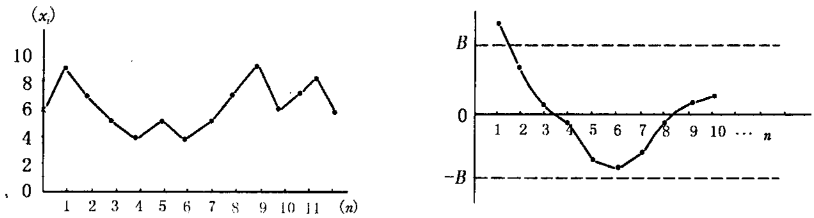


图 1-1-1　平稳时间序列的示意图（$B$=0.4277）

对平稳时间序列进行预测，是在预测式中对平滑指数自动调整，选取使得平均误差达到最小的平滑指数作为预测式中的平滑指数，用来进行最后的预测。

平滑指数法，主要用于等间隔平稳序列，如逐年的钻探进尺、油气勘探发现量、油气产量的预测等。

### 1.1.2　算法和步骤

（1）初始化：

设：时间序列为 $x_1,x_2,x_3,\cdots,x_n$。

如果令 $F_i$ 是 $x_i$ 的预测值，$E_i$ 是平滑误差，而 $M_i$ 是绝对平滑误差，$e_i$ 是预测误差，其值为：$e_i=x_i-F_i$。

在开始预测时要进行初始化，即：$F_2=x_1$；$E_1=0$；$M_1=0$；　$E_2=x_2-F_2=x_2-x_1$。

（2）计算 $i$ 时刻的平滑误差 $E_i$：

$$E_i=\beta e_i+(1-\beta)E_{i-1}\quad(i=2,3,\cdots,n)$$

其中，$\beta$ 为平滑指数，在区间（0.1～0.9）内变化。

（3）计算 $i$ 时刻的绝对平滑误差 $M_i$：

$$M_i=\beta|e_i|+(1-\beta)M_{i-1}\quad(i=2,3,\cdots,n)$$

（4）计算 $i$ 时刻的平滑值 $\alpha_i$：

$$\alpha_i=|E_i/M_{i-1}|\quad(i=2,3,\cdots,n)$$

（5）对 $i+1$ 时刻计算预测值：

$$F_{i+1}=\alpha_i x_i+(1-\alpha_i)F_i\quad(i=2,3,\cdots,n)\tag{1-1-1}$$

如此得到 $F_2,F_3,\cdots,F_n$，$F_{n+1}$，以及误差序列 $e_2,e_3,\cdots,e_n$。再由误差序列求出均方差 $S$（$\beta$）值：

$$S(\beta_j)=\frac{1}{n-1}\sum_{i=2}^{n}e_i^2$$

其中，$\beta_j$ 可取为 0.1，0.2，…，0.9。

改变 $\beta_j$ 值，重复第（2）步至第（5）步计算。

从 $S(\beta_j)$中选出最小的均方差 $S(\beta^*)=S^*$，确定 $\beta=\beta^*$ 为最佳平滑指数。

（6）固定 $\beta^*$执行第（2）步的计算。

用预测式（1-1-1）计算预测序列 $F_2,F_3,\cdots,F_n,F_{n+1}$ 和误差序列 $e_2,e_3,\cdots,e_n$，最后得到的 $F_{n+1}$ 即为预测结果。

（7）进行随机性检验。

求误差序列$\{e_i\}$的自相关系数$\{r_i\}(i=1,2,\cdots,k)$，其中 $k$ 是 $\frac{n-1}{2}$ 的整数部分。再计算检验值（统计量）：

$$Q=(n-1)\sum_{1}^{k} r_i^2$$

其中，$n$是时间序列的长度，$(n-1)$是误差序列的长度。查 $\chi^2$ 数据表，得到自由度为 $k$-1 置信度为 0.95 的 $\chi^2$ 值 xx，最后进行判断。当 $Q$≤xx 时，预测结果较好；否则，预测结果仅供参考。

注：程序中调用了求自相关系数子程序 zxgej 和查 $\chi^2$ 数据表子程序 xx95。

```
******a11 平滑指数法预测程序******
#include<fstream.h>
#include<math.h>
#include<string.h>
void main()
{  // a11.cpp 平滑指数法
   // 程序中调用子程序 (zxgej,xx95)
   // 输入文件 a11 的内容为:
   // n              (观测值的个数)
   // x1,x2,x3,…,xn (n 个观测值)
   // 例如
   // 16
   // 97 95 95 92 95 95 98 97 99 95 95 96 97 98 94 95
   cout<<"  平滑指数法=11"<<endl;
   cout<<endl;
   int zxgej(double [],double [],int);
   double xx95(int);
   double x[50],r[50],e[50],f[50],E[50];
   double m[50],a[50],t[50];
   double bt[10],s[10];
   double bb,b,ss,q,x2;
   int i,j,k,n;
   char fname[16];
   char *txt=".txt";
```

```
cout<<"->请输入参数文件名: ";
cin>>fname;
strcat(fname,txt);
ifstream rf(fname);
rf>>n;
cout<<"  数据个数  n="<<n<<endl;
for(i=0;i<n;i++)
    rf>>x[i];
cout<<"  x[i]:"<<endl;
for(i=0;i<n;i++)
    cout<<"  "<<x[i];
cout<<endl;
bb=1.96/sqrt(1.0*n);
cout<<"  B=1.96/sqrt("<<n<<")="<<bb<<endl;
cout<<"  计算 x[i]的自相关系数"<<endl;
k=zxgej(x,r,n);
for(i=0;i<k-1;i++)
    cout<<"    r["<<i<<"]="<<r[i]<<endl;
cout<<endl;
for(j=0;j<9;j++)
{
    bt[j]=0.1* (j+1);
    s[j]=0;
}
for(j=0;j<9;j++)
{
    ss=s[j];
    f[1]=x[0];
    E[0]=0;
    m[0]=0;
    for(i=1;i<n;i++)
    {
        e[i]=x[i]-f[i];
        E[i]=bt[j] *e[i]+(1-bt[j]) *E[i-1];
        m[i]=bt[j] *fabs(e[i])+(1-bt[j]) *m[i-1];
        t[i]=E[i]/m[i];
        a[i]=fabs(t[i]);
        f[i+1]=a[i] *x[i]+(1-a[i]) *f[i];
        s[j]=s[j]+e[i] *e[i]/(n-1);
    }
```

```
}
ss=s[0];
for(j=0;j<9;j++)
{
    cout<<"    平滑常数="<<bt[j]<<" "<<"均方差="<<s[j]<<endl;
    if(ss>s[j])
    {
        ss=s[j];
        b=bt[j];
    }
}
cout<<"    最小均方差="<<ss<<" 对应的平滑常数="<<b<<endl;
cout<<endl;
f[1]=x[0];
E[0]=0; m[0]=0;
for(i=1;i<n;i++)
{
    e[i]=x[i]-f[i];
    E[i]=b*e[i]+(1-b) *E[i-1];
    m[i]=b*fabs(e[i])+(1-b) *m[i-1];
    t[i]=E[i]/m[i];
    a[i]=fabs(t[i]);
    f[i+1]=a[i] *x[i]+(1-a[i]) *f[i];
}
cout<<"   i    x[i]      f[i]      e[i]"<<endl;
cout.setf(ios::fixed,ios::floatfield);
for(i=1;i<n;i++)
    cout<<"    "<<i<<" "<<x[i]<<" "<<f[i]<<" "<<e[i]<<endl;
cout<<endl;
cout<<"          bait 值 bt="<<b<<endl;
cout<<"          均方差 s="<<ss<<endl;
cout<<endl;
for(i=1;i<n;i++)
    x[i-1]=e[i];
bb=1.96/sqrt(1.0* (n-1));
cout<<"    B=1.96/sqrt("<<n-1<<")="<<bb<<endl;
cout<<"    计算 e[i]的自相关系数"<<endl;
k=zxgej(x,r,n-1);
for(i=0;i<k-1;i++)
    cout<<"     "<<r[i]<<endl;
```

```
    cout<<endl;
    cout<<"  计算 Q 值"<<endl;
    q=0;
    for(i=0;i<k-1;i++)
        q=q+r[i] *r[i];
    q=n*q;
    cout<<"      Q="<<q<<endl;
    cout<<"  查表找出自由度 df="<<k-1<<"置信度为 0.95 的表值 x2"<<endl;
    x2=xx95(k-1);
    cout<<"      x2="<<x2<<endl;
    cout<<endl;
    cout<<"      (预测值) x("<<n<<")="<<f[n]<<endl;
    if(q<=x2)
        cout<<"  因为 Q<x2 预测成功"<<endl;
    else
        cout<<"  因为 Q>x2 预测失败"<<endl;
    cout<<endl;
    cout<< "结束 a11 操作" <<endl;
}
```

**例 1.1 平滑指数法测试结果**

```
  平滑指数法=11
->请输入参数文件名:a11↙
  数据个数 n=16
  x[i]:
  97 95 95 92 95 95 98 97 99 95 95 96 97 98 94 95
  B=1.96/sqrt(16)=0.49
  计算 x[i]的自相关系数
   r[0]=0.202473 r[1]=0.0415545 r[2]=-0.45853 r[3]=-0.145693
   r[4]=-0.07781 r[5]=0.0587147 r[6]=0.032386
   平滑常数=0.1  均方差=6.15301
   平滑常数=0.2  均方差=6.20302
   平滑常数=0.3  均方差=6.18632
   平滑常数=0.4  均方差=5.67584
   平滑常数=0.5  均方差=5.36904
   平滑常数=0.6  均方差=5.07626
   平滑常数=0.7  均方差=4.80845
   平滑常数=0.8  均方差=4.62288
   平滑常数=0.9  均方差=4.6244
 最小均方差=4.62288  对应的平滑常数=0.8
```

| i | x[i] | f[i] | e[i] |
|---|---|---|---|
| 1 | 95.000000 | 97.000000 | -2.000000 |
| 2 | 95.000000 | 95.000000 | 0.000000 |
| 3 | 92.000000 | 95.000000 | -3.000000 |
| 4 | 95.000000 | 92.000000 | 3.000000 |
| 5 | 95.000000 | 93.977876 | 1.022124 |
| 6 | 98.000000 | 94.855699 | 3.144301 |
| 7 | 97.000000 | 97.955644 | -0.955644 |
| 8 | 99.000000 | 97.801507 | 1.198493 |
| 9 | 95.000000 | 98.698896 | -3.698896 |
| 10 | 95.000000 | 95.494043 | -0.494043 |
| 11 | 96.000000 | 95.040813 | 0.959187 |
| 12 | 97.000000 | 95.608980 | 1.391020 |
| 13 | 98.000000 | 96.915483 | 1.084517 |
| 14 | 94.000000 | 97.984737 | -3.984737 |
| 15 | 95.000000 | 94.523519 | 0.476481 |

bait 值 bt=0.800000

均方差 s=4.622882

B=1.96/sqrt(15)=0.506070

计算 e[i]的自相关系数

-0.193645 0.149049 -0.510476 -0.024122-0.060365 0.127119

计算 Q 值

Q=5.110277

查表找出自由度 df=6 置信度为 0.95 的 $\chi^2$ 值 x2

x2=12.590000

(预测值) x(16)=94.618395

因为 Q<x2 预测成功

结束 a11 操作

## 1.2 趋势分析法

### 1.2.1 功能

对具有趋势性变化的时间序列进行预测，也就是对线性增长或二次增长的非平稳的时间序列进行预测（此处的增长也可能是负增长）。所谓非平稳时间序列（图 1-2-1），是指它的自相关系数前几个（2～5 个）落在区间（$-B$，$B$）之外，与零有显著性差异，其他都落在区间之内。例如时间序列 $x$={1.0,0.6,2.0,2.2,2.6,4.0,4.0,5.4,6.8,6.0,6.8,7.5,10.0,8.9,9.3,9.0}

长度 $n=16$，$B=1.96/\sqrt{n}=0.49$，序列的自相关系数为：0.82913，0.64186，0.47415，0.26423，0.09649，-0.04233，-0.17368。

由此可见，自相关系数的前两个落在区间 $(-B,B)$ 之外，其他都落在区间之内。表明序列 $\{x\}$ 是一个非平稳的时间序列（图 1-2-1）。对于非平稳的时间序列应采用趋势分析法进行预测。

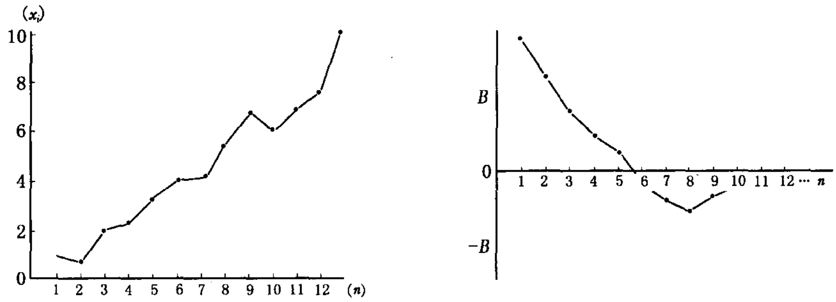


图 1-2-1　非平稳时间序列示意图($B$=0.49)

其用途：可以根据历年的油气勘探发现量，预测未来某时期的发现量或累计发现量。同样，对油田开发有关信息，或其他专业的时间序列信息也可进行趋势预测（已知时间序列应当是非平稳的）。

### 1.2.2　算法与计算步骤

根据趋势的变化形态的不同，有两种计算方式。

（1）线性指数平滑法（近似以线性增长的时间序列）。

①计算 $i$ 时刻的单指数平滑值 $S_i'$：

$$S_i' = \alpha x_i + (1-\alpha) S_{i-1}'$$

式中　$x_i$——$i$ 时刻的观测值；

$\alpha$——平滑常数；

$S_{i-1}'$——$i-1$ 时刻单指数平滑值。

②计算 $i$ 时刻的双指数平滑值 $S_i''$：

$$S_i'' = \alpha\, S_i' + (1-\alpha) S_{i-1}''$$

③计算 $i$ 时刻的水平值 $A_i$：

$$A_i = S_i' + (S_i' - S_i'') = 2S_i' - S_i''$$

④计算 $i$ 时刻的增量 $B_i$:

$$B_i = \frac{\alpha}{1-\alpha}(S_i' - S_i'')$$

⑤计算 $i$ 时刻以后的 $m$ 个预测值。

$$F_i = A_i + mB_i$$

其中，$m$ 是大于等于 1 的整数。

注：线性指数平滑法在计算之前，要对 $S_1'$ 和 $S_1''$ 进行初始化，即：

$$S_1' = x_1 \qquad S_1'' = x_1$$

在进行预测时，前面各步计算中 $i$ 是从 2 开始。

⑥进行 $\chi^2$ 检验。

在计算出 $F_2,F_3,\cdots,F_n$ 后，同时得到误差 $e_i=x_2-F_2,\cdots,e_n=x_n-F_n$ 然后再计算误差序列 { $e_i$ } 的 $k$ 个自相关系数 $r_1,r_2,\cdots,r_k$。

$$k=\frac{n-1}{2}$$

式中，$k$ 等于 $\frac{n-1}{2}$ 的整数部分。

计算统计量 $Q_1$ 值：$Q_1=(n-1)\sum_{k=1}^{k} r_k^2$。以自由度为 $k-1$，置信度为 0.95 查 $\chi^2$ 表，得到 xx 值。当 $Q_1\leqslant$xx 时，预测结果较好；否则，预测结果只能参考。

（2）二次趋势指数平滑法（近似以二次趋势增长的序列）。

①计算 $i$ 时刻单指数平滑值 $p_i'$：

$$p_i'=\alpha x_i+(1-\alpha)p_{i-1}'$$

②计算 $i$ 时刻双指数平滑值 $p_i''$：

$$p_i''=\alpha p_i'+(1-\alpha)p_{i-1}''$$

③计算 $i$ 时刻三重指数平滑值 $p_i'''$：

$$p_i'''=\alpha p_i''+(1-\alpha)p_{i-1}'''$$

④计算 $i$ 时刻的水平增量 $A_i$：

$$A_i=3p_i'-3p_i''+p_i'''$$

⑤计算 $i$ 时刻的线性增量 $B_i$：

$$B_i=\frac{\alpha}{2(1-\alpha)^2}[(6-5\alpha)p_i'-(10-8\alpha)p_i''+(4-3\alpha)p_i''']$$

⑥计算 $i$ 时刻的二次增量 $C_i$：

$$C_i=\frac{\alpha^2}{(1-\alpha)^2}(p_i'-2p_i''+p_i''')$$

⑦用式（1-2-1）预测 $i$ 时刻以后的 $m$ 个 $F_{i+m}$ 值：

$$F_{i+m}=A_i+m\cdot B_i+\frac{1}{2}c_i\cdot m^2 \quad (m\geqslant 1) \tag{1-2-1}$$

注：在开始计算时对 $p_1'$、$p_1''$、$p_1'''$ 进行初始化，即：

$$p_1'=p_1''=p_1'''=x_1$$

⑧进行 $\chi^2$ 检验。

计算出 $F_2$，$F_3$，…，$F_n$ 和 $e_2$，$e_3$，…，$e_n$ 后,再求误差序列 { $e_i$ } 的自相关系数和统计量 $Q_2$ 值。最后判断 $Q_2$ 和查 $\chi^2$ 表值 xx 的关系，来决定预测结果是否可用。

说明1：此程序对同一时间序列两种趋势都进行计算，如果 $Q_1 \leqslant xx$，$Q_2 \leqslant xx$，表明这两种预测结果都可用，但是最后决定选取哪个预测结果，应由各自的均方差来决定，即选出 $SS_1$ 和 $SS_2$ 中最小者，哪个最小就用哪个预测式。

说明2：在两种趋势预测中，都含有平滑参数 $\alpha$，在程序中 $\alpha$ 可取为 0.1～0.9，即：$\alpha_j = 0.1j\ (j=1,2,\cdots,9)$，在每种趋势中，都算出 9 个均方差 $SS_1(j)$ 和 $SS_2(j)$，再选出它们的最小者，即：

$$SS_1=\min\{SS_1(j), j=1\sim 9\},\ SS_2=\min\{SS_2(j), j=1\sim 9\}$$

如果 $SS_1<SS_2$，则认为线性指数平滑法的预测结果较好；否则认为，二次趋势指数平滑法的结果较好。

说明3：程序中调用了求自相关系数子程序 zxgej 和查 $\chi^2$ 数据表子程序 xx95。

```
******a12 趋势分析法预测程序******
#include<fstream.h>
#include<math.h>
#include<string.h>
void main()
{  // a12.cpp 趋势分析法
   // 程序中调用子程序 (zxgej,xx95)
   // 输入文件 a12 的内容为:
   // n                    (观测值的个数)
   // x1,x2,x3,…,xn (n 个观测值)
   // 例如
   // 18
   // 11.32 11.77 12.05 12.50 13.00 13.44 14.01 14.50 15.01
   // 15.43 16.00 16.38 16.77 17.19 17.60 18.15 18.53 19.00
   cout<<"  趋势分析法=12"<<endl;
   cout<<endl;
   int zxgej(double[],double[],int);
   double xx95(int);
   double x[50],y[50],d[50],e[50],f[50];
   double r[50],s[50],t[50],g[10],h[10];
   double bb,s1,s2,a,b,c,at,q1,q2,xx,f1,f2;
   int i,j,k,m,n,n1,n2;
   char fname[16];
   char *txt=".txt";
   cout<<"->请输入参数文件名: ";
   cin>>fname;
   strcat(fname,txt);
   ifstream rf(fname);
```

```
rf>>n;
cout<<"    数据个数  n="<<n<<endl;
for(i=0;i<n;i++)
    rf>>x[i];
cout<<"    x[i]:"<<endl;
for(i=0;i<n;i++)
    cout<<"    "<<x[i];
cout<<endl;
cout<<endl;
bb=1.96/sqrt(1.0*n);
cout<<"    B=1.96/sqrt("<<n<<")="<<bb<<endl;
cout<<"    计算 x[i]的自相关系数"<<endl;
k=zxgej(x,r,n);
for(i=0;i<k-1;i++)
    cout<<"     r["<<i<<"]="<<r[i]<<endl;
cout<<endl;
for(i=0;i<50;i++)
{
    y[i]=0; d[i]=0;
}
n2=n/2;
cout<<"    一次型: "<<endl;
at=0.1;
for(i=0;i<10;i++)
    h[i]=at+i*0.1;
for(i=0;i<10;i++)
{
    at=h[i];
    y[0]=x[0];y[1]=x[1];
    f[0]=x[0];t[0]=x[0];
    for(j=1;j<n;j++)
    {
        f[j]=at*x[j]+(1-at) *f[j-1];
        t[j]=at*f[j]+(1-at) *t[j-1];
        a=2*f[j]-t[j];
        b=(f[j]-t[j]) *at/(1-at);
        y[j+1]=a+b;
    }
    s1=0;
    for(j=0;j<n;j++)
```