FORTRAN 常用算法程序集

（第二版）

徐士良　编著

清华大学出版社

目　录

第1章 线性代数方程组的求解

1.1 全选主元高斯消去法

一、功能

用全选主元高斯(Gauss)消去法求解线性代数方程组 $AX=B$。

二、方法说明

高斯消去法分两步进行。

第一步 消去过程

在这一过程中，为了保证数值计算的稳定性，本子程序采用了全选主元。

对于 $k=1,2,\cdots,n-1$，作以下三步：

(1) 全选主元。即从系数矩阵的第 k 行、第 k 列开始的右下子阵中选取绝对值最大的元素，并将它交换到主元素的位置上。

(2) 归一化。即

$$a_{kj}/a_{kk}\Rightarrow a_{kj},\quad j=k+1,\cdots,n$$
$$b_k/a_{kk}\Rightarrow b_k$$

(3) 消去。即

$$a_{ij}-a_{ik}a_{kj}\Rightarrow a_{ij},\quad i,j=k+1,\cdots,n$$
$$b_i-a_{ik}b_k\Rightarrow b_i,\quad i=k+1,\cdots,n$$

第二步 回代过程

(1) $$b_n/a_{nn}\Rightarrow x_n$$

(2) $$b_i-\sum_{j=i+1}^{n}a_{ij}x_j\Rightarrow x_i,\quad i=n-1,\cdots,2,1$$

三、子程序语句

SUBROUTINE AGAUS(A,B,N,X,L,JS)

四、形参说明

A——双精度实型二维数组，体积为N×N，输入参数。存放方程组的系数矩阵，返回时将被破坏。

B——双精度实型一维数组，长度为N，输入参数。存放方程组右端向量，返回时将被破坏。

N——整型变量，输入参数。存放方程组的阶数。

X——双精度实型一维数组，长度为N，输出参数。返回方程组的解向量。

L——整型变量，输出参数。若返回L=0，说明方程组的系数矩阵奇异，求解失败；若L≠0，说明正常返回。

JS——整型一维数组，长度为N。本子程序的工作数组。

五、子程序(文件名:AGAUS.FOR)

```
      SUBROUTINE AGAUS(A,B,N,X,L,JS)
      DIMENSION A(N,N),X(N),B(N),JS(N)
      DOUBLE PRECISION A,B,X,T
      L=1
      DO 50 K=1,N-1
        D=0.0
        DO 210 I=K,N
        DO 210 J=K,N
          IF (ABS(A(I,J)).GT.D) THEN
            D=ABS(A(I,J))
            JS(K)=J
            IS=I
          END IF
210     CONTINUE
        IF (D+1.0.EQ.1.0) THEN
          L=0
        ELSE
          IF (JS(K).NE.K) THEN
            DO 220 I=1,N
              T=A(I,K)
              A(I,K)=A(I,JS(K))
              A(I,JS(K))=T
220         CONTINUE
          END IF
          IF (IS.NE.K) THEN
            DO 230 J=K,N
              T=A(K,J)
              A(K,J)=A(IS,J)
              A(IS,J)=T
230         CONTINUE
            T=B(K)
            B(K)=B(IS)
            B(IS)=T
          END IF
        END IF
        IF (L.EQ.0) THEN
          WRITE(*,100)
          RETURN
        END IF
```

```
      DO 10 J=K+1,N
        A(K,J)=A(K,J)/A(K,K)
10    CONTINUE
      B(K)=B(K)/A(K,K)
      DO 30 I=K+1,N
        DO 20 J=K+1,N
          A(I,J)=A(I,J)-A(I,K)*A(K,J)
20      CONTINUE
        B(I)=B(I)-A(I,K)*B(K)
30    CONTINUE
50    CONTINUE
      IF (ABS(A(N,N))+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,100)
        RETURN
      END IF
      X(N)=B(N)/A(N,N)
      DO 70 I=N-1,1,-1
        T=0.0
        DO 60 J=I+1,N
          T=T+A(I,J)*X(J)
60      CONTINUE
        X(I)=B(I)-T
70    CONTINUE
100   FORMAT(1X,' FAIL ')
      JS(N)=N
      DO 150 K=N,1,-1
        IF (JS(K).NE.K) THEN
          T=X(K)
          X(K)=X(JS(K))
          X(JS(K))=T
        END IF
150   CONTINUE
      RETURN
      END
```

六、例

求解下列方程组

$$\begin{cases} 0.2368x_1+0.2471x_2+0.2568x_3+1.2671x_4=1.8471 \\ 0.1968x_1+0.2071x_2+1.2168x_3+0.2271x_4=1.7471 \\ 0.1582x_1+1.1675x_2+0.1768x_3+0.1871x_4=1.6471 \\ 1.1161x_1+0.1254x_2+0.1397x_3+0.1490x_4=1.5471 \end{cases}$$

主程序(文件名:AGAUS0.FOR)为

```
      DIMENSION A(4,4),B(4),X(4),JS(4)
      DOUBLE PRECISION A,B,X
      DATA A/0.2368,0.1968,0.1582,1.1161,0.2471,0.2071,1.1675,0.1254
     *      ,0.2568,1.2168,0.1768,0.1397,1.2671,0.2271,0.1871,0.1490/
      DATA B/1.8471,1.7471,1.6471,1.5471/
      N=4
      CALL AGAUS(A,B,N,X,L,JS)
      IF (L.NE.0) THEN
        WRITE(*,10) (I,X(I),I=1,4)
      END IF
10    FORMAT(1X,'X(',I2,')=',D15.6)
      END
```

运行结果为

```
X( 1 )=    .104058D+01
X( 2 )=    .986956D+00
X( 3 )=    .935053D+00
X( 4 )=    .881297D+00
```

1.2　全选主元高斯-约当消去法

一、功能

用全选主元高斯-约当(Gauss-Jordan)消去法同时求解系数矩阵相同而常数向量不同的多组线性代数方程组　$AX=B$。

二、方法说明

设线性代数方程组为

$$AX=B$$

其中

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

$$B=\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix}$$

全选主元高斯-约当消去法的计算过程如下。

对于 $k=1,2,\cdots,n$ 作如下变换:

(1) 全选主元

从第 k 行、第 k 列开始的右下角子阵中选取绝对值最大的元素，并通过行、列交换将它交换到主元素的位置上。

(2) 归一化

$$a_{kj}/a_{kk} \Rightarrow a_{kj}, \quad j=k+1,\cdots,n$$
$$b_{kj}/a_{kk} \Rightarrow b_{kj}, \quad j=1,2,\cdots,m$$

(3) 消去

$$a_{ij}-a_{ik}a_{kj} \Rightarrow a_{ij}, \quad j=k+1,\cdots,n$$
$$i=1,2,\cdots,n;i\neq k$$
$$b_{ij}-a_{ik}b_{kj} \Rightarrow b_{ij}, \quad j=1,2,\cdots,m$$
$$i=1,2,\cdots,n;i\neq k$$

最后对 B 进行恢复，则 B 中的每一列即为每一个方程组的解。

三、子程序语句

SUBROUTINE AGJDN(A,B,N,M,L,JS)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入参数。存放方程组的系数矩阵，返回时将被破坏。

B——双精度实型二维数组，体积为 N×M，输入兼输出参数。调用时存放 M 组常数向量；返回 M 组解向量。

N——整型变量，输入参数。方程组阶数。

M——整型变量，输入参数。方程组右端常数向量的组数。

L——整型变量，输出参数。若返回 L=0，说明方程组系数矩阵奇异，求解失败；若 L≠0，表示正常返回。

JS——整型一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名：AGJDN.FOR)

```
      SUBROUTINE AGJDN(A,B,N,M,L,JS)
      DIMENSION A(N,N),B(N,M),JS(N)
      DOUBLE PRECISION A,B,D
      L=1
      DO 100 K=1,N
        Q=0.0
        DO 10 I=K,N
        DO 10 J=K,N
          IF (ABS(A(I,J)).GT.Q) THEN
            Q=ABS(A(I,J))
            JS(K)=J
            IS=I
          END IF
10      CONTINUE
        IF (Q+1.0.EQ.1.0) THEN
```

```
          WRITE(*,20)
          L=0
          RETURN
        END IF
20      FORMAT(1X,' FAIL ')
        DO 30 J=K,N
          D=A(K,J)
          A(K,J)=A(IS,J)
          A(IS,J)=D
30      CONTINUE
        DO 40 J=1,M
          D=B(K,J)
          B(K,J)=B(IS,J)
          B(IS,J)=D
40      CONTINUE
        DO 50 I=1,N
          D=A(I,K)
          A(I,K)=A(I,JS(K))
          A(I,JS(K))=D
50      CONTINUE
        DO 60 J=K+1,N
60      A(K,J)=A(K,J)/A(K,K)
        DO 70 J=1,M
70      B(K,J)=B(K,J)/A(K,K)
        DO 90 I=1,N
          IF (I.NE.K) THEN
            DO 80 J=K+1,N
80          A(I,J)=A(I,J)-A(I,K)*A(K,J)
            DO 85 J=1,M
85          B(I,J)=B(I,J)-A(I,K)*B(K,J)
          END IF
90      CONTINUE
100   CONTINUE
      DO 110 K=N,1,-1
      DO 110 J=1,M
        D=B(K,J)
        B(K,J)=B(JS(K),J)
        B(JS(K),J)=D
110   CONTINUE
      RETURN
      END
```

六、例

求解下列方程组

$$AX=B$$

其中

$$A=\begin{bmatrix}1 & 3 & 2 & 13\\ 7 & 2 & 1 & -2\\ 9 & 15 & 3 & -2\\ -2 & -2 & 11 & 5\end{bmatrix},\quad B=\begin{bmatrix}9 & 0\\ 6 & 4\\ 11 & 7\\ -2 & -1\end{bmatrix}$$

主程序(文件名:AGJDN0.FOR)为

```
      DIMENSION A(4,4),B(4,2),JS(4)
      DOUBLE PRECISION A,B
      DATA A/1.0,7.0,9.0,-2.0,3.0,2.0,15.0,-2.0,2.0,1.0,3.,11.,13.,
     *        -2.0,-2.0,5.0/
      DATA B/9.0,6.0,11.0,-2.0,0.0,4.0,7.0,-1.0/
      N=4
      M=2
      CALL AGJDN(A,B,N,M,L,JS)
      IF (L.NE.0) THEN
        WRITE(*,10) ((B(I,J),I=1,4),J=1,2)
      END IF
10    FORMAT(1X,4D15.6)
      END
```

运行结果为

```
.980745D+00    .267982D+00    -.222629D+00    .589274D+00
.497931D+00    .144494D+00    .628581D-01    -.813176D-01
```

1.3 复系数方程组的全选主元高斯消去法

一、功能

用全选主元高斯(Gauss)消去法求解复系数线性代数方程组 $AX=B$. 其中

$$A=AR+jAI$$
$$B=BR+jBI$$
$$X=XR+jXI$$

二、方法说明

基本方法同 1.1 节。

计算两个复数的乘积

$$e+jf=(a+jb)(c+jd)$$

本子程序采用如下算法:

$$p=ac,q=bd,s=(a+b)(c+d)$$

$$e=p-g, f=s-p-q$$

计算两个复数的商

$$e+jf=(c+jd)/(a+jb)$$

本子程序采用如下算法：

$$p=ac, q=-bd, s=(a-b)(c+d), w=a^2+b^2$$

$$e=(p-q)/w, f=(s-p-q)/w$$

三、子程序语句

SUBROUTINE ACGAS(AR,AI,N,BR,BI,L,JS)

四、形参说明

AR,AI——均为双精度实型二维数组，体积为N×N，输入参数。分别存放方程组系数矩阵的实部与虚部，返回时将被破坏。

N——整型变量，输入参数。方程组阶数。

BR,BI——均为双精度实型一维数组，长度为N，输入兼输出参数。调用时分别存放方程组右端常数向量的实部与虚部；分别返回方程组解向量的实部与虚部。

L——整型变量，输出参数。若返回L=0，说明方程组系数矩阵奇异，求解失败；若L≠0，表示正常返回。

JS——整型一维数组，长度为N。本子程序的工作数组。

五、子程序(文件名:ACGAS.FOR)

```
      SUBROUTINE ACGAS(AR,AI,N,BR,BI,L,JS)
      DIMENSION AR(N,N),AI(N,N),BR(N),BI(N),JS(N)
      DOUBLE PRECISION AR,AI,BR,BI,D,P,Q,S
      L=1
      DO 100 K=1,N-1
        D=0.0
        DO 10 I=K,N
        DO 10 J=K,N
          P=AR(I,J)*AR(I,J)+AI(I,J)*AI(I,J)
          IF (P.GT.D) THEN
            D=P
            JS(K)=J
            IS=I
          END IF
10      CONTINUE
        W=D
        IF (W+1.0.EQ.1.0) THEN
          WRITE(*,20)
          L=0
          RETURN
        END IF
20      FORMAT(1X,' ERR**FAIL ')
```

```
      DO 30 J=K,N
        P=AR(K,J)
        AR(K,J)=AR(IS,J)
        AR(IS,J)=P
        P=AI(K,J)
        AI(K,J)=AI(IS,J)
        AI(IS,J)=P
30    CONTINUE
      P=BR(K)
      BR(K)=BR(IS)
      BR(IS)=P
      P=BI(K)
      BI(K)=BI(IS)
      BI(IS)=P
      DO 50 I=1,N
        P=AR(I,K)
        AR(I,K)=AR(I,JS(K))
        AR(I,JS(K))=P
        P=AI(I,K)
        AI(I,K)=AI(I,JS(K))
        AI(I,JS(K))=P
50    CONTINUE
      DO 60 J=K+1,N
        P=AR(K,J)*AR(K,K)
        Q=-AI(K,J)*AI(K,K)
        S=(AR(K,K)-AI(K,K))*(AR(K,J)+AI(K,J))
        AR(K,J)=(P-Q)/D
        AI(K,J)=(S-P-Q)/D
60    CONTINUE
      P=BR(K)*AR(K,K)
      Q=-BI(K)*AI(K,K)
      S=(AR(K,K)-AI(K,K))*(BR(K)+BI(K))
      BR(K)=(P-Q)/D
      BI(K)=(S-P-Q)/D
      DO 90 I=K+1,N
        DO 80 J=K+1,N
          P=AR(I,K)*AR(K,J)
          Q=AI(I,K)*AI(K,J)
          S=(AR(I,K)+AI(I,K))*(AR(K,J)+AI(K,J))
          AR(I,J)=AR(I,J)-P+Q
          AI(I,J)=AI(I,J)-S+P+Q
80      CONTINUE
```

```
          P=AR(I,K)*BR(K)
          Q=AI(I,K)*BI(K)
          S=(AR(I,K)+AI(I,K))*(BR(K)+BI(K))
          BR(I)=BR(I)-P+Q
          BI(I)=BI(I)-S+P+Q
90      CONTINUE
100   CONTINUE
      D=AR(N,N)*AR(N,N)+AI(N,N)*AI(N,N)
      W=D
      IF (W+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,20)
        RETURN
      END IF
      P=AR(N,N)*BR(N)
      Q=-AI(N,N)*BI(N)
      S=(AR(N,N)-AI(N,N))*(BR(N)+BI(N))
      BR(N)=(P-Q)/D
      BI(N)=(S-P-Q)/D
      DO 200 I=N-1,1,-1
        DO 150 J=I+1,N
          P=AR(I,J)*BR(J)
          Q=AI(I,J)*BI(J)
          S=(AR(I,J)+AI(I,J))*(BR(J)+BI(J))
          BR(I)=BR(I)-P+Q
          BI(I)=BI(I)-S+P+Q
150     CONTINUE
200   CONTINUE
      JS(N)=N
      DO 110 K=N,1,-1
        P=BR(K)
        BR(K)=BR(JS(K))
        BR(JS(K))=P
        P=BI(K)
        BI(K)=BI(JS(K))
        BI(JS(K))=P
110   CONTINUE
      RETURN
      END
```

六、例

求解下列复系数方程组

$$AX=B$$

其中 $A=AR+jAI, B=BR+jBI$

$$AR=\begin{bmatrix}1 & 3 & 2 & 13\\ 7 & 2 & 1 & -2\\ 9 & 15 & 3 & -2\\ -2 & -2 & 11 & 5\end{bmatrix},\quad AI=\begin{bmatrix}3 & -2 & 1 & 6\\ -2 & 7 & 5 & 8\\ 9 & -3 & 15 & 1\\ -2 & -2 & 7 & 6\end{bmatrix}$$

$$BR=(2,7,3,9)^T,\ BI=(1,2,-2,3)^T$$

主程序(文件名:ACGAS0.FOR)为

```
      DIMENSION AR(4,4),AI(4,4),BR(4),BI(4),JS(4)
      DOUBLE PRECISION AR,AI,BR,BI
      DATA AR/1.0,7.0,9.0,-2.0,3.0,2.0,15.0,-2.0,
     *        2.0,1.0,3.0,11.0,13.0,-2.0,-2.0,5.0/
      DATA AI/3.0,-2.0,9.0,-2.0,-2.0,7.0,-3.0,-2.0,
     *        1.0,5.0,15.0,7.0,6.0,8.0,1.0,6.0/
      DATA BR/2.0,7.0,3.0,9.0/
      DATA BI/1.0,2.0,-2.0,3.0/
      CALL ACGAS(AR,AI,4,BR,BI,L,JS)
      IF (L.NE.0) THEN
        WRITE(*,10) (BR(I),I=1,4)
        WRITE(*,10) (BI(I),I=1,4)
      END IF
10    FORMAT(1X,4D14.6)
      END
```

运行结果为

```
.678233D-01   -.162341D+00   .598524D+00   .246456D+00
.707823D-01   -.761294D+00   -.437131D+00  .113996D+00
```

1.4 复系数方程组的全选主元高斯-约当消去法

一、功能

用全选主元高斯-约当(Gauss-Jordan)消去法求解系数矩阵相同而具有多组右端常数向量的复系数线性代数方程组 $AX=B$。

二、方法说明

基本方法同1.2节。

为了计算两个复数的乘积

$$e+jf=(a+jb)(c+jd)$$

本子程序采用如下算法:

$$p=ac, q=bd, s=(a+b)(c+d)$$

$$e=p-q, f=s-p-q$$

为了计算两个复数的商

$$e+jf=(c+jd)/(a+jb)$$

本子程序采用如下算法：

$$p=ac,q=-bd,s=(a-b)(c+d),w=a^2+b^2$$

$$e=(p-q)/w,f=(s-p-q)/w$$

三、子程序语句

SUBROUTINE ACJDN(AR,AI,N,BR,BI,M,L,JS)

四、形参说明

AR,AI——均为双精度实型二维数组，体积为N×N，输入参数。分别存放方程组系数矩阵的实部与虚部，返回时将被破坏。

N——整型变量，输入参数。方程组阶数。

BR,BI——均为双精度实型二维数组，体积为N×M，输入兼输出参数。调用时分别存放M组右端常数向量的实部与虚部；返回时分别存放M组解向量的实部与虚部。

M——整型变量，输入参数。方程组右端常数向量的组数。

L——整型变量，输出参数。若返回L=0，说明方程组的系数矩阵奇异，求解失败；若L≠0，表示正常返回。

JS——整型一维数组，长度为N。本子程序的工作数组。

五、子程序(文件名：ACJDN.FOR)

```
      SUBROUTINE ACJDN(AR,AI,N,BR,BI,M,L,JS)
      DIMENSION AR(N,N),AI(N,N),BR(N,M),BI(N,M),JS(N)
      DOUBLE PRECISION AR,AI,BR,BI,D,P,Q,S
      L=1
      DO 100 K=1,N
        D=0.0
        DO 10 I=K,N
        DO 10 J=K,N
          P=AR(I,J)*AR(I,J)+AI(I,J)*AI(I,J)
          IF (P.GT.D) THEN
            D=P
            JS(K)=J
            IS=I
          END IF
10      CONTINUE
        W=D
        IF (W+1.0.EQ.1.0) THEN
          WRITE(*,20)
          L=0
          RETURN
        END IF
20      FORMAT(1X,'ERR**FAIL')
```

```
      DO 30 J=K,N
        P=AR(K,J)
        AR(K,J)=AR(IS,J)
        AR(IS,J)=P
        P=AI(K,J)
        AI(K,J)=AI(IS,J)
        AI(IS,J)=P
30    CONTINUE
      DO 35 J=1,M
        P=BR(K,J)
        BR(K,J)=BR(IS,J)
        BR(IS,J)=P
        P=BI(K,J)
        BI(K,J)=BI(IS,J)
        BI(IS,J)=P
35    CONTINUE
      DO 50 I=1,N
        P=AR(I,K)
        AR(I,K)=AR(I,JS(K))
        AR(I,JS(K))=P
        P=AI(I,K)
        AI(I,K)=AI(I,JS(K))
        AI(I,JS(K))=P
50    CONTINUE
      DO 60 J=K+1,N
        P=AR(K,J)*AR(K,K)
        Q=-AI(K,J)*AI(K,K)
        S=(AR(K,K)-AI(K,K))*(AR(K,J)+AI(K,J))
        AR(K,J)=(P-Q)/D
        AI(K,J)=(S-P-Q)/D
60    CONTINUE
      DO 65 J=1,M
        P=BR(K,J)*AR(K,K)
        Q=-BI(K,J)*AI(K,K)
        S=(AR(K,K)-AI(K,K))*(BR(K,J)+BI(K,J))
        BR(K,J)=(P-Q)/D
        BI(K,J)=(S-P-Q)/D
65    CONTINUE
      DO 90 I=1,N
        IF (I.NE.K) THEN
          DO 80 J=K+1,N
            P=AR(I,K)*AR(K,J)
```

```
          Q=AI(I,K)*AI(K,J)
          S=(AR(I,K)+AI(I,K))*(AR(K,J)+AI(K,J))
          AR(I,J)=AR(I,J)-P+Q
          AI(I,J)=AI(I,J)-S+P+Q
80      CONTINUE
        DO 85 J=1,M
          P=AR(I,K)*BR(K,J)
          Q=AI(I,K)*BI(K,J)
          S=(AR(I,K)+AI(I,K))*(BR(K,J)+BI(K,J))
          BR(I,J)=BR(I,J)-P+Q
          BI(I,J)=BI(I,J)-S+P+Q
85      CONTINUE
       END IF
90    CONTINUE
100   CONTINUE
      DO 110 K=N,1,-1
      DO 110 J=1,M
        P=BR(K,J)
        BR(K,J)=BR(JS(K),J)
        BR(JS(K),J)=P
        P=BI(K,J)
        BI(K,J)=BI(JS(K),J)
        BI(JS(K),J)=P
110   CONTINUE
      RETURN
      END
```

六、例

求解复系数方程组 $AX=B$，其中

$$A=AR+\mathrm{j}AI, B=BR+\mathrm{j}BI$$

$$AR=\begin{bmatrix}1 & 3 & 2 & 13\\ 7 & 2 & 1 & -2\\ 9 & 15 & 3 & -2\\ -2 & -2 & 11 & 5\end{bmatrix},\quad AI=\begin{bmatrix}3 & -2 & 1 & 6\\ -2 & 7 & 5 & 8\\ 9 & -3 & 15 & 1\\ -2 & -2 & 7 & 6\end{bmatrix}$$

$$BR=\begin{bmatrix}2 & 1\\ 7 & 2\\ 3 & -2\\ 9 & 3\end{bmatrix},\quad BI=\begin{bmatrix}1 & 2\\ 2 & 7\\ -2 & 3\\ 3 & 9\end{bmatrix}$$

主程序(文件名:ACJDN0.FOR)为

```
      DIMENSION AR(4,4),AI(4,4),BR(4,2),BI(4,2),JS(4)
      DOUBLE PRECISION AR,AI,BR,BI
      DATA AR/1.0,7.0,9.0,-2.0,3.0,2.0,15.0,-2.0,
```

```
     *          2.0,1.0,3.0,11.0,13.0,-2.0,-2.0,5.0/
        DATA AI/3.0,-2.0,9.0,-2.0,-2.0,7.0,-3.0,-2.0,
     *          1.0,5.0,15.0,7.0,6.0,8.0,1.0,6.0/
        DATA BR/2.0,7.0,3.0,9.0,1.0,2.0,-2.0,3.0/
        DATA BI/1.0,2.0,-2.0,3.0,2.0,7.0,3.0,9.0/
        CALL ACJDN(AR,AI,4,BR,BI,2,L,IS)
        WRITE(*,*)
        IF (L.NE.0) THEN
          DO 5 J=1,2
            WRITE(*,10) (BR(I,J),I=1,4)
            WRITE(*,10) (BI(I,J),I=1,4)
            WRITE(*,*)
5         CONTINUE
        END IF
10      FORMAT(1X,4D14.6)
        END
```

运行结果为

```
 .678233D-01  -.162341D+00   .598524D+00   .246456D+00
 .707823D-01  -.761294D+00  -.437131D+00   .113996D+00

-.252514D-01   .330854D+00   .593462D+00   .134401D+00
 .227075D+00  -.497636D+00   .303802D+00   .153525D+00
```

1.5 三对角线方程组的追赶法

一、功能

用追赶法求解三对角线方程组 $AX=D$。其中

$$A=\begin{bmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ & a_{32} & a_{33} & a_{34} & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ & & & & a_{n,n-1} & a_{nn} \end{bmatrix}$$

$$X=(x_1,x_2,\cdots,x_n)^T,\ D=(d_1,d_2,\cdots,d_n)^T$$

二、方法说明

追赶法本质上是没有选主元的高斯消去法，只是在计算过程中考虑到了三对角线矩阵的特点，对于绝大部分的零元素不再作处理。其计算步骤为：

(1) 对于 $k=1,2,\cdots,n-1$ 作归一化及消元处理。即

$$a_{k,k+1}/a_{kk}\Rightarrow a_{k,k+1},\ d_k/a_{kk}\Rightarrow d_k$$

$$a_{k+1,k+1}-a_{k+1,k}a_{k,k+1}\Rightarrow a_{k+1,k+1}$$

$$d_{k+1}-a_{k+1,k}d_k \Rightarrow d_{k+1}$$

(2) 进行回代。即

$$d_n/a_{nn} \Rightarrow x_n$$

$$d_k-a_{k,k+1}x_{k+1} \Rightarrow x_k, k=n-1,\cdots,2,1$$

在本子程序中，为了便于三对角矩阵的输入，用一个实型一维数组 $B(1:3n-2)$ 以行为主存放三对角矩阵 A 中三条对角线上的元素。其中 n 为矩阵 A 的阶数。三对角线矩阵 A 与一维数组 B 之间有如下关系：

$$A(i,j)=\begin{cases}B[2(i-1)+j], & i-1\leqslant j\leqslant i+1\\ 0, & \text{其它}\end{cases}$$

根据以上关系，可以得到如下计算过程：

(1) 对于 $k=1,2,\cdots,n-1$ 作归一化及消元处理。即

$$B(3k-1)/B(3k-2) \Rightarrow B(3k-1)$$

$$d_k/B(3k-2) \Rightarrow d_k$$

$$B(3k+1)-B(3k)B(3k-1) \Rightarrow B(3k+1)$$

$$d_{k+1}-B(3k)d_k \Rightarrow d_{k+1}$$

(2) 进行回代。即

$$d_n/B(3n-2) \Rightarrow d_n$$

$$d_k-B(3k-1)d_{k+1} \Rightarrow d_k, k=n-1,\cdots,2,1$$

最后的解向量在 D 中。

特别要指出，由于追赶法本质上是没有选主元的高斯消去法，因此其计算过程是不稳定的。但当三对角矩阵满足下列条件：

$$|a_{11}|>|a_{12}|$$

$$|a_{ii}|\geqslant|a_{i,i-1}|+|a_{i,i+1}|, i=2,3,\cdots,n-1$$

$$|a_{nn}|>|a_{n,n-1}|$$

时，追赶法的计算过程中不会出现中间结果数量级的巨大增长和舍入误差的严重积累。

三、子程序语句

SUBROUTINE ATRDE(B,N,M,D,L)

四、形参说明

B——双精度实型一维数组，长度为 M=3＊N－2，输入参数。以行为主存放三对角矩阵 A 中三条对角线上的元素，即在 B 中依次存放元素：

$$a_{11},a_{12},a_{21},a_{22},a_{23},a_{32},a_{33},a_{34},\cdots,a_{n,n-1},a_{nn}$$

返回时该数组将被破坏。

N——整型变量，输入参数，方程组阶数。

M——整型变量，输入参数。M=3＊N－2 为三对角矩阵三条对角线上的元素个数。

D——双精度实型一维数组，长度为 N，输入兼输出参数。调用时存放方程组右端的常数向量；返回方程组的解向量。

L——整型变量，输出参数。若返回 L<0，说明 M 的值不正确（应为 M=3＊N－2）；若 L=0，说明程序工作失败；若 L>0，表示正常返回。

五、子程序(文件名:ATRDE.FOR)

```
      SUBROUTINE ATRDE(B,N,M,D,L)
      DIMENSION B(M),D(N)
      DOUBLE PRECISION B,D
      L=1
      IF (M.NE.(3*N-2)) THEN
        L=-1
        WRITE(*,10)
        RETURN
      END IF
10    FORMAT(1X,' ERR ')
      DO 20 K=1,N-1
        J=3*K-2
        IF (ABS(B(J))+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,10)
          RETURN
        END IF
        B(J+1)=B(J+1)/B(J)
        D(K)=D(K)/B(J)
        B(J+3)=B(J+3)-B(J+2)*B(J+1)
        D(K+1)=D(K+1)-B(J+2)*D(K)
20    CONTINUE
      IF (ABS(B(3*N-2))+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,10)
        RETURN
      END IF
      D(N)=D(N)/B(3*N-2)
      DO 30 K=N-1,1,-1
        D(K)=D(K)-B(3*K-1)*D(K+1)
30    CONTINUE
      RETURN
      END
```

六、例

求解三对角线方程组

$$
\begin{bmatrix} 13 & 12 & & & \\ 11 & 10 & 9 & & \\ & 8 & 7 & 6 & \\ & & 5 & 4 & 3 \\ & & & 2 & 1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} 3 \\ 0 \\ -2 \\ 6 \\ 8 \end{bmatrix}
$$

其中 N=5,M=3×5−2=13,

主程序(文件名:ATRDE0.FOR)为

```
      DIMENSION B(13),D(5)
      DOUBLE PRECISION B,D
      DATA B/13.0,12.0,11.0,10.0,9.0,8.0,
     *        7.0,6.0,5.0,4.0,3.0,2.0,1.0/
      DATA D/3.0,0.0,-2.0,6.0,8.0/
      CALL ATRDE(B,5,13,D,L)
      IF (L.GT.0) THEN
        WRITE(*,10) (D(I),I=1,5)
      END IF
 10   FORMAT(1X,D15.6)
      END
```

运行结果为

```
 .571837D+01
-.594490D+01
-.383673D+00
 .804082D+01
-.808163D+01
```

1.6 一般带型方程组

一、功能

用列选主元高斯(Gauss)消去法求解右端具有多组常向量的一般带型方程组 $AX=D$。其中 A 为 n 阶带型矩阵,即 A 中元素满足

$$a_{ij}\begin{cases}\neq 0, & i-l\leqslant j\leqslant i+l\\ =0, & \text{其它}\end{cases}$$

l 称为半带宽,$il=2l+1$ 称为带宽。D 为 m 列矩阵,每列表示一组常向量。

二、方法说明

设带型方程组 $AX=D$ 的系数矩阵为

$$A=\begin{array}{c} \\ l+1\left\{\begin{array}{c}\\ \\ \\ \end{array}\right. \\ \\ \\ \\ \\ \end{array}\begin{bmatrix} \overbrace{a_{11} & \cdots & a_{1,l+1}}^{l+1} & & & & \\ \vdots & \ddots & \vdots & \ddots & & & \\ a_{l+1,1} & \cdots & a_{l+1,l+1} & \cdots & a_{l+1,n} & & \\ & \ddots & & \ddots & & \ddots & \\ & & a_{n-1,n-2l} & \cdots & a_{n-1,n-1} & \cdots & a_{n-1,n} \\ & & & \ddots & \vdots & \ddots & \vdots \\ & & & & a_{n,n-l} & \cdots & a_{nn} \end{bmatrix}$$

m 组常向量 D 为

$$
D=\begin{bmatrix} d_{11} & \cdots & d_{1m} \\ d_{21} & \cdots & d_{2m} \\ \vdots & & \vdots \\ d_{n1} & \cdots & d_{nm} \end{bmatrix}
$$

由于系数矩阵呈带状，带区外均为零元素，因此只需存储带区内元素，可以用一个 n 行、il 列的矩形数组 B 表示。其表示格式如下：

$$
B=\begin{bmatrix} b_{11} & \cdots & b_{1,l+1} & \cdots & b_{1,il} \\ \vdots & & \vdots & & \vdots \\ b_{l+1,1} & \cdots & b_{l+1,l+1} & \cdots & b_{l+1,il} \\ \vdots & & \vdots & & \vdots \\ b_{n-l,1} & \cdots & b_{n-l,l+1} & \cdots & b_{n-l,il} \\ \vdots & & \vdots & & \vdots \\ b_{n1} & \cdots & b_{n,l+1} & \cdots & b_{n,il} \end{bmatrix}
$$

$$
=\begin{bmatrix} a_{11} & \cdots & a_{1,l+1} & & \\ & & & & 0 \\ \vdots & \ddots & \vdots & \ddots & \\ a_{l+1,1} & \cdots & a_{l+1,l+1} & \cdots & a_{l+1,2l+1} \\ \vdots & & \vdots & & \vdots \\ a_{n-l,n-2l} & \cdots & a_{n-l,n-l} & \cdots & a_{n-l,n} \\ \vdots & \iddots & & \iddots & \\ & & & & 0 \\ a_{n,n-l} & \cdots & a_{nn} & & \end{bmatrix}
$$

本子程序采用列选主元高斯消去法。

三、子程序语句

SUBROUTINE　ABAND(B,D,N,L,IL,M,IT)

四、形参说明

B ——双精度实型二维数组，体积为 N×IL，输入参数。存放带型矩阵 A 中带区内的元素，存放格式见方法说明，该数组在返回时被破坏。

D ——双精度实型二维数组，体积为 N×M，输入兼输出参数。调用时存放方程组右端的 M 组常数向量；返回 M 组解向量。

N ——整型变量，输入参数。方程组的阶数。

L——整型变量，输入参数。带型矩阵 A 的半带宽。

IL——整型变量，输入参数。存放带型矩阵 A 的带宽，要求 IL＝2＊L＋1。

M ——整型变量，输入参数。方程组右端常数向量的组数。

IT——整型变量，输出参数。若返回 IT＜0，说明输入参数 IL 与 L 的关系不对；若 IT＝0，说明系数矩阵 A 奇异，求解失败；若 IT＞0，表示正常返回。

五、子程序(文件名：ABAND.FOR)

```
      SUBROUTINE ABAND(B,D,N,L,IL,M,IT)
      DIMENSION B(N,IL),D(N,M)
```

```
      DOUBLE PRECISION B,D,T
      IT=1
      IF (IL.NE.2*L+1) THEN
        IT=-1
        WRITE(*,20)
        RETURN
      END IF
      LS=L+1
      DO 100 K=1,N-1
        P=0.0
        DO 10 I=K,LS
          IF (ABS(B(I,1)).GT.P) THEN
            P=ABS(B(I,1))
            IS=I
          END IF
10      CONTINUE
        IF (P+1.0.EQ.1.0) THEN
          IT=0
          WRITE(*,20)
          RETURN
        END IF
20      FORMAT(1X,'* * *FAIL* * *')
        DO 30 J=1,M
          T=D(K,J)
          D(K,J)=D(IS,J)
          D(IS,J)=T
30      CONTINUE
        DO 40 J=1,IL
          T=B(K,J)
          B(K,J)=B(IS,J)
          B(IS,J)=T
40      CONTINUE
        DO 50 J=1,M
50      D(K,J)=D(K,J)/B(K,1)
        DO 60 J=2,IL
60      B(K,J)=B(K,J)/B(K,1)
        DO 90 I=K+1,LS
          T=B(I,1)
          DO 70 J=1,M
70        D(I,J)=D(I,J)-T*D(K,J)
          DO 80 J=2,IL
80        B(I,J-1)=B(I,J)-T*B(K,J)
```

```
          B(I,IL)=0.0
90      CONTINUE
        IF (LS.NE.N) LS=LS+1
100   CONTINUE
      IF (ABS(B(N,1))+1.0.EQ.1.0) THEN
        IT=0
        WRITE(*,20)
        RETURN
      END IF
      DO 110 J=1,M
110   D(N,J)=D(N,J)/B(N,1)
      JS=2
      DO 150 I=N-1,1,-1
        DO 120 K=1,M
        DO 120 J=2,JS
120     D(I,K)=D(I,K)-B(I,J)*D(I+J-1,K)
        IF (JS.NE.IL) JS=JS+1
150   CONTINUE
      RETURN
      END
```

六、例

设五对角线方程组为

$$AX = D$$

其中

$$A = \begin{bmatrix} 3 & -4 & 1 & & & & & \\ -2 & -5 & 6 & 1 & & & & \\ 1 & 3 & -1 & 2 & -3 & & & \\ & 2 & 5 & -5 & 6 & -1 & & \\ & & -3 & 1 & -1 & 2 & -5 & \\ & & & 6 & 1 & -3 & 2 & -9 \\ & & & & -4 & 1 & -1 & 2 \\ & & & & & 5 & 1 & -7 \end{bmatrix}$$

$$D = \begin{bmatrix} 13 & 29 & -13 \\ -6 & 17 & -21 \\ -31 & -6 & 4 \\ 64 & 3 & 16 \\ -20 & 1 & -5 \\ -22 & -41 & 56 \\ -29 & 10 & -21 \\ 7 & -24 & 20 \end{bmatrix}$$

与带型矩阵 A 对应的矩形数组 B 为

$$B=\begin{bmatrix} 3 & -4 & 1 & 0 & 0 \\ -2 & -5 & 6 & 1 & 0 \\ 1 & 3 & -1 & 2 & -3 \\ 2 & 5 & 5 & 6 & -1 \\ -3 & 1 & -1 & 2 & -5 \\ 6 & 1 & -3 & 2 & -9 \\ -4 & 1 & -1 & 2 & 0 \\ 5 & 1 & -7 & 0 & 0 \end{bmatrix}$$

在本问题中，N=8，半带宽 L=2，带宽 IL=5，M=3。

主程序(文件名：ABAND0.FOR)为

```
      DIMENSION B(8,5),D(8,3)
      DOUBLE PRECISION B,D
      DATA B/3.0, 2.0,1.0,2.0,-3.0,6.0,-4.0,5.0,
     *       -4.0,-5.0,3.0,5.0,5*1.0,6.0,-1.0,
     *       -5.0, 1.0,-3.0, 1.0,-7.0,0.0,1.0,2.0,6.0,3*2.0,
     *       3*0.0,-3.0, 1.0,-5.0, 9.0,2*0.0/
      DATA D/13.0,-6.0, 31.0,64.0, 20.0,-22.0,-29.0,7.0,
     *       29.0,17.0,-6.0,3.0,1.0, 41.0,10.0, 21.0,
     *       -13.0, 21.0,1.0,16.0,-5.0,55.0,-21.0,20.0/
      CALL ABAND(B,D,8,2,5,3,IT)
      IF (IT.GT.0) THEN
        WRITE(*,10) ((D(I,J),J=1,3),I=1,8)
      END IF
10    FORMAT(1X,3D15.6)
      END
```

运行结果为

```
  .300000D+01     .400000D+01     .407091D-15
 -.100000D+01    -.300000D+01     .300000D+01
  .190887D-15     .200000D+01     .100000D+01
  .500000D+01    -.222045D-14    -.133227D-14
  .700000D+01     .968843D-15     .200000D+01
  .100000D+01     .100000D+01    -.300000D+01
  .200000D+01     .100000D+01    -.238524D-16
 -.928842D-15     .400000D+01    -.500000D-01
```

1.7 对称方程组的分解法

一、功能

用分解法求解系数矩阵为对称、且右端具有 m 组常数向量的线性代数方程组 $AX=C$。

二、方法说明

设系数矩阵 A 为对称，则可以分解为下列形式

$$A = LDL^T$$

其中 L 为单位下三角矩阵，D 为对角线矩阵。即

$$L = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \quad D = \begin{bmatrix} d_{11} & & & \\ & d_{22} & & \\ & & \ddots & \\ & & & d_{nn} \end{bmatrix}$$

矩阵 L 和 D 中的各元素由下列计算公式确定：

$$d_{11} = a_{11}$$

$$d_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 d_{kk}$$

$$l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} d_{kk} \right) / d_{jj},\ j<i$$

$$l_{ij} = 0,\ j>i$$

对于方程组

$$AX = B$$

当 L 和 D 确定后，令

$$DL^T X = Y$$

则由方程组

$$LY = B$$

解出 Y，再由方程组

$$DL^T X = Y$$

解出 X。其计算公式如下：

$$y_1 = b_1$$

$$y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k,\ i>1$$

$$x_n = y_n / d_{nn}$$

$$x_i = \left(y_i - \sum_{k=i+1}^{n} d_{ii} l_{ki} x_k \right) / d_{ii}$$

三、子程序语句

SUBROUTINE　ALDLE(A,N,M,C,L)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入参数。存放对称矩阵 A，返回时将被破坏。

N——整型变量，输入参数。方程组阶数。

M——整型变量，输入参数。方程组右端常数向量的组数。

C——双精度实型二维数组，体积为 N×M，输入兼输出参数。调用时存放 M 组常数

向量;返回 M 组解向量。

L——整型变量,输出参数。若返回 L=0,说明子程序工作失败;若 L≠0,表示正常返回。

五、子程序(文件名:ALDLE.FOR)

```
      SUBROUTINE ALDLE(A,N,M,C,L)
      DIMENSION A(N,N),C(N,M)
      DOUBLE PRECISION A,C
      L=1
      IF (ABS(A(1,1))+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,10)
        RETURN
      END IF
10    FORMAT(1X,'FAIL')
      DO 20 I=2,N
20    A(I,1)=A(I,1)/A(1,1)
      DO 60 I=2,N-1
        DO 30 J=2,I
30      A(I,I)=A(I,I)-A(I,J-1)*A(I,J-1)*A(J-1,J-1)
        DO 50 K=I+1,N
          DO 40 J=2,I
40        A(K,I)=A(K,I)-A(K,J-1)*A(I,J-1)*A(J-1,J-1)
          IF (ABS(A(I,I))+1.0.EQ.1.0) THEN
            L=0
            WRITE(*,10)
            RETURN
          END IF
          A(K,I)=A(K,I)/A(I,I)
50      CONTINUE
60    CONTINUE
      DO 70 J=2,N
70    A(N,N)=A(N,N)-A(N,J-1)*A(N,J-1)*A(J-1,J-1)
      DO 80 J=1,M
      DO 80 I=2,N
      DO 80 K=2,I
80    C(I,J)=C(I,J)-A(I,K-1)*C(K-1,J)
      DO 90 I=2,N
      DO 90 J=I,N
90    A(I-1,J)=A(I-1,I-1)*A(J,I-1)
      IF (ABS(A(N,N))+1.0.EQ.1.0) THEN
        L=0
```

```
      WRITE(*,10)
      RETURN
    END IF
    DO 150 J=1,M
      C(N,J)=C(N,J)/A(N,N)
      DO 140 K=2,N
        K1=N-K+2
        DO 130 K2=K1,N
          K3=N-K+1
          C(K3,J)=C(K3,J)-A(K3,K2)*C(K2,J)
130     CONTINUE
        C(K3,J)=C(K3,J)/A(K3,K3)
140   CONTINUE
150 CONTINUE
    RETURN
    END
```

六、例

求解对称方程组 $AX=C$，其中

$$A=\begin{bmatrix}5&7&6&5&1\\7&10&8&7&2\\6&8&10&9&3\\5&7&9&10&4\\1&2&3&4&5\end{bmatrix},\quad C=\begin{bmatrix}24&96\\34&136\\36&144\\35&140\\15&60\end{bmatrix}$$

主程序(文件名:ALDLE0.FOR)为

```
    DIMENSION A(5,5),C(5,2)
    DOUBLE PRECISION A,C
    DATA A/5.0,7.0,6.0,5.0,1.0,7.0,10.0,8.0,7.0,2.0,6.0,8.0,
  *       10.0,9.0,3.0,5.0,7.0,9.0,10.0,4.0,1.0,2.0,3.0,4.0,5.0/
    DATA C/24.0,34.0,36.0,35.0,15.0,96.0,136.0,144.0,140.0,60.0/
    CALL ALDLE(A,5,2,C,L)
    IF (L.NE.0) THEN
      WRITE(*,10) ((C(I,J),J=1,2),I=1,5)
    END IF
10  FORMAT(1X,2D15.6)
    END
```

运行结果为

```
 .100000D+01    .400000D+01
 .100000D+01    .400000D+01
 .100000D+01    .400000D+01
 .100000D+01    .400000D+01
 .100000D+01    .400000D+01
```

1.8 对称正定方程组的平方根法

一、功能

用乔里斯基(Cholesky)分解法(即平方根法)求解系数矩阵为对称正定、且右端具有多组常数向量的线性代数方程组 $AX = D$。

二、方法说明

对称正定矩阵 A 可以唯一地分解为

$$A = U^T U$$

其中 U 为上三角矩阵,即

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

U 矩阵中的各元素由下列计算公式确定:

$$u_{11} = \sqrt{a_{11}}$$

$$u_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2\right)^{\frac{1}{2}}, i = 2,3,\cdots,n$$

$$u_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}\right) / u_{ii}, j > i$$

于是,方程组 $AX = B$ 的解由下列公式计算:

$$y_i = \left(b_i - \sum_{k=1}^{i-1} u_{ki} y_k\right) / u_{ii}$$

$$x_i = \left(y_i - \sum_{k=i+1}^{n} u_{ik} x_k\right) / u_{ii}$$

三、子程序语句

SUBROUTINE ACHOL(A,N,M,D,L)

四、形参说明

A——双精度实型二维数组,体积为 N×N,输入兼输出参数,调用时存放对称正定的系数矩阵;返回分解式中的矩阵 U。

N——整型变量,输入参数。方程组的阶数。

M——整型变量,输入参数。方程组右端常数向量的组数。

D——双精度实型二维数组,体积为 N×M,输入兼输出参数。调用时存放方程组右端的 M 组常数向量;返回 M 组解向量。

L——整型变量,输出参数。若返回 L=0,说明子程序工作失败(如系数矩阵非对称正定);若 L≠0,表示正常返回。

五、子程序(文件名:ACHOL.FOR)

```
      SUBROUTINE ACHOL(A,N,M,D,L)
```

```
      DIMENSION A(N,N),D(N,M)
      DOUBLE PRECISION A,D
      L=1
      IF (A(1,1)+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,30)
        RETURN
      END IF
      A(1,1)=SQRT(A(1,1))
      DO 10 J=2,N
10    A(1,J)=A(1,J)/A(1,1)
      DO 100 I=2,N
        DO 20 J=2,I
20      A(I,I)=A(I,I)-A(J-1,I)*A(J-1,I)
        IF (A(I,I)+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,30)
          RETURN
        END IF
30      FORMAT(1X,'FAIL')
        A(I,I)=SQRT(A(I,I))
        IF (I.NE.N) THEN
          DO 50 J=I+1,N
          DO 40 K=2,I
40        A(I,J)=A(I,J)-A(K-1,I)*A(K-1,J)
50        A(I,J)=A(I,J)/A(I,I)
        END IF
100   CONTINUE
      DO 130 J=1,M
        D(1,J)=D(1,J)/A(1,1)
        DO 120 I=2,N
          DO 110 K=2,I
110       D(I,J)=D(I,J)-A(K-1,I)*D(K-1,J)
          D(I,J)=D(I,J)/A(I,I)
120     CONTINUE
130   CONTINUE
      DO 160 J=1,M
        D(N,J)=D(N,J)/A(N,N)
        DO 150 K=N,2,-1
          DO 140 I=K,N
140       D(K-1,J)=D(K-1,J)-A(K-1,I)*D(I,J)
          D(K-1,J)=D(K-1,J)/A(K-1,K-1)
```

```
160     CONTINUE
160   CONTINUE
      RETURN
      END
```

六、例

求解线性代数方程组

$$AX = D$$

其中 A 为对称正定矩阵。A 与 D 为

$$A = \begin{bmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{bmatrix}, \quad D = \begin{bmatrix} 23 & 92 \\ 32 & 128 \\ 33 & 132 \\ 31 & 124 \end{bmatrix}$$

主程序(文件名:ACHOL0.FOR)为

```
      DIMENSION A(4,4),D(4,2)
      DOUBLE PRECISION A,D
      DATA A/5.0,7.0,6.0,5.0,7.0,10.0,8.0,7.0,
     *          6.0,8.0,10.0,9.0,5.0,7.0,9.0,10.0/
      DATA D/23.0,32.0,33.0,31.0,92.0,128.0,132.0,124.0/
      CALL ACHOL(A,4,2,D,L)
      IF (L.NE.0) THEN
        WRITE(*,10) ((D(I,J),J=1,2),I=1,4)
      END IF
10    FORMAT(1X,2D15.6)
      END
```

运行结果为

```
.100000D+01    .400000D+01
.100000D+01    .400000D+01
.100000D+01    .400000D+01
.100000D+01    .400000D+01
```

1.9 大型稀疏方程组

一、功能

用全选主元高斯-约当(Gauss-Jordan)消去法求解系数矩阵为稀疏矩阵的大型方程组 $AX = B$。

二、方法说明

基本方法同 1.2 节,只是在消去过程中,避免了对零元素的运算,因而可以大大减少运算次数。本子程序不考虑稀疏矩阵的压缩存储问题。

三、子程序语句

SUBROUTINE　AGGJE(A,N,B,L,JS)

四、形参说明

A——双精度实型二维数组，体积为N×N，输入参数。存放系数矩阵，在返回时将被破坏。

N——整型变量。输入参数。方程组阶数。

B——双精度实型一维数组，长度为N，输入兼输出参数。调用时存放方程组右端的常数向量；返回方程组的解向量。

L——整型变量，输出参数。若返回L=0，说明系数矩阵奇异，求解失败；若L≠0，表示正常返回。

JS——整型一维数组，长度为N。本子程序的工作数组。

五、子程序(文件名：AGGJE.FOR)

```
      SUBROUTINE AGGJE(A,N,B,L,JS)
      DIMENSION A(N,N),B(N),JS(N)
      DOUBLE PRECISION A,B,T
      L=1
      DO 100 K=1,N
        D=0.0
        DO 10 I=K,N
        DO 10 J=K,N
          IF (ABS(A(I,J)).GT.D) THEN
            D=ABS(A(I,J))
            JS(K)=J
            IS=I
          END IF
10      CONTINUE
        IF (D+1.0.EQ.1.0) THEN
          WRITE(*,20)
          L=0
          RETURN
        END IF
20      FORMAT(1X,' FAIL ')
        DO 30 J=K,N
          T=A(K,J)
          A(K,J)=A(IS,J)
          A(IS,J)=T
30      CONTINUE
        T=B(K)
        B(K)=B(IS)
        B(IS)=T
        DO 50 I=1,N
          T=A(I,K)
          A(I,K)=A(I,JS(K))
```

```
          A(I,JS(K))=T
   50   CONTINUE
        T=A(K,K)
        DO 60 J=K+1,N
          IF (A(K,J).NE.0.0) A(K,J)=A(K,J)/T
   60   CONTINUE
        B(K)=B(K)/T
        DO 80 J=K+1,N
          IF (A(K,J).NE.0.0) THEN
            DO 70 I=1,N
              IF ((I.NE.K).AND.(A(I,K).NE.0.0)) THEN
                A(I,J)=A(I,J)-A(I,K)*A(K,J)
              END IF
   70       CONTINUE
          END IF
   80   CONTINUE
        DO 90 I=1,N
          IF ((I.NE.K).AND.(A(I,K).NE.0.0)) THEN
            B(I)=B(I)-A(I,K)*B(K)
          END IF
   90   CONTINUE
  100 CONTINUE
      DO 110 K=N,1,-1
        IF (K.NE.JS(K)) THEN
          T=B(K)
          B(K)=B(JS(K))
          B(JS(K))=T
        END IF
  110 CONTINUE
      RETURN
      END
```

六、例

求解稀疏方程组 $AX=B$。其中

$$A=\begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 6 & 0 & 0 & 0 & -6 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -4 \\ 3 & 0 & 0 & 0 & -2 & 0 & 1 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 5 & 0 \\ 1 & 0 & 0 & 0 & -3 & 0 & 0 & 2 \\ 0 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & -2 \end{bmatrix}$$

$$B = (4,6,-8,-2,27,-9,2,-4)^T$$

主程序(文件名:AGGJE0.FOR)为

```
      DIMENSION A(8,8),B(8),JS(8)
      DOUBLE PRECISION A,B
      DATA A/3*0.0,3.0,0.0,0.1.0,3*0.0,6.0,4*0.0,4.0,0.0,-1.0,
     *       3*0.0,6.0,2*0.0,1.0,2*0.0,2.0,3*0.0,-1.0,4*0.0,
     *       -2.0,0.0,-3.0,0.0,-1.0,0.0,-6.0,6*0.0,2.0,2*0.0,
     *       1.0,5.0,5*0.0,-4.0,2*0.0,2.0,0.0,-2.0/
      DATA B/4.0,6.0,-8.0,-2.0,27.0,-9.0,2.0,-4.0/
      CALL AGGJE(A,8,B,L,JS)
      IF (L.NE.0) THEN
        WRITE(*,10) (B(I),I=1,8)
      END IF
10    FORMAT(1X,D15.6)
      END
```

运行结果为

```
 .100000D+01
 .000000D+00
 .200000D+01
-.200000D+01
 .400000D+01
-.100000D+01
 .300000D+01
 .100000D+01
```

1.10 托伯利兹方程组的列文逊方法

一、功能

用列文逊(Levinson)递推算法求解托伯利兹(Toeplitz)型方程组。

二、方法说明

对称 Toeplitz 矩阵的形式如下

$$T^{(n)} = \begin{bmatrix} t_1 & t_2 & t_3 & \cdots & t_n \\ t_2 & t_1 & t_2 & \cdots & t_{n-1} \\ t_3 & t_2 & t_1 & \cdots & t_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_n & t_{n-1} & t_{n-2} & \cdots & t_1 \end{bmatrix}$$

简称对称 T 型矩阵。

设线性代数方程组 $AX = B$ 的系数矩阵为对称 T 型矩阵,则求解该方程组的递推算法如下。

取初值 $\alpha_1 = t_1, y_1^{(1)} = 1, x_1^{(1)} = b_1/t_1$。

对于 $k=1,2,\cdots,n-1$，依次作如下计算：

$$q_{k+1}=\sum_{j=1}^{k}x_j^{(k)}t_{k-j+2}$$

$$\beta_k=\sum_{j=1}^{k}y_j^{(k)}t_{j+1}$$

$$c_k=-\beta_k/\alpha_k$$

$$\begin{cases} y_1^{(k+1)}=c_k y_1^{(k)} \\ y_i^{(k+1)}=y_{i-1}^{(k)}+c_k y_{k-i+1}^{(k)},\ i=2,3,\cdots,k \\ y_{k+1}^{(k+1)}=y_k^{(k)} \end{cases}$$

$$\alpha_{k+1}=\alpha_k+c_k\beta_k$$

$$\omega_{k+1}=(b_{k+1}-q_{k+1})/\alpha_{k+1}$$

$$\begin{cases} x_i^{(k+1)}=x_i^{(k)}+\omega_{k+1}y_i^{(k+1)},\ i=1,2,\cdots,k \\ x_{k+1}^{(k+1)}=\omega_{k+1}y_{k+1}^{(k+1)} \end{cases}$$

三、子程序语句

SUBROUTINE　ATLVS(T,N,B,X,L,Y,S)

四、形参说明

T——双精度实型一维数组，长度为N，输入参数。对称 T 型矩阵中的元素，即 $T=(t_1,t_2,\cdots,t_n)$。

N——整型变量，输入参数。方程组阶数。

B——双精度实型一维数组，长度为N，输入参数。方程组右端的常数向量。

X——双精度实型一维数组，长度为N，输出参数。返回方程组的解向量。

L——整型变量，输出参数。若返回 L=0，说明子程序工作失败；若 L≠0，表示正常返回。

Y,S——均为双精度实型一维数组，长度为N。本子程序的工作数组。

五、子程序(文件名:ATLVS.FOR)

```
      SUBROUTINE ATLVS(T,N,B,X,L,Y,S)
      DIMENSION T(N),B(N),X(N),Y(N),S(N)
      DOUBLE PRECISION T,B,X,Y,S,A,C,H,Q,BETA
      L=1
      A=T(1)
      IF (ABS(A)+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,100)
        RETURN
      END IF
100   FORMAT(1X,' FAIL')
      Y(1)=1.0
      X(1)=B(1)/T(1)
```

```
      DO 40 K=1,N-1
        BETA=0.0
        Q=0.0
        DO 10 J=1,K
          BETA=BETA+Y(J)*T(J+1)
          Q=Q+X(J)*T(K-J+2)
10      CONTINUE
        IF (ABS(A)+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,100)
          RETURN
        END IF
        C=-BETA/A
        S(1)=C*Y(K)
        Y(K+1)=Y(K)
        IF (K.NE.1) THEN
          DO 20 I=2,K
20        S(I)=Y(I-1)+C*Y(K-I+1)
        END IF
        A=A+C*BETA
        IF (ABS(A)+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,100)
          RETURN
        END IF
        H=(B(K+1)-Q)/A
        DO 30 I=1,K
          X(I)=X(I)+H*S(I)
          Y(I)=S(I)
30      CONTINUE
        X(K+1)=H*Y(K+1)
40    CONTINUE
      RETURN
      END
```

六、例

求解托伯利兹方程组 $AX=B$。其中

$$A=\begin{bmatrix}6&5&4&3&2&1\\5&6&5&4&3&2\\4&5&6&5&4&3\\3&4&5&6&5&4\\2&3&4&5&6&5\\1&2&3&4&5&6\end{bmatrix},\quad B=\begin{bmatrix}11\\9\\9\\9\\13\\17\end{bmatrix}$$

即 $T=(6,5,4,3,2,1)$。

主程序(文件名:ATLVS0.FOR)为

```
      DIMENSION T(6),B(6),X(6),Y(6),S(6)
      DOUBLE PRECISION T,B,X,Y,S
      DATA T/6.0,5.0,4.0,3.0,2.0,1.0/
      DATA B/11.0,9.0,9.0,9.0,13.0,17.0/
      CALL ATLVS(T,6,B,X,L,Y,S)
      IF (L.NE.0) THEN
        WRITE(*,10) (X(I),I=1,6)
      END IF
10    FORMAT(1X,D15.6)
      END
```

运行结果为

```
  .300000D+01
 -.100000D+01
  .117321D-14
 -.200000D+01
 -.210942D-14
  .400000D+01
```

1.11 高斯-赛德尔迭代法

一、功能

用高斯-赛德尔(Gauss-Seidel)迭代法求解系数矩阵为主对角线占绝对优势的线性代数方程组 $AX=B$。

二、方法说明

如果方程组

$$\sum_{j=1}^{n}a_{ij}x_j=b_i,i=1,2,\cdots,n$$

的系数矩阵具有主对角线优势,即满足下列条件:

$$\sum_{\substack{j=1\\j\neq i}}^{n}|a_{ij}|<|a_{ii}|,i=1,2,\cdots,n$$

则高斯-赛德尔迭代格式

$$x_i^{(k+1)}=\left(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}\right)/a_{ii},i=1,2,\cdots,n$$

收敛。

在本子程序中,初值取

$$x_i^{(0)}=0,i=1,2,\cdots,n$$

结束迭代的条件为

$$\max_{1\le i\le n}\frac{|x_i^{(k+1)}-x_i^{(k)}|}{1+|x_i^{(k+1)}|}<\varepsilon$$

其中 ε 为给定的精度要求。

三、子程序语句

SUBROUTINE AGSDL(A,B,N,X,EPS,L)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入参数。方程组的系数矩阵，要求主对角线占绝对优势。

B——双精度实型一维数组，长度为 N，输入参数。方程组右端的常数向量。

N——整型变量，输入参数。方程组阶数。

X——双精度实型一维数组，长度为 N，输出参数。返回方程组的解向量。

EPS——实型变量，输入参数。控制精度要求。

L——整型变量，输出参数。若返回 L=0，说明子程序工作失败(如系数矩阵奇异，迭代不收敛)；若 L≠0，表示正常返回。

五、子程序(文件名：AGSDL.FOR)

```
      SUBROUTINE AGSDL(A,B,N,X,EPS,L)
      DIMENSION A(N,N),B(N),X(N)
      DOUBLE PRECISION A,B,X,T,S,P,Q
      DO 5 I=1,N
        IF (ABS(A(I,I))+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,100)
          RETURN
        END IF
5     CONTINUE
100   FORMAT(1X,' FAIL')
      L=100
      DO 10 I=1,N
10    X(I)=0.0
20    P=0.0
      L=L-1
      DO 50 I=1,N
        T=X(I)
        S=0.0
        DO 30 J=1,N
          IF (J.NE.I) S=S+A(I,J)*X(J)
30      CONTINUE
        X(I)=(B(I)-S)/A(I,I)
        Q=ABS(X(I)-T)/(1+ABS(X(I)))
        IF (Q.GT.P) P=Q
```

```
50    CONTINUE
      IF ((P.GE.EPS).AND.(L.NE.0)) GOTO 20
      IF (L.EQ.0) WRITE(*,100)
      RETURN
      END
```

六、例

用高斯-赛德尔迭代法求解下列线性代数方程组

$$\begin{cases} 7x_1+2x_2+x_3-2x_4=4 \\ 9x_1+15x_2+3x_3-2x_4=7 \\ -2x_1-2x_2+11x_3+5x_4=-1 \\ x_1+3x_2+2x_3+13x_4=0 \end{cases}$$

取 $\varepsilon=10^{-6}$。

主程序(文件名:AGSDL0.FOR)为

```
      DIMENSION A(4,4),B(4),X(4)
      DOUBLE PRECISION A,B,X
      DATA A/7.0,9.0,-2.0,1.0,2.0,15.0,-2.0,3.0,
     *       1.0,3.0,11.0,2.0,-2.0,-2.0,5.0,13.0/
      DATA B/4.0,7.0,-1.0,0.0/
      EPS=1.0E-06
      CALL AGSDL(A,B,4,X,EPS,L)
      IF (L.NE.0) THEN
        WRITE(*,10) (X(I),I=1,4)
      END IF
10    FORMAT(1X,D15.6)
      END
```

运行结果为

```
  .497931D+00
  .144494D+00
  .628581D-01
 -.813176D-01
```

1.12 对称正定方程组的共轭梯度法

一、功能

用共轭梯度法求解对称正定方程组 $AX=B$。

二、方法说明

共轭梯度法的递推计算公式如下。

取初值 $X_1=(0,0,\cdots,0)^T$,$R_1=P_1=B$。

对于 $i=1,2,\cdots,n$,作如下计算:

$$\alpha_i = \frac{(P_i, B)}{(P_i, AP_i)}$$

$$X_{i+1} = X_i + \alpha_i P_i$$

$$R_{i+1} = B - AX_{i+1}$$

$$\beta_i = \frac{(R_{i+1}, AP_i)}{(P_i, AP_i)}$$

$$P_{i+1} = R_{i+1} - \beta_i P_i$$

上述过程一直作到 $\| R_i \| < \varepsilon$ 或 $i = n$ 为止。

特别要指出，本方法只适用于对称正定方程组。

三、子程序语句

SUBROUTINE　AGRAD(A,N,B,EPS,X,P,R,S,Q)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入参数。存放对称正定矩阵。

N——整型变量，输入参数。方程组阶数。

B——双精度实型一维数组，长度为 N，输入参数。方程组右端的常数向量。

EPS——实型变量，输入参数。控制精度要求。

X——双精度实型一维数组，长度为 N，输出参数。返回方程组的解向量。

P,R,S,Q——均为双精度一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名：AGRAD.FOR)

```
      SUBROUTINE AGRAD(A,N,B,EPS,X,P,R,S,Q)
      DIMENSION A(N,N),B(N),X(N),P(N),R(N),S(N),Q(N)
      DOUBLE PRECISION A,B,X,P,R,S,Q,ALPHA,BETA,D,E
      DO 10 I=1,N
        X(I)=0.0
        P(I)=B(I)
        R(I)=B(I)
10    CONTINUE
      I=1
20    CALL BRMUL(A,P,N,N,1,S)
      D=0.0
      E=0.0
      DO  30  K=1,N
        D=D+P(K)*B(K)
        E=E+P(K)*S(K)
30    CONTINUE
      ALPHA=D/E
      DO  40  K=1,N
40    X(K)=X(K)+ALPHA*P(K)
      CALL BRMUL(A,X,N,N,1,Q)
      D=0.0
```

```
      DO  50  K=1,N
        R(K)=B(K)-Q(K)
        D=D+R(K)*S(K)
50    CONTINUE
      BETA=D/E
      D=0.0
      DO  55  K=1,N
55    D=D+R(K)*R(K)
      D=SQRT(D)
      IF (D.LT.EPS) RETURN
      DO  60  K=1,N
60    P(K)=R(K)-BETA*P(K)
      I=I+1
      IF (I.LE.N) GOTO 20
      RETURN
      END
```

六、例

用共轭梯度法求解对称正定方程组 $AX=B$，其中

$$A=\begin{bmatrix}5 & 7 & 6 & 5\\ 7 & 10 & 8 & 7\\ 6 & 8 & 10 & 9\\ 5 & 7 & 9 & 10\end{bmatrix},\ B=\begin{bmatrix}23\\32\\33\\31\end{bmatrix}$$

取 $\varepsilon=10^{-6}$。

主程序(文件名:AGRAD0.FOR)为

```
      DIMENSION A(4,4),B(4),X(4),P(4),R(4),S(4),Q(4)
      DOUBLE PRECISION A,B,X,P,R,S,Q
      DATA A/5.0,7.0,6.0,5.0,7.0,10.0,8.0,7.0,
     *        6.0,8.0,10.0,9.0,5.0,7.0,9.0,10.0/
      DATA B/23.0,32.0,33.0,31.0/
      EPS=0.000001
      CALL AGRAD(A,4,B,EPS,X,P,R,S,Q)
      WRITE(*,*)
      WRITE(*,10) (X(I),I=1,4)
      WRITE(*,*)
10    FORMAT(1X,D15.6)
      END
```

运行结果为

```
 .100000D+01
 .100000D+01
 .100000D+01
 .100000D+01
```

七、附注

本子程序需要调用实矩阵相乘子程序 BRMUL，参看 2.1 节。

1.13 线性最小二乘问题的豪斯荷尔德变换法

一、功能

用豪斯荷尔德(House holder)变换法求解最小二乘问题。

二、方法说明

设超定方程组为

$$AX = B$$

其中 A 为 $m \times n$ 阶 $(m \geqslant n)$ 列线性无关的矩阵，X 为 n 维列向量，B 为 m 维列向量。

用豪斯荷尔德变换将 A 进行 QR 分解，即

$$A = QR$$

关于 QR 分解的方法见 2.11 节。

设

$$E = B - AX$$

用 Q^T 左乘上式两端得

$$\begin{aligned} Q^TE &= Q^TB - Q^TAX \\ &= Q^TB - RX \end{aligned}$$

因为 Q^T 为正交矩阵，所以有

$$\| E \|_2^2 = \| Q^TE \|_2^2 = \| Q^TB - RX \|_2^2$$

若令

$$Q^TB = \begin{bmatrix} C \\ D \end{bmatrix}, \ RX = \begin{bmatrix} R_1 \\ O \end{bmatrix} X$$

其中 C 为 n 维列向量，D 为 $m-n$ 维列向量，R_1 为 $n \times n$ 阶上三角方阵。O 为 $(m-n) \times n$ 阶的零矩阵。

则有

$$\| E \|_2^2 = \| C - R_1X \|_2^2 + \| D \|_2^2$$

显然，当 X 满足

$$R_1X = C$$

时，$\| E \|_2^2$ 将取最小值。

由上所述，求解线性最小二乘问题

$$AX = B$$

的步骤如下：

(1) 对 A 进行 QR 分解，即

$$A = QR$$

其中 Q 为 $m \times m$ 阶正交矩阵，R 为右上三角矩阵。且令

$$R=\begin{bmatrix}R_1\\O\end{bmatrix}$$

其中 R_1 为 $n\times n$ 阶上三角矩阵。

(2) 计算

$$\begin{bmatrix}C\\D\end{bmatrix}=Q^TB$$

其中 C 为 n 维列向量。

(3) 利用回代求解方程组

$$R_1X=C$$

三、子程序语句

SUBROUTINE　AGMQR(A,M,N,B,Q,L,C)

四、形参说明

A——双精度实型二维数组，体积为 M×N，输入兼输出参数。调用时存放超定方程组的系数矩阵；返回 QR 分解式中的 R 矩阵。

M——整型变量，输入参数。方程组方程个数。M≥N。

N——整型变量，输入参数。未知数个数。N≤M。

B——双精度实型一维数组，长度为 M，输入兼输出参数。调用时存放方程组右端的常数向量；返回方程组的最小二乘解(前 N 个)。

Q——双精度实型二维数组，体积为 M×M，输出参数。返回 QR 分解式的 Q 矩阵。

L——整型变量，输出参数。若返回 L=0，说明矩阵 A 列线性相关，程序工作失败；若 L≠0，表示正常返回。

C——双精度实型一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名：AGMQR.FOR)

```
      SUBROUTINE AGMQR(A,M,N,B,Q,L,C)
      DIMENSION A(M,N),B(M),Q(M,M),C(N)
      DOUBLE PRECISION A,B,Q,C,D
      CALL BMAQR(A,M,N,Q,L)
      IF (L.EQ.0) RETURN
      DO  20  I=1,N
        D=0.0
        DO  10  J=1,M
10      D=D+Q(J,I)*B(J)
        C(I)=D
20    CONTINUE
      B(N)=C(N)/A(N,N)
      DO  40  I=N-1,1,-1
        D=0.0
        DO  30  J=I+1,N
30      D=D+A(I,J)*B(J)
```

```
      B(I)=(C(I)-D)/A(I,I)
40    CONTINUE
      RETURN
      END
```

六、例

求下列超定方程组的最小二乘解，并求系数矩阵的 QR 分解式：

$$
\begin{cases}
x_1 + x_2 - x_3 = 2 \\
2x_1 + x_2 = -3 \\
x_1 - x_2 = 1 \\
-x_1 + 2x_2 + x_3 = 4
\end{cases}
$$

主程序(文件名:AGMQR0.FOR)为

```
      DIMENSION A(4,3),B(4),Q(4,4),C(3)
      DOUBLE PRECISION A,B,Q,C
      DATA A/1.0,2.0,1.0,-1.0,1.0,1.0,-1.0,2.0,-1.0,0.0,0.0,1.0/
      DATA B/2.0,-3.0,1.0,4.0/
      M=4
      N=3
      CALL AGMQR(A,M,N,B,Q,L,C)
      IF (L.NE.0) THEN
        WRITE(*,*)
        WRITE(*,10) (I,B(I),I=1,N)
        WRITE(*,*)
        WRITE(*,20)
        WRITE(*,30) ((Q(I,J),J=1,M),I=1,M)
        WRITE(*,*)
        WRITE(*,40)
        WRITE(*,50) ((A(I,J),J=1,N),I=1,M)
        WRITE(*,*)
      END IF
10    FORMAT(1X,'X(',I2,')=',D15.6)
20    FORMAT(1X,'MAT Q IS:')
30    FORMAT(1X,4D15.6)
40    FORMAT(1X,'MAT R IS:')
50    FORMAT(1X,3D15.6)
      END
```

运行结果为

```
X( 1 )=   -.119048D+01
X( 2 )=    .952381D+00
X( 3 )=   -.666667D+00
```

MAT Q IS:

-.377964D+00	-.377964D+00	.755929D+00	.377964D+00
-.755929D+00	-.377964D+00	-.377964D+00	-.377964D+00
-.377964D+00	.377964D+00	-.377964D+00	.755929D+00
.377964D+00	-.755929D+00	-.377964D+00	.377964D+00

MAT R IS:

-.264575D+01	-.195590D-15	.755929D+00
.000000D+00	-.264575D+01	-.377964D+00
.000000D+00	.000000D+00	-.113389D+01
.000000D+00	.000000D+00	.000000D+00

七、附注

本子程序需要调用 QR 分解子程序 BMAQR，参看 2.11 节。

1.14 线性最小二乘问题的广义逆法

一、功能

利用奇异值分解求线性方程组 $AX=B$ 的系数矩阵 A 的广义逆 A^+ 及线性最小二乘解。

二、方法说明

设线性方程组为

$$AX = B$$

其中 A 为 $m\times n$ 阶矩阵，且列线性无关，则对于矩阵 A，存在一个 $m\times m$ 阶的列正交矩阵 U 和 $n\times n$ 阶的列正交矩阵 V，使

$$A = U\begin{bmatrix}\Sigma & O\\ O & O\end{bmatrix}V^T$$

成立。其中 $\Sigma=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_r)(r\leqslant\min\{m,n\})$，且 $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_r>0$。上式称为实矩阵 A 的奇异值分解式，$\sigma_i(i=1,2,\cdots,r)$ 称为 A 的奇异值。

利用 A 的奇异值分解式，可以计算 A 的广义逆 A^+ 及方程组的最小二乘解 X。

设 U_1 为 U 中前 r 列列正交向量组构成的 $m\times r$ 阶矩阵，V_1 为 V 中前 r 列列正交向量组构成的 $n\times r$ 阶矩阵，则 A 的广义逆为

$$A^+ = V_1\Sigma^{-1}U_1^T$$

方程组的最小二乘解为

$$X = A^+B = V_1\Sigma^{-1}U_1^TB$$

关于奇异值分解见 2.12 节。

三、子程序语句

SUBROUTINE AGMIV(M,N,A,B,AA,X,L,EPS,U,V,KA,S,E,WORK)

四、形参说明

M——整型变量,输入参数。方程个数。

N——整型变量,输入参数。未知数个数。

A——双精度实型二维数组,体积为 M×N,输入兼输出参数。调用时存放方程组的系数矩阵;返回奇异值 $\sigma_1,\sigma_2,\cdots,\sigma_r$(依次存放在对角线上)。

B——双精度实型一维数组,长度为 M,输入参数。方程组右端的常数向量。

AA——双精度实型二维数组,体积为 N×M,输出参数。返回 A 的广义逆 A^+ 。

X——双精度实型一维数组,长度为 N,输出参数。返回方程组的最小二乘解。

L——整型变量,输出参数,若返回 L=0,表示正常返回;若 L≠0,说明奇异值分解失败。

EPS——实型变量,输入参数。控制精度要求。

U——双精度实型二维数组,体积为 M×M,输出参数。返回奇异值分解式中的左奇异向量 U 。

V——双精度实型二维数组,体积为 N×N,输出参数,返回奇异值分解式中的右奇异向量 V^T 。

KA——整型变量,输入参数。KA=max{M,N}+1。

S,E,WORK——均为双精度实型一维数组,长度为 KA。本子程序的工作数组。

五、子程序(文件名:AGMIV. FOR)

```
      SUBROUTINE AGMIV(M,N,A,B,AA,X,L,EPS,U,V,KA,S,E,WORK)
      DIMENSION A(M,N),U(M,M),V(N,N),B(M),AA(N,M),X(N)
      DIMENSION S(KA),E(KA),WORK(KA)
      DOUBLE PRECISION A,U,V,B,AA,X,S,E,WORK
      CALL BMUAV(A,M,N,U,V,L,EPS,KA,S,E,WORK)
      IF (L.EQ.0) THEN
        K=1
10      IF (A(K,K).NE.0.0) THEN
          K=K+1
          IF (K.LE.MIN(M,N)) GOTO 10
        END IF
        K=K-1
        IF (K.NE.0) THEN
          DO 40 I=1,N
          DO 40 J=1,M
            AA(I,J)=0.0
            DO 30 II=1,K
30          AA(I,J)=AA(I,J)+V(II,I)*U(J,II)/A(II,II)
40        CONTINUE
        END IF
        DO 80 I=1,N
          X(I)=0.0
```

```
      DO 70 J=1,M
70      X(I)=X(I)+AA(I,J)*B(J)
80    CONTINUE
      END IF
      RETURN
      END
```

六、例

求下列超定方程组的最小二乘解，并求系数矩阵的广义逆 A^+，再求 A^+ 的广义逆 A^{++}：

$$\begin{cases} x_1 + x_2 - x_3 = 2 \\ 2x_1 + x_2 = -3 \\ x_1 - x_2 = 1 \\ -x_1 + 2x_2 + x_3 = 4 \end{cases}$$

主程序(文件名:AGMIV0.FOR)为

```
      DIMENSION A(4,3),U(4,4),V(3,3),B(4),C(3,4),X(3)
      DIMENSION S(5),E(5),WORK(5)
      DOUBLE PRECISION A,U,V,B,C,X,S,E,WORK
      DATA A/1.0,2.0,1.0,-1.0,1.0,1.0,-1.0,2.0,-1.0,2*0.0,1.0/
      DATA B/2.0,-3.0,1.0,4.0/
      M=4
      N=3
      KA=5
      EPS=0.000001
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(1X,'MAT A IS:')
      WRITE(*,200) ((A(I,J),J=1,N),I=1,M)
      CALL AGMIV(M,N,A,B,C,X,L,EPS,U,V,KA,S,E,WORK)
      IF (L.EQ.0) THEN
        WRITE(*,*)
        WRITE(*,50)
50      FORMAT(1X,'MAT A+ IS:')
        WRITE(*,60) ((C(I,J),J=1,M),I=1,N)
60      FORMAT(1X,4D13.6)
        WRITE(*,*)
        WRITE(*,*) 'THE SOLUTION TO THE LEAST SQUARES PROBLEM IS:'
        WRITE(*,100) (I,X(I),I=1,N)
100     FORMAT(1X,'X(',I2,')=',D15.6)
      END IF
      WRITE(*,*)
      CALL BGINV(N,M,C,A,L,EPS,U,V,KA,S,E,WORK)
      IF (L.EQ.0) THEN
```

```
      WRITE(*,70)
70    FORMAT(1X,'MAT A++ IS:')
      WRITE(*,200) ((A(I,J),J=1,N),I=1,M)
200   FORMAT(1X,3D15.6)
      WRITE(*,*)
    END IF
    END
```

运行结果为

```
MAT A IS:
   .100000D+01    .100000D+01   -.100000D+01
   .200000D+01    .100000D+01    .000000D+00
   .100000D+01   -.100000D+01    .000000D+00
  -.100000D+01    .200000D+01    .100000D+01
MAT A+ IS:
  -.476190D-01    .380952D+00    .238095D+00   -.476190D-01
   .238095D+00    .952381D-01   -.190476D+00    .238095D+00
  -.666667D+00    .333333D+00    .333333D+00    .333333D+00
THE SOLUTION TO THE LEAST SQUARES PROBLEM IS:
X( 1)=   -.119048D+01
X( 2)=    .952381D+00
X( 3)=   -.666667D+00
MAT A++ IS:
   .100000D+01    .100000D+01   -.100000D+01
   .200000D+01    .100000D+01   -.263446D-06
   .100000D+01   -.100000D+01    .658622D-07
  -.100000D+01    .200000D+01    .100000D+01
```

七、附注

本子程序需要调用奇异值分解的子程序 BMUAV，参看 2.12 节。

在本例的主程序中还需调用求广义逆的子程序 BGINV，参看 2.13 节。

1.15 病态线性方程组

一、功能

求解病态线性方程组 $AX=B$

二、方法说明

设线性代数方程组

$$AX=B$$

是病态的，求解病态方程组的步骤如下：

(1) 用全选主元高斯消去法求解，得到一组近似解 $X^{(1)}=(x_1^{(1)},x_2^{(1)},\cdots,x_n^{(1)})^T$。

(2) 计算剩余向量

$$R=B-AX^{(1)}$$

(3) 用全选主元高斯消去法求解线性方程组

$$AE=R$$

解出 $E=(e_1,e_2,\cdots,e_n)^T$

(4) 计算 $X^{(2)}=X^{(1)}+E$

(5) 令 $X^{(1)}=X^{(2)}$，转(2)重复这个过程。

上述过程一直作到

$$\max_{1\le i\le n}\frac{|x_i^{(2)}-x_i^{(1)}|}{1+|x_i^{(2)}|}<\varepsilon$$

为止。

三、子程序语句

SUBROUTINE　ABINT(A,N,B,EPS,X,L,P,JS,R,E)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入参数。方程组的系数矩阵。

N——整型变量，输入参数。方程组阶数。

B——双精度实型一维数组，长度为 N，输入参数。方程组右端的常数向量。

EPS——实型变量，输入参数。控制精度要求。

X——双精度实型一维数组，长度为 N，输出参数。返回方程组的解向量。

L——整型变量，输出参数。若返回 L=0，说明方程组求解失败；若 L≠0，表示正常返回。

P——双精度实型二维数组，体积为 N×N。本子程序的工作数组。

JS——整型一维数组，长度为 N。本子程序的工作数组。

R,E——均为双精度实型一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名：ABINT.FOR)

```
      SUBROUTINE ABINT(A,N,B,EPS,X,L,P,JS,R,E)
      DIMENSION A(N,N),B(N),X(N),JS(N),P(N,N),R(N),E(N)
      DOUBLE PRECISION A,B,X,P,R,E,Q,QQ
      I=60
      DO 10 K=1,N
      DO 10 J=1,N
10    P(K,J)=A(K,J)
      DO 20 K=1,N
20    E(K)=B(K)
      CALL AGAUS(P,E,N,X,L,JS)
      IF (L.EQ.0) RETURN
5     IF (I.EQ.0) THEN
        L=0
        RETURN
```

```
      END IF
      I=I-1
      CALL BRMUL(A,X,N,N,1,E)
      DO 30 K=1,N
30    E(K)=B(K)-E(K)
      DO 40 K=1,N
      DO 40 J=1,N
40    P(K,J)=A(K,J)
      CALL AGAUS(P,R,N,E,L,JS)
      IF (L.EQ.0) RETURN
      Q=0.0
      DO 50 K=1,N
        QQ=ABS(E(K))/(1.0+ABS(X(K)+E(K)))
        IF (QQ.GT.Q) Q=QQ
50    CONTINUE
      DO 60 K=1,N
60    X(K)=E(K)+X(K)
      IF (Q.GE.EPS) GOTO 5
      RETURN
      END
```

六、例

求解下列病态方程组

$$\begin{bmatrix} 3.4336 & -0.5238 & 0.67105 & -0.15272 \\ -0.5238 & 3.28326 & -0.73051 & -0.2689 \\ 0.67105 & -0.73051 & 4.02612 & 0.01835 \\ -0.15272 & -0.2689 & 0.01835 & 2.75702 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.5 \\ 2.5 \\ -2.0 \end{bmatrix}$$

$\varepsilon = 10^{-6}$。

主程序(文件名:ABINT0.FOR)为

```
      DIMENSION A(4,4),B(4),X(4),P(4,4),R(4),E(4),JS(4)
      DOUBLE PRECISION A,B,X,P,R,E
      DATA A/3.4336,-0.5238,0.67105,-0.15272,
     *       -0.5238,3.28326,-0.73051,-0.2689,
     *       0.67105,-0.73051,4.02612,0.01835,
     *       -0.15272,-0.2689,0.01835,2.75702/
      DATA B/-1.0,1.5,2.5,-2.0/
      N=4
      EPS=0.000001
      CALL ABINT(A,N,B,EPS,X,L,P,JS,R,E)
      WRITE(*,*)
      WRITE(*,*)'L=',L
      IF (L.NE.0) THEN
```

```
      WRITE(*,*)
      DO 10 K=1,N
10    WRITE(*,100) K,X(K)
      WRITE(*,*)
    END IF
100 FORMAT(3X,'X(',I2,')=',D15.6)
    END
```

运行结果为

```
L=          1

X( 1 )=   -.397718D+00
X( 2 )=    .510054D+00
X( 3 )=    .782984D+00
X( 4 )=   -.702916D+00
```

七、附注

本子程序需要调用全选主元高斯消去法子程序 AGAUS，参看 1.1 节；还要调用实矩阵相乘子程序，参看 2.1 节。

第2章 矩阵运算

2.1 实矩阵相乘

一、功能

求 $m \times n$ 阶矩阵 A 与 $n \times k$ 阶矩阵 B 的乘积矩阵。

二、方法说明

设

$$C = AB$$

如果 A 为 $m \times n$ 阶矩阵，B 为 $n \times k$ 阶矩阵，则 C 为 $m \times k$ 阶矩阵。乘积矩阵 C 的各元素为

$$c_{ij} = \sum_{l=1}^{n} a_{il} b_{lj}, i = 1,2,\cdots,m; j = 1,2,\cdots,k$$

三、子程序语句

SUBROUTINE BRMUL(A,B,M,N,K,C)

四、形参说明

A——双精度实型二维数组，体积为 M×N，输入参数。矩阵 A。

B——双精度实型二维数组，体积为 N×K，输入参数。矩阵 B。

M——整型变量，输入参数。矩阵 A 的行数。

N——整型变量，输入参数。矩阵 A 的列数，也是矩阵 B 的行数。

K——整型变量，输入参数。矩阵 B 的列数。

C——双精度实型二维数组，体积为 M×K，输出参数。返回乘积矩阵 $C = AB$。

五、子程序（文件名：BRMUL.FOR）

```
      SUBROUTINE BRMUL(A,B,M,N,K,C)
      DIMENSION A(M,N),B(N,K),C(M,K)
      DOUBLE PRECISION A,B,C
      DO 50 I=1,M
      DO 50 J=1,K
        C(I,J)=0.0
        DO 10 L=1,N
          C(I,J)=C(I,J)+A(I,L)*B(L,J)
10      CONTINUE
50    CONTINUE
      RETURN
      END
```

六、例

设矩阵 A 与 B 分别为

$$A = \begin{bmatrix} 1 & 3 & -2 & 0 & 4 \\ -2 & -1 & 5 & -7 & 2 \\ 0 & 8 & 4 & 1 & -5 \\ 3 & -3 & 2 & -4 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 4 & 5 & -1 \\ 2 & -2 & 6 \\ 7 & 8 & 1 \\ 0 & 3 & -5 \\ 9 & 8 & -6 \end{bmatrix}$$

求乘积矩阵 $C = AB$。其中 M=4,N=5,K=3。

主程序(文件名:BRMUL0.FOR)为

```
      DIMENSION A(4,5),B(5,3),C(4,3)
      DOUBLE PRECISION A,B,C
      DATA A/1.0,-2.0,0.0,3.0,3.0,-1.0,8.0,-3.0,-2.0,5.0,
     *        4.0,2.0,0.0,-7.0,1.0,-4.0,4.0,2.0,-5.0,1.0/
      DATA B/4.0,2.0,7.0,0.0,9.0,5.0,-2.0,8.0,
     *        3.0,8.0,-1.0,6.0,1.0,-5.0,-6.0/
      CALL BRMUL(A,B,4,5,3,C)
      WRITE(*,10) ((C(I,J),J=1,3),I=1,4)
10    FORMAT(1X,3D15.6)
      END
```

运行结果为

```
  .320000D+02    .150000D+02   -.900000D-01
  .430000D+02    .270000D+02    .240000D+02
 -.100000D+01   -.210000D+02    .770000D+02
  .290000D+02    .330000D+02   -.500000D+01
```

2.2 复矩阵相乘

一、功能

求 $m \times n$ 阶复矩阵 A 与 $n \times k$ 阶复矩阵 B 的乘积矩阵 $C = AB$。

二、方法说明

与实矩阵相乘的方法完全相同,参看 2.1 节。其中

$$A = AR + jAI, B = BR + jBI, C = AB = CR + jCI$$

两个复数相乘的方法如下:

设 $e + jf = (a + jb)(c + jd)$,令

$$p = ac, q = bd, s = (a + b)(c + d)$$

则

$$e = p - q, f = s - p - q$$

三、子程序语句

SUBROUTINE BCMUL(AR,AI,BR,BI,M,N,K,CR,CI)

四、形参说明

AR,AI——均为双精度实型二维数组,体积为 M×N,输入参数。分别存放矩阵 A 的实部与虚部。

BR,BI——均为双精度实型二维数组,体积为 N×K,输入参数。分别存放矩阵 B 的实部与虚部。

M——整型变量,输入参数。矩阵 A 的行数。

N——整型变量,输入参数。矩阵 A 的列数,也是矩阵 B 的行数。

K——整型变量,输入参数。矩阵 B 的列数 。

CR, CI——均为双精度实型二维数组，体积为 M×K，输出参数。返回乘积矩阵 $C = AB$ 的实部与虚部。

五、子程序(文件名:BCMUL.FOR)

```
      SUBROUTINE BCMUL(AR,AI,BR,BI,M,N,K,CR,CI)
      DIMENSION AR(M,N),AI(M,N),BR(N,K),BI(N,K)
      DIMENSION CR(M,K),CI(M,K)
      DOUBLE PRECISION AR,AI,BR,BI,CI,CR,P,Q,S
      DO 100 I=1,M
      DO 100 J=1,K
        CR(I,J)=0.0
        CI(I,J)=0.0
        DO 50 L=1,N
          P=AR(I,L)*BR(L,J)
          Q=AI(I,L)*BI(L,J)
          S=(AR(I,L)+AI(I,L))*(BR(L,J)+BI(L,J))
          CR(I,J)=CR(I,J)+P-Q
          CI(I,J)=CI(I,J)+S-P-Q
50      CONTINUE
100   CONTINUE
      RETURN
      END
```

六、例

设 3×4 阶矩阵 A 为

$$A = \begin{bmatrix} 1+j & 2-j & 3+2j & -2+j \\ 1-j & 5-j & 1+2j & 3+0j \\ 0-3j & 4-j & 2+2j & -1+2j \end{bmatrix}$$

4×4 阶矩阵 B 为

$$B=\begin{bmatrix}1-j & 4-j & 5+j & -2-j\\ 3+2j & 0+j & 2+0j & -1+5j\\ 6-3j & 3+2j & 1+j & 2-j\\ 2-j & -3-2j & -2+j & 1-2j\end{bmatrix}$$

即 $M=3,N=4,K=4$。且

$$AR=\begin{bmatrix}1 & 2 & 3 & -2\\ 1 & 5 & 1 & 3\\ 0 & 4 & 2 & -1\end{bmatrix},\qquad AI=\begin{bmatrix}1 & -1 & 2 & 1\\ -1 & -1 & 2 & 0\\ -3 & -1 & 2 & 2\end{bmatrix}$$

$$BR=\begin{bmatrix}1 & 4 & 5 & -2\\ 3 & 0 & 2 & -1\\ 6 & 3 & 1 & 2\\ 2 & -3 & -2 & 1\end{bmatrix},\quad BI=\begin{bmatrix}1 & -1 & 1 & -1\\ 2 & 1 & 0 & 5\\ -3 & 2 & 1 & -1\\ -1 & -2 & 1 & -2\end{bmatrix}$$

求 $C=AB$。

主程序(文件名:BCMUL0.FOR)为

```
      DIMENSION AR(3,4),AI(3,4),BR(4,4),BI(4,4),CR(3,4),CI(3,4)
      DOUBLE PRECISION AR,AI,BR,BI,CR,CI
      DATA AR/1.0,1.0,0.0,2.0,5.0,4.0,3.0,1.0,2.0,-2.0,3.0,-1.0/
      DATA AI/1.0,-1.0,-3.0,3*-1.0,3*2.0,1.0,0.0,2.0/
      DATA BR/1.0,3.0,6.0,2.0,4.0,0.0,3.0,-3.0,
     *        5.0,2.0,1.0,-2.0,-2.0,-1.0,2.0,1.0/
      DATA BI/-1.0,2.0,-3.0, 1.0,-1.0,1.0,2.0, 2.0,
     *        1.0,0.0,1.0,1.0,-1.0,5.0, 1.0, 2.0/
      CALL BCMUL(AR,AI,BR,BI,3,4,4,CR,CI)
      WRITE(*,10) ((CR(I,J),J=1,4),I=1,3)
      WRITE(*,20)
      WRITE(*,10) ((CI(I,J),J=1,4),I=1,3)
10    FORMAT(1X,4D15.6)
20    FORMAT(1X,'--------  ----------  ---------   ---')
      END
```

运行结果为

```
 .310000D+02    .190000D+02    .120000D+02    .100000D+02
 .350000D+02   -.600000D+01    .900000D+01    .400000D+01
 .290000D+02    .700000D+01    .110000D+02    .700000D+01
-----------    -------       --------       -------
 .800000D+01    .180000D+02    .500000D+01    .140000D+02
 .110000D+02    .200000D+01    .000000D+00    .240000D+02
 .130000D+02   -.200000D+01   -.180000D+02    .330000D+02
```

即乘积矩阵为

$$C = \begin{bmatrix} 31+8j & 19+18j & 12+5j & 10+14j \\ 35+11j & -6+2j & 9+0j & 4+24j \\ 29+13j & 7-2j & 11-18j & 7+33j \end{bmatrix}$$

2.3 实矩阵求逆的全选主元高斯-约当消去法

一、功能

采用全选主元高斯-约当(Gauss-Jordan)法求实矩阵的逆矩阵。

二、方法说明

高斯-约当法求实矩阵 A 的逆矩阵的步骤如下。

对于 k 从 1 到 n 作以下几步：

(1) 从第 k 行、第 k 列以下(包括第 k 行、第 k 列)的元素中选取绝对值最大的元素，并记下此元素所在的行号与列号，然后通过行交换与列交换将它交换到主元素位置上。

(2) $1/a_{kk} \Rightarrow a_{kk}$

(3) $a_{kj}a_{kk} \Rightarrow a_{kj}$，$j = 1,2,\cdots,n$；$j \neq k$

(4) $a_{ij} - a_{ik}a_{kj} \Rightarrow a_{ij}$，$i,j = 1,2,\cdots,n$；$i,j \neq k$

(5) $-a_{ik}a_{kk} \Rightarrow a_{ik}$，$i = 1,2,\cdots,n$；$i \neq k$

最后，根据在全选主元过程中所记录的行、列交换的信息进行恢复。恢复的原则如下：

在全选主元过程中，先交换的行、列后进行恢复；原来的行(列)交换用列(行)交换来恢复。

三、子程序语句

SUBROUTINE BRINV(A,N,L,IS,JS)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入兼输出参数。调用时存放原矩阵；返回逆矩阵。

N——整型变量，输入参数。存放矩阵的阶数。

L——整型变量，输出参数。若返回 L=0，说明原矩阵奇异，求逆失败；若 L≠0，则说明正常返回。

IS，JS——均为整型一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名:BRINV.FOR)

```
      SUBROUTINE BRINV(A,N,L,IS,JS)
      DIMENSION A(N,N),IS(N),JS(N)
      DOUBLE PRECISION A,T,D
      L=1
      DO 100 K=1,N
        D=0.0
        DO 10 I=K,N
        DO 10 J=K,N
          IF (ABS(A(I,J)).GT.D) THEN
```

```
          D=ABS(A(I,J))
          IS(K)=I
          JS(K)=J
        END IF
10    CONTINUE
      IF (D+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,20)
        RETURN
      END IF
20    FORMAT(1X,'ERR**NOT INV')
      DO 30 J=1,N
        T=A(K,J)
        A(K,J)=A(IS(K),J)
        A(IS(K),J)=T
30    CONTINUE
      DO 40 I=1,N
        T=A(I,K)
        A(I,K)=A(I,JS(K))
        A(I,JS(K))=T
40    CONTINUE
      A(K,K)=1/A(K,K)
      DO 50 J=1,N
        IF (J.NE.K) THEN
          A(K,J)=A(K,J)*A(K,K)
        END IF
50    CONTINUE
      DO 70 I=1,N
        IF (I.NE.K) THEN
          DO 60 J=1,N
            IF (J.NE.K) THEN
              A(I,J)=A(I,J)-A(I,K)*A(K,J)
            END IF
60        CONTINUE
        END IF
70    CONTINUE
      DO 80 I=1,N
        IF (I.NE.K) THEN
          A(I,K)=-A(I,K)*A(K,K)
        END IF
80    CONTINUE
100   CONTINUE
```

```
      DO 130 K=N,1,-1
        DO 110 J=1,N
          T=A(K,J)
          A(K,J)=A(JS(K),J)
          A(JS(K),J)=T
110     CONTINUE
        DO 120 I=1,N
          T=A(I,K)
          A(I,K)=A(I,IS(K))
          A(I,IS(K))=T
120     CONTINUE
130   CONTINUE
      RETURN
      END
```

六、例

求矩阵

$$A = \begin{bmatrix} 0.236800 & 0.247100 & 0.256800 & 1.26710 \\ 1.11610 & 0.125400 & 0.139700 & 0.149000 \\ 0.158200 & 1.16750 & 0.176800 & 0.187100 \\ 0.196800 & 0.207100 & 1.21680 & 0.227100 \end{bmatrix}$$

的逆矩阵 A^{-1}，并计算 AA^{-1} 以验证结果的正确性。

主程序(文件名:BRINV0.FOR)为

```
      DIMENSION A(4,4),B(4,4),C(4,4),IS(4),JS(4)
      DOUBLE PRECISION A,B,C
      DATA A/0.2368,1.1161,0.1582,0.1968,
     *       0.2471,0.1254,1.1675,0.2071,
     *       0.2568,0.1397,0.1768,1.2168,
     *       1.2671,0.1490,0.1871,0.2271/
      DO 5 I=1,4
      DO 5 J=1,4
5     B(I,J)=A(I,J)
      CALL BRINV(A,4,L,IS,JS)
      IF (L.NE.0) THEN
        WRITE(*,10) ((A(I,J),J=1,4),I=1,4)
        WRITE(*,*)
        CALL BRMUL(A,B,4,4,4,C)
        WRITE(*,10) ((C(I,J),J=1,4),I=1,4)
        WRITE(*,*)
      END IF
10    FORMAT(1X,4D15.6)
      END
```

运行结果为

A^{-1}

−.859208D−01	.937944D+00	−.684372D−01	−.796077D−01
−.105590D+00	−.885243D−01	.905983D+00	−.991908D−01
−.127073D+00	−.111351D+00	−.116967D+00	.878425D+00
.851606D+00	−.135456D+00	−.140183D+00	−.143807D+00

AA^{-1}

.100000D+01	.975613D−17	.721672D−17	.849405D−17
−.232799D−16	.100000D+01	−.215417D−16	−.263834D−16
−.277556D−16	.218331D−16	.100000D+01	.413217D−16
.328649D−18	.146215D−16	−.111537D−16	.100000D+01

2.4 复矩阵求逆的全选主元高斯-约当消去法

一、功能

采用全选主元高斯-约当(Gauss-Jordan)法求复矩阵的逆矩阵。

二、方法说明

基本方法同 2.3 节的方法说明。

关于复数乘除法可分别参看 15.1 节、15.2 节的方法说明。

三、子程序语句

SUBROUTINE　BCINV(AR,AI,N,L,IS,JS)

四、形参说明

AR,AI——均为双精度实型二维数组,体积为 N×N,输入兼输出参数。调用时分别存放原复矩阵的实部与虚部,返回逆矩阵的实部与虚部。

N——整型变量,输入参数。矩阵阶数。

L——整型变量,输出参数。若返回 L=0,说明矩阵奇异,求逆失败;若 L≠0,说明正常返回。

IS,JS——均为整型一维数组,长度为 N。本子程序的工作数组。

五、子程序(文件名:BCINV.FOR)

```
      SUBROUTINE BCINV(AR,AI,N,L,IS,JS)
      DIMENSION AR(N,N),AI(N,N),IS(N),JS(N)
      DOUBLE PRECISION AR,AI,D,P,T,Q,S,B
      L=1
      DO 100 K=1,N
        D=0.0
        DO 10 I=K,N
        DO 10 J=K,N
          P=AR(I,J)*AR(I,J)+AI(I,J)*AI(I,J)
          IF (P.GT.D) THEN
```

```
        D=P
        IS(K)=I
        JS(K)=J
      END IF
10    CONTINUE
      IF (D+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,20)
        RETURN
      END IF
20    FORMAT(1X,'ERR**NOT INV')
      DO 30 J=1,N
        T=AR(K,J)
        AR(K,J)=AR(IS(K),J)
        AR(IS(K),J)=T
        T=AI(K,J)
        AI(K,J)=AI(IS(K),J)
        AI(IS(K),J)=T
30    CONTINUE
      DO 40 I=1,N
        T=AR(I,K)
        AR(I,K)=AR(I,JS(K))
        AR(I,JS(K))=T
        T=AI(I,K)
        AI(I,K)=AI(I,JS(K))
        AI(I,JS(K))=T
40    CONTINUE
      AR(K,K)=AR(K,K)/D
      AI(K,K)=-AI(K,K)/D
      DO 50 J=1,N
        IF (J.NE.K) THEN
          P=AR(K,J)*AR(K,K)
          Q=AI(K,J)*AI(K,K)
          S=(AR(K,J)+AI(K,J))*(AR(K,K)+AI(K,K))
          AR(K,J)=P-Q
          AI(K,J)=S-P-Q
        END IF
50    CONTINUE
      DO 70 I=1,N
        IF (I.NE.K) THEN
          DO 60 J=1,N
            IF (J.NE.K) THEN
```

```
            P=AR(K,J)*AR(I,K)
            Q=AI(K,J)*AI(I,K)
            S=(AR(K,J)+AI(K,J))*(AR(I,K)+AI(I,K))
            T=P-Q
            B=S-P-Q
            AR(I,J)=AR(I,J)-T
            AI(I,J)=AI(I,J)-B
          END IF
60      CONTINUE
        END IF
70    CONTINUE
      DO 80 I=1,N
        IF (I.NE.K) THEN
          P=AR(I,K)*AR(K,K)
          Q=AI(I,K)*AI(K,K)
          S=(AR(I,K)+AI(I,K))*(AR(K,K)+AI(K,K))
          AR(I,K)=Q-P
          AI(I,K)=P+Q-S
        END IF
80    CONTINUE
100   CONTINUE
      DO 130 K=N,1,-1
        DO 110 J=1,N
          T=AR(K,J)
          AR(K,J)=AR(JS(K),J)
          AR(JS(K),J)=T
          T=AI(K,J)
          AI(K,J)=AI(JS(K),J)
          AI(JS(K),J)=T
110     CONTINUE
        DO 120 I=1,N
          T=AR(I,K)
          AR(I,K)=AR(I,IS(K))
          AR(I,IS(K))=T
          T=AI(I,K)
          AI(I,K)=AI(I,IS(K))
          AI(I,IS(K))=T
120     CONTINUE
130   CONTINUE
      RETURN
      END
```

六、例

设复矩阵为 $A = AR + \mathrm{i}AI$，其中

$$AR = \begin{bmatrix} 0.236800 & 0.247100 & 0.256800 & 1.26710 \\ 1.11610 & 0.125400 & 0.139700 & 0.149000 \\ 0.158200 & 1.16750 & 0.176800 & 0.187100 \\ 0.196800 & 0.207100 & 1.21680 & 0.227100 \end{bmatrix}$$

$$AI = \begin{bmatrix} 0.134500 & 0.167800 & 0.182500 & 1.11610 \\ 1.26710 & 0.201700 & 0.702400 & 0.272100 \\ -0.283600 & -1.19670 & 0.355800 & -0.207800 \\ 0.357600 & -1.23450 & 2.11850 & 0.477300 \end{bmatrix}$$

求逆矩阵 A^{-1}，并计算 AA^{-1} 以验证结果的正确性。

主程序(文件名:BCINV0.FOR)为

```
      DIMENSION AR(4,4),AI(4,4),BR(4,4)
      DIMENSION BI(4,4),CR(4,4),CI(4,4),IS(4),JS(4)
      DOUBLE PRECISION AR,AI,BR,BI,CR,CI
      DATA AR/0.2368,1.1161,0.1582,0.1968,
     *          0.2471,0.1254,1.1675,0.2071,
     *          0.2568,0.1397,0.1768,1.2168,
     *          1.2671,0.1490,0.1871,0.2271/
      DATA AI/0.1345,1.2671,-0.2836,0.3576,
     *          0.1678,0.2017,-1.1967,-1.2345,
     *          0.1825,0.7024,0.3558,2.1185,
     *          1.1161,0.2721,-0.2078,0.4773/
      DO 5 I=1,4
      DO 5 J=1,4
        BR(I,J)=AR(I,J)
        BI(I,J)=AI(I,J)
5     CONTINUE
      CALL BCINV(AR,AI,4,L,IS,JS)
      IF (L.NE.0) THEN
        WRITE(*,10) ((AR(I,J),J=1,4),I=1,4)
        WRITE(*,*)
        WRITE(*,10) ((AI(I,J),J=1,4),I=1,4)
        WRITE(*,*)
        CALL BCMUL(AR,AI,BR,BI,4,4,4,CR,CI)
        WRITE(*,10) ((CR(I,J),J=1,4),I=1,4)
        WRITE(*,*)
        WRITE(*,10) ((CI(I,J),J=1,4),I=1,4)
        WRITE(*,*)
```

```
      END IF
10    FORMAT(1X,4D15.6)
      END
```

运行结果为

A^{-1} 的实部

-.562735D-02	.485150D+00	.216563D-01	-.187479D+00
-.698275D-01	-.470249D-01	.554762D+00	-.561533D-01
-.176210D+00	-.141970D+00	.740790D-01	.261454D+00
.484182D+00	-.316556D-01	-.126789D+00	-.782208D-03

A^{-1} 的虚部

.450467D-01	-.481698D+00	-.238359D+00	.121296D+00
.116224D+00	.148688D+00	.512193D+00	-.142853D+00
.103341D+00	.114268D+00	.451289D+00	-.468983D+00
-.443286D+00	.409279D-01	-.121555D+00	.906060D-01

AA^{-1} 的实部

.100000D+01	-.346945D-16	.832667D-16	-.693889D-17
.659195D-16	.100000D+01	-.138778D-16	-.520417D-16
.902056D-16	-.693889D-16	.100000D+01	-.832667D-16
-.650521D-17	-.140404D-16	.780626D-17	.100000D+01

AA^{-1} 的虚部

-.832667D-16	-.485723D-16	-.555112D-16	-.277556D-16
-.659195D-16	-.693889D-17	.124900D-15	.312250D-16
.121431D-16	.693889D-16	.131839D-15	-.138778D-16
.997466D-17	-.275929D-16	.199493D-16	-.373237D-16

2.5 对称正定矩阵的求逆

一、功能

求 n 阶实对称正定矩阵的逆矩阵。

二、方法说明

若 A 为 n 阶对称正定矩阵，则其逆矩阵也是对称正定矩阵。

采用"变量循环重新编号法"，其计算公式如下：

$$a'_{nn} = 1/a_{11}$$

$$a'_{n,j-1} = -a_{1j}/a_{11}, j = 2,3,\cdots,n$$

$$a'_{i-1,n} = a_{i1}/a_{11}, i = 2,3,\cdots,n$$

$$a'_{i-1,j-1} = a_{ij} - a_{i1}a_{1j}/a_{11}, i,j = 2,3,\cdots,n$$

三、子程序语句

SUBROUTINE　BSSGJ(A,N,L,B)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入兼输出参数。调用时存放原矩阵 A，返回逆矩阵 A^{-1}。

N——整型变量，输入参数。矩阵阶数。

L——整型变量，输出参数。若返回 L=0，说明工作失败(如矩阵 A 非对称正定)；若 L≠0，则说明正常返回。

B——双精度实型一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名:BSSGJ.FOR)

```
      SUBROUTINE BSSGJ(A,N,L,B)
      DIMENSION A(N,N),B(N)
      DOUBLE PRECISION A,B,W,G
      L=1
      DO 100 K=1,N
        M=N-K+1
        W=A(1,1)
        IF (W+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,10)
          RETURN
        END IF
10      FORMAT(1X,'FAIL')
        DO 80 I=2,N
          G=A(I,1)
          B(I)=G/W
          IF (I.LE.M) B(I)=-B(I)
          DO 70 J=2,I
70        A(I-1,J-1)=A(I,J)+G*B(J)
80      CONTINUE
        A(N,N)=1.0/W
        DO 90 I=2,N
90      A(N,I-1)=B(I)
100     CONTINUE
        DO 110 I=1,N-1
        DO 110 J=I+1,N
110     A(I,J)=A(J,I)
        RETURN
        END
```

六、例

求对称正定矩阵

$$
A=\begin{bmatrix}5 & 7 & 6 & 5\\ 7 & 10 & 8 & 7\\ 6 & 8 & 10 & 9\\ 5 & 7 & 9 & 10\end{bmatrix}
$$

的逆矩阵 A^{-1}，并计算 AA^{-1}。

主程序(文件名:BSSGJ0.FOR)为

```
      DIMENSION A(4,4),B(4,4),C(4,4),D(4)
      DOUBLE PRECISION A,B,C,D
      DATA A/5.0,7.0,6.0,5.0,7.0,10.0,8.0,7.0,
     *       6.0,8.0,10.0,9.0,5.0,7.0,9.0,10.0/
      DO 5 I=1,4
      DO 5 J=1,4
5     B(I,J)=A(I,J)
      CALL BSSGJ(A,4,L,D)
      IF (L.NE.0) THEN
        WRITE(*,10) ((A(I,J),J=1,4),I=1,4)
        WRITE(*,*)
        CALL BRMUL(A,B,4,4,4,C)
        WRITE(*,10) ((C(I,J),J=1,4),I=1,4)
        WRITE(*,*)
      END IF
10    FORMAT(1X,4D15.6)
      END
```

运行结果为

A^{-1}

```
  .680000D+02   -.410000D+02   -.170000D+02    .100000D+02
 -.410000D+02    .250000D+02    .100000D+02   -.600000D+01
 -.170000D+02    .100000D+02    .500000D+01   -.300000D+01
  .100000D+02   -.600000D+01   -.300000D+01    .200000D+01
```

AA^{-1}

```
  .100000D+01    .603961D-13    .675016D-13    .710543D-14
  .239808D-13    .100000D+01    .381917D-13    .124345D-13
  .888178D-15   -.150990D-13    .100000D+01   -.177636D-14
  .000000D+00   -.532907D-14   -.355271D-14    .100000D+01
```

2.6 托伯利兹矩阵求逆的特兰持方法

一、功能

用特兰持(Trench)方法求托伯利兹(Toeplitz)矩阵的逆矩阵。

二、方法说明

设 $(n+1)$ 阶 T 型矩阵为

$$T^{(n)}=\begin{bmatrix} t_0 & t_1 & t_2 & \cdots & t_n \\ \tau_1 & t_0 & t_1 & \cdots & t_{n-1} \\ \tau_2 & \tau_1 & t_0 & \cdots & t_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_n & \tau_{n-1} & \tau_{n-2} & \cdots & t_0 \end{bmatrix}$$

其求逆过程如下。

取初值 $\alpha_0=t_0, c_1^{(0)}=\tau_1/t_0, r_1^{(0)}=t_1/t_0$。

1. 对于 k 从 0 到 $n-2$ 作如下工作，最后可算出 α_{n-1} 与 $c_i^{(n-1)}$、$r_i^{(n-1)}(i=1,2,\cdots,n)$

(1) $c_i^{(k+1)}=c_i^{(k)}+\dfrac{r_{k+2-i}^{(k)}}{\alpha_k}\left(\sum\limits_{j=1}^{k+1}c_{k+2-j}^{(k)}\tau_j-\tau_{k+2}\right), i=1,\cdots,k+1$

$$c_{k+2}^{(k+1)}=\frac{1}{\alpha_k}\left(\tau_{k+2}-\sum_{j=1}^{k+1}c_{k+2-j}^{(k)}\tau_j\right)$$

(2) $r_i^{(k+1)}=r_i^{(k)}+\dfrac{c_{k+2-i}^{(k)}}{\alpha_k}\left(\sum\limits_{j=1}^{k+1}r_{k+2-j}^{(k)}t_j-t_{k+2}\right), i=1,\cdots,k+1$

$$r_{k+2}^{(k+1)}=\frac{1}{\alpha_k}\left(t_{k+2}-\sum_{j=1}^{k+1}r_{k+2-j}^{(k)}t_j\right)$$

(3) $\alpha_{k+1}=t_0-\sum\limits_{j=1}^{k+2}t_jc_j^{(k+1)}$

2. 计算逆矩阵 $B^{(n)}$ 中的各元素

$$b_{11}^{(n)}=1/\alpha_{n-1}$$

$$b_{1,j+1}^{(n)}=-r_j^{(n-1)}/\alpha_{n-1}, j=1,2,\cdots,n$$

$$b_{i+1,1}^{(n)}=-c_i^{(n-1)}/\alpha_{n-1}, i=1,2,\cdots,n$$

$$b_{i+1,j+1}^{(n)}=b_{ij}^{(n)}+\frac{1}{\alpha_{n-1}}\left[c_i^{(n-1)}r_j^{(n-1)}-r_{n+1-i}^{(n-1)}c_{n+1-j}^{(n-1)}\right], i,j=1,2,\cdots,n$$

三、子程序语句

SUBROUTINE BTRCH(T0,T,TT,N,M,B,L,C,R,P)

四、形参说明

T0——双精度实型变量，输入参数。T 型矩阵对角线上的元素 t_0。

T——双精度实型一维数组，长度为N，输入参数。存放 T 型矩阵中上三角元素 $t_1,t_2,\cdots,t_n$。

TT——双精度实型一维数组，长度为N，输入参数，存放 T 型矩阵下三角元素 $\tau_1,\tau_2,\cdots,\tau_n$。

N——整型变量，输入参数。

M——整型变量，输入参数。M=N+1 为 T 型矩阵的阶数。

B——双精度实型二维数组，体积为M×M，输出参数。返回M阶 T 型矩阵的逆矩阵。

L——整型变量，输出参数。若返回 L=0，说明程序工作失败；若 L≠0，说明正常返回。

C，R，P——均为双精度实型一维数组，长度为 M。本子程序的工作数组。

五、子程序(文件名：BTRCH.FOR)

```
      SUBROUTINE BTRCH(T0,T,TT,N,M,B,L,C,R,P)
      DIMENSION T(N),TT(N),B(M,M)
      DIMENSION C(M),R(M),P(M)
      DOUBLE PRECISION T0,T,TT,B,C,R,P,A,S
      L=1
      IF (ABS(T0)+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,100)
        RETURN
      END IF
100   FORMAT(1X,' FAIL')
      A=T0
      C(1)=TT(1)/T0
      R(1)=T(1)/T0
      DO 60 K=0,N-2
        S=0.0
        DO 10 J=1,K+1
10      S=S+C(K+2-J)*TT(J)
        S=(S-TT(K+2))/A
        DO 20 I=1,K+1
20      P(I)=C(I)+S*R(K+2-I)
        C(K+2)=-S
        S=0.0
        DO 30 J=1,K+1
30      S=S+R(K+2-J)*T(J)
        S=(S-T(K+2))/A
        DO 40 I=1,K+1
          R(I)=R(I)+S*C(K+2-I)
          C(K+2-I)=P(K+2-I)
40      CONTINUE
        R(K+2)=-S
        A=0.0
        DO 50 J=1,K+2
50      A=A+T(J)*C(J)
        A=T0-A
        IF (ABS(A)+1.0.EQ.1.0) THEN
          L=0
```

```
        WRITE(*,100)
        RETURN
      END IF
60    CONTINUE
      B(1,1)=1.0/A
      DO 70 I=1,N
        B(1,I+1)=-R(I)/A
        B(I+1,1)=-C(I)/A
70    CONTINUE
      DO 80 I=1,N
      DO 80 J=1,N
        B(I+1,J+1)=B(I,J)-C(I)*B(1,J+1)
        B(I+1,J+1)=B(I+1,J+1)+C(N+1-J)*B(1,N+2-I)
80    CONTINUE
      RETURN
      END
```

六、例

设 $N+1$ 阶 T 型矩阵为

$$T^{(n)}=\begin{bmatrix}10 & 5 & 4 & 3 & 2 & 1\\ -1 & 10 & 5 & 4 & 3 & 2\\ -2 & -1 & 10 & 5 & 4 & 3\\ -3 & -2 & -1 & 10 & 5 & 4\\ -4 & -3 & -2 & -1 & 10 & 5\\ -5 & -4 & -3 & -2 & -1 & 10\end{bmatrix}$$

其中 N=5，且 T0=10，T=(5,4,3,2,1)，TT=(-1,-2,-3,-4,-5)求 $T^{(n)}$ 的逆矩阵 B，并计算 $BT^{(n)}$。

主程序(文件名:BTRCH0.FOR)为

```
      DIMENSION T(5),TT(5),B(6,6),C(6,6),A(6),R(6),P(6)
      DOUBLE PRECISION T,TT,B,T0,C,A,R,P
      DATA T/5.0,4.0,3.0,2.0,1.0/
      DATA TT/-1.0,-2.0,-3.0,-4.0,-5.0/
      T0=10.0
      CALL BTRCH(T0,T,TT,5,6,B,L,A,R,P)
      IF (L.NE.0) THEN
        WRITE(*,10) ((B(I,J),J=1,6),I=1,6)
        DO 50 I=1,6
        DO 50 J=1,6
          C(I,J)=0.0
          DO 20 K=1,J-1
20        C(I,J)=C(I,J)+B(I,K)*T(J-K)
          C(I,J)=C(I,J)+B(I,J)*T0
```

```
          DO 30 K=J+1,6
30        C(I,J)=C(I,J)+B(I,K)*TT(K-J)
50      CONTINUE
        WRITE(*,*)
        WRITE(*,10) ((C(I,J),J=1,6),I=1,6)
        WRITE(*,*)
      END IF
10    FORMAT(1X,6D13.6)
      END
```

运行结果为

$T^{(n)}$ 的逆矩阵 B

```
 .946884D-01-.469953D-01-.137211D-01-.177270D-02 .190745D-02 .380190D-02
-.427534D-02 .949232D-01-.469003D-01-.137122D-01-.181915D-02 .190745D-02
-.988143D-03-.473167D-02 .948032D-01-.469175D-01-.137122D-01-.177270D-02
 .177270D-02-.988082D-03-.474397D-02 .948032D-01-.469005D-01-.137211D-01
 .130644D-01 .209908D-02-.988082D-03-.473167D-02 .949232D-01-.469953D-01
 .469986D-01 .130644D-01 .177270D-02-.988143D-03-.427534D-02 .946884D-01
```

$BT^{(n)}$

```
 .100000D+01-.174668D-03-.688163D-02 .240309D-03-.843695D-04-.130104D-16
-.500000D-02 .100001D+01 .139531D-03-.675264D-02-.423289D-04-.372966D-16
 .970874D-02-.499818D-02 .100008D+01 .207183D-03-.689547D-02-.177809D-16
 .485723D-16 .970547D-02-.512863D-02 .100002D+01 .339164D-03 .693481D-02
 .971445D-16-.240994D-04 .874343D-02-.560017D-02 .100104D+01 .346741D-04
 .277556D-16-.866964D-04-.348283D-02 .713412D-02-.210127D-02 .100000D+01
```

2.7 求行列式值的全选主元高斯消去法

一、功能

用全选主元高斯(Gauss)消去法计算 n 阶方程 A 的行列式值。

二、方法说明

用高斯消去法对方阵 A 进行一系列行变换，使之成为上三角矩阵，其主对角线上各元素的乘积即为行列式值。其变换过程如下：

对于 $k=1,2,\cdots,n-1$ 作变换

$$a_{ij}-a_{ik}a_{kj}/a_{kk}\Rightarrow a_{ij},i,j=k+1,\cdots,n$$

为了保证数值稳定性，在变换过程中采用全选主元。

三、子程序语句

SUBROUTINE BSDET(A,N,DET)

四、形参说明

A——双精度实型二维数组，体积为N×N，输入参数。存放方阵 A 的元素，返回时将

被破坏。

N——整型变量，输入参数，方阵阶数。

DET——双精度实型变量，输出参数。返回方阵 A 的行列式值。

五、子程序(文件名:BSDET.FOR)

```
      SUBROUTINE BSDET(A,N,DET)
      DIMENSION A(N,N)
      DOUBLE PRECISION A,DET,F,D,Q
      F=1.0
      DET=1.0
      DO 100 K=1,N-1
        Q=0.0
        DO 10 I=K,N
        DO 10 J=K,N
          IF (ABS(A(I,J)).GT.Q) THEN
            Q=ABS(A(I,J))
            IS=I
            JS=J
          END IF
10      CONTINUE
        IF (Q+1.0.EQ.1.0) THEN
          DET=0.0
          RETURN
        END IF
        IF (IS.NE.K) THEN
          F=-F
          DO 20 J=K,N
            D=A(K,J)
            A(K,J)=A(IS,J)
            A(IS,J)=D
20        CONTINUE
        END IF
        IF (JS.NE.K) THEN
          F=-F
          DO 30 I=K,N
            D=A(I,JS)
            A(I,JS)=A(I,K)
            A(I,K)=D
30        CONTINUE
        END IF
        DET=DET*A(K,K)
        DO 50 I=K+1,N
```

```
      D=A(I,K)/A(K,K)
      DO 40 J=K+1,N
40    A(I,J)=A(I,J)-D*A(K,J)
50    CONTINUE
100   CONTINUE
      DET=F*DET*A(N,N)
      RETURN
      END
```

六、例

(1) 求方阵

$$A=\begin{bmatrix}1 & 2 & 3 & 4\\ 5 & 6 & 7 & 8\\ 9 & 10 & 11 & 12\\ 13 & 14 & 15 & 16\end{bmatrix}$$

的行列式值 det(A)。

主程序(文件名:BSDET0.FOR)为

```
      DIMENSION A(4,4)
      DOUBLE PRECISION A,DET
      DATA A/1.0,5.0,9.0,13.0,2.0,6.0,10.0,14.0,
     *       3.0,7.0,11.0,15.0,4.0,8.0,12.0,16.0/
      CALL BSDET(A,4,DET)
      WRITE(*,10) DET
10    FORMAT(1X,'DET=',D15.6)
      END
```

运行结果为

DET=   .000000D+00

(2) 求方阵

$$A=\begin{bmatrix}3 & -3 & -2 & 4\\ 5 & -5 & 1 & 8\\ 11 & 8 & 5 & -7\\ 5 & -1 & -3 & -1\end{bmatrix}$$

的行列式值 det(A)。

主程序(文件名:BSDET1.FOR)为

```
      DIMENSION A(4,4)
      DOUBLE PRECISION A,DET
      DATA A/3.0,5.0,11.0,5.0,-3.0,-5.0,8.0,-1.0,
     *       -2.0,1.0,5.0,-3.0,4.0,8.0,-7.0,-1.0/
      CALL BSDET(A,4,DET)
      WRITE(*,10) DET
10    FORMAT(1X,'DET=',D15.6)
```

```
      END
```

运行结果为

```
DET=     .595000D+03
```

2.8 求矩阵秩的全选主元高斯消去法

一、功能

用全选主元高斯(Gauss)消去法计算矩阵 A 的秩。

二、方法说明

设有 $m\times n$ 阶矩阵

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

取 $k=\min\{m,n\}$。对于 $r=1,2,\cdots,k$，用全选主元高斯消去法(参看 1.1 节) 将 A 变为上三角矩阵，直到某次 $a_{r+1,r+1}=0$ 为止，矩阵 A 的秩为 r。

三、子程序语句

SUBROUTINE BRANK(A,M,N,K)

四、形参说明

A——双精度实型二维数组，体积为 M×N，输入参数。返回时将被破坏。

M,N——均为整型变量，输入参数。矩阵 A 的行数与列数。

K——整型变量，输出参数。返回矩阵 A 的秩。

五、子程序(文件名:BRANK.FOR)

```
      SUBROUTINE BRANK(A,M,N,K)
      DIMENSION A(M,N)
      DOUBLE PRECISION A,D,Q
      IF (M.GE.N) THEN
        NN=N
      ELSE
        NN=M
      END IF
      K=0
      DO 100 L=1,NN
        Q=0.0
        DO 10 I=L,M
        DO 10 J=L,N
          IF (ABS(A(I,J)).GT.Q) THEN
            Q=ABS(A(I,J))
            IS=I
```

```
      JS=J
    END IF
10  CONTINUE
    IF (Q+1.0.EQ.1.0) RETURN
    K=K+1
    IF (IS.NE.L) THEN
      DO 20 J=L,N
        D=A(L,J)
        A(L,J)=A(IS,J)
        A(IS,J)=D
20    CONTINUE
    END IF
    IF (JS.NE.L) THEN
      DO 30 I=L,M
        D=A(I,JS)
        A(I,JS)=A(I,L)
        A(I,L)=D
30    CONTINUE
    END IF
    DO 50 I=L+1,N
      D=A(I,L)/A(L,L)
      DO 40 J=L+1,N
40    A(I,J)=A(I,J)-D*A(L,J)
50  CONTINUE
100 CONTINUE
    RETURN
    END
```

六、例

求矩阵

$$A=\begin{bmatrix}1&2&3&4\\5&6&7&8\\9&10&11&12\\13&14&15&16\\17&18&19&20\end{bmatrix}$$

的秩。

主程序(文件名:BRANK0.FOR)为

```
      DIMENSION A(5,4)
      DOUBLE PRECISION A
      DATA A/1.0,5.0,9.0,13.0,17.0,2.0,6.0,10.0,14.0,18.0,
     *       3.0,7.0,11.0,15.0,19.0,4.0,8.0,12.0,16.0,20.0/
      CALL BRANK(A,5,4,K)
```

```
      WRITE(*,10) K
10    FORMAT(1X,'RANK=',I3)
      END
```

运行结果为

RANK=4

2.9 对称正定矩阵的乔里斯基分解与行列式的求值

一、功能

用乔里斯基(Cholesky)分解法求对称正定矩阵的三角分解，并求行列式值。

二、方法说明

设矩阵 A 为对称正定矩阵，则存在一个实的非奇异下三角阵 L，使

$$A = LL^T$$

其中

$$L = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & 0 & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$$

乔里斯基分解的步骤为：

对于 $j = 1,2,\cdots,n$

(1) $l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2\right)^{\frac{1}{2}}$

(2) $l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk}\right)/l_{jj}, i = j+1,\cdots,n$

A 的行列式值为

$$\det(A) = \left(\prod_{k=1}^{n} l_{kk}\right)^2$$

三、子程序语句

SUBROUTINE　BCHOL(A,N,DET,L)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入兼输出参数。调用时存放对称正定矩阵；返回下三角矩阵 L。

N——整型变量，输入参数。矩阵阶数。

DET——双精度实型变量，输出参数。返回对称正定矩阵 A 的行列式值。

L——整型变量，输出参数。若返回 L=0，说明子程序工作失败(如 A 非对称正定)；若 L≠0，则说明正常返回。

五、子程序(文件名：BCHOL.FOR)

```
      SUBROUTINE BCHOL(A,N,DET,L)
```

```
      DIMENSION A(N,N)
      DOUBLE PRECISION A,DET
      L=1
      IF (A(1,1)+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,10)
        RETURN
      END IF
10    FORMAT(1X,' FAIL')
      A(1,1)=SQRT(A(1,1))
      DET=A(1,1)
      DO  20  I=2,N
20    A(I,1)=A(I,1)/A(1,1)
      DO  60  J=2,N
        DO  30  K=1,J-1
30      A(J,J)=A(J,J)-A(J,K)*A(J,K)
        IF (A(J,J)+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,10)
          RETURN
        END IF
        A(J,J)=SQRT(A(J,J))
        DET=DET*A(J,J)
        DO  50  I=J+1,N
          DO  40  K=1,J-1
40        A(I,J)=A(I,J)-A(I,K)*A(J,K)
          A(I,J)=A(I,J)/A(J,J)
50      CONTINUE
60    CONTINUE
      DET=DET*DET
      DO  70  I=1,N-1
      DO  70  J=I+1,N
70    A(I,J)=0.0
      RETURN
      END
```

六、例

对下列对称正定矩阵

$$A=\begin{bmatrix}5 & 7 & 6 & 5\\ 7 & 10 & 8 & 7\\ 6 & 8 & 10 & 9\\ 5 & 7 & 9 & 10\end{bmatrix}$$

进行乔里斯基分解，并求行列式值。

主程序(文件名:BCHOL0.FOR)为

```
      DIMENSION A(4,4)
      DOUBLE PRECISION A,DET
      DATA A/5.0,7.0,6.0,5.0,7.0,10.0,8.0,7.0,6.0,
     *       8.0,10.0,9.0,5.0,7.0,9.0,10.0/
      CALL BCHOL(A,4,DET,L)
      IF (L.NE.0) THEN
        WRITE(*,*)
        WRITE(*,10)
        WRITE(*,20) ((A(I,J),J=1,4),I=1,4)
        WRITE(*,*)
        WRITE(*,30) DET
        WRITE(*,*)
      END IF
10    FORMAT(1X,'MAT L IS:')
20    FORMAT(1X,4D15.6)
30    FORMAT(1X,'DET=',D15.6)
      END
```

运行结果为:

```
MAT L IS:
  .223607D+01    .000000D+00    .000000D+00    .000000D+00
  .313050D+01    .447214D+00    .000000D+00    .000000D+00
  .268328D+01   -.894427D+00    .141421D+01    .000000D+00
  .223607D+01   -.632240D-15    .212132D+01    .707107D+00

DET=    .100000D+01
```

2.10 矩阵的三角分解

一、功能

对 n 阶实矩阵进行 LU 分解，即 $A = LU$。其中

$$L = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & 0 \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & 0 & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

二、方法说明

令

$$Q = L + U - I_n = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1k} & \cdots & u_{1n} \\ l_{21} & u_{22} & \cdots & u_{2k} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ l_{k1} & l_{k2} & \cdots & u_{kk} & \cdots & u_{kn} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nk} & \cdots & u_{nn} \end{bmatrix}$$

则 LU 分解的问题变为求 Q 矩阵的问题。

将 n 阶实矩阵 A 变为矩阵 Q 的计算步骤为：

对于 $k = 1,2,\cdots,n-1$

(1) $a_{ik}/a_{kk} \Rightarrow a_{ik}, i = k+1,\cdots,n$

(2) $a_{ij} - a_{ik}a_{kj} \Rightarrow a_{ij}, i,j = k+1,\cdots,n$

由于本方法没有选主元，因此其数值计算是不稳定的。

三、子程序语句

SUBROUTINE BLLUU(A,N,L,U,JT)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入兼输出参数。调用时存放实矩阵 A；返回 Q 矩阵。

N——整型变量，输入参数。矩阵阶数。

L——双精度实型二维数组，体积为 N×N，输出参数。返回单位下三角矩阵 L。

U——双精度实型二维数组，体积为 N×N，输出参数。返回上三角矩阵 U。

JT——整型变量，输出参数。若返回 JT=0，则说明程序工作失败（如 $|a_{kk}| \approx 0$）；若 JT≠0，则说明正常返回。

五、子程序（文件名：BLLUU.FOR）

```
      SUBROUTINE BLLUU(A,N,L,U,JT)
      DIMENSION A(N,N),L(N,N),U(N,N)
      DOUBLE PRECISION A,L,U
      DO 100 K=1,N-1
        IF (ABS(A(K,K))+1.0.EQ.1.0) THEN
          WRITE(*,*)'* * *FAIL* * *'
          JT=0
        END IF
        DO 10 I=K+1,N
10      A(I,K)=A(I,K)/A(K,K)
        DO 20 I=K+1,N
        DO 20 J=K+1,N
20      A(I,J)=A(I,J)-A(I,K)*A(K,J)
100   CONTINUE
      DO 200 I=1,N
        DO 30 J=1,I-1
```

```
          L(I,J)=A(I,J)
          U(I,J)=0.0
30      CONTINUE
        L(I,I)=1.0
        U(I,I)=A(I,I)
        DO 40 J=I+1,N
          L(I,J)=0.0
          U(I,J)=A(I,J)
40      CONTINUE
200   CONTINUE
      JT=1
      END
```

六、例

对下列矩阵

$$A = \begin{bmatrix} 2 & 4 & 4 & 2 \\ 3 & 3 & 12 & 6 \\ 2 & 4 & -1 & 2 \\ 4 & 2 & 1 & 1 \end{bmatrix}$$

进行 LU 分解。

主程序(文件名:BLLUU0.FOR)为

```
      DIMENSION A(4,4),L(4,4),U(4,4)
      DOUBLE PRECISION A,L,U
      DATA A/2.0,3.0,2.0,4.0,4.0,3.0,4.0,2.0,
     *        4.0,12.0,-1.0,1.0,2.0,6.0,2.0,1.0/
      CALL BLLUU(A,4,L,U,K)
      IF (K.NE.0) THEN
        WRITE(*,*)
        WRITE(*,*) 'MAT  L IS :'
        WRITE(*,100) ((L(I,J),J=1,4),I=1,4)
        WRITE(*,*)
        WRITE(*,*) 'MAT  U IS :'
        WRITE(*,100) ((U(I,J),J=1,4),I=1,4)
        WRITE(*,*)
      END IF
100   FORMAT(1X,4D15.6)
      END
```

运行结果为

```
MAT L IS :
 .100000D+01   .000000D+00   .000000D+00   .000000D+00
 .150000D+01   .100000D+01   .000000D+00   .000000D+00
```

.100000D+01	.000000D+00	.100000D+01	.000000D+00
.200000D+01	.200000D+01	.380000D+01	.100000D+01

MAT U IS:

.200000D+01	.400000D+01	.400000D+01	.200000D+01
.000000D+00	-.300000D+01	.600000D+01	.300000D+01
.000000D+00	.000000D+00	-.500000D+01	.000000D+00
.000000D+00	.000000D+00	.000000D+00	-.900000D+01

2.11 一般实矩阵的 QR 分解

一、功能

用豪斯荷尔德(Householder)变换对一般实矩阵进行 QR 分解。

二、方法说明

设 $m \times n$ 阶实矩阵 A 列线性无关，则可以分解为

$$A = QR$$

的形式，其中 Q 为 $m \times m$ 阶的正交矩阵，R 为 $m \times n$ 阶的上三角矩阵。

豪斯荷尔德方法对 A 进行 QR 分解的步骤如下。

令 $s = \min\{m-1, n\}$，$Q = I_{m\times m}$。

对于 $k = 1, 2, \cdots, s$，作如下几步：

(1) 确定豪斯荷尔德变换

$$Q_k = \begin{bmatrix} I_{k-1} & O \\ O & \tilde{Q}_{m-k+1} \end{bmatrix}$$

其中

$$\tilde{Q}_{m-k+1} = \begin{bmatrix} (1-2u_k^2) & -2u_ku_{k+1} & \cdots & -2u_ku_m \\ -2u_{k+1}u_k & (1-2u_{k+1}^2) & \cdots & -2u_{k+1}u_m \\ \vdots & \vdots & & \vdots \\ -2u_mu_k & -2u_mu_{k+1} & \cdots & (1-2u_m^2) \end{bmatrix}$$

矩阵中 $u_j (j = k, k+1, \cdots, m)$ 的计算公式如下：

$$\eta = \max_{k\leqslant i\leqslant m}(|a_{ik}|)$$

$$\alpha = -\operatorname{sign}(a_{kk})\sqrt{\sum_{i=k}^{m}(a_{ik}/\eta)^2}\cdot\eta$$

$$\rho = \sqrt{2\alpha(\alpha - a_{kk})}$$

$$u_k = \frac{1}{\rho}(a_{kk} - \alpha) \Rightarrow a_{kk}$$

$$u_i = \frac{1}{\rho}a_{ik} \Rightarrow a_{ik},\ i = k+1, \cdots, m$$

(2) 用 Q_k 左乘 Q，即

$$Q_k Q \Rightarrow Q$$

其计算公式如下：

$$\left.\begin{array}{l} t = \sum_{l=k}^{m} a_{lk} q_{lj} \\ q_{ij} - 2a_{ik} t \Rightarrow q_{ij} \end{array}\right\} \begin{array}{l} j = 1,2\cdots,m \\ i = k,k+1,\cdots,m \end{array}$$

(3) 用 Q_k 左乘 A，即

$$Q_k A \Rightarrow A$$

其计算公式如下：

$$\left.\begin{array}{l} t = \sum_{l=k}^{m} a_{lk} a_{lj} \\ a_{ij} - 2a_{ik} t \Rightarrow a_{ij} \end{array}\right\} \begin{array}{l} j = k+1,\cdots,m \\ i = k,\cdots,m \end{array}$$

最后，矩阵 A 的右上三角部分即为 R，矩阵 Q^T 为正交矩阵。

三、子程序语句

SUBROUTINE　BMAQR(A,M,N,Q,L)

四、形参说明

A——双精度实型二维数组，体积为 M×N，输入兼输出参数。调用时存放实矩阵 A；返回 R 矩阵。

M——整型变量，输入参数。矩阵 A 的行数。

N——整型变量，输入参数。矩阵 A 的列数。

Q——双精度实型二维数组，体积为 M×M，输出参数。返回矩阵 Q。

L——整型变量，输出参数。若返回 L=0，则说明矩阵 A 列线性相关，程序工作失败；若 L≠0，说明正常返回。

五、子程序(文件名:BMAQR.FOR)

```
      SUBROUTINE BMAQR(A,M,N,Q,L)
      DIMENSION A(M,N),Q(M,M)
      DOUBLE PRECISION A,Q,ALPHA,T,U
      IF (M.LT.N) THEN
        L=0
        WRITE(*,40)
        RETURN
      END IF
40    FORMAT(1X,' FAIL')
      DO 10 I=1,M
      DO 10 J=1,M
        Q(I,J)=0.0
        IF (I.EQ.J) Q(I,J)=1.0
10    CONTINUE
      NN=N
      IF (M.EQ.N) NN=M-1
```

```
      DO 200 K=1,NN
        U=0.0
        DO 20 I=K,M
          IF (ABS(A(I,K)).GT.U) U=ABS(A(I,K))
 20     CONTINUE
        ALPHA=0.0
        DO 30 I=K,M
          T=A(I,K)/U
          ALPHA=ALPHA+T*T
 30     CONTINUE
        IF (A(K,K).GT.0.0) U=-U
        ALPHA=U*SQRT(ALPHA)
        IF (ABS(ALPHA)+1.0.EQ.1.0) THEN
          L=0
          WRITE(*,40)
          RETURN
        END IF
        U=SQRT(2.0*ALPHA*(ALPHA-A(K,K)))
        IF (U+1.0.NE.1.0) THEN
          A(K,K)=(A(K,K)-ALPHA)/U
          DO 50 I=K+1,M
 50       A(I,K)=A(I,K)/U
          DO 80 J=1,M
            T=0.0
            DO 60 L=K,M
 60         T=T+A(L,K)*Q(L,J)
            DO 70 I=K,M
 70         Q(I,J)=Q(I,J)-2.0*T*A(I,K)
 80       CONTINUE
          DO 110 J=K+1,N
            T=0.0
            DO 90 L=K,M
 90         T=T+A(L,K)*A(L,J)
            DO 100 I=K,M
 100        A(I,J)=A(I,J)-2.0*T*A(I,K)
 110      CONTINUE
          A(K,K)=ALPHA
          DO 120 I=K+1,M
 120      A(I,K)=0.0
        END IF
 200  CONTINUE
      L=1
```

```
      DO  210  I=1,M-1
      DO  210  J=I+1,M
        T=Q(I,J)
        Q(I,J)=Q(J,I)
        Q(J,I)=T
210   CONTINUE
      RETURN
      END
```

六、例

对矩阵

$$A=\begin{bmatrix}1 & 1 & -1\\ 2 & 1 & 0\\ 1 & -1 & 0\\ -1 & 2 & 1\end{bmatrix}$$

进行 QR 分解。

主程序(文件名:BMAQR0.FOR)为

```
      DIMENSION A(4,3),Q(4,4)
      DOUBLE PRECISION A,Q
      DATA A/1.0,2.0,1.0,-1.0,1.0,1.0,-1.0,2.0,-1.0,0.0,0.0,1.0/
      M=4
      N=3
      CALL BMAQR(A,M,N,Q,L)
      IF (L.NE.0) THEN
        WRITE(*,*)
        WRITE(*,10)
        WRITE(*,20) ((Q(I,J),J=1,M),I=1,M)
        WRITE(*,*)
        WRITE(*,30)
        WRITE(*,40) ((A(I,J),J=1,N),I=1,M)
        WRITE(*,*)
      END IF
10    FORMAT(1X,'MAT Q IS:')
20    FORMAT(1X,4D15.6)
30    FORMAT(1X,'MAT R IS:')
40    FORMAT(1X,3D15.6)
      END
```

运行结果为

```
MAT  Q  IS:
 -.377964D+00  -.377964D+00   .755929D+00   .377964D+00
 -.755929D+00  -.377964D+00  -.377964D+00  -.377964D+00
```

-.377964D+00	.377964D+00	-.377964D+00	.755929D+00
.377964D+00	-.755929D+00	-.377964D+00	.377964D+00

MAT R IS:

-.264575D+01	-.195590D-15	.755929D+00
.000000D+00	-.264575D+01	-.377964D+00
.000000D+00	.000000D+00	-.113389D+01
.000000D+00	.000000D+00	.000000D+00

2.12 一般实矩阵的奇异值分解

一、功能

利用豪斯荷尔德(Householder)变换及变形 QR 算法对一般实矩阵进行奇异值分解。

二、方法说明

设 A 为 $m\times n$ 阶的实矩阵，则存在一个 $m\times m$ 阶的列正交矩阵 U 和 $n\times n$ 阶的列正交矩阵 V，使

$$A=U\begin{bmatrix}\Sigma & 0\\ 0 & 0\end{bmatrix}V^T$$

成立。其中 $\Sigma=\mathrm{diag}\,(\sigma_1,\sigma_2,\cdots,\sigma_r)(r\leqslant\min\{m,n\})$，且 $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_r>0$。

上式称为实矩阵 A 的奇异值分解式，$\sigma_i(i=1,2,\cdots,r)$ 称为 A 的奇异值。

利用 A 的奇异值分解式，可以计算 A 的广义逆 A^+。

设 $U=(U_1,U_2)$，其中 U_1 为 U 中前 r 列列正交向量组构成的 $m\times r$ 阶矩阵；$V=(V_1,V_2)$，其中 V_1 为 V 中前 r 列列正交向量组构成的 $n\times r$ 阶矩阵。则 A 的广义逆为

$$A^+=V_1\Sigma^{-1}U_1^T$$

利用 A 的广义逆可以求解线性最小二乘问题

$$AX=B$$

其中 A 为 $m\times n$ 阶实矩阵，X 为 n 维列向量，B 为 m 维常数列向量。该问题的最小二乘解为

$$X=A^+B=V_1\Sigma^{-1}U_1^TB$$

奇异值分解的计算过程分两大步。

第一步　用豪斯荷尔德变换将 A 约化为双对角线矩阵，即

$$B=\tilde{U}^TA\tilde{V}=\begin{bmatrix}s_1 & e_1 & & & \\ & s_2 & e_2 & 0 & \\ & & \ddots & \ddots & \\ & 0 & & s_{p-1} & e_{p-1}\\ & & & & s_p\end{bmatrix}$$

其中

$$\tilde{U}=U_1U_2\cdots U_k,k=\min\{n,m-1\}$$
$$\tilde{V}=V_1V_2\cdots V_l,l=\min\{m,n-2\}$$

$\bar{U}$ 中的每一个变换 $U_j(j=1,2,\cdots,k)$将 A 中第 j 列主对角线以下的元素变为零；而 $\bar{V}$ 中的每一个向量 $V_j(j=1,2,\cdots,l)$将 A 中第 j 行中与主对角线紧邻的右次对角线元素右边的元素变为零。

对于每一个变换 V_j 具有如下形式：

$$I-\rho V_jV_j^T$$

其中 ρ 为一个比例因子，以避免计算过程中的溢出现象与误差的积累，V_j 是一个列向量

$$V_j=(v_1,v_2,\cdots,v_n)$$

则

$$AV_j=A-\rho AV_jV_j^T=A-WV_j^T$$

其中

$$W=\rho AV_j=\rho(\sum_{i=1}^{n}v_ia_{1i},\sum_{i=1}^{n}v_ia_{2i},\cdots,\sum_{i=1}^{n}v_ia_{mi})^T$$

第二步　用变形的 QR 算法进行迭代，计算所有的奇异值。

由第一步已经得到了一个双对角线矩阵 B。在这一步中，用一系列的平面旋转变换将 B 逐步变成对角矩阵。在每一次的迭代中，用变换

$$B'=U_{p-1,p}^T\cdots U_{23}^TU_{12}^TBV_{12}V_{23}\cdots V_{m-1,m}$$

其中变换 $U_{j,j+1}^T$ 将 B 中第 j 列主对角线下的一个非零元素变为零，同时在第 j 行的次对角线元素的右边出现一个非零元素；而变换 $V_{j,j+1}$ 将第 $j-1$ 行的次对角线元素右边的一个非零元素变为零，同时在第 j 列的主对角线元素的下方出现一个非零元素。由此可知，经过一次迭代($j=1,2,\cdots,p-1$)后，B' 仍为双对角线矩阵。但随着迭代的进行，最后收敛为对角矩阵，其对角线上的元素即为奇异值。

在每次迭代时，其初始变换 V_{12} 后，将在第1列的主对角线下方出现一个非零元素，在变换 V_{12} 中，选择位移值 μ 的计算式如下：

$$b=[(s_{p-1}+s_p)(s_{p-1}-s_p)+e_{p-1}^2]/2$$

$$c=(s_pe_{p-1})^2$$

$$d=\mathrm{sign}(b)\sqrt{b^2+c}$$

$$\mu=s_p^2-c/(b+d)$$

最后需要对奇异值按非递增次序进行排序。

在上述变换过程中，若对于某个次对角线元素 e_j 满足

$$|e_j|\leqslant\varepsilon(|s_{j+1}|+|s_j|)$$

则可认为 e_j 为零。

若对角线元素 s_j 满足

$$|s_j|\leqslant\varepsilon(|e_{j-1}|+|e_j|)$$

则可认为 s_j 为零(即为零奇异值)。

其中 ε 为预先给定的精度要求。

三、子程序语句

SUBROUTINE　BMUAV(A,M,N,U,V,L,EPS,KA,S,E,WORK)

四、形参说明

A——双精度实型二维数组，体积为M×N，输入兼输出参数。调用时存放实矩阵A；返回奇异值(以非递增次序存放在对角线上)。

M——整型变量，输入参数。矩阵A的行数。

N——整型变量，输入参数。矩阵A的列数。

U——双精度实型二维数组，体积为M×M，输出参数。返回左奇异向量U。

V——双精度实型二维数组，体积为N×N，输出参数。返回右奇异向量V。

L——整型变量，输出参数。若返回L=0，则说明正常返回；若L≠0，则说明工作失败。

EPS——实型变量，输入参数。控制精度要求。

KA——整型变量，输入参数。KA=max $\{M,N\}+1$。

S,E,WORK——均为双精度实型一维数组，长度为KA。本子程序的工作数组。

五、子程序(文件名:BMUAV.FOR)

```
      SUBROUTINE BMUAV(A,M,N,U,V,L,EPS,KA,S,E,WORK)
      DIMENSION A(M,N),U(M,M),V(N,N),S(KA),E(KA),WORK(KA)
      DOUBLE PRECISION A,U,V,S,E,D,WORK,DD,F,G,CS,SN,
     *                 SHH,SK,EK,B,C,SM,SM1,EM1
      IT=60
      K=N
      IF (M-1.LT.N) K=M-1
      L=M
      IF (N-2.LT.M) L=N-2
      IF (L.LT.0) L=0
      LL=K
      IF (L.GT.K) LL=L
      IF (LL.GE.1) THEN
        DO 150 KK=1,LL
          IF (KK.LE.K) THEN
            D=0.0
            DO 10 I=KK,M
10          D=D+A(I,KK)*A(I,KK)
            S(KK)=SQRT(D)
            IF (S(KK).NE.0.0) THEN
              IF (A(KK,KK).NE.0.0) S(KK)=SIGN(S(KK),A(KK,KK))
              DO 20 I=KK,M
20            A(I,KK)=A(I,KK)/S(KK)
              A(KK,KK)=1.0+A(KK,KK)
            END IF
            S(KK)=-S(KK)
          END IF
```

```
         IF (N.GE.KK+1) THEN
           DO  50  J=KK+1,N
             IF ((KK.LE.K).AND.(S(KK).NE.0.0)) THEN
               D=0.0
               DO  30  I=KK,M
30             D=D+A(I,KK) * A(I,J)
               D=-D/A(KK,KK)
               DO  40  I=KK,M
40             A(I,J)=A(I,J)+D * A(I,KK)
             END IF
             E(J)=A(KK,J)
50         CONTINUE
         END IF
         IF (KK.LE.K) THEN
           DO  60  I=KK,M
60         U(I,KK)=A(I,KK)
         END IF
         IF (KK.LE.L) THEN
           D=0.0
           DO  70  I=KK+1,N
70         D=D+E(I) * E(I)
           E(KK)=SQRT(D)
           IF (E(KK).NE.0.0) THEN
             IF (E(KK+1).NE.0.0) E(KK)=SIGN(E(KK),E(KK+1))
             DO  80  I=KK+1,N
80           E(I)=E(I)/E(KK)
             E(KK+1)=1.0+E(KK+1)
           END IF
           E(KK)=-E(KK)
           IF ((KK+1.LE.M).AND.(E(KK).NE.0.0)) THEN
             DO  90  I=KK+1,M
90           WORK(I)=0.0
             DO  110  J=KK+1,N
               DO  100  I=KK+1,M
100            WORK(I)=WORK(I)+E(J) * A(I,J)
110          CONTINUE
             DO  130  J=KK+1,N
               DO  120  I=KK+1,M
120            A(I,J)=A(I,J)-WORK(I) * E(J)/E(KK+1)
130          CONTINUE
           END IF
           DO  140  I=KK+1,N
```

```
140         V(I,KK)=E(I)
          END IF
150     CONTINUE
      END IF
      MM=N
      IF (M+1.LT.N) MM=M+1
      IF (K.LT.N) S(K+1)=A(K+1,K+1)
      IF (M.LT.MM) S(MM)=0.0
      IF (L+1.LT.MM) E(L+1)=A(L+1,MM)
      E(MM)=0.0
      NN=M
      IF (M.GT.N) NN=N
      IF (NN.GE.K+1) THEN
        DO 190 J=K+1,NN
          DO 180 I=1,M
180       U(I,J)=0.0
          U(J,J)=1.0
190     CONTINUE
      END IF
      IF (K.GE.1) THEN
        DO 250 LL=1,K
          KK=K-LL+1
          IF (S(KK).NE.0.0) THEN
            IF (NN.GE.KK+1) THEN
              DO 220 J=KK+1,NN
                D=0.0
                DO 200 I=KK,M
200             D=D+U(I,KK)*U(I,J)/U(KK,KK)
                D=-D
                DO 210 I=KK,M
210             U(I,J)=U(I,J)+D*U(I,KK)
220           CONTINUE
            END IF
            DO 225 I=KK,M
225         U(I,KK)=-U(I,KK)
            U(KK,KK)=1.0+U(KK,KK)
            IF (KK-1.GE.1) THEN
              DO 230 I=1,KK-1
230           U(I,KK)=0.0
            END IF
          ELSE
            DO 240 I=1,M
```

```
240          U(I,KK)=0.0
             U(KK,KK)=1.0
           END IF
250      CONTINUE
       END IF
       DO  300  LL=1,N
         KK=N-LL+1
         IF ((KK.LE.L).AND.(E(KK).NE.0.0)) THEN
           DO  280  J=KK+1,N
             D=0.0
             DO  260  I=KK+1,N
260          D=D+V(I,KK)*V(I,J)/V(KK+1,KK)
             D=-D
             DO  270  I=KK+1,N
270          V(I,J)=V(I,J)+D*V(I,KK)
280        CONTINUE
         END IF
         DO  290  I=1,N
290      V(I,KK)=0.0
         V(KK,KK)=1.0
300    CONTINUE
       DO  305  I=1,M
       DO  305  J=1,N
305    A(I,J)=0.0
       M1=MM
       IT=60
310    IF (MM.EQ.0) THEN
         L=0
         IF (M.GE.N) THEN
           I=N
         ELSE
           I=M
         END IF
         DO  315  J=1,I-1
           A(J,J)=S(J)
           A(J,J+1)=E(J)
315      CONTINUE
         A(I,I)=S(I)
         IF (M.LT.N) A(I,I+1)=E(I)
         DO  314  I=1,N-1
           DO  313  J=I+1,N
             D=V(I,J)
```

```
            V(I,J)=V(J,I)
            V(J,I)=D
313       CONTINUE
314     CONTINUE
        RETURN
      END IF
      IF (IT.EQ.0) THEN
        L=MM
        IF (M.GE.N) THEN
          I=N
        ELSE
          I=M
        END IF
        DO 316 J=1,I-1
          A(J,J)=S(J)
          A(J,J+1)=E(J)
316     CONTINUE
        A(I,I)=S(I)
        IF (M.LT.N) A(I,I+1)=E(I)
        DO 318 I=1,N-1
          DO 317 J=I+1,N
            D=V(I,J)
            V(I,J)=V(J,I)
            V(J,I)=D
317       CONTINUE
318     CONTINUE
        RETURN
      END IF
      KK=MM
320   KK=KK-1
      IF (KK.NE.0) THEN
        D=ABS(S(KK))+ABS(S(KK+1))
        DD=ABS(E(KK))
        IF (DD.GT.EPS*D) GOTO 320
        E(KK)=0.0
      END IF
      IF (KK.EQ.MM-1) THEN
        KK=KK+1
        IF (S(KK).LT.0.0) THEN
          S(KK)=-S(KK)
          DO 330 I=1,N
330       V(I,KK)=-V(I,KK)
```

```
         END IF
 335     IF (KK.NE.M1) THEN
           IF (S(KK).LT.S(KK+1)) THEN
             D=S(KK)
             S(KK)=S(KK+1)
             S(KK+1)=D
             IF (KK.LT.N) THEN
               DO 340 I=1,N
                 D=V(I,KK)
                 V(I,KK)=V(I,KK+1)
                 V(I,KK+1)=D
 340           CONTINUE
             END IF
             IF (KK.LT.M) THEN
               DO 350 I=1,M
                 D=U(I,KK)
                 U(I,KK)=U(I,KK+1)
                 U(I,KK+1)=D
 350           CONTINUE
             END IF
             KK=KK+1
             GOTO 335
           END IF
         END IF
         IT=60
         MM=MM-1
         GOTO 310
       END IF
       KS=MM+1
 360   KS=KS-1
       IF (KS.GT.KK) THEN
         D=0.0
         IF (KS.NE.MM) D=D+ABS(E(KS))
         IF (KS.NE.KK+1) D=D+ABS(E(KS-1))
         DD=ABS(S(KS))
         IF (DD.GT.EPS*D) GOTO 360
         S(KS)=0.0
       END IF
       IF (KS.EQ.KK) THEN
         KK=KK+1
         D=ABS(S(MM))
         IF (ABS(S(MM-1)).GT.D) D=ABS(S(MM-1))
```

```
      IF (ABS(E(MM-1)).GT.D) D=ABS(E(MM-1))
      IF (ABS(S(KK)).GT.D) D=ABS(S(KK))
      IF (ABS(E(KK)).GT.D) D=ABS(E(KK))
      SM=S(MM)/D
      SM1=S(MM-1)/D
      EM1=E(MM-1)/D
      SK=S(KK)/D
      EK=E(KK)/D
      B=((SM1+SM)*(SM1-SM)+EM1*EM1)/2.0
      C=SM*EM1
      C=C*C
      SHH=0.0
      IF ((B.NE.0.0).OR.(C.NE.0.0)) THEN
        SHH=SQRT(B*B+C)
        IF (B.LT.0.0) SHH=-SHH
        SHH=C/(B+SHH)
      END IF
      F=(SK+SM)*(SK-SM)-SHH
      G=SK*EK
      DO 400 I=KK,MM-1
        CALL SSS(F,G,CS,SN)
        IF (I.NE.KK) E(I-1)=F
        F=CS*S(I)+SN*E(I)
        E(I)=CS*E(I)-SN*S(I)
        G=SN*S(I+1)
        S(I+1)=CS*S(I+1)
        IF ((CS.NE.1.0).OR.(SN.NE.0.0)) THEN
          DO 370 J=1,N
            D=CS*V(J,I)+SN*V(J,I+1)
            V(J,I+1)=-SN*V(J,I)+CS*V(J,I+1)
            V(J,I)=D
370       CONTINUE
        END IF
        CALL SSS(F,G,CS,SN)
        S(I)=F
        F=CS*E(I)+SN*S(I+1)
        S(I+1)=-SN*E(I)+CS*S(I+1)
        G=SN*E(I+1)
        E(I+1)=CS*E(I+1)
        IF (I.LT.M) THEN
          IF ((CS.NE.1.0).OR.(SN.NE.0.0)) THEN
            DO 380 J=1,M
```

```
              D=CS*U(J,I)+SN*U(J,I+1)
              U(J,I+1)=-SN*U(J,I)+CS*U(J,I+1)
              U(J,I)=D
380         CONTINUE
          END IF
        END IF
400   CONTINUE
      E(MM-1)=F
      IT=IT-1
      GOTO 310
    END IF
    IF (KS.EQ.MM) THEN
      KK=KK+1
      F=E(MM-1)
      E(MM-1)=0.0
      DO 420 LL=KK,MM-1
        I=MM+KK-LL-1
        G=S(I)
        CALL SSS(G,F,CS,SN)
        S(I)=G
        IF (I.NE.KK) THEN
          F=-SN*E(I-1)
          E(I-1)=CS*E(I-1)
        END IF
        IF ((CS.NE.1.0).OR.(SN.NE.0.0)) THEN
          DO 410 J=1,N
            D=CS*V(J,I)+SN*V(J,MM)
            V(J,MM)=-SN*V(J,I)+CS*V(J,MM)
            V(J,I)=D
410       CONTINUE
        END IF
420   CONTINUE
      GOTO 310
    END IF
    KK=KS+1
    F=E(KK-1)
    E(KK-1)=0.0
    DO 450 I=KK,MM
      G=S(I)
      CALL SSS(G,F,CS,SN)
      S(I)=G
      F=-SN*E(I)
```

```
      E(1)=CS*E(1)
      IF ((CS.NE.1.0).OR.(SN.NE.0.0)) THEN
        DO 430 J=1,M
          D=CS*U(J,I)+SN*U(J,KK-1)
          U(J,KK-1)=-SN*U(J,I)+CS*U(J,KK-1)
          U(J,I)=D
430     CONTINUE
      END IF
450   CONTINUE
      GOTO 310
      END
      SUBROUTINE SSS(F,G,CS,SN)
      DOUBLE PRECISION F,G,CS,SN,D,R
      IF ((ABS(F)+ABS(G)).EQ.0.0) THEN
        CS=1.0
        SN=0.0
        D=0.0
      ELSE
        D=SQRT(F*F+G*G)
        IF (ABS(F).GT.ABS(G)) D=SIGN(D,F)
        IF (ABS(G).GE.ABS(F)) D=SIGN(D,G)
        CS=F/D
        SN=G/D
      END IF
      R=1.0
      IF (ABS(F).GT.ABS(G)) THEN
        R=SN
      ELSE
        IF (CS.NE.0.0) R=1.0/CS
      END IF
      F=D
      G=R
      RETURN
      END
```

六、例

(1) 求下列实矩阵

$$A = \begin{bmatrix} 1 & 1 & -1 \\ 2 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 2 & 1 \end{bmatrix}$$

的奇异值分解式，并验证结果的正确性。

主程序(文件名:BMUAV0.FOR)为

```
      DIMENSION A(4,3),U(4,4),V(3,3),C(4,3),S(5),E(5),WORK(5)
      DOUBLE PRECISION A,U,V,C,S,E,WORK
      DATA A/1.0,2.0,1.0,-1.0,2*1.0,-1.0,2.0,-1.0,2*0.0,1.0/
      M=4
      N=3
      KA=5
      EPS=0.000001
      CALL BMUAV(A,M,N,U,V,L,EPS,KA,S,E,WORK)
      WRITE(*,*)
      WRITE(*,10) L
10    FORMAT(1X,'L=',I3)
      WRITE(*,*)
      WRITE(*,20)
20    FORMAT(1X,'MAT U:')
      WRITE(*,30) ((U(I,J),J=1,M),I=1,M)
30    FORMAT(1X,4D15.6)
      WRITE(*,*)
      WRITE(*,40)
40    FORMAT(1X,'MAT A:')
      WRITE(*,50) ((A(I,J),J=1,N),I=1,M)
50    FORMAT(1X,3D15.6)
      WRITE(*,*)
      WRITE(*,60)
60    FORMAT(1X,'MAT V:')
      WRITE(*,70) ((V(I,J),J=1,N),I=1,N)
70    FORMAT(1X,3D15.6)
      WRITE(*,*)
      CALL BRMUL(U,A,4,4,3,C)
      CALL BRMUL(C,V,4,3,3,A)
      WRITE(*,80)
80    FORMAT(1X,'MAT UAV:')
      WRITE(*,50) ((A(I,J),J=1,N),I=1,M)
      WRITE(*,*)
      END
```

运行结果为

```
L=  0

MAT U:
  .276393D+00    .507093D+00    .723607D+00    .000000D+00
  .447214D+00    .676123D+00   -.447214D+00    .000000D+00
  .447214D+00   -.169031D+00   -.447214D+00    .000000D+00
```

```
 -.723607D+00    .507093D+00   -.276393D+00    .000000D+00
 MAT  A:
  .280252D+01    .000000D+00    .000000D+00
  .000000D+00    .264575D+01    .000000D+00
  .000000D+00    .000000D+00    .107047D+01
  .000000D+00    .000000D+00    .000000D+00
 MAT  V:
  .835549D+00   -.417775D+00   -.356822D+00
  .447214D+00    .894427D+00    .443009D-14
 -.319151D+00    .159576D+00   -.934172D+00
 MAT  UAV:
  .100000D+01    .100000D+01   -.100000D+01
  .200000D+01    .100000D+01    .165883D-15
  .100000D+01    .100000D+01    .165883D-15
 -.100000D+01    .200000D+01    .100000D+01
```

(2) 求下列实矩阵

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 1 & 0 & 2 \\ 1 & -1 & 0 & 1 \end{bmatrix}$$

的奇异值分解式；并验证结果的正确性。

主程序(文件名：BMUAV1.FOR)为

```
      DIMENSION A(3,4),U(3,3),V(4,4),C(3,4),S(5),E(5),WORK(5)
      DOUBLE PRECISION A,U,V,C,S,E,WORK
      DATA A/1.0,2.0,3*1.0,2*-1.0,2*0.0,-1.0,2.0,1.0/
      M=3
      N=4
      KA=5
      EPS=0.000001
      CALL BMUAV(A,M,N,U,V,L,EPS,KA,S,E,WORK)
      WRITE(*,*)
      WRITE(*,10) L
10    FORMAT(1X,'L=',I3)
      WRITE(*,*)
      WRITE(*,20)
20    FORMAT(1X,'MAT U:')
      WRITE(*,30) ((U(I,J),J=1,M),I=1,M)
30    FORMAT(1X,3D15.6)
      WRITE(*,*)
      WRITE(*,40)
```

```
40    FORMAT(1X,'MAT A:')
      WRITE(*,50) ((A(I,J),J=1,N),I=1,M)
50    FORMAT(1X,4D15.6)
      WRITE(*,*)
      WRITE(*,60)
60    FORMAT(1X,'MAT V:')
      WRITE(*,70) ((V(I,J),J=1,N),I=1,N)
70    FORMAT(1X,4D15.6)
      WRITE(*,*)
      CALL BRMUL(U,A,3,3,4,C)
      CALL BRMUL(C,V,3,4,4,A)
      WRITE(*,80)
80    FORMAT(1X,'MAT UAV:')
      WRITE(*,50) ((A(I,J),J=1,N),I=1,M)
      WRITE(*,*)
      END
```

运行结果为

```
L= 0

MAT  U:
 -.884994D-01   -.902880D+00   -.420685D+00
 -.925355D+00   -.817768D-01    .370177D+00

 -.368628D+00    .422043D+00   -.828247D+00
MAT  A:
  .320792D+01    .000000D+00    .000000D+00    .000000D+00
  .000000D+00    .213495D+01    .000000D+00    .000000D+00
  .000000D+00    .000000D+00    .107297D+01    .000000D+00

MAT  V:
 -.719419D+00   -.201136D+00    .275877D-01   -.664244D+00
 -.301829D+00   -.658891D+00    .422904D+00    .543980D+00
 -.473992D+00    .724850D+00    .392077D+00    .310161D+00
 -.408248D+00   -.412968D-16   -.816497D+00    .408248D+00

MAT  UAV:
  .100000D+01    .100000D+01   -.100000D+01   -.100000D+01
  .200000D+01    .100000D+01    .226617D-06    .200000D+01
  .100000D+01   -.100000D+01    .902758D-07    .100000D+01
```

2.13 求广义逆的奇异值分解法

一、功能

利用奇异值分解求一般实矩阵 A 的广义逆 A^+。

二、方法说明

设 A 为 $m\times n$ 阶实矩阵，其奇异值分解式为

$$A=U\begin{bmatrix}\Sigma & O\\ O & O\end{bmatrix}V^T$$

其中 U 为 $m\times m$ 阶的列正交矩阵(称为左奇异向量)；V 为 $n\times n$ 阶的列正交矩阵(称为右奇异向量)；$\Sigma=\mathrm{diag}\,(\sigma_1,\sigma_2,\cdots,\sigma_r)(r\leqslant\min\{m,n\})$，且 $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_r>0$，$\sigma_i(i=1,2,\cdots,r)$ 称为矩阵 A 的奇异值。关于奇异值分解参看 2.12 节。

则 A 的广义逆为

$$A^+=V_1\Sigma^{-1}U_1^T$$

其中 U_1 为 U 中前 r 列列正交向量组构成的 $m\times r$ 阶矩阵；V_1 为 V 中前 r 列列正交向量组构成的 $n\times r$ 阶矩阵。

三、子程序语句

SUBROUTINE BGINV(M,N,A,AA,L,EPS,U,V,KA,S,E,WORK)

四、形参说明

M——整型变量，输入参数。矩阵 A 的行数。

N——整型变量，输入参数。矩阵 A 的列数。

A——双精度实型二维数组，体积为 M×N，输入兼输出参数。调用时存放矩阵 A；返回矩阵 A 的奇异值(以非递增次序存放在对角线上)。

AA——双精度实型二维数组，体积为 N×M，输出参数。返回 A 的广义逆 A^+。

L——整型变量，输出参数。若返回 L=0，则说明正常返回；若 L≠0，说明在奇异值分解时失败。

EPS——实型变量，输入参数。奇异值分解时的控制精度要求。

U——双精度实型二维数组，体积为 M×M，输出参数。返回奇异值分解式中的左奇异向量 U。

V——双精度实型二维数组，体积为 N×N，输出参数，返回奇异值分解式中的右奇异向量 V。

KA——整型变量，输入参数。KA=max$\{M,N\}+1$。

S,E,WORK——均为双精度实型一维数组，长度为 KA。本子程序的工作数组。

五、子程序(文件名：BGINV.FOR)

```
      SUBROUTINE BGINV(M,N,A,AA,L,EPS,U,V,KA,S,E,WORK)
      DIMENSION A(M,N),U(M,M),V(N,N),AA(N,M)
      DIMENSION S(KA),E(KA),WORK(KA)
      DOUBLE PRECISION A,U,V,AA,S,E,WORK
      CALL BMUAV(A,M,N,U,V,L,EPS,KA,S,E,WORK)
      IF (L.EQ.0) THEN
        K=1
10      IF (A(K,K).NE.0.0) THEN
          K=K+1
          IF (K.LE.MIN(M,N)) GOTO 10
```

```
        END IF
        K=K-1
        IF (K.NE.0) THEN
          DO 40 I=1,N
          DO 40 J=1,M
            AA(I,J)=0.0
            DO 30 II=1,K
30          AA(I,J)=AA(I,J)+V(II,I)*U(J,II)/A(II,II)
40        CONTINUE
        END IF
      END IF
      RETURN
      END
```

六. 例

求矩阵

$$A=\begin{bmatrix}1 & 2 & 3 & 4 & 11\\ 6 & 7 & 8 & 9 & 10\\ 1 & 2 & 13 & 0 & 11\\ 16 & 17 & 8 & 9 & 13\\ 2 & 4 & 3 & 4 & 6\end{bmatrix}$$

的广义逆 A^+,并用 A^{++} 来验证结果的正确性。

主程序(文件名:BGINV0.FOR)为

```
      DIMENSION A(5,5),U(5,5),V(5,5),C(5,5)
      DIMENSION S(6),E(6),WORK(6)
      DOUBLE PRECISION A,U,V,C,S,E,WORK
      DATA A/1.0,6.0,1.0,16.0,2.0,2.0,7.0,2.0,17.0,
     *        4.0,3.0,8.0,13.0,8.0,3.0,4.0,9.0,0.0,
     *        9.0,4.0,11.0,10.0,11.0,13.0,6.0/
      M=5
      N=5
      KA=6
      EPS=0.000001
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(1X,'MAT A IS:')
      WRITE(*,60) ((A(I,J),J=1,N),I=1,M)
      CALL BGINV(M,N,A,C,L,EPS,U,V,KA,S,E,WORK)
      IF (L.EQ.0) THEN
        WRITE(*,*)
        WRITE(*,50)
50      FORMAT(1X,'MAT A+ IS:')
```

```
        WRITE(*,60) ((C(I,J),J=1,M),I=1,N)
60      FORMAT(1X,5D13.6)
        WRITE(*,*)
      END IF
      CALL BGINV(N,M,C,A,L,EPS,U,V,KA,S,E,WORK)
      IF (L.EQ.0) THEN
        WRITE(*,70)
70      FORMAT(1X,'MAT A++ IS:')
        WRITE(*,60) ((A(I,J),J=1,N),I=1,M)
        WRITE(*,*)
      END IF
      END
```

运行结果为

```
MAT  A  IS:
 .100000D+01   .200000D+01   .300000D+01   .400000D+01   .110000D+02
 .600000D+01   .700000D+01   .800000D+01   .900000D+01   .100000D+02
 .100000D+01   .200000D+01   .130000D+02   .000000D+00   .110000D+02
 .160000D+02   .170000D+02   .800000D+01   .900000D+01   .130000D+02
 .200000D+01   .400000D+01   .300000D+01   .400000D+01   .600000D+01

MAT  A+  IS:
 .162240D+00   .149440D+00  -.224000D-01   .835200D-01  -.686400D+00
-.207120D+00  -.230720D+00   .212000D-01   .102400D-01   .703200D+00
-.107200D+00   .768000D-01   .720000D-01  -.256000D-01  -.800000D-02
-.180000D-01   .192000D+00  -.700000D-01  -.640000D-01  -.200000D-01
 .149600D+00  -.624000D-01   .400000D-02   .208000D-01  -.560000D-01

MAT  A++  IS:
 .100000D+01   .200000D+01   .300000D+01   .400001D+01   .110000D+02
 .600000D+01   .700000D+01   .800000D+01   .900000D+01   .100000D+02
 .100001D+01   .200001D+01   .130000D+02   .892302D-06   .110000D+02
 .160000D+02   .170000D+02   .799999D+01   .900000D+01   .130000D+02
 .200000D+01   .400000D+01   .300000D+01   .400000D+01   .600000D+01
```

七、附注

本子程序需要调用奇异值分解子程序 BMUAV，参看 2.12 节。

第3章 矩阵特征值与特征向量的计算

3.1 约化对称矩阵为三对角阵的豪斯荷尔德变换法

一、功能

用豪斯荷尔德(Householder)变换将对称矩阵约化为对称三对角矩阵。

二、方法说明

化 n 阶对称矩阵 A 为三对角阵的豪斯荷尔德方法，就是使 A 经过 $n-2$ 次正交变换后变成三对角阵 A_{n-1}，即

$$A_{n-1} = P_{n-2}\cdots P_2 P_1 A P_1 P_2 \cdots P_{n-2}$$

每一次的正交变换具有如下形式

$$P_i = I - U_i U_i^T / H_i$$

P_i 为对称正交矩阵。其中

$$H_i = \frac{1}{2} U_i^T U_i$$

$$U_i = (a_{k1}^{(i)}, a_{k2}^{(i)}, \cdots, a_{k,k-2}^{(i)}, a_{k,k-1}^{(i)} \pm \sigma_i^{\frac{1}{2}}, 0, \cdots, 0)^T$$

$$\sigma_i = (a_{k1}^{(i)})^2 + (a_{k2}^{(i)})^2 + \cdots + (a_{k,k-1}^{(i)})^2$$

式中 $k = n - i + 1$

对 A 的每一次变换为

$$A_{i+1} = P_i A_i P_i = (I - U_i U_i^T / H_i) A_i (I - U_i U_i^T / H_i)$$

若令

$$S_i = A_i U_i / H_i$$

$$W_i = U_i^T S_i / 2H_i$$

$$Q_i = S_i - W_i U_i$$

则有

$$A_{i+1} = A_i - U_i Q_i^T - Q_i U_i^T$$

其中 S_i 的形式为

$$S_i = (p_{i1}, p_{i2}, \cdots, p_{ik}, 0, \cdots, 0)^T$$

Q_i 的形式为

$$Q_i = (p_{i1} - w_i u_{i1},\ p_{i2} - w_i u_{i2}, \cdots, p_{i,k-1} - w_i u_{i,k-1}, p_{ik}, 0, \cdots, 0)^T$$

三、子程序语句

SUBROUTINE CSTRQ(A,N,Q,B,C)

四、形参说明

A——双精度实型二维数组，体积为N×N，输入参数。存放对称矩阵，在返回时将被破坏。

N——整型变量，输入参数。实对称矩阵的阶数。

Q——双精度实型二维数组，体积为N×N，输出参数。返回豪斯荷尔德变换的乘积矩阵。在与3.2节中的子程序CSSTQ联用时，若将Q作为子程序CSSTQ的输入，则可计算A的特征值与相应的特征向量。

B——双精度实型一维数组，长度为N，输出参数。返回对称三对角阵中的主对角线元素。

C——双精度实型一维数组，长度为N，输出参数。前N－1个元素返回对称三对角阵中次对角线上的元素。

由返回的B与C可知，返回的对称三对角阵为

$$T=\begin{bmatrix} b_1 & c_1 & & & & \\ c_1 & b_2 & c_2 & & 0 & \\ & c_2 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & c_{n-2} & b_{n-1} & c_{n-1} \\ & & & & c_{n-1} & b_n \end{bmatrix}$$

五、子程序(文件名:CSTRQ.FOR)

```
      SUBROUTINE CSTRQ(A,N,Q,B,C)
      DIMENSION A(N,N),Q(N,N),B(N),C(N)
      DOUBLE PRECISION A,Q,B,C,F,H,G,H2
      DO 10 I=1,N
      DO 10 J=1,N
10    Q(I,J)=A(I,J)
      DO 80 I=N,2,-1
        H=0.0
        IF (I.GT.2) THEN
          DO 20 K=1,I-1
20        H=H+Q(I,K)*Q(I,K)
        END IF
        IF (H+1.0.EQ.1.0) THEN
          C(I)=0.0
          IF (I.EQ.2) C(I)=Q(I,I-1)
          B(I)=0.0
        ELSE
          C(I)=SQRT(H)
          IF (Q(I,I-1).GT.0.0) C(I)=-C(I)
          H=H-Q(I,I-1)*C(I)
```

```
          Q(I,I-1)=Q(I,I-1)-C(I)
          F=0.0
          DO 50 J=1,I-1
            Q(J,I)=Q(I,J)/H
            G=0.0
            DO 30 K=1,J
30          G=G+Q(J,K)*Q(I,K)
            IF (J+1.LE.I-1) THEN
              DO 40 K=J+1,I-1
40            G=G+Q(K,J)*Q(I,K)
            END IF
            C(J)=G/H
            F=F+G*Q(J,I)
50        CONTINUE
          H2=F/(H+H)
          DO 70 J=1,I-1
            F=Q(I,J)
            G=C(J)-H2*F
            C(J)=G
            DO 60 K=1,J
60          Q(J,K)=Q(J,K)-F*C(K)-G*Q(I,K)
70        CONTINUE
          B(I)=H
        END IF
80    CONTINUE
      DO 85 I=1,N-1
85    C(I)=C(I+1)
      C(N)=0.0
      B(1)=0.0
      DO 130 I=1,N
        IF ((B(I).NE.0.0).AND.(I-1.GE.1)) THEN
          DO 110 J=1,I-1
            G=0.0
            DO 90 K=1,I-1
90          G=G+Q(I,K)*Q(K,J)
            DO 100 K=1,I-1
100         Q(K,J)=Q(K,J)-G*Q(K,I)
110       CONTINUE
        END IF
        B(I)=Q(I,I)
        Q(I,I)=1.0
        IF (I-1.GE.1) THEN
```

```
          DO 120 J=1,I-1
            Q(I,J)=0.0
            Q(J,I)=0.0
120       CONTINUE
        END IF
130   CONTINUE
      RETURN
      END
```

六. 例

设实对称矩阵为

$$A=\begin{bmatrix}10 & 1 & 2 & 3 & 4\\ 1 & 9 & -1 & 2 & -3\\ 2 & -1 & 7 & 3 & -5\\ 3 & 2 & 3 & 12 & -1\\ 4 & -3 & -5 & -1 & 15\end{bmatrix}$$

将 A 约化为对称三对角阵，且给出豪斯荷尔德变换的乘积矩阵。

主程序（文件名：CSTRQ0.FOR）为

```
      DIMENSION A(5,5),Q(5,5),B(5),C(5)
      DOUBLE PRECISION A,Q,B,C
      DATA A/10.0,1.0,2.0,3.0,4.0,1.0,9.0,-1.0,2.0,-3.0,2.0,
     +       -1.0,7.0,3.0,-5.0,3.0,2.0,3.0,12.0,-1.0,4.0,
     *       -3.0,-5.0,-1.0,15.0/
      N=5
      CALL CSTRQ(A,N,Q,B,C)
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(1X,'MAT A IS:')
      WRITE(*,20) ((A(I,J),J=1,N),I=1,N)
20    FORMAT(1X,5D15.6)
      WRITE(*,*)
      WRITE(*,30)
30    FORMAT(1X,'MAT Q IS:')
      WRITE(*,20) ((Q(I,J),J=1,N),I=1,N)
      WRITE(*,*)
      WRITE(*,40)
40    FORMAT(1X,'MAT B IS:')
      WRITE(*,50) (B(I),I=1,N)
50    FORMAT(1X,5D15.6)
      WRITE(*,*)
      WRITE(*,60)
60    FORMAT(1X,'MAT C IS:')
```

```
      WRITE(*,70) (C(I),I=1,N)
70    FORMAT(1X,5D15.6)
      WRITE(*,*)
      END
```

运行结果为

```
MAT A IS:
  .100000D+02   .100000D+01   .200000D+01   .300000D+01   .400000D+01
  .100000D+01   .900000D+01  -.100000D+01   .200000D+01  -.300000D+01
  .200000D+01  -.100000D+01   .700000D+01   .300000D+01  -.500000D+01
  .300000D+01   .200000D+01   .300000D+01   .120000D+02  -.100000D+01
  .400000D+01  -.300000D+01  -.500000D+01  -.100000D+01   .150000D+02
MAT Q IS:
 -.384542D-01  -.826960D+00   .305490D-01   .560112D+00   .000000D+00
 -.868565D+00  -.240190D+00   .106922D+00  -.420084D+00   .000000D+00
  .449244D+00  -.503710D+00  -.232936D+00  -.700140D+00   .000000D+00
  .205658D+00  -.687167D-01   .966113D+00  -.140028D+00   .000000D+00
  .000000D+00   .000000D+00   .000000D+00   .000000D+00   .100000D+01
MAT B IS:
  .929520D+01   .116267D+02   .109604D+02   .611765D+01   .150000D+02
MAT C IS:
 -.749485D+00  -.449627D+01  -.215704D+01   .714143D+01   .000000D+00
```

即返回的对称三对角阵为

$$T=\begin{bmatrix} 9.29520 & -0.749485 & 0.000000 & 0.000000 & 0.000000 \\ -0.749485 & 11.6267 & -4.49627 & 0.000000 & 0.000000 \\ 0.000000 & -4.49627 & 10.9604 & -2.15704 & 0.000000 \\ 0.000000 & 0.000000 & -2.15704 & 6.11765 & 7.14143 \\ 0.000000 & 0.000000 & 0.000000 & 7.14143 & 15.0000 \end{bmatrix}$$

七、附注

本子程序与 3.2 节的子程序 CSSTQ 联用，可以求实对称矩阵的全部特征值与相应的特征向量。

3.2 实对称三对角阵全部特征值与相应特征向量的计算

一、功能

求实对称三对角阵的全部特征值与相应的特征向量。

二、方法说明

本方法是 QR 方法的变形。有关 QR 方法的说明请参看 3.4 节。

本子程序只能求实对称三对角阵的全部特征值，但如果与 3.1 节中的子程序 CSTRQ 联用，可以计算一般实对称矩阵的全部特征值与相应的特征向量。

三、子程序语句

SUBROUTINE　CSSTQ(N,B,C,Q,EPS,L)

四、形参说明

N——整型变量，输入参数。对称三对角阵的阶数。

B——双精度实型一维数组，长度为N，输入兼输出参数。调用时依次存放实对称三对角阵主对角线上的各元素；返回全部特征值。

C——双精度实型一维数组，长度为N，输入参数。其中前N−1个元素依次存放实对称三对角阵中次对角线上的元素。返回时将被破坏。

由B与C的定义可知，实对称三对角阵为

$$T=\begin{bmatrix} b_1 & c_1 & & & & \\ c_1 & b_2 & c_2 & & 0 & \\ & c_2 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & c_{n-2} & b_{n-1} & c_{n-1} \\ & & & & c_{n-1} & b_n \end{bmatrix}$$

Q——双精度实型二维数组，体积为N×N，输入兼输出参数。如果调用时Q为一个单位矩阵，则返回对称三对角阵T的特征向量组，其中第j列为与数组B中第j个特征值对应的特征向量；如果调用时Q为由3.1节中的子程序CSTRQ返回的豪斯荷尔德变换的乘积矩阵，则返回在子程序CSTRQ中输入的实对称矩阵A的特征向量组，其中第j列为与数组B中第j个特征值对应的特征向量。

EPS——实型变量，输入参数。控制精度要求。

L——整型变量，输出参数。若返回L=0，说明工作失败；若L≠0，则说明正常返回。

五、子程序(文件名:CSSTQ.FOR)

```
      SUBROUTINE CSSTQ(N,B,C,Q,EPS,L)
      DIMENSION B(N),C(N),Q(N,N)
      DOUBLE PRECISION B,C,Q,D,H,P,R,F,E,S,G
      C(N)=0.0
      D=0.0
      F=0.0
      DO 50 J=1,N
        IT=0
        H=EPS*(ABS(B(J))+ABS(C(J)))
        IF (H.GT.D) D=H
        M=J-1
10      M=M+1
        IF (M.LE.N) THEN
          IF (ABS(C(M)).GT.D) GOTO 10
        END IF
        IF (M.NE.J) THEN
```

```
15    IF (IT.EQ.60) THEN
        L=0
        WRITE(*,18)
18      FORMAT(1X,' FAIL')
        RETURN
      END IF
      IT=IT+1
      G=B(J)
      P=(B(J+1)-G)/(2.0*C(J))
      R=SQRT(P*P+1.0)
      IF (P.GE.0.0) THEN
        B(J)=C(J)/(P+R)
      ELSE
        B(J)=C(J)/(P-R)
      END IF
      H=G-B(J)
      DO 20 I=J+1,N
20    B(I)=B(I)-H
      F=F+H
      P=B(M)
      E=1.0
      S=0.0
      DO 40 I=M-1,J,-1
        G=E*C(I)
        H=E*P
        IF (ABS(P).GE.ABS(C(I))) THEN
          E=C(I)/P
          R=SQRT(E*E+1.0)
          C(I+1)=S*P*R
          S=E/R
          E=1.0/R
        ELSE
          E=P/C(I)
          R=SQRT(E*E+1.0)
          C(I+1)=S*C(I)*R
          S=1.0/R
          E=E/R
        END IF
        P=E*B(I)-S*G
        B(I+1)=H+S*(E*G+S*B(I))
        DO 30 K=1,N
          H=Q(K,I+1)
```

```
          Q(K,I+1)=S*Q(K,I)+E*H
          Q(K,I)=E*Q(K,I)-S*H
30       CONTINUE
40     CONTINUE
       C(J)=S*P
       B(J)=E*P
       IF (ABS(C(J)).GT.D) GOTO 15
     END IF
     B(J)=B(J)+F
50   CONTINUE
     DO 80 I=1,N
       K=I
       P=B(I)
       IF (I+1.LE.N) THEN
         J=I
60       J=J+1
         IF (J.LE.N) THEN
           IF (B(J).LE.P) THEN
             K=J
             P=B(J)
             GOTO 60
           END IF
         END IF
       END IF
       IF (K.NE.I) THEN
         B(K)=B(I)
         B(I)=P
         DO 70 J=1,N
           P=Q(J,I)
           Q(J,I)=Q(J,K)
           Q(J,K)=P
70       CONTINUE
       END IF
80   CONTINUE
     L=1
     RETURN
     END
```

六、例

设对称矩阵为

$$A=\begin{bmatrix}10 & 1 & 2 & 3 & 4\\ 1 & 9 & -1 & 2 & -3\\ 2 & -1 & 7 & 3 & -5\\ 3 & 2 & 3 & 12 & -1\\ 4 & -3 & -5 & -1 & 15\end{bmatrix}$$

与 3.1 节中的子程序 CSTRQ 联用，求 A 的全部特征值与相应的特征向量。取 EPS=0.000001。

主程序(文件名:CSSTQ0.FOR)为

```
      DIMENSION A(5,5),Q(5,5),B(5),C(5)
      DOUBLE PRECISION A,Q,B,C
      DATA A/10.0,1.0,2.0,3.0,4.0,1.0,9.0,-1.0,2.0,-3.0,2.0,
     *     -1.0,7.0,3.0,-5.0,3.0,2.0,3.0,12.0,-1.0,4.0,
     *     -3.0,-5.0,-1.0,15.0/
      N=5
      CALL CSTRQ(A,N,Q,B,C)
      EPS=0.000001
      CALL CSSTQ(N,B,C,Q,EPS,L)
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(1X,'MAT A IS:')
      WRITE(*,20) ((A(I,J),J=1,N),I=1,N)
20    FORMAT(1X,5D15.6)
      IF (L.NE.0) THEN
        WRITE(*,*)
        WRITE(*,30)
30      FORMAT(1X,'MAT Q IS:')
        WRITE(*,20) ((Q(I,J),J=1,N),I=1,N)
        WRITE(*,*)
        WRITE(*,40)
40      FORMAT(1X,'MAT B IS:')
        WRITE(*,50) (B(I),I=1,N)
50      FORMAT(1X,5D15.6)
      END IF
      WRITE(*,*)
      END
```

运行结果为

```
MAT A IS:
   .100000D+02    .100000D+01    .200000D+01    .300000D+01    .400000D+01
   .100000D+01    .900000D+01   -.100000D+01    .200000D+01   -.300000D+01
   .200000D+01   -.100000D+01    .700000D+01    .300000D+01   -.500000D+01
   .300000D+01    .200000D+01    .300000D+01    .120000D+02   -.100000D+01
```

.400000D+01	-.300000D+01	-.500000D+01	-.100000D+01	.150000D+02

MAT Q IS:

.554083D+00	.521511D-01	.387297D+00	-.623702D+00	.174505D+00
.199681D+00	-.859964D+00	-.366221D+00	-.159101D+00	-.247303D+00
.256510D+00	.505575D+00	-.704377D+00	-.227297D+00	-.361642D+00
-.860403D+00	.201167D-03	.118926D+00	-.692684D+00	-.264411D+00
-.174280D+00	-.462192D-01	-.453423D+00	-.232822D+00	.841244D+00

MAT B IS:

.699484D+01	.936555D+01	.165527D+01	.158089D+02	.191754D+02

3.3 约化一般实矩阵为赫申伯格矩阵的初等相似变换法

一、功能

用初等相似变换将一般实矩阵化为上 H 矩阵，即赫申伯格(Hessenberg)矩阵。

二、方法说明

为了将 n 阶实方阵 A 化为上 H 矩阵，只需依次将 A 中的第 $1,2,\cdots,n-2$ 列化为上 H 矩阵。为此，只要对于 $k=2,3,\cdots,n-1$ 作如下变换：

(1) 从第 $k-1$ 列的第 $k-1$ 个以下的元素中选出绝对值最大的元素 $a_{l,k-1}$。

(2) 交换第 l 行与第 k 行，交换第 l 列与第 k 列。

(3) 对于 $i=k+1,\cdots,n$ 作变换

$$a_{i,k-1}/a_{k,k-1}\Rightarrow m\ ,\ 0\Rightarrow a_{i,k-1}$$

$$a_{ij}-ma_{kj}\Rightarrow a_{ij},\ j=k,\cdots,n$$

$$a_{jk}+ma_{ji}\Rightarrow a_{jk},\ j=1,2,\cdots,n$$

三、子程序语句

SUBROUTINE CHHBG(A,N)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入兼输出参数。调用时存放一般实矩阵；返回上 H 矩阵。

N——整型变量，输入参数。矩阵的阶数。

五、子程序(文件名:CHHBG.FOR)

```
      SUBROUTINE CHHBG(A,N)
      DIMENSION A(N,N)
      DOUBLE PRECISION A,D,T
      DO 100 K=2,N-1
        D=0.0
        DO 10 J=K,N
          IF (ABS(A(J,K-1)).GT.ABS(D)) THEN
            D=A(J,K-1)
            I=J
```

```
      END IF
10    CONTINUE
      IF (ABS(D)+1.0.NE.1.0) THEN
        IF (L.NE.K) THEN
          DO 20 J=K-1,N
          T=A(L,J)
          A(L,J)=A(K,J)
20        A(K,J)=T
          DO 30 J=1,N
          T=A(J,L)
          A(J,L)=A(J,K)
30        A(J,K)=T
        END IF
        DO 90 I=K+1,N
          T=A(I,K-1)/D
          A(I,K-1)=0.0
          DO 40 J=K,N
40        A(I,J)=A(I,J)-T*A(K,J)
          DO 50 J=1,N
50        A(J,K)=A(J,K)+T*A(J,I)
90      CONTINUE
      END IF
100   CONTINUE
      RETURN
      END
```

六、例

设实方阵为

$$A=\begin{bmatrix} 1 & 6 & -3 & -1 & 7 \\ 8 & -15 & 18 & 5 & 4 \\ -2 & 11 & 9 & 15 & 20 \\ -13 & 2 & 21 & 30 & -6 \\ 17 & 22 & -5 & 3 & 6 \end{bmatrix}$$

用初等相似变换将 A 化为上 H 矩阵。

主程序(文件名:CHHBG0.FOR)为

```
      DIMENSION A(5,5)
      DOUBLE PRECISION A
      DATA A/1.0,8.0,-2.0,-13.0,17.0,6.0,-15.0,11.0,2.0,
     *       22.0,-3.0,18.0,9.0,21.0,-5.0,-1.0,5.0,
     *       15.0,30.0,3.0,7.0,4.0,20.0,-6.0,6.0/
      CALL CHHBG(A,5)
      WRITE(*,10) ((A(I,J),J=1,5),I=1,5)
```

```
10      FORMAT(1X,5D15.6)
        END
```

运行结果为

```
.100000D+01   .109412D+02   .618567D+01   .826783D+01  -.300000D+01
.170000D+02   .146471D+02   .248730D+02   .257797D+02  -.500000D+01
.000000D+00  -.192699D+02   .350096D+02   .583910D+01   .171765D+02
.000000D+00   .000000D+00  -.613733D+02  -.455544D+02   .618675D+01
.000000D+00   .000000D+00   .000000D+00  -.228526D+02   .258977D+02
```

七、附注

本子程序与 3.4 节中的子程序 CHHQR 联用，可以计算一般实矩阵的全部特征值。

3.4 求赫申伯格矩阵全部特征值的 QR 方法

一、功能

用带原点位移的双重步 QR 方法求上 H 矩阵的全部特征值。

二、方法说明

对于不可约上 H 矩阵 A，可以通过一系列双重步 QR 变换，使上 H 矩阵变为对角块全部是一阶块或二阶块，从而直接解出全部特征值。

双重步 QR 变换的步骤可以归纳如下。

(1) 确定一个初等正交对称矩阵 Q_0，对 A 作相似变换

$$A_1 = Q_0 A Q_0$$

Q_0 具有如下形式：

$$Q_0 = \begin{bmatrix} Q_0^{(0)} & O \\ O & I_{n-3} \end{bmatrix}$$

其中 $Q_0^{(0)}$ 为 3×3 的矩阵。若令

$$\begin{aligned}
\alpha &= a_{n-1,n-1} + a_{nn} \\
\beta &= a_{n-1,n-1}a_{nn} - a_{n-1,n}a_{n,n-1} \\
p_0 &= a_{11}(a_{11} - \alpha) + a_{12}a_{21} + \beta \\
q_0 &= a_{21}(a_{11} + a_{22} - \alpha) \\
r_0 &= a_{21}a_{32}
\end{aligned}$$

则有

$$Q_0^{(0)} = \begin{bmatrix} -\dfrac{p_0}{s_0} & -\dfrac{q_0}{s_0} & -\dfrac{r_0}{s_0} \\ -\dfrac{q_0}{s_0} & \dfrac{p_0}{s_0} + \dfrac{r_0^2}{s_0(p_0+s_0)} & -\dfrac{q_0 r_0}{s_0(p_0+s_0)} \\ -\dfrac{r_0}{s_0} & -\dfrac{q_0 r_0}{s_0(p_0+s_0)} & \dfrac{p_0}{s_0} + \dfrac{q_0^2}{s_0(p_0+s_0)} \end{bmatrix}$$

其中 $s_0 = \mathrm{sign}(p_0)\sqrt{p_0^2 + q_0^2 + r_0^2}$

由此可得

$$A_1 = Q_0 A Q_0$$

(2) 利用同样的方法,依次确定正交对称矩阵 $Q_1, Q_2, \cdots, Q_{n-2}$ 对 $A_1, A_2, \cdots, A_{n-2}$ 作相似变换

$$A_{i+1} = Q_i A_i Q_i,\ i = 1, 2, \cdots, n-2$$

最后可得到上 H 矩阵

$$A_{n-1} = Q_{n-2} A_{n-2} Q_{n-2}$$

在这个过程中,Q_i 具有如下形式:

$$Q_i = \begin{bmatrix} I_i & O & O \\ O & Q_i^{(0)} & O \\ O & O & I_{n-i-3} \end{bmatrix}$$

其中 $Q_i^{(0)}$ 为 3×3 矩阵。若令

$$p_i = a_{i+1,i}^{(i)},\quad q_i = a_{i+2,i}^{(i)},\quad r_i = a_{i+3,i}^{(i)}$$

但

$$p_{n-2} = a_{n-1,n-2}^{(n-2)},\quad q_{n-2} = a_{n,n-2}^{(n-2)},\quad r_{n-2} = 0$$

则有

$$Q_i^{(0)} = \begin{bmatrix} -\dfrac{p_i}{s_i} & -\dfrac{q_i}{s_i} & -\dfrac{r_i}{s_i} \\ -\dfrac{q_i}{s_i} & \dfrac{p_i}{s_i} + \dfrac{r_i^2}{s_i(p_i+s_i)} & -\dfrac{q_i r_i}{s_i(p_i+s_i)} \\ -\dfrac{r_i}{s_i} & -\dfrac{q_i r_i}{s_i(p_i+s_i)} & \dfrac{p_i}{s_i} + \dfrac{q_i^2}{s_i(p_i+s_i)} \end{bmatrix}$$

其中 $s_i = \mathrm{sign}(p_i)\sqrt{p_i^2 + q_i^2 + r_i^2}$

反复进行以上两步,直到将上 H 阵 A 变为对角块全部是一阶块或二阶块为止,此时就可以直接从各一阶块或二阶块中解出全部特征值。

QR 方法是一种迭代的方法。在每次迭代计算过程中,如果下次对角线元素的绝对值小到一定程度,就可以把它们当作零看待,将矩阵分割为各不可约上 H 阵,以便逐步降低主子阵的阶数,从而减少计算工作量。

在本子程序中,判断下次对角线元素绝对值是否可认为是零的准则为

$$|a_{k,k-1}| \leqslant \epsilon(|a_{k-1,k-1}| + |a_{kk}|)$$

本子程序与 3.3 节中的子程序 CHHBG 联用可以求一般实矩阵的全部特征值。

本子程序还可以用来求一般首一多项式的全部根,参看 4.5 节。

三、子程序语句

SUBROUTINE　CHHQR(A,N,U,V,EPS,JT)

四、形参说明

A——双精度实型二维数组,体积为 N×N,输入参数。存放上 H 矩阵,返回时将被破坏。

N——整型变量,输入参数。上 H 阵的阶数。

U,V——均为双精度实型一维数组,长度为N,输出参数。分别返回N个特征值的实部与虚部。

EPS——实型变量,输入参数。控制精度要求。

JT——整型变量,输出参数。若返回JT=0,则说明工作失败;若JT≠0,则说明正常返回。

五、子程序(文件名:CHHQR.FOR)

```
      SUBROUTINE CHHQR(A,N,U,V,EPS,JT)
      DIMENSION A(N,N),U(N),V(N)
      DOUBLE PRECISION A,U,V,W,B,C,X,Y,XY,P,Q,R,E,F,Z,G
      JT=1
      M=N
      IT=0
10    IF (M.EQ.0) RETURN
      L=M+1
40    L=L-1
      IF (ABS(A(L,L-1)).GT.EPS*(ABS(A(L-1,L-1))+ABS(A(L,L))).AND.
     *    L.GT.1) GOTO 40
      IF (L.EQ.M) THEN
        U(M)=A(M,M)
        V(M)=0.0
        M=M-1
        IT=0
        GOTO 10
      END IF
      IF (L.EQ.M-1) THEN
        B=-(A(M,M)+A(M-1,M-1))
        C=A(M,M)*A(M-1,M-1)-A(M,M-1)*A(M-1,M)
        W=B*B-4*C
        Y=SQRT(ABS(W))
        IF (W.GT.0.0) THEN
          XY=1.0
          IF (B.LT.0.0) XY=-1.0
          U(M)=(-B-XY*Y)/2.0
          U(M-1)=C/U(M)
          V(M)=0.0
          V(M-1)=0.0
        ELSE
          U(M)=-B/2.0
          U(M-1)=U(M)
          V(M)=Y/2.0
          V(M-1)=-V(M)
```

```
          END IF
          M=M-2
          IT=0
          GOTO 10
        END IF
        IF (IT.GE.60) THEN
          WRITE(*,50)
50        FORMAT(1X,'FAIL')
          JT=0
          RETURN
        END IF
        IT=IT+1
        DO 60 I=L+2,M
60      A(I,I-2)=0.0
        DO 70 I=L+3,M
70      A(I,I-3)=0.0
        DO 150 K=L,M-1
          IF (K.NE.L) THEN
            P=A(K,K-1)
            Q=A(K+1,K-1)
            R=0.0
            IF (K.NE.M-1) R=A(K+2,K-1)
          ELSE
            X=A(M,M)+A(M-1,M-1)
            Y=A(M-1,M-1)*A(M,M)-A(M-1,M)*A(M,M-1)
            P=A(L,L)*(A(L,L)-X)+A(L,L+1)*A(L+1,L)+Y
            Q=A(L+1,L)*(A(L,L)+A(L+1,L+1)-X)
            R=A(L+1,L)*A(L+2,L+1)
          END IF
          IF (ABS(P)+ABS(Q)+ABS(R).NE.0.0) THEN
            XY=1.0
            IF (P.LT.0.0) XY=-1.0
            S=XY*SQRT(P*P+Q*Q+R*R)
            IF (K.NE.L) A(K,K-1)=-S
            E=-Q/S
            F=-R/S
            X=-P/S
            Y=-X-F*R/(P+S)
            G=E*R/(P+S)
            Z=-X-E*Q/(P+S)
            DO 110 J=K,M
              P=X*A(K,J)+E*A(K+1,J)
```

```
            Q=E*A(K,J)+Y*A(K+1,J)
            R=F*A(K,J)+G*A(K+1,J)
            IF (K.NE.M-1) THEN
              P=P+F*A(K+2,J)
              Q=Q+G*A(K+2,J)
              R=R+Z*A(K+2,J)
              A(K+2,J)=R
            END IF
            A(K+1,J)=Q
            A(K,J)=P
110       CONTINUE
          J=K+3
          IF (J.GE.M) J=M
          DO 120 I=1,J
            P=X*A(I,K)+E*A(I,K+1)
            Q=E*A(I,K)+Y*A(I,K+1)
            R=F*A(I,K)+G*A(I,K+1)
            IF (K.NE.M-1) THEN
              P=P+F*A(I,K+2)
              Q=Q+G*A(I,K+2)
              R=R+Z*A(I,K+2)
              A(I,K+2)=R
            END IF
            A(I,K+1)=Q
            A(I,K)=P
120       CONTINUE
        END IF
150   CONTINUE
      GOTO 10
      END
```

六、例

设实矩阵为

$$A=\begin{bmatrix} 1 & 6 & -3 & -1 & 7 \\ 8 & -15 & 18 & 5 & 4 \\ -2 & 11 & 9 & 15 & 20 \\ -13 & 2 & 21 & 30 & -6 \\ 17 & 22 & -5 & 3 & 6 \end{bmatrix}$$

与3.3节中的子程序CHHBG联用，将A约化为上H矩阵，并求全部特征值。

主程序(文件名:CHHQR0.FOR)为

```
      DIMENSION A(5,5),U(5),V(5)
      DOUBLE PRECISION A,U,V
```

```
      DATA A/1.0,8.0,-2.0,-13.0,17.0,6.0,-15.0,11.0,2.0,
     *    22.0,-3.0,18.0,9.0,21.0,-5.0,-1.0,5.0,15.0,
     *    30.0,3.0,7.0,4.0,20.0,-6.0,6.0/
      CALL CHHBG(A,5)
      WRITE(*,10) ((A(I,J),J=1,5),I=1,5)
10    FORMAT(1X,5D15.6)
      EPS=1.0E-06
      CALL CHHQR(A,5,U,V,EPS,JT)
      IF (JT.NE.0) THEN
        DO 20 I=1,5
          WRITE(*,30) U(I),V(I)
20      CONTINUE
      END IF
30    FORMAT(1X,D15.6,2X,'+J',1X,D15.6)
      END
```

运行结果为

```
 .100000D+01   .109412D+02   .618567D+01   .826783D+01  -.300000D+01
 .170000D+02   .146471D+02   .248730D+02   .257797D+02   .500000D+01
 .000000D+00  -.192699D+02   .350096D+02   .583910D+01   .171765D+02
 .000000D+00   .000000D+00  -.613733D+02  -.455544D+02   .618675D+01
 .000000D+00   .000000D+00   .000000D+00  -.228526D+02   .258977D+02
-.661712D+00  +J   .000000D+00
 .429610D+02  +J   .000000D+00
-.153396D+02  +J  -.675565D+01
-.153396D+02  +J   .675565D+01
 .193800D+02  +J   .000000D+00
```

3.5 求实对称矩阵特征值与特征向量的雅可比法

一、功能

用雅可比(Jacobi)法求实对称矩阵的全部特征值与相应的特征向量。

二、方法说明

雅可比方法的基本思想如下。

设实对称矩阵为 A。在非对角线元素中选取一个绝对值最大者 a_{pq}，利用平面旋转变换矩阵 $R_0(p,q,\theta)$ 对 A 作正交相似变换：

$$A_1 = R_0(p,q,\theta)^T A R_0(p,q,\theta)$$

其中 $R_0(p,q,\theta)$ 的元素为

$$r_{pp} = \cos\theta,\quad r_{qq} = \cos\theta,\quad r_{pq} = -\sin\theta,\quad r_{qp} = \sin\theta$$

$$r_{ij} = 0,\quad i,j \neq p,q$$

如果按下式确定角度 θ：

$$\mathrm{tg}2\theta = 2a_{pq}/(a_{pp} - a_{qq})$$

则经上述变换后，非对角线元素的平方和将减少 $2a_{pq}^2$，对角线元素的平方和增加 $2a_{pq}^2$，矩阵中所有元素的平方和不变。反复进行上述变换，就可以逐步将矩阵 A 变为对角矩阵，对角线上的元素即为特征值，而每一步的平面旋转变换矩阵的乘积矩阵的第 i 列即为与第 i 个特征值对应的特征向量。

雅可比方法的基本步骤如下：

(1) 令 $S = I_n$(单位矩阵)。

(2) 选取非对角线元素中绝对值最大者 a_{pq}。

(3) 若 $|a_{pq}| < \varepsilon$，则迭代过程结束．此时，对角线元素即为特征值，即 $\lambda_i = a_{ii}(i = 1,2,\cdots,n)$，$S$ 中的第 i 列即为与 λ_i 对应的特征向量。否则继续下一步。

(4) 计算平面旋转变换矩阵的元素及其变换后的矩阵 A_1 的元素．由如下一系列公式给出：

$$x = -a_{pq}$$

$$y = \frac{1}{2}(a_{qq} - a_{pp})$$

$$\omega = \mathrm{sign}(y)\frac{x}{\sqrt{x^2 + y^2}}$$

$$\sin 2\theta = \omega$$

$$\sin\theta = \frac{\omega}{\sqrt{2 + (1 + \sqrt{1 - \omega^2})}}$$

$$\cos\theta = \sqrt{1 - \sin^2\theta}$$

$$a_{pp}^{(1)} = a_{pp}\cos^2\theta + a_{qq}\sin^2\theta + a_{pq}\sin 2\theta$$

$$a_{qq}^{(1)} = a_{pp}\sin^2\theta + a_{qq}\cos^2\theta - a_{pq}\sin 2\theta$$

$$a_{pq}^{(1)} = a_{qp}^{(1)} = 0$$

$$\left.\begin{aligned} a_{pj}^{(1)} &= a_{pj}\cos\theta + a_{qj}\sin\theta \\ a_{qj}^{(1)} &= -a_{pj}\sin\theta + a_{qj}\cos\theta \end{aligned}\right\} j \neq p,q$$

$$\left.\begin{aligned} a_{ip}^{(1)} &= a_{ip}\cos\theta + a_{iq}\sin\theta \\ a_{iq}^{(1)} &= -a_{ip}\sin\theta + a_{iq}\cos\theta \end{aligned}\right\} i \neq p,q$$

$$a_{ij}^{(1)} = a_{ij}, \quad i,j \neq p,q$$

(5) $S = SR(p,q,\theta)$，转(2)。

三、子程序语句

SUBROUTINE CJCBI(A,N,EPS,V,L)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入兼输出参数。调用时存放实对称矩阵；返回时，其对角线元素即为特征值，即 $\lambda_i = a_{ii}(i = 1,2,\cdots,n)$。

N——整型变量，输入参数．矩阵阶数．

EPS——实型变量，输入参数．控制精度要求．

V——双精度实型二维数组，体积为 N×N，输出参数。返回特征向量，第 i 列为与 λ_i 对应的特征向量。

L——整型变量；输出参数。若返回 L=0，说明程序工作可能失败；若 L≠0，说明正常返回。

五、子程序（文件名：CJCBI.FOR）

```
      SUBROUTINE CJCBI(A,N,EPS,V,L)
      DIMENSION A(N,N),V(N,N)
      DOUBLE PRECISION A,V,FM,CN,SN,OMEGA,X,Y
      INTEGER P,Q
      L=1
      DO 20 I=1,N
        V(I,I)=1.0
        DO 10 J=1,N
          IF (I.NE.J) V(I,J)=0.0
10      CONTINUE
20    CONTINUE
25    FM=0.0
      DO 30 I=2,N
      DO 30 J=1,I-1
        IF (ABS(A(I,J)).GT.FM) THEN
          FM=ABS(A(I,J))
          P=I
          Q=J
        END IF
30    CONTINUE
      IF (FM.LT.EPS) THEN
        L=1
        RETURN
      END IF
      IF (L.GT.100) THEN
        L=0
        RETURN
      END IF
      L=L+1
      X=-A(P,Q)
      Y=(A(Q,Q)-A(P,P))/2.0
      OMEGA=X/SQRT(X*X+Y*Y)
      IF (Y.LT.0.0) OMEGA=-OMEGA
      SN=1.0+SQRT(1.0-OMEGA*OMEGA)
      SN=OMEGA/SQRT(2.0*SN)
      CN=SQRT(1.0-SN*SN)
```

```
      FM=A(P,P)
      A(P,P)=FM*CN*CN+A(Q,Q)*SN*SN+A(P,Q)*OMEGA
      A(Q,Q)=FM*SN*SN+A(Q,Q)*CN*CN-A(P,Q)*OMEGA
      A(P,Q)=0.0
      A(Q,P)=0.0
      DO 60 J=1,N
        IF ((J.NE.P).AND.(J.NE.Q)) THEN
          FM=A(P,J)
          A(P,J)=FM*CN+A(Q,J)*SN
          A(Q,J)=-FM*SN+A(Q,J)*CN
        END IF
60    CONTINUE

      DO 70 I=1,N
        IF ((I.NE.P).AND.(I.NE.Q)) THEN
          FM=A(I,P)
          A(I,P)=FM*CN+A(I,Q)*SN
          A(I,Q)=-FM*SN+A(I,Q)*CN
        END IF
70    CONTINUE
      DO 80 I=1,N
        FM=V(I,P)
        V(I,P)=FM*CN+V(I,Q)*SN
        V(I,Q)=-FM*SN+V(I,Q)*CN
80    CONTINUE
      GOTO 25
      END
```

六、例

设实对称矩阵为

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

用雅可比法求全部特征值与相应的特征向量。取 EPS=0.001。

主程序(文件名:CJCBI0.FOR)为

```
      DIMENSION A(3,3),V(3,3)
      DOUBLE PRECISION A,V
      DATA A/2.0,-1.0,0.0,-1.0,2.0,-1.0,0.0,-1.0,2.0/
      EPS=0.001
      CALL CJCBI(A,3,EPS,V,L)
      IF (L.NE.0) THEN
        WRITE(*,20) (A(I,I),I=1,3)
```

```
      END IF
20    FORMAT(1X,3D15.6)
      WRITE(*,*)
      WRITE(*,20) ((V(I,J),J=1,3),I=1,3)
      END
```

运行结果为

```
.585786D+00    .341421D+01    .200000D+01

.500000D+00   -.500012D+00   -.707098D+00
.707107D+00    .707107D+00   -.120684D-04
.500000D+00   -.499988D+00    .707115D+00
```

本矩阵准确特征值为

$$\lambda_1 = 2 - \sqrt{2}, \quad \lambda_2 = 2 + \sqrt{2}, \quad \lambda_3 = 2.0$$

特征向量为

$$V_1 = \begin{bmatrix} 1/2 \\ \sqrt{2}/2 \\ 1/2 \end{bmatrix}, \quad V_2 = \begin{bmatrix} -1/2 \\ \sqrt{2}/2 \\ -1/2 \end{bmatrix}, \quad V_3 = \begin{bmatrix} -\sqrt{2}/2 \\ 0 \\ \sqrt{2}/2 \end{bmatrix}$$

3.6 求实对称矩阵特征值与特征向量的雅可比过关法

一、功能

用雅可比(Jacobi)过关法求实对称矩阵的全部特征值及对应的特征向量。

二、方法说明

基本方法同 3.5 节。在选取非对角线上的主元素时改用如下方法：

首先计算实对称矩阵 A 的非对角线元素的平方和的平方根

$$v_0 = \left(2\sum_{i=1}^{n}\sum_{j=1}^{i-1} a_{ij}^2\right)^{\frac{1}{2}}$$

然后设置关口 $v_1 = v_0/n$，在非对角线元素中按行扫描选取第一个绝对值大于或等于 v_1 的元素 a_{pq} 进行平面旋转变换，直到所有非对角线元素的绝对值均小于 v_1 为止。再设关口 $v_2 = v_1/n$ 重复这个过程，…以此类推，这个过程一直作到对于某个 $v_k < \epsilon$ 为止。

三、子程序语句

SUBROUTINE　CJCBJ(A,N,EPS,V)

四、形参说明

A——双精度实型二维数组，体积为 N×N，输入兼输出参数。调用时存放实对称矩阵；返回时其对角线上的元素为特征值。

N——整型变量，输入参数。实对称矩阵 A 的阶数。

EPS——实型变量，输入参数。控制精度要求。

V——双精度实型二维数组，体积为 N×N，输出参数。返回 N 个与特征值对应的特

征向量，其中 V 中第 i 列为与 $\lambda_i = a_{ii}$ 对应的特征向量。

五、子程序(文件名:CJCBJ.FOR)

```
      SUBROUTINE CJCBJ(A,N,EPS,V)
      DIMENSION A(N,N),V(N,N)
      DOUBLE PRECISION A,V,FF,FM,CN,SN,OMEGA,X,Y
      INTEGER P,Q
      DO 20 I=1,N
        V(I,I)=1.0
        DO 10 J=1,N
          IF (I.NE.J) V(I,J)=0.0
10      CONTINUE
20    CONTINUE
      FF=0.0
      DO 500 I=2,N
      DO 500 J=1,I-1
500   FF=FF+A(I,J)*A(I,J)
      FF=SQRT(2.0*FF)
205   FF=FF/(1.0*N)
25    DO 30 I=2,N
      DO 30 J=1,I-1
        IF (ABS(A(I,J)).GE.FF) THEN
          P=I
          Q=J
          GOTO 600
        END IF
30    CONTINUE
      IF (FF.GE.EPS) GOTO 205
      RETURN
600   X=-A(P,Q)
      Y=(A(Q,Q)-A(P,P))/2.0
      OMEGA=X/SQRT(X*X+Y*Y)
      IF (Y.LT.0.0) OMEGA=-OMEGA
      SN=1.0+SQRT(1.0-OMEGA*OMEGA)
      SN=OMEGA/SQRT(2.0*SN)
      CN=SQRT(1.0-SN*SN)
      FM=A(P,P)
      A(P,P)=FM*CN*CN+A(Q,Q)*SN*SN+A(P,Q)*OMEGA
      A(Q,Q)=FM*SN*SN+A(Q,Q)*CN*CN-A(P,Q)*OMEGA
      A(P,Q)=0.0
      A(Q,P)=0.0
      DO 60 J=1,N
```

```
      IF ((J.NE.P).AND.(J.NE.Q)) THEN
        FM=A(P,J)
        A(P,J)=FM*CN+A(Q,J)*SN
        A(Q,J)=-FM*SN+A(Q,J)*CN
      END IF
60    CONTINUE

      DO 70 I=1,N
        IF ((I.NE.P).AND.(I.NE.Q)) THEN
          FM=A(I,P)
          A(I,P)=FM*CN+A(I,Q)*SN
          A(I,Q)=-FM*SN+A(I,Q)*CN
        END IF
70    CONTINUE
      DO 80 I=1,N
        FM=V(I,P)
        V(I,P)=FM*CN+V(I,Q)*SN
        V(I,Q)=-FM*SN+V(I,Q)*CN
80    CONTINUE
      GOTO 25
      END
```

六、例

求实对称矩阵

$$A = \begin{bmatrix} 10 & 1 & 2 & 3 & 4 \\ 1 & 9 & -1 & 2 & -3 \\ 2 & -1 & 7 & 3 & -5 \\ 3 & 2 & 3 & 12 & -1 \\ 4 & -3 & -5 & -1 & 15 \end{bmatrix}$$

的全部特征值与特征向量，取 $\epsilon = 10^{-6}$。

主程序(文件名:CJCBJ0.FOR)为

```
      DIMENSION A(5,5),V(5,5)
      DOUBLE PRECISION A,V
      DATA A/10.0,1.0,2.0,3.0,4.0,1.0,9.0,-1.0,2.0,-3.0,
     *       2.0,-1.0,7.0,3.0,-5.0,3.0,2.0,3.0,12.0,-1.0,
     *       4.0,-3.0,-5.0,-1.0,15.0/
      EPS=0.000001
      CALL CJCBJ(A,5,EPS,V)
      WRITE(*,*)
      WRITE(*,20) (A(I,I),I=1,5)
20    FORMAT(1X,D15.6)
      WRITE(*,*)
```

```
      WRITE(*,30) ((V(I,J),J=1,5),I=1,5)
30    FORMAT(1X,5D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
 .699484D+01
 .936555D+01
 .165537D+01
 .158089D+02
 .191754D+02

 .654083D+00 -.521511D-01 -.387297D+00  .623702D+00  .174505D+00
 .199681D+00  .859964D+00  .366221D+00  .159101D+00 -.247303D+00
 .256510D+00 -.505575D+00  .704377D+00  .227297D+00 -.361642D+00
-.660403D+00 -.201167D-03 -.118926D+00  .692684D+00 -.264411D+00
-.174280D+00  .462192D-01  .453423D+00  .232822D+00  .841244D+00
```

第4章 非线性方程与方程组的求解

4.1 求非线性方程实根的对分法

一、功能

用对分法搜索非线性方程 $f(x)=0$ 在指定区间内的全部单实根。

二、方法说明

设非线性方程为 $f(x)=0$，搜索区间为 $[a,b]$。

从端点 $x_0=a$ 开始，以 h 为步长逐步往后搜索。对于每一个小区间 $[x_i,x_{i+1}]$：

若 $f(x_i)=0$，则 x_i 为其一个实根，再从 $x_i+\frac{h}{2}$ 开始往后搜索。

若 $f(x_i)f(x_{i+1})=0$，则 x_{i+1} 为其一个实根，再从 $x_{i+1}+\frac{h}{2}$ 开始往后搜索。

若 $f(x_i)f(x_{i+1})>0$，则说明当前小区间内无实根，从 x_{i+1} 开始再往后搜索。

若 $f(x_i)f(x_{i+1})<0$，则说明在当前小区间内有实根。此时，反复将区间折半，直到小区间长度小于 ε 为止，小区间的中点即为方程的一个实根。然后再从 x_{i+1} 开始往后搜索。其中 ε 为事先给定的精度要求。

以上过程一直作到区间端点 b 为止。

采用本子程序求方程的实根时，要注意步长 h 的选择。若步长 h 选得过大，可能会导致某些实根的丢失；若步长 h 选得过小，则计算工作量太大。

三、子程序语句

SUBROUTINE DDHRT(A,B,H,EPS,X,N,M,F)

四、形参说明

A,B——均为实型变量，输入参数。求根区间的左端点与右端点。

H——实型变量，输入参数。搜索的步长。

EPS——实型变量，输入参数。控制精度要求。

X——实型一维数组，长度为N，输出参数。前M个返回所有的实根。

N——整型变量，输入参数。在区间[A,B]内实根个数的预估值。

M——整型变量，输出参数。返回在区间[A,B]内方程实根的个数。如果返回的M=N，则有可能在区间[A,B]内的实根未搜索完。

F——实型函数子程序名，输入参数。用于计算方程左端的函数值 $f(x)$。在主程序中必须要用外部语句对相应的实参进行说明，并要说明其类型。该子程序由用户自编，其语句形式为

FUNCTION F(X)

其中：X为实型的自变量名，实型函数名返回函数值。

五、子程序(文件名:DDHRT.FOR)

```
      SUBROUTINE DDHRT(A,B,H,EPS,X,N,M,F)
      DIMENSION X(N)
      M=0
      Z=A
      Y=F(Z)
10    IF ((Z.GT.B+H/2.0).OR.(M.EQ.N)) RETURN
      IF (ABS(Y).LT.EPS) THEN
        M=M+1
        X(M)=Z
        Z=Z+H/2.0
        Y=F(Z)
        GOTO 10
      END IF
      Z1=Z+H
      Y1=F(Z1)
      IF (ABS(Y1).LT.EPS) THEN
        M=M+1
        X(M)=Z1
        Z=Z1+H/2.0
        Y=F(Z)
        GOTO 10
      END IF
      IF (Y*Y1.GT.0.0) THEN
        Y=Y1
        Z=Z1
        GOTO 10
      END IF
20    IF (ABS(Z1-Z).LT.EPS) THEN
        M=M+1
        X(M)=(Z1+Z)/2.0
        Z=Z1+H/2.0
        Y=F(Z)
        GOTO 10
      END IF
      Z0=(Z1+Z)/2.0
      Y0=F(Z0)
      IF (ABS(Y0).LT.EPS) THEN
        M=M+1
        X(M)=Z0
        Z=Z0+H/2.0
        Y=F(Z)
```

```
      GOTO 10
     END IF
     IF (Y * Y0.LT.0.0) THEN
      Z1=Z0
      Y1=Y0
     ELSE
      Z=Z0
      Y=Y0
     END IF
     GOTO 20
     END
```

六、例

求方程

$$f(x) = x^6 - 5x^5 + 3x^4 + x^3 - 7x^2 + 7x - 20 = 0$$

在区间[−2,5]内的全部单实根。取步长H=0.2,EPS=0.000001,N=6。

主程序及计算方程左端函数值的子程序(文件名:DDHRT0.FOR)为

```
     DIMENSION X(6)
     EXTERNAL F
     CALL DDHRT(-2.0,5.0,0.2,1.0E-06,X,6,M,F)
     WRITE(*,10) M
10   FORMAT(1X,'M=',I2)
     DO 20 I=1,M
20   WRITE(*,30) I,X(I)
30   FORMAT(1X,'X(',I2,')=',E15.6)
     END

     FUNCTION F(X)
     F=(((((X-5)*X+3)*X+1)*X-7)*X+7)*X-20.0
     RETURN
     END
```

运行结果为:

```
M= 2
X( 1)=  -.140246E+01
X( 2)=   .433376E+01
```

4.2 求非线性方程一个实根的牛顿法

一、功能

用牛顿(Newton)迭代法求方程的一个实根。

二、方法说明

设方程 $f(x) = 0$ 满足:

(1) $f(x)$ 在区间$[a,b]$上的一阶导数$f'(x)$与二阶导数$f''(x)$均存在，且各自保持固定符号；

(2) $f(a)f(b)<0$

(3) $f(x_0)f''(x)>0, x, x_0 \in [a,b]$

则方程 $f(x)=0$ 在区间$[a,b]$上有且只有一个实根，取初值 x_0，由牛顿迭代格式

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}$$

计算得到的序列 $x_0, x_1, \cdots, x_n, \cdots$ 收敛于方程 $f(x)=0$ 的根。

结束迭代的条件为

$$|f(x_{n+1})|<\varepsilon \text{ 和 } |x_{n+1}-x_n|<\varepsilon$$

其中 ε 为事先给定的精度要求。

三、子程序语句

SUBROUTINE DNEWT(X,EPS,FS,L)

四、形参说明

X——实型变量，输入兼输出参数。调用时存放迭代初值；返回方程实根的近似值。

EPS——实型变量，输入参数。控制精度要求。

FS——子程序名，输入参数。用于计算方程左端函数值 $f(x)$ 与一阶导数值 $f'(x)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE FS(X,F,DY,)

其中：X 为实型变量，自变量值；F、DY 均为实型变量，分别返回函数值 $f(x)$ 与一阶导数值 $f'(x)$。

L——整型变量，输出参数。返回迭代次数为 60－L。如果 L=0，则说明求根失败。

五、子程序（文件名：DNEWT.FOR）

```
      SUBROUTINE DNEWT(X,EPS,FS,L)
      L=60
      CALL FS(X,F,DY)
10    IF (ABS(DY)+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,20)
        RETURN
      END IF
20    FORMAT(1X,' ERR')
      X1=X-F/DY
      CALL FS(X1,F,DY)
      IF ((ABS(X1-X).GE.EPS).OR.(ABS(F).GE.EPS)) THEN
        L=L-1
        X=X1
        IF (L.EQ.0) RETURN
        GOTO 10
      END IF
```

```
      X=X1
      RETURN
      END
```

六、例

设方程为

$$f(x) = x^3 - x^2 - 1 = 0$$

求在 $x_0 = 1.0$ 附近的一个实根，取 EPS=10^{-6}。

主程序及计算左端函数值 $f(x)$，一阶导数值 $f'(x)$ 的子程序（文件名：DNEWT0.FOR）为

```
      EXTERNAL FS
      X0=1.5
      EPS=1.0E-06
      CALL DNEWT(X0,EPS,FS,L)
      IF (L.NE.0) THEN
        WRITE(*,10) X0
      END IF
10    FORMAT(1X,'X=',E15.6)
      END

      SUBROUTINE FS(X,F,DY)
      F=X*X*(X-1.0)-1.0
      DY=3.0*X*X-2.0*X
      RETURN
      END
```

运行结果为

```
X=     .146557E+01
```

4.3 求非线性方程一个实根的埃特金迭代法

一、功能

用埃特金(Aitken)迭代法求非线性方程的一个实根。

二、方法说明

设非线性方程为

$$f(x) = 0$$

首先将上述非线性方程化成等价形式

$$x = \varphi(x)$$

然后取初值 x_0，由下列埃特金迭代格式进行迭代：

预报 $\qquad u = \varphi(x_k)$

再预报 $\qquad v = \varphi(u)$

校正 $$x_{k+1} = v - \frac{(v-u)^2}{v-2u+x_k}$$

结束迭代过程的条件为

$$|v-u| < \varepsilon$$

此时 v 即为非线性方程的一个实根近似值。其中 ε 为预先给定的精度要求。

三、子程序语句

SUBROUTINE DATKN(X,EPS,F,L)

四、形参说明

X——双精度实型变量，输入兼输出参数。调用时存放初值，返回时存放满足精度要求的实根近似值。

EPS——实型变量，输入参数。精度要求。

F——双精度实型函数子程序名，输入参数。用于计算非线性方程 $f(x)=0$ 的等价形式 $x=\varphi(x)$ 中的右端函数值 $\varphi(x)$。子程序名的实参在主程序中必须用外部语句进行说明，该子程序的语句形式为

FUNCTION F(X)

其中：X 为双精度实型变量，自变量值；F 为双精度实型函数名，存放函数值 $\varphi(x)$，该子程序由用户自编。

L——整型变量，输入兼输出参数。调用时存放最大迭代次数，返回时存放实际迭代次数。如果返回的实际迭代次数等于预先指定的最大迭代次数，则返回的实根 X 没有满足精度要求，求根失败。

五、子程序（文件名：DATKN.FOR）

```
      SUBROUTINE DATKN(X,EPS,F,L)
      DOUBLE PRECISION X,F,U,V
      M=L
      L=0
10    L=L+1
      U=F(X)
      V=F(U)
      IF (ABS(U-V).LT.EPS*1.0D0) THEN
        X=V
      ELSE
        X=V-(V-U)*(V-U)/(V-2.0*U+X)
        IF (L.NE.M) GOTO 10
      END IF
      RETURN
      END
```

六、例

设非线性方程为

$$x^2+x-6=0$$

其等价形式为

$$x = 6 - x^2$$

用埃特金迭代法求一个实根。取初值 $x_0 = 0.0$，精度要求为 $\varepsilon = 10^{-7}$，最大迭代次数为20。

主程序及计算 $\varphi(x) = 6 - x^2$ 函数值的函数子程序(文件名:DATKN0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION X,F
      X=0.0
      EPS=1.0E-7
      L=20
      WRITE(*,*)
      CALL DATKN(X,EPS,F,L)
      WRITE(*,100) L,X
      WRITE(*,*)
100   FORMAT(1X,'L=',I2,5X,'X=',D13.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION X,F
      F=6.0-X*X
      END
```

运行结果为

```
L= 8      X=  .200000D+01
```

4.4 求非线性方程一个实根的连分式解法

一、功能

用连分式法求非线性方程 $f(x) = 0$ 的一个实根。

二、方法说明

设非线性方程为 $f(x) = 0$，其左端函数 $f(x)$ 的反函数为 $x = F(y)$，并用连分式表示为

$$x = a_0 + \cfrac{y - y_0}{a_1 + \cfrac{y - y_1}{a_2 + \cdots + \cfrac{y - y_{i-1}}{a_i + \cdots}}}$$

其中 $y_i = f(x_i)$。因此，满足方程 $f(x) = 0$ 的解为

$$\alpha = a_0 - \cfrac{y_0}{a_1 - \cfrac{y_1}{a_2 - \cdots - \cfrac{y_{i-1}}{a_i - \cdots}}}$$

具体求解过程如下。

选取初值 x_0, x_1，并计算出 $y_0 = f(x_0)$，$y_1 = f(x_1)$，根据 (x_0, y_0) 与 (x_1, y_1)，分别

计算出

$$a_0 = x_0$$
$$a_1 = (y_1 - y_0)/(x_1 - x_0)$$

由此可以计算出

$$x_2 = a_0 - \frac{y_0}{a_1}$$

一般来说，若已知点列

$$(x_0,y_0),(x_1,y_1),\cdots,(x_{i-1},y_{i-1})$$

并已求得 $a_0,a_1,\cdots,a_{i-1}$。则可以计算出

$$x_i = a_0 - \cfrac{y_0}{a_1 - \cfrac{y_1}{a_2 - \cdots - \cfrac{y_{i-2}}{a_{i-1}}}}$$

及

$$y_i = f(x_i)$$

然后通过下列递推公式再计算出 a_i：

$$u = x_i$$
$$u = (y_i - y_j)/(u - a_j),\ j = 0,1,\cdots,i-1$$
$$a_i = u$$

以上过程一直进行到 $|y_i| < \varepsilon$ 为止。其中 ε 为预先给定的精度要求。

在实际计算过程中，上述的连分式一般作到七节为止。如果此时还不满足精度要求，则将最后得到近似值 x_7 作为新的 x_0 再重新计算。

三、子程序语句

SUBROUTINE　DPQRT(X,EPS,F,L)

四、形参说明

X——实型变量，输入兼输出参数。调用时存放初值，返回方程实根近似值。

EPS——实型变量，输入参数。控制精度要求。

F——实型函数子程序名，输入参数。用于计算方程左端函数值 $f(x)$，在主程序中必须用外部语句对相应的实参进行说明，并要说明其类型。该子程序由用户自编，其语句形式为

FUNCTION　F(X)

其中，X 为实型自变量值，实型函数名 F 返回方程左端函数值。

L——整型变量，输出参数，返回的迭代次数为 10－L。如果返回 L=0，则说明求根失败。

五、子程序(文件名：DPQRT.FOR)

```
      SUBROUTINE DPQRT(X,EPS,F,L)
      DIMENSION A(10),Y(10)
      L=10
10    J=0
```

```
20    IF (J.LE.2) THEN
        Z=X+0.1*J
      ELSE
        Z=H2
      END IF
      Y(J+1)=F(Z)
      H2=Z
      IF (J.EQ.0) THEN
        A(1)=Z
      ELSE
        DO 30 I=1,J
          H2=H2-A(I)
          IF (ABS(H2)+1.0.EQ.1.0) THEN
            H2=SIGN(1.0E+35,H2)
            H2=H2*SIGN(1.0E0,Y(J+1)-Y(I))
          ELSE
            H2=(Y(J+1)-Y(I))/H2
          END IF
30      CONTINUE
        A(J+1)=H2
        H2=0.0
        DO 40 I=J,1,-1
          H2=H2+A(I+1)
          IF (ABS(H2)+1.0.EQ.1.0) THEN
            H2=SIGN(1.0E+35,H2)
            H2=H2*SIGN(1.0E0,-Y(I))
          ELSE
            H2=-Y(I)/H2
          END IF
40      CONTINUE
        H2=H2+A(1)
      END IF
      IF (ABS(Y(J+1)).GT.EPS) THEN
        J=J+1
        IF (J.LE.7) GOTO 20
        X=H2
        L=L-1
        IF (L.NE.0) GOTO 10
      END IF
      X=H2
      RETURN
      END
```

六、例

设非线性方程为

$$f(x) = x^3 - x^2 - 1 = 0$$

求在 $x_0 = 1.0$ 附近的一个实根。取初值 $x_0 = 1.0$, $EPS = 0.000001$。

主程序及计算方程左端函数值的子程序(文件名:DPQRT0.FOR)为

```
      EXTERNAL FA
      X=1.0
      EPS=1.0E-06
      CALL DPQRT(X,EPS,FA,L)
      WRITE(*,10) L,X
10    FORMAT(1X,'L=',I2,5X,'X=',E15.6)
      END

      FUNCTION FA(X)
      FA=X*X*(X-1.0)-1.0
      RETURN
      END
```

运行结果为

```
L=10     X=    .146557E+01
```

4.5 求实系数代数方程全部根的 *QR* 方法

一、功能

用 *QR* 方法求实系数高次多项式方程的全部根。

二、方法说明

设首一多项式方程为

$$P_n(x) = x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n = 0$$

由线性代数可知，$P_n(x) = 0$ 的 n 个根实际上就是实矩阵

$$A = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

的 n 个特征值，矩阵 A 为一个上 H 矩阵，可以直接用 *QR* 方法求出全部特征值。关于用 *QR* 方法求上 H 矩阵全部特征值，请参看第三章 3.4 节的方法说明。

三、子程序语句

SUBROUTINE DQRRT(A,N,XR,XI,EPS,B,L)

四、形参说明

A——双精度实型一维数组，长度为N，输入参数。依次存放N次首一多项式方程

$$x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n = 0$$

中的系数 $a_1, a_2, \cdots, a_{n-1}, a_n$。

N——整型变量，输入参数。首一多项式的次数。

XR，XI——均为双精度实型一维数组，长度为N，输出参数。分别返回方程N个根的实部与虚部。

EPS——实型变量，输入参数。控制精度要求。

B——双精度实型二维数组，体积为N×N。本子程序的工作数组。

L——整型变量，输出参数。若L=0，说明求根失败，否则正常返回。

五、子程序(文件名:DQRRT.FOR)

```
      SUBROUTINE DQRRT(A,N,XR,XI,EPS,B,L)
      DIMENSION B(N,N),XR(N),XI(N),A(N)
      DOUBLE PRECISION B,XR,XI,A
      DO 10 J=1,N
10    B(1,J)=-A(J)
      DO 20 I=2,N
      DO 20 J=1,N
20    B(I,J)=0.0
      DO 30 I=1,N-1
30    B(I+1,I)=1.0
      CALL CHHQR(B,N,XR,XI,EPS,L)
      RETURN
      END
```

六、例

设首一多项式方程为

$$P_6(x) = x^6 - 5x^5 + 3x^4 + x^3 - 7x^2 + 7x - 20 = 0$$

求全部根。取 $\epsilon = 0.000001$。

主程序(文件名:DQRRT0.FOR)为

```
      DIMENSION A(6),XR(6),XI(6),B(6,6)
      DOUBLE PRECISION A,XR,XI,B
      DATA A/-5.0,3.0,1.0,-7.0,7.0,-20.0/
      EPS=1.0E-06
      CALL DQRRT(A,6,XR,XI,EPS,B,L)
      IF (L.NE.0) THEN
        DO 10 I=1,6
10      WRITE(*,20) I,XR(I),XI(I)
      END IF
20    FORMAT(1X,'X(',I2,')=',2D15.6,' J')
      END
```

运行结果为

```
X( 1)=    .433376D+01    .000000D+00 J
X( 2)=   -.140246D+01    .000000D+00 J
X( 3)=    .118397D+01   -.936098D+00 J
X( 4)=    .118397D+01    .936098D+00 J
X( 5)=   -.149622D+00   -.119251D+01 J
X( 6)=   -.149622D+00    .119251D+01 J
```

七、附注

本子程序需要调用 QR 方法求上 H 矩阵全部特征值的子程序 CHHQR，参看 3.4 节。

4.6 求实系数代数方程全部根的牛顿-下山法

一、功能

用牛顿-下山法求实系数代数方程的全部根。

二、方法说明

设实系数代数方程为

$$f(z) = a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_{n-1} z + a_n = 0$$

牛顿-下山法的迭代格式为

$$z_{i+1} = z_i - t f(z_i)/f'(z_i)$$

选取适当的 t 可以保证有

$$|f(z_{i+1})|^2 < |f(z_i)|^2$$

迭代过程一直作到 $|f(z_i)|^2 < \varepsilon$ 为止。

迭代格式在鞍点或接近重根点时，可能因 $f'(z) \approx 0$ 而使 $|f(z)|^2 \neq 0$ 而失败。在本子程序中采用了撒网格的方法。选取适当的 d 与 c，用

$$x_{i+1} = x_i + d\cos(c)$$

$$y_{i+1} = y_i + d\sin(c)$$

计算，使 $|f(z_{i+1})|^2 < |f(z_i)|^2$ 而冲过鞍点或使 $|f(z_{i+1})|^2 < \varepsilon$ 而求得一个根。

每求得一个根 z^* 后，在 $f(z)$ 中劈去因子 $(z - z^*)$，再求另一个根。这个过程一直作到求出全部根为止。

在实际计算时，每求一个根都要作变换

$$z = \sqrt[n-1]{|a_n|}\, z'$$

以便使当 $a_1 = 1$ 时，$|a_n| = 1$，保证寻根在单位圆内进行。

三、子程序语句

SUBROUTINE　DSRRT(A,XR,XI,N,M,L,B)

四、形参说明

A——实型一维数组，长度为 N，输入参数。按降幂排列存放代数方程中的系数 a_1，$a_2, \cdots, a_n$。

XR,XI——均为实型一维数组，长度为 M=N−1，输出参数。分别返回方程 N−1 个

根的实部与虚部。

N——整型变量，输入参数．方程中的项数。

M——整型变量，输入参数。方程的最高次数，即 M=N−1。

L——整型变量，输出参数．若返回 L=0，则说明最高次系数为 0，求根失败；若 L≠0，说明正常返回。

B——实型一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名：DSRRT.FOR)

```
      SUBROUTINE DSRRT(A,XR,XI,N,M,L,B)
      DIMENSION A(N),XR(M),XI(M),B(N)
      IF (ABS(A(1))+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,5)
        RETURN
      END IF
5     FORMAT(1X,' ERR')
      L=1
      K=M
      IS=0
      W=1.0
      DO 10 I=1,N
10    B(I)=A(I)/A(1)
20    PP=ABS(B(K+1))
      IF (PP.LT.1.0E-12) THEN
        XR(K)=0.0
        XI(K)=0.0
        K=K-1
        IF (K.EQ.1) THEN
          XR(K)=-B(2)*W/B(1)
          XI(K)=0.0
          RETURN
        END IF
        GOTO 20
      END IF
      Q=PP**(1.0/K)
      P=Q
      W=W*P
      DO 30 I=1,K
        B(I+1)=B(I+1)/Q
        Q=Q*P
30    CONTINUE
      X=0.0001
```

```
      X1=X
      Y=0.2
      Y1=Y
      G=1.0E+37
      DX=1.0
40    U=B(1)
      V=0.0
      DO 50 I=1,K
        P=U * X1
        Q=V * Y1
        PQ=(U+V) * (X1+Y1)
        U=P-Q+B(I+1)
        V=PQ-P-Q
50    CONTINUE
      G1=U * U+V * V
      IF (G1.LT.G) GOTO 105
      IF (IS.NE.0) GOTO 80
60    T=T/1.67
      X1=X-T * DX
      Y1=Y-T * DY
      IF (K.GE.50) THEN
        P=SQRT(X1 * X1+Y1 * Y1)
        Q=EXP(85.0/K)
        IF (P.GE.Q) GOTO 60
      END IF
      IF (T.GE.1.0E-03) GOTO 40
      IF (G.LE.1.0E-18) GOTO 90
65    IS=1
      DD=SQRT(DX * DX+DY * DY)
      IF (DD.GT.1.0) DD=1.0
      DC=6.28/(K+4.5)
70    C=0.0
80    C=C+DC
      DX=DD * COS(C)
      DY=DD * SIN(C)
      X1=X+DX
      Y1=Y+DY
      IF (C.LE.6.29) GOTO 40
      DD=DD/1.67
      IF (DD.GT.1.0E-07) GOTO 70
90    IF (ABS(Y).LE.1.0E-06) THEN
        P=-X
```

```
           Y=0.0
           Q=0.0
         ELSE
           P=-2.0*X
           Q=X*X+Y*Y
           XR(K)=X*W
           XI(K)=-Y*W
           K=K-1
         END IF
         DO 100 I=1,K
           B(I+1)=B(I+1)-B(I)*P
           B(I+2)=B(I+2)-B(I)*Q
100      CONTINUE
         XR(K)=X*W
         XI(K)=Y*W
         K=K-1
         IF (K.EQ.1) THEN
           XR(K)=-B(2)*W/B(1)
           XI(K)=0.0
           RETURN
         END IF
         GOTO 20
105      G=G1
         X=X1
         Y=Y1
         IS=0
         IF (G.LE.1.0E-22) GOTO 90
         U1=K*B(1)
         V1=0.0
         DO 110 I=2,K
           P=U1*X
           Q=V1*Y
           PQ=(U1+V1)*(X+Y)
           U1=P-Q+(K-I+1)*B(I)
           V1=PQ-P-Q
110      CONTINUE
         P=U1*U1+V1*V1
         IF (P.LE.1.0E-20) GOTO 65
         DX=(U*U1+V*V1)/P
         DY=(U1*V-V1*U)/P
         T=1.0+4.0/K
         GOTO 60
```

```
      END
```

六、例

求实系数代数方程

$$f(x) = x^6 - 5x^5 + 3x^4 + x^3 - 7x^2 + 7x - 20 = 0$$

的全部根。

主程序(文件名:DSRRT0.FOR)为

```
      DIMENSION A(7),XR(6),XI(6),B(7)
      DATA A/1.0,-5.0,3.0,1.0,-7.0,7.0,-20.0/
      N=7
      M=6
      CALL DSRRT(A,XR,XI,N,M,L,B)
      IF (L.NE.0) THEN
        DO 10 I=1,M
10      WRITE(*,20) I,XR(I),XI(I)
      END IF
20    FORMAT(1X,'X(',I2,1X,')=',E13.6,2X,'J',2X,E13.6)
      END
```

运行结果为

```
X( 1 )=  .433376E+01  J   .000000E+00
X( 2 )=-.149622E+00  J   .119251E+01
X( 3 )=-.149622E+00  J  -.119251E+01
X( 4 )=-.140246E+01  J   .000000E+00
X( 5 )=  .118398E+01  J   .936099E+00
X( 6 )=  .118398E+01  J  -.936099E+00
```

4.7 求复系数代数方程全部根的牛顿-下山法

一、功能

用牛顿-下山法求复系数代数方程

$$f(z) = a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_{n-1} z + a_n = 0$$

的全部根。

二、方法说明

同4.6节。

三、子程序语句

SUBROUTINE DCSRT(AR,AI,XR,XI,N,M,L,BR,BI)

四、形参说明

AR,AI——均为实型一维数组,长度为N,输入参数。按降幂排列分别存放代数方程系数 $a_1, a_2, \cdots, a_n$ 的实部与虚部。

XR,XI——均为实型一维数组,长度为M=N-1,输出参数。分别返回方程N-1个

根的实部与虚部。

N——整型变量，输入参数。方程的项数。

M——整型变量，输入参数。方程的最高次数，即 M=N−1。

L——整型变量，输出参数。若返回 L=0，则说明最高次系数为 0，求根失败；若返回 L≠0，说明正常返回。

BR，BI——均为实型一维数组，长度为 N。本子程序的工作数组。

五、子程序(文件名：DCSRT.FOR)

```
      SUBROUTINE DCSRT(AR,AI,XR,XI,N,M,L,BR,BI)
      DIMENSION AR(N),AI(N),XR(M),XI(M),BR(N),BI(N)
      K=M
      IS=0
      W=1.0
      P=SQRT(AR(1)*AR(1)+AI(1)*AI(1))
      IF (P+1.0.EQ.1.0) THEN
        L=0
        WRITE(*,5)
        RETURN
      END IF
5     FORMAT(1X,' ERR')
      L=1
      DO 10 I=1,N
        BR(I)=AR(I)/P
        BI(I)=AI(I)/P
10    CONTINUE
20    PP=SQRT(BR(K+1)*BR(K+1)+BI(K+1)*BI(K+1))
      IF (PP.LT.1.0E-12) THEN
        XR(K)=0.0
        XI(K)=0.0
        K=K-1
        IF (K.EQ.1) THEN
          P=BR(1)*BR(1)+BI(1)*BI(1)
          XR(K)=-W*(BR(1)*BR(2)+BI(1)*BI(2))/P
          XI(K)=W*(BR(2)*BI(1)-BR(1)*BI(2))/P
          RETURN
        END IF
        GOTO 20
      END IF
      Q=PP**(1.0/K)
      P=Q
      W=W*P
      DO 30 I=1,K
```

```
      BR(I+1)=BR(I+1)/Q
      BI(I+1)=BI(I+1)/Q
      Q=Q*P
30    CONTINUE
      X=0.0001
      X1=X
      Y=0.2
      Y1=Y
      G=1.0E+37
      DX=1.0
40    U=BR(1)
      V=BI(1)
      DO 50 I=1,K
        P=U*X1
        Q=V*Y1
        PQ=(U+V)*(X1+Y1)
        U=P-Q+BR(I+1)
        V=PQ-P-Q+BI(I+1)
50    CONTINUE
      G1=U*U+V*V
      IF (G1.LT.G) GOTO 105
      IF (IS.NE.0) GOTO 80
60    T=T/1.67
      X1=X-T*DX
      Y1=Y-T*DY
      IF (K.GE.30) THEN
        P=SQRT(X1*X1+Y1*Y1)
        Q=EXP(75.0/K)
        IF (P.GE.Q) GOTO 60
      END IF
      IF (T.GE.1.0E-03) GOTO 40
      IF (G.LE.1.0E-18) GOTO 90
65    IS=1
      DD=SQRT(DX*DX+DY*DY)
      IF (DD.GT.1.0) DD=1.0
      DC=6.28/(K+4.5)
70    C=0.0
80    C=C+DC
      DX=DD*COS(C)
      DY=DD*SIN(C)
      X1=X+DX
      Y1=Y+DY
```

```
      IF (C.LE.6.28) GOTO 40
      DD=DD/1.67
      IF (DD.GT.1.0E-07) GOTO 70
90    DO 100 I=1,K
        BR(I+1)=BR(I+1)+BR(I)*X-BI(I)*Y
        BI(I+1)=BI(I+1)+BR(I)*Y+BI(I)*X
100   CONTINUE
      XR(K)=X*W
      XI(K)=Y*W
      K=K-1
      IF (K.EQ.1) THEN
        P=BR(1)*BR(1)+BI(1)*BI(1)
        XR(K)=-W*(BR(1)*BR(2)+BI(1)*BI(2))/P
        XI(K)=W*(BR(2)*BI(1)-BR(1)*BI(2))/P
        RETURN
      END IF
      GOTO 20
105   G=G1
      X=X1
      Y=Y1
      IS=0
      IF (G.LE.1.0E-22) GOTO 90
      U1=K*BR(1)
      V1=K*BI(1)
      DO 110 I=2,K
        P=U1*X
        Q=V1*Y
        PQ=(U1+V1)*(X+Y)
        U1=P-Q+(K-I+1)*BR(I)
        V1=PQ-P-Q+(K-I+1)*BI(I)
110   CONTINUE
      P=U1*U1+V1*V1
      IF (P.LE.1.0E-20) GOTO 65
      DX=(U*U1+V*V1)/P
      DY=(U1*V-V1*U)/P
      T=1.0+4.0/K
      GOTO 60
      END
```

六、例

求复系数代数方程

$$f(z) = z^5 + (3 + 2j)z^4 - 0.01jz^3 + (4.9 - 19j)z^2 + 21.33z + (0.1 - 100j) = 0$$

的全部根。

主程序(文件名:DCSRT0.FOR)为

```
      DIMENSION AR(6),AI(6),XR(5),XI(5),BR(6),BI(6)
      DATA AR/1.0,3.0,0.0,4.9,21.33,0.1/
      DATA AI/0.0,2.0,-0.01,-19.0,0.0,-100.0/
      N=6
      M=5
      CALL DCSRT(AR,AI,XR,XI,N,M,L,BR,BI)
      IF (L.NE.0) THEN
        DO 10 I=1,M
10      WRITE(*,20) I,XR(I),XI(I)
      END IF
20    FORMAT(1X,'X(',I2,1X,')=',E13.6,2X,'J',2X,E13.6)
      END
```

运行结果为

```
X( 1 )= -.216668E+01  J  -.276720E+01
X( 2 )=  .209386E+01  J   .110221E+01
X( 3 )= -.343095E+01  J   .304084E+00
X( 4 )=  .543454E+00  J  -.218745E+01
X( 5 )= -.396854E-01  J   .154836E+01
```

4.8 求非线性方程组一组实根的梯度法

一、功能

用梯度法(即最速下降法)求实函数方程组

$$f_i(x_1,x_2,\cdots,x_n)=0, i=1,2,\cdots,n$$

的一组实根。

二、方法说明

设非线性方程组为

$$f_i(x_1,x_2,\cdots,x_n)=0, i=1,2,\cdots,n$$

定义目标函数为

$$F=F(x_1,x_2,\cdots,x_n)=\sum_{i=1}^{n}f_i^2$$

则梯度法的计算过程如下。

(1) 选定一组初值

$$x_1,x_2,\cdots,x_n$$

(2) 计算目标函数值

$$F=F(x_1,x_2,\cdots,x_n)=\sum_{i=1}^{n}f_i^2$$

(3) 若 $F<\varepsilon$，则 $X=(x_1,x_2,\cdots,x_n)^T$ 即为方程组的一组实根，过程结束；否则继续。

(4) 计算目标函数在 $(x_1,x_2,\cdots,x_n)$ 点的偏导数

$$\frac{\partial F}{\partial x_i}=2\sum_{j=1}^{n}f_j\frac{\partial f_j}{\partial x_i},\quad i=1,2,\cdots,n$$

再计算

$$D=\sum_{j=1}^{n}\left(\frac{\partial F}{\partial x_j}\right)^2$$

(5) 计算

$$x_i-\lambda\frac{\partial F}{\partial x_i}\Rightarrow x_i,\quad i=1,2,\cdots,n$$

其中 $\lambda=F/D$

从(2)开始重复计算，直到满足精度要求为止。

三、子程序语句

SUBROUTINE DSNSE(N,EPS,X,Y,FS,L)

四、形参说明

N——整型变量，输入参数。方程的个数。

EPS——实型变量，输入参数。控制精度要求。

X——双精度实型一维数组，长度为 N，输入兼输出参数。调用时存放一组初值，返回一组实数解。

Y——双精度实型一维数组，长度为 N。本子程序的工作数组。

FS——子程序名，输入参数，用于计算目标函数值 $F(x_1,x_2,\cdots,x_n)$ 及 N 个偏导数值 $\frac{\partial F}{\partial x_i}(i=1,2,\cdots,N)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE FS(X,N,F,Y)

其中：X 为双精度实型一维数组，长度为 N，存放一组自变量值；F 为双精度实型变量，返回目标函数值 $F=\sum_{i=1}^{n}f_i^2$；Y 为双精度实型一维数组，长度为 N，返回目标函数的 N 个偏导数值 $\frac{\partial F}{\partial x_i}(i=1,2,\cdots,N)$。

L——整型变量，输出参数。若返回 L<0 说明在迭代过程中遇到了目标函数的局部极值点，即

$$D=\sum_{i=1}^{n}\left(\frac{\partial F}{\partial x_i}\right)^2=0$$

子程序工作失败；若 L=0，说明迭代可能不收敛；若 L>0，说明正常返回。

五、子程序(文件名：DSNSE.FOR)

```
SUBROUTINE DSNSE(N,EPS,X,Y,FS,L)
DIMENSION X(N),Y(N)
DOUBLE PRECISION X,Y,F,D,S
L=500
```

```
5       CALL FS(X,N,F,Y)
        IF (F.GE.EPS) THEN
          L=L-1
          IF (L.EQ.0) RETURN
          D=0.0
          DO 20 J=1,N
20        D=D+Y(J)*Y(J)
          IF (D+1.0.EQ.1.0) THEN
            L=-1
            RETURN
          END IF
          S=F/D
          DO 30 I=1,N
30        X(I)=X(I)-S*Y(I)
          GOTO 5
        END IF
        RETURN
        END
```

六、例

设非线性方程组为

$$\begin{cases} f_1 = x_1 - 5x_2^2 + 7x_3^2 + 12 = 0 \\ f_2 = 3x_1x_2 + x_1x_3 - 11x_1 = 0 \\ f_3 = 2x_2x_3 + 40x_1 = 0 \end{cases}$$

取初值(1.5,6.5,−5.0)，$\epsilon = 0.000001$，求一组实数解。

主程序及计算目标函数值、偏导数值的子程序(文件名:DSNSE0.FOR)为

```
      EXTERNAL FS
      DIMENSION X(3),Y(3)
      DOUBLE PRECISION X,Y
      DATA X/1.5,6.5,-5.0/
      EPS=1.0E-06
      CALL DSNSE(3,EPS,X,Y,FS,L)
      IF (L.GT.0) THEN
        DO 10 I=1,3
10      WRITE(*,20) I,X(I)
      END IF
20    FORMAT(1X,'X(',I2,')=',D15.6)
      END

      SUBROUTINE FS(X,N,F,Y)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y,F,F1,F2,F3,DF1,DF2,DF3
```

```
      F1=X(1)-5*X(2)*X(2)+7*X(3)*X(3)+12.0
      F2=3*X(1)*X(2)+X(1)*X(3)-11.0*X(1)
      F3=2.0*X(2)+X(3)+40.0*X(1)
      F=F1*F1+F2*F2+F3*F3
      DF1=1.0
      DF2=3.0*X(2)+X(3)-11.0
      DF3=40.0
      Y(1)=2.0*(F1*DF1+F2*DF2+F3*DF3)
      DF1=10.0*X(2)
      DF2=3.0*X(1)
      DF3=2.0*X(3)
      Y(2)=2.0*(F1*DF1+F2*DF2+F3*DF3)
      DF1=14.0*X(3)
      DF2=X(1)
      DF3=2.0*X(2)
      Y(3)=2.0*(F1*DF1+F2*DF2+F3*DF3)
      RETURN
      END
```

运行结果为

```
X(1)=    .100019D+01
X(2)=    .500043D+01
X(3)=  -.400039D+01
```

本问题的准确解为

$x_1 = 1,\ x_2 = 5,\ x_3 = -4$

4.9 求非线性方程组一组实根的拟牛顿法

一、功能

用拟牛顿法求非线性方程组

$$f_i(x_1,x_2,\cdots,x_n) = 0,\ i = 1,2,\cdots,n$$

的一组实数解。

二、方法说明

设非线性方程组为

$$f_i(X) = 0,\ i = 1,2,\cdots,n$$

其中 $X = (x_1,x_2,\cdots,x_n)^T$。并设 $X^{(k)} = (x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^T$ 为第 k 次迭代近似值，由牛顿法可计算第 $k+1$ 次迭代值。即

$$X^{(k+1)} = X^{(k)} - F(X^{(k)})^{-1}f(X^{(k)})$$

其中

$$f(X^{(k)}) = (f_1^{(k)},f_2^{(k)},\cdots,f_n^{(k)})^T$$

$$f_i^{(k)} = f_i(X^{(k)}),\ i = 1,2,\cdots,n$$

$F(X)$为雅可比矩阵，即

$$F(X) = \begin{bmatrix} \frac{\partial f_1(X)}{\partial x_1} & \frac{\partial f_1(X)}{\partial x_2} & \cdots & \frac{\partial f_1(X)}{\partial x_n} \\ \frac{\partial f_2(X)}{\partial x_1} & \frac{\partial f_2(X)}{\partial x_2} & \cdots & \frac{\partial f_2(X)}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n(X)}{\partial x_1} & \frac{\partial f_n(X)}{\partial x_2} & \cdots & \frac{\partial f_n(X)}{\partial x_n} \end{bmatrix}$$

令

$$\delta^{(k)} = F(X^{(k)})^{-1} f(X^{(k)})$$

其中

$$\delta^{(k)} = (\delta_1^{(k)}, \delta_2^{(k)}, \cdots, \delta_n^{(k)})^T$$

则有

$$F(X^{(k)})\delta^{(k)} = f(X^{(k)})$$
$$X^{(k+1)} = X^{(k)} - \delta^{(k)}$$

若在雅可比矩阵中用差商代替偏导数，即

$$\frac{\partial f_i(X^{(k)})}{\partial x_j} \approx \frac{f_i(X_j^{(k)}) - f_i(X^{(k)})}{h}$$

其中 h 足够小，且

$$f_i(X_j^{(k)}) = f_i(x_1^{(k)}, \cdots, x_{j-1}^{(k)}, x_j^{(k)} + h, x_{j+1}^{(k)}, \cdots, x_n^{(k)})$$

则有

$$\sum_{j=1}^{n} f_i(X_j^{(k)}) z_j^{(k)} = f_i(X^{(k)}), \ i = 1, 2, \cdots, n$$

其中

$$z_j^{(k)} = \frac{\delta_j^{(k)}}{h + \sum_{r=1}^{n} \delta_r^{(k)}}, \ j = 1, 2, \cdots, n$$

综上所述，拟牛顿法求解非线性方程组的计算过程如下。

取初值 $X = (x_1, x_2, \cdots, x_n)^T$, $h > 0$, $0 < t < 1$。

(1) 计算 $f_i(X) \Rightarrow B(i)$, $i = 1, 2, \cdots, n$。

(2) 若 $\max\limits_{1 \leqslant i \leqslant n} |B(i)| < \varepsilon$，则方程组的一组实数解即为

$$X = (x_1, x_2, \cdots, x_n)^T$$

计算过程结束。

(3) 计算

$$f_i(X_j) \Rightarrow A(i, j), \ i, j = 1, 2, \cdots, n$$

其中 $X_j = (x_1, x_2, \cdots, x_{j-1}, x_j + h, x_{j+1}, \cdots, x_n)^T$。

(4) 解方程组

$$AZ = B$$

其中 $Z = (z_1, z_2, \cdots, z_n)^T$，且计算

$$\beta = 1 - \sum_{j=1}^{n} z_j$$

(5) 计算

$$x_i - hz_i/\beta \Rightarrow x_i,\ i = 1,2,\cdots,n$$

(6) $t * h \Rightarrow h$，转(1)。

以上过程一直作到满足精度要求为止。

在使用本方法时，可能会遇到下列几种失败的情形：

迭代次数太多，可能不收敛；

线性方程组 $AZ = B$ 奇异；

$\beta = 0$，即 $\sum_{j=1}^{n} z_j = 1$。

在这种情况下，可以采取以下措施再试一试：

放宽精度要求 ε；

适当改变 t 与 h 的初值；

改变 X 的初值；

改变方程组的顺序等。

三、子程序语句

SUBROUTINE DNETN(N,X,Y,EPS,FS,T,H,A,B,L,JS)

四、形参说明

N——整型变量，输入参数。方程个数。

X——双精度实型一维数组，长度为N，输入兼输出参数。调用时存放初值；返回方程组的一组实数解。

Y——双精度实型一维数组，长度为N。本子程序的工作数组，用于存放方程组的左端函数值。

EPS——实型变量，输入参数。控制精度要求。

FS——子程序名，输入参数。用于计算N个方程中的左端函数值 $f_i(x_1,x_2,\cdots,x_n)(i=1,2,\cdots,N)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE FS(X,Y,N)

其中：X为双精度实型一维数组，长度为N，存放N个自变量 $x_1,x_2,\cdots,x_n$；Y为双精度实型一维数组，长度为N，返回N个方程的左端函数值 $f_i(x_1,x_2,\cdots,x_n)(i=1,2,\cdots,N)$。

T——双精度实型变量，输入参数。控制H大小的变量，要求0<T<1。

H——双精度实型变量，输入参数。增量初值，在本子程序工作时将逐步变小。

A——双精度实型二维数组，体积为N×N。本子程序的工作数组。

B——双精度实型一维数组，长度为N。本子程序的工作数组。

L——整型变量，输出参数。若返回L>0，说明正常返回；若L=0，说明迭代了100次还未满足精度要求，工作失败；若L=−1，说明线性方程组 $AZ = B$ 奇异，工作失败；若L

$=-2$,说明 $\beta=0$,即 $\sum_{j=1}^{n} x_j = 1$,工作失败。

JS——整型一维数组,长度为N。本子程序的工作数组。

五、子程序(文件名:DNETN.FOR)

```
      SUBROUTINE DNETN(N,X,Y,EPS,FS,T,H,A,B,L,JS)
      DIMENSION X(N),Y(N),A(N,N),B(N),JS(N)
      DOUBLE PRECISION X,Y,A,B,T,H,AM,Z,BETA,D
      L=100
10    CALL FS(X,B,N)
      AM=0.0
      DO 20 I=1,N
        IF (ABS(B(I)).GT.AM) AM=ABS(B(I))
20    CONTINUE
      IF (AM.LT.EPS) RETURN
      L=L-1
      IF (L.EQ.0) THEN
        WRITE(*,100)
        RETURN
      END IF
100   FORMAT(1X,' FAIL')
      DO 40 J=1,N
        Z=X(J)
        X(J)=X(J)+H
        CALL FS(X,Y,N)
        DO 30 I=1,N
30      A(I,J)=Y(I)
        X(J)=Z
40    CONTINUE
      CALL AGAUS(A,B,N,Y,K,JS)
      IF (K.EQ.0) THEN
        L=-1
        RETURN
      END IF
      BETA=1.0
      DO 50 I=1,N
50    BETA=BETA-Y(I)
      IF (ABS(BETA)+1.0.EQ.1.0) THEN
        L=-2
        WRITE(*,100)
        RETURN
      END IF
```

```
      D=H/BETA
      DO 60 I=1,N
60    X(I)=X(I)-D*Y(I)
      H=T*H
      GOTO 10
      END
```

六、例

设非线性方程组为

$$\begin{cases} f_1 = x_1^2 + x_2^2 + x_3^2 - 1.0 = 0 \\ f_2 = 2x_1^2 + x_2^2 - 4x_3 = 0 \\ f_3 = 3x_1^2 - 4x_2 + x_3^2 = 0 \end{cases}$$

取初值 $(1.0,1.0,1.0)^T$，$T=0.1$，$H=0.1$，$\varepsilon=10^{-7}$，求一组实数解。

主程序及计算各方程左端函数值的子程序(文件名：DNETN0.FOR)为

```
      EXTERNAL FS
      DIMENSION X(3),Y(3),A(3,3),B(3),JS(3)
      DOUBLE PRECISION X,Y,A,B,T,H
      DATA X/1.0,1.0,1.0/
      EPS=1.0E-07
      T=0.1
      H=0.1
      CALL DNETN(3,X,Y,EPS,FS,T,H,A,B,L,JS)
      WRITE(*,*)
      WRITE(*,30) L
      WRITE(*,*)
      DO 10 I=1,3
10    WRITE(*,20) I,X(I)
20    FORMAT(1X,'X(',I2,')=',D15.6)
30    FORMAT(1X,'L=',I4)
      WRITE(*,*)
      END

      SUBROUTINE FS(X,Y,N)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y
      Y(1)=X(1)*X(1)+X(2)*X(2)+X(3)*X(3)-1.0
      Y(2)=2.0*X(1)*X(1)+X(2)*X(2)-4.0*X(3)
      Y(3)=3.0*X(1)*X(1)-4.0*X(2)+X(3)*X(3)
      RETURN
      END
```

运行结果为

```
L=   95
```

X(1)=　.785197D+00

X(2)=　.496611D+00

X(3)=　.369923D+00

其中L=95,说明迭代了100−95=5次就满足了精度要求。

七、附注

本子程序需要调用全选主元高斯消去法求解线性代数方程组的子程序AGAUS,参看1.1节。

4.10　求非线性方程组最小二乘解的广义逆法

一、功能

利用广义逆求解无约束条件下的优化问题

$$f_i(x_1,x_2,\cdots,x_n)=0,\quad i=1,2,\cdots,m$$

其中$m\geqslant n$。当$m=n$时,即为求解非线性方程组。

二、方法说明

设非线性方程组为

$$f_i(x_1,x_2,\cdots,x_n)=0,\quad i=1,2,\cdots,m$$

其中$m\geqslant n$,雅可比矩阵为

$$A=\begin{bmatrix}\dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \cdots & \dfrac{\partial f_1}{\partial x_n}\\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \cdots & \dfrac{\partial f_2}{\partial x_n}\\ \vdots & \vdots & & \vdots\\ \dfrac{\partial f_m}{\partial x_1} & \dfrac{\partial f_m}{\partial x_2} & \cdots & \dfrac{\partial f_m}{\partial x_n}\end{bmatrix}$$

计算非线性方程组最小二乘解的迭代公式为

$$X^{(k+1)}=X^{(k)}-\alpha_k Z^{(k)}$$

其中$Z^{(k)}$为线性方程组

$$A^{(k)}Z^{(k)}=F^{(k)}$$

的线性最小二乘解,即

$$Z^{(k)}=(A^{(k)})^+F^{(k)}$$

式中$A^{(k)}$为k次迭代值$X^{(k)}$的雅可比矩阵,$F^{(k)}$为k次迭代值的方程组左端函数值,即

$$F^{(k)}=(f_1^{(k)},f_2^{(k)},\cdots,f_m^{(k)})^T$$

$$f_i^{(k)}=f_i(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)}),i=1,2,\cdots,m$$

α_k是为使α的一元函数

$$\sum_{i=1}^{m}(f_i^{(k+1)})^2$$

达到极小的点。在本子程序中用极值有理法计算α_k。

三、子程序语句

SUBROUTINE DNGIN(M,N,X,EPS1,EPS2,F,FJ,L,P,D,PP,DX,U,V,KA,S,E,WORK)

四、形参说明

M——整型变量,输入参数。方程个数。

N——整型变量,输入参数。未知数个数。

X——双精度实型一维数组,长度为N,输入兼输出参数。调用时存放初值$X^{(0)}$,要求各分量不全为0;返回最小二乘解,当M=N时,即为非线性方程组的解。

EPS1——实型变量,输入参数。控制最小二乘解的精度要求。

EPS2——实型变量,输入参数。用于奇异值分解的控制精度要求。

F——子程序名,输入参数。用于计算非线性方程组中各方程的左端函数值$f_i(x_1,x_2,\cdots,x_n)$ $(i=1,2,\cdots,M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE F(M,N,X,D)

其中:X为双精度实型一维数组,长度为N,N个自变量值$x_1,x_2,\cdots,x_n$;D为双精度实型一维数组,长度为M,返回M个方程的左端函数值$f_i(x_1,x_2,\cdots,x_n)(i=1,2,\cdots,M)$。

FJ——子程序名,输入参数。用于计算雅可比矩阵。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE FJ(M,N,X,P)

其中:X为双精度实型一维数组,长度为N,存放N个自变量值$x_1,x_2,\cdots,x_n$;P为双精度实型二维数组,体积为M×N,返回雅可比矩阵。

L——整型变量,输出参数。若返回L=0,说明正常返回,程序工作正常;若L>0,说明奇异值分解失败;若L<0,说明迭代了30次还未满足精度要求。

P——双精度实型二维数组,体积为M×N。本子程序的工作数组,用于存放雅可比矩阵。

D——双精度实型一维数组,长度为M。本子程序的工作数组,用于存放各方程的左端函数值。

PP——双精度实型二维数组,体积为N×M。本子程序的工作数组,用于存放雅可比矩阵的广义逆。

DX——双精度实型一维数组,长度为N。本子程序的工作数组,用于存放线性最小二乘解。

U——双精度实型二维数组,体积为M×M。本子程序的工作数组,用于奇异值分解。

V——双精度实型二维数组,体积为N×N。本子程序的工作数组,用于奇异值分解。

KA——整型变量,输入参数。KA=max{M,N}+1。

S,E,WORK——均为双精度实型一维数组,长度为KA。本子程序的工作数组。

五、子程序(文件名:DNGIN.FOR)

```
      SUBROUTINE DNGIN(M,N,X,EPS1,EPS2,F,FJ,L,P,D,PP,DX,U,
     *                 V,KA,S,E,WORK)
```

```
      DIMENSION X(N),P(M,N),D(M),PP(N,M),DX(N),WORK(KA)
      DIMENSION U(M,M),V(N,N),Y(10),B(10),S(KA),E(KA)
      DOUBLE PRECISION X,P,D,ALPHA,PP,DX,U,V,Y1,Y2,Y0,
     *                 Z,H2,H1,Y,B,S,E,WORK
      L=30
      ALPHA=1.0
5     CALL F(M,N,X,D)
      CALL FJ(M,N,X,P)
      CALL AGMIV(M,N,P,D,PP,DX,LL,EPS2,U,V,KA,S,E,WORK)
      IF (LL.NE.0) THEN
        L=1
        RETURN
      END IF
      J=0
10    IF (J.LE.2) THEN
        Z=ALPHA+J*0.01
      ELSE
        Z=H2
      END IF
      DO 20 I=1,N
20    V(I,1)=X(I)-Z*DX(I)
      CALL F(M,N,V,D)
      Y1=0.0
      DO 30 I=1,M
30    Y1=Y1+D(I)*D(I)
      DO 40 I=1,N
40    V(I,1)=X(I)-(Z+0.00001)*DX(I)
      CALL F(M,N,V,D)
      Y2=0.0
      DO 50 I=1,M
50    Y2=Y2+D(I)*D(I)
      Y0=(Y2-Y1)/0.00001
      IF (ABS(Y0).GT.1.0E-10) THEN
        H1=Y0
        H2=Z
        IF (J.EQ.0) THEN
          Y(1)=H1
          B(1)=H2
        ELSE
          Y(J+1)=H1
          DO 60 K=1,J
            H2=H2-B(K)
```

```
      IF (ABS(H2)+1.0.EQ.1.0) THEN
        H2=SIGN(1.0D+35,H2)*SIGN(1.0D0,H1-Y(K))
      ELSE
        H2=(H1-Y(K))/H2
      END IF
60    CONTINUE
      B(J+1)=H2
      H2=0.0
      DO 70 K=J,1,-1
        H2=B(K+1)+H2
        IF (ABS(H2)+1.0.EQ.1.0) THEN
          H2=SIGN(1.0D+35,H2)*SIGN(1.0D0, Y(K))
        ELSE
          H2=-Y(K)/H2
        END IF
70    CONTINUE
      H2=H2+B(1)
    END IF
    J=J+1
    IF (J.LE.7) GOTO 10
    Z=H2
  END IF
  ALPHA=Z
  Y1=0.0
  Y2=0.0
  DO 80 I=1,N
    DX(I)=-ALPHA*DX(I)
    X(I)=X(I)+DX(I)
    Y1=Y1+ABS(DX(I))
    Y2=Y2+ABS(X(I))
80 CONTINUE
  IF (Y1.LT.EPS1*Y2) THEN
    L=0
    RETURN
  END IF
  L=L-1
  IF (L.GE.0) GOTO 5
  RETURN
  END
```

六、例

(1) 求解非线性方程组

$$\begin{cases} x_1^2 + 10x_1x_2 + 4x_2^2 + 0.7401006 = 0 \\ x_1^2 - 3x_1x_2 + 2x_2^2 - 1.0201228 = 0 \end{cases}$$

其中 M=2,N=2。取 EPS1=0.000001,EPS2=0.000001。

主程序及计算各方程左端函数值、计算雅可比矩阵的子程序(文件名:DNGIN0.FOR)为

```
      EXTERNAL F,FJ
      DIMENSION X(2),P(2,2),D(2),PP(2,2),DX(2),U(2,2),V(2,2)
      DIMENSION S(3),E(3),WORK(3)
      DOUBLE PRECISION X,P,D,PP,DX,U,V,S,E,WORK
      DATA X/0.5,-1.0/
      M=2
      N=2
      KA=3
      EPS1=0.000001
      EPS2=0.000001
      CALL DNGIN(M,N,X,EPS1,EPS2,F,FJ,L,P,D,PP,DX,U,V,KA,S,E,WORK)
      WRITE(*,*)
      WRITE(*,5) L
 5    FORMAT(1X,'L=',I4)
      WRITE(*,*)
      IF (L.EQ.0) THEN
        WRITE(*,10) X(1),X(2)
        WRITE(*,*)
      END IF
 10   FORMAT(1X,'X(1)=',D13.6,3X,'X(2)=',D13.6)
      END

      SUBROUTINE F(M,N,X,D)
      DIMENSION X(N),D(M)
      DOUBLE PRECISION X,D
      D(1)=X(1)*X(1)+10.0*X(1)*X(2)+4.0*X(2)*X(2)+0.7401006
      D(2)=X(1)*X(1)-3.0*X(1)*X(2)+2.0*X(2)*X(2)-1.0201228
      RETURN
      END

      SUBROUTINE FJ(M,N,X,P)
      DIMENSION X(N),P(M,N)
      DOUBLE PRECISION X,P
      P(1,1)=2.0*X(1)+10.0*X(2)
      P(1,2)=10.0*X(1)+8.0*X(2)
      P(2,1)=2.0*X(1)-3.0*X(2)
```

```
      P(2,2)=-3.0*X(1)+4.0*X(2)
      RETURN
      END
```

运行结果为

```
L=   0

X(1)=  .376547D+00   X(2)= -.437953D+00
```

其中 L=0 表示工作正常，正常返回。

(2) 求最小二乘解

$$\begin{cases} x_1^2 + 7x_1x_2 + 3x_2^2 + 0.5 = 0 \\ x_1^2 - 2x_1x_2 + x_2^2 - 1.0 = 0 \\ x_1 + x_2 + 1.0 = 0 \end{cases}$$

其中 M=3,N=2。取 EPS1=EPS2=0.000001。

主程序及计算各方程左端函数值、计算雅可比矩阵的子程序(文件名:DNGIN1.FOR)为

```
      EXTERNAL F,FJ
      DIMENSION X(2),P(3,2),D(3),PP(2,3),DX(2),U(3,3),V(2,2)
      DIMENSION S(4),E(4),WORK(4)
      DOUBLE PRECISION X,P,D,PP,DX,U,V,S,E,WORK
      DATA X/1.0,-1.0/
      M=3
      N=2
      KA=4
      EPS1=0.000001
      EPS2=0.000001
      CALL DNGIN(M,N,X,EPS1,EPS2,F,FJ,L,P,D,PP,DX,U,V,KA,S,E,WORK)
      WRITE(*,*)
      WRITE(*,5) L
5     FORMAT(1X,'L=',I4)
      WRITE(*,*)
      IF (L.EQ.0) THEN
        WRITE(*,10) X(1),X(2)
        WRITE(*,*)
      END IF
10    FORMAT(1X,'X(1)=',D13.6,3X,'X(2)=',D13.6)
      END

      SUBROUTINE F(M,N,X,D)
      DIMENSION X(N),D(M)
      DOUBLE PRECISION X,D
      D(1)=X(1)*X(1)+7.0*X(1)*X(2)+3.0*X(2)*X(2)+0.5
```

```
      D(2)=X(1)*X(1)-2.0*X(1)*X(2)+1.0*X(2)*X(2)-1.0
      D(3)=X(1)+X(2)+1.0
      RETURN
      END

      SUBROUTINE FJ(M,N,X,P)
      DIMENSION X(N),P(M,N)
      DOUBLE PRECISION X,P
      P(1,1)=2.0*X(1)+7.0*X(2)
      P(1,2)=7.0*X(1)+6.0*X(2)
      P(2,1)=2.0*X(1)-2.0*X(2)
      P(2,2)=-2.0*X(1)+2.0*X(2)
      P(3,1)=1.0
      P(3,2)=1.0
      RETURN
      END
```

运行结果为

```
L=   0

X(1)=  .378947D+00   X(2)=  -.692576D+00
```

其中 L=0 表示正常返回。

4.11 求非线性方程一个实根的蒙特卡洛法

一、功能

用蒙特卡洛(Monte Carlo)法求非线性方程的一个实根。

二、方法说明

设实函数方程为

$$f(x) = 0$$

蒙特卡洛法求一个实根的过程如下：

选取一个初值 x,并计算 $F_0 = f(x)$。再选取一个 $b > 0$。

在区间$[-b,b]$上反复产生均匀分布的随机数 r,对于每一个 r 计算 $F_1 = f(x+r)$,直到发现一个 r,使 $|F_1| < |F_0|$ 为止,此时令 $x = x+r$, $F_0 = F_1$。如果连续产生了 m 个随机数 r 还不满足 $|F_1| < |F_0|$,则将 b 减半再进行。

重复上述过程,直到 $|F_0| < \varepsilon$ 为止,此时的 x 即为非线性方程 $f(x) = 0$ 的一个实根。

在使用本方法时,如遇到迭代不收敛时,可适当调整 b 与 m 的值 。

三、子程序语句

SUBROUTINE DMTCL(X,B,M,EPS,F)

四、形参说明

X——实型变量,输入兼输出参数。调用时存放初值;返回方程的一个实根。

B——实型变量，输入参数。均匀分布随机数的端点初值。

M——整型变量，输入参数。控制调节 B 的参数。

EPS——实型变量，输入参数。控制精度要求。

F——实型函数子程序名，输入参数。用于计算方程左端函数值 $f(x)$。在主程序中必须用外部语句对相应的实参进行说明，并说明其类型。该子程序由用户自编，其语句形式为

FUNCTION F(X)

其中：X 为实型变量，自变量值；实型函数名 F 返回函数值。

五、子程序（文件名：DMTCL. FOR）

```
      SUBROUTINE DMTCL(X,B,M,EPS,F)
      DOUBLE PRECISION R
      REAL NRND1
      A=B
      K=1
      R=1.0D0
      Y=F(X)
10    IF (A.GE.EPS) THEN
        L=L+1
        X1=-A+2.0*A*NRND1(R)
        X1=X+X1
        Y1=F(X1)
        K=K+1
        IF (ABS(Y1).GE.ABS(Y)) THEN
          IF (K.GT.M) THEN
            K=1
            A=A/2.0
          END IF
          GOTO 10
        ELSE
          K=1
          X=X1
          Y=Y1
          IF (ABS(Y).GE.EPS) GOTO 10
        END IF
      END IF
      END
```

六、例

求实函数方程

$$f(x)=e^{-x^3}-\mathrm{tg}x+800=0$$

在区间 $\left(0,\frac{\pi}{2}\right)$ 内的一个实根。

取初值 X=0.5,B=1.0,M=10,EPS=10^{-5}。

主程序及计算方程左端函数值的子程序(文件名:DMTCL0.FOR)为

```
      EXTERNAL F
      X=0.5
      B=1.0
      M=10
      EPS=1.0E-5
      CALL DMTCL(X,B,M,EPS,F)
      WRITE(*,*)
      WRITE(*,100) X
      WRITE(*,*)
100   FORMAT(5X,'X=',E13.6)
      END

      FUNCTION F(X)
      F=EXP(-X*X*X)-TAN(X)+800.0
      END
```

运行结果为

```
X=  .156955E+01
```

七、附注

本子程序需要调用产生 0 到 1 之间均匀分布的一个随机数的子程序 NRND1,参看 13.1 节。

4.12 求实函数或复函数方程一个复根的蒙特卡洛法

一、功能

用蒙特卡洛(Monte Carlo)法求实函数或复函数方程

$$f(z)=0$$

的一个复根。

二、方法说明

设非线性方程为

$$f(z)=0$$

左端函数 $f(z)$ 的模函数记为 $\|f(z)\|$。

蒙特卡洛法求一个复根的过程如下:

选取一个初值 $z=x+jy$,其中 $j=\sqrt{-1}$,并计算 $F_0=\|f(z)\|$,再选取一个 $b>0$。

在区间$[-b,b]$上反复产生均匀分布的随机数 r_x 与 r_y,对于每一对(r_x,r_y)计算 $F_1=\|f(x+r_x+j(y+r_y))\|$,直到发现一对$(r_x,r_y)$,使 $F_1<F_0$ 为止,此时令 $z=x+r_x+j(y+r_y)$,$F_0=F_1$。如果连续产生了 m 对随机数(r_x,r_y)还不满足 $F_1<F_0$,则将 b 减半再进

行。

重复上述过程，直到 $F_0 < \epsilon$ 为止，此时的 z 即为非线性方程 $f(z)$ 的一个复根。

在使用本方法时，如遇到迭代不收敛时，可适当调整 b 与 m 的值。

本方法可用于求只包含两个未知量的非线性方程组

$$\begin{cases} f_1(x,y) = 0 \\ f_2(x,y) = 0 \end{cases}$$

的一组实根，此时将 x 当作实部，y 当作虚部。

三、子程序语句

SUBROUTINE DCMTC(X,Y,B,M,EPS,F)

四、形参说明

X,Y——均为实型变量，输入兼输出参数。调用时存放复根初值的实部与虚部；返回方程复根的实部与虚部。

B——实型变量，输入参数。均匀分布随机数的端点初值。

M——整型变量，输入参数。控制调节 B 的参数。

EPS——实型变量，输入参数。控制精度要求。

F——实型函数子程序名，输入参数。用于计算方程左端函数值的模 $\|f(z)\|$。在主程序中必须用外部语句对相应的实参进行说明，并说明其类型。该子程序由用户自编，其语句形式为

FUNCTION F(X,Y)

其中：X 与 Y 均为实型变量，复变量的实部与虚部值；实型函数名 F 返回 $f(x+jy)$ 的模。

五、子程序（文件名：DCMTC.FOR）

```
      SUBROUTINE DCMTC(X,Y,B,M,EPS,F)
      DOUBLE PRECISION R
      REAL NRND1
      A=B
      K=1
      R=1.0D0
      Z=F(X,Y)
10    IF (A.GE.EPS) THEN
        L=L+1
        X1=-A+2.0*A*NRND1(R)
        X1=X+X1
        Y1=-A+2.0*A*NRND1(R)
        Y1=Y+Y1
        Z1=F(X1,Y1)
        K=K+1
        IF (Z1.GE.Z) THEN
          IF (K.GT.M) THEN
            K=1
```

```
      A=A/2.0
    END IF
    GOTO 10
  ELSE
    K=1
    X=X1
    Y=Y1
    Z=Z1
    IF (Z.GE.EPS) GOTO 10
  END IF
END IF
END
```

六、例

(1) 求实函数方程

$$f(z) = z^2 - 6z + 13 = 0$$

的一个复根。取初值 $z = 0.5 + 0.5j$，B=1.0，M=10，EPS=10^{-5}。

其中 $f(z) = (x^2 - y^2 - 6x + 13) + j(2xy - 6y)$。

主程序及计算方程左端函数值的模的子程序(文件名:DCMTC0.FOR)为

```
      EXTERNAL F
      X=0.5
      Y=0.5
      B=1.0
      M=10
      EPS=1.0E-5
      CALL DCMTC(X,Y,B,M,EPS,F)
      WRITE(*,*)
      WRITE(*,100) X,Y
      WRITE(*,*)
100   FORMAT(5X,'Z=',E13.6,'+j',E13.6)
      END

      FUNCTION F(X,Y)
      U=X*X-Y*Y-6*X+13
      V=2*X*Y-6*Y
      F=SQRT(U*U+V*V)
      END
```

运行结果为

```
Z=  .300000E+01+j  .200000E+01
```

(2) 求复函数方程

$$f(z) = z^2 + (1 + j)z - 2 + 2j = 0$$

的一个复根。取初值 $z = 0.5 + 0.5j$，B=1.0，M=10，EPS=10^{-6}。

其中 $f(z) = (x^2 - y^2 + x - y - 2) + \mathrm{j}(2xy + x + y + 2)$。

主程序及计算方程左端函数值模的子程序(文件名:DCMTC1.FOR)为

```
      EXTERNAL F
      X=0.5
      Y=0.5
      B=1.0
      M=10
      EPS=1.0E-5
      CALL DCMTC(X,Y,B,M,EPS,F)
      WRITE(*,*)
      WRITE(*,100) X,Y
      WRITE(*,*)
100   FORMAT(5X,'Z=',E13.6,' +j',E13.6)
      END

      FUNCTION F(X,Y)
      U=X*X-Y*Y+X-Y-2
      V=2*X*Y+X+Y+2
      F=SQRT(U*U+V*V)
      END
```

运行结果为

```
Z= -.200000E+01 +j .314415E-05
```

七、附注

本子程序需要调用产生 0 到 1 之间均匀分布随机数的子程序 NRND1,参看 13.1 节。

4.13 求非线性方程组一组实根的蒙特卡洛法

一、功能

用蒙特卡洛(Monte Carlo)法求非线性方程组

$$f_i(x_1, x_2, \cdots, x_n) = 0, \quad i = 1, 2, \cdots, n$$

的一组实根。

二、方法说明

设非线性方程组为

$$f_i(x_1, x_2, \cdots, x_n) = 0, \quad i = 1, 2, \cdots, n$$

定义模函数为

$$\| F \| = \sqrt{\sum_{i=1}^{n} f_i^2}$$

蒙特卡洛法求一组实根的过程如下。

选取初值 $X = (x_1, x_2, \cdots, x_n)^T$,并计算模函数值 $F_0 = \| F \|$。再选取一个 $b > 0$。

在区间$[-b,b]$上反复产生均匀分布的随机数$(r_1,r_2,\cdots,r_n)$，对于每一组$(r_1,r_2,\cdots,r_n)$计算$(x_1+r_1,x_2+r_2,\cdots,x_n+r_n)^T$的模函数值$F_1$，直到使$F_1<F_0$为止，此时令$x_i=x_i+r_i(i=1,2,\cdots,n)$，$F_0=F_1$。如果连续产生$m$组随机数还不满足$F_1<F_0$，则将$b$减半再进行。

重复上述过程，直到$F_0<\varepsilon$为止，此时的$X=(x_1,x_2,\cdots,x_n)^T$即为非线性方程组的一组实根。

在使用本方法时，如遇到迭代不收敛，可适当调整b与m的值。

三、子程序语句

SUBROUTINE DNMTC(X,N,B,M,EPS,F,Y)

四、形参说明

X——实型一维数组，长度为N，输入兼输出参数。调用时存放初值，返回方程组的一组实根。

N——整型变量，输入参数。方程的个数，也是未知数的个数。

B——实型变量，输入参数。随机数的端点初值。

M——整型变量，输入参数。控制调节B的参数。

EPS——实型变量，输入参数。控制精度要求。

F——实型函数子程序名，输入参数。用于计算模函数值$\|F\|=\sqrt{\sum_{i=1}^{n}f_i^2}$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

FUNCTION F(X,N)

其中：X为实型一维数组，长度为N，存放N个自变量值$x_1,x_2,\cdots,x_n$；实型函数名F返回模函数值$\|F\|$。

Y——实型一维数组，长度为N。本子程序的工作数组。

五、子程序(文件名：DNMTC.FOR)

```
      SUBROUTINE DNMTC(X,N,B,M,EPS,F,Y)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION R
      REAL NRND1
      A=B
      K=1
      R=1.0D0
      Z=F(X,N)
10    IF (A.GE.EPS) THEN
        L=L+1
        DO 20 I=1,N
20      Y(I)=-A+2.0*A*NRND1(R)+X(I)
        Z1=F(Y,N)
```

```
      K=K+1
      IF (Z1.GE.Z) THEN
        IF (K.GT.M) THEN
          K=1
          A=A/2.0
        END IF
        GOTO 10
      ELSE
        K=1
        DO 30 I=1,N
30      X(I)=Y(I)
        Z=Z1
        IF (Z.GE.EPS) GOTO 10
      END IF
     END IF
     END
```

六、例

求非线性方程组

$$\begin{cases} 3x_1 + x_2 + 2x_3^2 - 3 = 0 \\ -3x_1 + 5x_2^2 + 2x_1x_3 - 1 = 0 \\ 25x_1x_2 + 20x_3 + 12 = 0 \end{cases}$$

的一组实根。取初值 $X=(0,0,0)^T$，B=2.0，M=10，EPS=10^{-5}。

主程序及计算模函数值的子程序(文件名:DNMTC0.FOR)为

```
      EXTERNAL F
      DIMENSION X(3),Y(3)
      DATA X/3*0.0/
      B=2.0
      N=3
      M=10
      EPS=1.0E-5
      CALL DNMTC(X,N,B,M,EPS,F,Y)
      WRITE(*,*)
      DO 10 I=1,N
10    WRITE(*,100) I,X(I)
      WRITE(*,*)
100   FORMAT(5X,'X(',I2,1X,')=',E13.6)
      END

      FUNCTION F(X,N)
      DIMENSION X(N)
      F1=3*X(1)+X(2)+2*X(3)*X(3)-3.0
```

```
      F2=-3*X(1)+5*X(2)*X(2)+2*X(1)*X(3)-1
      F3=25*X(1)*X(2)+20*X(3)+12
      F=SQRT(F1*F1+F2*F2+F3*F3)
      END
```

运行结果为

```
X( 1 )=  .109341E+01
X( 2 )=-.799474E+00
X( 3 )=  .492661E+00
```

七、附注

本子程序需要调用产生0到1之间均匀分布随机数的子程序NRND1,参看13.1节。

第5章 插 值

5.1 一元全区间不等距插值

一、功能

根据给定结点上的函数值，用拉格朗日(Lagrange)插值公式，计算指定插值点处的函数值。

二、方法说明

在指定插值点 t 的前后各取四个结点，即选取满足

$$x_k < x_{k+1} < x_{k+2} < x_{k+3} < t < x_{k+4} < x_{k+5} < x_{k+6} < x_{k+7}$$

的八个结点 $(x_i, y_i)(i = k, \cdots, k+7)$，用七次拉格朗日插值公式计算指定插值点 t 处的函数近似值，即

$$z = \sum_{i=k}^{k+7} y_i \prod_{j=k, j\neq i}^{k+7} [(t - x_j)/(x_i - x_j)]$$

三、子程序语句

SUBROUTINE ENLGR(X,Y,N,T,Z)

四、形参说明

X,Y——均为双精度实型一维数组，长度为 N，输入参数。给定的 N 个数据点 $(x_i, y_i)(i = 1,2,\cdots,N)$，要求 $x_1 < x_2 < \cdots < x_{n-1} < x_n$。

N——整型变量，输入参数。给定结点的个数。

T——双精度实型变量，输入参数。指定插值点的值。

Z——双精度实型变量，输出参数。返回插值点 T 处的函数近似值。

五、子程序(文件名：ENLGR.FOR)

```
      SUBROUTINE ENLGR(X,Y,N,T,Z)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y,T,Z,S
      Z=0.0
      IF (N.LE.0) RETURN
      IF (N.EQ.1) THEN
        Z=Y(1)
        RETURN
      END IF
      IF (N.EQ.2) THEN
        Z=(Y(1)*(T-X(2))-Y(2)*(T-X(1)))/(X(1)-X(2))
        RETURN
```

```
      END IF
      I=1
10    IF (X(I).LT.T) THEN
        I=I+1
        IF (I.LE.N) GOTO 10
      END IF
      K=I-4
      IF (K.LT.1) K=1
      M=I+3
      IF (M.GT.N) M=N

      DO 30 I=K,M
        S=1.0
        DO 20 J=K,M
          IF (J.NE.I) THEN
            S=S*(T-X(J))/(X(I)-X(J))
          END IF
20      CONTINUE
        Z=Z+S*Y(I)
30    CONTINUE
      RETURN
      END
```

六、例

已知一元不等距列表函数如下：

i	1	2	3	4	5
x_i	0.10	0.15	0.25	0.40	0.50
y_i	0.904837	0.860708	0.778801	0.670320	0.606531
i	6	7	8	9	10
x_i	0.57	0.70	0.85	0.93	1.00
y_i	0.565525	0.496585	0.427415	0.394554	0.367879

用拉格朗日插值公式计算插值点 T=0.63 处的函数近似值。

主程序(文件名:ENLGR0.FOR)为

```
      DIMENSION X(10),Y(10)
      DOUBLE PRECISION X,Y,T,Z
      DATA X/0.10,0.15,0.25,0.40,0.50,0.57,0.70,
     *        0.85,0.93,1.0/
      DATA Y/0.904837,0.860708,0.778801,0.670320,0.606531,
```

```
*          0.565525,0.496585,0.427415,0.384554,0.362879/
      T=0.63
      CALL ENLGR(X,Y,10,T,Z)
      WRITE(*,20) T,Z
20    FORMAT(1X,'T=',F6.3,10X,'Z=',D15.6)
      END
```

运行结果为

```
T=  .630          Z=    .532591D+00
```

5.2 一元全区间等距插值

一、功能

根据给定等距结点上的函数值，计算指定插值点处的函数值。

二、方法说明

同5.1节。

三、子程序语句

SUBROUTINE　EELGR(X1,H,N,Y,T,Z)

四、形参说明

X1——双精度实型变量，输入参数。等距结点中的第一个结点值。

H——双精度实型变量，输入参数。等距结点的步长。

N——整型变量，输入参数。等距结点的个数。

Y——双精度实型一维数组，长度为N，输入参数。存放N个等距结点上的函数值。

T——双精度实型变量，输入参数。指定插值点的值。

Z——双精度实型变量，输出参数。返回指定插值点T处的函数近似值。

五、子程序（文件名：EELGR.FOR）

```
      SUBROUTINE EELGR(X1,H,N,Y,T,Z)
      DIMENSION Y(N)
      DOUBLE PRECISION X1,H,Y,T,Z,S,XX,P,Q
      Z=0.0
      IF (N.LE.0) RETURN
      IF (N.EQ.1) THEN
        Z=Y(1)
        RETURN
      END IF
      IF (N.EQ.2) THEN
        Z=(Y(2)*(T-X1)-Y(1)*(T-X1-H))/H
        RETURN
      END IF
      IF (T.GT.X1) THEN
        P=(T-X1)/H
```

```
        I=P
        Q=1.0*I
        I=I+1
        IF (P.GT.Q) I=I+1
      ELSE
        I=1
      END IF
      K=I-4
      IF (K.LT.1) K=1
      M=I+3
      IF (M.GT.N) M=N
      Z=0.0
      DO 30 I=K,M
        S=1.0
        XX=X1+(I-1)*H
        DO 20 J=K,M
          IF (J.NE.I) THEN
            XJ=X1+(J-1)*H
            S=S*(T-XJ)/(XX-XJ)
          END IF
20      CONTINUE
        Z=Z+S*Y(I)
30    CONTINUE
      RETURN
      END
```

六、例

设函数

$$f(x) = e^{-x}$$

的函数值列表如下：

i	1	2	3	4	5
x_i	0.1	0.2	0.3	0.4	0.5
y_i	0.904837	0.818731	0.740818	0.670320	0.606531
i	6	7	8	9	10
x_i	0.6	0.7	0.8	0.9	1.0
y_i	0.548812	0.496585	0.449329	0.406570	0.367879

计算在插值点 T=0.63 处的函数近似值。

主程序(文件名:EELGR0.FOR)为

```
      DIMENSION Y(10)
      DOUBLE PRECISION Y,H,X1,T,Z
      DATA Y/0.904837,0.818731,0.740818,0.670320,0.606531,
     *        0.548812,0.496585,0.449329,0.406570,0.367879/
      H=0.1
      N=10
      X1=0.1
      T=0.63
      CALL EELGR(X1,H,N,Y,T,Z)
      WRITE(*,20) T,Z
20    FORMAT(1X,'T=',F6.3,10X,'Z=',D15.6)
      END
```

运行结果为

```
T=  .630          Z=     .532592D+00
```

5.3 一元三点不等距插值

一、功能

根据给定结点上的函数值，用抛物插值计算指定插值点处的函数值。

二、方法说明

设给定 n 个不等距结点

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

及其相应的函数值

$$y_1, y_2, \cdots, y_{n-1}, y_n$$

为了计算指定插值点 t 处的函数值，选取最靠近插值点 t 的三个结点，用抛物插值公式计算 t 点的函数值。

三、子程序语句

SUBROUTINE ENLG3(X,Y,N,T,Z)

四、形参说明

X,Y——均为双精度实型一维数组，长度为 N，输入参数。存放给定的 N 个数据结点 $(x_i,y_i)(i = 1,2,\cdots,N)$。

N——整型变量，输入参数。给定结点的个数。

T——双精度实型变量，输入参数。指定的插值点值。

Z——双精度实型变量，输出参数。返回指定插值点 T 处的函数近似值。

五、子程序(文件名:ENLG3.FOR)

```
      SUBROUTINE ENLG3(X,Y,N,T,Z)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y,T,Z,S
      Z=0.0
```

```
      IF (N.LE.0) RETURN
      IF (N.EQ.1) THEN
        Z=Y(1)
        RETURN
      END IF
      IF (N.EQ.2) THEN
        Z=(Y(1)*(T-X(2))-Y(2)*(T-X(1)))/(X(1)-X(2))
        RETURN
      END IF
      IF (T.LE.X(2)) THEN
        K=1
        M=3
      ELSE IF (T.GE.X(N-1)) THEN
        K=N-2
        M=N
      ELSE
        K=1
        M=N
10      IF (IABS(K-M).NE.1) THEN
          L=(K+M)/2
          IF (T.LT.X(L)) THEN
            M=L
          ELSE
            K=L
          END IF
          GOTO 10
        END IF
        IF (ABS(T-X(K)).LT.ABS(T-X(M))) THEN
          K=K-1
        ELSE
          M=M+1
        END IF
      END IF
      Z=0.0
      DO 30 I=K,M
        S=1.0
        DO 20 J=K,M
          IF (J.NE.I) THEN
            S=S*(T-X(J))/(X(I)-X(J))
          END IF
20      CONTINUE
        Z=Z+S*Y(I)
```

```
30    CONTINUE
      RETURN
      END
```

六、例

设给定列表函数如下：

x	1.615	1.634	1.702	1.828	1.921
$y=f(x)$	2.41450	2.46459	2.65271	3.03035	3.34066

用抛物插值法计算 $f(1.682)$ 与 $f(1.813)$ 的近似值。

主程序(文件名:ENLG30.FOR)为

```
      DIMENSION X(5),Y(5)
      DOUBLE PRECISION X,Y,T1,T2,Z
      DATA X/1.615,1.634,1.702,1.828,1.921/
      DATA Y/2.41450,2.46459,2.65271,3.03035,3.34066/
      T1=1.682
      T2=1.813
      CALL ENLG3(X,Y,5,T1,Z)
      WRITE(*,10) T1,Z
10    FORMAT(1X,'X=',F6.3,10X,'F(X)=',D15.6)
      CALL ENLG3(X,Y,5,T2,Z)
      WRITE(*,10) T2,Z
      END
```

运行结果为

```
X= 1.682          F(X)=      .259624D+01
X= 1.813          F(X)=      .298281D+01
```

5.4 一元三点等距插值

一、功能

根据给定等距结点上的函数值，用抛物插值计算指定插值点处的函数值。

二、方法说明

同5.3节。

三、子程序语句

SUBROUTINE EELG3(X1,H,N,Y,T,Z)

四、形参说明

X1——双精度实型变量，输入参数。给定N个等距结点中的第一个结点值。

H——双精度实型变量，输入参数。给定N个等距结点的步长。

N——整型变量，输入参数。等距结点的个数。

Y——双精度实型一维数组，长度为 N，输入参数。给定 N 个等距结点上的函数值。

T——双精度实型变量，输入参数。指定的插值点值。

Z——双精度实型变量，输出参数。返回指定插值点 T 处的函数近似值。

五、子程序(文件名:EELG3.FOR)

```
SUBROUTINE EELG3(X1,H,N,Y,T,Z)
DIMENSION Y(N)
DOUBLE PRECISION X1,H,Y,T,Z,S,XI,XJ
Z=0.0
IF (N.LE.0) RETURN
IF (N.EQ.1) THEN
  Z=Y(1)
  RETURN
END IF
IF (N.EQ.2) THEN
  Z=(Y(2)*(T-X1)-Y(1)*(T-X1-H))/H
  RETURN
END IF
IF (T.LE.(X1+H)) THEN
  K=1
  M=3
ELSE IF (T.GE.(X1+(N-2)*H)) THEN
  K=N-2
  M=N
ELSE
  I=(T-X1)/H+1
  IF (ABS(T-X1-I*H).GE.ABS(T-X1-(I-1)*H)) THEN
    K=I-1
    M=I+1
  ELSE
    K=I
    M=I+2
  END IF
END IF
Z=0.0
DO 30 I=K,M
  S=1.0
  XI=X1+(I-1)*H
  DO 20 J=K,M
    IF (J.NE.I) THEN
      XJ=X1+(J-1)*H
      S=S*(T-XJ)/(XI-XJ)
```

```
      END IF
20    CONTINUE
      Z=Z+S*Y(I)
30    CONTINUE
      RETURN
      END
```

六、例

设函数

$$f(x) = e^{-x}$$

在 10 个等距结点上的函数值如下：

x	0.1	0.2	0.3	0.4	0.5
e^{-x}	0.904837	0.818731	0.740818	0.670320	0.606531
x	0.6	0.7	0.8	0.9	1.0
e^{-x}	0.548812	0.496585	0.449329	0.406570	0.367879

计算在插值点 T=0.63 处的函数近似值。

主程序(文件名:EELG30.FOR)为

```
      DIMENSION Y(10)
      DOUBLE PRECISION Y,H,X1,T,Z
      DATA Y/0.904837,0.818731,0.740818,0.670320,0.606531,
     *       0.548812,0.496585,0.449329,0.406570,0.367879/
      H=0.1
      N=10
      X1=0.1
      T=0.63
      CALL EELG3(X1,H,N,Y,T,Z)
      WRITE(*,20) T,Z
20    FORMAT(1X,'T=',F6.3,10X,'Z=',D15.6)
      END
```

运行结果为

```
T=  .630          Z=     .532567D+00
```

5.5 连分式不等距插值

一、功能

根据给定的不等距结点上的函数值，利用连分式插值法计算指定插值点上的函数值。

二、方法说明

设给定 n 个不等距结点

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

及相应的函数值

$$y_1, y_2, \cdots, y_{n-1}, y_n$$

可以构造一个 n 节连分式

$$\varphi(x) = b_1 + \cfrac{x - x_1}{b_2 + \cfrac{x - x_2}{b_3 + \cdots + \cfrac{x - x_{n-1}}{b_n}}}$$

其中 $b_1, b_2, \cdots, b_n$ 由下列递推公式计算：

$$b_1 = y_1$$

$$\begin{cases} u = y_j \\ u = (x_j - x_i)/(u - b_i), \ i = 1,2,\cdots,j-1 \\ b_j = u \end{cases}$$

$$j = 2,3,\cdots,n$$

在实际进行插值计算时，在指定插值点 t 的前后各取四个结点就足够了，此时计算八节连分式值 $\varphi(t)$ 即为插值点 t 处的函数近似值。

三、子程序语句

SUBROUTINE ENPQS(X,Y,N,T,Z)

四、形参说明

X,Y——均为双精度实型一维数组，长度为 N，输入参数。存放给定的 N 个数据结点 $(x_i, y_i)(i = 1,2,\cdots,N)$。

N——整型变量，输入参数。给定结点的个数。

T——双精度实型变量，输入参数。指定插值点值。

Z——双精度实型变量，输出参数。返回指定插值点 T 处的函数近似值。

五、子程序(文件名:ENPQS.FOR)

```
      SUBROUTINE ENPQS(X,Y,N,T,Z)
      DIMENSION X(N),Y(N),B(8)
      DOUBLE PRECISION X,Y,T,Z,B,H
      Z=0.0
      IF (N.LE.0) RETURN
      IF (N.EQ.1) THEN
        Z=Y(1)
        RETURN
      END IF
      IF (N.LE.8) THEN
        K=1
        M=N
      ELSE IF (T.LT.X(5)) THEN
        K=1
        M=8
```

```
      ELSE IF (T.GT.X(N-4)) THEN
        K=N-7
        M=8
      ELSE
        K=1
        J=N
10      IF (IABS(K-J).NE.1) THEN
          L=(K+J)/2
          IF (T.LT.X(L)) THEN
            J=L
          ELSE
            K=L
          END IF
          GOTO 10
        END IF
        K=K-3
        M=8
      END IF
      B(1)=Y(K)
      DO 30 I=2,M
        H=Y(I+K-1)
        DO 20 J=1,I-1
          H=H-B(J)
          IF (ABS(H)+1.0D0.EQ.1.0D0) THEN
            H=SIGN(1.0D+35,H)
            H=H*SIGN(1.0D0,X(I+K-1)-X(J+K-1))
          ELSE
            H=(X(I+K-1)-X(J+K-1))/H
          END IF
20      CONTINUE
        B(I)=H
30    CONTINUE
      Z=B(M)
      DO 40 I=M-1,1,-1
        IF (ABS(Z)+1.0D0.EQ.1.0D0) THEN
          Z=SIGN(1.0D+35,Z)
          Z=Z*SIGN(1.0D0,T-X(I+K-I))
          Z=B(I)+Z
        ELSE
          Z=B(I)+(T-X(I+K-1))/Z
        END IF
40    CONTINUE
```

```
      RETURN
      END
```

六、例

设函数

$$f(x)=\frac{1}{1+25x^2}$$

的 10 个不等距结点上的函数值如下：

x	−1.00	−0.80	−0.65	−0.40	−0.30
$f(x)$	0.0384615	0.0588236	0.0864865	0.200000	0.307692
x	0.00	0.20	0.45	0.80	1.00
$f(x)$	1.00000	0.500000	0.164948	0.0588236	0.0384615

用连分式插值计算 $f(-0.85)$ 及 $f(0.25)$ 的近似值。

主程序(文件名:ENPQS0.FOR)为

```
      DIMENSION X(10),Y(10)
      DOUBLE PRECISION X,Y,T,Z
      DATA X/-1.0,-0.8,-0.65,-0.4,-0.3,0.0,0.2,0.45,0.8,1.0/
      DATA Y/0.0384615,0.0588236,0.0864865,0.2,0.307692,
     *       1.0,0.5,0.164948,0.0588236,0.0384615/
      T=-0.85
      CALL ENPQS(X,Y,10,T,Z)
      WRITE(*,20) T,Z
20    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      T=0.25
      CALL ENPQS(X,Y,10,T,Z)
      WRITE(*,20) T,Z
      END
```

运行结果为

```
X= -.850          F(X)=     .524591D-01
X=  .250          F(X)=     .390244D+00
```

5.6 连分式等距插值

一、功能

根据给定的等距结点上的函数值，用连分式插值法计算指定插值点上的函数值。

二、方法说明

同 5.5 节。

三、子程序语句

SUBROUTINE　EEPQS(X1,H,N,Y,T,Z)

四、形参说明

X1——双精度实型变量，输入参数。给定 N 个等距结点中的第一个结点值。

H——双精度实型变量，输入参数。给定 N 个等距结点的步长。

N——整型变量，输入参数。给定等距结点的个数。

Y——双精度实型一维数组，长度为 N，输入参数。存放 N 个等距结点上的函数值。

T——双精度实型变量，输入参数。指定插值点值。

Z——双精度实型变量，输出参数。返回指定插值点 T 处的函数近似值。

五、子程序(文件名：EEPQS.FOR)

```
SUBROUTINE EEPQS(X1,H,N,Y,T,Z)
DIMENSION Y(N),B(8)
DOUBLE PRECISION X1,H,Y,T,Z,B,HH,X2,X3
Z=0.0
IF (N.LE.0) RETURN
IF (N.EQ.1) THEN
  Z=Y(1)
  RETURN
END IF
IF (N.LE.8) THEN
K=1
M=N
ELSE IF (T.LT.(X1+H*4.0)) THEN
K=1
M=8
ELSE IF (T.GT.(X1+(N-5)*H)) THEN
K=N-7
M=8
ELSE
  K=(T-X1)/H-2
  M=8
END IF
B(1)=Y(K)
DO 30 I=2,M
  HH=Y(I+K-1)
  X2=X1+(I+K-2)*H
  DO 20 J=1,I-1
    HH=HH-B(J)
    X3=X1+(J+K-2)*H
    IF (ABS(HH)+1.0D0.EQ.1.0D0) THEN
      HH=SIGN(1.0D+35,HH)
```

```
      HH=HH * SIGN(1.0D0,X2-X3)
     ELSE
      HH=(X2-X3)/HH
     END IF
20   CONTINUE
     B(I)=HH
30  CONTINUE
    Z=B(M)
    DO 40 I=M-1,1,-1
     HH=T-(X1+(I+K-2) * H)
     IF (ABS(Z)+1.0D0.EQ.1.0D0) THEN
      Z=SIGN(1.0D+35,Z)
      Z=Z * SIGN(1.0D0,HH)
      Z=B(I)+Z
     ELSE
      Z=B(I)+HH/Z
     END IF
40  CONTINUE
    RETURN
    END
```

六、例

设函数

$$f(x)=\frac{1}{1+25x^2}$$

的 11 个等距结点上的函数值如下：

x	-1.00	-0.80	-0.60	-0.40	-0.20	0.00
$f(x)$	0.0384615	0.0588236	0.100000	0.200000	0.500000	1.00000

x	0.20	0.40	0.60	0.80	1.00
$f(x)$	0.500000	0.200000	0.100000	0.0588236	0.0384615

用连分式插值法计算 $f(-0.75)$ 与 $f(0.05)$ 的近似值。

主程序(文件名：EEPQS0.FOR)为

```
    DIMENSION Y(11)
    DOUBLE PRECISION Y,H,X1,T,Z
    DATA Y/0.0384615,0.0588236,0.1,0.2,0.5,1.0,
   *        0.5,0.2,0.1,0.0588236,0.0384615/
    H=0.2
    X1=-1.0
    T=-0.75
```

```
      CALL EEPQS(X1,H,11,Y,T,Z)
      WRITE(*,20) T,Z
20    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      T=0.05
      CALL EEPQS(X1,H,11,Y,T,Z)
      WRITE(*,20) T,Z
      END
```

运行结果为

```
X= -.750          F(X)=    .663901D-01
X=  .050          F(X)=    .941177D+00
```

5.7 埃尔米特不等距插值

一、功能

根据给定的不等距结点上的函数值与一阶导数值，用埃尔米特(Hermite)插值法计算指定插值点上的函数值。

二、方法说明

设已知 $f(x)$ 在 n 个不等距结点

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

上的函数值与一阶导数值

$$y_1, y_2, \cdots, y_{n-1}, y_n$$
$$y'_1, y'_2, \cdots, y'_{n-1}, y'_n$$

则 $f(x)$ 可用埃尔米特插值多项式

$$P_{2n-1}(x) = \sum_{i=1}^{n}[y_i + (x - x_i)(y'_i - 2y_i l'_i(x_i))]l_i^2(x)$$

近似代替。其中

$$l_i(x) = \prod_{j=1,j\neq i}^{n}[(x - x_j)/(x_i - x_j)]$$

$$l'_i(x_i) = \sum_{j=1,j\neq i}^{n}\frac{1}{x_i - x_j}$$

在实际计算时，一般只需取插值点 t 的前后四个结点就够了。在本子程序中，所有给定的结点全部用来进行插值计算，为了减少计算工作量，用户在使用本子程序时，可以适当地将远离插值点的结点抛弃。

三、子程序语句

SUBROUTINE　ENHMT(X,Y,DY,N,T,Z)

四、形参说明

X,Y,DY——均为双精度实型一维数组，长度为 N，输入参数。存放给定的 N 个数据点值、函数值与一阶导数值 $(x_i, y_i, y'_i)(i = 1,2,\cdots,N)$。

N——整型变量，输入参数。给定结点的个数。

T——双精度实型变量,输入参数。指定插值点值。

Z——双精度实型变量,输出参数。返回指定插值点 T 处的函数近似值。

五、子程序(文件名:ENHMT.FOR)

```
      SUBROUTINE ENHMT(X,Y,DY,N,T,Z)
      DIMENSION X(N),Y(N),DY(N)
      DOUBLE PRECISION X,Y,DY,T,Z,S,P,Q
      Z=0.0
      DO 30 I=1,N
        S=1.0
        DO 10 J=1,N
          IF (J.NE.I) S=S*(T-X(J))/(X(I)-X(J))
10      CONTINUE
        S=S*S
        P=0.0
        DO 20 J=1,N
          IF (J.NE.I) P=P+1.0/(X(I)-X(J))
20      CONTINUE
        Q=Y(I)+(T-X(I))*(DY(I)-2*Y(I)*P)
        Z=Z+Q*S
30    CONTINUE
      RETURN
      END
```

六、例

设函数

$$f(x) = e^{-x}$$

在 10 个不等距结点上的函数值与一阶导数值如下:

x	0.10	0.15	0.30	0.45	0.55
$f(x)$	0.904837	0.860708	0.740818	0.637628	0.576950
$f'(x)$	−0.904837	−0.860708	−0.740818	−0.637628	−0.576950
x	0.60	0.70	0.85	0.90	1.00
$f(x)$	0.548812	0.496585	0.427415	0.406570	0.367879
$f'(x)$	−0.548812	−0.496585	−0.427415	−0.406570	−0.367879

用埃尔米特插值公式计算插值点 T=0.356 处的函数近似值。

主程序(文件名:ENHMT0.FOR)为

```
      DIMENSION X(10),Y(10),DY(10)
      DOUBLE PRECISION X,Y,DY,T,Z
```

```
      DATA X/0.1,0.15,0.30,0.45,0.55,0.6,0.7,0.85,0.9,1.0/
      DATA Y/0.904837,0.860708,0.740818,0.637628,0.576950,
     *      0.548812,0.496585,0.427415,0.406570,0.367879/
      DO 10 I=1,10
        DY(I)=-Y(I)
10    CONTINUE
      T=0.356
      CALL ENHMT(X,Y,DY,10,T,Z)
      WRITE(*,20) T,Z
20    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      END
```

运行结果为

```
X=   .356         F(X)=     .700478D+00
```

5.8 埃尔米特等距插值

一、功能

根据给定的等距结点上的函数值与一阶导数值，用埃尔米特(Hermite)等距插值公式计算指定插值点处的函数值。

二、方法说明

同5.7节。

三、子程序语句

SUBROUTINE　EEHMT(X1,H,N,Y,DY,T,Z)

四、形参说明

X1——双精度实型变量，输入参数。给定N个等距结点中的第一个结点值。

H——双精度实型变量，输入参数。等距结点的步长。

N——整型变量，输入参数。给定等距结点的个数。

Y,DY——均为双精度实型一维数组，长度为N，输入参数。给定N个等距结点上的函数值及一阶导数值。

T——双精度实型变量，输入参数。指定的插值点值。

Z——双精度实型变量，输出参数。返回指定插值点T处的函数近似值。

五、子程序(文件名：EEHMT.FOR)

```
      SUBROUTINE EEHMT(X1,H,N,Y,DY,T,Z)
      DIMENSION Y(N),DY(N)
      DOUBLE PRECISION X1,H,Y,DY,T,Z,S,P,Q
      Z=0.0
      DO 30 I=1,N
        S=1.0
        Q=X1+(I-1)*H
        DO 10 J=1,N
```

```
      P=X1+(J-1)*H
      IF (J.NE.I) S=S*(T-P)/(Q-P)
10    CONTINUE
      S=S*S
      P=0.0
      DO 20 J=1,N
        IF (J.NE.I) P=P+1.0/(Q-(X1+(J-1)*H))
20    CONTINUE
      Q=Y(I)+(T-Q)*(DY(I)-2*Y(I)*P)
      Z=Z+Q*S
30    CONTINUE
      RETURN
      END
```

六、例

设函数

$$f(x)=e^{-x}$$

在 10 个等距结点上的函数值与一阶导数值如下:

x	0.10	0.20	0.30	0.40	0.50
$f(x)$	0.904837	0.818731	0.740818	0.670320	0.606531
$f'(x)$	−0.904837	−0.818731	−0.740818	−0.670320	−0.606531
x	0.60	0.70	0.80	0.90	1.00
$f(x)$	0.548812	0.496585	0.449329	0.406570	0.367879
$f'(x)$	−0.548812	−0.496585	−0.449329	−0.406570	−0.367879

用埃尔米特插值法计算插值点 T=0.752 处的函数近似值。

主程序(文件名:EEHMT0.FOR)为

```
      DIMENSION Y(10),DY(10)
      DOUBLE PRECISION Y,DY,T,Z,X1,H
      DATA Y/0.904837,0.818731,0.740818,0.670320,0.606531,
     *       0.548812,0.496585,0.449329,0.406570,0.367879/
      X1=0.1
      H=0.1
      DO 10 I=1,10
        DY(I)=-Y(I)
10    CONTINUE
      T=0.752
      CALL EEHMT(X1,H,10,Y,DY,T,Z)
      WRITE(*,20) T,Z
```

```
20    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      END
```

运行结果为

```
X=   .752          F(X)=    .471423D+00
```

5.9 埃特金不等距逐步插值

一、功能

根据给定不等距结点上的函数值，用埃特金(Aitken)逐步插值法计算指定插值点处的函数值。

二、方法说明

设给定 n 个结点

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

上的函数值

$$y_1, y_2, \cdots, y_{n-1}, y_n$$

用埃特金逐步插值法计算指定插值点 t 处函数值的方法如下。

从 n 个结点中选取最靠近插值点 t，且尽量使 t 位于其中心的若干个结点进行插值计算，其埃特金逐步线性插值的公式为

$$P_{i,k+1}(t) = P_{i-1,k}(t) + \frac{t - x_k}{x_k - x_{k+1}}[P_{i-1,k}(t) - P_{i-1,k+1}(t)]$$

其中 $P_{i,j}(x)$ 表示通过结点 $x_1, x_2, \cdots, x_i$ 及 x_j 的 i 次插值多项式，且 $P_{1,j} = y_j (j = 1,2,\cdots)$。

三、子程序语句

SUBROUTINE　ENATK(X,Y,N,T,EPS,Z)

四、形参说明

X,Y——均为双精度实型一维数组，长度为 N，输入参数。存放给定的 N 个不等距结点值与函数值。

N——整型变量，输入参数。给定不等距结点的个数。

T——双精度实型变量，输入参数。指定的插值点值。

EPS——实型变量，输入参数。控制精度要求。

Z——双精度实型变量，输出参数。返回指定插值点 T 处的函数近似值。

五、子程序(文件名：ENATK.FOR)

```
      SUBROUTINE ENATK(X,Y,N,T,EPS,Z)
      DIMENSION X(N),Y(N),XM(10),YM(10)
      DOUBLE PRECISION X,Y,XM,YM,T,Z
      M=10
      IF (M.GT.N) M=N
      Z=0.0
      IF (M.LE.0) RETURN
      IF (N.EQ.1) THEN
```

```
          Z=Y(1)
          RETURN
        END IF
        IF (M.EQ.1) M=2
        IF (T.LE.X(1)) THEN
          K=1
        ELSE IF (T.GE.X(N)) THEN
          K=N
        ELSE
          K=1
          J=N
10        IF (IABS(K-J).NE.1) THEN
            L=(K+J)/2
            IF (T.LT.X(L)) THEN
              J=L
            ELSE
              K=L
            END IF
            GOTO 10
          END IF
          IF (ABS(T-X(L)).GT.ABS(T-X(J))) K=J
        END IF
        J=1
        L=0
        DO 20 I=1,M
          K=K+J*L
          IF ((K.LT.1).OR.(K.GT.N)) THEN
            L=L+1
            J=-J
            K=K+J*L
          END IF
          XM(I)=X(K)
          YM(I)=Y(K)
          L=L+1
          J=-J
20      CONTINUE
        I=2
        P=1.0+EPS
40      IF ((I.LE.M).AND.(P.GE.EPS)) THEN
          Z=YM(I)
          DO 30 J=2,I
30        Z=YM(J-1)+(T-XM(J-1))*(YM(J-1)-Z)/(XM(J-1)-XM(I))
```

```
      YM(I)=Z
      P=ABS(YM(I)-YM(I-1))
      I=I+1
      GOTO 40
    END IF
    RETURN
    END
```

六、例

设函数 $f(x)$ 在五个结点上的函数值如下：

x	1.615	1.634	1.702	1.828	1.921
$f(x)$	2.41450	2.46459	2.65271	3.03035	3.34066

用埃特金逐步线性插值法计算 $f(1.682)$ 与 $f(1.813)$ 的近似值。

主程序(文件名:ENATK0.FOR)为

```
      DIMENSION X(5),Y(5)
      DOUBLE PRECISION X,Y,T,Z
      DATA X/1.615,1.634,1.702,1.828,1.921/
      DATA Y/2.41450,2.46459,2.65271,3.03035,3.34066/
      EPS=0.000001
      T=1.682
      CALL ENATK(X,Y,5,T,EPS,Z)
      WRITE(*,10) T,Z
10    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      T=1.813
      CALL ENATK(X,Y,5,T,EPS,Z)
      WRITE(*,10) T,Z
      END
```

运行结果为

```
X=  1.682          F(X)=     .259612D+01
X=  1.813          F(X)=     .298332D+01
```

5.10 埃特金等距逐步插值

一、功能

根据给定等距结点上的函数值，用埃特金(Aitken)逐步插值法计算指定插值点处的函数值。

二、方法说明

同 5.9 节。

三、子程序语句

SUBROUTINE　EEATK(X1,H,N,Y,T,EPS,Z)

四、形参说明

X1——双精度实型变量，输入参数。给定N个等距结点中的第一个结点值。

H——双精度实型变量，输入参数。等距结点的步长。

N——整型变量，输入参数。给定等距结点的个数。

Y——双精度实型一维数组，长度为N，输入参数。存放N个等距结点上的函数值。

T——双精度实型变量，输入参数。指定的插值点值。

EPS——实型变量，输入参数。控制精度要求。

Z——双精度实型变量，输出参数。返回指定插值点T处的函数近似值。

五、子程序(文件名：EEATK.FOR)

```
      SUBROUTINE EEATK(X1,H,N,Y,T,EPS,Z)
      DIMENSION Y(N),XM(10),YM(10)
      DOUBLE PRECISION X1,H,Y,XM,YM,T,Z
      M=10
      IF (M.GT.N) M=N
      Z=0.0
      IF (M.LE.0) RETURN
      IF (N.EQ.1) THEN
        Z=Y(1)
        RETURN
      END IF
      IF (M.EQ.1) M=2
      IF (T.LE.X1) THEN
        K=1
      ELSE IF (T.GE.X1+(N-1)*H) THEN
        K=N
      ELSE
        K=1
        J=N
10      IF (IABS(K-J).NE.1) THEN
          L=(K+J)/2
          IF (T.LT.X1+(L-1)*H) THEN
            J=L
          ELSE
            K=L
          END IF
          GOTO 10
        END IF
        IF (ABS(T-X1-(L-1)*H).GT.ABS(T-X1-(J-1)*H)) K=J
      END IF
```

```
      J=1
      L=0
      DO 20 I=1,M
        K=K+J*L
        IF ((K.LT.1).OR.(K.GT.N)) THEN
          L=L+1
          J=-J
          K=K+J*L
        END IF
        XM(I)=X1+(K-1)*H
        YM(I)=Y(K)
        L=L+1
        J=-J
20    CONTINUE
      I=2
      P=1.0+EPS
100   IF ((I.LE.M).AND.(P.GE.EPS)) THEN
        Z=YM(I)
        DO 30 J=2,I
30      Z=YM(J-1)+(T-XM(J-1))*(YM(J-1)-Z)/(XM(J-1)-XM(I))
        YM(I)=Z
        P=ABS(YM(I)-YM(I-1))
        I=I+1
        GOTO 100
      END IF
      RETURN
      END
```

六、例

设函数 $f(x)$ 在 10 个等距结点上的函数值如下：

x	0.1	0.2	0.3	0.4	0.5
$f(x)$	0.904837	0.818731	0.740818	0.670320	0.606531
x	0.6	0.7	0.8	0.9	1.0
$f(x)$	0.548812	0.496585	0.449329	0.406570	0.367879

用埃特金线性逐步插值法计算 $f(0.15)$ 与 $f(0.55)$ 的近似值。

主程序(文件名:EEATK0.FOR)为

```
      DIMENSION Y(10)
      DOUBLE PRECISION X,H,Y,T,Z
      DATA Y/0.904837,0.818731,0.740818,0.670320,0.606531,
```

```
     *          0.548812,0.496585,0.449329,0.406570,0.367879/
      X=0.1
      H=0.1
      EPS=0.000001
      T=0.15
      CALL EEATK(X,H,10,Y,T,EPS,Z)
      WRITE(*,10) T,Z
10    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      T=0.55
      CALL EEATK(X,H,10,Y,T,EPS,Z)
      WRITE(*,10) T,Z
      END
```

运行结果为

```
X=   .150          F(X)=    .860708D+00
X=   .550          F(X)=    .576950D+00
```

5.11 光滑不等距插值

一、功能

根据给定的不等距结点上的函数值，用阿克玛(Akima)方法构造指定子区间上的三次插值多项式与计算指定插值点处的函数值。

二、方法说明

设给定 n 个不等距结点

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

上的函数值

$$y_1, y_2, \cdots, y_{n-1}, y_n$$

若在子区间 $[x_k, x_{k+1}](k = 1, 2, \cdots, n-1)$ 上的两个端点处有以下四个条件：

$$\begin{cases} y_k = f(x_k) \\ y_{k+1} = f(x_{k+1}) \\ y'_k = g_k \\ y'_{k+1} = g_{k+1} \end{cases}$$

则可以唯一确定一个三次多项式

$$S(x) = a + b(x - x_k) + c(x - x_k)^2 + d(x - x_k)^3$$

然后就用上述多项式计算该子区间中的插值点 t 处的函数值 $S(t)$ 。

根据阿克玛几何条件，g_k 与 g_{k+1} 由下式计算：

$$g_k = \frac{|u_{k+1} - u_k| u_{k-1} + |u_{k-1} - u_{k-2}| u_k}{|u_{k+1} - u_k| + |u_{k-1} - u_{k-2}|}$$

$$g_{k+1} = \frac{|u_{k+2} - u_{k+1}|u_k + |u_k - u_{k-1}|u_{k+1}}{|u_{k+2} - u_{k+1}| + |u_k - u_{k-1}|}$$

当 $u_{k+1}=u_k$ 与 $u_{k-1}=u_{k-2}$ 时

$$g_k = (u_{k-1} + u_k)/2$$

当 $u_{k+2} = u_{k+1}$ 与 $u_k = u_{k-1}$ 时

$$g_{k+1} = (u_k + u_{k+1})/2$$

其中 $u_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$，且在端点处有

$$u_0 = 2u_1 - u_2, \qquad u_{-1} = 2u_0 - u_1$$

$$u_n = 2u_{n-1} - u_{n-2}, \qquad u_{n+1} = 2u_n - u_{n-1}$$

最后可以求得在子区间 $[x_k, x_{k+1}]$ 上的三次多项式的系数为

$$a = y_k$$

$$b = g_k$$

$$c = (3u_k - 2g_k - g_{k+1})/(x_{k+1} - x_k)$$

$$d = (g_{k+1} + g_k - 2u_k)/(x_{k+1} - x_k)$$

插值点 t 处的函数近似值为

$$S(t) = a + b(t - x_k) + c(t - x_k)^2 + d(t - x_k)^3$$

三、子程序

SUBROUTINE　ENSPL(X,Y,N,K,T,Z,A,B,C,D)

四、形参说明

X,Y——均为双精度实型一维数组，长度为 N，输入参数。存放给定的 N 个不等距结点值与函数值。

N——整型变量，输入参数。不等距结点的个数。

K——整型变量，输入兼输出参数。在调用时，若 K≤0，则表示需要计算指定插值点 T 处的函数近似值，并计算插值点 T 所在子区间的三次多项式的系数 A,B,C,D；否则只需要计算第 K 个子区间上的三次多项式的系数 A,B,C,D，此时如果 K≥N，则按 K=N−1 处理。在返回时，K 中返回子区间号。

T——双精度实型变量，输入参数。当输入 K≤0 时，应存放指定的插值点值；当输入 K>0 时，此参数可任意。

Z——双精度实型变量，输出参数。当输入 K≤0 时，返回指定插值点 T 处的函数近似值；当输入 K>0 时，返回一个随机值。

A,B,C,D——均为双精度实型变量，输出参数。返回第 K 个子区间 $[x_k, x_{k+1}]$ 上的三次多项式

$$S(x) = A + B(x - x_k) + C(x - x_k)^2 + D(x - x_k)^3$$

的系数。

五、子程序(文件名:ENSPL.FOR)

```
      SUBROUTINE ENSPL(X,Y,N,K,T,Z,A,B,C,D)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y,T,Z,G1,G2,U1,U2,U3,U4,U5,A,B,C,D,S
```

```
      Z=0.0
      IF (N.LE.0) RETURN
      IF (N.EQ.1) THEN
        Z=Y(1)
        K=0
        A=Y(1)
        B=0.0
        C=0.0
        D=0.0
        RETURN
      END IF
      IF (N.EQ.2) THEN
        Z=(Y(1)*(T-X(2))-Y(2)*(T-X(1)))/(X(1)-X(2))
        K=1
        A=Y(1)
        B=(Y(2)-Y(1))/(X(2)-X(1))
        C=0.0
        D=0.0
        RETURN
      END IF
      S=1.0
      IF (K.LE.0) THEN
        S=-1.0
        IF (T.LE.X(2)) THEN
          K=1
        ELSE IF (T.GE.X(N)) THEN
          K=N-1
        ELSE
          K=1
          M=N
10        IF (IABS(K-M).NE.1) THEN
            L=(K+M)/2
            IF (T.LT.X(L)) THEN
              M=L
            ELSE
              K=L
            END IF
            GOTO 10
          END IF
        END IF
      END IF
      IF (K.GE.N) K=N-1
```

```
U3=(Y(K+1)-Y(K))/(X(K+1)-X(K))
IF (N.EQ.3) THEN
  IF (K.EQ.1) THEN
    U4=(Y(3)-Y(2))/(X(3)-X(2))
    U5=2.0*U4-U3
    U2=2.0*U3-U4
    U1=2.0*U2-U3
  ELSE IF (K.EQ.2) THEN
    U2=(Y(2)-Y(1))/(X(2)-X(1))
    U1=2.0*U2-U3
    U4=2.0*U3-U2
    U5=2.0*U4-U3
  END IF
ELSE
  IF (K.LE.2) THEN
    U4=(Y(K+2)-Y(K+1))/(X(K+2)-X(K+1))
    IF (K.EQ.2) THEN
      U2=(Y(2)-Y(1))/(X(2)-X(1))
      U1=2*U2-U3
      IF (N.EQ.4) THEN
        U5=2.0*U4-U3
      ELSE
        U5=(Y(5)-Y(4))/(X(5)-X(4))
      END IF
    ELSE
      U2=2*U3-U4
      U1=2*U2-U3
      U5=(Y(4)-Y(3))/(X(4)-X(3))
    END IF
  ELSE IF (K.GE.(N-2)) THEN
    U2=(Y(K)-Y(K-1))/(X(K)-X(K-1))
    IF (K.EQ.(N-2)) THEN
      U4=(Y(N)-Y(N-1))/(X(N)-X(N-1))
      U5=2*U4-U3
      IF (N.EQ.4) THEN
        U1=2.0*U2-U3
      ELSE
        U1=(Y(K-1)-Y(K-2))/(X(K-1)-X(K-2))
      END IF
    ELSE
      U4=2*U3-U2
      U5=2*U4-U3
```

```
      U1=(Y(K-1)-Y(K-2))/(X(K-1)-X(K-2))
    END IF
  ELSE
    U2=(Y(K)-Y(K-1))/(X(K)-X(K-1))
    U1=(Y(K-1)-Y(K-2))/(X(K-1)-X(K-2))
    U4=(Y(K+2)-Y(K+1))/(X(K+2)-X(K+1))
    U5=(Y(K+3)-Y(K+2))/(X(K+3)-X(K+2))
  END IF
END IF
A=ABS(U4-U3)
B=ABS(U1-U2)
IF ((A+1.0.EQ.1.0).AND.(B+1.0.EQ.1.0)) THEN
  G1=(U2+U3)/2.0
ELSE
  G1=(A*U2+B*U3)/(A+B)
END IF
A=ABS(U4-U5)
B=ABS(U3-U2)
IF ((A+1.0.EQ.1.0).AND.(B+1.0.EQ.1.0)) THEN
  G2=(U3+U4)/2.0
ELSE
  G2=(A*U3+B*U4)/(A+B)
END IF
A=Y(K)
B=G1
D=X(K+1)-X(K)
C=(3*U3-2*G1-G2)/D
D=(G2+G1-2*U3)/(D*D)
IF (S.LT.0.0) THEN
  S=T-X(K)
  Z=A+B*S+C*S*S+D*S*S*S
END IF
RETURN
END
```

六、例

设已知函数

$$f(x) = \frac{1}{1+25x^2}$$

在 11 个给定不等距结点上的函数值如下:

x	−1.00	−0.95	−0.75	−0.55	−0.30	0.00
$f(x)$	0.0384615	0.0424403	0.0663900	0.116788	0.307692	1.00000
x	0.20	0.45	0.60	0.80	1.00	
$f(x)$	0.500000	0.164948	0.100000	0.0588236	0.0384615	

用光滑插值法计算 $f(-0.85)$ 与 $f(0.15)$ 的近似值，并求插值点 T=−0.85 与 T=0.15 所在区间的三次多项式

$$S(x)=A+B(x-x_i)+C(x-x_i)^2+D(x-x_i)^3$$

的系数 A,B,C,D。

主程序(文件名:ENSPL0.FOR)为

```
      DIMENSION X(11),Y(11)
      DOUBLE PRECISION X,Y,T,Z,A,B,C,D
      DATA X/-1.0,-0.95,-0.75,-0.55,-0.3,0.0,
     *       0.2,0.45,0.6,0.8,1.0/
      DATA Y/0.0384615,0.0424403,0.06639,0.116788,0.307692,
     *       1.0,0.5,0.164948,0.1,0.0588236,0.0384615/
      T=-0.85
      K=-1
      CALL ENSPL(X,Y,11,K,T,Z,A,B,C,D)
      WRITE(*,*)
      WRITE(*,20) T,Z
      WRITE(*,30) K
      WRITE(*,40) A,B,C,D
      WRITE(*,*)
20    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      T=0.15
      K=-1
      CALL ENSPL(X,Y,11,K,T,Z,A,B,C,D)
      WRITE(*,20) T,Z
      WRITE(*,30) K
      WRITE(*,40) A,B,C,D
      WRITE(*,*)
30    FORMAT(1X,'K=',I3)
40    FORMAT(1X,'A=',D13.6,3X,'B=',D13.6,3X,
     *       'C=',D13.6,3X,'D=',D13.6)
      END
```

运行结果为

```
X=   -.850            F(X)=     .534043D-01
K=   2
```

```
A= .424403D-01  B= .869362D-01  C= .259985D+00  D= -.529619D+00

X=  .150          F(X)=    .616892D+00
K=  6
A= .100000D+01  B= -.437798D+00  C= -.255004D+02  D=  .759470D+02
```

5.12 光滑等距插值

一、功能

根据给定的等距结点上的函数值，用阿克玛(Akima)方法构造指定子区间上的三次插值多项式与计算指定插值点上的函数值。

二、方法说明

同5.11节。

三、子程序语句

SUBROUTINE EESPL(X1,H,N,Y,K,T,Z,A,B,C,D)

四、形参说明

X1——双精度实型变量，输入参数。给定N个等距结点中的第一个结点值。

H——双精度实型变量，输入参数。等距结点的步长。

N——整型变量，输入参数。等距结点的个数。

Y——双精度实型一维数组，长度为N，输入参数。存放给定N个等距结点上的函数值。

K——整型变量，输入兼输出参数。在调用时，如果K≤0，则表示需要计算指定插值点T处的函数近似值，并求插值点T所在子区间上的三次插值多项式的系数A,B,C,D；否则只计算第K个子区间上的三次插值多项式的系数A,B,C,D，此时如果K≥N，则按K=N−1处理。在返回时，返回子区间号。

T——双精度实型变量，输入参数。当输入K≤0时，应存放指定插值点的值；当输入K>0时，此参数可任意。

Z——双精度实型变量，输出参数，当输入K≤0时，返回指定插值点T处的函数近似值；当输入K>0时，返回随机值。

A,B,C,D——均为双精度实型变量，输出参数。返回第K个子区间$[x_k,x_{k+1}]$上的三次插值多项式

$$S(x)=A+B(x-x_k)+C(x-x_k)^2+D(x-x_k)^3$$

的系数。

五、子程序(文件名:EESPL.FOR)

```
      SUBROUTINE EESPL(X1,H,N,Y,K,T,Z,A,B,C,D)
      DIMENSION Y(N)
      DOUBLE PRECISION X1,H,Y,T,Z,A,B,C,D,
     *                 U1,U2,U3,U4,U5,S,G1,G2
      Z=0.0
```

```
IF (N.LE.0) RETURN
IF (N.EQ.1) THEN
  Z=Y(1)
  K=0
  A=Y(1)
  B=0.0
  C=0.0
  D=0.0
  RETURN
END IF
IF (N.EQ.2) THEN
  Z=(Y(2)*(T-X1)-Y(1)*(T-X1-H))/H
  K=1
  A=Y(1)
  B=(Y(2)-Y(1))/H
  C=0.0
  D=0.0
  RETURN
END IF
S=1.0
IF (K.LE.0) THEN
  S=-1.0
  A=X1+H
  B=X1+(N-1)*H
  IF (T.LE.A) THEN
    K=1
  ELSE IF (T.GE.B) THEN
    K=N-1
  ELSE
    K=1+(T-X1)/H
  END IF
END IF
IF (K.GE.N) K=N-1
U3=(Y(K+1)-Y(K))/H
IF (N.EQ.3) THEN
  IF (K.EQ.1) THEN
    U4=(Y(3)-Y(2))/H
    U5=2.0*U4-U3
    U2=2.0*U3-U4
    U1=2.0*U2-U3
  ELSE IF (K.EQ.2) THEN
    U2=(Y(2)-Y(1))/H
```

```
      U1=2.0*U2-U3
      U4=2.0*U3-U2
      U5=2.0*U4-U3
    END IF
  ELSE
    IF (K.LE.2) THEN
      U4=(Y(K+2)-Y(K+1))/H
      IF (K.EQ.2) THEN
        U2=(Y(2)-Y(1))/H
        U1=2*U2-U3
        IF (N.EQ.4) THEN
          U5=2.0*U4-U3
        ELSE
          U5=(Y(K+3)-Y(K+2))/H
        END IF
      ELSE
        U2=2*U3-U4
        U1=2*U2-U3
        U5=(Y(K+3)-Y(K+2))/H
      END IF
    ELSE IF (K.GE.(N-2)) THEN
      U1=(Y(K-1)-Y(K-2))/H
      U2=(Y(K)-Y(K-1))/H
      IF (K.EQ.(N-2)) THEN
        U4=(Y(N)-Y(N-1))/H
        U5=2*U4-U3
        IF (N.EQ.4) THEN
          U1=2.0*U2-U3
        ELSE
          U1=(Y(K-1)-Y(K-2))/H
        END IF
      ELSE
        U4=2*U3-U2
        U5=2*U4-U3
        U1=(Y(K-1)-Y(K-2))/H
      END IF
    ELSE
      U2=(Y(K)-Y(K-1))/H
      U1=(Y(K-1)-Y(K-2))/H
      U4=(Y(K+2)-Y(K+1))/H
      U5=(Y(K+3)-Y(K+2))/H
    END IF
```

```
END IF
A=ABS(U4-U3)
B=ABS(U1-U2)
IF ((A+1.0.EQ.1.0).AND.(B+1.0.EQ.1.0)) THEN
  G1=(U2+U3)/2.0
ELSE
  G1=(A*U2+B*U3)/(A+B)
END IF
A=ABS(U4-U5)
B=ABS(U3-U2)
IF ((A+1.0.EQ.1.0).AND.(B+1.0.EQ.1.0)) THEN
  G2=(U3+U4)/2.0
ELSE
  G2=(A*U3+B*U4)/(A+B)
END IF
A=Y(K)
B=G1
C=(3*U3-2*G1-G2)/H
D=(G2+G1-2*U3)/(H*H)
IF (S.LT.0.0) THEN
  S=T-(X1+(K-1)*H)
  Z=A+B*S+C*S*S+D*S*S*S
END IF
RETURN
END
```

六、例

设函数

$$f(x)=\frac{1}{1+25x^2}$$

在 11 个等距结点上的函数值如下：

x	−1.00	−0.80	−0.60	−0.40	−0.20	0.00
$f(x)$	0.0384615	0.0588236	0.100000	0.200000	0.500000	1.00000
x	0.20	0.40	0.60	0.80	1.00	
$f(x)$	0.500000	0.200000	0.100000	0.0588236	0.0384615	

用光滑插值法计算 $f(-0.65)$ 与 $f(0.25)$ 的近似值，并求插值点 T=−0.65 与 T=

0.25 所在子区间上的三次插值多项式

$$S(x) = A + B(x - x_i) + C(x - x_i)^2 + D(x - x_i)^3$$

的系数 A,B,C,D。

主程序(文件名:EESPL0.FOR)为

```
      DIMENSION Y(11)
      DOUBLE PRECISION Y,X1,H,T,Z,A,B,C,D
      DATA Y/0.0384615,0.0588236,0.1,0.2,0.5,1.0,
     *       0.5,0.2,0.1,0.0588236,0.0384615/
      X1=-1.0
      H=0.2
      T=-0.65
      K=-1
      CALL EESPL(X1,H,11,Y,K,T,Z,A,B,C,D)
      WRITE(*,*)
      WRITE(*,20) T,Z
      WRITE(*,30) K
      WRITE(*,40) A,B,C,D
      WRITE(*,*)
20    FORMAT(1X,'X=',F7.3,10X,'F(X)=',D15.6)
      T=0.25
      K=-1
      CALL EESPL(X1,H,11,Y,K,T,Z,A,B,C,D)
      WRITE(*,20) T,Z
      WRITE(*,30) K
      WRITE(*,40) A,B,C,D
      WRITE(*,*)
30    FORMAT(1X,'K=',I3)
40    FORMAT(1X,'A=',D13.6,3X,'B=',D13.6,3X,
     *       'C=',D13.6,3X,'D=',D13.6)
      END
```

运行结果为

```
X=   -.650          F(X)=    .882055D-01
K=  2
A=  .588236D-01   B=  .129011D+00   C=  .630092D+00   D= -.122868D+01

X=   .250           F(X)=    .413058D+00
K= 7
A=  .500000D+00   B= -.166667D+01   C= -.219697D+01   D=  .151515D+02
```

5.13 第一种边界条件的三次样条函数插值、微商与积分

一、功能

根据给定结点上的函数值及第一种边界条件，用三次样条函数计算各结点上的数值导数与积分，并对一元函数进行成组插值与成组微商。

二、方法说明

设已知函数 $y=f(x)$ 在给定结点

$$x_1<x_2<\cdots<x_{n-1}<x_n$$

上的函数值

$$y_1,y_2,\cdots,y_{n-1},y_n$$

以及两端点上的一阶导数值 $y'(x_1)$ 与 $y'(x_n)$。

计算其余 $n-2$ 个结点上的导数值 $y'(x_j)(j=2,3,\cdots,n-1)$ 的公式如下：

$$a_1=0\ ,\ b_1=y'(x_1)$$

$$h_j=x_{j+1}-x_j,\ j=1,2,\cdots,n-1$$

$$\alpha_j=h_{j-1}/(h_{j-1}+h_j),\ j=2,3,\cdots,n-1$$

$$\beta_j=3[(1-\alpha_j)(y_j-y_{j-1})/h_{j-1}+\alpha_j(y_{j+1}-y_j)/h_j]\qquad j=2,3,\cdots,n-1$$

$$a_j=-\alpha_j/[2+(1-\alpha_j)a_{j-1}],\ j=2,3,\cdots,n-1$$

$$b_j=[\beta_j-(1-\alpha_j)b_{j-1}]/[2+(1-\alpha_j)a_{j-1}],\ j=2,3,\cdots,n-1$$

$$y'(x_j)=a_jy'(x_{j+1})+b_j,\ j=n-1,n-2,\cdots,2$$

计算 n 个结点上的二阶导数值 $y''(x_j)(j=1,2,3,\cdots,n)$ 的公式如下：

$$y''(x_j)=6(y_{j+1}-y_j)/h_j^2-2(2y'(x_j)+y'(x_{j+1}))/h_j\qquad j=1,2,\cdots,n-1$$

$$y''(x_n)=6(y_{n-1}-y_n)/h_{n-1}^2+2(2y'(x_n)+y'(x_{n-1}))/h_{n-1}$$

利用各结点上的数值导数与辛卜生公式可以得到插值区间 $[x_1,x_n]$ 上积分值的计算公式为

$$T=\int_{x_1}^{x_n}y(x)\mathrm{d}x$$

$$=\frac{1}{2}\sum_{i=1}^{n-1}(x_{i+1}-x_i)(y_i+y_{i+1})-\frac{1}{24}\sum_{i=1}^{n-1}(x_{i+1}-x_i)^3(y''_i+y''_{i+1})$$

其中 $y''_i=y''(x_i)$。

利用数值导数 $y'(x_i)(i=1,2,\cdots,n)$ 计算插值点 s 处的函数值与导数值的公式如下：

$$y(s)=\left[\frac{3}{h_i^2}(x_{i+1}-s)^2-\frac{2}{h_i^3}(x_{i+1}-s)^3\right]y_i+\left[\frac{3}{h_i^2}(s-x_i)^2-\frac{2}{h_i^3}(s-x_i)^3\right]y_{i+1}$$

$$+h_i\left[\frac{1}{h_i^2}(x_{i+1}-s)^2-\frac{1}{h_i^3}(x_{i+1}-s)^3\right]y'(x_i)$$

$$-h_i\left[\frac{1}{h_i^2}(s-x_i)^2-\frac{1}{h_i^3}(s-x_i)^3\right]y'(x_{i+1})$$

$$y'(s)=\frac{6}{h_i}\left[\frac{1}{h_i^2}(x_{i+1}-s)^2-\frac{1}{h_i}(x_{i+1}-s)\right]y_i-\frac{6}{h_i}\left[\frac{1}{h_i^2}(s-x_i)^2-\frac{1}{h_i}(s-x_i)\right]y_{i+1}$$

$$+\left[\frac{3}{h_i^2}(x_{i+1}-s)^2-\frac{2}{h_i}(x_{i+1}-s)\right]y'(x_i)+\left[\frac{3}{h_i^2}(s-x_i)^2-\frac{2}{h_i}(s-x_i)\right]y'(x_{i+1})$$

$$y''(s)=\frac{1}{h_i^2}\left[6-\frac{12}{h_i}(x_{i+1}-s)\right]y_i+\frac{1}{h_i^2}\left[6-\frac{12}{h_i}(s-x_i)\right]y_{i+1}$$

$$+\frac{1}{h_i}\left[2-\frac{6}{h_i}(x_{i+1}-s)\right]y'(x_i)-\frac{1}{h_i}\left[2-\frac{6}{h_i}(s-x_i)\right]y'(x_{i+1})$$

其中 $s\in[x_i,x_{i+1}]$。

三、子程序语句

SUBROUTINE ESPL1(X,Y,N,DY1,DYN,XX,M,DY,DDY,S,DS,DDS,T,H)

四、形参说明

X,Y——均为双精度实型一维数组,长度为N,输入参数。存放给定N个结点值与函数值。

N——整型变量,输入参数。给定结点的个数。

DY1,DYN——均为双精度实型变量,输入参数。分别为第一个结点与最后一个结点的一阶导数值 $y'(x_1)$,$y'(x_n)$。

XX——双精度实型一维数组,长度为M,输入参数,存放指定的M个插值点值,要求 $x_1<XX(j)<x_n(j=1,2,\cdots,M)$。

M——整型变量,输入参数。指定插值点的个数。

DY,DDY——均为双精度实型一维数组,长度为N,输出参数。返回N个给定结点处的一阶导数值 $y'(x_j)$ 与二阶导数值 $y''(x_j)(j=1,2,\cdots,N)$。

S,DS,DDS——均为双精度实型一维数组,长度为M,输出参数。分别返回M个指定插值点 $XX(j)$ $(j=1,2,\cdots,M)$ 处的函数值、一阶导数值与二阶导数值。

T——双精度实型变量,输出参数。返回插值区间 $[x_1,x_n]$ 上的积分值。

H——双精度实型一维数组,长度为N。本子程序的工作数组。

五、子程序(文件名:ESPL1.FOR)

```
      SUBROUTINE ESPL1(X,Y,N,DY1,DYN,XX,M,DY,DDY,S,DS,DDS,T,H)
      DIMENSION X(N),Y(N),XX(M),DY(N),DDY(N)
      DIMENSION S(M),DS(M),DDS(M),H(N)
      DOUBLE PRECISION X,Y,XX,DY,DDY,S,DS,DDS,H,DY1,DYN,
     *                 T,H0,H1,BETA,ALPHA
      DY(1)=0.0
      H(1)=DY1
      H0=X(2)-X(1)
      DO 10 J=2,N-1
        H1=X(J+1)-X(J)
        ALPHA=H0/(H0+H1)
        BETA=(1.0-ALPHA)*(Y(J)-Y(J-1))/H0
        BETA=3.0*(BETA+ALPHA*(Y(J+1)-Y(J))/H1)
```

```
         DY(J)=-ALPHA/(2.0+(1.0-ALPHA)*DY(J-1))
         H(J)=(BETA-(1.0-ALPHA)*H(J-1))
         H(J)=H(J)/(2.0+(1.0-ALPHA)*DY(J-1))
         H0=H1
10    CONTINUE
      DY(N)=DYN
      DO 20 J=N-1,1,-1
20    DY(J)=DY(J)*DY(J+1)+H(J)
      DO 30 J=1,N-1
30    H(J)=X(J+1)-X(J)
      DO 40 J=1,N-1
        H1=H(J)*H(J)
        DDY(J)=6.0*(Y(J+1)-Y(J))/H1-
     *          2.0*(2.0*DY(J)+DY(J+1))/H(J)
40    CONTINUE
      H1=H(N-1)*H(N-1)
      DDY(N)=6.0*(Y(N-1)-Y(N))/H1+
     *          2.0*(2.0*DY(N)+DY(N-1))/H(N-1)
      T=0.0
      DO 50 I=1,N-1
        H1=0.5*H(I)*(Y(I)+Y(I+1))
        H1=H1-H(I)*H(I)*H(I)*(DDY(I)+DDY(I+1))/24.0
        T=T+H1
50    CONTINUE
      DO 70 J=1,M
        IF (XX(J).GE.X(N)) THEN
          I=N-1
        ELSE
          I=1
60        IF (XX(J).GT.X(I+1)) THEN
            I=I+1
            GOTO 60
          END IF
        END IF
        H1=(X(I+1)-XX(J))/H(I)
        S(J)=(3.0*H1*H1-2.0*H1*H1*H1)*Y(I)
        S(J)=S(J)+H(I)*(H1*H1-H1*H1*H1)*DY(I)
        DS(J)=6.0*(H1*H1-H1)*Y(I)/H(I)
        DS(J)=DS(J)+(3.0*H1*H1-2.0*H1)*DY(I)
        DDS(J)=(6.0-12.0*H1)*Y(I)/(H(I)*H(I))
        DDS(J)=DDS(J)+(2.0-6.0*H1)*DY(I)/H(I)
        H1=(XX(J)-X(I))/H(I)
```

```
      S(J)=S(J)+(3.0*H1*H1-2.0*H1*H1*H1)*Y(I+1)
      S(J)=S(J)-H(I)*(H1*H1-H1*H1*H1)*DY(I+1)
      DS(J)=DS(J)-6.0*(H1*H1-H1)*Y(I+1)/H(I)
      DS(J)=DS(J)+(3.0*H1*H1-2.0*H1)*DY(I+1)
      DDS(J)=DDS(J)+(6.0-12.0*H1)*Y(I+1)/(H(I)*H(I))
      DDS(J)=DDS(J)-(2.0-6.0*H1)*DY(I+1)/H(I)
70    CONTINUE
      RETURN
      END
```

六、例

已知某直升飞机旋转机翼外形曲线上的部分坐标值如下：

x	0.52	8.0	17.95	28.65	50.65	104.6
y	5.28794	13.8400	20.2000	24.9000	31.1000	36.5000
x	156.6	260.7	364.4	468.0	507.0	520.0
y	36.6000	31.0000	20.9000	7.80000	1.50000	0.200000

两端点上的一阶导数值为

$$y'_1 = 1.86548\ ,\ y'_n = -0.046115$$

计算各结点处的一阶与二阶导数值，在区间[0.52,520.0]上的积分值。并且计算在下列八个插值点

4.0,14.0,30.0,60.0,130.0,230.0,450.0,515.0

处的函数值、一阶导数值与二阶导数值。

主程序(文件名:ESPL10.FOR)为

```
      DIMENSION X(12),Y(12),XX(8),DY(12),DDY(12),H(12)
      DIMENSION S(8),DS(8),DDS(8)
      DOUBLE PRECISION X,Y,XX,DY,DDY,S,DS,DDS,T,DY1,DYN,H
      DATA X/0.52,8.0,17.95,28.65,50.65,104.6,156.6,
     *       260.7,364.4,468.0,507.0,520.0/
      DATA Y/5.28794,13.84,20.2,24.9,31.1,36.5,36.6,
     *       31.0,20.9,7.8,1.5,0.2/
      DATA XX/4.0,14.0,30.0,60.0,130.0,230.0,450.0,515.0/
      DY1=1.86548
      DYN=-0.046115
      N=12
      M=8
      CALL ESPL1(X,Y,N,DY1,DYN,XX,M,DY,DDY,S,DS,DDS,T,H)
      WRITE(*,*)
      WRITE(*,10)
```

```
10    FORMAT(6X,'X(I)',10X,'Y(I)',10X,'DY(I)',9X,'DDY(I)')
      WRITE(*,20) (X(I),Y(I),DY(I),DDY(I),I=1,N)
20    FORMAT(1X,4D14.6)
      WRITE(*,*)
      WRITE(*,30) T
30    FORMAT(1X,'T=',D15.6)
      WRITE(*,*)
      WRITE(*,40)
40    FORMAT(6X,'XX(I)',9X,' S(I)',9X,'DS(I)',9X,'DDS(I)')
      WRITE(*,20) (XX(I),S(I),DS(I),DDS(I),I=1,M)
      WRITE(*,*)
      END
```

运行结果为

X(I)	Y(I)	DY(I)	DDY(I)
.520000D+00	.528794D+01	.186548D+01	-.279319D+00
.800000D+01	.138400D+02	.743662D+00	-.2063[illegible]D-01
.179500D+02	.202000D+02	.532912D+00	-.217292D-01
.286500D+02	.249000D+02	.368185D+00	-.906091D-02
.506500D+02	.311000D+02	.208755D+00	-.543268D-02
.104600D+03	.365000D+02	.293142D-01	-.121944D-02
.156600D+03	.366000D+02	-.211539D-01	-.721639D-03
.260700D+03	.310000D+02	-.815142D-01	-.438021D-03
.364400D+03	.209000D+02	-.106449D+00	.428873D-04
.468000D+03	.780000D+01	-.164223D+00	-.107244D-02
.507000D+03	.150000D+01	-.135256D+00	.255796D-02
.520000D+03	.200000D+00	-.461150D-01	.111560D-01

T= .129044D+05

XX(I)	S(I)	DS(I)	DDS(I)
.400000D+01	.103314D+02	.110286D+01	-.158967D+00
.140000D+02	.179266D+02	.617882D+00	-.212939D-01
.300000D+02	.253889D+02	.356103D+00	-.883827D-02
.600000D+02	.328250D+02	.161373D+00	-.470249D-02
.130000D+03	.368774D+02	.142853D-02	-.976284D-03
.230000D+03	.332829D+02	-.667830D-01	-.521663D-03
.450000D+03	.105919D+02	-.146529D+00	-.893563D-03
.515000D+03	.556246D+00	-.936277D-01	.784907D-02

5.14 第二种边界条件的三次样条函数插值、微商与积分

一、功能

根据给定结点上的函数值与第二种边界条件，利用三次样条函数，计算各给定结点上的数值导数及积分，并对一元函数进行成组插值及成组微商。

二、方法说明

设已知函数 $y=f(x)$ 在给定 n 个结点

$$x_1<x_2<\cdots<x_{n-1}<x_n$$

上的函数值

$$y_1,y_2,\cdots,y_{n-1},y_n$$

以及两端点上的二阶导数值 $y''(x_1)$、$y''(x_2)$ 。

计算 n 个结点上的导数值 $y'(x_j)(j=1,2,\cdots,n)$ 的公式如下：

$$a_1=-0.5$$

$$b_1=\frac{3(y_2-y_1)}{2(x_2-x_1)}-\frac{x_2-x_1}{4}y''(x_1)$$

$$h_j=x_{j+1}-x_j,\ j=1,2,\cdots,n-1$$

$$\alpha_j=h_{j-1}/(h_{j-1}+h_j),\ j=2,3,\cdots,n-1$$

$$\beta_j=3[(1-\alpha_j)(y_j-y_{j-1})/h_{j-1}+\alpha_j(y_{j+1}-y_j)/h_j]$$

$$j=2,3,\cdots,n-1$$

$$a_j=-\alpha_j/[2+(1-\alpha_j)a_{j-1}],\ j=2,3,\cdots,n-1$$

$$b_j=[\beta_j-(1-\alpha_j)b_{j-1}]/[2+(1-\alpha_j)a_{j-1}],\ j=2,3,\cdots,n-1$$

$$y'(x_n)=[3(y_n-y_{n-1})/h_{n-1}+y''(x_n)h_{n-1}/2-b_{n-1}]/(2+a_{n-1})$$

$$y'(x_j)=a_jy'(x_{j+1})+b_j,\ j=n-1,n-2,\cdots,1$$

利用各结点上的一阶导数值计算二阶导数值、积分以及指定插值点上的函数值、一阶导数值、二阶导数值，其计算公式与 5.13 节相同。

三、子程序语句

SUBROUTINE ESPL2(X,Y,N,DY1,DYN,XX,M,DY,DDY,S,DS,DDS,T,H)

四、形参说明

X,Y——均为双精度实型一维数组，长度为 N，输入参数。存放给定的 N 个结点值与函数值。

N——整型变量，输入参数。给定结点的个数。

DY1,DYN——均为双精度实型变量，输入参数。分别为第一个结点与最后一个结点处的二阶导数值 $y''(x_1)$，$y''(x_n)$ 。

XX——双精度实型一维数组，长度为 M，输入参数。存放 M 个指定的插值点值，要求 $x_1<XX(j)<x_n(j=1,2,\cdots,M)$ 。

M——整型变量，输入参数。指定插值点个数。

DY,DDY——均为双精度实型一维数组,长度为N,输出参数。分别返回给定N个结点处的一阶导数值 $y'(x_j)$ 与二阶导数值。

S,DS,DDS——均为双精度实型一维数组,长度为M,输出参数。分别返回指定M个插值点处的函数值、一阶导数值、二阶导数值。

T——双精度实型变量,输出参数。返回插值区间 $[x_1,x_n]$ 上的积分值。

H——双精度实型一维数组,长度为N。本子程序的工作数组。

五、子程序(文件名:ESPL2.FOR)

```
      SUBROUTINE ESPL2(X,Y,N,DY1,DYN,XX,M,DY,DDY,S,DS,DDS,T,H)
      DIMENSION X(N),Y(N),XX(M),DY(N),DDY(N)
      DIMENSION S(M),DS(M),DDS(M),H(N)
      DOUBLE PRECISION X,Y,XX,DY,DDY,S,DS,DDS,H,DY1,DYN,
     *                 T,H0,H1,BETA,ALPHA
      DY(1)=-0.5
      H0=X(2)-X(1)
      H(1)=3.0*(Y(2)-Y(1))/(2.0*H0)-DY1*H0/4.0
      DO 10 J=2,N-1
        H1=X(J+1)-X(J)
        ALPHA=H0/(H0+H1)
        BETA=(1.0-ALPHA)*(Y(J)-Y(J-1))/H0
        BETA=3.0*(BETA+ALPHA*(Y(J+1)-Y(J))/H1)
        DY(J)=-ALPHA/(2.0+(1.0-ALPHA)*DY(J-1))
        H(J)=(BETA-(1.0-ALPHA)*H(J-1))
        H(J)=H(J)/(2.0+(1.0-ALPHA)*DY(J-1))
        H0=H1
10    CONTINUE
      DY(N)=(3.0*(Y(N)-Y(N-1))/H1+DYN*H1/2.0-H(N-1))
     *        /(2.0+DY(N-1))
      DO 20 J=N-1,1,-1
20    DY(J)=DY(J)*DY(J+1)+H(J)
      DO 30 J=1,N-1
30    H(J)=X(J+1)-X(J)
      DO 40 J=1,N-1
        H1=H(J)*H(J)
        DDY(J)=6.0*(Y(J+1)-Y(J))/H1-
     *         2.0*(2.0*DY(J)+DY(J+1))/H(J)
40    CONTINUE
      H1=H(N-1)*H(N-1)
      DDY(N)=6.0*(Y(N-1)-Y(N))/H1+
     *          2.0*(2.0*DY(N)+DY(N-1))/H(N-1)
      T=0.0
      DO 50 I=1,N-1
```

```
      H1=0.5*H(I)*(Y(I)+Y(I+1))
      H1=H1-H(I)*H(I)*H(I)*(DDY(I)+DDY(I+1))/24.0
      T=T+H1
50    CONTINUE
      DO 70 J=1,M
        IF (XX(J).GE.X(N)) THEN
          I=N-1
        ELSE
          I=1
60        IF (XX(J).GT.X(I+1)) THEN
            I=I+1
            GOTO 60
          END IF
        END IF
        H1=(X(I+1)-XX(J))/H(I)
        S(J)=(3.0*H1*H1-2.0*H1*H1*H1)*Y(I)
        S(J)=S(J)+H(I)*(H1*H1-H1*H1*H1)*DY(I)
        DS(J)=6.0*(H1*H1-H1)*Y(I)/H(I)
        DS(J)=DS(J)+(3.0*H1*H1-2.0*H1)*DY(I)
        DDS(J)=(6.0-12.0*H1)*Y(I)/(H(I)*H(I))
        DDS(J)=DDS(J)+(2.0-6.0*H1)*DY(I)/H(I)
        H1=(XX(J)-X(I))/H(I)
        S(J)=S(J)+(3.0*H1*H1-2.0*H1*H1*H1)*Y(I+1)
        S(J)=S(J)-H(I)*(H1*H1-H1*H1*H1)*DY(I+1)
        DS(J)=DS(J)-6.0*(H1*H1-H1)*Y(I+1)/H(I)
        DS(J)=DS(J)+(3.0*H1*H1-2.0*H1)*DY(I+1)
        DDS(J)=DDS(J)+(6.0-12.0*H1)*Y(I+1)/(H(I)*H(I))
        DDS(J)=DDS(J)-(2.0-6.0*H1)*DY(I+1)/H(I)
70    CONTINUE
      RETURN
      END
```

六、例

同5.13节中的例。其中边界条件为

$$y_1'' = -0.279319, \quad y_n'' = 0.0111560$$

主程序(文件名:ESPL20.FOR)为

```
      DIMENSION X(12),Y(12),XX(8),DY(12),DDY(12),H(12)
      DIMENSION S(8),DS(8),DDS(8)
      DOUBLE PRECISION X,Y,XX,DY,DDY,S,DS,DDS,T,DY1,DYN,H
      DATA X/0.52,8.0,17.95,28.65,50.65,104.6,156.6,
     *        260.7,364.4,468.0,507.0,520.0/
      DATA Y/5.28794,13.84,20.2,24.9,31.1,36.5,36.6,
```

```
     *          31.0,20.9,7.8,1.5,0.2/
      DATA XX/4.0,14.0,30.0,60.0,130.0,230.0,450.0,515.0/
      DY1=-0.279319
      DYN=0.011156
      N=12
      M=8
      CALL ESPL2(X,Y,N,DY1,DYN,XX,M,DY,DDY,S,DS,DDS,T,H)
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(6X,'X(I)',10X,'Y(I)',10X,'DY(I)',9X,'DDY(I)')
      WRITE(*,20) (X(I),Y(I),DY(I),DDY(I),I=1,N)
20    FORMAT(1X,4D14.6)
      WRITE(*,*)
      WRITE(*,30) T
30    FORMAT(1X,'T=',D15.6)
      WRITE(*,*)
      WRITE(*,40)
40    FORMAT(6X,'XX(I)',9X,' S(I)',9X,'DS(I)',9X,'DDS(I)')
      WRITE(*,20) (XX(I),S(I),DS(I),DDS(I),I=1,M)
      WRITE(*,*)
      END
```

运行结果为

```
      X(I)          Y(I)          DY(I)         DDY(I)
 .520000D+00   .528794D+01   .186548D+01  -.279319D+00
 .800000D+01   .138400D+02   .743662D+00   .206326D-01
 .179500D+02   .202000D+02   .532912D+00  -.217292D-01
 .286500D+02   .249000D+02   .368185D+00  -.906090D-02
 .508500D+02   .311000D+02   .203755D+00  -.543268D-02
 .104600D+03   .365000D+02   .293142D-01  -.121944D-02
 .156600D+03   .366000D+02  -.211539D-01  -.721639D-03
 .260700D+03   .310000D+02  -.815142D-01  -.438021D-03
 .364400D+03   .209000D+02  -.106449D+00  -.428872D-04
 .468000D+03   .780000D+01  -.164223D+00  -.107244D-02
 .507000D+03   .150000D+01  -.135256D+00   .255796D-02
 .520000D+03   .200000D+00  -.461151D-01   .111560D-01

 T=    .129044D+05

      XX(I)         S(I)          DS(I)         DDS(I)
 .400000D+01   .103314D+02   .110284D+01  -.158968D+00
 .140000D+02   .179266D+02   .617882D+00  -.212939D-01
 .300000D+02   .253889D+02   .356103D+00  -.683826D-02
```

.600000D+02	.328250D+02	.161373D+00	−.470249D−02
.130000D+03	.368774D+02	.142853D−02	−.976284D−03
.230000D+03	.332829D+02	.667830D−01	−.521663D−03
.450000D+03	.105919D+02	−.146529D+00	−.893563D−03
.515000D+03	.556246D+00	−.936277D−01	.784906D−02

5.15 第三种边界条件的三次样条函数插值、微商与积分

一、功能

根据给定结点上的函数值与第三种边界条件，利用三次样条函数计算各给定结点上的数值导数与积分，并对一元函数进行成组插值与成组微商。

二、方法说明

设已知函数 $y=f(x)$ 在给定 n 个结点

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

上的函数值

$$y_1, y_2, \cdots, y_{n-1}, y_n$$

以及边界条件

$$y_1 = y_n, \quad y'(x_1) = y'(x_n), \quad y''(x_1) = y''(x_n)$$

计算 n 个结点上的一阶导数值 $y'(x_j)$ $(j=1, 2, \cdots, n)$ 的公式如下：

$$a_1=0,\ b_1=1,\ c_1=0$$

$$h_0=h_{n-1},\ y_1-y_0=y_n-y_{n-1}$$

$$h_j=x_{j+1}-x_j,\ j=1, 2, \cdots, n-1$$

$$\alpha_j=h_{j-1}/(h_{j-1}+h_j),\ j=2, 3, \cdots, n-1$$

$$\beta_j=3[(1-\alpha_j)(y_j-y_{j-1})/h_{j-1}+\alpha_j(y_{j+1}-y_j)/h_j]$$

$$j=2, \cdots, n-1$$

$$a_j=-\alpha_{j-1}/[2+(1-\alpha_{j-1})a_{j-1}],\quad j=2, \cdots, n-1$$

$$b_j=-(1-\alpha_{j-1})b_{j-1}/[2+(1-\alpha_{j-1})a_{j-1}],\quad j=2,3, \cdots, n-1$$

$$c_j=[\beta_{j-1}-(1-\alpha_{j-1})c_{j-1}]/[2+(1-\alpha_{j-1})a_{j-1}],\quad j=2,3, \cdots, n-1$$

$$t_n=1,\quad v_n=0$$

$$t_j=a_jt_{j+1}+b_j,\quad j=n-1, n-2, \cdots, 2$$

$$v_j=a_jv_{j+1}+c_j,\quad j=n-1, n-2, \cdots, 2$$

$$y'(x_{n-1})=[\beta_{n-1}-\alpha_{n-1}v_2-(1-\alpha_{n-1})v_{n-1}]/[2+\alpha_{n-1}t_2+(1-\alpha_{n-1})t_{n-1}]$$

$$y'(x_j)=t_{j+1}y'(x_{n-1})+v_{j+1},\quad j=1, 2, \cdots, n-2$$

$$y'(x_n)=y'(x_1)$$

利用各结点上的一阶导数值计算二阶导数值、积分以及指定插值点处的函数值、一阶导数值与二阶导数值，其计算公式与 5.13 节相同。

三、子程序语句

SUBROUTINE ESPL3(X,Y,N,XX,M,DY,DDY,S,DS,DDS,T,H)

四、形参说明

X,Y——均为双精度实型一维数组，长度为N，输入参数。存放给定的N个结点值与函数值。

N——整型变量，输入参数。给定结点的个数。

XX——双精度实型一维数组，长度为M，输入参数。存放指定的M个插值点值。

M——整型变量，输入参数。指定插值点的个数。

DY,DDY——均为双精度实型一维数组，长度为N，输出参数。分别返回N个给定结点上的一阶导数值 $y'(x_j)$ 与二阶导数值 $y''(x_j)$ ($j=1,2,\cdots,N$)。

S,DS,DDS——均为双精度实型一维数组，长度为M，输出参数。分别返回指定的M个插值点处的函数值、一阶导数值与二阶导数值。

T——双精度实型变量，输出参数。返回插值区间$[x_1,x_n]$上的积分值。

H——双精度实型一维数组，长度为N。本子程序的工作数组。

五、子程序(文件名:ESPL3.FOR)

```
      SUBROUTINE ESPL3(X,Y,N,XX,M,DY,DDY,S,DS,DDS,T,H)
      DIMENSION X(N),Y(N),XX(M),DY(N),DDY(N)
      DIMENSION S(M),DS(M),DDS(M),H(N)
      DOUBLE PRECISION X,Y,XX,DY,DDY,S,DS,DDS,H,Y0,Y1,
     *                 T,H0,H1,BETA,ALPHA
      H0=X(N)-X(N-1)
      Y0=Y(N)-Y(N-1)
      DY(1)=0.0
      DDY(1)=0.0
      DDY(N)=0.0
      H(1)=1.0
      H(N)=1.0
      DO 10 J=2,N
        H1=H0
        Y1=Y0
        H0=X(J)-X(J-1)
        Y0=Y(J)-Y(J-1)
        ALPHA=H1/(H1+H0)
        BETA=3.0*((1.0-ALPHA)*Y1/H1+ALPHA*Y0/H0)
        IF (J.LT.N) THEN
          T=2.0+(1.0-ALPHA)*DY(J-1)
          DY(J)=-ALPHA/T
          H(J)=(ALPHA-1.0)*H(J-1)/T
          DDY(J)=(BETA-(1.0-ALPHA)*DDY(J-1))/T
        END IF
10    CONTINUE
      DO 20 J=N-1,2,-1
```

```
      H(J)=DY(J) * H(J+1)+H(J)
      DDY(J)=DY(J) * DDY(J+1)+DDY(J)
20    CONTINUE
      DY(N-1)=(BETA-ALPHA * DDY(2)-(1.0-ALPHA) * DDY(N-1))/
     *        (ALPHA * H(2)+(1.0-ALPHA) * H(N-1)+2.0)
      DO 25 J=3,N
25    DY(J-2)=H(J-1) * DY(N-1)+DDY(J-1)
      DY(N)=DY(1)
      DO 30 J=1,N-1
30    H(J)=X(J+1)-X(J)
      DO 40 J=1,N-1
        H1=H(J) * H(J)
        DDY(J)=6.0 * (Y(J+1)-Y(J))/H1-
     *        2.0 * (2.0 * DY(J)+DY(J+1))/H(J)
40    CONTINUE
      H1=H(N-1) * H(N-1)
      DDY(N)=6.0 * (Y(N-1)-Y(N))/H1+
     *          2.0 * (2.0 * DY(N)+DY(N-1))/H(N-1)
      T=0.0
      DO 50 I=1,N-1
        H1=0.5 * H(I) * (Y(I)+Y(I+1))
        H1=H1-H(I) * H(I) * H(I) * (DDY(I)+DDY(I+1))/24.0
        T=T+H1
50    CONTINUE
      DO 70 J=1,M
        H0=XX(J)
55      IF (H0.GE.X(N)) THEN
          H0=H0-(X(N)-X(1))
          GOTO 55
        END IF
56      IF (H0.LT.X(1)) THEN
          H0=H0+(X(N)-X(1))
          GOTO 56
        END IF
        I=1
60      IF (H0.GT.X(I+1)) THEN
          I=I+1
          GOTO 60
        END IF
        H1=(X(I+1)-H0)/H(I)
        S(J)=(3.0 * H1 * H1-2.0 * H1 * H1 * H1) * Y(I)
        S(J)=S(J)+H(I) * (H1 * H1-H1 * H1 * H1) * DY(I)
```

```
      DS(J)=6.0*(H1*H1-H1)*Y(I)/H(I)
      DS(J)=DS(J)+(3.0*H1*H1-2.0*H1)*DY(I)
      DDS(J)=(6.0-12.0*H1)*Y(I)/(H(I)*H(I))
      DDS(J)=DDS(J)+(2.0-6.0*H1)*DY(I)/H(I)
      H1=(H0-X(I))/H(I)
      S(J)=S(J)+(3.0*H1*H1-2.0*H1*H1*H1)*Y(I+1)
      S(J)=S(J)-H(I)*(H1*H1-H1*H1*H1)*DY(I+1)
      DS(J)=DS(J)-6.0*(H1*H1-H1)*Y(I+1)/H(I)
      DS(J)=DS(J)+(3.0*H1*H1-2.0*H1)*DY(I+1)
      DDS(J)=DDS(J)+(6.0-12.0*H1)*Y(I+1)/(H(I)*H(I))
      DDS(J)=DDS(J)-(2.0-6.0*H1)*DY(I+1)/H(I)
70    CONTINUE
      RETURN
      END
```

六、例

已知间隔为10°的sinx函数表，计算间隔为5°时的sinx函数表，并计算其一阶与二阶导数值，计算在一个周期内的积分值。

主程序（文件名：ESPL30.FOR）为

```
      DIMENSION X(37),Y(37),XX(36),DY(37),DDY(37),H(37)
      DIMENSION S(36),DS(36),DDS(36)
      DOUBLE PRECISION X,Y,XX,DY,DDY,S,DS,DDS,T,H
      DO 10 I=0,36
        X(I+1)=I*6.2831852/36.0
        Y(I+1)=SIN(X(I+1))
10    CONTINUE
      DO 20 I=0,35
20    XX(I+1)=(0.5+I)*6.2831852/36.0
      N=37
      M=36
      CALL ESPL3(X,Y,N,XX,M,DY,DDY,S,DS,DDS,T,H)
      WRITE(*,*)
      WRITE(*,30)
30    FORMAT(3X,'X(I)',5X,'Y(I)=SIN(X)',3X,'DY(I)=COS(X)',
     *           2X,'DDY(I)=-SIN(X)')
      WRITE(*,50) X(1),Y(1),DY(1),DDY(1)
      DO 40 I=1,36
        Z=XX(I)*36.0/0.62831852
        WRITE(*,50) Z,S(I),DS(I),DDS(I)
        Z=X(I+1)*36.0/0.62831852
        WRITE(*,50) Z,Y(I+1),DY(I+1),DDY(I+1)
40    CONTINUE
```

```
      WRITE(*,*)
50    FORMAT(1X,F5.1,3F14.6)
      WRITE(*,60) T
60    FORMAT(1X,'T=',D15.6)
      WRITE(*,*)
      END
```

运行结果为

X(I)	Y(I)=SIN(X)	DY(I)=COS(X)	DDY(I)=−SIN(X)
.0	.000000	.999995	.000000
5.0	.087156	.996197	—.087045
10.0	.173648	.984803	—.174089
15.0	.258818	.965928	—.258489
20.0	.342020	.939688	—.342889
25.0	.422617	.906310	—.422080
30.0	.500000	.866021	—.501271
35.0	.573575	.819154	—.572846
40.0	.642788	.766041	—.644421
45.0	.707105	.707108	—.706206
50.0	.766044	.642784	—.767991
55.0	.819150	.573578	—.818108
60.0	.866025	.499997	—.868226
65.0	.906306	.422619	—.905153
70.0	.939693	.342018	—.942080
75.0	.965923	.258820	—.964695
80.0	.984808	.173647	—.987310
85.0	.996192	.087156	—.994926
90.0	1.000000	.000000	—1.002541
95.0	.996192	—.087156	—.994926
100.0	.984808	—.173647	—.987310
105.0	.965923	—.258820	—.964695
110.0	.939693	—.342018	—.942080
115.0	.906306	—.422619	—.905153
120.0	.866025	—.499997	—.868226
125.0	.819150	—.573578	—.818109
130.0	.766045	—.642784	—.767991
135.0	.707105	—.707108	—.706206
140.0	.642788	—.766040	—.644421
145.0	.573575	—.819154	—.572846
150.0	.500000	—.866021	—.501271
155.0	.422617	—.906310	—.422080
160.0	.342020	—.939688	—.342889
165.0	.258819	—.965928	—.258489
170.0	.173648	—.984803	—.174090
175.0	.087156	—.996197	—.087045
180.0	.000000	—.999995	.000000
185.0	—.087155	—.996197	.087045
190.0	—.173648	—.984803	.174089
195.0	—.258818	—.965928	.258489
200.0	—.342020	—.939688	.342889
205.0	—.422617	—.906310	.422080
210.0	—.500000	—.866021	.501270
215.0	—.573575	—.819154	.572846

220.0	-.642787	-.766041	.644421
225.0	-.707105	-.707108	.706206
230.0	-.766044	-.642784	.767991
235.0	-.819150	-.573576	.818108
240.0	-.866025	-.499998	.868226
245.0	-.906305	-.422619	.905153
250.0	-.939693	-.342019	.942080
255.0	-.965923	-.258820	.964695
260.0	-.984808	-.173647	.987310
265.0	-.996192	-.087156	.994926
270.0	-1.000000	.000000	1.002541
275.0	-.996192	.087156	.994926
280.0	-.984808	.173647	.987310
285.0	-.965924	.258819	.964695
290.0	-.939693	.342018	.942081
295.0	-.906306	.422619	.905153
300.0	-.866026	.499997	.868226
305.0	-.819150	.573577	.818109
310.0	-.766045	.642784	.767991
315.0	-.707105	.707108	.706206
320.0	-.642788	.766040	.644421
325.0	-.573575	.819154	.572846
330.0	-.500000	.866021	.501271
335.0	-.422617	.906310	.422080
340.0	-.342020	.939688	.342889
345.0	-.258819	.965928	.258490
350.0	-.173648	.984803	.174090
355.0	-.087156	.996197	.087045
360.0	.000000	.999995	.000000

T=　.459112D-13

5.16　二元三点插值

一、功能

根据给定矩形域上 $n\times m$ 个结点上的函数值，用二元三点插值公式计算指定插值点处的函数值。

二、方法说明

已知矩形域上 $n\times m$ 个结点在两个方向上的坐标分别为

$$x_1<x_2<\cdots<x_{n-1}<x_n$$

$$y_1<y_2<\cdots<y_{m-1}<y_m$$

其相应的函数值为

$$z_{ij}=z(x_i,\ y_j),\ i=1,\ 2,\ \cdots,\ n;\ j=1,\ 2,\ \cdots,\ m$$

计算插值点$(u,\ v)$处的函数值 $w=z(u,\ v)$。

选取最靠近插值点$(u,\ v)$的 9 个结点，设其在两个方向上的坐标分别为 $x_p<x_{p+1}<x_{p+2}$及 $y_q<y_{q+1}<y_{q+2}$，然后用二元三点插值公式

$$z(x,y)=\sum_{i=p}^{p+2}\sum_{j=q}^{q+2}\left(\prod_{\substack{k=p\\k\neq i}}^{p+2}\frac{x-x_k}{x_i-x_k}\right)\left(\prod_{\substack{l=q\\l\neq j}}^{q+2}\frac{y-y_l}{y_j-y_l}\right)z_{ij}$$

计算插值点 (u, v) 处的函数近似值。

三、子程序语句

SUBROUTINE ESLQ3(X,Y,Z,N,M,U,V,W)

四、形参说明

X——双精度实型一维数组，长度为N，输入参数。矩形区域X方向的N个坐标。

Y——双精度实型一维数组，长度为M，输入参数。矩形区域Y方向的M个坐标。

Z——双精度实型二维数组，体积为N×M，输入参数。存放矩形域N×M个结点上的函数值

$$z_{ij}=z(x_i, y_j),\quad i=1,2,\cdots,N;\ j=1,2,\cdots,M$$

N——整型变量，输入参数。给定结点在X方向的坐标个数。

M——整型变量，输入参数。给定结点在Y方向的坐标个数。

U,V——均为双精度实型变量，输入参数。存放指定插值点的X坐标与Y坐标。

W——双精度实型变量，输出参数。返回指定插值点(U, V)处的二元函数近似值。

五、子程序(文件名:ESLQ3.FOR)

```
      SUBROUTINE ESLQ3(X,Y,Z,N,M,U,V,W)
      DIMENSION X(N),Y(M),Z(N,M),B(3)
      DOUBLE PRECISION X,Y,Z,U,V,W,B,HH
      NN=3
      IF (N.LE.3) THEN
        IP=1
        NN=N
      ELSE IF (U.LE.X(2)) THEN
        IP=1
      ELSE IF (U.GE.X(N-1)) THEN
        IP=N-2
      ELSE
        I=1
        J=N
10      IF (IABS(I-J).NE.1) THEN
          L=(I+J)/2
          IF (U.LT.X(L)) THEN
            J=L
          ELSE
            I=L
          END IF
          GOTO 10
        END IF
        IF (ABS(U-X(I)).LT.ABS(U-X(J))) THEN
```

```
          IP=I-1
        ELSE
          IP=I
        END IF
      END IF
      MM=3
      IF (M.LE.3) THEN
        IQ=1
        MM=M
      ELSE IF (V.LE.Y(2)) THEN
        IQ=1
      ELSE IF (V.GE.Y(M-1)) THEN
        IQ=M-2
      ELSE
        I=1
        J=M
20      IF (IABS(J-I).NE.1) THEN
          L=(I+J)/2
          IF (V.LT.Y(L)) THEN
            J=L
          ELSE
            I=L
          END IF
          GOTO 20
        END IF
        IF (ABS(V-Y(I)).LT.ABS(V-Y(J))) THEN
          IQ=I-1
        ELSE
          IQ=I
        END IF
      END IF
      DO 50 I=1,NN
        B(I)=0.0
        DO 40 J=1,MM
          HH=Z(IP+I-1,IQ+J-1)
          DO 30 K=1,MM
            IF (K.NE.J) THEN
              HH=HH*(V-Y(IQ+K-1))/(Y(IQ+J-1)-Y(IQ+K-1))
            END IF
30        CONTINUE
          B(I)=B(I)+HH
40      CONTINUE
```

```
50    CONTINUE
      W=0.0
      DO 70 I=1,NN
        HH=B(I)
        DO 60 J=1,NN
          IF (J.NE.I) THEN
            HH=HH*(U-X(IP+J-1))/(X(IP+I-1)-X(IP+J-1))
          END IF
60      CONTINUE
        W=W+HH
70    CONTINUE
      RETURN
      END
```

六、例

设二元函数

$$z(x, y) = e^{-(x-y)}$$

取以下 6×5 个结点：

$$x_i=0.2(i-1), \quad i=1, 2, \cdots, 6$$

$$y_j=0.25(j-1), \quad j=1, 2, \cdots, 5$$

上的函数值

$$z(x_i, y_j) = e^{-(x_i-y_j)}, \quad i = 1, 2, \cdots, 6; \ j = 1, 2, \cdots, 5$$

用二元三点插值公式计算插值点(0.9, 0.8)及(0.3, 0.9)处的函数近似值。

主程序(文件名:ESLQ30.FOR)为

```
      DIMENSION X(6),Y(5),Z(6,5)
      DOUBLE PRECISION X,Y,Z,U,V,W
      DO 10 I=1,6
10    X(I)=0.2*(I-1)
      DO 20 J=1,5
20    Y(J)=0.25*(J-1)
      DO 30 I=1,6
      DO 30 J=1,5
30    Z(I,J)=EXP(-(X(I)-Y(J)))
      U=0.9
      V=0.8
      CALL ESLQ3(X,Y,Z,6,5,U,V,W)
      WRITE(*,40) U,V,W
40    FORMAT(1X,'X=',F7.3,5X,'Y=',F7.3,10X,'Z(X,Y)=',D15.6)
      U=0.3
      V=0.9
```

```
      CALL ESLQ3(X,Y,Z,6,5,U,V,W)
      WRITE(*,40) U,V,W
      END
```

运行结果为

```
X=  .900   Y=  .800        Z(X,Y)=   .904786D+00
X=  .300   Y=  .900        Z(X,Y)=   .182460D+01
```

5.17 二元全区间插值

一、功能

根据给定矩形域上 $n \times m$ 个结点上的函数值，用二元插值公式计算指定插值点处的函数值。

二、方法说明

已知矩形域上 $n \times m$ 结点的两个方向的坐标分别为

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

$$y_1 < y_2 < \cdots < y_{m-1} < y_m$$

其相应的函数值为

$$z_{ij} = z(x_i, y_j),\ i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, m$$

计算插值点 (u, v) 处的函数值 $w = z(u, v)$。

以插值点 (u, v) 为中心，在 X 方向上，前后各取四个坐标

$$x_p < x_{p+1} < x_{p+2} < x_{p+3} < u < x_{p+4} < x_{p+5} < x_{p+6} < x_{p+7}$$

在 Y 方向上，前后也各取四个坐标

$$y_q < y_{q+1} < y_{q+2} < y_{q+3} < v < y_{q+4} < y_{q+5} < y_{q+6} < y_{q+7}$$

然后用二元插值公式

$$z(x, y) = \sum_{i=p}^{p+7} \sum_{j=q}^{q+7} \left(\prod_{\substack{k=p \\ k \neq i}}^{p+7} \frac{x - x_k}{x_i - x_k} \right) \left(\prod_{\substack{l=q \\ l \neq j}}^{q+7} \frac{y - y_l}{y_j - y_l} \right) z_{ij}$$

计算插值点 (u, v) 处的函数近似值。

三、子程序语句

SUBROUTINE　ESLGQ(X,Y,Z,N,M,U,V,W)

四、形参说明

X——双精度实型一维数组，长度为 N，输入参数。存放矩形区域 X 方向的 N 个坐标值。

Y——双精度实型一维数组，长度为 M，输入参数。存放矩形区域 Y 方向的 M 个坐标值。

Z——双精度实型二维数组，体积为 N×M，输入参数。存放矩形区域上 N×M 个结点上的函数值。

N——整型变量，输入参数。矩形区域 X 方向的坐标个数。

M——整型变量，输入参数。矩形区域 Y 方向的坐标个数。

U,V——均为双精度实型变量,输入参数。存放指定插值点的 X 坐标与 Y 坐标。

W——双精度实型变量,输出参数。返回指定插值点(U, V)处的函数近似值。

五、子程序(文件名:ESLGQ.FOR)

```
      SUBROUTINE ESLGQ(X,Y,Z,N,M,U,V,W)
      DIMENSION X(N),Y(M),Z(N,M),B(10)
      DOUBLE PRECISION X,Y,Z,U,V,W,B,HH
      IF (U.LE.X(1)) THEN
        IP=1
        IPP=4
      ELSE IF (U.GE.X(N)) THEN
        IP=N-3
        IPP=N
      ELSE
        I=1
        J=N
10      IF (IABS(I-J).NE.1) THEN
          L=(I+J)/2
          IF (U.LT.X(L)) THEN
            J=L
          ELSE
            I=L
          END IF
          GOTO 10
        END IF
        IP=I-3
        IPP=I+4
      END IF
      IF (IP.LT.1) IP=1
      IF (IPP.GT.N) IPP=N
      IF (V.LE.Y(1)) THEN
        IQ=1
        IQQ=4
      ELSE IF (V.GE.Y(M)) THEN
        IQ=M-3
        IQQ=M
      ELSE
        I=1
        J=M
20      IF (IABS(J-I).NE.1) THEN
          L=(I+J)/2
          IF (V.LT.Y(L)) THEN
```

```
            J=L
          ELSE
            I=L
          END IF
          GOTO 20
        END IF
        IQ=I-3
        IQQ=I+4
      END IF
      IF (IQ.LT.1) IQ=1
      IF (IQQ.GT.M) IQQ=M
      DO 50 I=IP,IPP
        B(I-IP+1)=0.0
        DO 40 J=IQ,IQQ
          HH=Z(I,J)
          DO 30 K=IQ,IQQ
            IF (K.NE.J) THEN
              HH=HH*(V-Y(K))/(Y(J)-Y(K))
            END IF
30        CONTINUE
          B(I-IP+1)=B(I-IP+1)+HH
40      CONTINUE
50    CONTINUE
      W=0.0
      DO 70 I=IP,IPP
        HH=B(I-IP+1)
        DO 60 J=IP,IPP
          IF (J.NE.I) THEN
            HH=HH*(U-X(J))/(X(I)-X(J))
          END IF
60      CONTINUE
        W=W+HH
70    CONTINUE
      RETURN
      END
```

六、例

设二元函数

$$z(x, y) = e^{-(x-y)}$$

取下列 11×11 个结点：

$$x_i=0.1(i-1), \quad i=1, 2, \cdots, 11$$

$$y_j=0.1(j-1), \quad j=1, 2, \cdots, 11$$

上的函数值

$$z(x_i, y_j) = e^{-(x_i-y_j)},\ i = 1, 2, \cdots, 11;\ j = 1, 2, \cdots, 11$$

用二元插值法计算插值点(0.35，0.65)及(0.45，0.55)处的函数近似值。

主程序(文件名:ESLGQ0.FOR)为

```
      DIMENSION X(11),Y(11),Z(11,11)
      DOUBLE PRECISION X,Y,Z,U,V,W
      DO 10 I=1,11
      X(I)=0.1*(I-1)
10    Y(I)=X(I)
      DO 30 I=1,11
      DO 30 J=1,11
30    Z(I,J)=EXP(-(X(I)-Y(J)))
      U=0.35
      V=0.65
      CALL ESLGQ(X,Y,Z,11,11,U,V,W)
      WRITE(*,40) U,V,W
40    FORMAT(1X,'X=',F7.3,5X,'Y=',F7.3,10X,'Z(X,Y)=',D15.6)
      U=0.45
      V=0.55
      CALL ESLGQ(X,Y,Z,11,11,U,V,W)
      WRITE(*,40) U,V,W
      END
```

运行结果为

```
X=   .350     Y=   .650          Z(X,Y)=     .134986D+01
X=   .450     Y=   .550          Z(X,Y)=     .110517D+01
```

第6章 数值积分

6.1 变步长梯形求积法

一、功能

用变步长梯形求积法计算定积分 $\int_a^b f(x)\mathrm{d}x$

二、方法说明

设定积分为

$$T=\int_a^b f(x)\mathrm{d}x$$

首先用梯形公式计算

$$T_n = h[f(a)+f(b)]/2$$

其中 n=1,h=b−a。

然后用递推公式

$$T_{2n}=\frac{1}{2}T_n+\frac{h}{2}\sum_{k=0}^{n-1} f\left(x_k+\frac{h}{2}\right)$$

$$2n \Rightarrow n, \quad \frac{h}{2} \Rightarrow h$$

计算每次二等分后的积分值,直到满足

$$|T_{2n}-T_n|<\varepsilon$$

为止。

三、子程序语句

SUBROUTINE　FFFTS(A,B,F,EPS,T)

四、形参说明

A,B——均为双精度实型变量,输入参数,积分的下限与上限。

F——双精度实型函数子程序名,输入参数。用于计算被积函数值 $f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编,其语句形式为:

DOUBLE PRECISION FUNCTION F(X)

其中:X 为双精度实型变量,自变量值;函数名 F 返回双精度实型函数值。

EPS——实型变量,输入参数,控制精度要求。

T——双精度实型变量,输出参数,返回积分值。

五、子程序(文件名:FFFTS.FOR)

```
      SUBROUTINE FFFTS(A,B,F,EPS,T)
```

```
      DOUBLE PRECISION A,B,F,T,FA,FB,H,T1,S,X
      FA=F(A)
      FB=F(B)
      N=1
      H=B-A
      T1=H*(FA+FB)/2.0
5     S=0.0
      DO 10 K=0,N-1
        X=A+(K+0.5)*H
        S=S+F(X)
10    CONTINUE
      T=(T1+H*S)/2.0
      IF (ABS(T1-T).GE.EPS) THEN
        T1=T
        N=N+N
        H=H/2.0
        GOTO 5
      END IF
      RETURN
      END
```

六、例

计算定积分

$$T = \int_0^1 e^{-x^2} dx$$

$\varepsilon=0.000001$。

主程序及计算被积函数的子程序(文件名:FFFTS0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,A,B,T
      A=0.0
      B=1.0
      EPS=0.000001
      CALL FFFTS(A,B,F,EPS,T)
      WRITE(*,10) T
10    FORMAT(1X,'T=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=EXP(-X*X)
      RETURN
      END
```

运行结果为

T=.746824D+00

6.2 变步长辛卜生求积法

一、功能

用变步长辛卜生(Simpson)求积法计算定积分

$$\int_a^b f(x)\mathrm{d}x$$

二、方法说明

设定积分为

$$S=\int_a^b f(x)\mathrm{d}x$$

(1) 首先用梯形公式计算

$$T_n=h[f(a)+f(b)]/2$$

其中n=1,h=b−a

(2) 然后用变步长梯形法则计算

$$T_{2n}=\frac{1}{2}T_n+\frac{h}{2}\sum_{k=0}^{n-1}f\left(x_k+\frac{h}{2}\right)$$

(3) 最后用辛卜生求积公式计算

$$S_{2n}=(4T_{2n}-T_n)/3$$

(4) 若$|S_{2n}-S_n|<\varepsilon$,则过程结束,S_{2n}即为所求积分的近似值;否则令 $2n\Rightarrow n$,$\frac{h}{2}\Rightarrow h$,重复(2)与(3)。

三、子程序语句

SUBROUTINE FSIMP(A,B,F,EPS,T)

四、形参说明

A,B——均为双精度实型变量,输入参数。积分的下限与上限。

F——双精度实型函数子程序名,输入参数。用于计算被积函数值$f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明,该子程序由用户自编,其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中,X为双精度实型变量,自变量值;函数名F返回双精度实型函数值。

EPS——实型变量,输入参数。控制精度要求。

T——双精度实型变量,输出参数。返回积分值。

五、子程序(文件名:FSIMP.FOR)

```
SUBROUTINE FSIMP(A,B,F,EPS,T)
DOUBLE PRECISION A,B,F,T,H,T1,S1,P,X,T2,S2
```

```
      N=1
      H=B-A
      T1=H*(F(A)+F(B))/2.0
      S1=T1
10    P=0.0
      DO 20 K=0,N-1
        X=A+(K+0.5)*H
        P=P+F(X)
20    CONTINUE
      T2=(T1+H*P)/2.0
      S2=(4*T2-T1)/3.0
      IF (ABS(S2-S1).GE.EPS) THEN
        T1=T2
        N=N+N
        H=H/2.0
        S1=S2
        GOTO 10
      END IF
      T=S2
      RETURN
      END
```

六、例

计算定积分

$$S=\int_0^1 \frac{\ln(1+x)}{1+x^2}\mathrm{d}x$$

$\varepsilon=0.000001$。

主程序及计算被积函数值的子程序(文件名:FSIMP0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,A,B,S
      A=0.0
      B=1.0
      EPS=0.000001
      CALL FSIMP(A,B,F,EPS,S)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=LOG(1+X)/(1+X*X)
      RETURN
```

```
      END
```

运行结果为

S=.272198D+00

6.3 自适应梯形求积法

一、功能

用自适应梯形求积法计算被积函数为强峰的定积分 $\int_a^b f(x)\,dx$。

二、方法说明

自适应梯形求积法的过程如下：

(1) 将积分区间$[a,b]$分成两个相等的子区间(称为1级子区间)$\Delta_0^{(1)}$与$\Delta_1^{(1)}$，在每一个子区间上分别用梯形公式计算积分近似值，其结果分别记为$t_0^{(1)}$与$t_1^{(1)}$。

(2) 将子区间$\Delta_0^{(1)}$分为两个相等的子区间(称为2级子区间)$\Delta_0^{(2)}$与$\Delta_1^{(2)}$，在每一个子区间上分别用梯形公式计算积分近似值$t_0^{(2)}$与$t_1^{(2)}$。如果满足不等式

$$|t_0^{(1)}-(t_0^{(2)}+t_1^{(2)})|<\varepsilon/1.4$$

则保留$t_0^{(2)}$与$t_1^{(2)}$。再将$\Delta_1^{(1)}$也分为两个相等的2级子区间$\Delta_2^{(2)}$与$\Delta_3^{(2)}$，其相应的积分近似值为$t_2^{(2)}$与$t_3^{(2)}$。如果满足不等式

$$|t_1^{(1)}-(t_2^{(2)}+t_3^{(2)})|<\varepsilon/1.4$$

则保留$t_2^{(2)}$与$t_3^{(2)}$。最后得到积分的近似值为

$$t=t_0^{(2)}+t_1^{(2)}+t_2^{(2)}+t_3^{(2)}$$

(3) 如果在上述不等式中有一个不成立，则将对应的2级子区间再分割成两个相等的3级子区间。在考虑3级子区间时，其精度要求变为$\varepsilon/1.4^2$。

对3级子区间中不满足精度要求的子区间又可以分割成两个相等的4级子区间，精度要求变为$\varepsilon/1.4^3$。以此类推，这个过程一直进行到所考虑的所有子区间上都满足精度要求为止。

在本子程序中，分割的最大深度为30。

三、子程序语句

SUBROUTINE FFPTS(A,B,F,EPS,T)

四、形参说明

A,B——均为双精度实型变量，输入参数。积分的下限与上限。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中：X为双精度实型变量，自变量值；函数名F返回双精度实型函数值。

EPS——实型变量，输入参数。控制精度要求。

T——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名:FFPTS.FOR)

```
      SUBROUTINE FFPTS(A,B,F,EPS,T)
      DIMENSION S(30,7)
      DOUBLE PRECISION A,B,F,T,S,F0,F1,P,H,X,F3,T1,T2
      T=0.0
      F0=F(A)
      F1=F(B)
      P=H*(F0+F1)/2.0
      K=1
      S(K,1)=A
      S(K,2)=B
      S(K,3)=B-A
      S(K,4)=F0
      S(K,5)=F1
      S(K,6)=P
      S(K,7)=EPS
10    IF (K.NE.0) THEN
        H=S(K,3)
        X=S(K,1)+H/2.0
        F3=F(X)
        T1=H*(S(K,4)+F3)/4.0
        T2=H*(F3+S(K,5))/4.0
        IF ((ABS(S(K,6)-T1-T2).LT.S(K,7)).OR.(K.GE.29)) THEN
          T=T+(T1+T2)
          K=K-1
        ELSE
          S(K+1,1)=X
          S(K+1,2)=S(K,2)
          S(K,2)=X
          S(K,3)=H/2.0
          S(K+1,3)=S(K,3)
          S(K+1,4)=F3
          S(K+1,5)=S(K,5)
          S(K,5)=F3
          S(K,7)=S(K,7)/1.4
          S(K+1,7)=S(K,7)
          S(K,6)=T1
          S(K+1,6)=T2
          K=K+1
        END IF
        GOTO 10
```

```
      END IF
      RETURN
      END
```

六、例

计算定积分

$$S=\int_{-1}^{1}\frac{dx}{1+25x^2}$$

$\varepsilon=0.000001$。

主程序及计算被积函数值的子程序(文件名:FFPTS0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,A,B,S
      A=-1.0
      B=1.0
      EPS=0.000001
      CALL FFPTS(A,B,F,EPS,S)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=1.0/(1.0+25*X*X)
      RETURN
      END
```

运行结果为

```
S=.549363D+00
```

6.4 龙贝格求积法

一、功能

用龙贝格(Romberg)求积法计算定积分

$$\int_a^b f(x)dx$$

二、方法说明

龙贝格求积公式为

$$T_{m+1}(h)=\frac{4^m T_m(h/2)-T_m(h)}{4^m-1}$$

其中 $T_m(h)$是步长为 h 时 2m－2 阶牛顿-柯特斯(Newton-Cotes)公式计算结果，$T_m\left(\frac{h}{2}\right)$是步长为$\frac{h}{2}$时 $2m-2$ 阶牛顿-柯特斯公式计算结果。$T_1(h)$是步长为 h 时梯形公式计算结果，$T_1\left(\frac{h}{2}\right)$是步长为 $\frac{h}{2}$时梯形公式计算结果。

在实际计算时，一直计算到$|T_{m+1}(h)-T_m(h)|<\varepsilon$为止。

在本子程序中，最多可以计算到$m=10$，如果此时还达不到精度要求，就取T_{11}作为结果。

三、子程序语句

SUBROUTINE FROMB(A,B,F,EPS,T)

四、形参说明

A,B——均为双精度实型变量，输入参数。积分的下限与上限。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值$f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明，该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中：X为双精度实型变量，自变量值；函数名F返回双精度实型函数值。

EPS——实型变量，输入参数。控制精度要求。

T——双精度实型变量，输出参数。返回积分值。

五、子程序（文件名：FROMB.FOR）

```
      SUBROUTINE FROMB(A,B,F,EPS,T)
      DIMENSION Y(10)
      DOUBLE PRECISION A,B,F,T,Y,H,P,S,Q
      H=B-A
      Y(1)=H*(F(A)+F(B))/2.0
      M=1
      N=1
10    P=0.0
      DO 20 I=0,N-1
20    P=P+F(A+(I+0.5)*H)
      P=(Y(1)+H*P)/2.0
      S=1.0
      DO 30 K=1,M
        S=4*S
        Q=(S*P-Y(K))/(S-1)
        Y(K)=P
        P=Q
30    CONTINUE
      IF ((ABS(Q-Y(M)).GE.EPS).AND.(M.LE.9)) THEN
        M=M+1
        Y(M)=Q
        N=N+N
        H=H/2.0
        GOTO 10
      END IF
```

```
      T=Q
      RETURN
      END
```

六、例

计算定积分

$$S=\int_0^1\frac{x}{4+x^2}dx$$

ε=0.000001。

主程序及计算被积函数值的子程序（文件名：FROMB0.FOR）为

```
      EXTERNAL F
      DOUBLE PRECISION F,A,B,S
      A=0.0
      B=1.0
      EPS=0.000001
      CALL FROMB(A,B,F,EPS,S)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=X/(4+X*X)
      RETURN
      END
```

运行结果为

S=.111572D+00

6.5 计算一维积分的连分式法

一、功能

用连分式法计算定积分

$$\int_a^b f(x)dx$$

二、方法说明

利用变步长梯形法则算出步长分别为

$$h_i=(b-a)/2^{i-1},i=1,2,\cdots$$

的一系列积分近似值 $S_i(i=1,2,\cdots)$。显然，积分近似值 S 是步长 h 的函数 S(h)。

将函数 S(h)用下列连分式表示：

$$S(h)=b_1+\frac{h-h_1}{b_2}+\frac{h-h_2}{b_3}+\cdots+\frac{h-h_i}{b_{i+1}}+\cdots$$

其中$b_1,b_2,\cdots,b_i,\cdots$可以由一系列的积分近似值点$(h_i,S_i)$ $(i=1,2,\cdots)$来确定，其计算公式为

$$\begin{cases} u = S_i \\ u = (h_i - h_{j-1})/(u - b_{j-1}),\ j = 2,\cdots,i \\ b_i = u \end{cases}$$

当$h=0$时，$S(0)$即为积分值。即

$$S=S(0)=b_1-\cfrac{h_1}{b_2-\cfrac{h_2}{b_3-\cdots-\cfrac{h_i}{b_{i+1}-\cdots}}}$$

在实际计算时，一般取到七节就能满足精度要求。在本子程序中，最多可以取到十节，如果此时还不满足精度要求，就直接取当前近似值。

三、子程序语句

SUBROUTINE FFPQG(A,B,F,EPS,G,L)

四、形参说明

A,B——均为双精度实型变量，输入参数。积分的下限与上限。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值$f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中：X 为双精度实型变量，自变量值；函数名 F 返回双精度实型函数值。

EPS——实型变量，输入参数。控制精度要求。

G——双精度实型变量，输出参数。返回积分值。

L——整型变量，输出参数。返回标志。若$L=0$，表示连分式取到十节还未满足精度要求；若$L\neq0$，表示正常返回。

五、子程序(文件名:FFPQG.FOR)

```
      SUBROUTINE FFPQG(A,B,F,EPS,G,L)
      DIMENSION BB(10),H(10)
      DOUBLE PRECISION A,B,F,G,BB,H,HH,T1,S1,S,X,T2
      M=1
      N=1
      H(1)=B-A
      HH=B-A
      T1=HH*(F(A)+F(B))/2.0
      S1=T1
      BB(1)=S1
5     S=0.0
      DO 10 K=0,N-1
        X=A+(K+0.5)*HH
        S=S+F(X)
10    CONTINUE
```

```
      T2=(T1+HH*S)/2.0
      M=M+1
      H(M)=H(M-1)/2.0
      G=T2
      DO 20 J=2,M
        S=G-BB(J-1)
        IF (ABS(S)+1.0.EQ.1.0) THEN
          S=SIGN(1.0D+35,S)
          G=S*SIGN(1.0D0,H(M)-H(J-1))
        ELSE
          G=(H(M)-H(J-1))/S
        END IF
20    CONTINUE
      BB(M)=G
      G=BB(M)
      DO 30 J=M,2,-1
        IF (ABS(G)+1.0.EQ.1.0) THEN
          G=SIGN(1.0D+35,G)
          G=G*SIGN(1.0D0,H(J-1))
          G=BB(J-1)-G
        ELSE
          G=BB(J-1)-H(J-1)/G
        END IF
30    CONTINUE
      IF ((ABS(G-S1).GE.EPS).AND.(M.LE.9)) THEN
        N=N+N
        T1=T2
        S1=G
        HH=HH/2.0
        GOTO 5
      END IF
      IF (M.GE.10) THEN
        L=0
      ELSE
        L=1
      END IF
      RETURN
      END
```

六、例

计算定积分

$$G=\int_0^{4.3}e^{-x^2}dx$$

$\varepsilon=0.000001$。

主程序及计算被积函数值的子程序(文件名:FFPQG0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,A,B,G
      A=0.0
      B=4.3
      EPS=0.000001
      CALL FFPQG(A,B,F,EPS,G,L)
      WRITE(*,10) L,G
10    FORMAT(1X,'L=',I2,5X,'G=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=EXP(-X*X)
      RETURN
      END
```

运行结果为

```
L=1    G=.886227D+00
```

6.6 高振荡函数求积法

一、功能

用分部积分法计算高振荡函数的积分

$$\int_a^b f(x)\sin mx\mathrm{d}x \text{ 与} \int_a^b f(x)\cos mx\mathrm{d}x$$

二、方法说明

当 m 较大时,积分

$$S_1(m)=\int_a^b f(x)\cos mx\mathrm{d}x$$

与

$$S_2(m)=\int_a^b f(x)\sin mx\mathrm{d}x$$

称为高振荡积分。

令

$$S(m)=\int_a^b f(x)\mathrm{e}^{jmx}\mathrm{d}x$$

其中 $\mathrm{e}^{jmx}=\cos mx+j\sin mx$,则有

$$S(m)=S_1(m)+jS_2(m)$$

反复利用分部积分法可以得到

$$S(m)=\int_a^b f(x)e^{jmx}dx$$
$$\approx\sum_{k=0}^{n-1}\left(\frac{j}{m}\right)^{k+1}\left[\left(f^{(k)}(b)\cos mb-f^{(k)}(a)\cos ma\right)+j\left(f^{(k)}(b)\sin mb-f^{(k)}(a)\sin ma\right)\right]$$

分离出实部与虚部后得到

$$S_1(m)=\int_a^b f(x)\cos mx\,dx$$
$$\approx\sum_{k=0}^{n-1}\frac{1}{m^{k+1}}\left[f^{(k)}(b)\sin\left(\frac{k\pi}{2}+mb\right)-f^{(k)}(a)\sin\left(\frac{k\pi}{2}+ma\right)\right]$$
$$S_2(m)=\int_a^b f(x)\sin mx\,dx$$
$$\approx\sum_{k=0}^{n-1}\frac{-1}{m^{k+1}}\left[f^{(k)}(b)\cos\left(\frac{k\pi}{2}+mb\right)-f^{(k)}(a)\cos\left(\frac{k\pi}{2}+ma\right)\right]$$

其中

$$\sin\left(\frac{k\pi}{2}+x\right)=\begin{cases}\sin x,\ \mathrm{mod}(k,4)=0\\ \cos x,\ \mathrm{mod}(k,4)=1\\ -\sin x,\ \mathrm{mod}(k,4)=2\\ -\cos x,\ \mathrm{mod}(k,4)=3\end{cases}$$

$$\cos\left(\frac{k\pi}{2}+x\right)=\begin{cases}\cos x,\ \mathrm{mod}(k,4)=0\\ -\sin x,\ \mathrm{mod}(k,4)=1\\ -\cos x,\ \mathrm{mod}(k,4)=2\\ \sin x,\ \mathrm{mod}(k,4)=3\end{cases}$$

在积分过程中，n 的大小可根据 $f(x)$ 的性质及 m 的大小而定．一般取 $n=3$ 左右。

三、子程序语句

SUBROUTINE FPART (AA,BB,N,M,A,B,S1,S2)

四、形参说明

AA,BB——均为双精度实型变量，输入参数．积分的下限与上限．

N——整型变量，输入参数。其意义见方法说明，但要注意，这里的 N 相当于方法说明中的 $n-1$。

M——整型变量，输入参数。被积函数中振荡函数的角频率。

A——双精度实型一维数组，长度为 N，输入参数。其中

$$A(i)=f^{(i-1)}(AA),\ i=1,2,\cdots,N$$

B——双精度实型一维数组，长度为 N，输入参数。其中

$$B(i)=f^{(i-1)}(BB),\ i=1,2,\cdots,N$$

S1——双精度实型变量，输出参数。返回积分 $\int_{AA}^{BB}f(x)\cos mx\,dx$。

S2——双精度实型变量，输出参数。返回积分 $\int_{AA}^{BB}f(x)\sin mx\,dx$。

五、子程序(文件名:FPART.FOR)

```
      SUBROUTINE FPART(AA,BB,N,M,A,B,S1,S2)
      DIMENSION A(N),B(N),SA(4),SB(4),CA(4),CB(4)
      DOUBLE PRECISION AA,BB,A,B,S1,S2,SA,SB,CA,CB
      SMA=SIN(M*AA)
      SMB=SIN(M*BB)
      CMA=COS(M*AA)
      CMB=COS(M*BB)
      SA(1)=SMA
      SA(2)=CMA
      SA(3)=-SMA
      SA(4)=-CMA
      SB(1)=SMB
      SB(2)=CMB
      SB(3)=-SMB
      SB(4)=-CMB
      CA(1)=CMA
      CA(2)=-SMA
      CA(3)=-CMA
      CA(4)=SMA
      CB(1)=CMB
      CB(2)=-SMB
      CB(3)=-CMB
      CB(4)=SMB
      S1=0.0
      S2=0.0
      MM=1
      DO 10 K=1,N
        L=K
20      IF (L.GE.5) THEN
          L=L-4
          GOTO 20
        END IF
        MM=MM*M
        S1=S1+(B(K)*SB(L)-A(K)*SA(L))/(1.0*MM)
        S2=S2+(B(K)*CB(L)-A(K)*CA(L))/(1.0*MM)
10    CONTINUE
      S2=-S2
      RETURN
      END
```

六、例

计算积分

$$S_1 = \int_0^{2\pi} x\cos x\cos 30x\mathrm{d}x \text{ 与 } S_2 = \int_0^{2\pi} x\cos x\sin 30x\mathrm{d}x$$

取 N=4，则有

A(1)=$f(0)$=0，　　A(2)=$f'(0)$=1.0

A(3)=$f''(0)$=0，　　A(4)=$f'''(0)$=−3.0

B(1)=$f(2\pi)$=6.2831852，　　B(2)=$f'(2\pi)$=1.0

B(3)=$f''(2\pi)$=−6.2831852，　　B(4)=$f'''(2\pi)$=−3.0

其中 $f(x)=x\cos x$。

主程序(文件名:FPART0.FOR)为

```
      DIMENSION A(4),B(4)
      DOUBLE PRECISION A,B,AA,BB,S1,S2
      DATA A/0.0,1.0,0.0,-3.0/
      DATA B/6.2831852,1.0,-6.2831852,-3.0/
      AA=0.0
      BB=6.2831852
      M=30
      N=4
      CALL FPART(AA,BB,N,M,A,B,S1,S2)
      WRITE(*,10) S1,S2
10    FORMAT(1X,'S1=',D15.6,'      S2=',D15.6)
      END
```

运行结果为

S1=−.189958D−05　　S2=−.209672D+00

七、附注

本子程序只对高振荡积分有效。

6.7 勒让德-高斯求积法

一、功能

用勒让德-高斯(Legendre-Gauss)求积法计算定积分

$$\int_a^b f(x)\mathrm{d}x。$$

二、方法说明

设定积分为

$$g = \int_a^b f(x)\mathrm{d}x$$

首先作变换

$$x = \frac{b-a}{2}t + \frac{b+a}{2}$$

将原积分化为在区间[−1,1]上的积分，即

$$g=\int_a^b f(x)\mathrm{d}x=\frac{b-a}{2}\int_{-1}^{1} f\left[\frac{b-a}{2}t+\frac{b+a}{2}\right]\mathrm{d}t=\frac{b-a}{2}\int_{-1}^{1}\varphi(t)\mathrm{d}t$$

根据插值求积公式有

$$\int_{-1}^{1}\varphi(t)\mathrm{d}t=\sum_{k=1}^{n}\lambda_k\varphi(t_k)$$

如果 n 个插值结点取勒让德多项式

$$P_n(t)=\frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}t^n}[(t^2-1)^n]$$

在区间[−1,1]上的 n 个零点，则插值求积公式的代数精确度为 2n−1。

在本子程序中，取 n=5，并且采用变步长求积法。

插值结点为

$t_1=-0.9061798459$,　$t_2=-0.5384693101$,　$t_3=0.0$,

$t_4=\ 0.5384693101$,　$t_5=0.9061798459$

求积系数为

$\lambda_1=0.2369268851$,　$\lambda_2=0.4786286705$,　$\lambda_3=0.5688888889$,

$\lambda_4=0.4786286705$,　$\lambda_5=0.2369268851$

三、子程序语句

SUBROUTINE FLRGS (A,B,F,EPS,G)

四、形参说明

A,B——均为双精度实型变量，输入参数。积分的下限与上限。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值 $f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中：X 为双精度实型变量，自变量值；函数名返回双精度实型函数值。

EPS——实型变量，输入参数。控制精度要求。

G——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名:FLRGS.FOR)

```
      SUBROUTINE FLRGS(A,B,F,EPS,G)
      DIMENSION T(5),C(5)
      DOUBLE PRECISION A,B,F,G,T,C,S,P,H,AA,BB,W,X,Q
      DATA T/-0.9061798459,-0.5384693101,0.0,
     *        0.5384693101,0.9061798459/
      DATA C/0.2369268851,0.4786286705,0.5688888889,
     *        0.4786286705,0.2369268851/
      M=1
      S=(B-A)*0.001
      P=0.0
10    H=(B-A)/M
```

```
      G=0.0
      DO 30  I=1,M
        AA=A+(I-1)*H
        BB=A+I*H
        W=0.0
        DO 20 J=1,5
          X=((BB-AA)*T(J)+(BB+AA))/2.0
          W=W+F(X)*C(J)
20      CONTINUE
        G=G+W
30    CONTINUE
      G=G*H/2.0
      Q=ABS(G-P)/(1.0+ABS(G))
      IF ((Q.GE.EPS).AND.(ABS(H).GT.ABS(S))) THEN
        P=G
        M=M+1
        GOTO 10
      END IF
      RETURN
      END
```

六、例

计算定积分

$$G = \int_{2.5}^{8.4} (x^2 + \sin x) dx$$

$\epsilon=0.000001$。

主程序及计算被积函数值的子程序(文件名:FLRGS0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,A,B,G
      A=2.5
      B=8.4
      EPS=0.000001
      CALL FLRGS(A,B,F,EPS,G)
      WRITE(*,10) G
10    FORMAT(1X,'G=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=X*X+SIN(X)
      RETURN
      END
```

运行结果为

G=.192078D+03

6.8 拉盖尔-高斯求积法

一、功能

用拉盖尔-高斯(Laguerre-Gauss)求积公式计算半无限区间[0,∞)上的积分

$$g=\int_0^{\infty} f(x)\mathrm{d}x$$

二、方法说明

设半无限区间[0,∞)上的积分为

$$g=\int_0^{\infty} f(x)\mathrm{d}x$$

n 点拉盖尔-高斯求积公式为

$$g=\sum_{i=1}^{n}\lambda_i f(x_i)$$

其中 $x_i(i=1,2,\cdots,n)$ 为 n 阶拉盖尔多项式

$$L_n(x)=\mathrm{e}^x\frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^n\mathrm{e}^{-x}),\ 0\leqslant x<+\infty$$

的 n 个零点；λ_i 为求积系数。

在本子程序中，取 $n=5$。

五阶拉盖尔-高斯求积公式的结点为

$x_1=0.26355990$，$x_2=1.41340290$，$x_3=3.59642600$，

$x_4=7.08580990$，$x_5=12.64080000$

求积系数为

$\lambda_1=0.6790941054$，$\lambda_2=1.638487956$，$\lambda_3=2.769426772$，

$\lambda_4=4.315944000$，$\lambda_5=7.104896230$

本方法特别适用于计算如下形式的积分：

$$\int_0^{\infty}\mathrm{e}^{-x}g(x)\mathrm{d}x$$

三、子程序语句

SUBROUTINE FLAGS(F,G)

四、形参说明

F——双精度实型函数子程序名，输入参数。用于计算被积函数值 $f(x)$，在主程序中必须用外部语句及类型说明语句对相应的实参进行说明，该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中：X 为双精度实型变量，自变量值；函数名 F 返回双精度实型函数值。

G——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名:FLAGS.FOR)

```
      SUBROUTINE FLAGS(F,G)
      DIMENSION T(5),C(5)
      DOUBLE PRECISION F,G,T,C,X
      DATA C/0.6790941054,1.638487956,2.769426772,
     *       4.31594400,7.104896230/
      DATA T/0.26355990,1.41340290,3.59642600,
     *       7.08580990,12.64080000/

      G=0.0D0
      DO 10 I=1,5
        X=T(I)
        G=G+F(X)*C(I)
10    CONTINUE
      END
```

六、例

计算半无限区间积分

$$G=\int_0^{\infty} x e^{-x} dx$$

主程序及计算被积函数值的子程序(文件名:FLAGS0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,G
      CALL FLAGS(F,G)
      WRITE(*,10) G
10    FORMAT(1X,'G=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=EXP(-X)*X
      RETURN
      END
```

运行结果为

```
G=.999995D+00
```

6.9 埃尔米特-高斯求积法

一、功能

用埃尔米特-高斯(Hermite-Gauss)求积公式计算无限区间$(-\infty,\infty)$上的积分

$$g=\int_{-\infty}^{\infty} f(x) dx$$

二、方法说明

设无限区间$(-\infty,\infty)$上的积分为

$$g=\int_{-\infty}^{\infty}f(x)\mathrm{d}x$$

n点埃尔米特-高斯求积公式为

$$g\doteq\sum_{i=1}^{n}\lambda_i f(x_i)$$

其中$x_i(i=1,2,\cdots,n)$为n阶埃尔米特多项式

$$H_n(x)=(-1)^n e^{x^2}\frac{\mathrm{d}^n}{\mathrm{d}x^n}(e^{-x^2}),\quad -\infty<x<+\infty$$

的n个零点;λ_i为求积系数。

在本子程序中,取$n=5$。

五阶埃尔米特-高斯求积公式的结点为

$x_1=-2.02018200,\quad x_2=-0.95857190,\quad x_3=0.0$

$x_4=0.95857190,\quad x_5=2.02018200$

求积系数为

$\lambda_1=1.181469599,\quad \lambda_2=0.9865791417,\quad \lambda_3=0.9453089237,$

$\lambda_4=0.9865791417,\quad \lambda_5=1.181469599$

本方法特别适用于计算如下形式的积分

$$\int_{-\infty}^{\infty}e^{-x^2}g(x)\mathrm{d}x$$

三、子程序语句

SUBROUTINE FHMGS(F,G)

四、形参说明

F——双精度实型函数子程序名,输入参数,用于计算被积函数值$f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中:X为双精度实型变量,自变量值;函数名F返回双精度实型函数值。

G——双精度实型变量,输出参数。返回积分值。

五、子程序(文件名:FHMGS.FOR)

```
      SUBROUTINE FHMGS(F,G)
      DIMENSION T(5),C(5)
      DOUBLE PRECISION F,G,T,C,X
      DATA C/1.181469599,0.9865791417,0.9453089237,
     *        0.9865791417,1.181469599/
      DATA T/-2.02018200,-0.95857190,0.0,
     *        0.95857190,2.02018200/
```

```
      G=0.0D0
      DO 10 I=1,5
        X=T(I)
      G=G+F(X)*C(I)
10    CONTINUE
      END
```

六、例

计算无限区间积分

$$G=\int_{-\infty}^{\infty}x^2e^{-x^2}dx$$

主程序及计算被积函数值的子程序(文件名:FHMGS0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,G
      CALL FHMGS(F,G)
      WRITE(*,10) G
10    FORMAT(1X,'G=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=EXP(-X*X)*X*X
      RETURN
      END
```

运行结果为

G=.886223D+00

6.10 切比雪夫求积法

一、功能

用切比雪夫(Chebyshev)求积法计算定积分 $\int_a^b f(x)dx$。

二、方法说明

设定积分为

$$S=\int_a^b f(x)dx$$

首先作变换

$$x=\frac{b-a}{2}t+\frac{b+a}{2}$$

将原积分化为在区间[-1,1]上的积分

$$S=\frac{b-a}{2}\int_{-1}^{1}f\left[\frac{b-a}{2}t+\frac{b+a}{2}\right]dt=\frac{b-a}{2}\int_{-1}^{1}\varphi(t)dt$$

切比雪夫求积公式为

$$\int_{-1}^{1}\varphi(t)\mathrm{d}t=\frac{2}{n}\sum_{k=1}^{n}\varphi(t_k)$$

在本子程序中，取 n=5，并且采用变步长求积法。

当 n=5 时，其结点为

$$t_1=-0.8324975,\quad t_2=-0.3745414,\quad t_3=0.0$$

$$t_4=0.3745414,\qquad t_5=0.8324975$$

三、子程序语句

SUBROUTINE FCBSV(A,B,F,EPS,S)

四、形参说明

A,B——均为双精度实型变量，输入参数。积分的下限与上限。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值 $f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中：X 为双精度实型变量，自变量值；函数名 F 返回双精度实型函数值。

EPS——实型变量，输入参数。控制精度要求。

S——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名：FCBSV.FOR)

```
      SUBROUTINE FCBSV(A,B,F,EPS,S)
      DIMENSION T(5)
      DOUBLE PRECISION A,B,F,S,T,D,P,H,G,Q,AA,BB,X
      DATA T/-0.8324975,-0.3745414,0.0,0.3745414,0.8324975/
      M=1
      D=(B-A)*0.001
      P=0.0
10    H=(B-A)/M
      G=0.0
      DO 30 I=1,M
        AA=A+(I-1)*H
        BB=A+I*H
        S=0.0
        DO 20 J=1,5
          X=((BB-AA)*T(J)+(BB+AA))/2.0
          S=S+F(X)
20      CONTINUE
        G=G+S
30    CONTINUE
      G=G*H/5.0
      Q=ABS(G-P)/(1+ABS(G))
```

```
      IF ((Q.GE.EPS).AND.(ABS(H).GT.ABS(D))) THEN
        P=G
        M=M+1
        GOTO 10
      END IF
      S=G
      RETURN
      END
```

六、例

计算定积分

$$S = \int_{2.5}^{8.4}(x^2 + \sin x)\mathrm{d}x$$

$\varepsilon=0.000001$。

主程序及计算被积函数值的子程序(文件名:FCBSV0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,A,B,S
      A=2.5
      B=8.4
      EPS=0.000001
      CALL FCBSV(A,B,F,EPS,S)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D15.6)
      END

      FUNCTION F(X)
      DOUBLE PRECISION F,X
      F=X*X+SIN(X)
      RETURN
      END
```

运行结果为

S=.192078D+03

6.11 计算一维积分的蒙特卡洛法

一、功能

用蒙特卡洛(Monte Carlo)法计算定积分。

二、方法说明

设定积分为

$$S = \int_a^b f(x)\mathrm{d}x$$

取 0 到 1 之间均匀分布的随机数序列 $t_i(i=1,2,\cdots,m)$，并令

$$x_i = a + (b-a)t_i,\quad i=1,2,\cdots,m$$

只要 m 足够大，则有

$$S \approx \frac{b-a}{m}\sum_{i=1}^{m} f(x_i)$$

在本子程序中，取 $m=10000$。

三、子程序语句

SUBROUTINE FMTCL(A,B,F,S)

四、形参说明

A,B——均为双精度实型变量，输入参数。积分的下限与上限。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值 $f(x)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X)

其中：X 为双精度实型变量，自变量值；函数名 F 返回双精度实型被积函数值。

S——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名：FMTCL.FOR)

```
      SUBROUTINE FMTCL(A,B,F,S)
      DOUBLE PRECISION A,B,F,S,R,X,K
      REAL M,NRND1
      R=1.0
      M=10000.0
      K=10000.0D0
      S=0.0D0
10    IF (M+1.0.NE.1.0) THEN
        M=M-1.0
        X=A+(B-A)*NRND1(R)
        S=S+F(X)/K
        GOTO 10
      END IF
      S=S*(B-A)
      END
```

六、例

用蒙特卡洛法计算定积分

$$S = \int_{2.5}^{8.4}(x^2 + \sin x)\mathrm{d}x$$

其中 A=2.5，B=8.4。

主程序及计算被积函数值的子程序(文件名：FMTCL0.FOR)为

```
      EXTERNAL F
      DOUBLE PRECISION F,S,A,B
```

```
      DATA A,B/2.5D0,8.4D0/
      CALL FMTCL(A,B,F,S)
      WRITE(*,*)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D13.6)
      WRITE(*,*)
      END

      DOUBLE PRECISION FUNCTION F(X)
      DOUBLE PRECISION X
      F=X*X+SIN(X)
      END
```

运行结果为

S=.191553D+03

七、附注

本子程序需要调用产生 0 到 1 之间均匀分布的一个随机数子程序 NRND1，参看 13.1 节。

6.12 变步长辛卜生二重积分法

一、功能

用变步长辛卜生(Simpson)方法计算二重积分。

二、方法说明

设二重积分为

$$S=\int_a^b \mathrm{d}x\int_{y_1(x)}^{y_2(x)} f(x,\ y)\mathrm{d}y$$

将此二重积分化为两个单积分，即

$$g(x)=\int_{y_1(x)}^{y_2(x)} f(x,\ y)\mathrm{d}y \text{ 与 } S=\int_a^b g(x)\mathrm{d}x$$

对每个单积分采用变步长辛卜生方法。其计算步骤如下：

1. 固定一个 x，设为 $\bar{x}$。

(1) 用梯形公式计算

$$t_1=[y_2(\bar{x})-y_1(\bar{x})][f(\bar{x},\ y_1(\bar{x}))+f(\bar{x},\ y_2(\bar{x}))]/2$$

(2) 将区间分半，每一个子区间长度为

$$h_k=[y_2(\bar{x})-y_1(\bar{x})]/2^k,\ k=1,\ 2,\ \cdots$$

用辛卜生公式计算

$$t_{k+1}=\frac{1}{2}t_k+h_k\sum_{i=1}^{n} f[\bar{x},\ y_1(\bar{x})+(2i-1)h_k]$$

$$g_k=(4t_{k+1}-t_k)/3$$

其中 $n=2^{k-1}$。

重复(2)，直到$|g_k-g_{k-1}|<\varepsilon(1+|g_k|)$为止，此时有$g(x)\approx g_k$。

2. 利用1. 中所计算的一系列$g(x)$的值计算二重积分的近似值S。

(1) 用梯形公式计算

$$T_1=(b-a)[g(b)+g(a)]/2$$

(2) 将区间二等分，其子区间长度为

$$H_k=(b-a)/2^k,\ k=1,2,\cdots$$

用辛卜生公式计算

$$T_{k+1}=\frac{1}{2}T_k+H_k\sum_{i=1}^{n}g[a+(2i-1)H_k]$$

$$S_k=(4T_{k+1}-T_k)/3$$

其中$n=2^{k-1}$。

重复(2)，直到$|S_k-S_{k-1}|<\varepsilon(1+|S_k|)$为止，此时有$S\approx S_k$。

三、子程序语句

SUBROUTINE FSIM2 (A,B,FS,F,EPS,S)

四、形参说明

A,B——均为双精度实型变量，输入参数。外层积分的下限与上限。

FS——子程序名，输入参数。用于计算内层积分的下限$y_1(x)$与上限$y_2(x)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE FS(X,Y1,Y2)

其中：X,Y1,Y2均为双精度实型变量，分别为自变量值、积分下限$y_1(x)$、积分上限$y_2(x)$。

F——双精度实型函数子程序名，输入参数，用于计算被积函数值$f(x,y)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X,Y)

其中：X,Y均为双精度实型变量，自变量值；函数名F返回双精度实型函数值。

EPS——实型变量，输入参数。控制精度要求。

S——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名:FSIM2.FOR)

```
      SUBROUTINE FSIM2(A,B,FS,F,EPS,S)
      EXTERNAL FS,F
      DOUBLE PRECISION A,B,F,S,H,S1,S2,TS1,TS2,X,G,S0,C
      N=1
      H=0.5*(B-A)
      C=(B-A)*1.0E-06
      CALL SIMP1(A,FS,F,EPS,S1)
      CALL SIMP1(B,FS,F,EPS,S2)
      TS1=H*(S1+S2)
 10   X=A-H
```

```
      TS2=0.5*TS1
      DO 20 J=1,N
        X=X+2*H
        CALL SIMP1(X,FS,F,EPS,G)
        TS2=TS2+H*G
20    CONTINUE
      S=(4*TS2-TS1)/3.0
      N=N+N
      IF (N.GE.16) THEN
        IF (ABS(S-S0).LE.EPS*(ABS(S)+1.0)) RETURN
      END IF
      S0=S
      TS1=TS2
      H=0.5*H
      IF (ABS(H).GE.ABS(C)) GOTO 10
      RETURN
      END

      SUBROUTINE SIMP1(X,FS,F,EPS,G)
      DOUBLE PRECISION X,F,G,Y1,Y2,H,C,TS1,TS2,Y,G0
      N=1
      CALL FS(X,Y1,Y2)
      H=0.5*(Y2-Y1)
      C=(Y2-Y1)*1.0E-06
      TS1=H*(F(X,Y1)+F(X,Y2))
10    Y=Y1-H
      TS2=0.5*TS1
      DO 20 I=1,N
        Y=Y+2*H
        TS2=TS2+H*F(X,Y)
20    CONTINUE
      G=(4*TS2-TS1)/3.0
      N=N+N
      IF (N.GE.16) THEN
        IF (ABS(G-G0).LE.EPS*(ABS(G)+1.0)) RETURN
      END IF
      G0=G
      TS1=TS2
      H=0.5*H
      IF (ABS(H).GE.ABS(C)) GOTO 10
      RETURN
      END
```

六、例

计算二重积分

$$S=\int_0^1 dx\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} e^{x^2+y^2}dy$$

$\varepsilon=0.0001$。

主程序与计算内层积分上下限、计算被积函数值的子程序(文件名:FSIM20.FOR)为

```
      EXTERNAL FS,F
      DOUBLE PRECISION F,A,B,S
      A=0.0
      B=1.0
      EPS=0.0001
      CALL FSIM2(A,B,FS,F,EPS,S)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D13.6)
      END

      SUBROUTINE FS(X,Y1,Y2)
      DOUBLE PRECISION X,Y1,Y2,Q
      Q=SQRT(1.0-X*X)
      Y1=-Q
      Y2=Q
      RETURN
      END

FUNCTION F(X,Y)
DOUBLE PRECISION F,X,Y
F=EXP(X*X+Y*Y)
RETURN
END
```

运行结果为

S=.269893D+01

6.13 计算多重积分的高斯方法

一、功能

用高斯(Gauss)方法计算多重积分。

二、方法说明

设多重积分为

$$S=\int_a^b dx_1\int_{c_1(x_1)}^{d_1(x_1)} dx_2\cdots\int_{c_{n-1}(x_1,\cdots,x_{n-1})}^{d_{n-1}(x_1,\cdots,x_{n-1})} f(x_1,x_2,\cdots,x_n)dx_n$$

在计算 n 重积分时，分别将 1，2，…，n 层区间分为相等的 $JS_1, JS_2, \cdots, JS_n$ 个子区间。首先求出各积分区间上的第一个子区间中第一个高斯型点($\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_n$)；然后固定 $\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_{n-1}$，按高斯方法计算最内层(即第 n 层)积分，再从内到外计算各层积分值。最后就得到所求的 n 重积分的近似值。

计算单重积分的高斯方法见 6.7 节的方法说明。

三、子程序语句

SUBROUTINE FGAUS(N,JS,X,FS,F,S)

四、形参说明

N——整型变量，输入参数。积分重数，要求 N≤10。

JS——整型一维数组，长度为 N，输入参数。其中 JS(i) (i=1, 2, …, N)表示第 i 层积分区间所分的子区间个数。

X——双精度实型一维数组，长度为 N。本子程序的工作数组，用于存放 N 个积分变量 $x_1, x_2, \cdots, x_n$ 的值。

FS——子程序名，输入参数。用于计算各层积分的上下限，在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE FS(J,N,X,DN,UP)

其中：J 为整型变量，积分层号；N 为整型变量，积分重数；X 为双精度实型一维数组，长度为 N，存放 N 个积分变量的值；DN 与 UP 均为双精度实型变量，返回第 J 层积分的下限与上限值。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值 $f(x_1, x_2, \cdots, x_n)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(N,X)

其中：N 为整型变量，积分重数；X 为双精度实型一维数组，长度为 N，存放 N 个积分变量的值，函数名 F 返回双精度实型函数值。

S——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名：FGAUS.FOR)

```
      SUBROUTINE FGAUS(N,JS,X,FS,F,S)
      DIMENSION JS(N),X(N)
      DIMENSION T(5),C(5),D(2,11),CC(11),IS(2,11)
      DOUBLE PRECISION X,F,S,T,C,D,CC,DN,UP,P
      DATA T/-0.9061798459,-0.5384693101,0.0,
     *           0.5384693101,0.9061798459/
      DATA C/0.2369268851,0.4786286705,0.5688888889,
     *           0.4786286705,0.2369268851/
      M=1
      D(1,N+1)=1.0
      D(2,N+1)=1.0
10    DO 20 J=M,N
```

```
      CALL FS(J,N,X,DN,UP)
      D(1,J)=0.5*(UP-DN)/JS(J)
      CC(J)=D(1,J)+DN
      X(J)=D(1,J)*T(1)+CC(J)
      D(2,J)=0.0
      IS(1,J)=1
      IS(2,J)=1
20    CONTINUE
      J=N
30    K=IS(1,J)
      IF (J.EQ.N) THEN
        P=F(N,X)
      ELSE
        P=1.0
      END IF
      D(2,J)=D(2,J+1)*D(1,J+1)*P*C(K)+D(2,J)
      IS(1,J)=IS(1,J)+1
      IF (IS(1,J).GT.5) THEN
        IF (IS(2,J).GE.JS(J)) THEN
          J=J-1
          IF (J.EQ.0) THEN
            S=D(2,1)*D(1,1)
            RETURN
          END IF
          GOTO 30
        END IF
        IS(2,J)=IS(2,J)+1
        CC(J)=CC(J)+D(1,J)*2.0
        IS(1,J)=1
      END IF
      K=IS(1,J)
      X(J)=D(1,J)*T(K)+CC(J)
      IF (J.EQ.N) GOTO 30
      M=J+1
      GOTO 10
      END
```

六、例

计算三重积分

$$S=\int_0^1 dx\int_0^{\sqrt{1-x^2}}dy\int_{\sqrt{x^2+y^2}}^{\sqrt{2-x^2-y^2}}z^2dz$$

其中第 1,2,3 层的积分区间均分成 4 个子区间，变量 x,y,z 分别用 x_1，x_2，x_3 表示，

主程序及计算各层积分的上下限、计算被积函数的子程序(文件名:FGAUS0.FOR)为

```
      EXTERNAL FS,F
      DIMENSION JS(3),X(3)
      DOUBLE PRECISION F,S,X
      DATA JS/4,4,4/
      N=3
      CALL FGAUS(N,JS,X,FS,F,S)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D13.6)
      END

      SUBROUTINE FS(J,N,X,DN,UP)
      DIMENSION X(N)
      DOUBLE PRECISION X,DN,UP,Q
      IF (J.EQ.1) THEN
        DN=0.0
        UP=1.0
      ELSE IF (J.EQ.2) THEN
        DN=0.0
        UP=SQRT(1.0-X(1)*X(1))
      ELSE IF (J.EQ.3) THEN
        Q=X(1)*X(1)+X(2)*X(2)
        DN=SQRT(Q)
        UP=SQRT(2.0-Q)
      END IF
      RETURN
      END

      FUNCTION F(N,X)
      DIMENSION X(N)
      DOUBLE PRECISION F,X
      F=X(3)*X(3)
      RETURN
      END
```

运行结果为

$S=.382944D+00$

6.14 计算二重积分的连分式法

一、功能

用连分式法计算二重积分。

二、方法说明

设二重积分为

$$S=\int_a^b \mathrm{d}x\int_{y_1(x)}^{y_2(x)} f(x,y)\mathrm{d}y$$

将此二重积分化为两个单积分，即

$$S(x)=\int_{y_1(x)}^{y_2(x)} f(x,y)\mathrm{d}y,\qquad S=\int_a^b S(x)\mathrm{d}x$$

利用连分式法计算每个单积分。其计算步骤如下。

1. 固定一个 x，设为 $\bar{x}$。

用连分式法计算

$$S_k(\bar{x})=b_1-\frac{h_1}{b_2}-\frac{h_2}{b_3}-\cdots-\frac{h_{k-1}}{b_k}$$

其中 $b_1,b_2,\cdots,b_k$ 满足条件

$$g_k(\bar{x})=b_1+\frac{h-h_1}{b_2}+\frac{h-h_2}{b_3}+\cdots+\frac{h-h_{k-1}}{b_k}$$

式中

$$h_k=[y_2(\bar{x})-y_1(\bar{x})]/2^{k-1}$$

$g_k(\bar{x})$ 为当步长为 h_k 时，积分

$$\int_{y_1(\bar{x})}^{y_2(\bar{x})} f(\bar{x},y)\mathrm{d}y$$

用复化梯形公式计算得到的近似值。

这个过程一直做到满足条件

$$|S_k(\bar{x})-S_{k-1}(\bar{x})|<\varepsilon$$

为止。此时有 $S(\bar{x})\approx S_k(\bar{x})$。

2. 利用 1. 中所计算得到的一系列 $S(\bar{x})$ 值，再利用连分式法计算二重积分 S 的近似值。即

$$S_k=b_1'-\frac{h_1'}{b_2'}-\frac{h_2'}{b_3'}-\cdots-\frac{h_{k-1}'}{b_k'}$$

其中 $b_1',b_2',\cdots,b_k'$ 满足条件

$$g_k'=b_1'+\frac{h-h_1'}{b_2'}+\frac{h-h_2'}{b_3'}+\cdots+\frac{h-h_{k-1}'}{b_k'}$$

式中

$$h'_k = (b-a)/2^{k-1}$$

g'_i 为当步长为 h'_k 时，积分

$$\int_a^b S(x)\mathrm{d}x$$

用复化梯形公式计算得到的近似值。

这个过程也一直做到满足条件

$$|S_k - S_{k-1}| < \varepsilon$$

为止。此时有 $S \approx S_k$。

关于用连分式法计算单积分的详细叙述见 6.5 节的方法说明。

三、子程序语句

SUBROUTINE FPQG2 (A,B,FS,F,EPS,S)

四、形参说明

A,B——均为双精度实型变量，输入参数。外层积分的下限与上限。

FS——子程序名，输入参数。用于计算内层积分的下限 $y_1(x)$ 与上限 $y_2(x)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE FS (X,Y1,Y2)

其中 X,Y1,Y2 均为双精度实型变量，分别为自变量值、内层积分的下限值与上限值。

F——双精度实型函数子程序名，输入参数，用于计算被积函数值 $f(x,y)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F(X,Y)

其中：X 与 Y 均为双精度实型变量，自变量值；函数名 F 返回双精度实型函数值。

EPS——实型变量，输入参数。控制精度要求。

S——双精度实型变量，输出参数。返回积分值。

五、子程序（文件名：FPQG2.FOR）

```
      SUBROUTINE FPQG2(A,B,FS,F,EPS,S)
      EXTERNAL FS,F
      DIMENSION BB(10),H(10)
      DOUBLE PRECISION A,B,F,S,BB,H,S1,S2,TS1,TS2,S0,X,HH,G
      M=1
      N=1
      H(1)=B-A
      HH=B-A
      CALL FQG1(A,FS,F,EPS,S1)
      CALL FQG1(B,FS,F,EPS,S2)
      TS1=HH*(S1+S2)/2.0
      S0=TS1
      BB(1)=TS1
10    TS2=0.5*TS1
```

```
      DO 20 K=0,N-1
        X=A+(K+0.5)*HH
        CALL PQG1(X,FS,F,EPS,S1)
        TS2=TS2+0.5*S1*HH
20    CONTINUE
      M=M+1
      H(M)=H(M-1)/2.0
      G=TS2
      DO 30 J=2,M
        S=G-BB(J-1)
        IF (ABS(S)+1.0.EQ.1.0) THEN
          G=SIGN(1.0D+35,S)
          G=G*SIGN(1.0D0,H(M)-H(J-1))
        ELSE
          G=(H(M)-H(J-1))/S
        END IF
30    CONTINUE
      BB(M)=G
      S=BB(M)
      DO 40 J=M,2,-1
        IF (ABS(S)+1.0.EQ.1.0) THEN
          S=SIGN(1.0D+35,S)
          S=S*SIGN(1.0D0,H(J-1))
          S=BB(J-1)-S
        ELSE
          S=BB(J-1)-H(J-1)/S
        END IF
40    CONTINUE
      IF ((ABS(S-S0).GE.EPS*(ABS(S)+1.0)).AND.(M.LE.9)) THEN
        N=N+N
        TS1=TS2
        S0=S
        HH=HH/2.0
        GOTO 10
      END IF
      RETURN
      END

      SUBROUTINE PQG1(X,FS,F,EPS,S)
      DIMENSION B(10),H(10)
      DOUBLE PRECISION X,F,S,B,H,Y1,Y2,HH,TS1,TS2,S0,Y,G
      M=1
```

```
      N=1
      CALL FS(X,Y1,Y2)
      H(1)=Y2-Y1
      HH=Y2-Y1
      TS1=0.5*HH*(F(X,Y1)+F(X,Y2))
      S0=TS1
      B(1)=TS1
10    TS2=0.5*TS1
      DO 20 K=0,N-1
        Y=Y1+(K+0.5)*HH
        TS2=TS2+0.5*HH*F(X,Y)
20    CONTINUE
      M=M+1
      H(M)=H(M-1)/2.0
      G=TS2
      DO 30 J=2,M
        S=G-B(J-1)
        IF (ABS(S)+1.0.EQ.1.0) THEN
          S=SIGN(1.0D+35,S)
          G=S*SIGN(1.0D0,H(M)-H(J-1))
        ELSE
          G=(H(M)-H(J-1))/S
        END IF
30    CONTINUE
      B(M)=G
      S=B(M)
      DO 40 J=M,2,-1
        IF (ABS(S)+1.0.EQ.1.0) THEN
          S=SIGN(1.0D+35,S)
          S=S*SIGN(1.0D0,H(J-1))
          S=B(J-1)-S
        ELSE
          S=B(J-1)-H(J-1)/S
        END IF
40    CONTINUE
      IF ((ABS(S-S0).GE.EPS*(ABS(S)+1.0)).AND.(M.LE.9)) THEN
        N=N+N
        TS1=TS2
        S0=S
        HH=0.5*HH
        GOTO 10
      END IF
```

```
      RETURN
      END
```

六、例

计算二重积分

$$S=\int_0^1 dx\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} e^{x^2+y^2} dy$$

$\epsilon=0.00005$。

主程序及计算内层积分上下限值、被积函数值的子程序(文件名:FPQG20.FOR)为

```
      EXTERNAL FS,F
      DOUBLE PRECISION F,A,B,S
      A=0.0
      B=1.0
      EPS=0.00005
      CALL FPQG2(A,B,FS,F,EPS,S)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D13.6)
      END

      SUBROUTINE FS(X,Y1,Y2)
      DOUBLE PRECISION X,Y1,Y2,Q
      Q=SQRT(1.0-X*X)
      Y1=-Q
      Y2=Q
      RETURN
      END

      FUNCTION F(X,Y)
      DOUBLE PRECISION F,X,Y
      F=EXP(X*X+Y*Y)
      RETURN
      END
```

运行结果为

```
S=.269911D+01
```

6.15 计算多重积分的蒙特卡洛法

一、功能

用蒙特卡洛(Monte Carlo)法计算多重积分

$$S=\int_{a_1}^{b_1}\int_{a_2}^{b_2}\cdots\int_{a_n}^{b_n} f(x_1, x_2, \cdots, x_n)\, dx_1 dx_2 \cdots dx_n$$

二、方法说明

取0到1之间均匀分布的随机数点列

$$(t_1^{(i)}, t_2^{(i)}, \cdots, t_n^{(i)}),\ i = 1, 2, \cdots, m$$

并令

$$x_j^{(i)} = a_j + (b_j - a_j)t_j^{(i)},\ j = 1, 2, \cdots, n$$

只要 m 足够大，则有

$$S \approx \frac{1}{m}\left[\sum_{i=1}^{m} f(x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)})\right] \cdot \prod_{j=1}^{n}(b_j - a_j)$$

在本子程序中，取 $m=10000$。

三、子程序语句

SUBROUTINE FMTML (N,A,B,F,X,S)

四、形参说明

N——整型变量，输入参数。积分的重数。

A——双精度实型一维数组，长度为N，输入参数。积分的下限值 $a_1, a_2, \cdots, a_n$。

B——双精度实型一维数组，长度为N，输入参数。积分的上限值 $b_1, b_2, \cdots, b_n$。

F——双精度实型函数子程序名，输入参数。用于计算被积函数值 $f(x_1, x_2, \cdots, x_n)$。在主程序中必须用外部语句及类型说明语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

DOUBLE PRECISION FUNCTION F (N,X)

其中：N为整型变量，被积函数中自变量个数，也是积分的重数；X为双精度实型一维数组，长度为N，存放自变量值 $x_1, x_2, \cdots, x_n$；函数名F返回双精度实型函数值 $f(x_1, x_2, \cdots, x_n)$。

X——双精度实型一维数组，长度为N。本子程序的工作数组，用于存放N个自变量值 $x_1, x_2, \cdots, x_n$。

S——双精度实型变量，输出参数。返回积分值。

五、子程序(文件名：FMTML.FOR)

```
      SUBROUTINE FMTML(A,B,N,F,X,S)
      DOUBLE PRECISION A(N),B(N),F,S,R,X(N),K
      REAL M,NRND1
      R=1.0D0
      M=10000.0
      K=10000.0D0
      S=0.0D0
10    IF (M+1.0.NE.1.0) THEN
        M=M-1.0
        DO 20 I=1,N
          X(I)=A(I)+(B(I)-A(I))*NRND1(R)
20      CONTINUE
        S=S+F(N,X)/K
```

```
      GOTO 10
      END IF
      DO 30 I=1,N
30    S=S*(B(I)-A(I))
      END
```

六、例

用蒙特卡洛法计算三重积分

$$S=\int_1^2\int_1^2\int_1^2(x_1^2+x_2^2+x_3^2)\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x_3$$

主程序及计算被积函数值的子程序(文件名:FMTML0.FOR)为

```
      EXTERNAL F
      DIMENSION X(3),A(3),B(3)
      DOUBLE PRECISION X,F,S,A,B
      DATA A,B/3*1.0D0,3*2.0D0/
      N=3
      CALL FMTML(A,B,N,F,X,S)
      WRITE(*,*)
      WRITE(*,10) S
10    FORMAT(1X,'S=',D13.6)
      WRITE(*,*)
      END

      FUNCTION F(N,X)
      DIMENSION X(N)
      DOUBLE PRECISION X,F
      F=0.0D0
      DO 10 I=1,N
10    F=F+X(I)*X(I)
      END
```

运行结果为

S=.697043D+01

七、附注

本子程序需要调用产生 0 到 1 之间均匀分布的一个随机数子程序 NRND1,参看 13.1 节。

第7章 常微分方程组的求解

7.1 全区间积分的定步长欧拉方法

一、功能

用改进的欧拉(Euler)公式，对一阶微分方程组用定步长进行全区间积分。

二、方法说明

设一阶微分方程组

$$y_i'=f_i(t,y_1,y_2,\cdots,y_m),\ y_i(t_0)=y_{i0}$$
$$i=1,2,\cdots,m$$

已知 $t=t_{j-1}$ 点处的函数值 $y_{i,j-1}(i=1,2,\cdots,m)$，求 $t_j=t_{j-1}+h$ 点处的函数值 $y_{ij}(i=1,2,\cdots,m)$。

改进的欧拉公式为

$$p_i=y_{i,j-1}+hf_i(t_{j-1},y_{1,j-1},\cdots,y_{m,j-1}),\ i=1,2,\cdots,m$$
$$q_i=y_{i,j-1}+hf_i(t_j,p_1,p_2,\cdots,p_m),i=1,2,\cdots,m$$
$$y_{ij}=\frac{1}{2}(p_i+q_i),\ i=1,2,\cdots,m$$

三、子程序语句

SUBROUTINE GELR1(T,Y,M,H,N,Z,F,D)

四、形参说明

T——双精度实型变量，输入参数。微分方程组的初值点，即积分的起始点。

Y——双精度实型一维数组，长度为M，输入参数。微分方程组在初始点T处的M个未知函数的初值。返回时其值在Z(i, 1) (i=1, 2, …, M)中。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

H——双精度实型变量，输入参数。从初值点开始，对微分方程组积分一步的步长。

N——整型变量，输入参数，对微分方程组积分的步数(包括初值点这一步)。

Z——双精度实型二维数组，体积为M×N，输出参数。存放N个积分点(包括初值点)上的未知函数值，即

$$Z(i,j)=y_{ij},\ i=1,2,\cdots,M;\ j=1,2,\cdots,N$$

其中Z(i, 1) (i=1, 2, …, M)为初值。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值 $f_i(t,y_1,y_2,\cdots,y_m)$ (i=1, 2, …, M)。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F (T,Y,M,D)

其中，T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未知函数的函数值 $y_i(T)$；D 为双精度实型一维数组，长度为 M，返回 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

D——双精度实型一维数组，长度为 M. 本子程序的工作数组，用于存放 M 个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

五、子程序(文件名:GELR1.FOR)

```
      SUBROUTINE GELR1(T,Y,M,H,N,Z,F,D)
      DIMENSION Y(M),Z(M,N),D(M)
      DOUBLE PRECISION Y,Z,D,T,H,X
      DO 10 I=1,M
10    Z(I,1)=Y(I)
      DO 50 J=2,N
        X=T+(J-2)*H
        CALL F(X,Y,M,D)
        DO 20 I=1,M
20      Y(I)=Z(I,J-1)+H*D(I)
        X=T+(J-1)*H
        CALL F(X,Y,M,D)
        DO 30 I=1,M
30      D(I)=Z(I,J-1)+H*D(I)
        DO 40 I=1,M
          Y(I)=(Y(I)+D(I))/2.0
          Z(I,J)=Y(I)
40      CONTINUE
50    CONTINUE
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = -1.0 \\ y_2' = -y_1, & y_2(0) = 0.0 \\ y_3' = -y_3, & y_3(0) = 1.0 \end{cases}$$

用改进欧拉公式计算当 $h=0.01$ 时，11 个积分点(包括初值点)

$$t_i = (i-1)h, \ i = 1, 2, \cdots, 11$$

上的未知函数的近似值 $y_{1i}, y_{2i}, y_{3i}(i=1, 2, \cdots, 11)$。

其中初值点 T=0.0，M=3，N=11，H=0.01。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GELR10.FOR)为

```
      EXTERNAL F
      DIMENSION Y(3),Z(3,11),D(3)
      DOUBLE PRECISION Y,Z,D,T,H,X
```

```
      T=0.0
      Y(1)=-1.0
      Y(2)=0.0
      Y(3)=1.0
      H=0.01
      M=3
      N=11
      CALL GELR1(T,Y,M,H,N,Z,F,D)
      WRITE(*,*)
      DO 10 I=1,N
        X=(I-1)*H
        WRITE(*,20) X
        WRITE(*,30) (Z(J,I),J=1,M)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F5.2)
30    FORMAT(1X,'Y(1)=',D13.6,3X,'Y(2)=',D13.6,
     *   3X,'Y(3)=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=-Y(3)
      RETURN
      END
```

运行结果为

```
T=  .00
Y(1)= -.100000D+01   Y(2)=  .000000D+00   Y(3)=  .100000D+01

T=  .01
Y(1)= -.999950D+00   Y(2)=  .100000D-01   Y(3)=  .990050D+00

T=  .02
Y(1)= -.999800D+00   Y(2)=  .199990D-01   Y(3)=  .980199D+00

T=  .03
Y(1)= -.999550D+00   Y(2)=  .299960D-01   Y(3)=  .970446D+00

T=  .04
```

Y(1)= -.999200D+00   Y(2)= .399900D-01   Y(3)= .960790D+00

T= .05
Y(1)= -.998750D+00   Y(2)= .499800D-01   Y(3)= .951230D+00

T= .06
Y(1)= -.998200D+00   Y(2)= .599650D-01   Y(3)= .941765D+00

T= .07
Y(1)= -.997551D+00   Y(2)= .699440D-01   Y(3)= .932395D+00

T= .08
Y(1)= -.996802D+00   Y(2)= .799160D-01   Y(3)= .923118D+00

T= .09
Y(1)= -.995953D+00   Y(2)= .898800D-01   Y(3)= .913933D+00

T= .10
Y(1)= -.995004D+00   Y(2)= .998351D-01   Y(3)= .904839D+00

本问题的精确解为

$$\begin{cases} y_1 = -\cos t \\ y_2 = \sin t \\ y_3 = e^{-t} \end{cases}$$

7.2 积分一步的变步长欧拉方法

一、功能

用变步长欧拉(Euler)方法对一阶微分方程组积分一步。

二、方法说明

基本方法同 7.1 节的方法说明。

根据改进的欧拉公式，以 h 为步长由 $y_{i,j-1}$ 计算 $y_{ij}^{(h)}$ $(i=1, 2, \cdots, m)$；再以 $h/2$ 为步长由 $y_{i,j-1}$ 跨两步计算 $y_{ij}^{(h/2)}$ $(i=1, 2, \cdots, m)$。若

$$\max_{1\leqslant i\leqslant m} |y_{ij}^{(h/2)} - y_{ij}^{(h)}| < \varepsilon$$

则停止计算，取 $y_{ij}=y_{ij}^{(h/2)}$ $(i=1, 2, \cdots, m)$；否则将步长折半再进行计算。

上述过程一直做到

$$\max_{1\leqslant i\leqslant m} |y_{ij}^{(h/2^k)} - y_{ij}^{(h/2^{k-1})}| < \varepsilon$$

为止，最后可取

$$y_{ij} = y_{ij}^{(h/2^k)}, \ i = 1, 2, \cdots, m$$

三、子程序语句

SUBROUTINE GELR2 (T,H,Y,M,F,D,EPS,A,B,C)

四、形参说明

T——双精度实型变量,输入参数。自变量的起始点值。

H——双精度实型变量,输入参数。积分步长。

Y——双精度实型一维数组,长度为M,输入兼输出参数。调用时存放起始点T处的未知函数值 $y_i(T)$ ($i=1, 2, \cdots, M$);返回时存放下一点的未知函数值 $y_i(T+H)$ ($i=1, 2, \cdots, M$)。

M——整型变量,输入参数。方程组中方程的个数,也是未知函数的个数。

F——子程序名,输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ ($i=1, 2, \cdots, M$)。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE F(T,Y,M,D)

其中,T为双精度实型变量,自变量值;Y为双精度实型一维数组,长度为M,存放M个未知函数值 $y_i(T)$ ($i=1, 2, \cdots, M$);D为双精度实型一维数组,长度为M,返回M个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ ($i=1, 2, \cdots, M$)。

D——双精度实型一维数组,长度为M。本子程序的工作数组,用于存放M个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ ($i=1, 2, \cdots, M$)。

EPS——实型变量,输入参数。控制精度要求。

A,B,C——均为双精度实型一维数组,长度为M。本子程序的工作数组。

五、子程序(文件名:GELR2.FOR)

```
      SUBROUTINE GELR2(T,H,Y,M,F,D,EPS,A,B,C)
      DIMENSION Y(M),D(M),A(M),B(M),C(M)
      DOUBLE PRECISION T,H,Y,D,HH,X,A,B,C
      HH=H
      N=1
      P=1.0+EPS
      DO 10 I=1,M
10    A(I)=Y(I)
20    IF (P.GE.EPS) THEN
        DO 30 I=1,M
          B(I)=Y(I)
          Y(I)=A(I)
30      CONTINUE
        DO 70 J=1,N
          DO 35 I=1,M
35        C(I)=Y(I)
          X=T+(J-1)*HH
          CALL F(X,Y,M,D)
```

```
      DO 40 I=1,M
40      Y(I)=C(I)+HH*D(I)
      X=T+J*HH
      CALL F(X,Y,M,D)
      DO 50 I=1,M
50      D(I)=C(I)+HH*D(I)
      DO 60 I=1,M
60      Y(I)=(Y(I)+D(I))/2.0
70    CONTINUE
      P=0.0
      DO 80 I=1,M
        Q=ABS(Y(I)-B(I))
        IF (Q.GT.P) P=Q
80    CONTINUE
      HH=HH/2.0
      N=N+N
      GOTO 20
    END IF
    RETURN
    END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = -1.0 \\ y_2' = -y_1, & y_2(0) = 0.0 \\ y_3' = -y_3, & y_3(0) = 1.0 \end{cases}$$

用积分一步的变步长欧拉方法计算当 $h=0.01$ 时各积分点

$$t_i = (i-1)h, \quad i = 1, 2, \cdots, 11$$

上未知函数的近似值 y_{1i}, y_{2i}, y_{3i} $(i=1, 2, \cdots, 11)$。取 $\varepsilon=0.00001$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GELR20.FOR)为

```
    EXTERNAL F
    DIMENSION Y(3),D(3),A(3),B(3),C(3)
    DOUBLE PRECISION Y,D,T,H,A,B,C
    T=0.0
    Y(1)=-1.0
    Y(2)=0.0
    Y(3)=1.0
    H=0.01
    M=3
    EPS=0.00001
    WRITE(*,*)
```

```
      WRITE(*,30) T,Y(1),Y(2),Y(3)
      WRITE(*,*)
      DO 10 I=1,10
        CALL GELR2(T,H,Y,M,F,D,EPS,A,B,C)
        T=T+H
        WRITE(*,30) T,Y(1),Y(2),Y(3)
        WRITE(*,*)
10    CONTINUE
30    FORMAT(1X,'T=',F4.2,3X,'Y(1)=',D13.6,3X,'Y(2)=',D13.6,
     *       3X,'Y(3)=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=-Y(3)
      RETURN
      END
```

运行结果为

```
T=.00   Y(1)= -.100000D+01   Y(2)=  .000000D+00   Y(3)=  .100000D+01

T=.01   Y(1)= -.999950D+00   Y(2)=  .999987D-02   Y(3)=  .990050D+00

T=.02   Y(1)= -.999800D+00   Y(2)=  .199987D-01   Y(3)=  .980199D+00

T=.03   Y(1)= -.999550D+00   Y(2)=  .299956D-01   Y(3)=  .970446D+00

T=.04   Y(1)= -.999200D+00   Y(2)=  .399895D-01   Y(3)=  .960790D+00

T=.05   Y(1)= -.998750D+00   Y(2)=  .499794D-01   Y(3)=  .951230D+00

T=.06   Y(1)= -.998201D+00   Y(2)=  .599643D-01   Y(3)=  .941765D+00

T=.07   Y(1)= -.997551D+00   Y(2)=  .699431D-01   Y(3)=  .932394D+00

T=.08   Y(1)= -.996802D+00   Y(2)=  .799150D-01   Y(3)=  .923117D+00

T=.09   Y(1)= -.995953D+00   Y(2)=  .898769D-01   Y(3)=  .913932D+00

T=.10   Y(1)= -.995004D+00   Y(2)=  .998338D-01   Y(3)=  .904838D+00
```

本问题的精确解为

$$\begin{cases} y_1 = -\cos t \\ y_2 = \sin t \\ y_3 = e^{-t} \end{cases}$$

7.3 定步长维梯方法

一、功能

用定步长维梯(Witty)方法对一阶微分方程组进行全区间积分。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), \quad y_i(t_0) = y_{i0}$$

$$i = 1, 2, \cdots, m$$

用维梯方法由 t_j 积分一步到 $t_{j+1} = t_j + h$ 的计算公式如下：

$$y_{i, j+\frac{1}{2}} = y_{ij} + \frac{1}{2} h y_{ij}', \quad i = 1, 2, \cdots, m$$

$$y_{i, j+\frac{1}{2}}' = f_i\left(t_j + \frac{h}{2}, y_{1, j+\frac{1}{2}}, \cdots, y_{m, j+\frac{1}{2}}\right), \quad i = 1, 2, \cdots, m$$

$$y_{i, j+1} = y_{ij} + h y_{i, j+\frac{1}{2}}', \quad i = 1, 2, \cdots, m$$

$$y_{i, j+1}' = 2 y_{i, j+\frac{1}{2}}' - y_{ij}', \quad i = 1, 2, \cdots, m$$

三、子程序语句

SUBROUTINE GWITY (T,Y,M,H,N,Z,F,D,A)

四、形参说明

T——双精度实型变量，输入参数。积分的起始点。

Y——双精度实型一维数组，长度为 M，输入参数。未知函数在初始点 T 处的 M 个初值 $y_i(T)$ $(i=1, 2, \cdots, M)$。返回时其值在 $Z(i, 1)$ $(i=1, 2, \cdots, M)$中。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

H——双精度实型变量，输入参数。积分步长。

N——整型变量，输入参数。从起始点开始积分的步数(包括起始点这一步)。

Z——双精度实型二维数组，体积为 M×N，输出参数。存放 N 个积分点(包括起始点 T)上的未知函数值，即

$$Z(i, j) = y_{ij}, \quad i = 1, 2, \cdots, M; \ j = 1, 2, \cdots, N$$

其中 $Z(i, 1)$ $(i=1, 2, \cdots, M)$为初值。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未

知函数值 $y_i(T)$ ($i=1, 2, \cdots, M$);D 为双精度实型一维数组，长度为 M，返回 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ ($i=1, 2, \cdots, M$)。

D——双精度实型一维数组，长度为 M. 本子程序的工作数组，用于存放 M 个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$。

A——双精度实型一维数组，长度为 M. 本子程序的工作数组。

五、子程序(文件名:GWITY.FOR)

```
      SUBROUTINE GWITY(T,Y,M,H,N,Z,F,D,A)
      DIMENSION Y(M),D(M),Z(M,N),A(M)
      DOUBLE PRECISION Y,D,Z,T,H,X,A
      DO 10 I=1,M
10    Z(I,1)=Y(I)
      CALL F(T,Y,M,D)
      DO 40 J=2,N
        DO 20 I=1,M
20      A(I)=Z(I,J-1)+H*D(I)/2.0
        X=T+(J-1.5)*H
        CALL F(X,A,M,Y)
        DO 30 I=1,M
          D(I)=2.0*Y(I)-D(I)
          Z(I,J)=Z(I,J-1)+H*Y(I)
30    CONTINUE
40    CONTINUE
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = -1.0 \\ y_2' = -y_1, & y_2(0) = 0.0 \\ y_3' = -y_3, & y_3(0) = 1.0 \end{cases}$$

用维梯方法计算步长 $h=0.1$ 时，11 个积分点

$$t_i = (i-1)h, \quad i = 1, 2, \cdots, 11$$

上的未知函数的近似值 y_{1i}, y_{2i}, y_{3i} ($i=1, 2, \cdots, 11$)。

其中 T=0.0, M=3, N=11, H=0.1.

主程序及计算方程组中 M 个方程右端函数值的子程序(文件名:GWITY0.FOR)为

```
      EXTERNAL F
      DIMENSION Y(3),Z(3,11),D(3),A(3)
      DOUBLE PRECISION Y,Z,D,T,H,X,A
      T=0.0
      Y(1)=-1.0
      Y(2)=0.0
```

```
      Y(3)=1.0
      H=0.1
      M=3
      N=11
      CALL GWITY(T,Y,M,H,N,Z,F,D,A)
      WRITE(*,*)
      DO 10 I=1,N
        X=(I-1)*H
        WRITE(*,20) X
        WRITE(*,30) (Z(J,I),J=1,M)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F5.2)
30    FORMAT(1X,'Y(1)=',D13.6,3X,'Y(2)=',D13.6,
     *       3X,'Y(3)=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=-Y(3)
      RETURN
      END
```

运行结果为

```
T=   .00
Y(1)= -.100000D+01   Y(2)=   .000000D+00   Y(3)=   .100000D+01

T=   .10
Y(1)= -.995000D+00   Y(2)=   .100000D+00   Y(3)=   .905000D+00

T=   .20
Y(1)= -.980000D+00   Y(2)=   .199000D+00   Y(3)=   .819000D+00

T=   .30
Y(1)= -.955200D+00   Y(2)=   .296000D+00   Y(3)=   .741200D+00

T=   .40
Y(1)= -.920800D+00   Y(2)=   .390040D+00   Y(3)=   .670760D+00

T=   .50
Y(1)= -.877192D+00   Y(2)=   .480160D+00   Y(3)=   .607048D+00

T=   .60
Y(1)= -.824768D+00   Y(2)=   .565476D+00   Y(3)=   .549350D+00
```

```
T=   .70
Y(1)= -.764096D+00   Y(2)=   .645114D+00   Y(3)=   .497178D+00
T=   .80
Y(1)= -.695745D+00   Y(2)=   .718298D+00   Y(3)=   .449915D+00
T=   .90
Y(1)= -.620437D+00   Y(2)=   .784263D+00   Y(3)=   .407195D+00
T=   1.00
Y(1)= -.538893D+00   Y(2)=   .842385D+00   Y(3)=   .368476D+00
```

本问题的精确解为

$$\begin{cases} y_1 = -\cos t \\ y_2 = \sin t \\ y_3 = e^{-t} \end{cases}$$

7.4 全区间积分的定步长龙格-库塔法

一、功能

用定步长四阶龙格-库塔(Runge-Kutta)法对一阶微分方程组进行全区间积分。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), \quad y_i(t_0) = y_{i0}$$
$$i = 1, 2, \cdots, m$$

由 t_j 积分一步到 $t_{j+1} = t_j + h$ 的四阶龙格-库塔方法的计算公式为:

$$\begin{cases} K_{1i} = f_i(t_j, y_{1j}, y_{2j}, \cdots, y_{mj}), \ i = 1, 2, \cdots, m \\ K_{2i} = f_i\left(t_j + \dfrac{h}{2}, y_{1j} + \dfrac{h}{2}K_{11}, \cdots, y_{mj} + \dfrac{h}{2}K_{1m}\right), \ i = 1, 2, \cdots, m \\ K_{3i} = f_i\left(t_j + \dfrac{h}{2}, y_{1j} + \dfrac{h}{2}K_{21}, \cdots, y_{mj} + \dfrac{h}{2}K_{2m}\right), \ i = 1, 2, \cdots, m \\ K_{4i} = f_i(t_j + h, y_{1j} + hK_{31}, \cdots, y_{mj} + hK_{3m}), \ i = 1, 2, \cdots, m \\ y_{i,j+1} = y_{ij} + \dfrac{h}{6}(K_{1i} + 2K_{2i} + 2K_{3i} + K_{4i}), \ i = 1, 2, \cdots, m \end{cases}$$

三、子程序语句

SUBROUTINE GRKT1 (T,Y,M,H,N,Z,F,D,B)

四、形参说明

T——双精度实型变量,输入参数。积分起始点。

Y——双精度实型一维数组,长度为M,输入参数。M个未知函数在起始点T处的初值 y_i(T) (i=1, 2, …, M)。返回时其值在 Z(i, 1) (i=1, 2, …, M)中。

M——整型变量,输入参数。方程组中方程的个数,也是未知函数的个数。

H——双精度实型变量,输入参数。积分步长。

N——整型变量,输入参数。积分的步数(包括起始点这一步)。

Z——双精度实型二维数组,体积为M×N,输出参数。返回M个未知函数在N个积分点上的函数值,即

$$Z(i, j) = y_{ij}; \ i = 1, 2, \cdots, M; \ j = 1, 2, \cdots, N$$

其中Z(i, 1) (i=1, 2, …, M)为初值。

F——子程序名,输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE F(T,Y,M,D)

其中:T为双精度实型变量,自变量值;Y为双精度实型一维数组,长度为M,存放M个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$;D为双精度实型一维数组,长度为M,返回M个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

D——双精度实型一维数组,长度为M。本子程序的工作数组,用于存放M个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

B——双精度实型一维数组,长度为M。本子程序的工作数组。

五、子程序(文件名:GRKT1.FOR)

```
      SUBROUTINE GRKT1(T,Y,M,H,N,Z,F,D,B)
      DIMENSION Y(M),D(M),Z(M,N),A(4),B(M)
      DOUBLE PRECISION Y,D,Z,A,B,T,H,X,TT
      A(1)=H/2.0
      A(2)=A(1)
      A(3)=H
      A(4)=H
      DO 5 I=1,M
5     Z(I,1)=Y(I)
      X=T
      DO 100 J=2,N
        CALL F(T,Y,M,D)
        DO 10 I=1,M
10      B(I)=Y(I)
        DO 30 K=1,3
          DO 20 I=1,M
            Y(I)=Z(I,J-1)+A(K)*D(I)
            B(I)=B(I)+A(K+1)*D(I)/3.0
20        CONTINUE
          TT=T+A(K)
          CALL F(TT,Y,M,D)
30      CONTINUE
        DO 40 I=1,M
40      Y(I)=B(I)+H*D(I)/6.0
```

```
      DO 50 I=1,M
50      Z(I,J)=Y(I)
      T=T+H
100   CONTINUE
      T=X
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = -1.0 \\ y_2' = -y_1, & y_2(0) = 0.0 \\ y_3' = -y_3, & y_3(0) = 1.0 \end{cases}$$

用四阶龙格-库塔公式计算步长 $h=0.01$ 时，11 个积分点

$$t_i = (i-1)h,\ i = 1,\ 2,\ \cdots,\ 11$$

上的未知函数值 $y_{1i}, y_{2i}, y_{3i}(i=1,2,\cdots,11)$。

其中 T=0.0，M=3，N=11，H=0.01。

主程序及计算方程组中 M 个方程右端函数值的子程序(文件名:GRKT10.FOR)为

```
      EXTERNAL F
      DIMENSION Y(3),D(3),Z(3,11),B(3)
      DOUBLE PRECISION Y,D,Z,T,H,B
      T=0.0
      Y(1)=-1.0
      Y(2)=0.0
      Y(3)=1.0
      H=0.01
      M=3
      N=11
      CALL GRKT1(T,Y,M,H,N,Z,F,D,B)
      WRITE(*,*)
      DO 10 I=1,N
        T=(I-1)*H
        WRITE(*,50) T
        WRITE(*,100) (Z(J,I),J=1,M)
        WRITE(*,*)
10    CONTINUE
50    FORMAT(1X,'T=',F7.3)
100   FORMAT(1X,'Y(1)=',D13.6,3X,'Y(2)=',D13.6,3X,'Y(3)=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
```

```
      DIMENSION Y(m),D(m)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=-Y(3)
      RETURN
      END
```

运行结果为

```
T=   .000
Y(1)= -.100000D+01   Y(2)=   .000000D+00   Y(3)=   .100000D+01

T=   .010
Y(1)= -.999950D+00   Y(2)=   .999983D-02   Y(3)=   .990050D+00

T=   .020
Y(1)= -.999800D+00   Y(2)=   .199987D-01   Y(3)=   .980199D+00

T=   .030
Y(1)= -.999550D+00   Y(2)=   .299955D-01   Y(3)=   .970446D+00

T=   .040
Y(1)= -.999200D+00   Y(2)=   .399893D-01   Y(3)=   .960789D+00

T=   .050
Y(1)= -.998750D+00   Y(2)=   .499792D-01   Y(3)=   .951229D+00

T=   .060
Y(1)= -.998201D+00   Y(2)=   .599640D-01   Y(3)=   .941765D+00

T=   .070
Y(1)= -.997551D+00   Y(2)=   .699428D-01   Y(3)=   .932394D+00

T=   .080
Y(1)= -.996802D+00   Y(2)=   .799147D-01   Y(3)=   .923116D+00

T=   .090
Y(1)= -.995953D+00   Y(2)=   .898785D-01   Y(3)=   .913931D+00

T=   .100
Y(1)= -.995004D+00   Y(2)=   .998334D-01   Y(3)=   .904837D+00
```

本问题的精确解为

$$\begin{cases} y_1 = -\cos t \\ y_2 = \sin t \\ y_3 = e^{-t} \end{cases}$$

7.5 积分一步的变步长龙格-库塔法

一、功能

用变步长四阶龙格-库塔(Runge-Kutta)法对一阶微分方程组积分一步。

二、方法说明

四阶龙格-库塔公式见7.4节的方法说明。

变步长的处理见7.2节的方法说明。

三、子程序语句

SUBROUTINE GRKT2 (T,H,Y,M,F,EPS,D,B,C,G,E)

四、形参说明

T——双精度实型变量,输入参数。积分的起始点。

H——双精度实型变量,输入参数。积分一步的步长。

Y——双精度实型一维数组,长度为M,输入兼输出参数。调用时存放起始点T处的M个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$;返回时存放下一点T+H处的M个未知函数值 $y_i(T+H)$ $(i=1, 2, \cdots, M)$。

M——整型变量,输入参数。方程组中方程的个数,也是未知函数的个数。

F——子程序名,输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_n)$ $(i=1, 2, \cdots, M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE F(T,Y,M,D)

其中:T为双精度实型变量,自变量值;Y为双精度实型一维数组,长度为M,存放M个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$;D为双精度实型一维数组,长度为M,返回方程组中M个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

EPS——实型变量,输入参数。控制精度要求。

D——双精度实型一维数组,长度为M。本子程序的工作数组,用于存放M个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

B,C,G,E——均为双精度实型一维数组,长度为M。本子程序的工作数组。

五、子程序(文件名:GRKT2.FOR)

```
      SUBROUTINE GRKT2(T,H,Y,M,F,EPS,D,B,C,G,E)
      DIMENSION Y(M),D(M),A(4),B(M),C(M),G(M),E(M)
      DOUBLE PRECISION Y,D,A,B,C,G,T,H,X,HH,E
      HH=H
      N=1
      P=1+EPS
      X=T
      DO 5 I=1,M
5     C(I)=Y(I)
10    IF (P.GE.EPS) THEN
```

```
      A(1)=HH/2.0
      A(2)=A(1)
      A(3)=HH
      A(4)=HH
      DO 20 I=1,M
        G(I)=Y(I)
        Y(I)=C(I)
20    CONTINUE
      DT=H/N
      T=X
      DO 100 J=1,N
        CALL F(T,Y,M,D)
        DO 30 I=1,M
        E(I)=Y(I)
30      B(I)=Y(I)
        DO 50 K=1,3
          DO 40 I=1,M
            Y(I)=E(I)+A(K)*D(I)
            B(I)=B(I)+A(K+1)*D(I)/3.0
40        CONTINUE

          TT=T+A(K)
          CALL F(TT,Y,M,D)
50      CONTINUE
        DO 60 I=1,M
60      Y(I)=B(I)+HH*D(I)/6.0
        T=T+DT
100   CONTINUE
      P=0.0
      DO 110 I=1,M
        Q=ABS(Y(I)-G(I))
        IF (Q.GT.P) P=Q
110   CONTINUE
      HH=HH/2.0
      N=N+N
      GOTO 10
     END IF
     T=X
     RETURN
     END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = 0.0 \\ y_2' = -y_1, & y_2(0) = 1.0 \end{cases}$$

用变步长四阶龙格-库塔法计算当 $h=0.1$ 时各积分点

$$t_i = (i-1)h, \quad i = 1, 2, \cdots, 11$$

上的函数值 y_{1i}，$y_{2i}(i=1, 2, \cdots, 11)$。取 $\varepsilon=0.00001$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GRKT20.FOR)为

```
      EXTERNAL F
      DIMENSION Y(2),D(2),B(2),C(2),G(2),E(2)
      DOUBLE PRECISION Y,D,T,H,B,C,G,E
      T=0.0
      Y(1)=0.0
      Y(2)=1.0
      H=0.1
      M=2
      EPS=1.0E-05
      WRITE(*,*)
      WRITE(*,50) T,Y(1),Y(2)
      WRITE(*,*)
      DO 10 I=1,10
        CALL GRKT2(T,H,Y,M,F,EPS,D,B,C,G,E)
        T=T+H
        WRITE(*,50) T,Y(1),Y(2)
        WRITE(*,*)
10    CONTINUE
50    FORMAT(1X,'T=',F7.3,5X,'Y(1)=',D13.6,5X,'Y(2)=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      RETURN
      END
```

运行结果为

```
T=   .000   Y(1)=   .000000D+00   Y(2)=   .100000D+01

T=   .100   Y(1)=   .998334D-01   Y(2)=   .995004D+00

T=   .200   Y(1)=   .198669D+00   Y(2)=   .980067D+00

T=   .300   Y(1)=   .295520D+00   Y(2)=   .955336D+00
```

```
T=  .400   Y(1)=  .389418D+00   Y(2)=  .921061D+00
T=  .500   Y(1)=  .479426D+00   Y(2)=  .877583D+00
T=  .600   Y(1)=  .564642D+00   Y(2)=  .825336D+00
T=  .700   Y(1)=  .644218D+00   Y(2)=  .764842D+00
T=  .800   Y(1)=  .717356D+00   Y(2)=  .696707D+00
T=  .900   Y(1)=  .783327D+00   Y(2)=  .621610D+00
T= 1.000   Y(1)=  .841471D+00   Y(2)=  .540302D+00
```

本问题的精确解为

$$\begin{cases} y_1 = \sin t \\ y_2 = \cos t \end{cases}$$

7.6 积分一步的变步长基尔方法

一、功能

用变步长基尔(Gill)方法对一阶微分方程组积分一步。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m),\ y_i(t_0) = y_{i0}$$
$$i = 1, 2, \cdots, m$$

由 t 点积分到 $t+h$ 点的基尔方法的计算公式如下：

$$\begin{cases} K_{1i} = hf_i(t, y_1^{(0)}, y_2^{(0)}, \cdots, y_m^{(0)}),\ i = 1, 2, \cdots, m \\ K_{2i} = hf_i\left(t + \dfrac{h}{2}, y_1^{(1)}, y_2^{(1)}, \cdots, y_m^{(1)}\right),\ i = 1, 2, \cdots, m \\ K_{3i} = hf_i\left(t + \dfrac{h}{2}, y_1^{(2)}, y_2^{(2)}, \cdots, y_m^{(2)}\right),\ i = 1, 2, \cdots, m \\ K_{4i} = hf_i(t + h, y_1^{(3)}, y_2^{(3)}, \cdots, y_m^{(3)}),\ i = 1, 2, \cdots, m \end{cases}$$

$$\begin{cases} y_i^{(1)} = y_i^{(0)} + \dfrac{1}{2}(K_{1i} - 2q_i^{(0)}),\ i = 1, 2, \cdots, m \\ y_i^{(2)} = y_i^{(1)} + \left[1 - \sqrt{\dfrac{1}{2}}\right](K_{2i} - q_i^{(1)}),\ i = 1, 2, \cdots, m \\ y_i^{(3)} = y_i^{(2)} + \left[1 + \sqrt{\dfrac{1}{2}}\right](K_{3i} - q_i^{(2)}),\ i = 1, 2, \cdots, m \\ y_i^{(4)} = y_i^{(3)} + \dfrac{1}{6}(K_{4i} - 2q_i^{(3)}),\ i = 1, 2, \cdots, m \end{cases}$$

$$
\begin{cases}
q_i^{(1)} = q_i^{(0)} + 3\left[\dfrac{1}{2}(K_{1i} - 2q_i^{(0)})\right] - \dfrac{1}{2}K_{1i},\ i = 1,\ 2,\ \cdots,\ m \\
q_i^{(2)} = q_i^{(1)} + 3\left[\left(1 - \sqrt{\dfrac{1}{2}}\right)(K_{2i} - q_i^{(1)})\right] - \left(1 - \sqrt{\dfrac{1}{2}}\right)K_{2i},\ i = 1,\ 2,\ \cdots,\ m \\
q_i^{(3)} = q_i^{(2)} + 3\left[\left(1 + \sqrt{\dfrac{1}{2}}\right)(K_{3i} - q_i^{(2)})\right] - \left(1 + \sqrt{\dfrac{1}{2}}\right)K_{3i},\ i = 1,\ 2,\ \cdots,\ m \\
q_i^{(4)} = q_i^{(3)} + 3\left[\dfrac{1}{6}(K_{4i} - 2q_i^{(3)})\right] - \dfrac{1}{2}K_{4i},\ i = 1,\ 2,\ \cdots,\ m
\end{cases}
$$

其中，$y_i^{(0)}$ 为 t 点的未知函数值 $y_i(t)$；$y_i^{(4)}$ 为积分一步后的未知函数值 $y_i(t+h)$；$q_i^{(0)}$ 在初始时赋初值为 0，以后每积分一步，将 $q_i^{(4)}$ 作为下一步的 $q_i^{(0)}$。

这种方法具有抵消在每一步中所积累的舍入误差的作用，可以提高精度。

本子程序在积分一步中还具有自动选择步长的作用。即根据给定的精度要求，自动将步长 h 等分，然后跨多步积分到 $t+h$。关于自动选择步长的方法参看 7-2 节的方法说明。

三、子程序语句

SUBROUTINE GGIL1(T,H,Y,M,F,EPS,Q,D,U,V,G)

四、形参说明

T——双精度实型变量，输入参数。自变量的起始点。

H——双精度实型变量，输入参数。积分的步长。

Y——双精度实型一维数组，长度为 M，输入兼输出参数。调用时存放起始点的 M 个未知函数值 $y_i(T)$ $(i=1,\ 2,\ \cdots,\ M)$，返回时存放 T+H 点的 M 个未知函数值 $y_i(T+H)$ $(i=1,\ 2,\ \cdots,\ M)$。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值 $f_i(t,\ y_1,\ y_2,\ \cdots,\ y_m)$ $(i=1,\ 2,\ \cdots,\ M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未知函数值 $y_i(T)$ $(i=1,\ 2,\ \cdots,\ M)$；D 为双精度实型一维数组，长度为 M，返回方程组中 M 个方程的右端函数值 $f_i(T,\ y_1,\ y_2,\ \cdots,\ y_m)$ $(i=1,\ 2,\ \cdots,\ M)$。

EPS——实型变量，输入参数。控制精度要求。

Q——双精度实型一维数组，长度为 M，输入兼输出参数。主程序中第一次调用本子程序时，应该置初值

$$Q(i) = 0,\ i = 1,\ 2,\ \cdots,\ M$$

以后每调用一次本子程序(即每积分一步)，将由本子程序返回值以便循环使用。

D——双精度实型一维数组，长度为 M。本子程序的工作数组，用于存放方程组中各方程右端函数值 $f_i(t,\ y_1,\ y_2,\ \cdots,\ y_m)$ $(i=1,\ 2,\ \cdots,\ M)$。

U,V,G——均为双精度实型一维数组，长度为 M。本子程序的工作数组。

五、子程序(文件名:GGIL1.FOR)

```
      SUBROUTINE GGIL1(T,H,Y,M,F,EPS,Q,D,U,V,G)
      DIMENSION Y(M),D(M),Q(M),A(4),B(4),C(4),E(4)
      DIMENSION U(M),V(M),G(M)
      DOUBLE PRECISION Y,D,Q,A,B,C,E,U,V,G,T,H,X,HH,DT,R,S,T0
      DATA A/0.5,0.29289321881,1.7071067812,0.166666667/
      DATA B/2.0,1.0,1.0,2.0/
      DATA E/0.5,0.5,1.0,1.0/
      DO 10 I=1,3
10    C(I)=A(I)
      C(4)=0.5
      X=T
      P=1+EPS
      NN=1
      HH=H
      DO 40 J=1,M
40    U(J)=Y(J)
50    IF (P.GE.EPS) THEN
        DO 60 J=1,M
          V(J)=Y(J)
          Y(J)=U(J)
          G(J)=Q(J)
60      CONTINUE
        DT=H/NN
        T=X
        DO 180 K=1,NN
          CALL F(T,Y,M,D)
          DO 160 II=1,4
            DO 70 J=1,M
70          D(J)=D(J)*HH
            DO 140 J=1,M
              R=(A(II)*(D(J)-B(II)*G(J))+Y(J))-Y(J)
              Y(J)=Y(J)+R
              S=G(J)+3.0*R
              G(J)=S-C(II)*D(J)
140         CONTINUE
            T0=T+E(II)*HH
            CALL F(T0,Y,M,D)
160       CONTINUE
          T=T+DT
180     CONTINUE
```

```
      P=0.0
      DO 190 J=1,M
        QQ=ABS(Y(J)-V(J))
        IF (QQ.GT.P) P=QQ
190   CONTINUE
      HH=HH/2.0
      NN=NN+NN
      GOTO 50
     ELSE
      DO 195 J=1,M
        Q(J)=G(J)
195   CONTINUE
     END IF
     T=X
     RETURN
     END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = 0.0 \\ y_2' = -y_1, & y_1(0) = 1.0 \\ y_3' = -y_3, & y_3(0) = 1.0 \end{cases}$$

用变步长基尔方法计算当 $h=0.1$ 时,各积分点

$$t_i = (i-1)h, \ i = 1, 2, \cdots, 11$$

上的未知函数值 $y_{1i}, y_{2i}, y_{3i}(i=1, 2, \cdots, 11)$。取 $\varepsilon=0.000001$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GGIL10.FOR)为

```
      EXTERNAL F
      DIMENSION Y(3),D(3),Q(3),U(3),V(3),G(3)
      DOUBLE PRECISION Y,D,Q,T,H,U,V,G
      DATA Y/0.0,1.0,1.0/
      DATA Q/3*0.0/
      T=0.0
      H=0.1
      M=3
      EPS=0.000001
      WRITE(*,*)
      WRITE(*,20) T
      WRITE(*,30) (Y(J),J=1,M)
      WRITE(*,*)
      DO 10 I=1,10
        CALL GGIL1(T,H,Y,M,F,EPS,Q,D,U,V,G)
```

```
          T=I*H
          WRITE(*,20) T
          WRITE(*,30) (Y(J),J=1,M)
          WRITE(*,*)
10      CONTINUE
20      FORMAT(1X,'T=',F7.3)
30      FORMAT(1X,'Y1=',D13.6,5X,'Y2=',D13.6,5X,'Y3=',D13.6)
        END

        SUBROUTINE F(T,Y,M,D)
        DIMENSION Y(M),D(M)
        DOUBLE PRECISION Y,D,T
        D(1)=Y(2)
        D(2)=-Y(1)
        D(3)=-Y(3)
        RETURN
        END
```

运行结果为

```
T=   .000
Y1=  .000000D+00     Y2=  .100000D+01     Y3=  .100000D+01

T=   .100
Y1=  .998334D-01     Y2=  .995004D+00     Y3=  .904837D+00

T=   .200
Y1=  .198669D+00     Y2=  .980067D+00     Y3=  .818731D+00

T=   .300
Y1=  .295520D+00     Y2=  .955336D+00     Y3=  .740818D+00

T=   .400
Y1=  .389418D+00     Y2=  .921061D+00     Y3=  .670320D+00

T=   .500
Y1=  .479426D+00     Y2=  .877583D+00     Y3=  .606531D+00

T=   .600
Y1=  .564642D+00     Y2=  .825336D+00     Y3=  .548812D+00

T=   .700
Y1=  .644218D+00     Y2=  .764842D+00     Y3=  .496585D+00

T=   .800
Y1=  .717356D+00     Y2=  .696707D+00     Y3=  .449329D+00

T=   .900
Y1=  .783327D+00     Y2=  .621610D+00     Y3=  .406570D+00
```

T= 1.000

Y1= .841471D+00 Y2= .540302D+00 Y3= .367879D+00

本问题的精确解为

$$\begin{cases} y_1 = \sin t \\ y_2 = \cos t \\ y_3 = e^{-t} \end{cases}$$

7.7 全区间积分的变步长基尔方法

一、功能

用变步长基尔(Gill)方法对一阶微分方程组进行全区间积分。

二、方法说明

见 7.6 节的方法说明。

三、子程序语句

SUBROUTINE GGIL2 (T,H,N,Y,M,Z,F,EPS,D,Q,U,V,G)

四、形参说明

T——双精度实型变量,输入参数。积分的起始点。

H——双精度实型变量,输入参数。积分的步长。

N——整型变量,输入参数。积分的步数(包括起始点这一步)。

Y——双精度实型一维数组,长度为 M,输入参数。存放起始点 T 处的 M 个未知函数的初值 $y_i(T)$ $(i=1, 2, \cdots, M)$。返回时其值在 $Z(i, 1)$ $(i=1, 2, \cdots, M)$中。

M——整型变量,输入参数。方程组中方程的个数,也是未知函数的个数。

Z——双精度实型二维数组,体积为 M×N,输出参数。存放 N 个积分点上的 M 个未知函数值,即

$$Z(i, j) = y_{ij}, \ i = 1, 2, \cdots, M; \ j = 1, 2, \cdots, N$$

其中 $Z(i, 1)$为初值 $(i=1, 2, \cdots, M)$。

F——子程序名,输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE F(T,Y,M,D)

其中:T 为双精度实型变量,自变量值;Y 为双精度实型一维数组,长度为 M,存放 M 个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$;D 为双精度实型一维数组,长度为 M,返回方程组中各方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

EPS——实型变量,输入参数。控制精度要求。

D——双精度实型一维数组,长度为 M。本子程序的工作数组,用于存放方程组中 M 个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

Q,U,V,G——均为双精度实型一维数组,长度为 M。本子程序的工作数组。

五、子程序(文件名:GGIL2.FOR)

```
      SUBROUTINE GGIL2(T,H,N,Y,M,Z,F,EPS,D,Q,U,V,G)
      DIMENSION Y(M),D(M),Z(M,N),A(4),B(4),C(4),E(4)
      DIMENSION Q(M),U(M),V(M),G(M)
      DOUBLE PRECISION Y,D,Z,A,B,C,E,Q,U,V,G,
     *                 T,H,AA,X,HH,DT,R,S,T0
      DATA A/0.5,0.29289321881,1.7071067812,0.166666667/
      DATA B/2.0,1.0,1.0,2.0/
      DATA E/0.5,0.5,1.0,1.0/
      AA=T
      DO 10 I=1,3
10    C(I)=A(I)
      C(4)=0.5
      DO 20 I=1,M
20    Z(I,1)=Y(I)
      DO 30 I=1,M
30    Q(I)=0.0
      DO 200 I=2,N
        X=AA+(I-2)*H
        P=1+EPS
        NN=1
        HH=H
        DO 40 J=1,M
40      U(J)=Y(J)
50      IF (P.GE.EPS) THEN
          DO 60 J=1,M
            V(J)=Y(J)
            Y(J)=U(J)
            G(J)=Q(J)
60        CONTINUE
          DT=H/NN
          T=X
          DO 180 K=1,NN
            CALL F(T,Y,M,D)
            DO 160 II=1,4
              DO 70 J=1,M
70            D(J)=D(J)*HH
              DO 140 J=1,M
                R=(A(II)*(D(J)-B(II)*G(J))+Y(J))-Y(J)
                Y(J)=Y(J)+R
                S=G(J)+3.0*R
```

```
         G(J)=S-C(II)*D(J)
140      CONTINUE
         T0=T+E(II)*HH
         CALL F(T0,Y,M,D)
160    CONTINUE
       T=T+DT
180   CONTINUE
      P=0.0
      DO 190 J=1,M
        QQ=ABS(Y(J)-V(J))
        IF (QQ.GT.P) P=QQ
190   CONTINUE
      HH=HH/2.0
      NN=NN+NN
      GOTO 50
     ELSE
      DO 195 J=1,M
        Q(J)=G(J)
        Z(J,I)=Y(J)
195   CONTINUE
     END IF
200 CONTINUE
    T=AA
    RETURN
    END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = 0.0 \\ y_2' = -y_1, & y_2(0) = 1.0 \\ y_3' = -y_3, & y_3(0) = 1.0 \end{cases}$$

用变步长基尔方法计算当 $h=0.1$ 时，各积分点

$$t_i = (i-1)h,\ i = 1,\ 2,\ \cdots,\ 11$$

上的未知函数值 y_{1i}，y_{2i}，$y_{3i}$$(i=1,\ 2,\ \cdots,\ 11)$。取 $\varepsilon=0.000001$。

主程序及计算方程组中各方程右端函数值的子程序（文件名：GGIL20.FOR）为

```
EXTERNAL F
DIMENSION Y(3),D(3),Z(3,11),Q(3),U(3),V(3),G(3)
DOUBLE PRECISION Y,D,Z,A,H,T,Q,U,V,G
DATA Y/0.0,1.0,1.0/
A=0.0
```

```
      H=0.1
      N=11
      M=3
      EPS=0.000001
      CALL GGIL2(A,H,N,Y,M,Z,F,EPS,D,Q,U,V,G)
      WRITE(*,*)
      DO 10 I=1,N
        T=(I-1)*H
        WRITE(*,20) T
        WRITE(*,30) (Z(J,I),J=1,M)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F7.3)
30    FORMAT(1X,'Y1=',D13.6,5X,'Y2=',D13.6,5X,'Y3=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=-Y(3)
      RETURN
      END
```

运行结果为

```
T=   .000
Y1=   .000000D+00     Y2=   .100000D+01     Y3=   .100000D+01

T=   .100
Y1=   .998334D-01     Y2=   .995004D+00     Y3=   .904837D+00

T=   .200
Y1=   .198669D+00     Y2=   .980067D+00     Y3=   .818731D+00

T=   .300
Y1=   .295520D+00     Y2=   .955336D+00     Y3=   .740818D+00

T=   .400
Y1=   .389418D+00     Y2=   .921061D+00     Y3=   .670320D+00

T=   .500
Y1=   .479426D+00     Y2=   .877583D+00     Y3=   .606531D+00

T=   .600
Y1=   .564642D+00     Y2=   .825336D+00     Y3=   .548812D+00
```

T=   .700
Y1=   .644218D+00   Y2=   .764842D+00   Y3=   .496585D+00

T=   .800
Y1=   .717356D+00   Y2=   .696707D+00   Y3=   .449329D+00

T=   .900
Y1=   .783327D+00   Y2=   .621610D+00   Y3=   .406570D+00

T=   1.000
Y1=   .841471D+00   Y2=   .540302D+00   Y3=   .367879D+00

本问题的精确解为

$$\begin{cases} y_1 = \sin t \\ y_2 = \cos t \\ y_3 = e^{-t} \end{cases}$$

7.8 全区间积分的变步长默森方法

一、功能

用变步长默森(Merson)方法对一阶微分方程组进行全区间积分。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), \; y_i(t_0) = y_{i0} \quad i = 1, 2, \cdots, m$$

由 t_j 积分一步到 $t_{j+1} = t_j + h$ 的默森方法的计算公式如下：

$$y_i^{(1)} = y_i^{(0)} + \frac{h}{3} f_i(t_j, y_1^{(0)}, y_2^{(0)}, \cdots, y_m^{(0)}), \quad i = 1, 2, \cdots, m$$

$$y_i^{(2)} = y_i^{(1)} + \frac{h}{6}\left[f_i\left(t_j + \frac{h}{3}, y_1^{(1)}, y_2^{(1)}, \cdots, y_m^{(1)}\right) - f_i(t_j, y_1^{(0)}, y_2^{(0)}, \cdots, y_m^{(0)})\right]$$
$$i = 1, 2, \cdots, m$$

$$y_i^{(3)} = y_i^{(2)} + \frac{3}{8}h\left[f_i\left(t_j + \frac{h}{3}, y_1^{(2)}, y_2^{(2)}, \cdots, y_m^{(2)}\right) - \frac{4}{9}\left(f_i\left(t_j + \frac{h}{3}, y_1^{(1)}, y_2^{(1)}, \cdots, y_m^{(1)}\right) + \frac{1}{4} f_i(t_j, y_1^{(0)}, y_2^{(0)}, \cdots, y_m^{(0)})\right)\right]$$
$$i = 1, 2, \cdots, m$$

$$y_i^{(4)} = y_i^{(3)} + 2h\left[f_i\left(t_j + \frac{h}{2}, y_1^{(3)}, y_2^{(3)}, \cdots, y_m^{(3)}\right) - \frac{15}{16}\left(f_i\left(t_j + \frac{h}{3}, y_1^{(2)}, y_2^{(2)}, \cdots, y_m^{(2)}\right) - \frac{1}{5} f_i(t_j, y_1^{(0)}, y_2^{(0)}, \cdots, y_m^{(0)})\right)\right]$$
$$i = 1, 2, \cdots, m$$

$$y_i^{(5)} = y_i^{(4)} + \frac{h}{6}\left[f_i(t_j + h, y_1^{(4)}, y_2^{(4)}, \cdots, y_m^{(4)}) - 8\left(f_i\left(t_j + \frac{h}{2}, y_1^{(3)}, y_2^{(3)}, \cdots, y_m^{(3)}\right) - \frac{9}{8}\left(f_i\left(t_j + \frac{h}{3}, y_1^{(2)}, y_2^{(2)}, \cdots, y_m^{(2)}\right) - \frac{2}{9} f_i(t_j, y_1^{(0)}, y_2^{(0)}, \cdots, y_m^{(0)})\right)\right)\right]$$

$$i=1,2,\cdots,m$$

其中

$$y_i^{(0)}=y_i(t_j),\quad i=1,2,\cdots,m$$

$$y_i^{(5)}=y_i(t_j+h),\quad i=1,2,\cdots,m$$

关于自动选择步长问题参看7.2节的方法说明。

三、子程序语句

SUBROUTINE GMRSN (T,H,N,Y,M,Z,F,EPS,D,U,A,B,C,V)

四、形参说明

T——双精度实型变量，输入参数。积分的起始点值。

H——双精度实型变量，输入参数。积分的步长。

N——整型变量，输入参数。积分的步数(包括起始点这一步)。

Y——双精度实型一维数组，长度为M，输入参数。存放在起始点T处的M个未知函数值y_i(T) ($i=1,2,\cdots$, M)。返回时其值在Z(i, 1) ($i=1,2,\cdots$, M)中。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

Z——双精度实型二维数组，体积为M×N，输出参数。存放N个积分点上的M个未知函数值，即

$$Z(i,j)=y_{ij},\ i=1,2,\cdots,M;\ j=1,2,\cdots,N$$

其中Z(i, 1) ($i=1,2,\cdots$, M)为初值。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值$f_i(t,y_1,y_2,\cdots,y_m)$ ($i=1,2,\cdots$, M)。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T为双精度实型变量，自变量值，Y为双精度实型一维数组，长度为M，存放M个未知函数值y_i(T) ($i=1,2,\cdots$, M)；D为双精度实型一维数组，长度为M，返回方程组中M个方程的右端函数值$f_i(T,y_1,y_2,\cdots,y_m)$ ($i=1,2,\cdots$, M)。

EPS——实型变量，输入参数。控制精度要求。

D——双精度实型一维数组，长度为M。本子程序的工作数组，用于存放方程组中各方程的右端函数值$f_i(t,y_1,y_2,\cdots,y_m)$ ($i=1,2,\cdots$, M)。

U,A,B,C,V——均为双精度实型一维数组，长度为M。本子程序的工作数组。

五、子程序(文件名：GMRSN.FOR)

```
      SUBROUTINE GMRSN(T,H,N,Y,M,Z,F,EPS,D,U,A,B,C,V)
      DIMENSION Y(M),D(M),U(M),A(M),B(M)
      DIMENSION C(M),V(M),Z(M,N)
      DOUBLE PRECISION Y,D,U,A,B,C,V,Z,T,H,AA,X,HH,DT,T0,QQ
      AA=T
      DO 90 I=1,M
90    Z(I,1)=Y(I)
      DO 200 I=2,N
```

```
      X=AA+(I-2)*H
      P=1+EPS
      NN=1
      HH=H
      DO 110 J=1,M
110   U(J)=Y(J)
120   IF (P.GE.EPS) THEN
        DO 130 J=1,M
          V(J)=Y(J)
          Y(J)=U(J)
130     CONTINUE
        DT=H/NN
        T=X
        DO 160 K=1,NN
          CALL F(T,Y,M,D)
          DO 135 J=1,M
            A(J)=D(J)
            Y(J)=Y(J)+HH*D(J)/3.0
135       CONTINUE
          T0=T+HH/3.0
          CALL F(T0,Y,M,D)
          DO 140 J=1,M
            B(J)=D(J)
            Y(J)=Y(J)+HH*(D(J)-A(J))/6.0
140       CONTINUE
          CALL F(T0,Y,M,D)
          DO 145 J=1,M
            B(J)=D(J)
            Y(J)=Y(J)+3*HH*(D(J)-4*(B(J)+A(J)/4.0)/9.0)/8.0
145       CONTINUE
          T0=T+HH/2.0
          CALL F(T0,Y,M,D)
          DO 150 J=1,M
            C(J)=D(J)
            QQ=D(J)-15*(B(J)-A(J)/5.0)/16.0
            Y(J)=Y(J)+2*HH*QQ
150       CONTINUE
          T0=T+HH
          CALL F(T0,Y,M,D)
          DO 155 J=1,M
            QQ=D(J)-8*(C(J)-9*(B(J)-2*A(J)/9.0)/8.0)
            Y(J)=Y(J)+HH*QQ/6.0
```

```
155       CONTINUE
          T=T+DT
160     CONTINUE
        P=0.0
        DO 165 J=1,M
          QQ=ABS(Y(J)-V(J))
          IF (QQ.GT.P) P=QQ
165     CONTINUE
        HH=HH/2.0
        NN=NN+NN
        GOTO 120
      ELSE
        DO 170 J=1,M
170     Z(J,I)=Y(J)
      END IF
200   CONTINUE
      T=AA
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = 60[0.06 + t(t - 0.6)]y_2, & y_1(0) = 0.0 \\ y_2' = -60[0.06 + t(t - 0.6)]y_1, & y_2(0) = 1.0 \end{cases}$$

用默森方法计算当 $h=0.1$ 时，各积分点

$$t_i = (i - 1)h, \quad i = 1, 2, \cdots, 11$$

上的未知函数值 $y_1(t_i)$，$y_2(t_i)$，$(i=1, 2, \cdots, 11)$。取 $\varepsilon=0.00001$。

主程序及计算方程组中各方程右端函数值的子程序（文件名：GMRSN0.FOR）为

```
      EXTERNAL F
      DIMENSION Y(2),D(2),Z(2,11),U(2),AA(2),B(2),C(2),V(2)
      DOUBLE PRECISION Y,D,Z,A,T,H,U,AA,B,C,V
      A=0.0
      H=0.1
      N=11
      M=2
      EPS=0.00001
      Y(1)=0.0
      Y(2)=1.0
      CALL GMRSN(A,H,N,Y,M,Z,F,EPS,D,U,AA,B,C,V)
      WRITE(*,*)
      DO 10 I=1,N
        T=(I-1)*H
```

```
      WRITE(*,20) T,Z(1,I),Z(2,I)
      WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,F7.3,5X,'Y1=',D13.6,5X,'Y2=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T,Q
      Q=60*(0.06+T*(T-0.6))
      D(1)=Q*Y(2)
      D(2)=-Q*Y(1)
      RETURN
      END
```

运行结果为

```
  .000    Y1=   .000000D+00    Y2=   .100000D+01

  .100    Y1=   .198669D+00    Y2=   .980064D+00

  .200    Y1=   .159318D+00    Y2=   .987226D+00

  .300    Y1=  -.231730D-06    Y2=   .100000D+01

  .400    Y1=  -.159319D+00    Y2=   .987226D+00

  .500    Y1=  -.198670D+00    Y2=   .980064D+00

  .600    Y1=  -.685894D-06    Y2=   .100000D+01

  .700    Y1=   .531188D+00    Y2=   .847256D+00

  .800    Y1=   .999577D+00    Y2=  -.292033D-01

  .900    Y1=  -.982552D-01    Y2=  -.995165D+00

 1.000    Y1=  -.631263D+00    Y2=   .775574D+00
```

本问题的精确解为

$$
\begin{cases}
y_1 = \sin[20t(t-0.3)(t-0.6)] \\
y_2 = \cos[20t(t-0.3)(t-0.6)]
\end{cases}
$$

7.9 积分一步的连分式法

一、功能

用连分式法对一阶微分方程组积分一步。

二、方法说明

设一阶微分方程组为

$$y_i'=f_i(t,\ y_1,\ y_2,\ \cdots,\ y_m),\ y_i(t_0)=y_{i0}\quad i=1,\ 2,\ \cdots,\ m$$

现已知 $y_{ij}=y_i(t_j)$，求 $y_{i,j+1}=y_i(t_{j+1})$ $(i=1,\ 2,\ \cdots,\ m)$

用定步长欧拉方法由 t_j 跨 $n=2^{l-1}$ 步到 t_{j+1}，其中每一小步的步长为

$$h_l=(t_{j+1}-t_j)/2^{l-1},\ l=1,\ 2,\ \cdots$$

由此可以计算得到

$$y_{i,j+1}^{(l)}\overset{记作}{=}S_{il},\ i=1,\ 2,\ \cdots,\ m$$

显然，S_{il}是步长 h_l 的函数，即可以表示成

$$S_i=S_i(h)$$

并且，当 h 越小时，计算得到的 $S_i(h)$越接近于真值 $y_{i,j+1}=y_i(t_{j+1})$。

将 $S_i(h)$用连分式表示为

$$S_i(h)=b_{1i}+\cfrac{h-h_1}{b_{2i}+\cfrac{h-h_2}{b_{3i}+\cdots+\cfrac{h-h_{l-1}}{b_{li}+\cdots}}}$$

$$i=1,\ 2,\ \cdots,\ m$$

则有

$$\begin{aligned}y_i(t_{j+1})&=S_i(0)\\&=b_{1i}-\cfrac{h_1}{b_{2i}-\cfrac{h_2}{b_{3i}-\cdots-\cfrac{h_{l-1}}{b_{li}-\cdots}}}\end{aligned}$$

$$i=1,\ 2,\ \cdots,\ m$$

其中 $b_{li}(l=1,\ 2,\ \cdots)$ 可由以下递推公式计算：

$$b_{1i}=S_{i1},\quad i=1,\ 2,\ \cdots,\ m$$

$$\begin{cases}u=S_{il}\\u=(h_l-h_k)/(u-b_{ki}),\quad k=1,\ 2,\ \cdots,\ l-1\\b_{li}=u\end{cases}$$

$$i=1,\ 2,\ \cdots,\ m$$

通常，以上连分式做到七节就能满足精度要求。

最后可以得到积分一步的连分式法的计算步骤如下：

1. 用定步长改进欧拉公式由 t_j 跨一步到 t_{j+1} 计算 $S_{i1}(i=1,\ 2,\ \cdots,\ m)$，其中

$$h_1=t_{j+1}-t_j$$

$$b_{1i}=S_{i1},\quad y_i^{(1)}=S_{i1},\quad i=1,\ 2,\ \cdots,\ m$$

2. 对于$l=2, 3, \cdots$，令

$$h_l = h_{l-1}/2$$

(1) 以h_l为步长，用定步长改进欧拉公式由t_j跨$n=2^{l-1}$步到t_{j+1}计算$S_{li}(i=1, 2, \cdots, m)$。然后用下列递推公式计算$b_{li}(i=1, 2, \cdots, m)$

$$\begin{cases} u = S_{li} \\ u = (h_l - h_k)/(u - b_{ki}),\ k = 1, 2, \cdots, l-1 \\ b_{li} = u \end{cases}$$

(2) 计算

$$y_i^{(l)} = b_{1i} - \frac{h_1}{b_{2i}} - \frac{h_2}{b_{3i}} - \cdots - \frac{h_{l-1}}{b_{li}} \qquad i=1, 2, \cdots, m$$

反复做第2步，直到

$$\max_{1\leqslant i\leqslant m} |y_i^{(l)} - y_i^{(l-1)}| < \varepsilon$$

为止。

关于改进欧拉公式，参看7.1节的方法说明。本子程序中自带定步长改进欧拉公式积分一步的子程序。

三、子程序语句

SUBROUTINE GPBS1 (T,H,Y,M,F,EPS,D,B,U,V,W)

四、形参说明

T——双精度实型变量，输入参数。积分起始点值。

H——双精度实型变量，输入参数。积分的步长。

Y——双精度实型一维数组，长度为M，输入兼输出参数。调用时存放起始点T处的M个未知函数值$y_i(T)$ $(i=1, 2, \cdots, M)$；返回时存放T+H点处的M个未知函数值$y_i(T+H)$ $(i=1, 2, \cdots, M)$。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值$f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$，在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T为双精度实型变量，自变量值；Y为双精度实型一维数组，长度为M，存放M个未知函数值$y_i(T)$ $(i=1, 2, \cdots, M)$；D为双精度实型一维数组，长度为M，返回方程组中M个方程的右端函数值$f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

EPS——实型变量，输入参数，控制精度要求。

D——双精度实型一维数组，长度为M。本子程序的工作数组，用于存放方程组中M个方程的右端函数值$f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

B——双精度实型二维数组，体积为10×M。本子程序的工作数组。

U,V,W——均为双精度实型一维数组，长度为M。本子程序的工作数组。

五、子程序(文件名:GPBS1.FOR)

```
      SUBROUTINE GPBS1(T,H,Y,M,F,EPS,D,B,U,V,W)
      DIMENSION Y(M),D(M),B(10,M),G(10)
      DIMENSION U(M),V(M),W(M)
      DOUBLE PRECISION Y,D,B,G,U,V,T,H,X,HH,DD,Q,W
      DO 20 J=1,M
20    V(J)=Y(J)
      X=T
      NN=1
      HH=H
      G(1)=HH
      CALL EULER(X,Y,M,HH,F,D,W)
      DO 30 J=1,M
        B(1,J)=Y(J)
        U(J)=Y(J)
30    CONTINUE
      K=2
40    NN=NN+NN
      HH=HH/2.0
      G(K)=HH
      DO 50 J=1,M
50    Y(J)=V(J)
      T=X
      DO 60 J=1,NN
        CALL EULER(T,Y,M,HH,F,D,W)
        T=T+HH
60    CONTINUE
      DO 80 J=1,M
        DD=Y(J)
        DO 70 II=1,K-1
          Q=DD-B(II,J)
          IF (ABS(Q)+1.0.EQ.1.0) THEN
            DD=SIGN(1.0D+35,Q)
            DD=DD*SIGN(1.0D0,G(K)-G(II))
          ELSE
            DD=(G(K)-G(II))/Q
          END IF
70      CONTINUE
        B(K,J)=DD
80    CONTINUE
      DO 100 J=1,M
```

```
      DD=0.0
      DO 90 II=K-1,1,-1
        DD=B(II+1,J)+DD
        IF (ABS(DD)+1.0D0.EQ.1.0D0) THEN
          DD=SIGN(1.0D+35,DD)
          DD=DD*SIGN(1.0D0,-G(II))
        ELSE
          DD=-G(II)/DD
        END IF
90      CONTINUE
      Y(J)=DD+B(1,J)
100   CONTINUE
      P=0.0
      DO 110 J=1,M
        Q=ABS(Y(J)-U(J))
        IF (Q.GT.P) P=Q
110   CONTINUE
      IF ((P.GE.EPS).AND.(K.LT.7)) THEN
        DO 120 J=1,M
120     U(J)=Y(J)
        K=K+1
        GOTO 40
      END IF
      T=X
      RETURN
      END

      SUBROUTINE EULER(T,Y,M,H,F,D,A)
      DIMENSION Y(M),D(M),A(M)
      DOUBLE PRECISION A,Y,D,T,H,X
      CALL F(T,Y,M,D)
      DO 20 I=1,M
20    A(I)=Y(I)+H*D(I)
      X=T+H
      CALL F(X,A,M,D)
      DO 30 I=1,M
30    D(I)=Y(I)+H*D(I)
      DO 40 I=1,M
40    Y(I)=(A(I)+D(I))/2.0
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = -y_2, & y_1(0) = 1.0 \\ y_2' = y_1, & y_2(0) = 0.0 \end{cases}$$

用积分一步的连分式法计算当 $h=0.1$ 时，各积分点

$$t_i = (i-1)h, \quad i = 1, 2, \cdots, 11$$

上的未知函数值 y_{1i}，y_{2i}($i=1, 2, \cdots, 11$)。取 $\epsilon=0.000001$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GPBS10.FOR)为

```
      EXTERNAL F
      DIMENSION Y(2),D(2),B(10,2),U(2),V(2),W(2)
      DOUBLE PRECISION Y,D,T,H,B,U,V,W
      T=0.0
      H=0.1
      M=2
      EPS=0.000001
      Y(1)=1.0
      Y(2)=0.0
      WRITE(*,*)
      WRITE(*,20) T,Y(1),Y(2)
      WRITE(*,*)
      DO 10 I=1,10
        CALL GPBS1(T,H,Y,M,F,EPS,D,B,U,V,W)
        T=T+H
        WRITE(*,20) T,Y(1),Y(2)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F7.3,5X,'Y1=',D13.6,5X,'Y2=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=-Y(2)
      D(2)=Y(1)
      RETURN
      END
```

运行结果为

```
T=   .000   Y1=   .100000D+01   Y2=   .000000D+00

T=   .100   Y1=   .995004D+00   Y2=   .998334D-01

T=   .200   Y1=   .980067D+00   Y2=   .198669D+00
```

T=	.300	Y1=	.955337D+00	Y2=	.295520D+00
T=	.400	Y1=	.921061D+00	Y2=	.389418D+00
T=	.500	Y1=	.877583D+00	Y2=	.479426D+00
T=	.600	Y1=	.825336D+00	Y2=	.564642D+00
T=	.700	Y1=	.764842D+00	Y2=	.644218D+00
T=	.800	Y1=	.696707D+00	Y2=	.717356D+00
T=	.900	Y1=	.621610D+00	Y2=	.783327D+00
T=	1.000	Y1=	.540302D+00	Y2=	.841471D+00

本问题的精确解为

$$\begin{cases} y_1 = \cos t \\ y_2 = \sin t \end{cases}$$

7.10 全区间积分的连分式法

一、功能

用连分式法对一阶微分方程组进行全区间积分

二、方法说明

同 7.9 节的方法说明。

三、子程序语句

SUBROUTINE GPBS2 (A,H,N,Y,M,Z,F,EPS,D,B,U,V,W)

四、形参说明

A——双精度实型变量，输入参数。积分的起始点值。

H——双精度实型变量，输入参数。积分的步长。

N——整型变量，输入参数。积分的步数(包括起始点这一步)。

Y——双精度实型一维数组，长度为 M，输入参数。存放起始点 A 处的 M 个未知函数值 $y_i(A)$ $(i=1, 2, \cdots, M)$，返回时其值在 $Z(i, 1)$ $(i=1, 2, \cdots, M)$中。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

Z——双精度实型二维数组，体积为 M×N，输出参数。存放 N 个积分点上的 M 个未知函数值，即

$$Z(i, j) = y_{ij}, \ i = 1, 2, \cdots, M; \ j = 1, 2, \cdots, N$$

其中 $Z(i, 1)$ $(i=1, 2, \cdots, M)$为初值。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$，在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未

知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$；D 为双精度实型一维数组，长度为 M，返回方程组中 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

EPS——实型变量，输入参数。控制精度要求。

D——双精度实型一维数组，长度为 M。本子程序的工作数组，用于存放方程组中 M 个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

B——双精度实型二维数组，体积为 10×M。本子程序的工作数组。

U，V，W——均为双精度实型一维数组，长度为 M。本子程序的工作数组。

五、子程序（文件名：GPBS2.FOR）

```
      SUBROUTINE GPBS2(A,H,N,Y,M,Z,F,EPS,D,B,U,V,W)
      DIMENSION Y(M),D(M),Z(M,N),B(10,M),G(10)
      DIMENSION U(M),V(M),W(M)
      DOUBLE PRECISION Y,D,Z,B,G,U,V,A,H,X,HH,DD,Q,W
      DO 10 I=1,M
10    Z(I,1)=Y(I)
      DO 200 I=2,N
        DO 20 J=1,M
20      V(J)=Y(J)
        X=A+(I-2)*H
        NN=1
        HH=H
        G(1)=HH
        CALL EULER(X,Y,M,HH,F,D,W)
        DO 30 J=1,M
          B(1,J)=Y(J)
          U(J)=Y(J)
30      CONTINUE
        K=2
40      NN=NN+NN
        HH=HH/2.0
        G(K)=HH
        DO 50 J=1,M
50      Y(J)=V(J)
        T=X
        DO 60 J=1,NN
          CALL EULER(T,Y,M,HH,F,D,W)
          T=T+HH
60      CONTINUE
        DO 80 J=1,M
          DD=Y(J)
          DO 70 II=1,K-1
            Q=DD-B(II,J)
```

```
            IF (ABS(Q)+1.0.EQ.1.0) THEN
              DD=SIGN(1.0D+35,Q)
              DD=DD*SIGN(1.0D0,G(K)-G(II))
            ELSE
              DD=(G(K)-G(II))/Q
            END IF
70        CONTINUE
          B(K,J)=DD
80      CONTINUE

        DO 100 J=1,M
          DD=0.0
          DO 90 II=K-1,1,-1
            DD=DD+B(II+1,J)
            IF (ABS(DD)+1.0D0.EQ.1.0D0) THEN
              DD=SIGN(1.0D+35,DD)
              DD=DD*SIGN(1.0D0,-G(II))
            ELSE
              DD=-G(II)/DD
            END IF
90        CONTINUE
          Y(J)=DD+B(1,J)
100     CONTINUE
        P=0.0
        DO 110 J=1,M
          Q=ABS(Y(J)-U(J))
          IF (Q.GT.P) P=Q
110     CONTINUE
        IF ((P.GE.EPS).AND.(K.LT.7)) THEN
          DO 120 J=1,M
120       U(J)=Y(J)
          K=K+1
          GOTO 40
        END IF
        DO 130 J=1,M
130     Z(J,I)=Y(J)
200   CONTINUE
      RETURN
      END

      SUBROUTINE EULER(T,Y,M,H,F,D,A)
      DIMENSION Y(M),D(M),A(M)
```

```
      DOUBLE PRECISION A,Y,D,T,H,X
      CALL F(T,Y,M,D)
      DO 20 I=1,M
20    A(I)=Y(I)+H*D(I)
      X=T+H
      CALL F(X,A,M,D)
      DO 30 I=1,M
30    D(I)=Y(I)+H*D(I)
      DO 40 I=1,M
40    Y(I)=(A(I)+D(I))/2.0
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = -y_2, & y_1(0) = 1.0 \\ y_2' = y_1, & y_2(0) = 0.0 \end{cases}$$

用全区间积分的连分式法计算当 $h=0.1$ 时，各积分点

$$t_i = (i-1)h,\ i = 1, 2, \cdots, 11$$

上的未知函数值 y_{1i}、y_{2i} $(i=1, 2, \cdots, 11)$。取 $\varepsilon=0.000001$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GPBS20.FOR)为

```
      EXTERNAL F
      DIMENSION Y(2),D(2),Z(2,11),B(10,2),U(2),V(2),W(2)
      DOUBLE PRECISION Y,D,Z,A,H,T,B,U,V,W
      A=0.0
      H=0.1
      N=11
      M=2
      EPS=0.000001
      Y(1)=1.0
      Y(2)=0.0
      CALL GPBS2(A,H,N,Y,M,Z,F,EPS,D,B,U,V,W)
      WRITE(*,*)
      DO 10 I=1,N
        T=(I-1)*H
        WRITE(*,20) T,Z(1,I),Z(2,I)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F7.3,5X,'Y1=',D13.6,5X,'Y2=',D13.6)
      END
```

```
SUBROUTINE F(T,Y,M,D)
DIMENSION Y(M),D(M)
DOUBLE PRECISION Y,D,T
D(1)=-Y(2)
D(2)=Y(1)
RETURN
END
```

运行结果为

```
T=   .000   Y1=  .100000D+01   Y2=  .000000D+00
T=   .100   Y1=  .995004D+00   Y2=  .998334D-01
T=   .200   Y1=  .980067D+00   Y2=  .198669D+00
T=   .300   Y1=  .955337D+00   Y2=  .295520D+00
T=   .400   Y1=  .921061D+00   Y2=  .389418D+00
T=   .500   Y1=  .877583D+00   Y2=  .479426D+00
T=   .600   Y1=  .825336D+00   Y2=  .564642D+00
T=   .700   Y1=  .764842D+00   Y2=  .644218D+00
T=   .800   Y1=  .696707D+00   Y2=  .717356D+00
T=   .900   Y1=  .621610D+00   Y2=  .783327D+00
T=  1.000   Y1=  .540302D+00   Y2=  .841471D+00
```

本问题的精确解为

$$\begin{cases} y_1 = \cos t \\ y_2 = \sin t \end{cases}$$

7.11 全区间积分的双边法

一、功能

用双边法对一阶微分方程组进行全区间积分。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), \quad y_i(t_0) = y_{i0}$$
$$i = 1, 2, \cdots, m$$

用双边法积分一步的计算公式为：

$$\begin{cases} p_i^{(j+2)} = -4y_i^{(j+1)} + 5y_i^{(j)} + 2h[2f_i^{(j+1)} + f_i^{(j)}] \\ q_i^{(j+2)} = 4y_i^{(j+1)} - 3y_i^{(j)} + \dfrac{2}{3}h[f_i^{(j+2)} - 2f_i^{(j+1)} - 2f_i^{(j)}] \\ y_i^{(j+2)} = \dfrac{1}{2}[p_i^{(j+2)} + q_i^{(j+2)}] \end{cases}$$

$$i=1, 2, \cdots, m$$

其中

$$y_i^{(j)}=y_i(t_j)$$

$$f_i^{(j)}=f_i(t_j, y_1^{(j)}, y_2^{(j)}, \cdots, y_m^{(j)})$$

$$f_i^{(j+1)}=f_i(t_j+h, y_1^{(j+1)}, y_2^{(j+1)}, \cdots, y_m^{(j+1)})$$

$$f_i^{(j+2)}=f_i(t_j+2h, p_1^{(j+2)}, p_2^{(j+2)}, \cdots, p_m^{(j+2)})$$

在本子程序中，采用积分一步的变步长基尔(Gill)方法起步算出 $y_i(t_1)$ $(i=1, 2, \cdots, m)$。

三、子程序语句

SUBROUTINE GGJFQ (A,H,N,Y,M,Z,F,D,EPS,P,F1,F2,F3,G)

四、形参说明

A——双精度实型变量，输入参数。积分的起始点值。

H——双精度实型变量，输入参数。积分的步长。

N——整型变量，输入参数，积分的步数(包括起始点这一步)。

Y——双精度实型一维数组，长度为 M，输入参数。存放在起始点 A 处的 M 个未知函数值 $y_i(A)$ $(i=1, 2, \cdots, M)$。返回时其值在 $Z(i, 1)$ $(i=1, 2, \cdots, M)$中。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

Z——双精度实型二维数组，体积为 M×N，输出参数。存放 N 个积分点上的 M 个未知函数值，即

$$Z(i, j)=y_{ij}, i=1, 2, \cdots, M; j=1, 2, \cdots, N$$

其中 $Z(i, 1)$ $(i=1, 2, \cdots, M)$为初值。

F——子程序名，输入参数，用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$；D 为双精度实型一维数组，长度为 M，返回方程组中 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

D——双精度实型一维数组，长度为 M。本子程序的工作数组，主要用于存放方程组中 M 个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

EPS——实型变量，输入参数。控制精度要求。

P,F1,F2,F3,G——均为双精度实型一维数组，长度为 M。本子程序的工作数组。

五、子程序(文件名:GGJFQ.FOR)

```
      SUBROUTINE GGJFQ(A,H,N,Y,M,Z,F,D,EPS,P,F1,F2,F3,G)
      DIMENSION Y(M),D(M),Z(M,N),P(M),F1(M),F2(M),F3(M),G(M)
      DOUBLE PRECISION Y,D,Z,P,F1,F2,F3,A,H,T,QQ,G
      DO 5 I=1,M
5     P(I)=0.0
```

```
      DO 10 I=1,M
10    Z(I,1)=Y(I)
      T=A
      CALL F(T,Y,M,D)
      DO 20 J=1,M
20    F1(J)=D(J)
      CALL GGIL1(T,H,Y,M,F,EPS,P,D,F3,F2,G)
      T=A+H
      CALL F(T,Y,M,D)
      DO 30 J=1,M
        Z(J,2)=Y(J)
        F2(J)=D(J)
30    CONTINUE
      DO 40 J=1,M
        P(J)=-4*Z(J,2)+5*Z(J,1)+2*H*(2*F2(J)+F1(J))
        Y(J)=P(J)
40    CONTINUE
      T=A+2*H
      CALL F(T,Y,M,D)
      DO 50 J=1,M
        QQ=2*H*(D(J)-2*F2(J)-2*F1(J))/3.0
        QQ=QQ+4*Z(J,2)-3*Z(J,1)
        Z(J,3)=(P(J)+QQ)/2.0
        Y(J)=Z(J,3)
50    CONTINUE
      DO 100 I=4,N
        T=A+(I-2)*H
        CALL F(T,Y,M,D)
        DO 60 J=1,M
          F1(J)=F2(J)
          F2(J)=D(J)
60      CONTINUE
        DO 70 J=1,M
          P(J)=-4*Z(J,I-1)+5*Z(J,I-2)+2*H*(2*F2(J)+F1(J))
          Y(J)=P(J)
70      CONTINUE
        T=T+H
        CALL F(T,Y,M,D)
        DO 80 J=1,M
          QQ=2*H*(D(J)-2*F2(J)-2*F1(J))/3.0
          QQ=QQ+4*Z(J,I-1)-3*Z(J,I-2)
          Z(J,I)=(P(J)+QQ)/2.0
```

```
      Y(J)=Z(J,I)
 80     CONTINUE
 100  CONTINUE
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = -y_2, & y_1(0) = 1.0 \\ y_2' = y_1, & y_2(0) = 0.0 \end{cases}$$

用双边法计算当 $h=0.1$ 时，各积分点

$$t_i = (i-1)h, \ i = 1, 2, \cdots, 11$$

上的未知函数值 $y_1(t_i)$，$y_2(t_i)$。取 $\varepsilon=0.000001$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GGJFQ0.FOR)为

```
      EXTERNAL F
      DIMENSION Y(2),D(2),Z(2,11),P(2),F1(2),F2(2),F3(2),G(2)
      DOUBLE PRECISION Y,D,Z,A,H,T,P,F1,F2,F3,G
      DATA Y/1.0,0.0/
      A=0.0
      H=0.1
      N=11
      M=2
      EPS=0.000001
      CALL GGJFQ(A,H,N,Y,M,Z,F,D,EPS,P,F1,F2,F3,G)
      WRITE(*,*)
      DO 10 I=1,N
        T=(I-1)*H
        WRITE(*,20) T,Z(1,I),Z(2,I)
        WRITE(*,*)
 10   CONTINUE
 20   FORMAT(1X,'T=',F6.3,5X,'Y1=',D13.6,5X,'Y2=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=-Y(2)
      D(2)=Y(1)
      RETURN
      END
```

运行结果为

```
T=   .000    Y1=   .100000D+01    Y2=   .000000D+00
T=   .100    Y1=   .995004D+00    Y2=   .998334D-01
T=   .200    Y1=   .980067D+00    Y2=   .198669D+00
T=   .300    Y1=   .955337D+00    Y2=   .295520D+00
T=   .400    Y1=   .921061D+00    Y2=   .389417D+00
T=   .500    Y1=   .877583D+00    Y2=   .479425D+00
T=   .600    Y1=   .825336D+00    Y2=   .564641D+00
T=   .700    Y1=   .764843D+00    Y2=   .644217D+00
T=   .800    Y1=   .696708D+00    Y2=   .717355D+00
T=   .900    Y1=   .621611D+00    Y2=   .783326D+00
T=  1.000    Y1=   .540304D+00    Y2=   .841470D+00
```

本问题的精确解为

$$\begin{cases} y_1 = \cos t \\ y_2 = \sin t \end{cases}$$

七、附注

本子程序需要调用变步长基尔方法积分一步子程序 GGIL1，参看 7.6 节。

7.12 全区间积分的阿当姆斯预报-校正法

一、功能

用阿当姆斯(Adams)预报-校正公式对一阶微分方程组进行全区间积分。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), \quad y_i(t_0) = y_{i0}$$

$$i = 1, 2, \cdots, m$$

积分步长为 h。

阿当姆斯预报-校正公式为

预报公式 $\bar{y}_{i,j+1} = y_{ij} + h[55f_{ij} - 59f_{i,j-1} + 37f_{i,j-2} - 9f_{i,j-3}]/24$

校正公式 $y_{i,j+1} = y_{ij} + h[9\bar{f}_{i,j+1} + 19f_{ij} - 5f_{i,j-1} + f_{i,j-2}]/24$

$$i = 1, 2, \cdots, m$$

其中

$$f_{ik} = f_i(t_k, y_{1k}, y_{2k}, \cdots, y_{mk}), \quad k = j-3, j-2, j-1, j$$

$$\bar{f}_{i,j+1} = f_i(t_{j+1}, \bar{y}_{1,j+1}, \bar{y}_{2,j+1}, \cdots, \bar{y}_{m,j+1})$$

由于阿当姆斯方法是线性多步法，在计算新点上的各未知函数值时要用到前面四个点上的各未知函数数值。因此，在用这个方法时，必须要先用别的单步法计算开始四个点上

的各未知函数值。在本子程序中，采用变步长四阶龙格-库塔公式计算开始四个点上的各未知函数值 y_{i1}，y_{i2}，y_{i3} 与 y_{i4}，其中 y_{i1} 为给定的初值($i=1, 2, \cdots, m$)。

三、子程序语句

SUBROUTINE GADMS (T,H,N,Y,M,Z,F,D,EPS,B,E,S,G)

四、形参说明

T——双精度实型变量，输入参数。积分的起始点值。

H——双精度实型变量，输入参数。积分的步长。

N——整型变量，输入参数。积分的步数(包括起始点这一步)。

Y——双精度实型一维数组，长度为 M，输入参数。存放起始点 T 处的 M 个未知函数值 $y_i(T)$ ($i=1, 2, \cdots, M$)。返回时其值在 Z(i, 1) ($i=1, 2, \cdots, M$)中。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

Z——双精度实型二维数组，体积为 M×N，输出参数。存放 N 个积分点上的 M 个未知函数值，即

$$Z(i, j) = y_{ij},\ i = 1, 2, \cdots, M;\ j = 1, 2, \cdots, N$$

其中 Z(i, 1) ($i=1, 2, \cdots, M$)为初值。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ ($i=1, 2, \cdots, M$)。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未知函数值 $y_i(T)$ ($i=1, 2, \cdots, M$)；D 为双精度实型一维数组，长度为 M，返回方程组中 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ ($i=1, 2, \cdots, M$)。

D——双精度实型一维数组，长度为 M。本子程序的工作数组，用于存放方程组中 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ ($i=1, 2, \cdots, M$)。

EPS——实型变量，输入参数。控制精度要求。

B——双精度实型二维数组，体积为 4×M。本子程序的工作数组。

E,S,G——均为双精度实型一维数组，长度为 M。本子程序的工作数组。

五、子程序(文件名：GADMS. FOR)

```
      SUBROUTINE GADMS(T,H,N,Y,M,Z,F,D,EPS,B,E,S,G)
      EXTERNAL F
      DIMENSION Y(M),D(M),Z(M,N),B(4,M),E(M),S(M),G(M)
      DOUBLE PRECISION Y,Z,D,B,T,H,A,Q,E,S,G
      A=T
      DO 5 I=1,M
5     Z(I,1)=Y(I)
      CALL F(T,Y,M,D)
      DO 10 I=1,M
10    B(1,I)=D(I)
      DO 40 I=2,4
```

```
      IF (I.LE.N) THEN
        T=A+(I-1)*H
        CALL GRKT2(T,H,Y,M,F,EPS,D,E,S,G,Z(1,N))
        DO 20 K=1,M
20      Z(K,I)=Y(K)
        CALL F(T,Y,M,D)
        DO 30 K=1,M
30      B(I,K)=D(K)
      END IF
40    CONTINUE
      DO 100 I=5,N
        DO 50 J=1,M
          Q=55*B(4,J)-59*B(3,J)+37*B(2,J)-9*B(1,J)
          Y(J)=Z(J,I-1)+H*Q/24.0
          B(1,J)=B(2,J)
          B(2,J)=B(3,J)
          B(3,J)=B(4,J)
50      CONTINUE
        T=A+(I-1)*H
        CALL F(T,Y,M,D)
        DO 60 K=1,M
60      B(4,K)=D(K)
        DO 70 J=1,M
          Q=9*B(4,J)+19*B(3,J)-5*B(2,J)+B(1,J)
          Z(J,I)=Z(J,I-1)+H*Q/24.0
          Y(J)=Z(J,I)
70      CONTINUE
        CALL F(T,Y,M,D)
        DO 80 K=1,M
80      B(4,K)=D(K)
100   CONTINUE
      T=A
      RETURN
      END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = 0.0 \\ y_2' = -y_1, & y_2(0) = 1.0 \\ y_3' = -y_3, & y_3(0) = 1.0 \end{cases}$$

用阿当姆斯预报-校正公式计算当 $h=0.05$ 时，各积分点

$$t_i=(i-1)h,\ i=1,2,\cdots,11$$

上的未知函数值 $y_{1i},y_{2i},y_{3i}(i=1,2,\cdots,11)$。取 $\varepsilon=0.0001$。

主程序及计算方程组中各方程右端函数值的子程序（文件名：GADMS0.FOR）为

```
      EXTERNAL F
      DIMENSION Y(3),D(3),Z(3,11),B(4,3),E(3),P(3),G(3)
      DOUBLE PRECISION Y,D,Z,A,T,H,B,E,P,G
      A=0.0
      H=0.05
      Y(1)=0.0
      Y(2)=1.0
      Y(3)=1.0
      N=11
      M=3
      EPS=0.0001
      CALL GADMS(A,H,N,Y,M,Z,F,D,EPS,B,E,P,G)
      WRITE(*,*)
      DO 10 I=1,N
        T=(I-1)*H
        WRITE(*,20) T,(Z(J,I),J=1,M)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F6.3,3X,'Y1=',D13.6,3X,'Y2=',D13.6,
     *       3X,'Y3=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=-Y(3)
      RETURN
      END
```

运行结果为

```
T=  .000   Y1=  .000000D+00   Y2=  .100000D+01   Y3=  .100000D+01

T=  .050   Y1=  .499792D-01   Y2=  .998750D+00   Y3=  .951229D+00

T=  .100   Y1=  .998334D-01   Y2=  .995004D+00   Y3=  .904837D+00

T=  .150   Y1=  .149438D+00   Y2=  .988771D+00   Y3=  .860708D+00

T=  .200   Y1=  .198669D+00   Y2=  .980067D+00   Y3=  .818731D+00

T=  .250   Y1=  .247404D+00   Y2=  .968912D+00   Y3=  .778801D+00
```

```
T=  .300  Y1=  .295520D+00  Y2=  .955336D+00  Y3=  .740818D+00
T=  .350  Y1=  .342898D+00  Y2=  .939373D+00  Y3=  .704688D+00
T=  .400  Y1=  .389418D+00  Y2=  .921061D+00  Y3=  .670320D+00
T=  .450  Y1=  .434966D+00  Y2=  .900447D+00  Y3=  .637628D+00
T=  .500  Y1=  .479426D+00  Y2=  .877583D+00  Y3=  .606531D+00
```

本问题的精确解为

$$\begin{cases} y_1 = \sin t \\ y_2 = \cos t \\ y_3 = e^{-t} \end{cases}$$

七、附注

本子程序需要调用变步长四阶龙格-库塔法积分一步子程序 GRKT2，参看 7.5 节。

7.13 全区间积分的哈明方法

一、功能

用哈明(Hamming)方法对一阶微分方程组进行全区间积分。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), y_i(t_0) = y_{i0}$$
$$i = 1, 2, \cdots, m$$

积分步长为 h。

哈明方法的计算公式为

预报 $\quad \overline{P}_{i,j+1} = y_{i,j-3} + 4h(2f_{ij} - f_{i,j-1} + 2f_{i,j-2})/3$

$$i = 1, 2, \cdots, m$$

修正 $\quad P_{i,j+1} = \overline{P}_{i,j+1} + \frac{112}{121}(C_{ij} - P_{ij}), \quad i = 1, 2, \cdots, m$

校正 $\quad C_{i,j+1} = \frac{1}{8}[9y_{ij} - y_{i,j-2} + 3h(f_{i,j+1} + 2f_{ij} - f_{i,j-1})]$

$$i = 1, 2, \cdots, m$$

终值 $\quad y_{i,j+1} = C_{i,j+1} - \frac{9}{121}(C_{i,j+1} - P_{i,j+1}), \quad i = 1, 2, \cdots, m$

其中

$$f_{ik} = f_i(t_k, y_{1k}, y_{2k}, \cdots, y_{mk}), \quad k = j-2, j-1, j$$
$$f_{i,j+1} = f_i(t_{j+1}, P_{1,j+1}, P_{2,j+1}, \cdots, P_{m,j+1})$$

哈明方法是线性多步法，在计算新点上的各未知函数值要用到前四个点上的各未知函数值。因此，在用本方法时，要先用别的单步法计算开始三个点上的各未知函数值。在本子程序中，采用变步长四阶龙格-库塔公式计算开始三个点上的各函数值 y_{i1}，y_{i2}，y_{i3}($i=1, 2, \cdots, m$)，其中 y_{i0}为初值。

三、子程序语句

SUBROUTINE GHAMG (T,H,N,Y,M,Z,F,D,EPS,B,U,BB,CC,GG)

四、形参说明

T——双精度实型变量,输入参数。积分的起始点值。

H——双精度实型变量,输入参数。积分的步长。

N——整型变量,输入参数。积分的步数(包括起始点这一步)。

Y——双精度实型一维数组,长度为M,输入参数。存放在起始点T处的M个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$。返回时其值在 $Z(i=1)$ $(i=1, 2, \cdots, M)$中。

M——整型变量,输入参数。方程组中方程的个数,也是未知函数的个数。

Z——双精度实型二维数组,体积为M×N,输出参数。存放N个积分点上的M个未知函数值,即

$$Z(i, j) = y_{ij}, i = 1, 2, \cdots, M; i = 1, 2, \cdots, N$$

其中 $Z(i, 1)$ $(i=1, 2, \cdots, M)$为初值。

F——子程序名,输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$,在主程序中必须用外部语句对相应的实参进行说明,该子程序由用户自编,其语句形式为

SUBROUTINE F(T,Y,M,D)

其中:T为双精度实型变量,自变量值;Y为双精度实型一维数组,长度为M,存放M个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$;D为双精度实型一维数组,长度为M,返回方程组中M个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

D——双精度实型一维数组,长度为M。本子程序的工作数组,主要用于存放方程组中M个方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

EPS——实型变量,输入参数。控制精度要求。

B——双精度实型二维数组,体积为4×M。本子程序的工作数组。

U,BB,CC,GG——均为双精度实型一维数组,长度为M。本子程序的工作数组。

五、子程序(文件名:GHAMG.FOR)

```
      SUBROUTINE GHAMG(T,H,N,Y,M,Z,F,D,EPS,B,U,BB,CC,GG)
      EXTERNAL F
      DIMENSION Y(M),D(M),Z(M,N),B(4,M),U(M),BB(M),CC(M),GG(M)
      DOUBLE PRECISION Y,D,Z,B,U,A,T,H,Q,BB,CC,GG
      A=T
      DO 5 I=1,M
5     Z(I,1)=Y(I)
      CALL F(T,Y,M,D)
      DO 10 I=1,M
10    B(1,I)=D(I)
      DO 40 I=2,4
        IF (I.LE.N) THEN
          T=A+(I-1)*H
```

```
            CALL GRKT2(T,H,Y,M,F,EPS,D,BB,CC,GG,Z(1,N))
            DO 20 K=1,M
20          Z(K,I)=Y(K)
            CALL F(T,Y,M,D)
            DO 30 K=1,M
30          B(I,K)=D(K)
          END IF
40      CONTINUE
        DO 45 I=1,M
45      U(I)=0.0
        DO 100 I=5,N
          DO 50 J=1,M
            Q=2*B(4,J)-B(3,J)+2*B(2,J)
            IF (I.EQ.5) THEN
              Y(J)=Z(J,1)+4*H*Q/3.0
            ELSE
              Y(J)=Z(J,I-4)+4*H*Q/3.0
            END IF
50        CONTINUE
          DO 70 J=1,M
70        Y(J)=Y(J)+112*U(J)/121.0
          T=A+(I-1)*H
          CALL F(T,Y,M,D)
          DO 80 J=1,M
            Q=(9*Z(J,I-1)-Z(J,I-3)+3*H*(D(J)+2*B(4,J)-B(3,J)))/8.0
            U(J)=Q-Y(J)
            Z(J,I)=Q-9*U(J)/121.0
            Y(J)=Z(J,I)
            B(2,J)=B(3,J)
            B(3,J)=B(4,J)
80        CONTINUE
          CALL F(T,Y,M,D)
          DO 90 K=1,M
90        B(4,K)=D(K)
100     CONTINUE
        T=A
        RETURN
        END
```

六、例

设一阶微分方程组为

$$\begin{cases} y_1' = y_2, & y_1(0) = 1.0 \\ y_2' = -y_1, & y_2(0) = 1.0 \\ y_3' = y_3, & y_3(0) = 1.0 \end{cases}$$

用哈明方法计算当 $h=0.05$ 时，各积分点

$$t_i = (i-1)h, \quad i = 1, 2, \cdots, 11$$

上的函数值 $y_{1i}, y_{2i}, y_{3i}(i=1, 2, \cdots, 11)$。取 $\varepsilon=0.0001$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GHAMG0.FOR)为

```
      EXTERNAL F
      DIMENSION Y(3),D(3),Z(3,11),B(4,3),U(3),BB(3),CC(3),GG(3)
      DOUBLE PRECISION Y,D,Z,A,T,H,B,U,BB,CC,GG
      A=0.0
      H=0.05
      Y(1)=1.0
      Y(2)=1.0
      Y(3)=1.0
      N=11
      M=3
      EPS=0.0001
      CALL GHAMG(A,H,N,Y,M,Z,F,D,EPS,B,U,BB,CC,GG)
      WRITE(*,*)
      DO 10 I=1,N
        T=(I-1)*H
        WRITE(*,20) T,(Z(J,I),J=1,M)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F6.3,3X,'Y1=',D13.6,3X,'Y2=',D13.6,
     *       3X,'Y3=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=Y(2)
      D(2)=-Y(1)
      D(3)=Y(3)
      RETURN
      END
```

运行结果为

```
T=  .000   Y1=  .100000D+01   Y2=  .100000D+01   Y3=  .100000D+01

T=  .050   Y1=  .104873D+01   Y2=  .948771D+00   Y3=  .105127D+01
```

```
T=  .100  Y1=  .109484D+01  Y2=  .895171D+00  Y3=  .110517D+01
T=  .150  Y1=  .113821D+01  Y2=  .839333D+00  Y3=  .116183D+01
T=  .200  Y1=  .117874D+01  Y2=  .781397D+00  Y3=  .122140D+01
T=  .250  Y1=  .121632D+01  Y2=  .721508D+00  Y3=  .128403D+01
T=  .300  Y1=  .125086D+01  Y2=  .659816D+00  Y3=  .134986D+01
T=  .350  Y1=  .128227D+01  Y2=  .596475D+00  Y3=  .141907D+01
T=  .400  Y1=  .131048D+01  Y2=  .531643D+00  Y3=  .149182D+01
T=  .450  Y1=  .133541D+01  Y2=  .465482D+00  Y3=  .156831D+01
T=  .500  Y1=  .135701D+01  Y2=  .398157D+00  Y3=  .164872D+01
```

本问题的精确解为

$$\begin{cases} y_1 = \sin t + \cos t \\ y_2 = \cos t - \sin t \\ y_3 = e^t \end{cases}$$

七、附注

本子程序需要调用变步长四阶龙格-库塔法积分一步的子程序 GRKT2，参看 7.5 节。

7.14 积分一步的特雷纳方法

一、功能

用特雷纳(Treanor)方法对一阶刚性(Stiff)微分方程组积分一步。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), \quad y_i(t_0) = y_{i0}$$
$$i = 1, 2, \cdots, m$$

如果矩阵

$$\begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots & \frac{\partial f_1}{\partial y_m} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \cdots & \frac{\partial f_2}{\partial y_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial y_1} & \frac{\partial f_m}{\partial y_2} & \cdots & \frac{\partial f_m}{\partial y_m} \end{bmatrix}$$

的特征值 λ_i 具有如下特性：

$$\mathrm{Re}\lambda_i < 0 \text{ 且 } \max_{1 \le i \le m} |\mathrm{Re}\lambda_i| \gg \min_{1 \le i \le m} |\mathrm{Re}\lambda_i|$$

则称此微分方程组为刚性的。

求解刚性微分方程组的特雷纳方法如下。

设已知 t_j 点处的未知函数值 $y_{ij}(i=1, 2, \cdots, m)$，则计算 $t_{j+1} = t_j + h$ 点处未知函数

值 $y_{i,j+1}$ 的公式为

$$y_{i,j+1} = y_{ij} + \Delta y_i,\ i = 1,2,\cdots,m$$

其中

$$\Delta y_i = \begin{cases} \dfrac{h}{6}[g_i^{(1)} + 2(g_i^{(2)} + g_i^{(3)}) + g_i^{(4)}], \quad p_i \leqslant 0 \\ h\{g_i^{(1)} r_i^{(2)} + [-3(g_i^{(1)} + p_i Z_i^{(1)}) + 2(g_i^{(2)} + p_i Z_i^{(2)}) \\ \qquad + 2(g_i^{(3)} + p_i Z_i^{(3)}) - (g_i^{(4)} + p_i Z_i^{(4)})] r_i^{(3)} \\ \qquad + 4[(g_i^{(1)} + p_i Z_i^{(1)}) - (g_i^{(2)} + p_i Z_i^{(2)}) - (g_i^{(3)} + p_i Z_i^{(3)}) \\ \qquad + (g_i^{(4)} + p_i Z_i^{(4)})] r_i^{(4)}\}, \quad p_i > 0 \end{cases}$$

式中

$$p_i = \frac{g_i^{(3)} - g_i^{(2)}}{Z_i^{(3)} - Z_i^{(2)}},\quad i=1,2,\cdots,m$$

$$r_i^{(1)} = e^{-p_i h},\quad r_i^{(2)} = \frac{r_i^{(1)} - 1}{-p_i h},\quad r_i^{(3)} = \frac{r_i^{(2)} - 1}{-p_i h},$$

$$r_i^{(4)} = \frac{r_i^{(3)} - \dfrac{1}{2}}{-p_i h},\quad i=1,2,\cdots,m$$

$$Z_i^{(1)} = y_{ij},\qquad g_i^{(1)} = f_i(t_j, Z_1^{(1)}, Z_2^{(1)}, \cdots, Z_m^{(1)})$$

$$Z_i^{(2)} = Z_i^{(1)} + \frac{h}{2} g_i^{(1)},\qquad g_i^{(2)} = f_i\left(t_j + \frac{h}{2}, Z_1^{(2)}, Z_2^{(2)}, \cdots, Z_m^{(2)}\right)$$

$$Z_i^{(3)} = Z_i^{(1)} + \frac{h}{2} g_i^{(2)},\qquad g_i^{(3)} = f_i\left(t_j + \frac{h}{2}, Z_1^{(3)}, Z_2^{(3)}, \cdots, Z_m^{(3)}\right)$$

$$Z_i^{(4)} = Z_i^{(1)} + q_i h,\qquad g_i^{(4)} = f_i(t_j + h, Z_1^{(4)}, Z_2^{(4)}, \cdots, Z_m^{(4)})$$

q_i 由下列公式确定：

$$q_i = \begin{cases} g_i^{(3)},\ p_i \leqslant 0 \\ 2(g_i^{(3)} - g_i^{(1)}) r_i^{(3)} + (g_i^{(1)} - g_i^{(2)}) r_i^{(2)} + g_i^{(2)},\ p_i > 0 \end{cases}$$

$$i=1,2,\cdots,m$$

由上述计算公式可知，当 $p_i \leqslant 0$ 时，此方法便退化为四阶龙格-库塔方法。

本方法也适用于求解非刚性的一阶微分方程组。

三、子程序语句

SUBROUTINE GTNR1(T,H,Y,M,F,D,P,Z,G,R)

四、形参说明

T——双精度实型变量，输入参数。积分的起始点值。

H——双精度实型变量，输入参数。积分的步长。

Y——双精度实型一维数组，长度为 M，输入兼输出参数。调用时存放起始点处的未知函数值 $y_i(T)$ $(i=1,2,\cdots,M)$；返回时存放 T+H 点处的未知函数值 $y_i(T+H)$ $(i=1,2,\cdots,M)$。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值 $f_i(t,y_1,y_2,\cdots,y_m)$ $(i=1,2,\cdots,M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程

序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$；D 为双精度实型一维数组，长度为 M，返回方程组中 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

D——双精度实型一维数组，长度为 M。本子程序的工作数组，主要用于存放方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

P——双精度实型一维数组，长度为 M。本子程序的工作数组。

Z,G,R——均为双精度实型二维数组，体积为 4×M。本子程序的工作数组。

五、子程序(文件名：GTNR1.FOR)

```
      SUBROUTINE GTNR1(T,H,Y,M,F,D,P,Z,G,R)
      DIMENSION Y(M),D(M),Z(4,M),G(4,M)
      DIMENSION P(M),R(4,M)
      DOUBLE PRECISION Y,D,Z,G,P,R,T,H,X,S,AA,BB,DD,Q,DY,DY1
      X=T
      DO 50 J=1,M
50    Z(1,J)=Y(J)
      CALL F(T,Y,M,D)
      DO 60 J=1,M
        G(1,J)=D(J)
        Y(J)=Z(1,J)+H*D(J)/2.0
        Z(2,J)=Y(J)
60    CONTINUE
      S=T+H/2.0
      CALL F(S,Y,M,D)
      DO 70 J=1,M
        G(2,J)=D(J)
        Y(J)=Z(1,J)+H*D(J)/2.0
        Z(3,J)=Y(J)
70    CONTINUE
      CALL F(S,Y,M,D)
      DO 80 J=1,M
80    G(3,J)=D(J)
      DO 90 J=1,M
        AA=G(3,J)-G(2,J)
        BB=Z(3,J)-Z(2,J)
        IF (-AA*BB*H.GT.0.0) THEN
          P(J)=-AA/BB
          DD=-P(J)*H
          R(1,J)=EXP(DD)
          R(2,J)=(R(1,J)-1)/DD
```

```
      R(3,J)=(R(2,J)-1)/DD
      R(4,J)=(R(3,J)-1)/DD
    ELSE
      P(J)=0.0
    END IF
    IF (P(J).LE.0.0) THEN
      Q=G(3,J)
    ELSE
      Q=2*(G(3,J)-G(1,J))*R(3,J)
      Q=Q+(G(1,J)-G(2,J))*R(2,J)+G(2,J)
    END IF
    Z(4,J)=Z(1,J)+Q*H
    Y(J)=Z(4,J)
90  CONTINUE
    S=T+H
    CALL F(S,Y,M,D)
    DO 100 J=1,M
100 G(4,J)=D(J)


    DO 110 J=1,M
      IF (P(J).LE.0.0) THEN
        DY=(G(1,J)+2*(G(2,J)+G(3,J))+G(4,J))*H/6.0
      ELSE
        DY=-3*(G(1,J)+P(J)*Z(1,J))+2*(G(2,J)+P(J)*Z(2,J))
        DY=DY+2*(G(3,J)+P(J)*Z(3,J))-(G(4,J)+P(J)*Z(4,J))
        DY=DY*R(3,J)+G(1,J)*R(2,J)
        DY1=G(1,J)-G(2,J)-G(3,J)+G(4,J)
        DY1=DY1+(Z(1,J)-Z(2,J)-Z(3,J)+Z(4,J))*P(J)
        DY=(DY+4*DY1*R(4,J))*H
      END IF
      Y(J)=Z(1,J)+DY
110 CONTINUE
    T=X
    RETURN
    END
```

六、例

设一阶刚性微分方程组为

$$\begin{cases} y_1' = -21y_1 + 19y_2 - 20y_3, & y_1(0) = 1.0 \\ y_2' = 19y_1 - 21y_2 + 20y_3, & y_2(0) = 0.0 \\ y_3' = 40y_1 - 40y_2 - 40y_3, & y_3(0) = -1.0 \end{cases}$$

用积分一步的特雷纳方法计算当 $h=0.001$ 时各积分点

$$t_i=(i-1)h,\ i=1,2,\cdots,101$$

上的未知函数值 $y_1(t_i)$，$y_2(t_i)$，$y_3(t_i)$，$(i=1,2,\cdots,101)$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GTNR10.FOR)为

```
      EXTERNAL F
      DIMENSION Y(3),D(3),P(3),Z(4,3),G(4,3),R(4,3)
      DOUBLE PRECISION Y,D,T,H,P,Z,G,R
      DATA Y/1.0,0.0,-1.0/
      T=0.0
      H=0.001
      M=3
      WRITE(*,*)
      WRITE(*,20) T,Y(1),Y(2),Y(3)
      WRITE(*,*)
      DO 10 I=1,100
        CALL GTNR1(T,H,Y,M,F,D,P,Z,G,R)
        T=T+H
        WRITE(*,20) T,Y(1),Y(2),Y(3)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F6.3,3X,'Y1=',D13.6,3X,'Y2=',D13.6,
     *       3X,'Y3=',D13.6)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=-21*Y(1)+19*Y(2)-20*Y(3)
      D(2)=19*Y(1)-21*Y(2)+20*Y(3)
      D(3)=40*Y(1)-40*Y(2)-40*Y(3)
      RETURN
      END
```

运行结果为

```
T=   .000   Y1=   .100000D+01   Y2=   .000000D+00   Y3=  -.100000D+01

T=   .001   Y1=   .998243D+00   Y2=  -.240944D-03   Y3=  -.921600D+00

T=   .002   Y1=   .995011D+00   Y2=   .896761D-03   Y3=  -.845389D+00

T=   .003   Y1=   .990406D+00   Y2=   .361195D-02   Y3=  -.772932D+00

T=   .004   Y1=   .984554D+00   Y2=   .747766D-02   Y3=  -.703802D+00

T=   .005   Y1=   .977570D+00   Y2=   .124796D-01   Y3=  -.637904D+00
```

```
T=   .006   Y1=   .969562D+00   Y2=   .185105D-01   Y3= -.575151D+00
T=   .007   Y1=   .960829D+00   Y2=   .254688D-01   Y3= -.515481D+00
T=   .008   Y1=   .950869D+00   Y2=   .332583D-01   Y3= -.458830D+00
T=   .009   Y1=   .940374D+00   Y2=   .417877D-01   Y3= -.405136D+00
T=   .010   Y1=   .929229D+00   Y2=   .509707D-01   Y3= -.354333D+00
T=   .011   Y1=   .917515D+00   Y2=   .607257D-01   Y3= -.306354D+00
T=   .012   Y1=   .905311D+00   Y2=   .709756D-01   Y3= -.261126D+00
T=   .013   Y1=   .892688D+00   Y2=   .816478D-01   Y3= -.218576D+00
T=   .014   Y1=   .879715D+00   Y2=   .926741D-01   Y3= -.178628D+00
T=   .015   Y1=   .866456D+00   Y2=   .103990D+00   Y3= -.141204D+00
T=   .016   Y1=   .852971D+00   Y2=   .115536D+00   Y3= -.106226D+00
T=   .017   Y1=   .839316D+00   Y2=   .127257D+00   Y3= -.736113D-01
T=   .018   Y1=   .825543D+00   Y2=   .139098D+00   Y3= -.432796D-01
T=   .019   Y1=   .811700D+00   Y2=   .151014D+00   Y3= -.151488D-01
T=   .020   Y1=   .797831D+00   Y2=   .162959D+00   Y3=   .108605D-01
T=   .021   Y1=   .783971D+00   Y2=   .174899D+00   Y3=   .348586D-01
T=   .022   Y1=   .770169D+00   Y2=   .186786D+00   Y3=   .569026D-01
T=   .023   Y1=   .756458D+00   Y2=   .198584D+00   Y3=   .770751D-01
T=   .024   Y1=   .742872D+00   Y2=   .210262D+00   Y3=   .954583D-01
T=   .025   Y1=   .729439D+00   Y2=   .221791D+00   Y3=   .112133D+00
T=   .026   Y1=   .716187D+00   Y2=   .233142D+00   Y3=   .127180D+00
T=   .027   Y1=   .703140D+00   Y2=   .244293D+00   Y3=   .140678D+00
T=   .028   Y1=   .690318D+00   Y2=   .255221D+00   Y3=   .152705D+00
T=   .029   Y1=   .677743D+00   Y2=   .265907D+00   Y3=   .163337D+00
T=   .030   Y1=   .665430D+00   Y2=   .276335D+00   Y3=   .172648D+00
T=   .031   Y1=   .653395D+00   Y2=   .286489D+00   Y3=   .180712D+00
T=   .032   Y1=   .641650D+00   Y2=   .296355D+00   Y3=   .187600D+00
T=   .033   Y1=   .630207D+00   Y2=   .305925D+00   Y3=   .193381D+00
T=   .034   Y1=   .619074D+00   Y2=   .315187D+00   Y3=   .198123D+00
T=   .035   Y1=   .608260D+00   Y2=   .324134D+00   Y3=   .201890D+00
T=   .036   Y1=   .597771D+00   Y2=   .332760D+00   Y3=   .204747D+00
T=   .037   Y1=   .587610D+00   Y2=   .341062D+00   Y3=   .206754D+00
```

T=	.038	Y1=	.577782D+00	Y2=	.349034D+00	Y3=	.207971D+00
T=	.039	Y1=	.568289D+00	Y2=	.356676D+00	Y3=	.208455D+00
T=	.040	Y1=	.559130D+00	Y2=	.363987D+00	Y3=	.208260D+00
T=	.041	Y1=	.550176D+00	Y2=	.371096D+00	Y3=	.207440D+00
T=	.042	Y1=	.541636D+00	Y2=	.377795D+00	Y3=	.206035D+00
T=	.043	Y1=	.533450D+00	Y2=	.384144D+00	Y3=	.204101D+00
T=	.044	Y1=	.525603D+00	Y2=	.390157D+00	Y3=	.201685D+00
T=	.045	Y1=	.518088D+00	Y2=	.395842D+00	Y3=	.198833D+00
T=	.046	Y1=	.510897D+00	Y2=	.401207D+00	Y3=	.195588D+00
T=	.047	Y1=	.504023D+00	Y2=	.406258D+00	Y3=	.191990D+00
T=	.048	Y1=	.497461D+00	Y2=	.411001D+00	Y3=	.188077D+00
T=	.049	Y1=	.491204D+00	Y2=	.415444D+00	Y3=	.183886D+00
T=	.050	Y1=	.485243D+00	Y2=	.419592D+00	Y3=	.179452D+00
T=	.051	Y1=	.479573D+00	Y2=	.423455D+00	Y3=	.174806D+00
T=	.052	Y1=	.474185D+00	Y2=	.427038D+00	Y3=	.169979D+00
T=	.053	Y1=	.469072D+00	Y2=	.430351D+00	Y3=	.165001D+00
T=	.054	Y1=	.464225D+00	Y2=	.433401D+00	Y3=	.159898D+00
T=	.055	Y1=	.459636D+00	Y2=	.436196D+00	Y3=	.154695D+00
T=	.056	Y1=	.455297D+00	Y2=	.438745D+00	Y3=	.149416D+00
T=	.057	Y1=	.451200D+00	Y2=	.441056D+00	Y3=	.144084D+00
T=	.058	Y1=	.447335D+00	Y2=	.443139D+00	Y3=	.138718D+00
T=	.059	Y1=	.443694D+00	Y2=	.445000D+00	Y3=	.133337D+00
T=	.060	Y1=	.440267D+00	Y2=	.446652D+00	Y3=	.127956D+00
T=	.061	Y1=	.437047D+00	Y2=	.448100D+00	Y3=	.122595D+00
T=	.062	Y1=	.434025D+00	Y2=	.449354D+00	Y3=	.117270D+00
T=	.063	Y1=	.431192D+00	Y2=	.450421D+00	Y3=	.111992D+00
T=	.064	Y1=	.428542D+00	Y2=	.451310D+00	Y3=	.106776D+00
T=	.065	Y1=	.426064D+00	Y2=	.452030D+00	Y3=	.101633D+00
T=	.066	Y1=	.423751D+00	Y2=	.452588D+00	Y3=	.965717D-01
T=	.067	Y1=	.421594D+00	Y2=	.452994D+00	Y3=	.916029D-01
T=	.068	Y1=	.419586D+00	Y2=	.453254D+00	Y3=	.867343D-01
T=	.069	Y1=	.417719D+00	Y2=	.453378D+00	Y3=	.819732D-01

```
T=  .070  Y1=  .415985D+00  Y2=  .453371D+00  Y3=  .773259D-01
T=  .071  Y1=  .414377D+00  Y2=  .453243D+00  Y3=  .727981D-01
T=  .072  Y1=  .412887D+00  Y2=  .452999D+00  Y3=  .683945D-01
T=  .073  Y1=  .411509D+00  Y2=  .452647D+00  Y3=  .641190D-01
T=  .074  Y1=  .410235D+00  Y2=  .452195D+00  Y3=  .599750D-01
T=  .075  Y1=  .409059D+00  Y2=  .451648D+00  Y3=  .559651D-01
T=  .076  Y1=  .407974D+00  Y2=  .451013D+00  Y3=  .520913D-01
T=  .077  Y1=  .406974D+00  Y2=  .450296D+00  Y3=  .483551D-01
T=  .078  Y1=  .406054D+00  Y2=  .449504D+00  Y3=  .447574D-01
T=  .079  Y1=  .405207D+00  Y2=  .448641D+00  Y3=  .412987D-01
T=  .080  Y1=  .404428D+00  Y2=  .447714D+00  Y3=  .379085D-01
T=  .081  Y1=  .403713D+00  Y2=  .446727D+00  Y3=  .347042D-01
T=  .082  Y1=  .403056D+00  Y2=  .445685D+00  Y3=  .314488D-01
T=  .083  Y1=  .402451D+00  Y2=  .444594D+00  Y3=  .287349D-01
T=  .084  Y1=  .401895D+00  Y2=  .443458D+00  Y3=  .259589D-01
T=  .085  Y1=  .401382D+00  Y2=  .442282D+00  Y3=  .233178D-01
T=  .086  Y1=  .400908D+00  Y2=  .441070D+00  Y3=  .208089D-01
T=  .087  Y1=  .400469D+00  Y2=  .439827D+00  Y3=  .184296D-01
T=  .088  Y1=  .400062D+00  Y2=  .438555D+00  Y3=  .161769D-01
T=  .089  Y1=  .399682D+00  Y2=  .437260D+00  Y3=  .140479D-01
T=  .090  Y1=  .399326D+00  Y2=  .435943D+00  Y3=  .120397D-01
T=  .091  Y1=  .398992D+00  Y2=  .434608D+00  Y3=  .101491D-01
T=  .092  Y1=  .398676D+00  Y2=  .433259D+00  Y3=  .837273D-02
T=  .093  Y1=  .398375D+00  Y2=  .431898D+00  Y3=  .670731D-02
T=  .094  Y1=  .398086D+00  Y2=  .430528D+00  Y3=  .514940D-02
T=  .095  Y1=  .397807D+00  Y2=  .429151D+00  Y3=  .369550D-02
T=  .096  Y1=  .397536D+00  Y2=  .427770D+00  Y3=  .234209D-02
T=  .097  Y1=  .397270D+00  Y2=  .426387D+00  Y3=  .108569D-02
T=  .098  Y1=  .397008D+00  Y2=  .425003D+00  Y3= -.770904D-04
T=  .099  Y1=  .396748D+00  Y2=  .423621D+00  Y3= -.115094D-02
T=  .100  Y1=  .396487D+00  Y2=  .422242D+00  Y3= -.213904D-02
```

本问题的精确解为

$$
\begin{cases}
y_1(t) = \dfrac{1}{2}e^{-2t} + \dfrac{1}{2}e^{-40t}(\cos 40t + \sin 40t) \\
y_2(t) = \dfrac{1}{2}e^{-2t} + \dfrac{1}{2}e^{-40t}(\cos 40t + \sin 40t) \\
y_3(t) = -e^{-40t}(\cos 40t - \sin 40t)
\end{cases}
$$

7.15 全区间积分的特雷纳方法

一、功能

用特雷纳(Treanor)方法对一阶刚性(stiff)微分方程组进行全区间积分。

二、方法说明

同 7.14 节的方法说明

三、子程序语句

SUBROUTINE GTNR2 (A,H,N,Y,M,W,F,D,P,Z,G,R)

四、形参说明

A——双精度实型变量,输入参数。积分的起始点值。

H——双精度实型变量,输入参数。积分的步长。

N——整型变量,输入参数。积分的步数(包括起始点这一步)。

Y——双精度实型一维数组,长度为M,输入参数,存放起始点 A 处的 M 个未知函数值 $y_i(A)$ $(i=1, 2, \cdots, M)$。返回时其值在 $W(i, 1)$ $(i=1, 2, \cdots, M)$中。

M——整型变量,输入参数。方程组中方程的个数,也是未知函数的个数。

W——双精度实型二维数组,体积为 M×N,输出参数。存放 N 个积分点上的 M 个未知函数值,即

$$W(i, j) = y_{ij}, \; i = 1, 2, \cdots, M; \; j = 1, 2, \cdots, N$$

其中 $W(i, 1)$ $(i=1, 2, \cdots, M)$为初值。

F——子程序名,输入参数。用于计算方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE F(T,Y,M,D)

其中,T 为双精度实型变量,自变量值;Y 为双精度实型一维数组,长度为 M,存放 M 个未知函数值 $y_i(T)$ $(i=1, 2, \cdots, M)$;D 为双精度实型一维数组,长度为 M,返回方程组中 M 个方程的右端函数值 $f_i(T, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

D——双精度实型一维数组,长度为 M。本子程序的工作数组,主要用于存放方程组中各方程的右端函数值 $f_i(t, y_1, y_2, \cdots, y_m)$ $(i=1, 2, \cdots, M)$。

P——双精度实型一维数组,长度为 M。本子程序的工作数组。

Z,G,R——均为双精度实型二维数组,体积为 4×M。本子程序的工作数组。

五、子程序(文件名:GTNR2.FOR)

```
      SUBROUTINE GTNR2(A,H,N,Y,M,W,F,D,P,Z,G,R)
```

```
      DIMENSION Y(M),D(M),W(M,N),Z(4,M),G(4,M)
      DIMENSION P(M),R(4,M)
      DOUBLE PRECISION Y,D,W,Z,G,P,R,A,H,T,S,AA,BB,DD,Q,DY,DY1
      DO 10 J=1,M
 10   W(J,1)=Y(J)
      DO 200 I=2,N
        T=A+(I-2)*H
        DO 50 J=1,M
 50     Z(1,J)=Y(J)
        CALL F(T,Y,M,D)
        DO 60 J=1,M
          G(1,J)=D(J)
          Y(J)=Z(1,J)+H*D(J)/2.0
          Z(2,J)=Y(J)
 60     CONTINUE
        S=T+H/2.0
        CALL F(S,Y,M,D)
        DO 70 J=1,M
          G(2,J)=D(J)
          Y(J)=Z(1,J)+H*D(J)/2.0
          Z(3,J)=Y(J)
 70     CONTINUE
        CALL F(S,Y,M,D)
        DO 80 J=1,M
 80     G(3,J)=D(J)
        DO 90 J=1,M
          AA=G(3,J)-G(2,J)
          BB=Z(3,J)-Z(2,J)
          IF (-AA*BB*H.GT.0.0) THEN
            P(J)=-AA/BB
            DD=-P(J)*H
            R(1,J)=EXP(DD)
            R(2,J)=(R(1,J)-1)/DD
            R(3,J)=(R(2,J)-1)/DD
            R(4,J)=(R(3,J)-1)/DD
          ELSE
            P(J)=0.0
          END IF
          IF (P(J).LE.0.0) THEN
            Q=G(3,J)
          ELSE
            Q=2*(G(3,J)-G(1,J))*R(3,J)
```

```
      Q=Q+(G(1,J)-G(2,J))*R(2,J)+G(2,J)
     END IF
     Z(4,J)=Z(1,J)+Q*H
     Y(J)=Z(4,J)
90   CONTINUE
     S=T+H
     CALL F(S,Y,M,D)
     DO 100 J=1,M
100  G(4,J)=D(J)
     DO 110 J=1,M
      IF (P(J).LE.0.0) THEN
        DY=(G(1,J)+2*(G(2,J)+G(3,J))+G(4,J))*H/6.0
      ELSE
        DY=-3*(G(1,J)+P(J)*Z(1,J))+2*(G(2,J)+P(J)*Z(2,J))
        DY=DY+2*(G(3,J)+P(J)*Z(3,J))-(G(4,J)+P(J)*Z(4,J))
        DY=DY*R(3,J)+G(1,J)*R(2,J)
        DY1=G(1,J)-G(2,J)-G(3,J)+G(4,J)
        DY1=DY1+(Z(1,J)-Z(2,J)-Z(3,J)+Z(4,J))*P(J)
        DY=(DY+4*DY1*R(4,J))*H
      END IF
      Y(J)=Z(1,J)+DY
      W(J,I)=Y(J)
110   CONTINUE
200  CONTINUE
     RETURN
     END
```

六、例

设一阶刚性微分方程组为

$$\begin{cases} y_1' = -21y_1 + 19y_2 - 20y_3, & y_1(0) = 1.0 \\ y_2' = 19y_1 - 21y_2 + 20y_3, & y_2(0) = 0.0 \\ y_3' = 40y_1 - 40y_2 - 40y_3, & y_3(0) = -1.0 \end{cases}$$

用全区间积分的特雷纳方法计算当 $h=0.001$ 各积分点

$$t_i = (i-1)h,\ i = 1, 2, \cdots, 101$$

上的未知函数值 $y_1(t_i)$, $y_2(t_i)$, $y_3(t_i)$ $(i=1, 2, \cdots, 101)$。

主程序及计算方程组中各方程右端函数值的子程序(文件名:GTNR20.FOR)为

```
      EXTERNAL F
      DIMENSION Y(3),D(3),W(3,101),P(3),Z(4,3),G(4,3),R(4,3)
      DOUBLE PRECISION Y,D,W,A,H,T,P,Z,G,R
      DATA Y/1.0,0.0,-1.0/
      A=0.0
```

```
      H=0.001
      N=101
      M=3
      CALL GTNR2(A,H,N,Y,M,W,F,D,P,Z,G,R)
      WRITE(*,*)
      DO 10 I=1,N,10
        T=(I-1)*H
        WRITE(*,20) T,W(1,I),W(2,I),W(3,I)
        WRITE(*,*)
10    CONTINUE
20    FORMAT(1X,'T=',F6.3,3X,'Y1=',D13.6,3X,'Y2=',D13.6,
     *       3X,'Y3=',D13.6)
      END
      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=-21*Y(1)+19*Y(2)-20*Y(3)
      D(2)=19*Y(1)-21*Y(2)+20*Y(3)
      D(3)=40*Y(1)-40*Y(2)-40*Y(3)
      RETURN
      END
```

部分运行结果为

```
T=  .000  Y1=  .100000D+01  Y2=  .000000D+00  Y3= -.100000D+01
T=  .010  Y1=  .929229D+00  Y2=  .509707D-01  Y3= -.354333D+00
T=  .020  Y1=  .797831D+00  Y2=  .162959D+00  Y3=  .108605D 01
T=  .030  Y1=  .665430D+00  Y2=  .276335D+00  Y3=  .173648D+00
T=  .040  Y1=  .559130D+00  Y2=  .363987D+00  Y3=  .208260D+00
T=  .050  Y1=  .485243D+00  Y2=  .419592D+00  Y3=  .179452D+00
T=  .060  Y1=  .440267D+00  Y2=  .446652D+00  Y3=  .127956D+00
T=  .070  Y1=  .415985D+00  Y2=  .453371D+00  Y3=  .773259D-01
T=  .080  Y1=  .404428D+00  Y2=  .447714D+00  Y3=  .379085D-01
T=  .090  Y1=  .399326D+00  Y2=  .435943D+00  Y3=  .120397D-01
T=  .100  Y1=  .396487D+00  Y2=  .422242D+00  Y3= -.213904D-02
```

本问题的精确解为

$$
\begin{cases}
y_1(t) = \dfrac{1}{2}e^{-2t} + \dfrac{1}{2}e^{-40t}(\cos 40t + \sin 40t) \\
y_2(t) = \dfrac{1}{2}e^{-2t} + \dfrac{1}{2}e^{-40t}(\cos 40t + \sin 40t) \\
y_3(t) = -e^{-40t}(\cos 40t - \sin 40t)
\end{cases}
$$

7.16 积分刚性方程组的吉尔方法

一、功能

用吉尔(Gear)方法积分一阶刚性微分方程组。本子程序也能积分非刚性方程组。

二、方法说明

设一阶微分方程组为

$$y_i' = f_i(t, y_1, y_2, \cdots, y_m), y_i(t_0) = y_{i0}$$
$$i = 1, 2, \cdots, m$$

吉尔方法的计算公式如下：

预报 $$Z_{n,(0)} = PZ_{n-1}$$

校正 $$Z_{n,(k+1)} = Z_{n,(k)} - L\left[\left[l_1 I - l_0 \frac{\partial F}{\partial Y}\right]^{-1} G(Z_{n,(k)})\right]^T$$

终值 $$Z_n = Z_{n,(M)}$$

其中

$$Z_n = (Y_n, hY_n', \cdots, h^p Y_n^{(p)}/p!)^T$$

为 $(p+1) \times m$ 阶的矩阵；

$$G(Z_{n,(k)}) = hF(t_n, Y_{n,(k)}) - hY'_{n,(k)}$$

为 m 维列向量；$\frac{\partial F}{\partial Y}$ 为 $m \times m$ 阶的雅可比(Jacobi)矩阵

$$\frac{\partial F}{\partial Y} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots & \frac{\partial f_1}{\partial y_m} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \cdots & \frac{\partial f_2}{\partial y_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial y_1} & \frac{\partial f_m}{\partial y_2} & \cdots & \frac{\partial f_m}{\partial y_m} \end{bmatrix}$$

P 为$(p+1)$阶的巴斯卡尔(Pascal)三角矩阵

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ & 1 & 2 & 3 & \cdots & k-1 & k \\ & & 1 & 3 & \cdots & k-1 & k \\ & & & 1 & \cdots & k-1 & k \\ & 0 & & & \ddots & \vdots & \vdots \\ & & & & & 1 & k \\ & & & & & & 1 \end{bmatrix}$$

L 为

$$L = (l_0, l_1, \cdots, l_p)^T$$

在本子程序中，$M = 3$(即迭代校正 3 次)。向量 L 在本子程序中自带。

上述计算过程是自开始、自动变步长且自动变阶的。

在本子程序中，为了满足精度要求，首先考虑减小步长，只有当达到最小步长但还没有满足精度要求时，才考虑提高方法的阶数，并且在用某个阶数的方法连续进行几步的计算中均满足精度要求时，考虑降低方法的阶数。

三、子程序语句

SUBROUTINE GGEAR(A,B,H,HMIN,HMAX,EPS,M,N,Y,T,Z,KF,F,JACOBI,D,P,S,S0Z,YM,ER,IIS,JIS)

四、形参说明

A,B——均为双精度实型变量，输入参数。积分区间的左端点与右端点。

H——双精度实型变量，输入参数。积分的拟定步长，在积分过程中将自动放大或缩小。要求 HMIN<<H<HMAX。

HMIN,HMAX——均为双精度实型变量，输入参数。积分过程中允许的最小步长与最大步长。

EPS——实型变量，输入参数。误差检验常数。

M——整型变量，输入参数。方程组中方程的个数，也是未知函数的个数。

N——整型变量，输入参数。拟定输出点数。

Y——双精度实型二维数组，体积为 M×8，输入参数。其中 $Y(i,1)(i=1,2,\cdots,M)$ 存放起始点 A 处的 M 个未知函数值 $y_i(A)$ $(i=1,2,\cdots,M)$；$Y(i,j)(i=1,2,\cdots,M;j=2,3,\cdots,8)$ 为本子程序的工作单元，返回时，初值在 $Z(i,1)(i=1,2,\cdots,M)$ 中。

T——双精度实型一维数组，长度为 N，输出参数。存放各输出点的自变量值(包括起始点 A)。

Z——双精度实型二维数组，体积为 M×N，输出参数。存放 N 个输出点处的 M 个未知函数值，即

$$Z(i,j)=y_i(t_j),i=1,2,\cdots,M;j=1,2,\cdots,N$$

其 $Z(i,1)(i=1,2,\cdots,M)$ 为初值。

KF——整型变量，输出参数。其中：

KF=+1　表示全区间积分成功。

KF=−1　表示步长已小于最小步长 HMIN，但精度仍未达到，积分停止(前若干点输出有效)。

FK=−2　表示阶数大于 6，积分停止(前若干点输出有效)。

FK=−3　表示对于 H≥HMIN 校正迭代不收敛，积分停止(前若干点输出有效)。

KF=−4　表示对所处理的问题，精度要求太高，积分停止(前若干点输出有效)。

F——子程序名，输入参数。用于计算方程组中各方程的右端函数值。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE F(T,Y,M,D)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未知函数值 $y_i(T)(i=1,2,\cdots,M)$；D 为双精度实型一维数组，长度为 M，返回方程组中 M

个方程的右端函数值 $f_i(T,y_1,y_2,\cdots,y_m)(i=1,2,\cdots,M)$。

JACOBI——子程序名，输入参数。用于计算雅可比矩阵

$$\begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots & \frac{\partial f_1}{\partial y_m} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \cdots & \frac{\partial f_2}{\partial y_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial y_1} & \frac{\partial f_m}{\partial y_2} & \cdots & \frac{\partial f_m}{\partial y_m} \end{bmatrix}$$

在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE JACOBI(T,Y,P,M)

其中：T 为双精度实型变量，自变量值；Y 为双精度实型一维数组，长度为 M，存放 M 个未知函数值 $y_i(T)(i=1,2,\cdots,M)$；P 为双精度实型二维数组，体积为 M×M，返回雅可比矩阵中各元素，即 $P(i,j)=\frac{\partial f_i}{\partial y_j}(i=1,2,\cdots,M;j=1,2,\cdots,M)$。

D——双精度实型一维数组，长度为 M。本子程序的工作数组，主要用于存放方程组中 M 个方程的右端函数值 $f_i(t,y_1,y_2,\cdots,y_m)(i=1,2,\cdots,M)$。

P——双精度实型二维数组，体积为 M×M。本子程序的工作数组，用于雅可比矩阵的处理。

S——双精度实型二维数组，体积为 M×10。本子程序的工作数组。

S02,YM,ER——均为双精度实型一维数组，长度为 M。本子程序的工作数组。

IIS,JJS——均为整型一维数组，长度为 M。本子程序的工作数组。

五、子程序(文件名:GGEAR.FOR)

```
      SUBROUTINE GGEAR(A,B,H,HMIN,HMAX,EPS,M,N,Y,T,Z,KF,
     *                 F,JACOBI,D,P,S,S02,YM,ER,IIS,JJS)
      DIMENSION T(N),Z(M,N),P(M,M),Y(M,8),D(M),S(M,10)
      DIMENSION YM(M),S02(M),ER(M),AA(7),PP(7,3),IIS(M),JJS(M)
      DOUBLE PRECISION T,Z,D,P,Y,S,YM,S02,ER,AA,PP,A,B,H,
     *     HMIN,HMAX,HW,HD,RM,T0,TD,R,DD,PR1,PR2,PR3,RR
      DATA PP/2.0,4.5,7.333,10.42,13.7,17.15,1.0,
     *        3.0,6.0,9.167,12.5,15.98,1.0,1.0,
     *        1.0,1.0,0.5,0.1667,0.04133,0.008267,1.0/
      AA(2)=-1.0
      JT=0
      NN=0
      KF=1
      T0=A
      NQ=1
      CALL F(T0,Y,M,D)
      DO 10 I=1,M
```

```
10    Y(I,2)=D(1)*H
      HW=H
      K=2
      DO 15 I=1,M
15    YM(I)=1.0
20    IRT=1
      KF=1
      NN=NN+1
      T(NN)=T0
      DO 30 I=1,M
30    Z(I,NN)=Y(I,1)
      IF (T0.GE.B) RETURN
      IF (NN.EQ.N) RETURN
      DO 40 I=1,M
      DO 40 J=1,K
40    S(I,J)=Y(I,J)
      HD=HW
      IF (H.NE.HD) THEN
        RM=H/HD
        IRT1=1
        CALL G75(HMIN,HD,RM,HMAX,Y,S,H,IDB,K,M)
      END IF
      NQD=NQ
      TD=T0
      RM=1.0
      IF (JT.GT.0) GOTO 80
60    IF (NQ.GT.6) THEN
        KF=-2
        RETURN
      END IF
      IF (NQ.EQ.1) THEN
        AA(1)=-1.0
      ELSE IF (NQ.EQ.2) THEN
        AA(1)=-2.0/3.0
        AA(3)=-1.0/3.0
      ELSE IF (NQ.EQ.3) THEN
        AA(1)=-6.0/11.0
        AA(3)=AA(1)
        AA(4)=-1.0/11.0
      ELSE IF (NQ.EQ.4) THEN
        AA(1)=-0.48
        AA(3)=-0.7
```

```
      AA(4)=-0.2
      AA(5)=-0.02
    ELSE IF (NQ.EQ.5) THEN
      AA(1)=-120.0/274.0
      AA(3)=-225.0/274.0
      AA(4)=-85.0/274.0
      AA(5)=-15.0/274.0
      AA(6)=-1.0/274.0
    ELSE
      AA(1)=-720.0/1764.0
      AA(3)=-1624.0/1764.0
      AA(4)=-735.0/1764.0
      AA(5)=-175.0/1764.0
      AA(6)=-21.0/1764.0
      AA(7)=-1.0/1764.0
    END IF
    K=NQ+1
    IDB=K
    ENQ2=0.5/(NQ+1.0)
    ENQ3=0.5/(NQ+2.0)
    ENQ1=0.5/(NQ+0.0)
    EUP=FP(NQ,2)*EPS
    EUP=EUP*EUP
    E=FP(NQ,1)*EPS
    E=E*E
    EDWN=FP(NQ,3)*EPS
    EDWN=EDWN*EDWN
    IF (EDWN.EQ.0.0) THEN
      KF=-4
      CALL G47(M,K,Y,S,H,HD,NQ,NQD,JT)
      RETURN
    END IF
    BND=EPS*ENQ3/(M+0.0)
    IW=1
    IF (IRT.EQ.2) THEN
      CALL G68(M,K,R,Y,YM,IDB,NQ,JT)
      GOTO 20
    END IF
80  T0=T0+H
    DO 110 J=2,K
      DO 100 J1=J,K
        J2=K-J1+J-1
```

```
        DO 90 I=1,M
90      Y(I,J2)=Y(I,J2)+Y(I,J2+1)
100   CONTINUE
110   CONTINUE
      DO 120 I=1,M
120   ER(I)=0.0
      J1=1
      NT=1
      DO 190 L=1,3
        IF ((J1.NE.0).AND.(NT.NE.0)) THEN
          CALL F(T0,Y,M,D)
          IF (IW.GE.1) THEN
            CALL JACOBI(T0,Y,P,M)
            R=AA(1)*H
            DO 130 I=1,M
            DO 130 J=1,M
130         P(I,J)=P(I,J)*R
            DO 140 I=1,M
140         P(I,I)=1.0+P(I,I)
            IW=-1
            CALL BRINV(P,M,J1,JIS,JJS)
          END IF
          IF (J1.GT.0) THEN
            DO 150 I=1,M
150         S02(I)=Y(I,2)-D(I)*H
            DO 170 I=1,M
              DD=0.0
              DO 160 J=1,M
160           DD=DD+S02(J)*P(I,J)
              S(I,9)=DD
170         CONTINUE
            NT=M
            DO 180 I=1,M
              Y(I,1)=Y(I,1)+AA(1)*S(I,9)
              Y(I,2)=Y(I,2)-S(I,9)
              ER(I)=ER(I)+S(I,9)
              IF (ABS(S(I,9)).LE.(EPS*YM(I))) NT=NT-1
180         CONTINUE
          END IF
        END IF
190   CONTINUE
      IF (NT.GT.0) THEN
```

```
      T0=TD
      IF ((H.GT.(HMIN * 1.00001)).OR.(IW.GE.0)) THEN
        IF (IW.NE.0) RM=0.25 * RM
        IW=1
        IRT1=2
        CALL G75(HMIN,HD,RM,HMAX,Y,S,H,IDB,K,M)
        GOTO 80
      END IF
      KF=-3
      CALL G47(M,K,Y,S,H,HD,NQ,NQD,JT)
      RETURN
    END IF
    DD=0.0
    DO 200 I=1,M
200 DD=DD+(ER(I)/YM(I)) * (ER(I)/YM(I))
    IW=0
    IF (DD.LE.E) THEN
      IF (K.GE.3) THEN
        DO 210 J=3,K
        DO 210 I=1,M
210     Y(I,J)=Y(I,J)+AA(J) * ER(I)
      END IF
      KF=1
      HW=H
      IF (IDB.GT.1) THEN
        IDB=IDB-1
        IF (IDB.LE.1) THEN
          DO 220 I=1,M
220       S(I,10)=ER(I)
        END IF
        DO 230 I=1,M
          IF (YM(I).LT.ABS(Y(I,1))) YM(I)=ABS(Y(I,1))
230     CONTINUE
        JT=NQ
        GOTO 20
      END IF
    END IF
    IF (DD.GT.E) THEN
      KF=KF-2
      IF (H.LE.(HMIN * 1.00001)) THEN
        KF=-1
        HW=H
```

```
          JT=NQ
          RETURN
        END IF
        T0=TD
        IF (KF.LE.-5) THEN
          IF (NQ.EQ.1) THEN
            KF=-4
            CALL G47(M,K,Y,S,H,HD,NQ,NQD,JT)
            RETURN
          END IF
          CALL F(T0,Y,M,D)
          R=H/HD
          DO 240 I=1,M
            Y(I,1)=S(I,1)
            S(I,2)=HD*D(I)
            Y(I,2)=S(I,2)*R
240       CONTINUE
          NQ=1
          KF=1
          GOTO 60
        END IF
      END IF
      PR2=((DD/E)**ENQ2)*1.2
      PR3=1.0E+20
      IF (NQ.LT.7) THEN
        IF (KF.GT.-1) THEN
          DD=0.0
          DO 250 I=1,M
            PR3=(ER(I)-S(I,10))/YM(I)
            DD=DD+PR3*PR3
250       CONTINUE
          PR3=((DD/EUP)**ENQ3)*1.4
        END IF
      END IF
      PR1=1.0E+20
      IF (NQ.GT.1) THEN
        DD=0.0
        DO 260 I=1,M
          PR1=Y(I,K)/YM(I)
          DD=DD+PR1*PR1
260     CONTINUE
        PR1=((DD/EDWN)**ENQ1)*1.3
```

```
      END IF
      IF (PR2.LE.PR3) THEN
        IF (PR2.GT.PR1) THEN
          R=1.0E+4
          IF (PR1.GT.1.0E-4) R=1.0/PR1
          NQW=NQ-1
        ELSE
          NQW=NQ
          R=1.0E+4
          IF (PR2.GT.1.0E-4) R=1.0/PR2
        END IF
      ELSE IF (PR3.LT.PR1) THEN
        R=1.0E+4
        IF (PR3.GT.1.0E-4) R=1.0/PR3
        NQW=NQ+1
      ELSE
        R=1.0E+4
        IF (PR1.GT.1.0E-4) R=1.0/PR1
        NQW=NQ-1
      END IF
      IDB=10
      IF (KF.EQ.1) THEN
        IF (R.LT.1.1) THEN
          DO 270 I=1,M
            IF (YM(I).LT.ABS(Y(I,1))) YM(I)=ABS(Y(I,1))
270       CONTINUE
          JT=NQ
          GOTO 20
        END IF
      END IF
      IF (NQW.GT.NQ) THEN
        DO 280 I=1,M
280     Y(I,NQW+1)=ER(I)*AA(K)/(K+0.0)
      END IF
      K=NQW+1
      IF (KF.EQ.1) THEN
        IRT=2
        RR=HMAX/ABS(H)
        IF (R.GT.RR) R=RR
        H=H*R
        HW=H
        IF (NQ.EQ.NQW) THEN
```

```
    CALL G68(M,K,R,Y,YM,IDB,NQ,JT)
    GOTO 20
   END IF
   NQ=NQW
   GOTO 60
  END IF
  RM=RM * R
  IRT1=3
  CALL G75(HMIN,HD,RM,HMAX,Y,S,H,IDB,K,M)
  IF (NQW.EQ.NQ) GOTO 80
  NQ=NQW
  GOTO 60
  END

  SUBROUTINE G75(HMIN,HD,RM,HMAX,Y,S,H,IDB,K,M)
  DIMENSION Y(M,8),S(M,10)
  DOUBLE PRECISION HMIN,HMAX,H,RM,Y,S,HD,R,RR
  RR=ABS(HMIN/HD)
  IF (RM.LT.RR) RM=RR
  RR=ABS(HMAX/HD)
  IF (RM.GT.RR) RM=RR
  R=1.0
  DO 20 J=2,K
   R=R * RM
   DO 10 I=1,M
10  Y(I,J)=S(I,J) * R
20 CONTINUE
  H=HD * RM
  DO 30 I=1,M
30 Y(I,1)=S(I,1)
  IDB=K
  RETURN
  END

  SUBROUTINE G47(M,K,Y,S,H,HD,NQ,NQD,JT)
  DIMENSION Y(M,8),S(M,10)
  DOUBLE PRECISION Y,S,H,HD
  DO 10 I=1,M
  DO 10 J=1,K
10 Y(I,J)=S(I,J)
  H=HD
  NQ=NQD
  JT=NQ
```

```
      RETURN
      END
      SUBROUTINE G68(M,K,R,Y,YM,IDB,NQ,JT)
      DIMENSION Y(M,8),YM(M)
      DOUBLE PRECISION Y,YM,R,R1
      R1=1.0
      DO 20 J=2,K
        R1=R1*R
        DO 10 I=1,M
10      Y(I,J)=Y(I,J)*R1
20    CONTINUE
      IDB=K
      DO 30 I=1,M
        IF (YM(I).LT.ABS(Y(I,1))) YM(I)=ABS(Y(I,1))
30    CONTINUE
      JT=NQ
      RETURN
      END
```

六、例

设一阶刚性微分方程组为

$$\begin{cases} y_1' = -21y_1 + 19y_2 - 20y_3, & y_1(0) = 1.0 \\ y_2' = 19y_1 - 21y_2 + 20y_3, & y_2(0) = 0.0 \\ y_3' = 40y_1 - 40y_2 - 40y_3, & y_3(0) = -1.0 \end{cases}$$

用吉尔方法求在区间[0,1]上的数值解。其中 A=0.0,B=1.0,M=3。取 N=30。H,HMIN,HMAX,EPS 由键盘输入。

主程序及计算方程组中各方程的右端函数值、雅可比矩阵的子程序(文件名:GGEAR0.FOR)为

```
      EXTERNAL F,JACOBI
      DIMENSION Y(3,8),S(3,10),YM(3),D(3),P(3,3),Z(3,30)
      DIMENSION S02(3),ER(3),T(30),IIS(3),JJS(3)
      DOUBLE PRECISION Y,S,YM,D,P,Z,S02,ER,T,A,B,H,HMIN,HMAX
      DATA Y/1.0,0.0,-1.0,21*0.0/
      A=0.0
      B=1.0
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(1X,'H,HMIN,HMAX,EPS=?')
      READ(*,*) H,HMIN,HMAX,EPS
      WRITE(*,*)
      N=30
```

```
      M=3
      CALL GGEAR(A,B,H,HMIN,HMAX,EPS,M,N,Y,T,Z,KF,
     *           F,JACOBI,D,P,S,S02,YM,ER,IIS,JJS)
      WRITE(*,20) KF
20    FORMAT(1X,'KF=',I4)
      WRITE(*,*)
      WRITE(*,30)
30    FORMAT(7X,'T',14X,'Y(1)',11X,'Y(2)',11X,'Y(3)')
      DO 40 I=1,N
        WRITE(*,50) T(I),Z(1,I),Z(2,I),Z(3,I)
40    CONTINUE
50    FORMAT(1X,F10.6,5X,3D15.6)
      WRITE(*,*)
      END

      SUBROUTINE F(T,Y,M,D)
      DIMENSION Y(M),D(M)
      DOUBLE PRECISION Y,D,T
      D(1)=-21.0*Y(1)+19.0*Y(2)-20.0*Y(3)
      D(2)=19.0*Y(1)-21.0*Y(2)+20.0*Y(3)
      D(3)=40.0*Y(1)-40.0*Y(2)-40.0*Y(3)
      RETURN
      END

      SUBROUTINE JACOBI(T,Y,P,M)
      DIMENSION Y(M),P(M,M)
      DOUBLE PRECISION Y,P,T
      P(1,1)=-21.0
      P(1,2)=19.0
      P(1,3)=-20.0
      P(2,1)=19.0
      P(2,2)=-21.0
      P(2,3)=20.0
      P(3,1)=40.0
      P(3,2)=-40.0
      P(3,3)=-40.0
      RETURN
      END
```

运行结果为

(对应四组不同的 H,HMIN,HMAX,EPS 值)

```
H,HMIN,HMAX,EPS=?
0.01,0.0001,0.1,0.001
```

KF= 1

T	Y(1)	Y(2)	Y(3)
.000000	.100000D+01	.000000D+00	−.100000D+01
.000690	.998591D+00	.315899D−04	−.946338D+00
.001380	.996502D+00	.744849D−03	−.894192D+00
.003234	.988675D+00	.488066D−02	−.759460D+00
.005088	.976732D+00	.131464D−01	−.635490D+00
.006942	.961290D+00	.249243D−01	−.522061D+00
.010099	.928377D+00	.516304D−01	−.352411D+00
.013256	.889586D+00	.842532D−01	−.210762D+00
.016413	.847441D+00	.120268D+00	−.948459D−01
.019569	.803844D+00	.157775D+00	−.218473D−02
.023568	.748863D+00	.205096D+00	.858297D−01
.027567	.696222D+00	.250139D+00	.145945D+00
.031565	.647232D+00	.291531D+00	.183266D+00
.035564	.602954D+00	.328392D+00	.202440D+00
.039562	.563735D+00	.360192D+00	.207577D+00
.045876	.512593D+00	.399741D+00	.195664D+00
.052190	.473985D+00	.426901D+00	.169112D+00
.058503	.446209D+00	.443372D+00	.136401D+00
.064817	.427137D+00	.451281D+00	.10[illegible]212D+00
.071131	.414653D+00	.452743D+00	.730871D−01
.078731	.405714D+00	.448597D+00	.432409D−01
.086331	.400860D+00	.440564D+00	.211851D−01
.093931	.398047D+00	.430683D+00	.631603D−02
.101532	.395908D+00	.420320D+00	−.265137D−02
.109132	.393628D+00	.410287D+00	−.719698D−02
.119089	.389743D+00	.398321D+00	−.872038D−02
.129046	.384662D+00	.387864D+00	−.757989D−02
.139003	.378578D+00	.378716D+00	.552278D−02
.148960	.371820D+00	.370543D+00	−.349394D−02
.158917	.364699D+00	.363027D+00	.188095D−02

H,HMIN.HMAX.EPS=?

0.01,0.0001,0.1,0.0001

KF= 1

T	Y(1)	Y(2)	Y(3)
.000000	.100000D+01	.000000D+00	−.100000D+01
.000218	.999707D+00	−.143222D−03	−.982702D+00
.000436	.999341D+00	−.212843D−03	−.965556D+00
.001286	.997381D+00	.496717D−04	−.899883D+00

.002137	.994389D+00	.134717D−02	−.836513D+00
.002987	.990432D+00	.361242D−02	−.775437D+00
.004397	.981900D+00	.934551D−02	−.679183D+00
.005806	.971188D+00	.172665D−01	−.589113D+00
.007216	.958607D+00	.270648D−01	−.505108D+00
.008626	.944427D+00	.384696D−01	−.427027D+00
.010509	.923405D+00	.557956D−01	−.331659D+00
.012393	.900503D+00	.750160D−01	−.246140D+00
.014276	.876207D+00	.956413D−01	−.169995D+00
.016160	.850950D+00	.117247D+00	−.102715D+00
.019069	.810926D+00	.151654D+00	−.149844D−01
.021978	.770725D+00	.186271D+00	.548703D−01
.024887	.731274D+00	.220170D+00	.108895D+00
.027797	.693289D+00	.252635D+00	.149087D+00
.030706	.657317D+00	.283120D+00	.177348D+00
.034273	.616516D+00	.317232D+00	.198334D+00
.037840	.579765D+00	.347348D+00	.207108D+00
.041407	.547237D+00	.373285D+00	.206393D+00
.044975	.518920D+00	.395058D+00	.198575D+00
.048542	.494658D+00	.412823D+00	.185694D+00
.052901	.470134D+00	.429469D+00	.165550D+00
.057260	.450670D+00	.441124D+00	.142891D+00
.061619	.435577D+00	.448476D+00	.119615D+00
.065979	.424148D+00	.452231D+00	.970890D−01
.070338	.415701D+00	.453070D+00	.762414D−01
.075490	.408704D+00	.451161D+00	.545202D−01

H,HMIN,HMAX,EPS=?

0.01,0.00001,0.1,0.0001

KF= 1

T	Y(1)	Y(2)	Y(3)
.000000	.100000D+01	.000000D+00	−.100000D+01
.000218	.999707D+00	−.143222D−03	−.982702D+00
.000436	.999341D+00	−.212843D−03	−.965556D+00
.001286	.997381D+00	.496717D−04	−.899883D+00
.002137	.994389D+00	.134717D−02	−.836513D+00
.002987	.990432D+00	.361242D−02	−.775437D+00
.004397	.981900D+00	.934551D−02	−.679183D+00
.005806	.971188D+00	.172665D−01	−.589113D+00
.007216	.958607D+00	.270648D−01	−.505108D+00
.008626	.944427D+00	.384696D−01	−.427027D+00

.010509	.923405D+00	.557956D−01	−.331659D+00
.012393	.900503D+00	.750160D−01	−.246140D+00
.014276	.876207D+00	.956443D−01	−.169995D+00
.016160	.850950D+00	.117247D+00	−.102715D+00
.019069	.810926D+00	.151654D+00	−.149844D−01
.021978	.770725D+00	.186271D+00	.548703D−01
.024887	.731274D+00	.220170D+00	.108895D+00
.027797	.693289D+00	.252635D+00	.149087D+00
.030706	.657317D+00	.283120D+00	.177348D+00
.034273	.616518D+00	.317232D+00	.198334D+00
.037840	.579765D+00	.347348D+00	.207108D+00
.041407	.547237D+00	.373285D+00	.206393D+00
.044975	.518920D+00	.395058D+00	.198575D+00
.048542	.494658D+00	.412823D+00	.185694D+00
.052901	.470134D+00	.429469D+00	.165650D+00
.057260	.450670D+00	.441124D+00	.142891D+00
.061619	.435577D+00	.448476D+00	.119615D+00
.065979	.424148D+00	.452231D+00	.970890D−01
.070338	.415701D+00	.453070D+00	.762414D−01
.075490	.408704D+00	.451161D+00	.545202D−01

H,HMIN,HMAX.EPS=?

0.01,0.00001,0.1,0.00001

KF= 1

T	Y(1)	Y(2)	Y(3)
.000000	.100000D+01	.000000D+00	−.100000D+01
.000069	.999923D+00	−.613974D−04	−.994497D+00
.000138	.999839D+00	−.115256D−03	−.989009D+00
.000531	.999241D+00	−.302322D−03	−.957984D+00
.000924	.998407D+00	−.253065D−03	−.927454D+00
.001317	.997344D+00	.255856D−04	−.897417D+00
.001956	.995144D+00	.951994D−03	−.849651D+00
.002595	.992368D+00	.243622D−02	−.803183D+00
.003234	.989108D+00	.444563D−02	−.758009D+00
.003872	.985335D+00	.695039D−02	−.714120D+00
.004810	.978964D+00	.114625D−01	−.652041D+00
.005747	.971679D+00	.168931D−01	−.592682D+00
.006684	.963562D+00	.231585D−01	−.536005D+00
.007621	.954694D+00	.301793D−01	−.481972D+00
.008414	.935907D+00	.454416D−01	−.385808D+00
.011207	.915141D+00	.626946D−01	−.298804D+00
.012999	.892843D+00	.814927D−01	−.220566D+00

.014792	.869413D+00	.101436D+00	−.150669D+00
.016585	.845210D+00	.122164D+00	−.886666D−01
.018378	.820554D+00	.143358D+00	−.340891D−01
.020171	.795725D+00	.164737D+00	.135040D−01
.021963	.770970D+00	.186055D+00	.546174D−01
.023756	.746499D+00	.207100D+00	.897148D−01
.025549	.722493D+00	.227693D+00	.119264D+00
.027342	.699104D+00	.247681D+00	.143724D+00
.029134	.676459D+00	.266938D+00	.163540D+00
.030927	.654658D+00	.285362D+00	.179142D+00
.032720	.633784D+00	.302872D+00	.190946D+00
.034513	.613895D+00	.319408D+00	.199344D+00
.037424	.583809D+00	.344076D+00	.206683D+00

本问题的精确解为

$$\begin{cases} y_1(t) = \dfrac{1}{2}e^{-2t} + \dfrac{1}{2}e^{-40t}(\cos 40t + \sin 40t) \\ y_2(t) = \dfrac{1}{2}e^{-2t} - \dfrac{1}{2}e^{-40t}(\cos 40t + \sin 40t) \\ y_3(t) = -e^{-40t}(\cos 40t - \sin 40t) \end{cases}$$

七、附注

本子程序需要调用全选主元矩阵求逆子程序 BRINV，参看 2.3 节。

7.17 二阶微分方程边值问题的数值解法

一、功能

用有限差分法求二阶线性微分方程边值问题的数值解。

二、方法说明

设二阶线性微分方程的边值问题为

$$\begin{cases} u(x)y'' + v(x)y' + w(x)y = f(x) \\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

现要求未知函数 $y(x)$ 在区间$[a,b]$上的 n 个等距离散点上的近似解，其中 n 个等距离散点为

$$x_i = a + (i-1)h, i = 1,2,\cdots,n$$

$$h = (b-a)/(n-1)$$

显然，$y(x_1) = y(a) = \alpha, y(x_n) = y(b) = \beta$。

用中心差分近似代替 $y''(x)$ 与 $y'(x)$，即

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}$$

$$y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

其中 $y_i = y(x_i)$。将它们代入原微分方程得

$$\frac{u(x_i)}{h^2}(y_{i+1} - 2y_i + y_{i-1}) + \frac{v(x_i)}{2h}(y_{i+1} - y_{i-1}) + w(x_i)y_i = f(x_i)$$

$$i = 2,3,\cdots,n-1$$

经整理后可得到如下关于 $y_i(i = 1,2,\cdots,n)$ 的方程组：

$$\begin{cases} y_1 = \alpha \\ A_i y_{i-1} + B_i y_i + C_i y_{i+1} = D_i, i = 2,3,\cdots,n-1 \\ y_n = \beta \end{cases}$$

其中

$$\begin{cases} A_i = u(x_i) - \dfrac{h}{2}v(x_i) \\ B_i = h^2 w(x_i) - 2u(x_i) \\ C_i = u(x_i) + \dfrac{h}{2}v(x_i) \\ D_i = h^2 f(x_i) \end{cases}$$

用矩阵表示为

$$\begin{bmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & A_n & B_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_{n-1} \\ D_n \end{bmatrix}$$

其中 $B_1 = 1, C_1 = 0, D_1 = \alpha; A_n = 0, B_n = 1, D_n = \beta$。

此为三对角线方程组，可用追赶法求解。

在实际计算时，为了提高精度，采用如下方法：

用步长 $h = (b-a)/(n-1)$ 计算得到 $n-2$ 个等距离散点上的一组近似解

$$y_i^{(h)} = y_i(a + (i-1)h), i = 2,3,\cdots,n-1$$

再用步长 $\frac{h}{2} = (b-a)/(2(n-1))$ 计算得到 $n-2$ 个等距离散点上的另一组近似解

$$y_i^{(h/2)} = y_i(a + (i-1)h), i = 2,3,\cdots,n-1$$

最后令

$$y_i = \frac{4y_i^{(h/2)} - y_i^{(h)}}{3}, i = 2,3,\cdots,n-1$$

三、子程序语句

SUBROUTINE GDFTE(A,B,Y0,YN,N,Y,FS,N2,N6,G,D)

四、形参说明

A,B——均为双精度实型变量，输入参数。求解区间的左端点与右端点值。

Y0,YN——均为双精度实型变量,输入参数。未知函数在求解区间左、右端点处的函数值 y(A)与 y(B)。

N——整型变量,输入参数。求解区间[A,B]上的等分点数,即各等分点为

$$h = (B - A)/(N - 1)$$

$$x_i = A + (i - 1)h, i = 1,2,\cdots,N$$

Y——双精度实型一维数组,长度为N,输出参数。存放N个等分点上的未知函数值,即

$$y_i = y(x_i), i = 1,2,\cdots,N$$

FS——子程序名,输入参数。用于计算微分方程

$$u(x)y'' + v(x)y' + w(x)y = f(x)$$

中的函数 $u(x)$,$v(x)$,$w(x)$,$f(x)$ 的值。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE FS(X,U,V,W,F)

其中:X,U,V,W,F 均为双精度实型变量,X 为自变量值,U,V,W,F 分别返回 $u(x)$,$v(x)$,$w(x)$,$f(x)$ 的值。

N2——整型变量,输入参数。要求 N2=2*N。

N6——整型变量,输入参数。要求 N6=6*N。

G——双精度实型一维数组,长度为 N6。本子程序的工作数组。

D——双精度实型一维数组,长度为 N2。本子程序的工作数组。

五、子程序(文件名:GDFTE.FOR)

```
      SUBROUTINE GDFTE(A,B,Y0,YN,N,Y,FS,N2,N6,G,D)
      DIMENSION Y(N),G(N6),D(N2)
      DOUBLE PRECISION Y,G,D,A,B,Y0,YN,H,X,U,V,W,F
      H=(B-A)/(N-1.0)
      NN=2*N-1
      G(1)=1.0
      G(2)=0.0
      Y(1)=Y0
      Y(N)=YN
      G(3*N-2)=1.0
      G(3*N-3)=0.0
      DO 10 I=2,N-1
        X=A+(I-1)*H
        CALL FS(X,U,V,W,F)
        K=3*(I-1)-1
        G(K+1)=U-H*V/2.0
        G(K+2)=H*H*W-2.0*U
        G(K+3)=U+H*V/2.0
        Y(I)=H*H*F
10    CONTINUE
```

```
      M1=3*N-2
      CALL ATRDE(G,N,M1,Y,L)
      H=H/2.0
      G(1)=1.0
      G(2)=0.0
      D(1)=Y0
      D(NN)=YN
      G(3*NN-2)=1.0
      G(3*NN-3)=0.0
      DO 20 I=2,NN-1
        X=A+(I-1)*H
        CALL FS(X,U,V,W,F)
        K=3*(I-1)-1
        G(K+1)=U-H*V/2.0
        G(K+2)=H*H*W-2.0*U
        G(K+3)=U+H*V/2.0
        D(I)=H*H*F
20    CONTINUE
      M1=3*NN-2
      CALL ATRDE(G,NN,M1,D,L)
      DO 30 I=2,N-1
        K=2*I-1
        Y(I)=(4.0*D(K)-Y(I))/3.0
30    CONTINUE
      RETURN
      END
```

六、例

设二阶微分方程边值问题为

$$\begin{cases} -y'' + \dfrac{2}{x^2}y = \dfrac{1}{x} \\ y(2) = 0, y(3) = 0 \end{cases}$$

求解区间为[2,3]，等分点数为N=11(即步长为 $h = 0.1$)。其中 $u(x) = -1, v(x) = 0$, $w(x) = \dfrac{2}{x^2}, f(x)\dfrac{1}{x}$。

主程序及计算 $u(x), v(x), w(x), f(x)$ 的子程序(文件名:GDFTE0.FOR)为

```
      EXTERNAL FS
      DIMENSION Y(11),G(66),D(22)
      DOUBLE PRECISION Y,A,B,Y0,YN,G,D
      A=2.0
      B=3.0
      N=11
```

```
      N2=22
      N6=66
      Y0=0.0
      YN=0.0
      CALL GDFTE(A,B,Y0,YN,N,Y,FS,N2,N6,G,D)
      WRITE(*,*)
      WRITE(*,10) (I,Y(I),I=1,N)
10    FORMAT(1X,'Y(',I2,' )=',D15.6)
      WRITE(*,*)
      END
      SUBROUTINE FS(X,U,V,W,F)
      DOUBLE PRECISION X,U,V,W,F
      U=-1.0
      V=0.0
      W=2.0/(X*X)
      F=1.0/X
      RETURN
      END
```

运行结果为

```
Y( 1 )=    .000000D+00
Y( 2 )=    .186090D-01
Y( 3 )=    .325359D-01
Y( 4 )=    .420480D-01
Y( 5 )=    .473684D-01
Y( 6 )=    .486842D-01
Y( 7 )=    .461538D-01
Y( 8 )=    .399123D-01
Y( 9 )=    .300752D-01
Y(10 )=    .167423D-01
Y(11 )=    .000000D+00
```

本问题的精确解为

$$y(x) = \left(19x - 5x^3 - \frac{36}{x}\right)/38$$

七、附注

本子程序需要调用追赶法求解三对角线方程组的子程序 ATRDE，参看 1.5 节。

第8章 拟合与逼近

8.1 最小二乘曲线拟合

一、功能

用最小二乘法求 n 个数据点的拟合多项式。

二、方法说明

设已知 n 个数据点 $(x_i,y_i)(i=1,2,\cdots,n)$，要求 $(m-1)$ 次最小二乘拟合多项式

$$P_m(x)=a_1+a_2x+a_3x^2+\cdots+a_mx^{m-1}$$

其中 $m\leqslant n$ 且 $m\leqslant 20$。

设拟合多项式为各正交多项式 $Q_j(x)$ $(j=1,2,\cdots,m)$ 的线性组合，即

$$P_m(x)=q_1Q_1(x)+q_2Q_2(x)+\cdots+q_mQ_m(x)$$

其中 $Q_j(x)$ $(j=1,2,\cdots,m)$ 由下列递推公式构造：

$$Q_1(x)=1$$

$$Q_2(x)=(x-\alpha_1)$$

$$Q_{j+1}(x)=(x-\alpha_{j+1})Q_j(x)-\beta_jQ_{j-1}(x),\ j=2,3,\cdots,m-1$$

若设

$$d_j=\sum_{i=1}^{n}Q_j^2(x_i),\ j=1,2,\cdots,m$$

则

$$\alpha_{j+1}=\sum_{i=1}^{n}x_iQ_j^2(x_i)/d_j,\ j=1,2,\cdots,m-1$$

$$\beta_j=d_j/d_{j-1},\qquad j=2,3,\cdots,m-1$$

可以证明，由上面构造的多项式 $\{Q_j\}(j=1,2,\cdots,m)$ 是互相正交的。根据最小二乘原理，可得

$$q_j=\sum_{i=1}^{n}y_iQ_j(x_i)/d_j,\ j=1,2,\cdots,m$$

最后可以化成一般的 $m-1$ 次多项式

$$P_m(x)=a_1+a_2x+a_3x^2+\cdots+a_mx^{m-1}$$

具体计算步骤如下。

(1) $Q_1(x)=1$，可得

$$b_1=1,\ d_1=n,\ q_1=\sum_{i=1}^{n}y_i/d_1,\ \alpha=\sum_{i=1}^{n}x_i/d_1$$

$$a_1=q_1b_1$$

(2) $Q_2(x) = x - \alpha$, 可得

$$t_2 = 1,\ t_1 = -\alpha$$

$$d_2 = \sum_{i=1}^{n} Q_2^2(x_i),\ q_2 = \sum_{i=1}^{n} y_i Q_2(x_i)/d_2$$

$$\alpha = \sum_{i=1}^{n} x_i Q_2^2(x_i)/d_2,\ \beta = d_2/d_1$$

$$a_2 = q_2 t_2,\ q_2 t_1 + a_1 \Rightarrow a_1$$

(3) 对于 $j = 3,4,\cdots,m$, 作以下各步:

$$\begin{aligned}Q_j(x) &= (x-\alpha)Q_{j-1}(x) + \beta Q_{j-2}(x)\\ &= (x-\alpha)(t_{j-1}x^{j-2} + \cdots + t_2 x + t_1)\\ &\quad -\beta(b_{j-2}x^{j-3} + \cdots + b_2 x + b_1)\\ &\triangleq s_j x^{j-1} + s_{j-1}x^{j-2} + \cdots + s_2 x + s_1\end{aligned}$$

其中 s_k 由下列递推公式计算:

$$\begin{cases} s_j = t_{j-1} \\ s_{j-1} = -\alpha\, t_{j-1} + t_{j-2} \\ s_k = -\alpha\, t_k + t_{k-1} - \beta b_k,\ k = j-2,\cdots,2 \\ s_1 = -\alpha\, t_1 - \beta b_1 \end{cases}$$

再计算

$$d_j = \sum_{i=1}^{n} Q_j^2(x_i),\ q_j = \sum_{i=1}^{n} y_i Q_j(x_i)/d_j$$

$$\alpha = \sum_{i=1}^{n} x_i Q_j^2(x_i)/d_j,\ \beta = d_j/d_{j-1}$$

由此可以计算相应的 a_k:

$$\begin{cases} a_j = q_j s_j \\ a_k + q_j s_k \Rightarrow a_k,\ k = j-1,\cdots,1 \end{cases}$$

且

$$\begin{cases} t_j = s_j \\ b_k = t_k,\ t_k = s_k,\ k = j-1,\cdots,1 \end{cases}$$

在实际计算过程中,为了防止运算溢出, x_i 用

$$x_i' = x_i - \bar{x},\ i = 1,2,\cdots,n$$

代替。其中

$$\bar{x} = \sum_{i=1}^{n} x_i/n$$

此时,拟合多项式的形式为

$$P_m(x) = a_1 + a_2(x-\bar{x}) + a_3(x-\bar{x})^2 + \cdots + a_m(x-\bar{x})^{m-1}$$

三、子程序语句

SUBROUTINE HPIR1(X,Y,A,N,M,DT1,DT2,DT3)

四、形参说明

X,Y——均为双精度实型一维数组，长度为N，输入参数。分别存放N个数据点的X坐标与Y坐标。

A——双精度实型一维数组，长度为M，输出参数。存放M−1次拟合多项式的M个系数，即

$$P_M(x) = a_1 + a_2(x - \bar{x}) + a_3(x - \bar{x})^2 + \cdots + a_M(x - \bar{x})^{M-1}$$

N——整型变量，输入参数。数据点数。

M——整型变量，输入参数。拟合多项式的项数，即次数为M−1，要求M≤N且M≤20。

DT1,DT2,DT3——均为双精度实型变量，输出参数。分别为拟合多项式与数据点偏差的平方和、绝对值之和、绝对值的最大值。

五、子程序（文件名：HPIR1.FOR）

```
      SUBROUTINE HPIR1(X,Y,A,N,M,DT1,DT2,DT3)
      DIMENSION X(N),Y(N),A(M),S(20),T(20),B(20)
      DOUBLE PRECISION X,Y,A,S,T,B,DT1,DT2,DT3,
     *                 Z,D1,P,C,D2,G,Q,DT
      DO 5 I=1,M
5     A(I)=0.0
      IF (M.GT.N) M=N
      IF (M.GT.20) M=20
      Z=0.0
      DO 10 I=1,N
10    Z=Z+X(I)/N
      B(1)=1.0
      D1=N
      P=0.0
      C=0.0
      DO 20 I=1,N
        P=P+(X(I)-Z)
        C=C+Y(I)
20    CONTINUE
      C=C/D1
      P=P/D1
      A(1)=C*B(1)
      IF (M.GT.1) THEN
        T(2)=1.0
        T(1)=-P
        D2=0.0
        C=0.0
        G=0.0
```

```
      DO 30 I=1,N
        Q=X(I)-Z-P
        D2=D2+Q*Q
        C=Y(I)*Q+C
        G=(X(I)-Z)*Q*Q+G
30    CONTINUE
      C=C/D2
      P=G/D2
      Q=D2/D1
      D1=D2
      A(2)=C*T(2)
      A(1)=C*T(1)+A(1)
    END IF
    DO 100 J=3,M
      S(J)=T(J-1)
      S(J-1)=-P*T(J-1)+T(J-2)
      IF (J.GE.4) THEN
        DO 40 K=J-2,2,-1
40      S(K)=-P*T(K)+T(K-1)-Q*B(K)
      END IF
      S(1)=-P*T(1)-Q*B(1)
      D2=0.0
      C=0.0
      G=0.0
      DO 70 I=1,N
        Q=S(J)
        DO 60 K=J-1,1,-1
60      Q=Q*(X(I)-Z)+S(K)
        D2=D2+Q*Q
        C=Y(I)*Q+C
        G=(X(I)-Z)*Q*Q+G
70    CONTINUE
      C=C/D2
      P=G/D2
      Q=D2/D1
      D1=D2
      A(J)=C*S(J)
      T(J)=S(J)
      DO 80 K=J-1,1,-1
        A(K)=C*S(K)+A(K)
        B(K)=T(K)
        T(K)=S(K)
```

```
80     CONTINUE
100   CONTINUE
      DT1=0.0
      DT2=0.0
      DT3=0.0
      DO 120 I=1,N
        Q=A(M)
        DO 110 K=M-1,1,-1
110     Q=Q*(X(I)-Z)+A(K)
        DT=Q-Y(I)
        IF (ABS(DT).GT.DT3) DT3=ABS(DT)
        DT1=DT1+DT*DT
        DT2=DT2+ABS(DT)
120   CONTINUE
      RETURN
      END
```

六、例

设给定函数

$$f(x) = x - e^{-x}$$

从 $x_1 = 0$ 开始，取步长 $h = 0.1$ 的 20 个数据点，求 5 次最小二乘拟合多项式

$$P_5(x) = a_1 + a_2(x - \bar{x}) + a_3(x - \bar{x})^2 + \cdots + a_6(x - \bar{x})^5$$

其中

$$\bar{x} = \sum_{i=1}^{20} x_i / 20 = 0.95$$

在本问题中，N=20，M=6。

主程序(文件名：HPIR10.FOR)为

```
      DIMENSION X(20),Y(20),A(6)
      DOUBLE PRECISION X,Y,A,DT1,DT2,DT3,B
      B=0.0
      DO 10 I=1,20
        X(I)=B+(I-1)*0.1
        Y(I)=X(I)-EXP(-X(I))
10    CONTINUE
      N=20
      M=6
      CALL HPIR1(X,Y,A,N,M,DT1,DT2,DT3)
      WRITE(*,*)
      WRITE(*,20) (I,A(I),I=1,M)
20    FORMAT(1X,'A(',I2,')=',D15.6)
      WRITE(*,*)
      WRITE(*,30) DT1,DT2,DT3
```

```
30    FORMAT(1X,'DT1=',D12.6,5X,'DT2=',D12.6,5X,'DT3=',D12.6)
      END
```

运行结果为

```
A( 1 )=    .563248D+00
A( 2 )=    .138675D+01
A( 3 )=   -.193134D+00
A( 4 )=    .644035D-01
A( 5 )=   -.168412D-01
A( 6 )=    .334429D-02

DT1= .380174D-08    DT2= .168505D-03    DT3= .153940D-04
```

8.2 切比雪夫曲线拟合

一、功能

给定 n 个数据点，求切比雪夫(Chebyshev)意义下的最佳拟合多项式。

二、方法说明

设给定 n 个数据点

$$(x_i, y_i), i = 1, 2, \cdots, n$$

其中 $x_1 < x_2 < \cdots < x_{n-1} < x_n$。求 $m-1(m<n)$ 次多项式

$$P(x) = a_1 + a_2 x + \cdots + a_m x^{m-1}$$

使在 n 个给定点上的偏差最大值最小，即

$$\max_{1 \leqslant i \leqslant n} |P(x_i) - y_i| = \min$$

其计算步骤如下。

从给定的 n 个数据点的自变量值 $x_i (i = 1, 2, \cdots, n)$ 中选取 $m+1$ 个不同点 $u_1, u_2, \cdots, u_{m+1}$ 组成初始参考点集。

设在初始点集 $u_1, u_2, \cdots, u_{m+1}$ 上，参考多项式 $\Phi(x)$ 的偏差为 h，即参考多项式 $\Phi(x)$ 在初始点集上的取值为

$$\Phi(u_i) = f(u_i) + (-1)^i h, i = 1, 2, \cdots, m+1$$

且 $\Phi(u_i)$ 的各阶差商是 h 的线性函数。

由于 $\Phi(x)$ 为 $m-1$ 次多项式，其 m 阶差商等于零，由此可以求出 h。再根据 $\Phi(u_i)$ 的各阶差商，由牛顿插值公式可求出 $\Phi(x)$：

$$\Phi(x) = a_1 + a_2 x + a_3 x^2 + \cdots + a_m x^{m-1}$$

令

$$hh = \max_{1 \leqslant i \leqslant n} |\Phi(x_i) - y_i|$$

若 $hh = h$，则 $\Phi(x)$ 即为所求的拟合多项式。

若 $hh > h$，则用达到偏差最大值的点 x_j 代替点集 $\{u_j\} (i = 1, 2, \cdots, m+1)$ 中离 x_j 最近、且具有与 $(\Phi(x_j) - y_j)$ 的符号相同的点，从而构造一个新的参考点集。用这个新的参

考点集重复以上过程，直到最大逼近误差等于参考偏差为止。

三、子程序语句

SUBROUTINE HCHIR(X,Y,N,A,M,M1)

四、形参说明

X,Y——均为双精度实型一维数组，长度为N，输入参数。存放N个数据点的X坐标与Y坐标。

N——整型变量，输入参数。数据点数。

A——双精度实型一维数组，长度为M1=M+1，输出参数。其中A(i) (i=1,2,…,M)存放M−1次拟合多项式

$$P(x)=a_1+a_2x+a_3x^2+\cdots+a_mx^{m-1}$$

的系数a_i；A(M+1)存放拟合多项式的偏差。若A(M+1)为负值，则说明在迭代过程中参考偏差绝对值不再增大，其绝对值为当前选择的参考偏差的绝对值。

M——整型变量，输入参数。拟合多项式的项数，即次数为M−1。要求M<N，且M≤19。

M1——整型变量，输入参数。M1=M+1。

五、子程序(文件名：HCHIR.FOR)

```
      SUBROUTINE HCHIR(X,Y,N,A,M,M1)
      DIMENSION X(N),Y(N),A(M1),IX(20),H(20)
      DOUBLE PRECISION X,Y,A,H,HA,HH,Y1,Y2,H1,H2,D,HM
      DO 5 I=1,M1
5     A(I)=0.0
      IF (M.GE.N) M=N-1
      IF (M.GE.20) M=19
      M1=M+1
      HA=0.0
      IX(1)=1
      IX(M1)=N
      L=(N-1)/M
      J=L
      DO 10 I=2,M
        IX(I)=J+1
        J=J+L
10    CONTINUE
20    HH=1.0
      DO 30 I=1,M1
        A(I)=Y(IX(I))
        H(I)=-HH
        HH=-HH
30    CONTINUE
      DO 50 J=1,M
```

```
      II=M1
      Y2=A(II)
      H2=H(II)
      DO 40 I=J,M
        D=X(IX(II))-X(IX(M1-I))
        Y1=A(M-I+J)
        H1=H(M-I+J)
        A(II)=(Y2-Y1)/D
        H(II)=(H2-H1)/D
        II=M-I+J
        Y2=Y1
        H2=H1
40      CONTINUE
50    CONTINUE
      HH=-A(M1)/H(M1)
      DO 60 I=1,M1
60    A(I)=A(I)+H(I)*HH
      DO 80 J=1,M-1
        II=M-J
        D=X(IX(II))
        Y2=A(II)
        DO 70 K=M1-J,M
          Y1=A(K)
          A(II)=Y2-D*Y1
          Y2=Y1
          II=K
70      CONTINUE
80    CONTINUE
      HM=ABS(HH)
      IF (HM.LE.HA) THEN
        A(M1)=-HM
        RETURN
      END IF
      A(M1)=HM
      HA=HM
      IM=IX(1)
      H1=HH
      J=1
      DO 100 I=1,N
        IF (I.EQ.IX(J)) THEN
          IF (J.LT.M1) J=J+1
        ELSE
```

```
      H2=A(M)
      DO 90 K=M-1,1,-1
90    H2=H2*X(I)+A(K)
      H2=H2-Y(I)
      IF (ABS(H2).GT.HM) THEN
        HM=ABS(H2)
        H1=H2
        IM=I
      END IF
    END IF
100 CONTINUE
    IF (IM.EQ.IX(1)) RETURN
    I=1
110 IF (IM.GE.IX(I)) THEN
      I=I+1
      IF (I.LE.M1) GOTO 110
    END IF
    IF (I.GT.M1) I=M1
    IF (I.EQ.(I/2)*2) THEN
      H2=HH
    ELSE
      H2=-HH
    END IF
    IF (H1*H2.GE.0.0) THEN
      IX(I)=IM
      GOTO 20
    END IF
    IF (IM.LT.IX(1)) THEN
      DO 120 J=M,1,-1
120   IX(J+1)=IX(J)
      IX(1)=IM
      GOTO 20
    END IF
    IF (IM.GT.IX(M1)) THEN
      DO 130 J=2,M1
130   IX(J-1)=IX(J)
      IX(M1)=IM
      GOTO 20
    END IF
    IX(I-1)=IM
    GOTO 20
    END
```

六、例

取函数 $f(x)=\mathrm{arctg}x$ 在区间[-1,1]上的 101 个等距点

$$x_i=-1.0+(i-1)*0.02, i=1,2,\cdots,101$$

上的函数值

$$y_i=f(x_i), i=1,2,\cdots,101$$

根据此 101 个数据点构造切比雪夫意义下的五次拟合多项式

$$P(x)=a_1+a_2x+a_3x^2+a_4x^3+a_5x^4+a_6x^5$$

在本例中，N=101，M=6，M1=7。

主程序(文件名：HCHIR0.FOR)为

```
      DIMENSION X(101),Y(101),A(7)
      DOUBLE PRECISION X,Y,A
      N=101
      M=6
      M1=7
      DO 10 I=1,N
        X(I)=-1.0+(I-1)*0.02
        Y(I)=ATAN(X(I))
10    CONTINUE
      CALL HCHIR(X,Y,N,A,M,M1)
      WRITE(*,*)
      WRITE(*,20) (I,A(I),I=1,M)
      WRITE(*,*)
      WRITE(*,30) A(M1)
      WRITE(*,*)
20    FORMAT(1X,'A(',I2,' )=',D15.6)
30    FORMAT(1X,'HMAX=',D15.6)
      END
```

运行结果为

```
A( 1 )=   -.430478D-11
A( 2 )=    .995364D+00
A( 3 )=    .548259D-11
A( 4 )=   -.288716D+00
A( 5 )=   -.372085D-10
A( 6 )=    .793575D-01

HMAX=      .607072D-03
```

8.3 最佳一致逼近的里米兹方法

一、功能

用里米兹(Remez)方法求函数的最佳一致逼近多项式。

二、方法说明

若函数 $f(x)$ 在区间$[a,b]$上的 $n-1$ 次最佳一致逼近多项式为

$$P_n(x)=p_1+p_2x+p_3x^2+\cdots+p_nx^{n-1}$$

则存在 $n+1$ 个点的交错点组$\{x_i\}$满足

$$f(x_i)-P_n(x_i)=(-1)^i\mu,\ i=1,2,\cdots,n+1$$

其中

$$\mu=\max_{x\in[a,b]}|f(x)-P_n(x)|$$

求函数 $f(x)$ 在区间$[a,b]$上的 $n-1$ 次最佳一致逼近多项式的里米兹方法如下。

(1) 在区间$[a,b]$上取 n 次切比雪夫(Chebyshev)多项式的交错点组

$$x_k=\frac{1}{2}\left[b+a+(b-a)\cos\frac{n-k+1}{n}\pi\right](k=1,2,\cdots,n+1$$

作为初始参考点集。

(2) 以参考点集$\{x_i\}(i=1,2,\cdots,n+1)$构造一个参考多项式

$$P_n(x)=p_1+p_2x+p_3x^2+\cdots+p_nx^{n-1}$$

满足

$$P_n(x_i)-f(x_i)=(-1)^i\mu,\ i=1,2,\cdots,n+1$$

由于 $P_n(x)$ 为 $n-1$ 次多项式，在 $n+1$ 个点$\{x_i\}(i=1,2,\cdots,n+1)$上的 n 阶差商为零，由此可以确定出参考偏差 μ。

然后根据 $P_n(x)$ 在 $n+1$ 个点$\{x_i\}(i=1,2,\cdots,n+1)$上的各阶差商，利用牛顿插值公式确定出参考多项式 $P_n(x)$ 的各系数 $p_1,p_2,\cdots,p_n$。

(3) 找出使函数 $|f(x)-P_n(x)|$ 在区间$[a,b]$上取最大值的点 $\hat{x}$，并按如下原则替换原参考点集$\{x_i\}(i=1,2,\cdots,n+1)$中的某一点：

若 $\hat{x}$ 在 a 与 x_1 之间，且 $f(x_1)-P_n(x_1)$ 与 $f(\hat{x})-P_n(\hat{x})$ 同号，则将 $\hat{x}$ 代替 x_1 构成新点集

$$\{\hat{x},x_2,\cdots,x_n,x_{n+1}\}$$

否则新点集为

$$\{\hat{x},x_1,x_2,\cdots,x_n\}$$

若 $\hat{x}$ 在 x_{n+1} 与 b 之间，且 $f(x_{n+1})-P_n(x_{n+1})$ 与 $f(\hat{x})-P_n(\hat{x})$ 同号，则将 $\hat{x}$ 代替 x_{n+1} 构成新点集

$$\{x_1,x_2,\cdots,x_n,\hat{x}\}$$

否则取新点集

$$\{x_2,x_3,\cdots,x_{n+1},\hat{x}\}$$

若 $\hat{x}$ 在 x_i 与 $x_{i+1}(i=1,2,\cdots,n)$ 之间，且 $f(x_i)-P_n(x_i)$ 与 $f(\hat{x})-P_n(\hat{x})$ 同号，则将 $\hat{x}$ 代替 x_i，否则将 $\hat{x}$ 代替 x_{i+1} 构成新点集。

重复(2)与(3)，直到相邻两次求得的参考偏差接近相等为止。此时，最后获得的参考多项式 $P_n(x)$ 即为近似的 $n-1$ 次最佳一致逼近多项式。

三、子程序语句

SUBROUTINE HREMZ(A,B,F,P,N,M,EPS)

四、形参说明

A,B——均为双精度实型变量，输入参数。区间的左、右端点值。

F——实型函数名，输入参数。用于计算函数 $f(x)$ 的值。在主程序中必须用外部语句对相应的实参进行说明。该函数子程序由用户自编，其语句形式为

FUNCTION F(X)

P——双精度实型一维数组，长度为 M=N+1，输出参数。其中前 N 个元素依次存放 N-1 次最佳一致逼近多项式 $P_n(x)$ 的 N 个系数 $p_1,p_2,\cdots,p_n$；P (M)存放 $P_n(x)$ 的偏差绝对值。

N——整型变量，输入参数。最佳一致逼近多项式的项数，即次数为 N-1。要求 N<20。

M——整型变量，输入参数。M=N+1。

EPS——实型变量，输入参数。控制精度要求，一般取 10^{-10}～10^{-35}之间的数。

五、子程序(文件名：HREMZ.FOR)

```
      SUBROUTINE HREMZ(A,B,F,P,N,M,EPS)
      DIMENSION P(M),X(20),G(20)
      DOUBLE PRECISION P,X,G,H,U,D,A,B,XX,X0,S,T,YY
      IF (N.GE.20) N=19
      D=1.0D+35
      DO 10 K=0,N
        T=COS((N-K)*3.1415926/(1.0*N))
        X(K+1)=(B+A+(B-A)*T)/2.0
10    CONTINUE
15    U=1.0
      DO 20 I=1,M
        P(I)=F(X(I))
        G(I)=-U
        U=-U
20    CONTINUE
      DO 30 J=1,N
        K=M
        S=P(K)
        XX=G(K)
        DO 25 I=J,N
          T=P(N-I+J)
          X0=G(N-I+J)
          P(K)=(S-T)/(X(K)-X(M-I))
          G(K)=(XX-X0)/(X(K)-X(M-I))
          K=N-I+J
          S=T
          XX=X0
```

```
25      CONTINUE
30    CONTINUE
      U=-P(M)/G(M)
      DO 35 I=1,M
35    P(I)=P(I)+G(I) * U
      DO 50 J=1,N-1
        K=N-J
        H=X(K)
        S=P(K)
        DO 40 I=M-J,N
          T=P(I)
          P(K)=S-H * T
          S=T
          K=I
40      CONTINUE
50    CONTINUE
      P(M)=ABS(U)
      U=P(M)
      IF (ABS(U-D).LE.EPS) RETURN
      D=U
      H=0.1 * (B-A)/N
      XX=A
      X0=A
60    IF (X0.LE.B) THEN
        S=F(X0)
        T=P(N)
        DO 70 I=N-1,1,-1
70      T=T * X0+P(I)
        S=ABS(S-T)
        IF (S.GT.U) THEN
          U=S
          XX=X0
        END IF
        X0=X0+H
        GOTO 60
      END IF
      S=F(XX)
      T=P(N)
      DO 75 I=N-1,1,-1
75    T=T * XX+P(I)
      YY=S-T
      I=1
```

```
      J=N+1
80    IF ((J-I).NE.1) THEN
        K=(I+J)/2
        IF (XX.LT.X(K)) THEN
          J=K
        ELSE
          I=K
        END IF
        GOTO 80
      END IF
      IF (XX.LT.X(1)) THEN
        S=F(X(1))
        T=P(N)
        DO 90 K=N-1,1,-1
90      T=T*X(1)+P(K)
        S=S-T
        IF (S*YY.GT.0.0) THEN
          X(1)=XX
        ELSE
          DO 95 K=N,1,-1
95        X(K+1)=X(K)
          X(1)=XX
        END IF
      ELSE IF (XX.GT.X(N+1)) THEN
        S=F(X(N+1))
        T=P(N)
        DO 100 K=N-1,1,-1
100     T=T*X(N+1)+P(K)
        S=S-T
        IF (S*YY.GT.0.0) THEN
          X(N+1)=XX
        ELSE
          DO 105 K=1,N
105       X(K)=X(K+1)
          X(N+1)=XX
        END IF
      ELSE
        S=F(X(I))
        T=P(N)
        DO 110 K=N-1,1,-1
110     T=T*X(I)+P(K)
        S=S-T
```

```
      IF (S*YY.GT.0.0) THEN
        X(I)=XX
      ELSE
        X(J)=XX
      END IF
    END IF
    GOTO 15
    RETURN
    END
```

六、例

(1) 求函数 $f(x) = e^x$ 在区间$[-1,1]$上的三次最佳一致逼近多项式及其偏差的绝对值.其中 $A=-1,B=1,N=4,M=5,\epsilon=10^{-10}$。

(2) 求函数$f(x)=\mathrm{arctg}x$在区间$[-1,1]$上的五次最佳一致逼近多项式及其偏差的绝对值.其中 $A=-1,B=1,N=6,M=7,\epsilon=10^{-10}$。

主程序及计算函数值 $f(x)$ 的子程序(文件名:HREMZ0.FOR)为

```
      EXTERNAL F1,F2
      DIMENSION P(5),Q(7)
      DOUBLE PRECISION P,Q,A,B
      A=-1.0
      B=1.0
      EPS=1.0E-10
      WRITE(*,*)
      CALL HREMZ(A,B,F1,P,4,5,EPS)
      WRITE(*,10) (I,P(I),I=1,4)
10    FORMAT(1X,'P(',I2,' )=',D15.6)
      WRITE(*,30) P(5)
      WRITE(*,*)
      CALL HREMZ(A,B,F2,Q,6,7,EPS)
      WRITE(*,20) (I,Q(I),I=1,6)
20    FORMAT(1X,'Q(',I2,' )=',D15.6)
      WRITE(*,30) Q(7)
30    FORMAT(1X,'HMAX=',D15.6)
      END

      FUNCTION F1(X)
      DOUBLE PRECISION X
      F1=EXP(X)
      RETURN
      END

      FUNCTION F2(X)
      DOUBLE PRECISION X
```

```
      F2=ATAN(X)
      RETURN
      END
```

运行结果为

$f(x)=e^x$ 的三次最佳一致逼近多项式的系数与偏差的绝对值：

```
P( 1 )=     .994594D+00
P( 2 )=     .995483D+00
P( 3 )=     .542974D+00
P( 4 )=     .179518D+00
HMAX=       .551301D-02
```

$f(x)=\mathrm{arctg}x$ 的五次最佳一致逼近多项式的系数与偏差的绝对值：

```
Q( 1 )=     .404098D-08
Q( 2 )=     .995373D+00
Q( 3 )=    -.932517D-07
Q( 4 )=    -.288758D+00
Q( 5 )=     .140570D-06
Q( 6 )=     .793890D-01
HMAX=       .605692D-03
```

8.4 矩形域的最小二乘曲面拟合

一、功能

用最小二乘法求矩形域上 $n\times m$ 个数据点的拟合曲面。

二、方法说明

设已知矩形区域内 $n\times m$ 个网点 $(x_i,y_j)(i=1,2,\cdots,n;j=1,2,\cdots,m)$ 上的函数值 z_{ij}，求最小二乘拟合多项式

$$f(x,y)=\sum_{i=1}^{p}\sum_{j=1}^{q}a_{ij}x^{i-1}y^{j-1}$$

首先，固定 y，对 x 构造 m 个最小二乘拟合多项式

$$G_j(x)=\sum_{k=1}^{p}\lambda_{kj}\Phi_k(x),\ j=1,2,\cdots,m$$

其中各 $\Phi_k(x)(k=1,2,\cdots,p)$ 为互相正交的多项式，并由下列递推公式构造：

$$\Phi_1(x)=1$$

$$\Phi_2(x)=x-\alpha_1$$

$$\Phi_{k+1}(x)=(x-\alpha_k)\Phi_k(x)-\beta_k\Phi_{k-1}(x),k=2,3,\cdots,p-1$$

若令

$$d_k=\sum_{i=1}^{n}\Phi_k^2(x_i),\ k=1,2,\cdots,p$$

则有

$$\alpha_k = \sum_{i=1}^{n} x_i \Phi_k^2(x_i)/d_k,\ k = 1,2,\cdots,p-1$$

$$\beta_k = d_k/d_{k-1},\qquad k = 1,2,\cdots,p-1$$

根据最小二乘原理可得

$$\lambda_{kj} = \sum_{i=1}^{n} z_{ij}\Phi_k(x_i)/d_k,\ j = 1,2,\cdots,m$$

$$k = 1,2,\cdots,p$$

然后，再构造 y 的最小二乘拟合多项式

$$H_k(y) = \sum_{l=1}^{q} \mu_{kl}\Psi_l(y),\ k = 1,2,\cdots,p$$

其中各 $\Psi_l(y)(l = 1,2,\cdots,q)$ 也为互相正交的多项式，并由下列递推公式构造：

$$\Psi_1(y) = 1$$

$$\Psi_2(y) = y - \alpha_1'$$

$$\Psi_{l+1}(y) = (y - \alpha_l')\Psi_l(y) - \beta_l'\Psi_{l-1}(y),\ l = 2,3,\cdots,q-1$$

若令

$$\delta_l = \sum_{j=1}^{m} \Psi_l^2(y_j),\ l = 1,2,\cdots,q$$

则有

$$\alpha_l' = \sum_{j=1}^{m} y_j\Psi_l^2(y_j)/\delta_l,\ l = 1,2,\cdots,q-1$$

$$\beta_l' = \delta_l/\delta_{l-1},\qquad l = 1,2,\cdots,q-1$$

根据最小二乘原理可得

$$\mu_{kl} = \sum_{j=1}^{m} \lambda_{kj}\Psi_l(y_j)/\delta_l,\ k = 1,2,\cdots,p$$

$$l = 1,2,\cdots,q$$

最后可得二元函数的拟合多项式为

$$f(x,y) = \sum_{k=1}^{p}\sum_{l=1}^{q} \mu_{kl}\Phi_k(x)\Psi_l(y)$$

再转换成标准的多项式形式

$$f(x,y) = \sum_{i=1}^{p}\sum_{j=1}^{q} a_{ij}x^{i-1}y^{j-1}$$

在实际计算过程中，为了防止运算溢出，x_i 与 y_j 分别用

$$x_i' = x_i - \overline{x},\ i = 1,2,\cdots,n$$

$$y_j' = y_j - \overline{y},\ j = 1,2,\cdots,m$$

代替。其中

$$\overline{x} = \sum_{i=1}^{n} x_i/n,\ \overline{y} = \sum_{j=1}^{m} y_j/m$$

此时，二元拟合多项式的形式为

$$f(x,y)=\sum_{i=1}^{p}\sum_{j=1}^{q}a_{ij}(x-\bar{x})^{i-1}(y-\bar{y})^{j-1}$$

三、子程序语句

SUBROUTINE HPIR2(X,Y,Z,N,M,A,P,Q,DT1,DT2,DT3,V)

四、形式说明

X——双精度实型一维数组，长度为N，输入参数。自变量 x 的 N 个点值。

Y——双精度实型一维数组，长度为M，输入参数。自变量 y 的 M 个点值。

Z——双精度实型二维数组，体积为 N×M，输入参数。矩形区域内 N×M 个网点 $(x_i,y_j)(i=1,2,\cdots,N;j=1,2,\cdots,M)$ 上的函数值 Z_{ij}。

N——整型变量，输入参数。X 方向上的结点个数。

M——整型变量，输入参数。Y 方向上的结点个数。

A——双精度实型二维数组，体积为 P×Q，输出参数。存放二元拟合多项式

$$f(x,y)=\sum_{i=1}^{p}\sum_{j=1}^{q}a_{ij}(x-\bar{x})^{i-1}(y-\bar{y})^{j-1}$$

的系数 $a_{ij}(i=1,2,\cdots,P;j=1,2,\cdots,Q)$。

P——整型变量，输入参数。拟合多项式中 x 的最高次数加1，即 x 的最高次为P−1。要求P≤N，且 P≤20。

Q——整型变量，输入参数。拟合多项式中 y 的最高次数加1，即 y 的最高次为Q−1。要求Q≤M，且Q≤20。

DT1,DT2,DT3——均为双精度实型变量，输出参数。分别为拟合多项式与数据点的偏差平方和、偏差绝对值之和、偏差绝对值的最大值，即

$$DT1=\sum_{i=1}^{N}\sum_{j=1}^{M}[f(x_i,y_j)-Z_{ij}]^2$$

$$DT2=\sum_{i=1}^{N}\sum_{j=1}^{M}|f(x_i,y_j)-Z_{ij}|$$

$$DT3=\max_{\substack{1\le i\le N\\1\le j\le M}}|f(x_i,y_j)-Z_{ij}|$$

V——双精度实型二维数组，体积为 20×M。本子程序的工作数组。

五、子程序(文件名:HPIR2.FOR)

```
      SUBROUTINE HPIR2(X,Y,Z,N,M,A,P,Q,DT1,DT2,DT3,V)
      INTEGER P,Q
      DIMENSION X(N),Y(M),Z(N,M),A(P,Q)
      DIMENSION APX(20),APY(20),BX(20),BY(20),U(20,20),V(20,M)
      DIMENSION T(20),T1(20),T2(20)
      DOUBLE PRECISION X,Y,Z,A,APX,APY,BX,BY,U,V,T,T1,T2,DT1,DT2,
     *                  DT3,XX,YY,D1,D2,G,G1,G2,X1,X2,Y1,DD,DT
      DO 5 I=1,P
      DO 5 J=1,Q
5     A(I,J)=0.0
```

```
      IF (P.GT.N) P=N
      IF (P.GT.20) P=20
      IF (Q.GT.M) Q=M
      IF (Q.GT.20) Q=20
      XX=0.0
      DO 10 I=1,N
10    XX=XX+X(I)/N
      YY=0.0
      DO 20 I=1,M
20    YY=YY+Y(I)/M
      D1=N
      APX(1)=0.0
      DO 30 I=1,N
30    APX(1)=APX(1)+X(I)-XX
      APX(1)=APX(1)/D1
      DO 50 J=1,M
        V(1,J)=0.0
        DO 40 I=1,N
40      V(1,J)=V(1,J)+Z(I,J)
        V(1,J)=V(1,J)/D1
50    CONTINUE
      IF (P.GT.1) THEN
        D2=0.0
        APX(2)=0.0
        DO 60 I=1,N
          G=X(I)-XX   APX(1)
          D2=D2+G*G
          APX(2)=APX(2)+(X(I)-XX)*G*G
60      CONTINUE
        APX(2)=APX(2)/D2
        BX(2)=D2/D1
        DO 80 J=1,M
          V(2,J)=0.0
          DO 70 I=1,N
            G=X(I)-XX-APX(1)
            V(2,J)=V(2,J)+Z(I,J)*G
70        CONTINUE
          V(2,J)=V(2,J)/D2
80      CONTINUE
        D1=D2
      END IF
```

```
      DO 140 K=3,P
        D2=0.0
        APX(K)=0.0
        DO 90 J=1,M
90      V(K,J)=0.0
        DO 120 I=1,N
          G1=1.0
          G2=X(I)-XX-APX(1)
          DO 100 J=3,K
            G=(X(I)-XX-APX(J-1))*G2-BX(J-1)*G1
            G1=G2
            G2=G
100       CONTINUE
          D2=D2+G*G
          APX(K)=APX(K)+(X(I)-XX)*G*G
          DO 110 J=1,M
110       V(K,J)=V(K,J)+Z(I,J)*G
120     CONTINUE
        DO 130 J=1,M
130     V(K,J)=V(K,J)/D2
        APX(K)=APX(K)/D2
        BX(K)=D2/D1
        D1=D2
140   CONTINUE
      D1=M
      APY(1)=0.0
      DO 150 I=1,M
150   APY(1)=APY(1)+Y(I)-YY
      APY(1)=APY(1)/D1
      DO 170 J=1,P
        U(J,1)=0.0
        DO 160 I=1,M
160     U(J,1)=U(J,1)+V(J,I)
        U(J,1)=U(J,1)/D1
170   CONTINUE
      IF (Q.GT.1) THEN
        D2=0.0
        APY(2)=0.0
        DO 180 I=1,M
          G=Y(I)-YY-APY(1)
          D2=D2+G*G
          APY(2)=APY(2)+(Y(I)-YY)*G*G
```

```
180     CONTINUE
        APY(2)=APY(2)/D2
        BY(2)=D2/D1
        DO 200 J=1,P
          U(J,2)=0.0
          DO 190 I=1,M
            G=Y(I)-YY-APY(1)
            U(J,2)=U(J,2)+V(J,I)*G
190       CONTINUE
          U(J,2)=U(J,2)/D2
200     CONTINUE
        D1=D2
      END IF
      DO 260 K=3,Q
        D2=0.0
        APY(K)=0.0
        DO 210 J=1,P
210     U(J,K)=0.0
        DO 240 I=1,M
          G1=1.0
          G2=Y(I)-YY-APY(1)
          DO 220 J=3,K
            G=(Y(I)-YY-APY(J-1))*G2-BY(J-1)*G1
            G1=G2
            G2=G
220       CONTINUE
          D2=D2+G*G
          APY(K)=APY(K)+(Y(I)-YY)*G*G
          DO 230 J=1,P
230       U(J,K)=U(J,K)+V(J,I)*G
240     CONTINUE
        DO 250 J=1,P
250     U(J,K)=U(J,K)/D2
        APY(K)=APY(K)/D2
        BY(K)=D2/D1
        D1=D2
260   CONTINUE
      V(1,1)=1.0
      V(2,1)=-APY(1)
      V(2,2)=1.0
      DO 265 I=1,P
      DO 265 J=1,Q
```

```
265   A(I,J)=0.0
      DO 280 I=3,Q
        V(I,I)=V(I-1,I-1)
        V(I,I-1)=-APY(I-1)*V(I-1,I-1)+V(I-1,I-2)
        IF (I.GE.4) THEN
          DO 270 K=I-2,2,-1
270       V(I,K)=-APY(I-1)*V(I-1,K)+V(I-1,K-1)-BY(I-1)*V(I-2,K)
        END IF
        V(I,1)=-APY(I-1)*V(I-1,1)-BY(I-1)*V(I-2,1)
280   CONTINUE
      DO 320 I=1,P
        IF (I.EQ.1) THEN
          T(1)=1.0
          T1(1)=1.0
        ELSE IF (I.EQ.2) THEN
          T(1)=-APX(1)
          T(2)=1.0
          T2(1)=T(1)
          T2(2)=T(2)
        ELSE
          T(I)=T2(I-1)
          T(I-1)=-APX(I-1)*T2(I-1)+T2(I-2)
          IF (I.GE.4) THEN
            DO 290 K=I-2,2,-1
290         T(K)=-APX(I-1)*T2(K)+T2(K-1)-BX(I-1)*T1(K)
          END IF
          T(1)=-APX(I-1)*T2(1)-BX(I-1)*T1(1)
          T2(I)=T(I)
          DO 310 K=I-1,1,-1
            T1(K)=T2(K)
            T2(K)=T(K)
310       CONTINUE
        END IF
        DO 300 J=1,Q
        DO 300 K=I,1,-1
        DO 300 L=J,1,-1
300     A(K,L)=A(K,L)+U(I,J)*T(K)*V(J,L)
320   CONTINUE
      DT1=0.0
      DT2=0.0
      DT3=0.0
      DO 400 I=1,N
```

```
        X1=X(I)-XX
        DO 350 J=1,M
          Y1=Y(J)-YY
          X2=1.0
          DD=0.0
          DO 340 K=1,P
            G=A(K,Q)
            DO 330 KK=Q-1,1,-1
330         G=G*Y1+A(K,KK)
            G=G*X2
            DD=DD+G
            X2=X2*X1
340       CONTINUE
          DT=DD-Z(I,J)
          IF (ABS(DT).GT.DT3) DT3=ABS(DT)
          DT1=DT1+DT*DT
          DT2=DT2+ABS(DT)
350     CONTINUE
400   CONTINUE
      RETURN
      END
```

六、例

设二元函数

$$Z(x,y)=e^{x^2-y^2}$$

取矩形区域内 11×21 个网点

$$\begin{cases} x_i=0.2(i-1),\ i=1,2,\cdots,11 \\ y_j=0.1(j-1),\ j=1,2,\cdots,21 \end{cases}$$

上的函数值 $Z_{ij}=Z(x_i,y_j)$，由这些数据点构造一个最小二乘拟合多项式

$$f(x,y)=\sum_{i=1}^{4}\sum_{j=1}^{5}a_{ij}(x-\bar{x})^{i-1}(y-\bar{y})^{j-1}$$

其中

$$\bar{x}=\sum_{i=1}^{11}x_i/11=1.0,\bar{y}=\sum_{j=1}^{21}y_j/21=1.0$$

并分别计算偏差平方和 DT1、偏差绝对值之和 DT2 和偏差绝对值最大者 DT3。

主程序(文件名:HPIR20.FOR)为

```
      DIMENSION X(11),Y(21),Z(11,21),A(6,5),V(20,21)
      INTEGER P,Q
      DOUBLE PRECISION X,Y,Z,A,DT1,DT2,DT3,V
      N=11
      M=21
```

```
      P=6
      Q=5
      DO 10 I=1,N
10    X(I)=0.2*(I-1)
      DO 20 I=1,M
20    Y(I)=0.1*(I-1)
      DO 40 I=1,N
      DO 40 J=1,M
40    Z(I,J)=EXP(X(I)*X(I)-Y(J)*Y(J))
      CALL HPIR2(X,Y,Z,N,M,A,P,Q,DT1,DT2,DT3,V)
      WRITE(*,*)
      WRITE(*,50)
50    FORMAT(1X,27X,'A(I,J)')
      WRITE(*,*)
      WRITE(*,60) ((A(I,J),J=1,Q),I=1,P)
60    FORMAT(1X,5D13.6)
      WRITE(*,*)
      WRITE(*,70) DT1,DT2,DT3
70    FORMAT(1X,'DT1=',D13.6,3X,'DT2=',D13.6,3X,'DT3=',D13.6)
      END
```

运行结果为

```
                    A(I,J)

 .111924D+01 -.222009D+01  .997902D+00  .738116D+00 -.581947D+00
 .225202D+01 -.446704D+01  .200788D+01  .148516D+01 -.117094D+01
 .916373D+00 -.181769D+01  .817029D+00  .604331D+00 -.476468D+00
 .134836D+01 -.267457D+01  .120219D+01  .889219D+00 -.701080D+00
 .819701D+01 -.162593D+02  .730837D+01  .540577D+01 -.426203D+01
 .630257D+01 -.125016D+02  .561931D+01  .415642D+01 -.327702D+01

 DT1=  .580139D+01   DT2=  .253761D+02   DT3=  .555972D+00
```

第9章 数据处理与回归分析

9.1 随机样本分析

一、功能

(1) 计算给定一维随机样本的算术平均值、方差与标准差。

(2) 按高斯分布计算出在给定各区间上的近似理论样本点数。

(3) 打印经验直方图。

二、方法说明

设给定随机变量 x 的 n 个样本点值 $x_i(i=1,2,\cdots,n)$。

(1) 计算样本参数值

随机样本算术平均值

$$\bar{x}=\sum_{i=1}^{n} x_i/n$$

样本方差

$$s=\sum_{i=1}^{n}(x_i-\bar{x})^2/n$$

样本标准差

$$t=\sqrt{s}$$

(2) 按高斯分布计算出给定各区间上的近似理论样本点数。

设随机变量 x 的起始值为 x_0，区间长度为 h，则第 i 个区间的中点为

$$x_i^*=x_0+(i-0.5)h,\quad i=1,2,\cdots$$

在第 i 个区间上，按高斯分布所应有的近似理论样本点数为

$$F_i=\frac{n}{\sqrt{2\pi s}}\mathrm{EXP}\left(-\frac{(x_i^*-\bar{x})^2}{2s}\right)\cdot h$$

(3) 打印经验直方图

在直方图上方打印样本点数 n，直方图中随机变量的起始值 x_0，直方图中随机变量的等区间长度值 h，直方图中区间总数 m，随机变量样本的算术平均值 xa，随机样本的均方差 s，以及标准偏差 t。

在打印的直方图中：

左起第一列为从小到大打印各区间的中点值；

第二列打印出随机样本中落在对应区间中的实际点数；

右边是直方图本身。各区间行上的"X"的个数代表样本中随机变量值落在该区间中的个数，而"*"号所占的序数则为按高斯分布计算得到的近似理论点数。

三、子程序语句

SUBROUTINE IRHIS(N,X,XA,S,T,X0,H,M,F,E,K)

四、形参说明

N——整型变量,输入参数。随机样本点数。

X——实型一维数组,长度为N,输入参数。存放随机变量的N个样本点值。

XA——实型变量,输出参数。随机样本的算术平均值。

S——实型变量,输出参数。随机样本的方差。

T——实型变量,输出参数。随机样本的标准差。

X0——实型变量,输入参数。直方图中随机变量的起始值。

H——实型变量,输入参数。直方图中随机变量等区间长度值。

M——整型变量,输入参数。直方图中区间总数。

F——实型一维数组,长度为M,输出参数。存放M个区间的每一个区间上按高斯分布所应有的近似理论样本点数。

E——实型一维数组,长度为M,输出参数。存放落在M个区间的每一个区间上的随机样本实际点数。

K——整型变量,输入参数。若K=0,表示不打印直方图;若K≠0,表示需要打印直方图;格式见方法说明并参看本节例。

五、子程序(文件名:IRHIS.FOR)

```
      SUBROUTINE IRHIS(N,X,XA,S,T,X0,H,M,F,E,K)
      CHARACTER A(50)
      DIMENSION X(N),F(M),E(M)
      XA=0.0
      DO 10 I=1,N
10    XA=XA+X(I)/N
      S=0.0
      DO 20 I=1,N
20    S=S+(X(I)-XA)*(X(I)-XA)
      S=S/(N*1.0)
      T=SQRT(S)
      DO 30 I=1,M
        E(I)=0.0
        Z=X0+(I-0.5)*H-XA
        Z=EXP(-Z*Z/(2.0*S))
        F(I)=N*Z*H/(T*2.5066)
30    CONTINUE
      XE=X0+M*H
      DO 40 I=1,N
        IF ((X(I)-X0).GE.0.0) THEN
          IF ((XE-X(I)).GE.0.0) THEN
            J=(X(I)-X0)/H+1
```

```
            E(J)=E(J)+1
          END IF
        END IF
40    CONTINUE
      IF (K.EQ.0) RETURN
      WRITE(*,*)
      WRITE(*,50) N
50    FORMAT(1X,'N=',I10)
      WRITE(*,*)
      WRITE(*,60) X0,H,M
60    FORMAT(1X,'X0=',E13.6,3X,'H=',E13.6,3X,'M=',I5)
      WRITE(*,*)
      WRITE(*,70) XA,S,T
70    FORMAT(1X,'XA=',E13.6,3X,'S=',E13.6,3X,'T=',E13.6)
      WRITE(*,*)
      XE=1.0
      Z=0.0
      DO 80 I=1,M
        IF (E(I).GT.Z) Z=E(I)
80    CONTINUE
90    IF (Z.GT.50.0) THEN
        Z=Z/2
        XE=XE/2
        GOTO 90
      END IF
      DO 120 I=1,M
        Z=X0+(I-0.5)*H
        DO 100 J=1,50
100     A(J)=' '
        J=E(I)*XE+0.5
        IF (J.GT.0) THEN
          DO 110 L=1,J
110       A(L)='X'
        END IF
        J=F(I)*XE+0.5
        IF ((J.GT.0).AND.(J.LE.50)) A(J)='*'
        WRITE(*,130) Z,E(I),A
        WRITE(*,*)
120   CONTINUE
130   FORMAT(1X,E13.6,1X,E13.6,1X,50A1)
      RETURN
      END
```

六、例

给定随机变量的 100 个样本点(参看主程序中的 DATA 语句),打印出直方图。

其中 N=100,X0=192.000,H=2.00000,M=10,K≠0.

主程序(文件名:IRHIS0.FOR)为

```
      DIMENSION X(100),F(10),E(10)
      DATA X/193.199,195.673,195.757,196.051,196.092,196.596,
     *       196.579,196.763,196.847,197.267,197.392,197.477,
     *       198.189,193.850,198.944,199.070,199.111,199.153,
     *       199.237,199.698,199.572,199.614,199.824,199.908,
     *       200.188,200.160,200.243,200.285,200.453,200.704,
     *       200.746,200.830,200.872,200.914,200.956,200.998,
     *       200.998,201.123,201.208,201.333,201.375,201.543,
     *       201.543,201.584,201.711,201.878,201.919,202.004,
     *       202.004,202.088,202.172,202.172,202.297,202.339,
     *       202.381,202.507,202.591,202.716,202.633,202.884,
     *       203.051,203.052,203.094,203.094,203.177,203.178,
     *       203.219,203.764,203.765,203.848,203.890,203.974,
     *       204.184,204.267,204.352,204.352,204.729,205.106,
     *       205.148,205.231,205.357,205.400,205.483,206.070,
     *       206.112,206.154,206.155,206.615,206.657,206.993,
     *       207.243,207.621,208.124,208.375,208.502,208.628,
     *       208.670,208.711,210.012,211.394/
      N=100
      M=10
      X0=192.0
      H=2.0
      K=1
      CALL IRHIS(N,X,XA,S,T,X0,H,M,F,E,K)
      END
```

运行结果为

```
N=            100
X0 =  .192000E+03    H=  .200000E+01    M--    10
XA=  .202230E+03   S=  .129867E+02    T--   .360370E+01
.193000E+03        .200000E+01   *X
.195000E+03        .200000E+01  XX*
.197000E+03        .900000E+01  XXXXXXX*X
.199000E+03        .110000E+02  XXXXXXXXXXX    *
.201000E+03        .230000E+02  XXXXXXXXXXXXXXXXXXXX*XX
.203000E+03        .250000E+02  XXXXXXXXXXXXXXXXXXXXXX*XXX
.205000E+03        .110000E+02  XXXXXXXXXXX     *
.207000E+03        .900000E+01  XXXXXXXX*
```

.209000E+03 .600000E+01 XXX*XX
.211000E+03 .200000E+01 *X

9.2 一元线性回归分析

一、功能

对于给定的 n 个数据点 $(x_i,y_i)(i=1,2,\cdots,n)$ 用直线 $y=ax+b$ 作回归分析。

二、方法说明

设随机变量 y 随自变量 x 变化。已知 n 组观测数据 (x_i,y_i) $(i=1,2,\cdots,n)$，用直线

$$y=ax+b$$

作回归分析。其中 a 与 b 为回归系数。

为确定回归系数 a 与 b，一般采用最小二乘法，即使得

$$q=\sum_{i=1}^{n}[y_i-(ax_i+b)]^2$$

达到最小。根据极值原理，a 与 b 满足下列方程：

$$\frac{\partial q}{\partial a}=2\sum_{i=1}^{n}[y_i-(ax_i+b)](-x_i)=0$$

$$\frac{\partial q}{\partial b}=2\sum_{i=1}^{n}[y_i-(ax_i+b)](-1)=0$$

从而解得

$$a=\frac{dxy}{dx},\ b=\bar{y}-a\bar{x}$$

其中

$$\bar{x}=\sum_{i=1}^{n}\frac{x_i}{n},\ \bar{y}=\sum_{i=1}^{n}\frac{y_i}{n}$$

$$dx=\sum_{i=1}^{n}(x_i-\bar{x})^2,\ dxy=\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})$$

最后计算出以下几个量：

(1) 偏差平方和

$$q=\sum_{i=1}^{n}[y_i-(ax_i+b)]^2$$

(2) 平均标准偏差

$$s=\sqrt{q/n}$$

(3) 回归平方和

$$p=\sum_{i=1}^{n}[(ax_i+b)-\bar{y}]^2$$

(4) 最大偏差

$$umax=\max_{1\leqslant i\leqslant n}|y_i-(ax_i+b)|$$

(5) 最小偏差

$$umin=\min_{1\leqslant i\leqslant n}|y_i-(ax_i+b)|$$

(6) 偏差的平均值

$$u=\sum_{i=1}^{n}|y_i-(ax_i+b)|/n$$

三、子程序语句

SUBROUTINE ISQT1(X,Y,N,A,B,Q,S,P,UMAX,UMIN,U)

四、形参说明

X,Y——均匀双精度实型一维数组,长度为N,输入参数。存放N个观测值(x_i,y_i) $(i=1,2,\cdots,N)$。

N——整型变量,输入参数。观测点数。

A,B——均为双精度实型变量,输出参数。回归系数,即线性函数为

$$y=Ax+B$$

Q,S,P——均为双精度实型变量,输出参数。分别为偏差平方和、平均标准偏差与回归平方和。

UMAX,UMIN——均为双精度实型变量,输出参数。分别为最大偏差与最小偏差。

U——双精度实型变量,输出参数。偏差的平均值。

五、子程序(文件名:ISQT1.FOR)

```
      SUBROUTINE ISQT1(X,Y,N,A,B,Q,S,P,UMAX,UMIN,U)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y,A,B,Q,S,P,UMAX,UMIN,U
      DOUBLE PRECISION XX,YY,DX,DXY
      XX=0.0
      YY=0.0
      DO 10 I=1,N
        XX=XX+X(I)/N
        YY=YY+Y(I)/N
10    CONTINUE
      DX=0.0
      DXY=0.0
      DO 20 I=1,N
        Q=X(I)-XX
        DX=DX+Q*Q
        DXY=DXY+Q*(Y(I)-YY)
20    CONTINUE
      A=DXY/DX
      B=YY-A*XX
      Q=0.0
      U=0.0
      P=0.0
      UMAX=0.0
      UMIN=1.0D+30
```

```
      DO 30 I=1,N
        S=A*X(I)+B
        Q=Q+(Y(I)-S)*(Y(I)-S)
        P=P+(S-YY)*(S-YY)
        DX=ABS(Y(I)-S)
        IF (DX.GT.UMAX) UMAX=DX
        IF (DX.LT.UMIN) UMIN=DX
        U=U+DX/N
30    CONTINUE
      S=SQRT(Q/N)
      RETURN
      END
```

六、例

给定 11 个观测值如下：

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
y	2.75	2.84	2.965	3.01	3.20	3.25	3.38	3.43	3.55	3.66	3.74

求回归系数 A 与 B、偏差平方和 Q、平均标准偏差 S、回归平方和 P、偏差最大值 UMAX，偏差最小值 UMIN、偏差平均值 U。

主程序(文件名:ISQT10.FOR)为

```
      DIMENSION X(11),Y(11)
      DOUBLE PRECISION X,Y,A,B,Q,S,P,UMAX,UMIN,U
      DATA X/0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0/
      DATA Y/2.75,2.84,2.965,3.01,3.20,3.25,3.38,3.43,3.55,
     *         3.66,3.74/
      N=11
      CALL ISQT1(X,Y,N,A,B,Q,S,P,UMAX,UMIN,U)
      WRITE(*,*)
      WRITE(*,10) A,B
10    FORMAT(1X,'A=',D13.6,3X,'B=',D13.6)
      WRITE(*,*)
      WRITE(*,20) Q,S,P
      WRITE(*,*)
      WRITE(*,30) UMAX,UMIN,U
      WRITE(*,*)
20    FORMAT(1X,'Q=',D13.6,3X,'S=',D13.6,3X,'P=',D13.6)
30    FORMAT(1X,'UMAX=',D13.6,3X,'UMIN=',D13.6,3X,'U=',D13.6)
      END
```

运行结果为

A= .100045D+01 B= .275205D+01

Q= .586796D−02 S= .230966D−01 P= .110100D+01

UMAX= .477728D−01 UMIN= .204541D−02 U= .174298D−01

9.3 多元线性回归分析

一、功能

对随机变量 y 及 m 个自变量 $x_1,x_2,\cdots,x_m$ 的 n 组观测值 $(x_{1i},x_{2i},\cdots,x_{mi})$ $(i=1,2,\cdots,n)$ 作线性回归分析。

二、方法说明

设随机变量 y 及 m 个自变量 $x_1,x_2,\cdots,x_m$。已知 n 组观测数据 $(x_{1i},x_{2i},\cdots,x_{mi})$ $(i=1,2,\cdots,n)$，用线性表达式

$$y = a_1 + a_2x_1 + a_3x_2 + \cdots + a_{m+1}x_m$$

对观测数据进行回归分析。其中 $a_1,a_2,\cdots,a_{m+1}$ 为回归系数。

与一元线性回归分析一样(参看 9.2 节的方法说明)，根据最小二乘原理，为使

$$q = \sum_{i=1}^{n}[y_i - (a_1 + a_2x_{1i} + a_3x_{2i} + \cdots + a_{m+1}x_{mi})]^2$$

达到最小，回归系数 $a_1,a_2,a_3,\cdots,a_{m+1}$ 应满足下列方程组：

$$(CC^T)\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{m+1} \end{bmatrix} = C\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

其中

$$C=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \end{bmatrix}$$

采用乔里斯基(Cholesky)分解法解出回归系数。

为了衡量回归效果，还要计算以下五个量：

(1) 偏差平方和 q

$$q = \sum_{i=1}^{n}[y_i - (a_1 + a_2x_{1i} + \cdots + a_{m+1}x_{mi})]^2$$

(2) 平均标准偏差 s

$$s=\sqrt{q/n}$$

(3) 复相关系数 r

$$r=\sqrt{1-q/dyy}$$

其中 $dyy=\sum_{i=1}^{n}(y_i-\bar{y})^2$，$\bar{y}=\sum_{i=1}^{n}y_i/n$。

当 r 接近于 1 时，说明相对误差 q/dyy 接近于零，线性回归效果好。

(4) 偏相关系数 V_j

$$V_j=\sqrt{1-q/Q_j},\ j=1,2,\cdots,m$$

其中

$$Q_j=\sum_{i=1}^{n}[y_i-(a_1+\sum_{\substack{k=1\\k\neq j}}^{m}a_{k+1}x_{ki})]^2$$

当 V_j 越大时，说明 x_j 对于 y 的作用越显著，此时不能把 x_j 剔除。

(5) 回归平方和 u

$$u=\sum_{i=1}^{n}[\bar{y}-(a_1+a_2x_{1i}+\cdots+a_{m+1}x_{mi})]^2$$

三、子程序语句

SUBROUTINE ISQT2(X,Y,M,MM,N,A,Q,S,R,V,U,B)

四、形参说明

X——双精度实型二维数组，体积为 M×N，输入参数。其中每一列存放 m 个自变量的一组观测值，即每一列为

$$(x_{1i},x_{2i},\cdots,x_{mi})^T,\ i=1,2,\cdots,N$$

Y——双精度实型一维数组，长度为 N，输入参数。存放随机变量 y 的 N 个观测值。

M——整型变量，输入参数。自变量个数。

MM——整型变量，输入参数。回归系数的个数。MM=M+1。

N——整型变量，输入参数。观测数据的组数。

A——双精度实型一维数组，长度为 MM=M+1，输出参数。存放 M+1 个回归系数 $a_1,a_2,\cdots,a_{m+1}$。

Q,S,R——均为双精度实型变量，输出参数。分别为偏差平方和、平均标准偏差与复相关系数。

V——双精度实型一维数组，长度为 M，输出参数。M 个自变量的偏相关系数。

U——双精度实型变量，输出参数。回归平方和。

B——双精度实型二维数组，体积为 MM×MM。本子程序的工作数组，用于存放 CC^T（见方法说明）。

五、子程序(文件名:ISQT2.FOR)

```
      SUBROUTINE ISQT2(X,Y,M,MM,N,A,Q,S,R,V,U,B)
      DIMENSION X(M,N),Y(N),A(MM),B(MM,MM),V(M)
      DOUBLE PRECISION X,Y,A,B,V,Q,S,R,U,YY,DYY,P,PP
      B(1,1)=N
      DO 20 J=2,MM
        B(1,J)=0.0
        DO 10 I=1,N
```

```
10     B(1,J)=B(1,J)+X(J-1,I)
       B(J,1)=B(1,J)
20    CONTINUE
      DO 50 I=2,MM
        DO 40 J=I,MM
          B(I,J)=0.0
          DO 30 K=1,N
30        B(I,J)=B(I,J)+X(I-1,K)*X(J-1,K)
          B(J,I)=B(I,J)
40      CONTINUE
50    CONTINUE
      A(1)=0.0
      DO 60 I=1,N
60    A(1)=A(1)+Y(I)
      DO 80 I=2,MM
        A(I)=0.0
        DO 70 J=1,N
70      A(I)=A(I)+X(I-1,J)*Y(J)
80    CONTINUE
      CALL ACHOL(B,MM,1,A,L)
      YY=0.0
      DO 90 I=1,N
90    YY=YY+Y(I)/N
      Q=0.0
      DYY=0.0
      U=0.0
      DO 110 I=1,N
        P=A(1)
        DO 100 J=1,M
100     P=P+A(J+1)*X(J,I)
        Q=Q+(Y(I)-P)*(Y(I)-P)
        DYY=DYY+(Y(I)-YY)*(Y(I)-YY)
        U=U+(YY-P)*(YY-P)
110   CONTINUE
      S=SQRT(Q/N)
      R=SQRT(1.0-Q/DYY)
      DO 150 J=1,M
        P=0.0
        DO 140 I=1,N
          PP=A(1)
          DO 130 K=1,M
            IF (K.NE.J) PP=PP+A(K+1)*X(K,I)
```

```
130         CONTINUE
            P=P+(Y(I)-PP)*(Y(I)-PP)
140       CONTINUE
          V(J)=SQRT(1.0-Q/P)
150     CONTINUE
        RETURN
        END
```

六、例

对随机变量 y 及自变量 x_1, x_2, x_3 的下列五组观测数据作多元线性回归分析：

i	x_{1i}	x_{2i}	x_{3i}	y_i
1	1.1	2.0	3.2	10.1
2	1.0	2.0	3.2	10.2
3	1.2	1.8	3.0	10.0
4	1.1	1.9	2.9	10.1
5	0.9	2.1	2.9	10.0

其中 N=5，M=3，MM=4。

主程序(文件名：ISQT20.FOR)为

```
      DIMENSION X(3,5),Y(5),A(4),B(4,4),V(3)
      DOUBLE PRECISION X,Y,A,B,V,Q,S,R,U
      DATA X/1.1,2.0,3.2,1.0,2.0,3.2,1.2,1.8,3.0,
     *       1.1,1.9,2.9,0.9,2.1,2.9/
      DATA Y/10.1,10.2,10.0,10.1,10.0/
      N=5
      M=3
      MM=4
      CALL ISQT2(X,Y,M,MM,N,A,Q,S,R,V,U,B)
      WRITE(*,*)
      WRITE(*,10) (I,A(I),I=1,MM)
10    FORMAT(1X,'A(',I2,' )=',D13.6)
      WRITE(*,*)
      WRITE(*,20) Q,S,R
20    FORMAT(1X,'Q=',D13.6,3X,'S=',D13.6,3X,'R=',D13.6)
      WRITE(*,*)
      WRITE(*,30) (I,V(I),I=1,M)
30    FORMAT(1X,'V(',I2,' )=',D13.6)
      WRITE(*,*)
      WRITE(*,40) U
```

```
40      FORMAT(1X,'U=',D13.6)
        WRITE(*,*)
        END
```

运行结果为

```
A( 1 )=    .107800D+02
A( 2 )= -.799993D+00
A( 3 )= -.699994D+00
A( 4 )=    .499899D+00
Q=   .120001D-01     S=    .489899D-01     R=    .755927D+00
V( 1 )=    .998351D+00
V( 2 )=    .999365D+00
V( 3 )=    .999482D+00
U= .159999D-01
```

由复相关系数 R=0.755927 可以看出，线性回归效果并非特别好。

由求得的偏相关系数 V_1，V_2，V_3 可知，自变量 x_1，x_2，x_3 对 y 的作用均比较显著，即这三个变量均不能剔除。

七、附注

本子程序需要调用乔里斯基分解法(即平方根法)求解对称正定方程组的子程序 ACHOL(参看 1.8 节)。

9.4 逐步回归分析

一、功能

对多元线性回归进行因子筛选，最后给出一定显著性水平下各因子均为显著的回归方程中的诸回归系数，偏回归平方和、估计的标准偏差、复相关系数及 F-检验值、各回归系数的标准偏差、应变量条件期望值的估计值及残差。

二、方法说明

设 n 个自变量为 $x_j(j=1,2,\cdots,n)$，因变量为 y，有 k 个观测点为

$$(x_{i1},x_{i2},\cdots,x_{in},y_i),\ i=1,2,\cdots,k$$

根据最小二乘原理，y 的估计值为

$$\hat{y}=b_0+b_1'x_1'+b_2'x_2'+\cdots+b_l'x_l'$$

其中 $l\leqslant n$，且各 $x_i'(i=1,2,\cdots,l)$ 是从 n 个 x 中按一定显著性水平筛选出的统计检验为显著的因子。筛选的过程如下。

(1) 首先作出 $(n+1)\times(n+1)$ 阶规范化的系数初始相关阵

$$R=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} & r_{1y} \\ r_{21} & r_{22} & \cdots & r_{2n} & r_{2y} \\ \vdots & \vdots & & \vdots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} & r_{ny} \\ r_{y1} & r_{y2} & \cdots & r_{yn} & r_{yy} \end{bmatrix}$$

式中

$$r_{ij}=\frac{d_{ij}}{\sqrt{d_i d_j}},\ i,j=1,2,\cdots,n\ ,n+1\ 对应\ y$$

$$d_{ij}=\sum_{t=1}^{k}(x_{it}-\bar{x}_i)(x_{tj}-\bar{x}_j)$$

$$d_i=\sum_{t=1}^{k}(x_{it}-\bar{x}_i)^2,\ d_j=\sum_{t=1}^{k}(x_{tj}-\bar{x}_j)^2$$

其中

$$\bar{x}_i=\sum_{t=1}^{k}x_{it}/k\ ,\ i=1,2,\cdots,n+1$$

(2) 计算偏回归平方和

$$V_i=\frac{r_{iy}r_{yi}}{r_{ii}}\ ,\ i=1,2,\cdots,n$$

(3) 若 $V_i<0$，则对应的 x_i 是已被选入回归方程的因子。

从所有 $V_i<0$ 的 V_i 中选出 $V_{min}=\min\{|V_i|\}$，其对应的因子为 x_{min}。且检验因子 x_{min} 的显著性。若

$$\frac{\varphi V_{min}}{r_{yy}}<F_2$$

则剔除因子 x_{min}，并对系数相关阵 R 进行该因子的消元变换。转(2)。

(4) 若 $V_i>0$，则对应的 x_i 为尚待选入的因子。

从所有 $V_i>0$ 的 V_i 中选出 $V_{max}=\max\{|V_i|\}$，其对应的因子为 x_{max}。且检验因子 x_{max} 的显著性。若

$$\frac{(\varphi-1)V_{max}}{r_{yy}-V_{max}}\geqslant F_1$$

则因子 x_{max} 应选入，并对系数相关阵 R 进行该因子的消元变换。转(2)。

上述过程一直进行到无因子可剔可选为止。

在上述步骤中，φ 为相应的残差平方和之自由度；F_1 及 F_2 均是 F-Y 分布值，它们取决于观测点数、已选入的因子数及选定的取舍显著性水平，一般 $F_1>F_2$，且在观测点较多时可取为常数。

当要剔除或选入某个因子 x_l 时，均需要对系数相关阵 R 进行消元变换，其算法如下：

$$r_{ij}=r_{ij}-\frac{r_{lj}}{r_{ll}}\cdot r_{il},\ i,j=1,2,\cdots,n+1;i,j\neq l$$

$$r_{lj}=r_{lj}/r_{ll},\ j=1,2,\cdots,n+1\ ;\ j\neq l$$

$$r_{il}=-\ r_{il}/r_{ll},i=1,2,\cdots,n+1\ ;\ i\neq l$$

$$r_{ll}=1/r_{ll}$$

当筛选结束时，就可得出规格化回归方程的各回归系数

$$b_0,b_1,b_2,\cdots,b_n$$

其中值为 0 的系数表示对应的自变量可剔除。回归模型的各有关值由下列各式计算：

选入回归方程各因子的回归系数 $b_i=\sqrt{\frac{d_y}{d_i}}\cdot r_{iy}$

回归方程的常数项 $b_0=\bar{y}-\sum_{j=1}^{t}b_j\bar{x}_j$

各因子的偏回归平方和 $V_i=r_{iy}r_{yi}/r_{ii}$

估计的标准偏差 $s=d_y\sqrt{r_{yy}/\varphi}$

各回归系数的标准偏差 $S_i=s\sqrt{r_{ii}/d_i}$

复相关系数 $C=\sqrt{1-r_{yy}}$

F-Y 检验值 $F=\dfrac{\varphi(1-r_{yy})}{(k-\varphi-1)r_{yy}}$

残差平方和 $q=d_yr_{yy}$

因变量条件期望值的估计值 $YE_i=b_0+\sum_{j=1}^{t}b_jx_{ij}$

残差 $YR_i=y_i-YE_i$

三、子程序语句

SUBROUTINE ISQT3(M,K,X,F1,F2,EPS,XX,B,V,S,C,F,YE,YR,R)

四、形参说明

M——整型变量,输入参数。M=N+1,其中N为自变量个数。

K——整型变量,输入参数。观测点数。

X——双精度实型二维数组,体积为K×M,输入参数。存放K个观测点数据,其中前N列中的每一列对应一个自变量因子的K个观测值,最后一列(即第M列)存放因变量 y 的K个观测值。

F1,F2——均为实型变量,输入参数。分别存放欲选入与剔除因子时显著性检验的F-分布值。

EPS——实型变量,输入参数。置防止系数相关阵退化的判据。

XX——双精度实型一维数组,长度为M,输出参数。前N个存放自变量因子的算术平均值,最后一个(即第M个)存放因变量 y 的算术平均值。

B——双精度实型一维数组,长度为M,输出参数。前N个存放各因子的回归系数 b_1, $b_2,\cdots,b_n$,最后一个存放回归方程的常数项 b_0。

V——双精度实型一维数组,长度为M,输出参数。前N个存放各因子的偏回归平方和,最后一个存放残差平方和 q。

S——双精度实型一维数组,长度为M,输出参数。前N个存放各因子回归系数的标准偏差,最后一个存放估计的标准偏差。

C——双精度实型变量,输出参数。存放复相关系数。

F——双精度实型变量,输出参数。存放F-Y检验值。

YE——双精度实型一维数组,长度为K,输出参数。存放因变量条件期望值的K个估计值(对应于K个观测值)。

YR——双精度实型一维数组,长度为 K,输出参数。存放因变量的 K 个观测值的残差。

R——双精度实型二维数组,体积为 M×M,输出参数。存放最终的规格化的系数相关阵。

五、子程序(文件名:ISQT3.FOR)

```
      SUBROUTINE ISQT3(M,K,X,F1,F2,EPS,XX,B,V,S,C,F,YE,YR,R)
      DIMENSION X(K,M),B(M),V(M),S(M),YE(M),YR(M),R(M,M),XX(M)
      DOUBLE PRECISION X,B,V,S,YE,YR,R,XX,F,C,FMI,FMX,
     *                  PHI,VMI,VMX,Z,Q,SD
      DO 50 J=1,M
        Z=0.0
        DO 40 I=1,K
40       Z=Z+X(I,J)/K
        XX(J)=Z
50    CONTINUE
      DO 80 I=1,M
        DO 70 J=1,I
          Z=0.0
          DO 60 II=1,K
60        Z=Z+(X(II,I)-XX(I))*(X(II,J)-XX(J))
          R(I,J)=Z
70      CONTINUE
80    CONTINUE
      DO 90 I=1,M
90    YE(I)=SQRT(R(I,I))
      DO 100 I=1,M
      DO 100 J=1,I
      R(I,J)=R(I,J)/(YE(I)*YE(J))
      R(J,I)=R(I,J)
100   CONTINUE
      PHI=K-1
      SD=YE(M)/SQRT(K-1.0)
105   VMI=1.0E+35
      VMX=0.0
      IMI=0
      IMX=0
      DO 110 I=1,M
        V(I)=0.0
        B(I)=0.0
        S(I)=0.0
110   CONTINUE
```

```
      I=0
120   I=I+1
      IF (R(I,I).GE.EPS) THEN
        V(I)=R(I,M)*R(M,I)/R(I,I)
        IF (V(I).GE.0.0) THEN
          IF (V(I).GT.VMX) THEN
            VMX=V(I)
            IMX=I
          END IF
        ELSE
          B(I)=R(I,M)*YE(M)/YE(I)
          S(I)=SQRT(R(I,I))*SD/YE(I)
          IF (ABS(V(I)).LT.VMI) THEN
            VMI=ABS(V(I))
            IMI=I
          END IF
        END IF
      END IF
      IF (I.NE.M-1) GOTO 120
      IF (PHI.NE.M-2) THEN
        Z=0.0
        DO 130 I=1,M-1
130     Z=Z+B(I)*XX(I)
        B(M)=XX(M)-Z
        S(M)=SD
        V(M)=Q
      ELSE
        B(M)=XX(M)
        S(M)=SD
      END IF
      FMI=VMI*PHI/R(M,M)
      FMX=(PHI-1.0)*VMX/(R(M,M)-VMX)
      IF ((FMI.LT.F2).OR.(FMX.GE.F1)) THEN
        IF (FMI.LT.F2) THEN
          PHI=PHI+1.0
          L=IMI
        ELSE
          PHI=PHI-1.0
          L=IMX
        END IF
        DO 150 I=1,M
          IF (I.NE.L) THEN
```

```
                DO 140 J=1,M
                  IF (J.NE.L) THEN
                    R(I,J)=R(I,J)-(R(L,J)/R(L,L))*R(I,L)
                  END IF
140             CONTINUE
              END IF
150         CONTINUE
            DO 160 J=1,M
              IF (J.NE.L) THEN
                R(L,J)=R(L,J)/R(L,L)
              END IF
160         CONTINUE
            DO 170 I=1,M
              IF (I.NE.L) THEN
                R(I,L)=-R(I,L)/R(L,L)
              END IF
170         CONTINUE
            R(L,L)=1.0/R(L,L)
            Q=R(M,M)*YE(M)*YE(M)
            SD=SQRT(R(M,M)/PHI)*YE(M)
            C=SQRT(1.0-R(M,M))
            F=(PHI*(1.0-R(M,M)))/((K-PHI-1.0)*R(M,M))
            GOTO 105
          END IF
          DO 190 I=1,K
            Z=0.0
            DO 180 J=1,M-1
180         Z=Z+B(J)*X(I,J)
            YE(I)=B(M)+Z
            YR(I)=X(I,M)-YE(I)
190       CONTINUE
          RETURN
          END
```

六、例

设自变量有 4 个，分别为 x_1,x_2,x_3,x_4，因变量为 y。

13 个观测点值如下：

k	x_1	x_2	x_3	x_4	y
1	7.000	26.000	6.000	60.000	78.500
2	1.000	29.000	15.000	52.000	74.300

续表

k	x_1	x_2	x_3	x_4	y
3	11.000	56.000	8.000	20.000	104.300
4	11.000	31.000	8.000	47.000	87.600
5	7.000	52.000	6.000	33.000	95.900
6	11.000	55.000	9.000	22.000	109.200
7	3.000	71.000	17.000	6.000	102.700
8	1.000	31.000	22.000	44.000	72.500
9	2.000	54.000	18.000	22.000	93.100
10	21.000	47.000	4.000	26.000	115.900
11	1.000	40.000	23.000	34.000	83.800
12	11.000	66.000	9.000	12.000	113.300
13	10.000	68.000	8.000	12.000	109.400

对于不同的 F1 及 F2 值进行逐步回归分析。

主程序(文件名:ISQT30.FOR)为

```
      DIMENSION X(13,5),XX(5),B(5),V(5),S(5),
     *          YE(13),YR(13),R(5,5)
      DOUBLE PRECISION X,XX,B,V,S,YE,YR,R,F,C
      DATA X/7.0,1.0,11.0,11.0,7.0,11.0,3.0,1.0,2.0,21.0,
     *       1.0,11.0,10.0,26.0,29.0,56.0,31.0,52.0,55.0,
     *       71.0,31.0,54.0,47.0,40.0,66.0,68.0,6.0,15.0,
     *       8.0,8.0,6.0,9.0,17.0,22.0,18.0,4.0,23.0,9.0,
     *       8.0,60.0,52.0,20.0,47.0,33.0,22.0,6.0,44.0,
     *       22.0,26.0,34.0,12.0,12.0,78.5,74.3,104.3,87.6,
     *       95.9,109.2,102.7,72.5,93.1,115.9,83.8,
     *       113.3,109.4/
      EPS=1.0E-30
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(1X,'F1,F2=?')
      READ(*,*) F1,F2
      M=5
      K=13
      CALL ISQT3(M,K,X,F1,F2,EPS,XX,B,V,S,C,F,YE,YR,R)
      WRITE(*,*)
      WRITE(*,20)
20    FORMAT(1X,'ORIGINAL X(I) AND Y VALUES:')
```

```
      WRITE(*,40)
40    FORMAT(6X,'X1',8X,'X2',8X,'X3',8X,'X4',8X,'Y')
      DO 30 I=1,13
30    WRITE(*,50) (X(I,J),J=1,5)
50    FORMAT(1X,5F10.3)
      WRITE(*,*)
      WRITE(*,60)
60    FORMAT(1X,'MEAN OF X(I) AND Y:')
      WRITE(*,40)
      WRITE(*,50) (XX(I),I=1,5)
      WRITE(*,*)
      WRITE(*,70)
70    FORMAT(1X,'REGRESSION COEFFI B(I) AND B0:')
      WRITE(*,80)
80    FORMAT(4X,'B0',11X,'B1',11X,'B2',11X,'B3',11X,'B4')
      WRITE(*,90) B(5),B(1),B(2),B(3),B(4)
90    FORMAT(1X,5D13.6)
      WRITE(*,*)
      WRITE(*,100)
100   FORMAT(1X,'STANDARD PARTIAL SUM OF SQUARE OF REGRESSION FOR
     * X(I)')
      WRITE(*,110)
110   FORMAT(1X,'AND SUM OF SQUARE OF RESIDUALS:')
      WRITE(*,120)
120   FORMAT(5X,'V1',11X,'V2',11X,'V3',11X,'V4',11X,'Q')
      WRITE(*,90) (V(I),I=1,5)
      WRITE(*,*)
      WRITE(*,140)
140   FORMAT(1X,'STANDARD DEVIATION OF REGRESSION COEFFI AND
     * REGRES SION EQUATION:')
      WRITE(*,150)
150   FORMAT(5X,'S1',11X,'S2',11X,'S3',11X,'S4',11X,'SS')
      WRITE(*,90) (S(I),I=1,5)
      WRITE(*,*)
      WRITE(*,170) C
170   FORMAT(1X,'MULTI-CORRELATION COEFFI C IS:',D13.6)
      WRITE(*,*)
      WRITE(*,180) F
180   FORMAT(1X,'THE F VALUE=',D13.6)
      WRITE(*,*)
      WRITE(*,190)
190   FORMAT(1X,'ESTIMATED VALUES AND RESIDUALS:')
      WRITE(*,200)
200   FORMAT(5X,'YE',20X,'YR')
      DO 210 I=1,13
```

```
210     WRITE(*,220) YE(I),YR(I)
220     FORMAT(1X,D13.6,7X,D13.6)
        WRITE(*,*)
        WRITE(*,230)
230     FORMAT(1X,'MATRIX R:')
        WRITE(*,90) ((R(I,J),J=1,M),I=1,M)
        WRITE(*,*)
        END
```

当取 $\alpha=0.25$，查 *F*-*Y* 分布表得 F1=1.46，F2=1.45。

运行结果为

```
F1,F2=?
1.46,1.45
ORIGINAL X(I) AND Y VALUES:
```

X1	X2	X3	X4	Y
7.000	26.000	6.000	60.000	78.500
1.000	29.000	15.000	52.000	74.300
11.000	56.000	8.000	20.000	104.300
11.000	31.000	8.000	47.000	87.600
7.000	52.000	6.000	33.000	95.900
11.000	55.000	9.000	22.000	109.200
3.000	71.000	17.000	6.000	102.700
1.000	31.000	22.000	44.000	72.500
2.000	54.000	18.000	22.000	93.100
21.000	47.000	4.000	26.000	115.900
1.000	40.000	23.000	34.000	83.800
11.000	66.000	9.000	12.000	113.300
10.000	68.000	8.000	12.000	109.400

MEAN OF X(I) AND Y:

X1	X2	X3	X4	Y
7.462	48.154	11.769	30.000	95.423

REGRESSION COEFFI B(I) AND B0:

B0	B1	B2	B3	B4
.716483D+02	.145194D+01	.416110D+00	.000000D+00	−.236540D+00

STANDARD PARTIAL SUM OF SQUARE OF REGRESSION FOR X(I)

AND SUM OF SQUARE OF RESIDUALS:

V1	V2	V3	V4	Q
−.302275D+00	−.986440D−02	.401693D−04	−.365708D−02	.479727D+02

STANDARD DEVIATION OF REGRESSION COEFFI AND REGRES SION EQUATION:

S1	S2	S3	S4	SS
.116998D+00	.185610D+00	.000000D+00	.173288D+00	.230874D+01

MULTI-CORRELATION COEFFI C IS: .991128D+00

THE F VALUE= .166832D+03

ESTIMATED VALUES AND RESIDUALS:

YE	YR
.784383D+02	.616853D-01
.728673D+02	.143267D+01
.106191D+03	-.189096D+01
.894016D+02	-.180164D+01
.956438D+02	.256248D+00
.105302D+03	.389822D+01
.104129D+03	-.142868D+01
.755919D+02	-.309188D+01
.918182D+02	.128177D+01
.115546D+03	.353833D+00
.817023D+02	.209773D+01
.112244D+03	.105562D+01
.111625D+03	-.222467D+01

MATRIX R:

.106633D+01	.204390D+00	-.893654D+00	.460588D+00	.567737D+00
.204390D+00	.187803D+02	-.224227D+01	.183226D+02	.430414D+00
.893654D+00	.224227D+01	.213363D-01	.237143D+01	.925779D-03
.460588D+00	.183226D+02	-.237143D+01	.189401D+02	-.263183D+00
-.567737D+00	-.430414D+00	.925779D-03	.263183D+00	.176645D-01

当取 $\alpha=0.05$，查 F-Y 分布表得 F1=4.75，F2=4.67。

运行结果为

F1,F2=?

4.75,4.67

ORIGINAL X(I) AND Y VALUES:

X1	X2	X3	X4	Y
7.000	26.000	6.000	60.000	78.500
1.000	29.000	15.000	52.000	74.300
11.000	56.000	8.000	20.000	104.300
11.000	31.000	8.000	47.000	87.600
7.000	52.000	6.000	33.000	95.900
11.000	55.000	9.000	22.000	109.200
3.000	71.000	17.000	6.000	102.700
1.000	31.000	22.000	44.000	72.500
2.000	54.000	18.000	22.000	93.100
21.000	47.000	4.000	26.000	115.900
1.000	40.000	23.000	34.000	83.800
11.000	66.000	9.000	12.000	113.300
10.000	68.000	8.000	12.000	109.400

MEAN OF X(I) AND Y:

X1	X2	X3	X4	Y
7.462	48.154	11.769	30.000	95.423

REGRESSION COEFFI B(I) AND B0:

B0	B1	B2	B3	B4
.525774D+02	.146831D+01	.662250D+00	.000000D+00	.000000D+00

STANDARD PARTIAL SUM OF SQUARE OF REGRESSION FOR X(I)

AND SUM OF SQUARE OF RESIDUALS:

V1	V2	V3	V4	Q
-.312410D+00	-.444730D+00	.360630D-02	.365708D-02	.579045D+02

STANDARD DEVIATION OF REGRESSION COEFFI AND REGRES SION EQUATION:

S1	S2	S3	S4	SS
.121301D+00	.458547D-01	.000000D+00	.000000D+00	.240633D+01

MULTI--CORRELATION COEFFI C IS: .989282D+00

THE F VALUE= .229504D+03

ESTIMATED VALUES AND RESIDUALS:

YE	YR
.800740D+02	-.157400D+01
.732509D+02	.104908D+01
.105815D+03	-.151474D+01
.892585D+02	-.165848D+01
.972925D+02	.139251D+01
.105152D+03	.404751D+01
.104002D+03	-.130205D+01
.745754D+02	-.207542D+01
.912755D+02	.182451D+01
.114538D+03	.136246D+01
.805357D+02	.326433D+01
.112437D+03	.862758D+00
.112293D+03	-.289344D+01

MATRIX R:

.105513D+01	-.241181D+00	-.835985D+00	-.243182D-01	.574137D+00
-.241181D+00	.105513D+01	.518466D-01	-.967396D+00	.685017D+00
.835985D+00	-.518466D-01	.318256D+00	-.125207D+00	.338781D-01
.243182D-01	.967396D+00	-.125207D+00	.527981D-01	-.138956D-01
-.574137D+00	-.685017D+00	.338781D-01	-.138956D-01	.213216D-01

当取 $\alpha=0.01$，查 F-Y 分布表得 F1=9.33，F2=9.07，

运行结果为

F1,F2=?

9.33,9.07

ORIGINAL X(I) AND Y VALUES:

X1	X2	X3	X4	Y
7.000	26.000	6.000	60.000	78.500
1.000	29.000	15.000	52.000	74.300
11.000	56.000	8.000	20.000	104.300
11.000	31.000	8.000	47.000	87.600
7.000	52.000	6.000	33.000	95.900
11.000	55.000	9.000	22.000	109.200
3.000	71.000	17.000	6.000	102.700
1.000	31.000	22.000	44.000	72.500
2.000	54.000	18.000	22.000	93.100
21.000	47.000	4.000	26.000	115.900
1.000	40.000	23.000	34.000	83.800
11.000	66.000	9.000	12.000	113.300
10.000	68.000	8.000	12.000	109.400

MEAN OF X(I) AND Y:

X1	X2	X3	X4	Y
7.462	48.154	11.769	30.000	95.423

REGRESSION COEFFI B(I) AND B0:

B0	B1	B2	B3	B4
.103097D+03	.143996D+01	.000000D+00	.000000D+00	−.613954D+00

STANDARD PARTIAL SUM OF SQUARE OF REGRESSION FOR X(I)

AND SUM OF SQUARE OF RESIDUALS:

V1	V2	V3	V4	Q
−.297929D+00	.986440D−02	.881004D−02	−.438523D+00	.747621D+02

STANDARD DEVIATION OF REGRESSION COEFFI AND REGRES SION EQUATION:

S1	S2	S3	S4	SS
.138417D+00	.000000D+00	.000000D+00	.486446D−01	.273427D+01

MULTI−CORRELATION COEFFI C IS: .986139D+00

THE F VALUE= .176627D+03

ESTIMATED VALUES AND RESIDUALS:

YE	YR
.763399D+02	.216013D+01
.726118D+02	.168825D+01
.106658D+03	−.235786D+01
.900811D+02	−.248110D+01
.929166D+02	.298338D+01
.105430D+03	.377005D+01
.103734D+03	−.103354D+01
.775234D+02	−.502338D+01

.924703D+02	.629680D+00
.117374D+03	−.147371D+01
.836629D+02	.137086D+00
.111569D+03	.173052D+01
.110130D+03	−.729520D+00

MATRIX R:

.106411D+01	−.106832D−01	−.869251D+00	.261179D+00	.563052D+00
.106832D−01	.532473D−01	−.119395D+00	.975626D+00	.229184D−01
.869251D+00	−.119395D+00	.289051D+00	.183816D+00	−.504633D−01
.261179D+00	−.975626D+00	−.183816D+00	.106411D+01	−.683107D+00
−.563052D+00	.229184D−01	−.504633D−01	.683107D+00	.275290D−01

七、附注

(1) 本子程序适用于自变量个数较多，且观测点较多的问题。

(2) F1 与 F2 的选取一般可以查 F-Y 分布表：假设选入单个因子的显著性水平取为 α，则可以在 F-Y 分布表中查到当 $m=1$、n 为观测点数时的 F_α 值。此时取 $F2=F_\alpha$。F1 取为当 $m=1$，n 为观测点数减 1 时的 F_α 值。

9.5 半对数数据相关

一、功能

对于给定的 n 个数据点 $(x_i, y_i)(i=1,2,\cdots,n)$ 用函数

$$y=be^{ax} \quad , t>0$$

作拟合。

二、方法说明

设给定 n 个数据点 (x_i, y_i) $(i=1,2,\cdots,n)$，且 $y_i>0(i=1,2,\cdots,n)$，用函数

$$y=be^{ax} \quad , t>0$$

进行拟合。为了求拟合参数，两边取对数，即

$$\log_e y=\log_e b+ax$$

令

$$\bar{y}=\bar{a}\bar{x}+\bar{b}$$

其中

$$\tilde{y}=\log_e y,\ \tilde{x}=x,\ \tilde{a}=a,\ \tilde{b}=\log_e b$$

此时，问题化为对 n 个数据点 $(\tilde{x}_i, \tilde{y}_i)$ 作线性拟合。求出 $\tilde{a}$ 与 $\tilde{b}$ 后，就可以得到

$$a=\tilde{a}\ ,\ b=e^{\tilde{b}}$$

关于一元线性拟合参看 9.2 节的方法说明。

三、子程序语句

SUBROUTINE ILOG1(X,Y,N,T,A,B,Q,S,UMAX,UMIN,U)

四、形参说明

X,Y——均为双精度实型一维数组,长度为N,输入参数。存放N个数据点(x_i,y_i) $(i=1,2,\cdots,N)$,要求所有的$y_i>0(i=1,2,\cdots,N)$。

N——整型变量,输入参数。数据点数。

T——双精度实型变量,输入参数。指数函数的底,要求T>0。

A,B——均为双精度实型变量,输出参数。拟合函数的参数。即拟合函数形式为

$$y=BT^{Ax}$$

Q——双精度实型变量,输出参数。偏差平方和,即

$$Q=\sum_{i=1}^{N}(y_i-BT^{Ax_i})^2$$

S——双精度实型变量,输出参数。平均标准偏差,即

$$S=\sqrt{Q/N}$$

UMAX——双精度实型变量,输出参数。最大偏差,即

$$UMAX=\max_{1\leq i\leq N}|y_i-BT^{Ax_i}|$$

UMIN——双精度实型变量,输出参数。最小偏差,即

$$UMIN=\min_{1\leq i\leq N}|y_i-BT^{Ax_i}|$$

U——双精度实型变量,输出参数。偏差平均值,即

$$U=\sum_{i=1}^{N}|y_i-BT^{Ax_i}|/N$$

五、子程序(文件名:ILOG1.FOR)

```
      SUBROUTINE ILOG1(X,Y,N,T,A,B,Q,S,UMAX,UMIN,U)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y,T,A,B,Q,S,UMAX,UMIN,U
      DOUBLE PRECISION XX,YY,DX,DXY
      XX=0.0
      YY=0.0
      DO 10 I=1,N
        XX=XX+X(I)/N
        YY=YY+(LOG(Y(I))/LOG(T))/N
10    CONTINUE
      DX=0.0
      DXY=0.0
      DO 20 I=1,N
        Q=X(I)-XX
        DX=DX+Q*Q
        DXY=DXY+Q*(LOG(Y(I))/LOG(T)-YY)
20    CONTINUE
      A=DXY/DX
      B=YY-A*XX
```

```
      B=T**B
      Q=0.0
      U=0.0
      UMAX=0.0
      UMIN=1.0D+30
      DO 30 I=1,N
        S=B*T**(A*X(I))
        Q=Q+(Y(I)-S)*(Y(I)-S)
        DX=ABS(Y(I)-S)
        IF (DX.GT.UMAX) UMAX=DX
        IF (DX.LT.UMIN) UMIN=DX
        U=U+DX/N
30    CONTINUE
      S=SQRT(Q/N)
      RETURN
      END
```

六、例

给定 12 个数据点如下：

x	0.96	0.94	0.92	0.90	0.88	0.86	0.84	0.82	0.80	0.78	0.76	0.74
y	558.0	313.0	174.0	97.0	55.8	31.3	17.4	9.70	5.58	3.13	1.74	1.00

用函数 $y=bt^{ax}$ 进行拟合，并求偏差平方和 Q、平均标准偏差 S、最大偏差 UMAX、最小偏差 UMIN，偏差平均值 U。其中 $t=10.0$。

主程序(文件名：ILOG10.FOR)为

```
      DIMENSION X(12),Y(12)
      DOUBLE PRECISION X,Y,T,A,B,Q,S,UMAX,UMIN,U
      DATA X/0.96,0.94,0.92,0.90,0.88,0.86,0.84,
     *       0.82,0.80,0.78,0.76,0.74/
      DATA Y/558.0,313.0,174.0,97.0,55.8,31.3,17.4,
     *       9.70,5.58,3.13,1.74,1.0/
      T=10.0
      N=12
      CALL ILOG1(X,Y,N,T,A,B,Q,S,UMAX,UMIN,U)
      WRITE(*,*)
      WRITE(*,10) A,B
10    FORMAT(1X,'A=',D13.6,3X,'B=',D13.6)
      WRITE(*,*)
      WRITE(*,20) Q,S
      WRITE(*,*)
      WRITE(*,30) UMAX,UMIN,U
```

```
      WRITE(*,*)
20    FORMAT(1X,'Q=',D13.6,3X,'S=',D13.6)
30    FORMAT(1X,'UMAX=',D13.6,3X,'UMIN=',D13.6,3X,'U=',D13.6)
      END
```

运行结果为

A=	.124927D+02	B=	.561931D-09		
Q=	.327159D+02	S=	.165116D+01		
UMAX=	.506521D+01	UMIN=	.111965D-01	U=	.853078D+00

9.6 对数数据相关

一、功能

对于给定的 n 个数据点 $(x_i,y_i)(i=1,2,\cdots,n)$ 用函数

$$y=bx^a$$

作拟合。

二、方法说明

设给定 n 个数据点 $(x_i,y_i)(i=1,2,\cdots,n)$，且 $x_i,y_i>0(i=1,2,\cdots,n)$，用函数

$$y=bx^a,\quad x,y>0$$

进行拟合。为了求拟合参数 a 与 b，两边取对数，即

$$\ln y=\ln b+a\ln x$$

令

$$\tilde{y}=\tilde{a}\,\tilde{x}+\tilde{b}$$

其中 $\tilde{y}=\ln y,\tilde{x}=\ln x,\tilde{a}=a,\tilde{b}=\ln b$

此时，问题化为对 n 个数据点 $(\tilde{x}_i,\tilde{y}_i)$ 作线性拟合。求出 $\tilde{a}$ 与 $\tilde{b}$ 后，就可以得到

$$a=\tilde{a}\ ,\ b=e^{\tilde{b}}$$

关于一元线性拟合参看 9.2 节的方法说明。

三、子程序语句

SUBROUTINE ILOG2(X,Y,N,A,B,Q,S,UMAX,UMIN,U)

四、形参说明

X,Y——均为双精度实型一维数组，长度为 N，输入参数。存放 N 个数据点 $(x_i,y_i)(i=1,2,\cdots,N)$，要求所有的 $x_i,y_i>0\ (i=1,2,\cdots,N)$。

N——整型变量，输入参数。数据点数。

A,B——均为双精度实型变量，输出参数。拟合函数的参数。即拟合函数形式为

$$y=Bx^A$$

Q——双精度实型变量，输出参数。偏差平方和，即

$$Q=\sum_{i=1}^{N}(y_i-Bx_i^A)^2$$

S——双精度实型变量，输出参数。平均标准差，即

$$S=\sqrt{Q/N}$$

UMAX——双精度实型变量,输出参数。最大偏差,即

$$UMAX=\max_{1\leqslant i\leqslant N}|y_i-Bx_i^A|$$

UMIN——双精度实型变量,输出参数。最小偏差,即

$$UMIN=\min_{1\leqslant i\leqslant N}|y_i-Bx_i^A|$$

U——双精度实型变量,输出参数。偏差平均值,即

$$U=\sum_{i=1}^{N}|y_i-Bx_i^A|/N$$

五、子程序(文件名:ILOG2.FOR)

```fortran
      SUBROUTINE ILOG2(X,Y,N,A,B,Q,S,UMAX,UMIN,U)
      DIMENSION X(N),Y(N)
      DOUBLE PRECISION X,Y,A,B,Q,S,UMAX,UMIN,U
      DOUBLE PRECISION XX,YY,DX,DXY
      XX=0.0
      YY=0.0
      DO 10 I=1,N
        XX=XX+LOG(X(I))/N
        YY=YY+LOG(Y(I))/N
10    CONTINUE
      DX=0.0
      DXY=0.0
      DO 20 I=1,N
        Q=LOG(X(I))-XX
        DX=DX+Q*Q
        DXY=DXY+Q*(LOG(Y(I))-YY)
20    CONTINUE
      A=DXY/DX
      B=YY-A*XX
      B=EXP(B)
      Q=0.0
      U=0.0
      UMAX=0.0
      UMIN=1.0D+30
      DO 30 I=1,N
        S=B*X(I)**A
        Q=Q+(Y(I)-S)*(Y(I)-S)
        DX=ABS(Y(I)-S)
        IF (DX.GT.UMAX) UMAX=DX
        IF (DX.LT.UMIN) UMIN=DX
        U=U+DX/N
```

```
30      CONTINUE
        S=SQRT(Q/N)
        RETURN
        END
```

六、例

给定 10 个数据点如下：

x	0.1	1.0	3.0	5.0	8.0	10.0	20.0	50.0	80.0	100.0
y	0.1	0.9	2.5	4.0	6.3	7.8	14.8	36.0	54.0	67.0

用函数 $y=bx^a$ 进行拟合，并求偏差平方和 Q、平均标准偏差 S、最大偏差 UMAX、最小偏差 UMIN、偏差平均值 U。

主程序(文件名：ILOG20.FOR)为

```
      DIMENSION X(10),Y(10)
      DOUBLE PRECISION X,Y,A,B,Q,S,UMAX,UMIN,U
      DATA X/0.1,1.0,3.0,5.0,8.0,10.0,20.0,50.0,80.0,100.0/
      DATA Y/0.1,0.9,2.5,4.0,6.3,7.8,14.8,36.0,54.0,67.0/
      N=10
      CALL ILOG2(X,Y,N,A,B,Q,S,UMAX,UMIN,U)
      WRITE(*,*)
      WRITE(*,10) A,B
10    FORMAT(1X,'A=',D13.6,3X,'B=',D13.6)
      WRITE(*,*)
      WRITE(*,20) Q,S
      WRITE(*,*)
      WRITE(*,30) UMAX,UMIN,U
      WRITE(*,*)
20    FORMAT(1X,'Q=',D13.6,3X,'S=',D13.6)
30    FORMAT(1X,'UMAX=',D13.6,3X,'UMIN=',D13.6,3X,'U=',D13.6)
      END
```

运行结果为

```
A=      .941576D+00     B=         .885773D+00
Q=      .178560D+01     S=         .422682D+00
UMAX=    .856169D+00     UMIN=       .133199D-02     U=    .250615D+00
```

第10章 极值问题

10.1 一维极值有理法

一、功能

用连分式法求目标函数 $f(x)$ 的极值点。

二、方法说明

设目标函数 $f(x)$ 在区间 $[a,b]$ 上连续且单峰（或单谷）。$f(x)$ 的极值点为函数

$$y(x)=\frac{d[f(x)]}{dx}$$

的零点。

设 $y(x)$ 的反函数为 $x=\varphi(y)$，且用如下连分式表示：

$$x=\varphi(y)=b_1+\cfrac{y-y_1}{b_2+\cfrac{y-y_2}{b_3+\cdots+\cfrac{y-y_{i-1}}{b_i+\cdots}}}$$

令 $y=0$，即得到 $y(x)$ 的零点（即 $f(x)$ 的极值点）为

$$\bar{x}=b_1-\cfrac{y_1}{b_2-\cfrac{y_2}{b_3-\cdots-\cfrac{y_{i-1}}{b_i-\cdots}}}$$

由此可得计算 $f(x)$ 极值点（即 $y(x)$ 的零点）的过程如下。

在区间 $[a,b]$ 上选取三个初始点 x_1,x_2,x_3，并分别计算得到相应的导数值

$$y_1=f'(x_1),\ y_2=f'(x_2),\ y_3=f'(x_3)$$

利用连分式插值法计算得到系数 b_1,b_2 及 b_3。如此又可以求得一个新的点

$$x_4=b_1-\cfrac{y_1}{b_2-\cfrac{y_2}{b_3}}$$

一般来说，假设已经取得点列

$$(x_1,y_1),\ (x_2,y_2),\cdots,(x_{i-1},y_{i-1})$$

并由此确定了系数 $b_1,b_2,\cdots,b_{i-1}$。则又可以求出新的点

$$x_i=b_1-\cfrac{y_1}{b_2-\cfrac{y_1}{b_3-\cdots-\cfrac{y_{i-2}}{b_{i-1}}}}$$

及

$$y_i=f'(x_i)$$

再由如下递推关系计算出新的 b_i：

$$u=x_i$$

$$u=(y_i-y_j)/(u-b_j)\ ,\ j=1,2,i-1$$

$$b_i=u$$

以上过程一直进行到对于某个 x_i 满足

$$|y_i|<\epsilon$$

为止。此时，x_i 即为所求极值点 $\bar{x}$。

求出极值点后，取一个很小的 Δx，若不等式

$$f(\bar{x}+\Delta x)-2f(\bar{x})+f(\bar{x}-\Delta x)\leqslant 0$$

成立，则 $\bar{x}$ 为极大值点；否则为极小值点。

三、子程序语句

SUBROUTINE JMAX1(X,Z,K,M,EPS,FS)

四、形参说明

X——双精度实型变量，输入兼输出参数。调用时存放初值；返回时存放极值点 $\bar{x}$。

Z——双精度实型变量，输出参数。存放极值点 $\bar{x}$ 处的函数值 $f(\bar{x})$。

K——整型变量，输出参数。若 K=0，表示返回的 X 为极小值点；若 K=1，表示返回的 X 为极大值点。

M——整型变量，输出参数，若 M=0，表示本子程序迭代了 10 次还未满足精度要求，此时返回的 X 与 Z 值仅供参考；若 M≠0，表示正常返回。

EPS——实型变量，输入参数。控制精度要求，一般取 10^{-10}～10^{-35}之间的数。

FS——子程序名，输入参数。该子程序用于计算目标函数值 $f(x)$ 及导数值 $f'(x)$，在主程序中必须用外部语句对相应的实参进行说明，该子程序由用户自编，其语句形式为

SUBROUTINE FS(X,Y,DY)

其中：X 为双精度实型自变量值，Y 为双精度实型函数值 $f(x)$，DY 为双精度实型导数值 $f'(x)$。

五、子程序(文件名：JMAX1.FOR)

```
      SUBROUTINE JMAX1(X,Z,K,M,EPS,FS)
      DIMENSION Y(10),B(10)
      DOUBLE PRECISION Y,B,X,Z,XX,DY,H1,H2,DX
      M=10
10    J=0
20    IF (J.LE.2) THEN
        XX=X+0.01*J
      ELSE
        XX=H2
      END IF
      CALL FS(XX,Z,DY)
      IF (ABS(DY).GT.EPS) THEN
        H1=DY
        H2=XX
        IF (J.EQ.0) THEN
```

```
         Y(1)=H1
         B(1)=H2
       ELSE
         Y(J+1)=H1
         DO 30 I=1,J
           H2=H2-B(I)
           IF (ABS(H2)+1.0D0.EQ.1.0D0) THEN
             H2=SIGN(1.0D+35,H2)
             H2=H2*SIGN(1.0D0,H1-Y(I))
           ELSE
             H2=(H1-Y(I))/H2
           END IF
30       CONTINUE
         B(J+1)=H2
         H2=0.0
         DO 40 I=J,1,-1
           H2=B(I+1)+H2
           IF (ABS(H2)+1.0D0.EQ.1.0D0) THEN
             H2=SIGN(1.0D+35,H2)
             H2=H2*SIGN(1.0D0,-Y(I))
           ELSE
             H2=-Y(I)/H2
           END IF
40       CONTINUE
         H2=H2+B(1)
       END IF
       J=J+1
       IF (J.LE.7) GOTO 20
       X=H2
       M=M-1
       IF (M.NE.0) GOTO 10
       XX=X
     END IF
     X=XX
     IF (ABS(X).LE.1.0) THEN
       DX=1.0E-05
     ELSE
       DX=ABS(X*1.0E-05)
     END IF
     XX=X-DX
     CALL FS(XX,H1,DY)
     XX=X+DX
```

```
      CALL FS(XX,H2,DY)
      K=0
      IF ((H1+H2-2*Z).LE.0.0) K=1
      RETURN
      END
```

六.例

计算目标函数

$$f(x)=(x-1)(10-x)$$

的极值点及极值点处的函数值。取初值 X=1.0。

主程序及计算目标函数值、导数值的子程序(文件名:JMAX10.FOR)为

```
      EXTERNAL FA
      DOUBLE PRECISION X,Z
      WRITE(*,*)
      WRITE(*,5)
      READ(*,*) EPS
      X=1.0
      CALL JMAX1(X,Z,K,M,EPS,FA)
      WRITE(*,*)
      WRITE(*,40) M
      WRITE(*,*)
      IF (K.EQ.0) THEN
        WRITE(*,10)
      ELSE
        WRITE(*,20)
      END IF
      WRITE(*,30) X,Z
      WRITE(*,*)
5     FORMAT(1X,'EPS=')
10    FORMAT(1X,'MIN:')
20    FORMAT(1X,'MAX:')
30    FORMAT(1X,'X=',D15.6,5X,'F(X)=',D15.6)
40    FORMAT(1X,'M=',I2)
      END

      SUBROUTINE FA(X,Y,DY)
      DOUBLE PRECISION X,Y,DY
      Y=(X-1.0)*(10.0-X)
      DY=-2.0*X+11.0
      RETURN
      END
```

运行结果为

当输入 $\varepsilon=10^{-10}$时

EPS=

1.0E−10

M=10

MAX:

X=   .550000D+01   F(X)=   .202500D+02

当输入 $\varepsilon=10^{-25}$时

EPS=

1.0E−25

M=10

MAX:

X=   .550000D+01   F(X)=   .202500D+02

当输入 $\varepsilon=0.0$ 时

EPS=

0.0

M=10

MAX:

X=   .550000D+01   F(X)=   .202500D+02

10.2 n 维极值有理法

一、功能

用连分式法求多元函数的极值点与极值。

二、方法说明

设 n 元函数

$$Z=F(x_1,x_2,\cdots,x_n)$$

单峰(或单谷),则 F 的极值点为方程组

$$\frac{\partial F}{\partial x_i}=0\ ,\ i=1,2,\cdots,n$$

的一组实数解。

利用连分式法,轮流求某个 x_i 方向上的极值点(其余 $n-1$ 个变量保持当前值不变)。即对于 $i=1,2,\cdots,n$,分别求函数

$$y(x_i)=\frac{\partial F}{\partial x_i}$$

的零点(具体方法参看 10.1 节的方法说明)。

反复进行上述过程,直到满足

$$\sum_{i=1}^{n}\left|\frac{\partial F}{\partial x_i}\right|<\varepsilon$$

为止。

三、子程序语句

SUBROUTINE JMAXN(X,N,FS,EPS,Z,K,L)

四、形参说明

X——双精度实型一维数组，长度为N，输入兼输出参数。调用时存放N元函数极值点的初值；返回时存放极值点的N个自变量值。

N——整型变量，输入参数。自变量个数。

FS——子程序名，输入参数。该子程序用于计算N元函数的偏导数值。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编，其语句形式为

SUBROUTINE FS(X,N,J,F)

其中：X为双精度实型一维数组，长度为N，存放N个自变量值；N为整型变量，自变量的个数；J为整型变量，表示对x_j求偏导数，即

$$F_{x_j}=\frac{\partial F}{\partial x_j}\ ,\ j=1,2,\cdots,N$$

当J=0时，表示求函数值$F=F(x_1,x_2,\cdots,x_n)$；F为双精度实型变量，存放函数值（当J=0时）或偏导数值$\frac{\partial F}{\partial x_j}(j=1,2,\cdots,N)$。

EPS——实型变量，输入参数。控制精度要求。

Z——双精度实型变量，输出参数。存放极值点处的函数值。

K——整型变量，输出参数。若K=1，表示求出的为极大值点；若K=−1，表示求出的为极小值点；若K=0，表示求出的为鞍点。

L——整型变量，输出参数。若L=0，说明迭代不收敛；若L≠0，表示正常返回。

五、子程序（文件名：JMAXN.FOR）

```
      SUBROUTINE JMAXN(X,N,FS,EPS,Z,K,L)
      DIMENSION X(N),Y(10),B(10)
      DOUBLE PRECISION X,Y,B,Z,T,H1,H2,F,DX
      L=10
10    T=0.0
      DO 15 I=1,N
        CALL FS(X,N,I,F)
        T=T+ABS(F)
15    CONTINUE
      IF (T.GE.EPS) THEN
        DO 60 I=1,N
          IL=5
20        J=0
          T=X(I)
30        IF (J.LE.2) THEN
            Z=T+J*0.01
          ELSE
            Z=H2
```

```
          END IF
          X(I)=Z
          CALL FS(X,N,I,F)
          IF (ABS(F)+1.0.NE.1.0) THEN
            H1=F
            H2=Z
            IF (J.EQ.0) THEN
              Y(1)=H1
              B(1)=H2
            ELSE
              Y(J+1)=H1
              DO 40 K=1,J
                H2=H2-B(K)
                IF (ABS(H2)+1.0D0.EQ.1.0D0) THEN
                  H2=SIGN(1.0D+35,H2)
                  H2=H2*SIGN(1.0D0,H1-Y(K))
                ELSE
                  H2=(H1-Y(K))/H2
                END IF
40            CONTINUE
              B(J+1)=H2
              H2=0.0
              DO 50 K=J,1,-1
                H2=H2+B(K+1)
                IF (ABS(H2)+1.0D0.EQ.1.0D0) THEN
                  H2=SIGN(1.0D+35,H2)
                  H2=H2*SIGN(1.0D0,-Y(K))
                ELSE
                  H2=-Y(K)/H2
                END IF
50            CONTINUE
              H2=H2+B(1)
            END IF
            J=J+1
            IF (J.LE.7) GOTO 30
            X(I)=H2
            IL=IL-1
            IF (IL.NE.0) GOTO 20
          END IF
          X(I)=Z
60      CONTINUE
        L=L-1
```

```
      IF (L.NE.0) GOTO 10
      END IF
      K=1
      DX=0.00001
      T=X(1)
      CALL FS(X,N,0,Z)
      X(1)=T+DX
      CALL FS(X,N,0,H1)
      X(1)=T-DX
      CALL FS(X,N,0,H2)
      X(1)=T
      T=H1+H2-2*Z
      IF (T.GT.0.0) K=-1
      J=1
70    J=J+1
      DX=0.00001
      T=X(J)
      X(J)=T+DX
      CALL FS(X,N,0,H2)
      X(J)=T-DX
      CALL FS(X,N,0,H1)
      X(J)=T
      T=H1+H2-2*Z
      IF ((T*K.LT.0.0).AND.(J.LT.N)) GOTO 70
      IF (T*K.GT.0.0) K=0
      RETURN
      END
```

六、例

设二元函数为

$$Z=(x-1)^2+(y+2)^2+2$$

求其极值点与极值。

用 x_1 表示自变量 x，x_2 表示自变量 y。并取初值 $x_1=0$，$x_2=0$，精度要求 $\varepsilon=10^{-4}$。

主程序及计算函数值或偏导数值的子程序(文件名:JMAXN0.FOR)为

```
      EXTERNAL FS
      DIMENSION X(2)
      DOUBLE PRECISION X,Z
      DATA X/2*0.0/
      EPS=0.000001
      CALL JMAXN(X,2,FS,EPS,Z,K,L)
      WRITE(*,*)
      WRITE(*,60) L
```

```
      DO 10 I=1,2
10    WRITE(*,20) I,X(I)
20    FORMAT(1X,'X(',I2,' )=',D13.6)
      WRITE(*,*)
      IF (K.EQ.1) THEN
        WRITE(*,30) Z
      ELSE IF (K.EQ.(-1)) THEN
        WRITE(*,40) Z
      ELSE
        WRITE(*,50) Z
      END IF
30    FORMAT(1X,'MAX:',5X,'Z=',D13.6)
40    FORMAT(1X,'MIN:',5X,'Z=',D13.6)
50    FORMAT(1X,'NOT',5X,'Z=',D13.6)
60    FORMAT(1X,'L=',I3)
      END

      SUBROUTINE FS(X,N,J,F)
      DIMENSION X(N)
      DOUBLE PRECISION X,F
      F=(X(1)-1.0)*(X(1)-1.0)+(X(2)+2.0)*(X(2)+2.0)+2.0
      IF (J.EQ.0) RETURN
      IF (J.EQ.1) THEN
        F=2.0*(X(1)-1.0)
      ELSE
        F=2.0*(X(2)+2.0)
      END IF
      RETURN
      END
```

运行结果为

```
L=   9
X( 1 )=  .100000D+01
X( 2 )= -.200000D+01
MIN:     Z=   .200000D+01
```

10.3 不等式约束线性规划问题

一、功能

求解不等式约束条件下的线性规划问题。

二、方法说明

设给定 m 阶 n 维不等式约束条件

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leqslant b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant b_m \end{cases}$$

且 $x_j \geqslant 0, j=1,2,\cdots,n$

求一组$(x_1,x_2,\cdots,x_n)$值，使目标函数

$$f = \sum_{j=1}^{n} c_j x_j$$

达到极小值。

如果需要求极大值，则只要令

$$\tilde{f} = -f$$

此时就化为求目标函数

$$\tilde{f} = -\sum_{j=1}^{n} c_j x_j$$

的极小值。

引进 m 个非负松弛变量 $x_{n+1},\cdots,x_{n+m}$，则上述问题化为：

寻找 X 值，使满足

$$AX=B \tag{1}$$

$$X \geqslant 0 \tag{2}$$

且使目标函数

$$f=C^T X \tag{3}$$

达到极小值。其中

$$X=(x_1,x_2,\cdots,x_n,x_{n+1},\cdots,x_{n+m})^T$$

$$B=(b_1,b_2,\cdots,b_m)^T$$

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$C=(c_1,c_2,\cdots,c_n,0,\cdots,0)^T$$

称满足(1)和(2)的解为容许解，其中正分量个数不多于 m 个的容许解称为基本解，而使(3)取极小值的解称为最优解，最优解必在基本解中。

寻找最优解的过程如下。

假定已得到一个基本解，其正分量个数为 m，分别为 $x_{i_1},x_{i_2},\cdots,x_{i_m}$，且与之对应的矩阵 A 中的列向量 $P_{i_1},P_{i_2},\cdots,P_{i_m}$ 为线性无关，这组向量称为基底向量。对于矩阵 A 中的每一列向量均可用基底向量的线性组合表示，即

$$A=PD$$

其中

$$P=[P_{i_1},P_{i_2},\cdots,P_{i_m}]$$

$$D=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1,n+m} \\ d_{21} & d_{22} & \cdots & d_{2,n+m} \\ \vdots & \vdots & & \\ d_{m1} & d_{m2} & \cdots & d_{m,n+m} \end{bmatrix}$$

因此，组合系数矩阵 D 可用下式计算：

$$D=P^{-1}A$$

令

$$z_j=c_{i_1}d_{1j}+c_{i_2}d_{2j}+\cdots+c_{i_m}d_{mj},\quad j=1,2,\cdots,n+m$$

如果对于所有的 j ($j=1,2,\cdots,n+m$)满足

$$z_j-c_j\leqslant 0$$

则取 X 为

$$(x_{i_1},x_{i_2},\cdots,x_{i_m})^T$$

而 X 的其余 n 个分量均为 0。X 即为最优解。

否则选择对应于

$$\max_{1\leqslant j\leqslant n+m}\{(z_j-c_j)\}=z_k-c_k$$

的向量 P_k 进入基底向量组；而将对应于

$$\min_{\substack{1\leqslant l\leqslant m\\ d_{lk}>0}}\{x_{i_l}/d_{lk}\}=x_{i_l}/d_{lk}$$

的向量 P_l 从基底向量组中消去。这样，对于新的基底，其目标函数值比原先的下降了。如果所有的 $d_{lk}\leqslant 0(l=1,2,\cdots,m)$，则说明目标函数值无界。

重复以上过程，直至求出最优解，或者确定目标函数值无界为止。

在上述过程的每一步中，新的解可用下式计算：

$$(x_{i_1},x_{i_2},\cdots,x_{i_m})^T=P^{-1}B$$

基底向量组的初值取单位矩阵，即

$$P=I_m$$

也就是取初始解为

$$X=(0,0,\cdots,0,b_1,b_2,\cdots,b_m)^T$$

三、子程序语句

SUBROUTINE JLPLQ(A,B,C,X,M,N,MN,S,P,D,L,JS,IIS,JJS)

四、形参说明

A——双精度实型二维数组，体积为 M×MN，输入参数。其存放格式如下：

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

B——双精度实型一维数组，长度为 M，输入参数。存放约束条件的右端项 $b_1,b_2,\cdots,b_m$。

C——双精度实型一维数组，长度为 MN，输入参数。存放顺序如下：

$$C=(c_1,c_2,\cdots,c_n,0,\cdots,0)$$

X——双精度实型一维数组，长度为 MN，输出参数。存放顺序为

$$X=(x_1,x_2,\cdots,x_n,x_{n+1},\cdots,x_{n+m})$$

其中前 N 个分量表示目标函数 F 的极小值点，即最优解。

M——整型变量，输入参数。不等式约束条件的个数。

N——整型变量，输入参数。目标函数中自变量的个数。

MN——整型变量，输入参数。MN=M+N。

S——双精度实型变量，输出参数。目标函数 F 的极小值。

P——双精度实型二维数组，体积为 M×M。本子程序的工作数组。

D——双精度实型二维数组，体积为 M×MN。本子程序的工作数组。

L——整型变量，输出参数。若 L=0，表示目标函数值无界；若 L≠0，表示正常返回。

JS，IIS，JJS—— 均为整型一维数组，长度为 M。本子程序的工作数组。

五、子程序(文件名：JLPLQ.FOR)

```
      SUBROUTINE JLPLQ(A,B,C,X,M,N,MN,S,P,D,L,JS,IIS,JJS)
      DIMENSION A(M,MN),B(M),C(MN),X(MN),P(M,M)
      DIMENSION JS(M),D(M,MN),IIS(M),JJS(M)
      DOUBLE PRECISION A,B,C,D,P,X,S,Z,DD
      DO 10 I=1,M
10    JS(I)=N+I
15    L=1
      DO 20 I=1,M
      DO 20 J=1,M
20    P(I,J)=A(I,JS(J))
      CALL BRINV(P,M,K,IIS,JJS)
      IF (K.EQ.0) THEN
        L=0
        RETURN
      END IF
      CALL BRMUL(P,A,M,M,MN,D)
      DO 30 I=1,MN
30    X(I)=0.0
      DO 50 I=1,M
        S=0.0
        DO 40 J=1,M
40      S=S+P(I,J)*B(J)
        X(JS(I))=S
50    CONTINUE
      K=0
      DD=1.0D-35
```

```
      DO 70 J=1,MN
        Z=0.0
        DO 60 I=1,M
60      Z=Z+C(JS(I))*D(I,J)
        Z=Z-C(J)
        IF (Z.GT.DD) THEN
          DD=Z
          K=J
        END IF
70    CONTINUE
      IF (K.EQ.0) THEN
        S=0.0
        DO 80 J=1,N
80      S=S+C(J)*X(J)
        RETURN
      END IF
      J=0
      DD=1.0D+20
      DO 90 I=1,M
        IF (D(I,K).GE.1.0D-20) THEN
          Y=X(JS(I))/D(I,K)
          IF (Y.LT.DD) THEN
            DD=Y
            J=I
          END IF
        END IF
90    CONTINUE
      IF (J.EQ.0) THEN
        L=0
        RETURN
      END IF
      JS(J)=K
      GOTO 15
      END
```

六. 例

设约束条件为

$$\begin{cases} x_1+2x_2+7x_3\leqslant 10 \\ x_1+4x_2+13x_3\leqslant 18 \\ \qquad 2x_2+8x_3\leqslant 13 \\ x_1, x_2, x_3\geqslant 0 \end{cases}$$

求目标函数

$$F=4x_1+9x_2+26x_3$$

的极大值。

化为极小值问题后的目标函数为

$$\tilde{F}=-4x_1-9x_2-26x_3$$

在本例中，M=3，N=3，MN=6。

$$A=\begin{bmatrix}1 & 2 & 7 & 1 & 0 & 0\\ 1 & 4 & 13 & 0 & 1 & 0\\ 0 & 2 & 8 & 0 & 0 & 1\end{bmatrix}$$

$$B=(10,18,13)^T$$

$$C=(-4,-9,-26,0,0,0)^T$$

$$X=(x_1,x_2,x_3,x_4,x_5,x_6)^T$$

主程序(文件名:JLPLQ0.FOR)为

```
      DIMENSION A(3,6),B(3),C(6),X(6),P(3,3),D(3,6)
      DIMENSION JS(3),IIS(3),JJS(3)
      DOUBLE PRECISION A,B,C,D,P,X,S
      DATA A/2*1.0,0.0,2.0,4.0,2.0,7.0,13.0,
     *        8.0,1.0,3*0.0,1.0,3*0.0,1.0/
      DATA B/10.0,18.0,13.0/
      DATA C/-4.0,-9.0,-26.0,3*0.0/
      N=3
      M=3
      MN=6
      CALL JLPLQ(A,B,C,X,M,N,MN,S,P,D,L,JS,IIS,JJS)
      WRITE(*,*)
      IF (L.NE.0) THEN
        WRITE(*,10) (I,X(I),I=1,MN)
        WRITE(*,*)
        WRITE(*,20) S
        WRITE(*,*)
      END IF
10    FORMAT(1X,'X(',I2,')=',D15.6)
20    FORMAT(1X,'S=',D15.6)
      END
```

运行结果为

```
X( 1 )=          .200000D+01
X( 2 )=          .400000D+01
X( 3 )=          .000000D+00
X( 4 )=          .000000D+00
X( 5 )=          .000000D+00
X( 6 )=          .500000D+01
```

S=   -.440000D+02

因此，本问题的极大值点为

$x_1=2.00000, x_2=4.00000, x_3=0.00000$

目标函数的极大值为

F=44.0000

七、附注

本子程序需要调用全选主元高斯-约当法求逆的子程序 BRINV(参看 2.3 节)以及实矩阵相乘的子程序 BRMUL(参看 2.1 节)。

10.4 求 n 维极值的单形调优法

一、功能

用单形调优法求解无约束条件下的 n 维极值问题。

二、方法说明

设具有 n 个变量的目标函数为

$$J=f(x_1,x_2,\cdots,x_n)$$

单形调优法求目标函数 J 的极小值点的迭代过程如下。

(1) 在 n 维变量空间中确定一个由 $n+1$ 个顶点构成的初始单形

$$X_{(i)}=(x_{1i},x_{2i},\cdots,x_{ni}),\ i=1,2,\cdots,n+1$$

并计算函数值

$$f_{(i)}=f(X_{(i)}),\ i=1,2,\cdots,n+1$$

(2) 确定

$$f_{(R)}=f(X_{(R)})=\max_{1\leqslant i\leqslant n+1}\{f_{(i)}\}$$

$$f_{(G)}=f(X_{(G)})=\max_{\substack{1\leqslant i\leqslant n+1\\ i\neq R}}\{f_{(i)}\}$$

$$f_{(L)}=f(X_{(L)})=\min_{1\leqslant i\leqslant n+1}\{f_{(i)}\}$$

其中 $X_{(R)}$ 称为最坏点。

(3) 求出最坏点 $X_{(R)}$ 的对称点

$$X_T=2X_F-X_{(R)}$$

其中

$$X_F=\frac{1}{n}\sum_{\substack{i=1\\ i\neq R}}^{n+1}X_{(i)}$$

(4) 确定新的顶点代替原顶点，构成一个新的单形。依次按照如下原则进行替代：

若 $f(X_T)<f_{(L)}$，则需由下式将 X_T 扩大为 X_E

$$X_E=(1+\mu)X_T-\mu X_F$$

其中 μ 称为扩张系数，一般取 $1.2<\mu<2.0$。在这种情况下，如果 $f(X_E)<f_{(L)}$，则

$$X_E\Rightarrow X_{(R)},\ f(X_E)\Rightarrow f_{(R)}$$

否则

$$X_T \Rightarrow X_{(R)},\ f(X_T) \Rightarrow f_{(R)}$$

若 $f(X_T) \leqslant f_{(G)}$，则 $X_T \Rightarrow X_{(R)}$，$f(X_T) \Rightarrow f_{(R)}$。

若 $f(X_T) > f_{(G)}$，如果 $f(X_T) \leqslant f_{(R)}$，则

$$X_T \Rightarrow X_{(R)},\ f(X_T) \Rightarrow f_{(R)}$$

然后由下式将 X_T 缩小为 X_E：

$$X_E = \lambda X_{(R)} + (1-\lambda) X_F$$

其中 λ 称为收缩系数，一般取 $0.0 < \lambda < 1.0$。在这种情况下，如果 $f(X_E) > f_{(R)}$，则新的单形的 $n+1$ 个顶点为

$$X_{(i)} = \frac{1}{2}(X_{(i)} + X_{(L)}),\ i=1,2,\cdots,n+1$$

且

$$f_{(i)} = f(X_{(i)}),\ i=1,2,\cdots,n+1$$

否则 $X_E \Rightarrow X_{(R)}, f(X_E) = f_{(R)}$。

重复(2)～(4)，直到单形中各顶点距离小于预先给定的精度要求为止。

如果实际问题中需要求极大值，则只要令目标函数为

$$\bar{J} = -J = -f(x_1, x_2, \cdots, x_n)$$

即可，此时 $\bar{J}$ 的极小值的绝对值即为 J 的极大值。

三、子程序语句

SUBROUTINE JJSIM (N,M,D,U,V,X,Z,EPS,K,FS,XX,F,XT,XF)

四、形参说明

N——整型变量，输入参数。自变量个数。

M——整型变量，输入参数。单形顶点数，M=N+1

D——实型变量，输入参数。初始单形中任意两顶点间的距离。

U——实型变量，输入参数。扩张系数 μ，一般取 $1.2 < \mu < 2.0$。

V——实型变量，输入参数。收缩系数 λ，一般取 $0.0 < \lambda < 1.0$。

X——双精度实型一维数组，长度为 N，输出参数。存放极小值点各坐标值。

Z——双精度实型变量，输出参数。存放极小值。

EPS——实型变量，输入参数。控制精度要求。

K——整型变量，输出参数。迭代次数。如果 K=201，则说明迭代了 200 次还没有满足精度要求，输出的极值点 X 与极值 Z 只作为参考；如果 K≤200，表示正常返回。

FS——双精度实型函数名，输入参数。该函数子程序用于计算目标函数值。在主程序中必须用外部语句对相应的实参进行说明。该函数子程序由用户自编，其语句形式为

FUNCTION FS(N,X)

其中：N 为整型变量，自变量个数；X 为双精度实型一维数组，长度为 N，存放 N 个自变量值。

XX——双精度实型二维数组，体积为 N×M，输出参数。存放最后单形的 M(即 N+1)个顶点坐标

$$(x_{1i}, x_{2i}, \cdots, x_{ni}),\ i=1,2,\cdots,M$$

F——双精度实型一维数组，长度为 M，输出参数，存放最后单形的 M 个顶点上的目标函数值。

XT，XF——均为双精度实型一维数组，长度为 M。本子程序的工作数组。

五、子程序（文件名：JJSIM.FOR）

```
      SUBROUTINE JJSIM(N,M,D,U,V,X,Z,EPS,K,FS,XX,F,XT,XF)
      DIMENSION X(N),F(M),XX(N,M),XT(N),XF(N)
      DOUBLE PRECISION X,Z,F,XX,XT,XF,FR,FL,FG,FT,SF,FS
      INTEGER R,G
      K=1
      FR=SQRT(1.0D0*M)
      FL=D*(FR-1.0)/(1.414*N)
      FG=D*(FR+N-1.0)/(1.414*N)
      DO 10 I=2,M
      DO 10 J=1,N
10    XX(J,I)=FL
      DO 20 I=2,M
20    XX(I-1,I)=FG
      DO 40 I=1,M
        DO 30 J=1,N
30      X(J)=XX(J,I)
        F(I)=FS(N,X)
40    CONTINUE
50    FR=F(1)
      FL=F(1)
      R=1
      L=1
      DO 60 I=2,M
        IF (F(I).GT.FR) THEN
          R=I
          FR=F(I)
        END IF
        IF (F(I).LT.FL) THEN
          L=I
          FL=F(I)
        END IF
60    CONTINUE
      G=1
      FG=F(1)
      J=1
      IF (R.EQ.1) THEN
        G=2
```

```
        FG=F(2)
        J=2
      END IF
      DO 70 I=J+1,M
        IF ((I.NE.R).AND.(F(I).GT.FG)) THEN
          G=I
          FG=F(I)
        END IF
70    CONTINUE
      DO 90 J=1,N
        XF(J)=0.0
        DO 80 I=1,M
          IF (I.NE.R) XF(J)=XF(J)+XX(J,I)/N
80      CONTINUE
        XT(J)=2.0*XF(J)-XX(J,R)
90    CONTINUE
      FT=FS(N,XT)
      IF (FT.LT.F(L)) THEN
        DO 100 J=1,N
100     XF(J)=(1.0+U)*XT(J)-U*XF(J)
        FF=FS(N,XF)
        IF (FF.LT.F(L)) THEN
          DO 110 J=1,N
110       XX(J,R)=XF(J)
          F(R)=FF
        ELSE
          DO 120 J=1,N
120       XX(J,R)=XT(J)
          F(R)=FT
        END IF
      ELSE IF (FT.LE.F(G)) THEN
        DO 130 J=1,N
130     XX(J,R)=XT(J)
        F(R)=FT
      ELSE
        IF (FT.LE.F(R)) THEN
          DO 140 J=1,N
140       XX(J,R)=XT(J)
          F(R)=FT
        END IF
        DO 150 J=1,N
150     XF(J)=V*XX(J,R)+(1.0-V)*XF(J)
```

```
      FF=FS(N,XF)
      IF (FF.GT.F(R)) THEN
      DO 170 I=1,M
        DO 160 J=1,N
          XX(J,I)=(XX(J,I)+XX(J,L))/2.0
          X(J)=XX(J,I)
160     CONTINUE
        F(I)=FS(N,X)
170   CONTINUE
      ELSE
        DO 180 J=1,N
180     XX(J,R)=XF(J)
        F(R)=FF
      END IF
    END IF
    FF=0.0
    FT=0.0
    DO 190 I=1,M
      FF=FF+F(I)/M
      FT=FT+F(I)*F(I)
190 CONTINUE
    FT=(FT-M*FF*FF)/N
    IF (FT.GE.EPS) THEN
      K=K+1
      IF (K.LT.201) GOTO 50
    END IF
    DO 210 J=1,N
      X(J)=0.0
      DO 200 I=1,M
200   X(J)=X(J)+XX(J,I)/M
210 CONTINUE
    Z=FS(N,X)
    RETURN
    END
```

六、例

求目标函数

$$J=100(x_1-x_1^2)^2+(1-x_1)^2$$

的极小值点与极小值。取 $\varepsilon=10^{-30}$，$D=1.0$，$\mu=1.6$，$\lambda=0.4$。其中 $N=2$，$M=3$，

主程序及计算目标函数的子程序(文件名：JJSIM0.FOR)为

```
      EXTERNAL FS
      DIMENSION X(2),XX(2,3),F(3),XT(2),XF(2)
```

```
      DOUBLE PRECISION X,Z,XX,F,FS,XT,XF
      N=2
      M=3
      D=1.0
      U=1.6
      V=0.4
      EPS=1.0E-30
      CALL JJSIM(N,M,D,U,V,X,Z,EPS,K,FS,XX,F,XT,XF)
      WRITE(*,*)
      WRITE(*,10) K
10    FORMAT(1X,'K=',I4)
      WRITE(*,*)
      WRITE(*,20)
20    FORMAT(7X,'X(1)',11X,'X(2)',11X,'F')
      DO 30 I=1,M
30    WRITE(*,40) XX(1,I),XX(2,I),F(I)
40    FORMAT(1X,3D15.6)
      WRITE(*,*)
      WRITE(*,50) (I,X(I),I=1,N)
50    FORMAT(1X,'X(',I2,')=',D15.6)
      WRITE(*,60) Z
60    FORMAT(1X,'Z=',D15.6)
      WRITE(*,*)
      END

      FUNCTION FS(N,X)
      DIMENSION X(N)
      DOUBLE PRECISION X,S,FS
      S=X(2)-X(1)*X(1)
      S=100.0*S*S
      FS=S+(1.0-X(1))*(1.0-X(1))
      RETURN
      END
```

运行结果为

```
K=   84
       X(1)           X(2)           F
     .100000D+01    .100000D+01    .537804D-15
     .100000D+01    .100000D+01    .576446D-15
     .100000D+01    .100000D+01    .257919D-15
X( 1 )=    .100000D+01
X( 2 )=    .100000D+01
```

Z=　　.168347D−15

本问题的理论极小值点为：$x_1=1.0$，$x_2=1.0$；极小值为0.0。

10.5　求约束条件下 n 维极值的复形调优法

一、功能

用复形调优法求解等式与不等式约束条件下的 n 维极值问题。

二、方法说明

设多变量目标函数为

$$J=f(x_1,x_2,\cdots,x_n)$$

n 个常量约束条件为

$$a_i\leqslant x_i\leqslant b_i,\ i=1,2,\cdots,n$$

k 个函数约束条件为

$$C_j(x_1,x_2,\cdots,x_n)\leqslant W_j(x_1,x_2,\cdots,x_n)\leqslant D_j(x_1,x_2,\cdots,x_n),\ j=1,2,\cdots,k$$

求 n 维目标函数 J 的极小值点与极小值。

若需要求极大值，则令

$$\tilde{J}=-J=-f(x_1,x_2,\cdots,x_n)$$

$\tilde{J}$ 的极小值的绝对值即为 J 的极大值。

复形调优法求目标函数 J 的极小值点的迭代过程如下。

复形共有 $2n$ 个顶点。假设给定初始复形中的第一个顶点坐标为

$$X_{(1)}=(x_{11},x_{21},\cdots,x_{n1})$$

且此顶点坐标满足 n 个常数约束条件与 k 个函数约束条件。

(1) 在 n 维变量空间中确定出初始复形的其余 $2n-1$ 个顶点。方法如下：

利用伪随机数按常量约束条件产生第 j 个顶点

$$X_{(j)}=(x_{1j},x_{2j},\cdots,x_{nj})\ ,\ j=2,3,\cdots,2n$$

即其中的各分量值为

$$x_{ij}=a_i+RN(b_i-a_i),\ i=1,2,\cdots,n;j=2,3,\cdots,2n$$

其中 RN 为[0,1]之间均匀分布的一个伪随机数。

显然，由上述方法产生的初始复形的各顶点满足常量约束条件。然后再检查它们是否符合函数约束条件，如果不符合，则需要作调整，直到全部符合函数约束条件及常量约束条件为止。调整的原则为：

假设前 $j-1$ 个顶点已满足所有的约束条件，而第 j 个顶点不满足约束条件，则作如下调整变换($j=2,3,\cdots,2n$)：

$$X_{(j)}=(X_{(j-1)}+T)/2$$

其中

$$T=\frac{1}{j-1}\sum_{i=1}^{j-1}X_{(i)}$$

这个过程一直作到满足所有约束条件为止。

初始复形的 $2n$ 个顶点确定以后，计算各顶点处的目标函数值

$$f_{(j)}=f(X_{(j)}),\ j=1,2,\cdots,2n$$

(2) 确定

$$f_{(R)}=f(X_{(R)})=\max_{1\leqslant i\leqslant 2n}\{f_{(i)}\}$$

$$f_{(G)}=f(X_{(G)})=\max_{\substack{1\leqslant i\leqslant 2n\\ i\neq R}}\{f_{(i)}\}$$

其中 $X_{(R)}$称为最坏点。

(3) 计算最坏点 $X_{(R)}$的对称点

$$X_T=(1+\alpha)X_F-\alpha X_{(R)}$$

其中

$$X_F=\frac{1}{2n-1}\sum_{\substack{i=1\\ i\neq R}}^{2n}X_{(i)}$$

α 称为反射系数，一般取 1.3 左右。

(4) 确定一个新的顶点替代最坏点 $X_{(R)}$构成新的复形。其方法如下：

如果 $f(X_T)>f_{(G)}$，则用下式修改 X_T：

$$X_T=(X_F+X_T)/2$$

直到 $f(X_T)\leqslant f_{(G)}$为止。

然后检查 X_T 是否满足所有约束条件。如果对于某个分量 $X_T(j)$不满足常量约束条件，即如果 $X_T(j)<a_j$ 或 $X_T(j)>b_j$，则令

$$X_T(j)=a_j+\delta \text{ 或 } X_T(j)=b_j-\delta$$

δ 为很小的一个常数，一般取 $\delta=10^{-6}$。然后重复(4)。

如果 X_T 不满足函数约束条件，则用下式修改 X_T：

$$X_T=(X_F+X_T)/2$$

重复(4)。

直到 $f(X_T)\leqslant f_{(G)}$且 X_T 满足所有约束条件为止。此时令

$$X_{(R)}=X_T,\ f_{(R)}=f(X_T)$$

重复(2)到(4)，直到复形中各顶点距离小于预先给定的精度要求为止。

三、子程序语句

SUBROUTINE JCPLX(N,K,M,A,B,C,D,W,ALPHA,EPS,FJ,FCN,XX,F,X,Z,L,XT,XF)

四、形参说明

N——整型变量，输入参数。自变量个数。

K——整型变量，输入参数。函数约束条件的个数。

M——整型变量，输入参数。复形顶点个数，M=2N。

A——双精度实型一维数组，长度为 N，输入参数。依次存放常量约束条件中自变量 $x_i(i=1,2,\cdots,N)$的下界。

B——双精度实型一维数组，长度为 N，输入参数。依次存放常量约束条件中自变量

$x_i(i=1,2,\cdots,N)$的上界。

C,D,W——均为双精度实型一维数组,长度为K。本子程序的工作数组,分别存放函数约束条件中的下限值$C_j(x_1,x_2,\cdots,x_n)$、上限值$D_j(x_1,x_2,\cdots,x_n)$及计算值$W_j(x_1,x_2,\cdots,x_n)(j=1,2,\cdots,K)$。

ALPHA——实型变量,输入参数。反射系数α,一般取值为1.3左右。

EPS——实型变量,输入参数。控制精度要求。

FJ——双精度实型函数名,输入参数。该函数子程序用于计算目标函数值。在主程序中必须用外部语句对相应的实参进行说明。该函数子程序由用户自编,其语句形式为

FUNCTION FJ(N,X)

其中:N为整型变量,自变量的个数;X为双精度实型一维数组,长度为N,存放N个自变量值。

FCN——子程序名,输入参数。该子程序用于计算函数约束条件中的下限值$C_j(x_1,x_2,\cdots,x_n)$、上限值$D_j(x_1,x_2,\cdots,x_n)$与计算值$W_j(x_1,x_2,\cdots,x_n)(j=1,2,\cdots,K)$。在主程序中必须用外部语句对相应的实参进行说明。该子程序由用户自编,其语句形式为

SUBROUTINE FCN(N,K,X,C,D,W)

其中:X为双精度实型一维数组,长度为N,存放N个自变量值;C、D、W均为双精度实型一维数组,长度为K,分别存放函数约束条件中的下限值$C_j(x_1,x_2,\cdots,x_n)$、上限值$D_j(x_1,x_2,\cdots,x_n)$与计算值$W_j(x_1,x_2,\cdots,x_n)(j=1,2,\cdots,K)$。

XX——双精度实型二维数组,体积为N×M,输出参数。存放最后复形的M个顶点坐标

$$(x_{1i},x_{2i},\cdots,x_{ni}),\ i=1,2,\cdots,M$$

F——双精度实型一维数组,长度为M,输出参数。存放最后复形的M个顶点上的目标函数值。

X——双精度实型一维数组,长度为N,输入兼输出参数。调用时存放初始复形的第一个顶点的各坐标值,要求满足所有的约束条件;返回时存放满足约束条件的极小值点各坐标值。

Z——双精度实型变量,输出参数。存放极小值。

L——整型变量,输出参数。迭代次数。如果L=201,说明迭代了200次还没有满足精度要求,输出的极小值点X与极小值Z只作为参考;如果L≤200,正常返回。

XT,XF——均为双精度实型一维数组,长度为N。本子程序的工作数组。

五、子程序(文件名:JCPLX.FOR)

```
      SUBROUTINE JCPLX(N,K,M,A,B,C,D,W,ALPHA,EPS,
     *                 FJ,FCN,XX,F,X,Z,L,XT,XF)
      DIMENSION A(N),B(N),C(K),D(K),W(K),X(N)
      DIMENSION XX(N,M),F(M),XT(N),XF(N)
      DOUBLE PRECISION A,B,C,D,W,XX,F,X,Z,XT,XF,FJ,FR,FG,T
      REAL NRND1
      INTEGER R,G
```

```
      DO 10 I=1,N
10    XX(I,1)=X(I)
      F(1)=FJ(N,X)
      DO 100 J=2,M
        DO 20 I=1,N
          XX(I,J)=A(I)+(B(I)-A(I))*NRND1(T)
          X(I)=XX(I,J)
20      CONTINUE
30      R=1
        G=0
40      IF ((A(R).LE.X(R)).AND.(B(R).GE.X(R))) THEN
          R=R+1
          IF (R.LE.N) GOTO 40
        ELSE
          G=1
        END IF
        IF (G.EQ.0) THEN
          CALL FCN(N,K,X,C,D,W)
          R=1
60        IF ((C(R).LE.W(R)).AND.(D(R).GE.W(R))) THEN
            R=R+1
            IF (R.LE.K) GOTO 60
          ELSE
            G=1
          END IF
        END IF
        IF (G.NE.0) THEN
          DO 80 R=1,N
            Z=0.0
            DO 70 G=1,J-1
70          Z=Z+XX(R,G)/(J-1.0)
            XX(R,J)=(XX(R,J)+Z)/2.0
            X(R)=XX(R,J)
80        CONTINUE
          GOTO 30
        END IF
        F(J)=FJ(N,X)
100   CONTINUE
      L=1
110   FR=F(1)
      R=1
      DO 120 I=2,M
```

```
        IF (F(I).GT.FR) THEN
          R=I
          FR=F(I)
        END IF
120   CONTINUE
      G=1
      J=1
      FG=F(1)
      IF (R.EQ.1) THEN
        G=2
        J=2
        FG=F(2)
      END IF
      DO 130 I=J+1,M
        IF (I.NE.R) THEN
          IF (F(I).GT.FG) THEN
            G=I
            FG=F(I)
          END IF
        END IF
130   CONTINUE
      DO 150 I=1,N
        XF(I)=0.0
        DO 140 J=1,M
          IF (J.NE.R) XF(I)=XF(I)+XX(I,J)/(M-1.0)
140     CONTINUE
        XT(I)=(1.0+ALPHA)*XF(I)-ALPHA*XX(I,R)
150   CONTINUE
160   Z=FJ(N,XT)
      IF (Z.GT.FG) THEN
        DO 180 I=1,N
180     XT(I)=(XT(I)+XF(I))/2.0
        GOTO 160
      END IF
      J=0
      DO 190 I=1,N
        IF (A(I).GT.XT(I)) THEN
          XT(I)=XT(I)+0.000001
          J=1
        END IF
        IF (B(I).LT.XT(I)) THEN
          XT(I)=XT(I)-0.000001
```

```
      J=1
    END IF
190 CONTINUE
    IF (J.NE.0) GOTO 160
    CALL FCN(N,K,XT,C,D,W)
    J=1
200 IF ((C(J).LE.W(J)).AND.(D(J).GE.W(J))) THEN
      J=J+1
      IF (J.LE.K) GOTO 200
    END IF
    IF (J.LE.K) THEN
      DO 210 I=1,N
210   XT(I)=(XT(I)+XF(I))/2.0
      GOTO 160
    END IF
    DO 220 I=1,N
220 XX(I,R)=XT(I)
    F(R)=Z
    FR=0.0
    FG=0.0
    DO 230 J=1,M
      FR=FR+F(J)/M
      FG=FG+F(J)*F(J)
230 CONTINUE
    FR=(FG-M*FR*FR)/(M-1.0)
    IF (FR.GE.EPS) THEN
      L=L+1
      IF (L.LT.201) GOTO 110
    END IF
    DO 250 I=1,N
      X(I)=0.0
      DO 240 J=1,M
240   X(I)=X(I)+XX(I,J)/M
250 CONTINUE
    Z=FJ(N,X)
    RETURN
    END
```

六、例

求目标函数

$$J=f(x_1,x_2)=-\frac{[9-(x_1-3)^2]x_2^3}{27\sqrt{3}}$$

满足约束条件

$$x_1 \geqslant 0, \ x_2 \geqslant 0$$

$$0 \leqslant x_2 \leqslant \frac{x_1}{\sqrt{3}}, \ 0 \leqslant x_1 + \sqrt{3} x_2 \leqslant 6$$

的极小值点与极小值。

取 $\varepsilon = 10^{-30}$，反射系数 $\alpha = 1.3$。

初始复形的第一个顶点为(0.0,0.0)。

其中常量约束条件的下界为 $a_1 = 0.0$，$a_2 = 0.0$，上界取 $b_1 = 10^{35}$，$b_2 = 10^{35}$。

N=2，K=2，M=4。

主程序及计算目标函数值子程序、计算函数约束条件中各值的子程序(文件名：JCPLX0.FOR)为

```
      EXTERNAL FJ,FCN
      DIMENSION X(2),XX(2,4),F(4),A(2),B(2),C(2),D(2),W(2)
      DIMENSION XT(2),XF(2)
      DOUBLE PRECISION X,Z,XX,F,FJ,A,B,C,D,W,XT,XF
      DATA X/2*0.0/
      DATA A/2*0.0/
      B(1)=1.0D+35
      B(2)=B(1)
      N=2
      K=2
      M=4
      EPS=1.0E-30
      ALPHA=1.3
      CALL JCPLX(N,K,M,A,B,C,D,W,ALPHA,EPS,
     *            FJ,FCN,XX,F,X,Z,L,XT,XF)
      WRITE(*,*)
      WRITE(*,10) L
10    FORMAT(1X,'L=',I4)
      WRITE(*,*)
      WRITE(*,20)
20    FORMAT(7X,'X(1)',11X,'X(2)',11X,'F')
      DO 30 I=1,M
30    WRITE(*,40) XX(1,I),XX(2,I),F(I)
40    FORMAT(1X,3D15.6)
      WRITE(*,*)
      WRITE(*,50) (I,X(I),I=1,N)
50    FORMAT(1X,'X(',I2,')=',D15.6)
      WRITE(*,60) Z
60    FORMAT(1X,'Z=',D15.6)
      WRITE(*,*)
```

```
      END
      FUNCTION FJ(N,X)
      DIMENSION X(N)
      DOUBLE PRECISION X,FJ,S
      S=-(9.0-(X(1)-3.0)*(X(1)-3.0))
      S=S*X(2)*X(2)*X(2)/(27.0*SQRT(3.0))
      FJ=S
      RETURN
      END
      SUBROUTINE FCN(N,K,X,C,D,W)
      DIMENSION X(N),C(K),D(K),W(K)
      DOUBLE PRECISION X,C,D,W
      C(1)=0.0
      C(2)=0.0
      D(1)=X(1)/SQRT(3.0)
      D(2)=6.0
      W(1)=X(2)
      W(2)=X(1)+X(2)*SQRT(3.0)
      RETURN
      END
```

运行结果为

```
L=     71
      X(1)              X(2)              F
    .300000D+01      .173205D+01      -.100000D+01
    .300000D+01      .173205D+01      -.100000D+01
    .300000D+01      .173205D+01      -.100000D+01
    .300000D+01      .173205D+01      -.100000D+01
X( 1 )=     .300000D+01
X( 2 )=     .173205D+01
Z=     -.100000D+01
```

本问题的理论极小值点为 $x_1=3.0$，$x_2=\sqrt{3}$，极小值为-1.0。

七、附注

本子程序需要调用产生 0 到 1 之间均匀分布的随机数子程序 NRND1(参看 13.1 节)。

第11章 数学变换与滤波

11.1 傅里叶级数逼近

一、功能

根据函数 $f(x)$ 在区间 $[0,2\pi]$ 上的 $(2n+1)$ 个等距点

$$x_i=\frac{2\pi}{2n+1}(i+0.5)\quad, i=0,1,2,\cdots,2n$$

处的函数值 f_i，求傅里叶(Fourier)级数

$$f(x)=\frac{1}{2}A_0+\sum_{k=1}^{n}(A_k\cos kx+B_k\sin kx)$$

的前 $(2n+1)$ 个系数

$$A_k(k=0,1,\cdots,n)\text{与} B_k(k=1,2,\cdots,n)$$

的近似值。

二、方法说明

已知函数 $f(x)$ 在区间 $[0,2\pi]$ 上 $(2n+1)$ 个点

$$x_i=\frac{2\pi}{2n+1}(i+0.5),\ i=0,1,2,\cdots,2n$$

处的函数值 $f_i(i=0,1,\cdots,2n)$。计算傅里叶级数的前 $(2n+1)$ 个系数

$$A_k(k=0,1,\cdots,n)\text{与} B_k(k=1,2,\cdots,n)$$

近似值的方法如下。

对于 $k=0,1,2,\cdots,n$，作如下运算：

(1) 按下列迭代公式计算 u_1 与 u_2：

$$\begin{cases}u_{2n+2}=u_{2n+1}=0\\ u_j=f_j+2\cos k\theta\cdot u_{j+1}-u_{j+2},\ j=2n,2n-1,\cdots,2,1\end{cases}$$

其中 $\theta=\dfrac{2\pi}{2n+1}$。计算 $\cos k\theta$ 与 $\sin k\theta$ 用如下递推公式：

$$\cos k\theta=\cos\theta\cos(k-1)\theta-\sin\theta\sin(k-1)\theta$$

$$\sin k\theta=\sin\theta\cos(k-1)\theta+\cos\theta\sin(k-1)\theta$$

(2) 按下列公式计算 A_k 与 B_k：

$$\begin{cases}A_k=\dfrac{2}{2n+1}(f_0+u_1\cos k\theta-u_2)\\[2ex] B_k=\dfrac{2}{2n+1}u_1\sin k\theta\end{cases}$$

三、子程序语句

SUBROUTINE KFOUR(F,N1,N2,A,B)

四、形参说明

F——双精度实型一维数组，长度为 N2=2n+1，输入参数。存放区间[0,2π]上(2n+1)个等距点。

$$x_i=\frac{2\pi}{2n+1}(i-0.5),\ i=1,2,\cdots,N2$$

处的函数值 $f(x_i)$ $(i=1,2,\cdots,N2)$。

N1——整型变量，输入参数。其值为 N1=n+1。

N2——整型变量，输入参数。其值为 N2=2n+1。

A——双精度实型一维数组，长度为 N1=n+1，输出参数。存放计算结果 $A_k(k=1,2,\cdots,N1)$。

B——双精度实型一维数组，长度为 N1=n+1，输出参数。存放计算结果 $B_k(k=1,2,\cdots,N1)$，其中 $B_1=0.0$。

五、子程序(文件名:KFOUR.FOR)

```
      SUBROUTINE KFOUR(F,N1,N2,A,B)
      DIMENSION F(N2),A(N1),B(N1)
      DOUBLE PRECISION F,A,B,C,S,T,C1,S1,U1,U2,U0
      T=6.283185306/N2
      C=COS(T)
      S=SIN(T)
      T=2.0/N2
      C1=1.0
      S1=0.0
      DO 20 I=1,N1
        U1=0.0
        U2=0.0
        DO 10 J=N2,2,-1
          U0=F(J)+2.0*C1*U1-U2
          U2=U1
          U1=U0
10      CONTINUE
        A(I)=T*(F(1)+U1*C1-U2)
        B(I)=T*U1*S1
        U0=C*C1-S*S1
        S1=C*S1+S*C1
        C1=U0
20    CONTINUE
      RETURN
      END
```

六、例

给定函数

$$f(x)=x^2$$

在区间$[0,2\pi]$上 101 个等距点

$$x_i=\frac{2\pi}{101}(i-0.5),\ i=1,2,\cdots,101$$

处的函数值

$$f_i=x_i^2,\ i=1,2,\cdots,101$$

求傅里叶级数的系数

$$A_k(k=1,2,\cdots,51)与 B_k(k=1,2,\cdots,51)$$

其中 $n=50$, N1=51, N2=101。

主程序(文件名:KFOUR0.FOR)为

```
      DIMENSION F(101), A(51), B(51)
      DOUBLE PRECISION F,A,B,C,H
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(3X,'N',12X,'A(N)',15X,'B(N)')
      H=6.2831853606/101.0
      DO 20 I=1,101
        C=H*(I-0.5)
        F(I)=C*C
20    CONTINUE
      N1=51
      N2=101
      CALL KFOUR(F,N1,N2,A,B)
      DO 30 I=1,N1
        K=I-1
        WRITE(*,40) K,A(I),B(I)
30    CONTINUE
      WRITE(*,*)
40    FORMAT(1X,I3,5X,D15.6,4X,D15.6)
      END
```

运行结果为

N	A(N)	B(N)
0	.263183D+02	.000000D+00
1	.360655D+01	−.126867D+02
2	.606548D+00	−.633721D+01
3	.509934D−01	−.421798D+01
4	−.143449D+00	−.315631D+01
5	−.233447D+00	−.251766D+01
6	−.282333D+00	−.209052D+01
7	−.311808D+00	−.178423D+01

8	-.330937D+00	-.155347D+01
9	-.344050D+00	-.137306D+01
10	-.353428D+00	-.122788D+01
11	-.360365D+00	-.110833D+01
12	-.365639D+00	-.100799D+01
13	-.369741D+00	-.922419D+00
14	-.372995D+00	-.848449D+00
15	-.375617D+00	-.783751D+00
16	-.377762D+00	-.726583D+00
17	-.379537D+00	-.675607D+00
18	-.381023D+00	-.629787D+00
19	-.382278D+00	-.588301D+00
20	-.383347D+00	-.550491D+00
21	-.384265D+00	-.515827D+00
22	-.385059D+00	-.483871D+00
23	-.385749D+00	-.454263D+00
24	-.386352D+00	-.426702D+00
25	-.386881D+00	-.400935D+00
26	-.387349D+00	-.376746D+00
27	-.387762D+00	-.353953D+00
28	-.388130D+00	-.332397D+00
29	-.388457D+00	-.311941D+00
30	-.388750D+00	-.292468D+00
31	-.389011D+00	-.273872D+00
32	-.389246D+00	-.256062D+00
33	-.389456D+00	-.238957D+00
34	-.389645D+00	-.222484D+00
35	-.389814D+00	-.206579D+00
36	-.389966D+00	-.191183D+00
37	-.390103D+00	-.176245D+00
38	-.390225D+00	-.161716D+00
39	-.390334D+00	-.147553D+00
40	-.390430D+00	-.133716D+00
41	-.390516D+00	-.120167D+00
42	-.390591D+00	-.106872D+00
43	-.390656D+00	-.937995D-01
44	-.390712D+00	-.809189D-01
45	-.390759D+00	-.682015D-01
46	-.390798D+00	-.556201D-01
47	-.390829D+00	-.431486D-01
48	-.390852D+00	-.307616D-01
49	-.390867D+00	-.184345D-01
50	-.390874D+00	-.614315D-02

七、附注

在区间[0,2π]上取 $2n+1$ 个等距点时，应该取各小区间的中点，即用公式

$$x_i=\frac{2\pi}{2n+1}(i-0.5),\ i=1,2,\cdots,2n+1$$

来计算，而不要用公式

$$x_i=\frac{2\pi}{2n+1}(i+1),\ i=1,2,\cdots,2n+1$$

计算。

11.2 快速傅里叶变换

一、功能

用 FFT 计算离散傅里叶(Fourier)变换。

二、方法说明

计算 n 个采样点

$$P=\{p_0,p_1,\cdots,p_{n-1}\}$$

的离散傅里叶变换，可以归结为计算多项式

$$F(x)=p_0+p_1x+p_2x^2+\cdots+p_{n-1}x^{n-1}$$

在各 n 次单位根 $1,\omega,\omega^2,\cdots,\omega^{n-1}$ 上的值，即

$$F_0=p_0+p_1+p_2+\cdots+p_{n-1}$$
$$F_1=p_0+p_1\omega+p_2\omega^2+\cdots+p_{n-1}\omega^{n-1}$$
$$F_2=p_0+p_1\omega^2+p_2(\omega^2)^2+\cdots+p_{n-1}(\omega^2)^{n-1}$$
$$\cdots\cdots$$
$$F_{n-1}=p_0+p_1\omega^{n-1}+p_2(\omega^{n-1})^2+\cdots+p_{n-1}(\omega^{n-1})^{n-1}$$

其中 $\omega=e^{-j2\pi/n}$($j=\sqrt{-1}$)为 n 次单位元根。

若 n 是 2 的 k 次幂，即 $n=2^k(k>0)$，则 $F(x)$ 可以分解为关于 x 的偶次幂与奇次幂两部分，即

$$F(x)=p_0+p_2x^2+\cdots+p_{n-2}x^{n-2}+x(p_1+p_3x^2+\cdots+p_{n-1}x^{n-2})$$

若令

$$P_{even}(x^2)=p_0+p_2x^2+\cdots+p_{n-2}x^{n-2}$$
$$P_{odd}(x^2)=p_1+p_3x^2+\cdots+p_{n-1}x^{n-2}$$

则有

$$F(x)=P_{even}(x^2)+xP_{odd}(x^2)$$

并且有

$$F(-x)=P_{even}(x^2)-xP_{odd}(x^2)$$

由此可以看出，为求 F 在各 n 次单位根上的值，只要求 P_{even} 和 P_{odd} 在 $1,\omega^2,\cdots,(\omega^{(n/2)-1})^2$ 上的值就行了。

而 P_{even} 和 P_{odd} 同样可以分解成关于 x 的偶次幂和奇次幂两部分，以此类推，一直分解

下去，最后只要求二次单位根 1 与 −1 上的值。

在实际计算时，可以将上述过程倒过来进行，这就是 FFT 算法。

三、子程序语句

SUBROUTINE KKFFT(PR,PI,N,K,FR,FI,L,IL)

四、形参说明

PR,PI——均为双精度实型一维数组，长度为 N，输入兼输出参数。调用时分别存放 N 个采样输入的实部与虚部(当 L=0 时，即计算傅里叶变换)或傅里叶变换的 N 个实部与虚部(当 L=1 时，即计算逆傅里叶变换)，返回时分别存放傅里叶变换的模与幅角(当 L=0 时)或逆傅里叶变换的模与幅角(当 L=1 时)，其中幅角的单位为度。

N——整型变量，输入参数。输入的点数，要求 $N=2^K(K>0)$。

K——整型变量，输入参数。要求满足 $N=2^K(K>0)$。

FR,FI——均为双精度实型一维数组，长度为 N，输出参数。分别存放傅里叶变换的实部与虚部(当 L=0 时)或逆傅里叶变换的实部与虚部(当 L=1 时)。

L——整型变量，输入参数。若 L=0，表示计算傅里叶变换；若 L=1，表示计算逆傅里叶变换。

IL——整型变量，输入参数，若 IL=0，表示不计算傅里叶变换或逆变换的模与幅角；若 IL=1，表示要计算傅里叶变换或逆变换的模与幅角。

五、子程序(文件名:KKFFT.FOR)

```
      SUBROUTINE KKFFT(PR,PI,N,K,FR,FI,L,IL)
      DIMENSION PR(N),PI(N),FR(N),FI(N)
      DOUBLE PRECISION PR,PI,FR,FI,P,Q,S,VR,VI,PODDR,PODDI
      DO 20 IT=0,N-1
        M=IT
        IS=0
        DO 10 I=0,K-1
          J=M/2
          IS=2*IS+(M-2*J)
          M=J
10      CONTINUE
        FR(IT+1)=PR(IS+1)
        FI(IT+1)=PI(IS+1)
20    CONTINUE
      PR(1)=1.0
      PI(1)=0.0
      PR(2)=COS(6.283185306/N)
      PI(2)=-SIN(6.283185306/N)
      IF (L.NE.0) PI(2)=-PI(2)
      DO 30 I=3,N
        P=PR(I-1)*PR(2)
        Q=PI(I-1)*PI(2)
```

```
        S=(PR(I-1)+PI(I-1))*(PR(2)+PI(2))
        FR(I)=P-Q
        FI(I)=S-P-Q
 30   CONTINUE
      DO 40 IT=0,N-2,2
        VR=FR(IT+1)
        VI=FI(IT+1)
        FR(IT+1)=VR+FR(IT+2)
        FI(IT+1)=VI+FI(IT+2)
        FR(IT+2)=VR-FR(IT+2)
        FI(IT+2)=VI-FI(IT+2)
 40   CONTINUE
      M=N/2
      NV=2
      DO 70 L0=K-2,0,-1
        M=M/2
        NV=2*NV
        DO 60 IT=0,(M-1)*NV,NV
        DO 60 J=0,(NV/2)-1
          P=PR(M*J+1)*FR(IT+J+1+NV/2)
          Q=PI(M*J+1)*FI(IT+J+1+NV/2)
          S=PR(M*J+1)+PI(M*J+1)
          S=S*(FR(IT+J+1+NV/2)+FI(IT+J+1+NV/2))
          PODDR=P-Q
          PODDI=S-P-Q
          FR(IT+J+1+NV/2)=FR(IT+J+1)-PODDR
          FI(IT+J+1+NV/2)=FI(IT+J+1)-PODDI
          FR(IT+J+1)=FR(IT+J+1)+PODDR
          FI(IT+J+1)=FI(IT+J+1)+PODDI
 60     CONTINUE
 70   CONTINUE
      IF (L.NE.0) THEN
        DO 80 I=1,N
          FR(I)=FR(I)/N
          FI(I)=FI(I)/N
 80     CONTINUE
      END IF
      IF (IL.NE.0) THEN
        DO 90 I=1,N
          PR(I)=SQRT(FR(I)*FR(I)+FI(I)*FI(I))
          PI(I)=ATAN(FI(I)/FR(I))*360.0/6.283185306
 90     CONTINUE
```

```
      END IF
      RETURN
      END
```

六、例

设函数

$$P(t)=e^{-t}, t\geqslant 0$$

取 N=64,K=6,周期 T=6.4,则步长 $h=\frac{T}{N}=0.1$,且取等分区间的中点,即

$$P_i=e^{-(i-0.5)h},\ i=1,2,\cdots,N$$

计算:

(1) P_i 的离散傅里叶变换 F_i,且计算 F_i 的模与幅角。即 L=0,IL=1。

(2) F_i 的逆傅里叶变换 p_i 的模。即 L=1,IL=1。

主程序(文件名:KKFFT0.FOR)为

```
      DIMENSION PR(64),PI(64),FR(64),FI(64)
      DOUBLE PRECISION PR,PI,FR,FI
      N=64
      K=6
      DO 10 I=1,N
        PR(I)=EXP(-0.1*(I-0.5))
        PI(I)=0.0
10    CONTINUE
      WRITE(*,*)
      WRITE(*,20) (PR(I),I=1,N)
20    FORMAT(1X,4D15.6)
      WRITE(*,30)
30    FORMAT(2X,' ')
      CALL KKFFT(PR,PI,N,K,FR,FI,0,1)
      WRITE(*,20) (FR(I),I=1,N)
      WRITE(*,30)
      WRITE(*,20) (FI(I),I=1,N)
      WRITE(*,30)
      WRITE(*,20) (PR(I),I=1,N)
      WRITE(*,30)
      WRITE(*,20) (PI(I),I=1,N)
      WRITE(*,30)
      CALL KKFFT(FR,FI,N,K,PR,PI,1,1)
      WRITE(*,20) (FR(I),I=1,N)
      WRITE(*,*)
      END
```

运行结果为

序列 $P_i(i=1,2,\cdots,64)$；

.951229D+00	.860708D+00	.778801D+00	.704688D+00
.637628D+00	.576950D+00	.522046D+00	.472367D+00
.427415D+00	.386741D+00	.349938D+00	.316637D+00
.286505D+00	.259240D+00	.234570D+00	.212248D+00
.192050D+00	.173774D+00	.157237D+00	.142274D+00
.128735D+00	.116484D+00	.105399D+00	.953691D−01
.862936D−01	.780817D−01	.706512D−01	.639279D−01
.578443D−01	.523397D−01	.473589D−01	.428521D−01
.387742D−01	.350843D−01	.317456D−01	.287246D−01
.259911D−01	.235177D−01	.212797D−01	.192547D−01
.174224D−01	.157644D−01	.142642D−01	.129068D−01
.116786D−01	.105672D−01	.956160D−02	.865170D−02
.782838D−02	.708341D−02	.640933D−02	.579940D−02
.524752D−02	.474815D−02	.429630D−02	.388746D−02
.351752D−02	.318278D−02	.287990D−02	.260584D−02
.235786D−02	.213348D−02	.193045D−02	.174675D−02

P_i 的傅里叶变换 $F_i(i=1,2,\cdots,64)$的实部；

.997923D+01	.531844D+01	.243865D+01	.146438D+01
.106110D+01	.861244D+00	.748901D+00	.679837D+00
.634482D+00	.603158D+00	.580650D+00	.563957D+00
.551251D+00	.541370D+00	.533549D+00	.527264D+00
.522149D+00	.517944D+00	.514456D+00	.511543D+00
.509098D+00	.507039D+00	.505301D+00	.503836D+00
.502604D+00	.501574D+00	.500723D+00	.500030D+00
.499482D+00	.499067D+00	.498775D+00	.498603D+00
.498546D+00	.498603D+00	.498775D+00	.499067D+00
.499482D+00	.500030D+00	.500723D+00	.501574D+00
.502604D+00	.503836D+00	.505301D+00	.507039D+00
.509098D+00	.511543D+00	.514456D+00	.517944D+00
.522149D+00	.527264D+00	.533549D+00	.541370D+00
.551251D+00	.563957D+00	.580650D+00	.603157D+00
.634482D+00	.679837D+00	.748900D+00	.861244D+00
.106110D+01	.146438D+01	.243865D+01	.531844D+01

P_i 的傅里叶变换 $F_i(i=1,2,\cdots,64)$的虚部；

.000000D+00	−.473967D+01	−.382485D+01	−.286773D+01
−.223986D+01	−.181854D+01	−.152015D+01	−.129842D+01
−.112707D+01	−.990376D+00	−.878443D+00	−.784767D+00
−.704911D+00	−.635744D+00	−.575000D+00	−.520997D+00
−.472460D+00	−.428403D+00	−.388053D+00	−.350793D+00
−.316123D+00	−.283634D+00	−.252985D+00	−.223890D+00
−.196104D+00	−.169417D+00	−.143644D+00	−.118622D+00
−.942032D−01	−.702537D−01	−.466479D−01	−.232677D−01

.000000D+00	.232682D−01	.466483D−01	.702540D−01
.942034D−01	.118622D+00	.143644D+00	.169417D+00
.196104D+00	.223890D+00	.252985D+00	.283634D+00
.316123D+00	.350793D+00	.388054D+00	.428404D+00
.472460D+00	.520997D+00	.575000D+00	.635744D+00
.704911D+00	.784767D+00	.878443D+00	.990377D+00
.112707D+01	.129842D+01	.152015D+01	.181854D+01
.223985D+01	.286773D+01	.382485D+01	.473967D+01

P_i 的傅里叶变换 $F_i(i=1,2,\cdots,64)$的模:

.997923D+01	.712393D+01	.453613D+01	.321998D+01
.247849D+01	.201217D+01	.169461D+01	.146563D+01
.129339D+01	.115959D+01	.105300D+01	.966389D+00
.894861D+00	.835016D+00	.784410D+00	.741246D+00
.704172D+00	.672157D+00	.644399D+00	.620268D+00
.599262D+00	.580979D+00	.565094D+00	.551341D+00
.539507D+00	.529413D+00	.520919D+00	.513908D+00
.508288D+00	.503987D+00	.500952D+00	.499146D+00
.498546D+00	.499146D+00	.500952D+00	.503987D+00
.508288D+00	.513908D+00	.520919D+00	.529414D+00
.539507D+00	.551341D+00	.565094D+00	.580979D+00
.599262D+00	.620268D+00	.644399D+00	.672157D+00
.704172D+00	.741246D+00	.784410D+00	.835016D+00
.894861D+00	.966389D+00	.105300D+01	.115959D+01
.129339D+01	.146563D+01	.169461D+01	.201217D+01
.247849D+01	.321998D+01	.453613D+01	.712392D+01

P_i 的傅里叶变换 $F_i(i=1,2,\cdots,64)$的幅角(单位为度):

.000000D+00	−.417067D+02	−.574792D+02	−.629494D+02
−.646513D+02	−.646581D+02	.637729D+02	−.623640D+02
−.606228D+02	−.586578D+02	−.565353D+02	−.542079D+02
−.519741D+02	−.495838D+02	−.471414D+02	−.446575D+02
−.421400D+02	−.395949D+02	.370272D+02	−.344406D+02
−.318381D+02	−.292223D+02	−.265954D+02	−.239589D+02
−.213145D+02	−.186634D+02	−.160067D+02	−.133455D+02
−.106806D+02	−.801289D+01	−.534304D+01	.267181D+01
.000000D+00	.267187D+01	.534309D+01	.801292D+01
.106806D+02	.133455D+02	.160068D+02	.186634D+02
.213145D+02	.239589D+02	.265954D+02	.292224D+02
.318381D+02	.344406D+02	.370272D+02	.395950D+02
.421400D+02	.446575D+02	.471414D+02	.495838D+02
.519741D+02	.542979D+02	.565353D+02	.586578D+02
.606228D+02	.623640D+02	.637729D+02	.646581D+02
.646513D+02	.629494D+02	.574792D+02	.417067D+02

F_i 的逆傅里叶变换 $P_i(i=1,2,\cdots,64)$的模:

.951229D+00	.860708D+00	.778801D+00	.704688D+00
.637628D+00	.576950D+00	.522046D+00	.472366D+00
.427415D+00	.386741D+00	.349938D+00	.316637D+00
.286505D+00	.259240D+00	.234570D+00	.212248D+00
.192050D+00	.173774D+00	.157237D+00	.142274D+00
.128735D+00	.116484D+00	.105399D+00	.953692D−01
.862936D−01	.780817D−01	.706512D−01	.639279D−01
.578443D−01	.523397D−01	.473590D−01	.428522D−01
.387742D−01	.350844D−01	.317457D−01	.287247D−01
.259912D−01	.235178D−01	.212798D−01	.192548D−01
.174224D−01	.157645D−01	.142643D−01	.129069D−01
.116786D−01	.105673D−01	.956167D−02	.865179D−02
.782840D−02	.708346D−02	.640938D−02	.579948D−02
.524756D−02	.474822D−02	.429637D−02	.388755D−02
.351755D−02	.318285D−02	.287996D−02	.260593D−02
.235792D−02	.213357D−02	.193053D−02	.174685D−02

11.3 快速沃什变换

一、功能

计算给定序列的沃什(walsh)变换序列。

二、方法说明

设给定序列

$$p_1, p_2, \cdots, p_n$$

其中 $n=2^k(k\geqslant 1)$。则沃什变换定义为

$$x_i = \sum_{j=1}^{n} W_{ij}^{(n)} p_j, \ i = 1,2,\cdots,n$$

沃什变换可以看作是矩阵与向量的乘积，即

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = W^{(n)} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}$$

其中

$$W^{(n)} = \begin{bmatrix} W_{11}^{(n)} & W_{12}^{(n)} & \cdots & W_{1n}^{(n)} \\ W_{21}^{(n)} & W_{22}^{(n)} & \cdots & W_{2n}^{(n)} \\ \vdots & \vdots & & \vdots \\ W_{n1}^{(n)} & W_{n2}^{(n)} & \cdots & W_{nn}^{(n)} \end{bmatrix}$$

是一个特殊的矩阵，其元素值只有两个：+1 与 −1。$W^{(n)}$中的各元素有以下递推关系($i=1,2,\cdots,n$)：

$$W^{(1)} = W_{11}^{(1)} = 1$$

$$W_{2i,j}^{(2n)} = (W_{i,j}^{(n)}, (-1)^{i-1} W_{i,j}^{(n)}),\ i=1,2,\cdots,n$$

$$W^{(2n)}_{2i+1,j}=(W^{(n)}_{i},(-1)^{i}W^{(n)}_{i}),\ i=1,2,\cdots,n$$

若利用通常的矩阵相乘的方法，计算沃什变换序列需要作 $n(n-1)$ 次加减法．本子程序采用与快速傅里叶变换类似的方法，加减法总次数为 $n\log_2 n$。

三、子程序语句

SUBROUTINE KKFWT(P,K,N,X)

四、形参说明

P——实型一维数组，长度为N，输入参数．存放长度为 $N=2^K$ 的输入序列。

K——整型变量，输入参数．要求满足 $N=2^K(K>0)$。

N——整型变量，输入参数。序列的长度，要求 $N=2^K(K>0)$。

X——实型一维数组，长度为N，输出参数。输入序列 $p_i(i=1,2,\cdots,n)$ 的沃什变换序列。

五、子程序(文件名：KKFWT.FOR)

```
      SUBROUTINE KKFWT(P,K,N,X)
      DIMENSION P(N),X(N)
      M=1
      L=N
      IT=2
      X(1)=1
      II=N/2+1
      X(II)=2
      DO 20 I=1,K-1
        M=M+M
        L=L/2
        IT=IT+IT
        DO 10 J=0,M-1
10      X(J*L+L/2+1)=IT+1-X(J*L+1)
20    CONTINUE
      DO 30 I=1,N
        II=X(I)
        X(I)=P(II)
30    CONTINUE
      L=1
      DO 60 I=1,K
        M=N/(2*L)-1
        DO 50 J=0,M
          IT=2*L*J
          DO 40 IS=0,L-1
            Q=X(IT+IS+1)+X(IT+IS+L+1)
            X(IT+IS+L+1)=X(IT+IS+1)-X(IT+IS+L+1)
            X(IT+IS+1)=Q
```

```
40        CONTINUE
50      CONTINUE
        L=2*L
60    CONTINUE
      RETURN
      END
```

六、例

设输入序列为

$$p_i=i, \quad i=1,2,\cdots,8$$

计算沃什变换序列 $x_i(i=1,2,\cdots,8)$。

其中 K=3,N=8。

主程序(文件名:KKFWT0.FOR)为

```
      DIMENSION P(8),X(8)
      DO 10 I=1,8
10    P(I)=I
      CALL KKFWT(P,3,8,X)
      WRITE(*,*)
      WRITE(*,20) (I,X(I),I=1,8)
20    FORMAT(1X,'x(',I2,')=',E13.6)
      WRITE(*,*)
      END
```

运行结果为

```
x( 1 )=  .360000E+02
x( 2 )= -.160000E+02
x( 3 )=  .000000E+00
x( 4 )= -.800000E+01
x( 5 )=  .000000E+00
x( 6 )=  .000000E+00
x( 7 )=  .000000E+00
x( 8 )= -.400000E+01
```

11.4 五点三次平滑

一、功能

用五点三次平滑公式对等距点上的观测数据进行平滑。

二、方法说明

设已知 n 个等距点

$$x_1<x_2<\cdots<x_{n-1}<x_n$$

上的观测(或实验)数据为

$$y_1,y_2,\cdots,y_{n-1},y_n$$

则在每个数据点的前后各取两个相邻的数据，用三次多项式

$$y=a_1+a_2x+a_3x^2+a_4x^3$$

根据最小二乘原理确定系数 a_1,a_2,a_3,a_4。最后可得到五点三次平滑公式如下：

$$\bar{y}_{i-2}=\frac{1}{70}(69y_{i-2}+4y_{i-1}-6y_i+4y_{i+1}-y_{i+2})$$

$$\bar{y}_{i-1}=\frac{1}{35}(2y_{i-2}+27y_{i-1}+12y_i-8y_{i+1}+2y_{i+2})$$

$$\bar{y}_i=\frac{1}{35}(-3y_{i-2}+12y_{i-1}+17y_i+12y_{i+1}-3y_{i+2})$$

$$\bar{y}_{i+1}=\frac{1}{35}(2y_{i-2}-8y_{i-1}+12y_i+27y_{i+1}+2y_{i+2})$$

$$\bar{y}_{i+2}=\frac{1}{70}(-y_{i-2}+4y_{i-1}-6y_i+4y_{i+1}+69y_{i+2})$$

其中 $\bar{y}_i$ 表示 y_i 的平滑值。

对于开始两点和最后两点分别由上述第 1，2 与 4，5 公式进行平滑。

本方法要求数据点数 $n\geqslant 5$。

三、子程序语句

SUBROUTINE KKSPT(N,Y,YY)

四、形参说明

N——整型变量，输入参数。等距观测点的个数，要求 N≥5。

Y——实型一维数组，长度为 N，输入参数。等距观测点上的数据。

YY——实型一维数组，长度为 N，输出参数。N 个等距观测点上的平滑结果。

五、子程序(文件名：KKSPT.FOR)

```
      SUBROUTINE KKSPT(N,Y,YY)
      DIMENSION Y(N),YY(N)
      IF (N.LT.5) THEN
        DO 10 I=1,N
10      YY(I)=Y(I)
        RETURN
      END IF
      YY(1)=(69.*Y(1)+4.*Y(2)-6.*Y(3)+4.*Y(4)-Y(5))/70.
      YY(2)=(2.*Y(1)+27.*Y(2)+12.*Y(3)-8.*Y(4)+2.*Y(5))/35.
      DO 20 I=3,N-2
20    YY(I)=(-3.*Y(I-2)+12.*Y(I-1)+17.*Y(I)+
     *       12.*Y(I+1)-3.*Y(I+2))/35.
      YY(N-1)=(2.*Y(N-4)-8.*Y(N-3)+12.*Y(N-2)+
     *       27.*Y(N-1)+2.*Y(N))/35.
      YY(N)=(-Y(N-4)+4.*Y(N-3)-6.*Y(N-2)+
     *       4.*Y(N-1)+69.*Y(N))/70.
      RETURN
      END
```

六、例

设等距观测点数据为

i	1	2	3	4	5	6	7	8	9
y_i	54.0	145.0	227.0	359.0	401.0	342.0	259.0	112.0	65.0

用五点三次平滑公式对上述观测数据进行平滑。

主程序(文件名:KKSPT0.FOR)为

```
      DIMENSION Y(9),YY(9)
      DATA Y/54.0,145.0,227.0,359.0,401.0,342.0,
     *         259.0,112.0,65.0/
      N=9
      CALL KKSPT(N,Y,YY)
      WRITE(*,*)
      WRITE(*,10)
      WRITE(*,20) (I,Y(I),YY(I),I=1,N)
      WRITE(*,*)
10    FORMAT(1X,5X,'I',12X,'Y(I)',20X,'YY(I)')
20    FORMAT(1X,4X,I2,6X,E15.6,9X,E15.6)
      END
```

运行结果为

```
I        Y(I)                     YY(I)
1     .540000E+02             .568429E+02
2     .145000E+03             .133629E+03
3     .227000E+03             .244057E+03
4     .359000E+03             .347943E+03
5     .401000E+03             .393457E+03
6     .342000E+03             .352029E+03
7     .259000E+03             .241514E+03
8     .112000E+03             .123657E+03
9     .650000E+02             .620857E+02
```

11.5 离散随机线性系统的卡尔曼滤波

一、功能

对时间离散点上的采样数据进行卡尔曼(Kalman)滤波。

二、方法说明

设 n 维线性动态系统与 m 维线性观测系统由下列差分方程组描述:

$$\begin{cases}X_k=\Phi_{k,k-1}X_{k-1}+W_{k-1}\\ Y_k=H_kX_k+V_k\end{cases},\quad k=1,2,\cdots$$

其中：

X_k 为 n 维向量，表示系统在第 k 时刻的状态。

$\Phi_{k,k-1}$是一个 $n\times n$ 阶矩阵，称为系统的状态转移矩阵，它反映了系统从第 $k-1$ 个采样时刻的状态到第 k 个采样时刻的状态的变换。

W_k 是一个 n 维向量，表示在第 k 时刻作用于系统的随机干扰，称为模型噪声。为简单起见，一般假设$\{W_k\}(k=1,2,\cdots)$为高斯白噪声序列，具有已知的零均值和协方差阵 Q_k。

Y_k 为 m 维的观测向量。

H_k 为 $m\times n$ 阶的观测矩阵，表示从状态量 X_k 到观测量 Y_k 的转换。

V_k 为 m 维的观测噪声，同样假设$\{V_k\}(k=1,2,\cdots)$为高斯白噪声序列，具有已知的零均值和协方差阵 R_k。

经推导（推导过程略），可得到如下递推滤波公式：

$$G_k=P_kH_k^T[H_kP_kH_k^T+R_k]^{-1}$$

$$\hat{X}_k=\Phi_{k,k-1}\hat{X}_{k-1}+G_k[Y_k-H_k\Phi_{k,k-1}\hat{X}_{k-1}]$$

$$C_k=(I-G_kH_k)P_k$$

$$P_{k+1}=\Phi_{k+1,k}C_k\Phi_{k+1,k}^T+Q_k$$

其中：

Q_k 为 $n\times n$ 阶的模型噪声 W_k 的协方差阵；

R_k 为 $m\times m$ 阶的观测噪声 V_k 的协方差阵；

G_k 为 $n\times m$ 阶的增益矩阵；

$\hat{X}_k$ 为 n 维向量，第 k 时刻经滤波后的估值；

C_k 为 $n\times n$ 阶的估计误差协方差阵。

根据上述公式，可以从 $\hat{X}_0=E\{X_0\}$、P_0（给定）出发，利用已知的矩阵 Q_k，R_k，H_k，$\Phi_{k,k-1}$与 k 时刻的观测值 Y_k，递推地算出每个时刻的状态估计 $\hat{X}_k(k=1,2,\cdots)$。

如果线性系统是定常的，则有 $\Phi_{k,k-1}=\Phi$，$H_k=H$，即它们都是常阵；如果模型噪声 W_k 与观测噪声 V_k 都是平稳随机序列，则 Q_k 与 R_k 都是常阵。在这种情况下，常增益的离散 Kalman 滤波是渐近稳定的。

三、子程序语句

SUBROUTINE KLMAN(N,M,K,F,Q,R,H,Y,X,P,G,L,E,IA,A,B,IS,JS)

四、形参说明

N——整型变量，输入参数．动态系统的维数．

M——整型变量，输入参数。观测系统的维数。

K——整型变量，输入参数．观测序列的长度。

F——双精度实型二维数组，体积为 N×N，输入参数。系统状态转移矩阵 Φ。

Q——双精度实型二维数组，体积为 N×N，输入参数．模型噪声 W_k 的协方差阵。

R——双精度实型二维数组，体积为 M×M，输入参数．观测噪声 V_k 的协方差阵。

H——双精度实型二维数组，体积为 M×N，输入参数。观测矩阵．

Y——双精度实型二维数组，体积为 K×M，输入参数。观测向量序列，Y(i,j) ($j=1$, 2,…,M)表示第 i 时刻的观测向量。

X——双精度实型二维数组,体积为 K×N,输入兼输出参数。动态向量估值序列。调用时其第 1 行存放初值 X(1,j)(j=1,2,…,N);返回时存放 K 个时刻的状态向量估值序列,X(i,j)(j=1,2,…,N)表示第 i 时刻的状态向量估值。

P——双精度实型二维数组,体积为 N×N,输入兼输出参数。调用时存放初值 P_0;返回时存放最后时刻的估计误差协方差阵。

G——双精度实型二维数组,体积为 N×M,输出参数。存放最后时刻的稳定增益矩阵。

L——整型变量,输出参数。若 L=0,说明求增益矩阵过程中求逆失败;若 L≠0,正常返回。

E——双精度实型二维数组,体积为 M×M。本子程序的工作数组。

IA——整型变量,输入参数。要求 IA=max{M,N}。

A,B——均为双精度实型二维数组,体积为 IA×IA。本子程序的工作数组。

IS,JS——均为整型一维数组,长度为 M。本子程序的工作数组。

五、子程序(文件名:KLMAN.FOR)

```
      SUBROUTINE KLMAN(N,M,K,F,Q,R,H,Y,X,P,G,L,E,IA,A,B,IS,JS)
      DIMENSION F(N,N),Q(N,N),R(M,M),H(M,N),Y(K,M),X(K,N)
      DIMENSION P(N,N),G(N,M),E(M,M),A(IA,IA),B(IA,IA)
      DIMENSION IS(M),JS(M)
      DOUBLE PRECISION F,Q,R,H,Y,X,P,G,E,A,B
      DO 2 I=1,N
      DO 2 J=1,N
        A(I,J)=0.0
        DO 1 KK=1,N
1       A(I,J)=A(I,J)+P(I,KK)*F(J,KK)
2     CONTINUE
      DO 4 I=1,N
      DO 4 J=1,N
        P(I,J)=Q(I,J)
        DO 3 KK=1,N
3       P(I,J)=P(I,J)+F(I,KK)*A(KK,J)
4     CONTINUE
      DO 300 II=2,K
        DO 20 I=1,N
        DO 20 J=1,M
          A(I,J)=0.0
          DO 10 KK=1,N
10        A(I,J)=A(I,J)+P(I,KK)*H(J,KK)
20      CONTINUE
        DO 40 I=1,M
```

```
      DO 40 J=1,M
        E(I,J)=R(I,J)
        DO 30 KK=1,N
30      E(I,J)=E(I,J)+H(I,KK)*A(KK,J)
40    CONTINUE
      CALL BRINV(E,M,L,IS,JS)
      IF (L.EQ.0) RETURN
      DO 60 I=1,N
      DO 60 J=1,M
        G(I,J)=0.0
        DO 50 KK=1,M
50      G(I,J)=G(I,J)+A(I,KK)*E(J,KK)
60    CONTINUE
      DO 80 I=1,N
        X(II,I)=0.0
        DO 70 J=1,N
70      X(II,I)=X(II,I)+F(I,J)*X(II-1,J)
80    CONTINUE
      DO 100 I=1,M
        B(I,1)=Y(II,I)
        DO 90 J=1,N
90      B(I,1)=B(I,1)-H(I,J)*X(II,J)
100   CONTINUE
      DO 120 I=1,N
        DO 110 J=1,M
110     X(II,I)=X(II,I)+G(I,J)*B(J,1)
120   CONTINUE
      IF (II.LT.K) THEN
        DO 140 I=1,N
        DO 140 J=1,N
          A(I,J)=0.0
          DO 130 KK=1,M
130       A(I,J)=A(I,J)-G(I,KK)*H(KK,J)
          IF (I.EQ.J) A(I,J)=1.0+A(I,J)
140     CONTINUE
        DO 160 I=1,N
        DO 160 J=1,N
          B(I,J)=0.0
          DO 150 KK=1,N
150       B(I,J)=B(I,J)+A(I,KK)*P(KK,J)
160     CONTINUE
        DO 180 I=1,N
```

```
      DO 180 J=1,N
        A(I,J)=0.0
        DO 170 KK=1,N
170     A(I,J)=A(I,J)+B(I,KK)*F(J,KK)
180   CONTINUE
      DO 200 I=1,N
      DO 200 J=1,N
        P(I,J)=Q(I,J)
        DO 190 KK=1,N
190     P(I,J)=P(I,J)+F(I,KK)*A(J,KK)
200   CONTINUE
    END IF
300 CONTINUE
    L=1
    RETURN
    END
```

六、例

设信号源运动方程为

$$s(t)=5-2t+3t^2+v(t)$$

其中 $v(t)$ 是一个均值为零、方差为 0.25 的高斯白噪声。

状态向量为 $X_k=(s,s',s'')^T$，取初值为 $X_0=(5.0,-2.0,6.0)^T$

状态转移矩阵为

$$F=\Phi_{k,k-1}=\begin{bmatrix}1.0 & T & \frac{T^2}{2}\\ 0.0 & 1.0 & T\\ 0.0 & 0.0 & 1.0\end{bmatrix}$$

其中 T 为采样间隔，在本例中取 $T=0.05$。

动态系统维数为 N=3，观测系统维数为 M=1。

模型噪声协方差阵为

$$Q=\begin{bmatrix}0.1 & 0.0 & 0.0\\ 0.0 & 0.1 & 0.0\\ 0.0 & 0.0 & 0.1\end{bmatrix}$$

观测矩阵为 $H=(1.0,0.0,0.0)$

观测噪声协方差阵为 $R=0.25$

初始估计误差协方差阵为

$$P_0=\begin{bmatrix}0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.0\end{bmatrix}$$

取观测向量序列长度为 K=200。

对输出的状态向量序列每隔 5 个时刻打印一次。

主程序(文件名:KLMAN0.FOR)为

```
      DIMENSION F(3,3),Q(3,3),R(1,1),H(1,3),Y(200,1),JS(1)
      DIMENSION X(200,3),P(3,3),G(3,1),E(1,1),A(3,3),B(3,3),IS(1)
      DOUBLE PRECISION F,Q,R,H,Y,X,P,G,E,S,A,B,RR
      REAL NGRN1
      DATA F/1.0,0.0,0.0,0.05,1.0,0.0,0.00125,0.05,1.0/
      DATA Q/0.1,3*0.0,0.1,3*0.0,0.1/
      DATA R/0.25/
      DATA H/1.0,0.0,0.0/
      DATA P/9*0.0/
      DATA X/5.0,199*0.0,-2.0,199*0.0,6.0,199*0.0/
      RR=3.0
      M=1
      N=3
      IA=3
      K=200
      DO 10 I=1,K
        T=(I-1)*0.05
        U=NGRN1(0.0,0.5,RR)
        Y(I,1)=5.0-2.0*T+3.0*T*T+U
10    CONTINUE
      CALL KLMAN(N,M,K,F,Q,R,H,Y,X,P,G,L,E,IA,A,B,IS,JS)
      WRITE(*,*)
      IF (L.NE.0) THEN
        WRITE(*,30)
        DO 20 I=1,K,5
          T=(I-1)*0.05
          S=5.0-2.0*T+3.0*T*T
          WRITE(*,40) T,S,Y(I,1),X(I,1),X(I,2),X(I,3)
20      CONTINUE
      END IF
30    FORMAT(4X,'T',8X,'S',12X,'Y',12X,
     *         'X(1)',9X,'X(2)',9X,'X(3)')
40    FORMAT(1X,F6.2,5D13.6)
      WRITE(*,*)
      END
```

在上述主程序中,调用了子程序 NGRN1,产生一个正态分布的随机数,参看 13.5 节。

运行结果为(其中:T 为采样时刻值;S 为真值,Y 为迭加有高斯白噪声的采样值;X(1),X(2),X(3)为状态向量各分量的估值)。

T	S	Y	X(1)	X(2)	X(3)
.00	.500000D+01	.452919D+01	.500000D+01	−.200000D+01	.600000D+01
.25	.468750D+01	.486378D+01	.493571D+01	−.490743D+00	.600090D+01
.50	.475000D+01	.428627D+01	.430317D+01	.908784D+00	.598912D+01
.75	.518750D+01	.545290D+01	.534466D+01	.261525D+01	.603734D+01
1.00	.600000D+01	.601991D+01	.613120D+01	.409616D+01	.602796D+01
1.25	.718750D+01	.764357D+01	.752414D+01	.572228D+01	.607342D+01
1.50	.875000D+01	.848012D+01	.881456D+01	.698739D+01	.596894D+01
1.75	.106875D+02	.106858D+02	.107348D+02	.844523D+01	.594568D+01
2.00	.130000D+02	.129169D+02	.130712D+02	.995959D+01	.595479D+01
2.25	.156875D+02	.148296D+02	.152859D+02	.109318D+02	.588636D+01
2.50	.187500D+02	.185802D+02	.189215D+02	.130511D+02	.600474D+01
2.75	.221875D+02	.218250D+02	.220604D+02	.142252D+02	.584478D+01
3.00	.260000D+02	.257201D+02	.257853D+02	.156549D+02	.582221D+01
3.25	.301875D+02	.299219D+02	.301774D+02	.173756D+02	.593115D+01
3.50	.347500D+02	.340866D+02	.344844D+02	.186204D+02	.581384D+01
3.75	.396875D+02	.398704D+02	.400457D+02	.208251D+02	.617171D+01
4.00	.450000D+02	.459296D+02	.452854D+02	.222537D+02	.614011D+01
4.25	.506875D+02	.509204D+02	.505342D+02	.232704D+02	.589734D+01
4.50	.567500D+02	.564992D+02	.565427D+02	.247190D+02	.587064D+01
4.75	.631875D+02	.633220D+02	.634017D+02	.267550D+02	.616404D+01
5.00	.700000D+02	.695453D+02	.696776D+02	.276312D+02	.583184D+01
5.25	.771875D+02	.773252D+02	.773505D+02	.297220D+02	.614479D+01
5.50	.847500D+02	.863180D+02	.854181D+02	.317995D+02	.643817D+01
5.75	.926875D+02	.936799D+02	.930965D+02	.329776D+02	.625914D+01
6.00	.101000D+03	.101067D+03	.101109D+03	.341047D+02	.605347D+01
6.25	.109688D+03	.109136D+03	.109484D+03	.352076D+02	.583299D+01
6.50	.118750D+03	.118043D+03	.118377D+03	.365433D+02	.576208D+01
6.75	.128188D+03	.128944D+03	.128611D+03	.390052D+02	.627828D+01
7.00	.138000D+03	.138495D+03	.138290D+03	.403375D+02	.618392D+01
7.25	.148188D+03	.147854D+03	.148205D+03	.414579D+02	.595973D+01
7.50	.158750D+03	.158674D+03	.158702D+03	.429216D+02	.595984D+01
7.75	.169688D+03	.169115D+03	.169373D+03	.441717D+02	.585179D+01
8.00	.181000D+03	.180830D+03	.180888D+03	.459171D+02	.597831D+01
8.25	.192688D+03	.192477D+03	.192356D+03	.471874D+02	.586385D+01
8.50	.204750D+03	.204712D+03	.204616D+03	.490023D+02	.605487D+01
8.75	.217188D+03	.217191D+03	.217208D+03	.506779D+02	.613373D+01
9.00	.230000D+03	.230571D+03	.230244D+03	.524492D+02	.627211D+01
9.25	.243188D+03	.243507D+03	.243722D+03	.541689D+02	.633364D+01
9.50	.256750D+03	.256156D+03	.256552D+03	.547093D+02	.579432D+01
9.75	.270688D+03	.270674D+03	.270456D+03	.562002D+02	.581294D+01

若将状态向量初值取为 $X_0=(1.0,0.0,0.0)^T$，则运行结果为(主程序略)

T	S	Y	X(1)	X(2)	X(3)
.00	.500000D+01	.452919D+01	.100000D+01	.000000D+00	.000000D+00
.25	.468750D+01	.486378D+01	.472087D+01	.648678D−01	.351771D−02

.50	.475000D+01	.428627D+01	.426731D+01	.175159D−02	−.288177D−02
.75	.518750D+01	.545290D+01	.524170D+01	.335827D+00	.758458D−01
1.00	.600000D+01	.601991D+01	.596410D+01	.682197D+00	.178819D+00
1.25	.718750D+01	.764357D+01	.731066D+01	.149384D+01	.465890D+00
1.50	.875000D+01	.848012D+01	.857383D+01	.228946D+01	.747628D+00
1.75	.106875D+02	.106858D+02	.104842D+02	.359532D+01	.123529D+01
2.00	.130000D+02	.129169D+02	.128245D+02	.521390D+01	.183701D+01
2.25	.156875D+02	.148296D+02	.150530D+02	.647414D+01	.219465D+01
2.50	.187500D+02	.185802D+02	.187088D+02	.899784D+01	.313079D+01
2.75	.221875D+02	.218250D+02	.218712D+02	.106379D+02	.355022D+01
3.00	.260000D+02	.257201D+02	.256209D+02	.125544D+02	.405011D+01
3.25	.301875D+02	.299219D+02	.300377D+02	.147540D+02	.461543D+01
3.50	.347500D+02	.340866D+02	.343681D+02	.164510D+02	.488597D+01
3.75	.396875D+02	.398704D+02	.399510D+02	.190697D+02	.556518D+01
4.00	.450000D+02	.459296D+02	.452100D+02	.208678D+02	.579315D+01
4.25	.506875D+02	.509204D+02	.504758D+02	.222073D+02	.575445D+01
4.50	.567500D+02	.564992D+02	.564990D+02	.239320D+02	.588323D+01
4.75	.631875D+02	.633220D+02	.633703D+02	.261997D+02	.629057D+01
5.00	.700000D+02	.695453D+02	.696565D+02	.272664D+02	.603753D+01
5.25	.771875D+02	.773252D+02	.773377D+02	.295105D+02	.640116D+01
5.50	.847500D+02	.863180D+02	.854117D+02	.317081D+02	.672242D+01
5.75	.926875D+02	.936799D+02	.930952D+02	.329779D+02	.655352D+01
6.00	.101000D+03	.101067D+03	.101112D+03	.341722D+02	.634455D+01
6.25	.109688D+03	.109136D+03	.109489D+03	.353220D+02	.611104D+01
6.50	.118750D+03	.118043D+03	.118384D+03	.366880D+02	.602043D+01
6.75	.128188D+03	.128944D+03	.128619D+03	.391668D+02	.651276D+01
7.00	.138000D+03	.138495D+03	.138298D+03	.405057D+02	.639231D+01
7.25	.148188D+03	.147854D+03	.148213D+03	.416247D+02	.614138D+01
7.50	.158750D+03	.158674D+03	.158710D+03	.430814D+02	.611523D+01
7.75	.169688D+03	.169115D+03	.169381D+03	.443204D+02	.598225D+01
8.00	.181000D+03	.180830D+03	.180895D+03	.460524D+02	.608570D+01
8.25	.192688D+03	.192477D+03	.192362D+03	.473078D+02	.595039D+01
8.50	.204750D+03	.204712D+03	.204622D+03	.491075D+02	.612295D+01
8.75	.217188D+03	.217191D+03	.217213D+03	.507680D+02	.618578D+01
9.00	.230000D+03	.230571D+03	.230248D+03	.525249D+02	.631048D+01
9.25	.243188D+03	.243507D+03	.243725D+03	.542314D+02	.636058D+01
9.50	.256750D+03	.256156D+03	.256555D+03	.547597D+02	.581189D+01
9.75	.270688D+03	.270674D+03	.270458D+03	.562400D+02	.582299D+01

若将状态向量初值取为 $X_0=(0.0,0.0,0.0)^T$。则运行结果为(主程序略)

T	S	Y	X(1)	X(2)	X(3)
.00	.500000D+01	.452919D+01	.000000D+00	.000000D+00	.000000D+00
.25	.468750D+01	.486378D+01	.465351D+01	.810319D−01	.446234D−02
.50	.475000D+01	.428627D+01	.426573D+01	.268803D−01	−.936815D−03
.75	.518750D+01	.545290D+01	.524295D+01	.360104D+00	.774544D−01
1.00	.600000D+01	.601991D+01	.596532D+01	.703757D+00	.179455D+00

1.25	.718750D+01	.764357D+01	.731168D+01	.151192D+01	.465180D+00
1.50	.875000D+01	.848012D+01	.857465D+01	.230375D+01	.745426D+00
1.75	.106875D+02	.106858D+02	.104848D+02	.360585D+01	.123167D+01
2.00	.130000D+02	.129169D+02	.128249D+02	.522099D+01	.183223D+01
2.25	.156875D+02	.148296D+02	.150532D+02	.647823D+01	.218904D+01
2.50	.187500D+02	.185802D+02	.187089D+02	.899956D+01	.312469D+01
2.75	.221875D+02	.218250D+02	.218712D+02	.106378D+02	.354395D+01
3.00	.260000D+02	.257201D+02	.256209D+02	.125529D+02	.404393D+01
3.25	.301875D+02	.299219D+02	.300376D+02	.147514D+02	.460953D+01
3.50	.347500D+02	.340866D+02	.343680D+02	.164479D+02	.488049D+01
3.75	.396875D+02	.398704D+02	.399508D+02	.190663D+02	.556022D+01
4.00	.450000D+02	.459296D+02	.452099D+02	.208612D+02	.578874D+01
4.25	.506875D+02	.509204D+02	.504756D+02	.222037D+02	.575062D+01
4.50	.567500D+02	.564992D+02	.564988D+02	.239286D+02	.587996D+01
4.75	.631875D+02	.633220D+02	.633702D+02	.261966D+02	.628783D+01
5.00	.700000D+02	.695453D+02	.696563D+02	.272635D+02	.603527D+01
5.25	.771875D+02	.773252D+02	.773375D+02	.295079D+02	.639935D+01
5.50	.847500D+02	.863180D+02	.854116D+02	.317059D+02	.672100D+01
5.75	.926875D+02	.936799D+02	.930951D+02	.329760D+02	.655244D+01
6.00	.101000D+03	.101067D+03	.101112D+03	.341706D+02	.634376D+01
6.25	.109688D+03	.109136D+03	.109489D+03	.353207D+02	.611049D+01
6.50	.118750D+03	.118043D+03	.118384D+03	.366870D+02	.602008D+01
6.75	.128188D+03	.128944D+03	.128619D+03	.391659D+02	.651256D+01
7.00	.138000D+03	.138495D+03	.138298D+03	.405050D+02	.639224D+01
7.25	.148188D+03	.147854D+03	.148213D+03	.416242D+02	.614140D+01
7.50	.158750D+03	.158674D+03	.158710D+03	.430811D+02	.611532D+01
7.75	.169688D+03	.169115D+03	.169381D+03	.443202D+02	.598238D+01
8.00	.181000D+03	.180830D+03	.180895D+03	.460523D+02	.608586D+01
8.25	.192688D+03	.192477D+03	.192362D+03	.473078D+02	.595057D+01
8.50	.204750D+03	.204712D+03	.204622D+03	.491075D+02	.612313D+01
8.75	.217188D+03	.217191D+03	.217213D+03	.507680D+02	.618596D+01
9.00	.230000D+03	.230571D+03	.230248D+03	.525250D+02	.631065D+01
9.25	.243188D+03	.243507D+03	.243725D+03	.542314D+02	.636074D+01
9.50	.256750D+03	.256156D+03	.256555D+03	.547598D+02	.581203D+01
9.75	.270688D+03	.270674D+03	.270158D+03	.562401D+02	.582311D+01

若将状态向量初值取为 $X_0=(0.0,0.0,0.0)^T$，模型噪声协方差阵取为

$$Q=\begin{bmatrix}0.25 & 0.0 & 0.0\\ 0.0 & 0.25 & 0.0\\ 0.0 & 0.0 & 0.25\end{bmatrix}$$

则运行结果为(主程序略)

T	S	Y	X(1)	X(2)	X(3)
.00	.500000D+01	.452919D+01	.000000D+00	.000000D+00	.000000D+00
.25	.468750D+01	.486378D+01	.490800D+01	.306350D 01	.134930D−02
.50	.475000D+01	.428627D+01	.426808D+01	−.387102D−01	.471116D 02
.75	.518750D+01	.545290D+01	.534922D+01	.276091D+00	.712244D−01

1.00	.600000D+01	.601991D+01	.600831D+01	.552067D+00	.154007D+00
1.25	.718750D+01	.764357D+01	.743148D+01	.130909D+01	.425992D+00
1.50	.875000D+01	.848012D+01	.861174D+01	.197961D+01	.661003D+00
1.75	.106875D+02	.106858D+02	.105559D+02	.320745D+01	.112391D+01
2.00	.130000D+02	.129169D+02	.128833D+02	.471237D+01	.168346D+01
2.25	.156875D+02	.148296D+02	.149870D+02	.584547D+01	.199617D+01
2.50	.187500D+02	.185802D+02	.187237D+02	.831572D+01	.291856D+01
2.75	.221875D+02	.218250D+02	.218914D+02	.998011D+01	.337084D+01
3.00	.260000D+02	.257201D+02	.256373D+02	.119015D+02	.388955D+01
3.25	.301875D+02	.299219D+02	.300136D+02	.140612D+02	.444804D+01
3.50	.347500D+02	.340866D+02	.343379D+02	.158220D+02	.476455D+01
3.75	.396875D+02	.398704D+02	.399489D+02	.184486D+02	.545244D+01
4.00	.450000D+02	.459296D+02	.453829D+02	.205102D+02	.582670D+01
4.25	.506875D+02	.509204D+02	.505727D+02	.218939D+02	.581366D+01
4.50	.567500D+02	.564992D+02	.564587D+02	.235495D+02	.590096D+01
4.75	.631875D+02	.633220D+02	.634102D+02	.259067D+02	.634328D+01
5.00	.700000D+02	.695453D+02	.696069D+02	.270221D+02	.611136D+01
5.25	.771875D+02	.773252D+02	.773503D+02	.293253D+02	.648873D+01
5.50	.847500D+02	.863180D+02	.856373D+02	.317468D+02	.691316D+01
5.75	.926875D+02	.936799D+02	.932291D+02	.330101D+02	.672733D+01
6.00	.101000D+03	.101067D+03	.101113D+03	.341525D+02	.647605D+01
6.25	.109688D+03	.109136D+03	.109379D+03	.353684D+02	.621017D+01
6.50	.118750D+03	.118043D+03	.118306D+03	.367105D+02	.614156D+01
6.75	.128188D+03	.128944D+03	.128742D+03	.393086D+02	.667273D+01
7.00	.138000D+03	.138495D+03	.138379D+03	.406401D+02	.653417D+01
7.25	.148188D+03	.147854D+03	.148106D+03	.416209D+02	.619930D+01
7.50	.158750D+03	.158674D+03	.158730D+03	.432169D+02	.623233D+01
7.75	.169688D+03	.169115D+03	.169352D+03	.444233D+02	.606812D+01
8.00	.181000D+03	.180830D+03	.180864D+03	.461229D+02	.614257D+01
8.25	.192688D+03	.192477D+03	.192366D+03	.474360D+02	.603103D+01
8.50	.204750D+03	.204712D+03	.204686D+03	.492466D+02	.619672D+01
8.75	.217188D+03	.217191D+03	.217220D+03	.508220D+02	.620883D+01
9.00	.230000D+03	.230571D+03	.230362D+03	.526567D+02	.637017D+01
9.25	.243188D+03	.243507D+03	.243679D+03	.541767D+02	.632136D+01
9.50	.256750D+03	.256156D+03	.256403D+03	.547125D+02	.578505D+01
9.75	.270688D+03	.270674D+03	.270494D+03	.563791D+02	.589060D+01

若将状态向量初值取为$X_0=(0.0,0.0,0.0)^T$,模型噪声协方差阵取为

$$Q=\begin{bmatrix}0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.1\end{bmatrix}$$

则运行结果为(主程序略)

T	S	Y	X(1)	X(2)	X(3)
.00	.500000D+01	.452919D+01	.000000D+00	.000000D+00	.000000D+00
.25	.468750D+01	.486378D+01	.167563D−02	.169908D 01	.898157D−01

.50	.475000D+01	.428627D+01	.118204D+00	.568803D+00	.142718D+01
.75	.518750D+01	.545290D+01	.139357D+01	.446020D+01	.742770D+01
1.00	.600000D+01	.601991D+01	.482101D+01	.114214D+02	.139999D+02
1.25	.718750D+01	.764357D+01	.821343D+01	.150829D+02	.141014D+02
1.50	.875000D+01	.848012D+01	.104143D+02	.147863D+02	.101077D+02
1.75	.106875D+02	.106858D+02	.122199D+02	.131373D+02	.577423D+01
2.00	.130000D+02	.129169D+02	.139828D+02	.113198D+02	.244568D+01
2.25	.156875D+02	.148296D+02	.159096D+02	.996056D+01	.402039D+00
2.50	.187500D+02	.185802D+02	.187363D+02	.107044D+02	.104647D+01
2.75	.221875D+02	.218250D+02	.217639D+02	.116044D+02	.169080D+01
3.00	.260000D+02	.257201D+02	.254062D+02	.133739D+02	.301530D+01
3.25	.301875D+02	.299219D+02	.298507D+02	.160365D+02	.483520D+01
3.50	.347500D+02	.340866D+02	.343202D+02	.178632D+02	.545422D+01
3.75	.396875D+02	.398704D+02	.397956D+02	.208402D+02	.701239D+01
4.00	.450000D+02	.459296D+02	.451104D+02	.224167D+02	.688946D+01
4.25	.506875D+02	.509204D+02	.505772D+02	.234563D+02	.622986D+01
4.50	.567500D+02	.564992D+02	.566819D+02	.250630D+02	.623591D+01
4.75	.631875D+02	.633220D+02	.632429D+02	.268761D+02	.654496D+01
5.00	.700000D+02	.695453D+02	.698761D+02	.279253D+02	.594547D+01
5.25	.771875D+02	.773252D+02	.772553D+02	.298428D+02	.638406D+01
5.50	.847500D+02	.863180D+02	.850798D+02	.318027D+02	.679099D+01
5.75	.926875D+02	.936799D+02	.929263D+02	.329715D+02	.636172D+01
6.00	.101000D+03	.101067D+03	.101128D+03	.341239D+02	.596117D+01
6.25	.109688D+03	.109136D+03	.109721D+03	.353212D+02	.562239D+01
6.50	.118750D+03	.118043D+03	.118539D+03	.363443D+02	.522935D+01
6.75	.128188D+03	.128944D+03	.128247D+03	.385817D+02	.618028D+01
7.00	.138000D+03	.138495D+03	.138078D+03	.401675D+02	.627818D+01
7.25	.148188D+03	.147854D+03	.148360D+03	.417838D+02	.628188D+01
7.50	.158750D+03	.158674D+03	.158700D+03	.428133D+02	.580109D+01
7.75	.169688D+03	.169115D+03	.169361D+03	.438808D+02	.547503D+01
8.00	.181000D+03	.180830D+03	.180856D+03	.459038D+02	.607992D+01
8.25	.192688D+03	.192477D+03	.192431D+03	.472260D+02	.586565D+01
8.50	.204750D+03	.204712D+03	.204421D+03	.487643D+02	.600702D+01
8.75	.217188D+03	.217191D+03	.217103D+03	.508457D+02	.656453D+01
9.00	.230000D+03	.230571D+03	.230042D+03	.525792D+02	.670727D+01
9.25	.243188D+03	.243507D+03	.243753D+03	.548958D+02	.728076D+01
9.50	.256750D+03	.256156D+03	.257105D+03	.554637D+02	.596719D+01
9.75	.270688D+03	.270674D+03	.270649D+03	.559621D+02	.497524D+01

七、附注

本子程序需要调用全选主元高斯-约当法求逆子程序 BRINV，参看 2.3 节。

11.6 α-β-γ 滤波

一、功能

对等间隔的量测数据进行滤波估值。

二、方法说明

设一个过程的量测数据为 $X^*(t)$，且

$$X^*(t)=X(t)+\eta(t)$$

其中 $X(t)$ 为有用信号的准确值；$\eta(t)$ 是均值为零的白噪声过程，即有

$$E\{\eta(t)\}=0$$

$$E\{\eta(t)\eta(\tau)\}=\gamma\delta(t-\tau)$$

采用 α-β-γ 滤波方法对量测数据序列 X^* 进行估值的计算公式如下：

由上时刻对本时刻的一步预测估值公式为

$$\hat{X}_{n+1/n}=\hat{X}_n+\hat{X}'_nT+\hat{X}''_n(T^2/2)$$

$$\hat{X}'_{n+1/n}=\hat{X}'_n+\hat{X}''_nT$$

$$\hat{X}''_{n+1/n}=\hat{X}''_n$$

本时刻的滤波估值公式为

$$\hat{X}_{n+1}=\hat{X}_{n+1/n}+\alpha(X^*_{n+1}-\hat{X}_{n+1/n})$$

$$\hat{X}'_{n+1}=\hat{X}'_{n+1/n}+\frac{\beta}{T}(X^*_{n+1}-\hat{X}_{n+1/n})$$

$$\hat{X}''_{n+1}=\hat{X}''_{n+1/n}+\frac{2\gamma}{T^2}(X^*_{n+1}-\hat{X}_{n+1/n})$$

其中：

X^*_{n+1} 为本时刻的量测值；

$\hat{X}_{n+1/n}$ 为上时刻对本时刻的位置一步预测估值；

$\hat{X}'_{n+1/n}$ 为上时刻对本时刻的速度一步预测估值；

$\hat{X}''_{n+1/n}$ 为上时刻对本时刻的加速度一步预测估值；

$\hat{X}_{n+1}$ 为本时刻的位置滤波估值；

$\hat{X}'_{n+1}$ 为本时刻的速度滤波估值；

$\hat{X}''_{n+1}$ 为本时刻的加速度滤波估值；

T 为采样间隔；

α,β,γ 为滤波器结构参数。

三、子程序语句

SUBROUTINE KKABG(N,X,T,A,B,C,Y)

四、形参说明

N——整型变量，输入参数。量测值的点数。

X——实型一维数组，长度为 N，输入参数。N 个等间隔点上的量测值。

T——实型变量，输入参数。采样周期，即量测点的间隔 Δt。

A,B,C——均为实型变量，输入参数。滤波器结构参数 α,β,γ。

Y——实型一维数组，长度为 N，输出参数。存放 N 个等间隔点上的滤波估值。

五、子程序(文件名:KKABG.FOR)

```
      SUBROUTINE KKABG(N,X,T,A,B,C,Y)
      DIMENSION X(N),Y(N)
      DO 10 I=1,N
```

```
      S1=SS+T*VV+T*T*AA/2.0
      V1=VV+T*AA
      A1=AA
      SS=S1+A*(X(I)-S1)
      Y(I)=SS
      VV=V1+B*(X(I)-S1)
      AA=A1+2.0*C*(X(I)-S1)/(T*T)
10    CONTINUE
      RETURN
      END
```

六、例

设准确信号为

$$Z(t)=3t^2-2t+5$$

以 $\Delta t=0.04$ 采样 250 个点，并且迭加上一个均值为零、标准差为 0.5 的正态分布白噪声。

取 $\alpha=0.271$，$\beta=0.0285$，$\gamma=0.0005$。对此 250 个采样点值进行滤波估值。

主程序(文件名，KKABG0.FOR)为

(在本主程序中，调用子程序 NGRN1，产生一个正态分布随机数，参看 13.5 节。)

```
      DIMENSION X(250),Y(250),Z(250)
      REAL NGRN1
      DOUBLE PRECISION R
      N=250
      WRITE(*,*)
      WRITE(*,5)
5     FORMAT(1X,'A,B,C=?')
      READ(*,*) A,B,C
      WRITE(*,100) A,B,C
100   FORMAT(1X,3F10.6)
      DT=0.04
      R=1.0
      DO 20 I=1,N
        S=NGRN1(0.0,0.5,R)
        T=I*DT
        Z(I)=3.0*T*T-2.0*T+5.0
        X(I)=Z(I)+S
20    CONTINUE
      WRITE(*,*)
      CALL KKABG(N,X,DT,A,B,C,Y)
      WRITE(*,30)
30    FORMAT(4X,'T',16X,'X(T)',11X,'Y(T)',11X,'Z(T)')
      DO 40 I=1,N,5
        T=I*DT
```

```
      WRITE(*,50) T,X(I),Y(I),Z(I)
40    CONTINUE
50    FORMAT(1X,F5.2,10X,3E15.6)
      WRITE(*,*)
      END
```

运行结果为

```
A,B,C=?
 .271000      .028500      .000500
```

T	X(T)	Y(T)	Z(T)
.04	.440364E+01	.119339E+01	.492480E+01
.24	.498963E+01	.434103E+01	.469280E+01
.44	.388789E+01	.467183E+01	.470080E+01
.64	.500466E+01	.521915E+01	.494880E+01
.84	.524620E+01	.570225E+01	.543680E+01
1.04	.601876E+01	.639615E+01	.616480E+01
1.24	.672858E+01	.744597E+01	.713280E+01
1.44	.928192E+01	.893706E+01	.834080E+01
1.64	.100850E+02	.102175E+02	.978880E+01
1.84	.110441E+02	.115928E+02	.114768E+02
2.04	.140635E+02	.135817E+02	.134048E+02
2.24	.160534E+02	.156284E+02	.155728E+02
2.44	.179201E+02	.177037E+02	.179808E+02
2.64	.20[illegible]658E+02	.202964E+02	.206288E+02
2.84	.238087E+02	.234925E+02	.235168E+02
3.04	.273252E+02	.265535E+02	.266448E+02
3.24	.307511E+02	.296828E+02	.300128E+02
3.44	.330836E+02	.329198E+02	.336208E+02
3.64	.382281E+02	.372218E+02	.374688E+02
3.84	.410912E+02	.415027E+02	.415568E+02
4.04	.460790E+02	.459679E+02	.458848E+02
4.24	.505978E+02	.508325E+02	.504528E+02
4.44	.555538E+02	.557039E+02	.552608E+02
4.64	.613534E+02	.611842E+02	.603088E+02
4.84	.654028E+02	.662041E+02	.655968E+02
5.04	.711082E+02	.714238E+02	.711248E+02
5.24	.768758E+02	.772506E+02	.768928E+02
5.44	.831119E+02	.833998E+02	.829008E+02
5.64	.897228E+02	.896347E+02	.891488E+02
5.84	.966148E+02	.965031E+02	.956368E+02
6.04	.101694E+03	.102524E+03	.102365E+03
6.24	.109367E+03	.109063E+03	.10933[illegible]E+03
6.44	.116539E+03	.116180E+03	.116541E+03
6.64	.123618E+03	.123621E+03	.123989E+03

6.84	.131508E+03	.130807E+03	.131677E+03
7.04	.140118E+03	.138759E+03	.139605E+03
7.24	.147352E+03	.146838E+03	.147773E+03
7.44	.156117E+03	.155397E+03	.156181E+03
7.64	.164819E+03	.164242E+03	.164829E+03
7.84	.174365E+03	.173491E+03	.173717E+03
8.04	.182161E+03	.182345E+03	.182845E+03
8.24	.192612E+03	.192239E+03	.192213E+03
8.44	.201626E+03	.202012E+03	.201821E+03
8.64	.211408E+03	.211908E+03	.211669E+03
8.84	.221966E+03	.222637E+03	.221757E+03
9.04	.232104E+03	.232681E+03	.232085E+03
9.24	.242429E+03	.243477E+03	.242653E+03
9.44	.253848E+03	.254342E+03	.253461E+03
9.64	.264267E+03	.265051E+03	.264509E+03
9.84	.276092E+03	.276487E+03	.275797E+03

第12章 特殊函数

12.1 伽马函数

一、功能

计算实变量 x 的伽马(Gamma)函数值

$$\Gamma(x)=\int_0^{\infty} e^{-t} t^{x-1} dt \qquad (x>0)$$

二、方法说明

当 $2<x\leqslant 3$ 时,$\Gamma(x)$用下列切比雪夫(Chebyshev)多项式逼近:

$$\Gamma(x)=\sum_{k=0}^{10} a_k (x-2)^{10-k}$$

其中

$a_0=0.0000677106$,　　$a_1=-0.0003442342$

$a_2=0.0015397681$,　　$a_3=-0.0024467480$

$a_4=0.0109736958$,　　$a_5=-0.0002109075$

$a_6=0.0742379071$,　　$a_7=0.0815782188$

$a_8=0.4118402518$,　　$a_9=0.4227843370$

$a_{10}=1.0$

当 $0<x\leqslant 2$ 时,利用公式

$$\Gamma(x)=\frac{1}{x}\Gamma(x+1),\ 1<x\leqslant 2$$

或

$$\Gamma(x)=\frac{1}{x(x+1)}\Gamma(x+2),\ 0<x\leqslant 1$$

当 $x>3$ 时,利用公式

$$\Gamma(x)=(x-1)(x-2)\cdots(x-k)\Gamma(x-k)$$

直到满足 $2<(x-k)\leqslant 3$ 为止。

利用伽马函数可以计算贝塔(Beta)函数

$$B(x,y)=B(y,x)=\int_0^1 t^{x-1}(1-t)^{y-1}dt,\ x,y>0$$

其计算公式为

$$B(x,y)=\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

三、子程序语句

FUNCTION MGAM1(X)

四、形参说明

X——双精度实型变量，输入参数。自变量值，要求X>0。若X≤0，本子程序返回函数值－1，并在子程序中输出信息"ERR * * X<0!"。

函数名 MGAM1 返回双精度实型伽马函数值Γ(X)。

五、子程序(文件名:MGAM1.FOR)

```
      FUNCTION MGAM1(X)
      DOUBLE PRECISION MGAM1,X
      DOUBLE PRECISION Y,T,S,U,A(11)
      DATA A/0.0000677106,-0.0003442342,0.0015397681,
     *       -0.0024467480,0.0109736958,-0.0002109075,
     *       0.0742379071,0.0815782188,0.4118402518,
     *       0.4227843370,1.0/
      IF (X.LE.0.) THEN
        WRITE(*,*)'ERR * * X<0!'
        MGAM1=-1.0
        RETURN
      END IF
      Y=X
      IF (Y.LE.1.0) THEN
        T=1.0/(Y*(Y+1.0))
        Y=Y+2.0
      ELSE IF (Y.LE.2.0) THEN
        T=1.0/Y
        Y=Y+1.0
      ELSE IF (Y.LE.3.0) THEN
        T=1.0
      ELSE
        T=1.0
10      IF (Y.GT.3.0) THEN
          Y=Y-1.0
          T=T*Y
          GOTO 10
        END IF
      END IF
      S=A(1)
      U=Y-2.0
      DO 20 I=1,10
20    S=S*U+A(I+1)
      S=S*T
      MGAM1=S
      RETURN
```

```
      END
```

六、例

计算 0.5 到 5.0 之间,每隔 0.5 的伽马函数值与贝塔函数值 B(1.5,2.5)。

主程序(文件名:MGAM10.FOR)为

```
      DOUBLE PRECISION MGAM1,X,Y
      WRITE(*,*)
      DO 10 I=1,10
        X=0.5*I
        Y=MGAM1(X)
        WRITE(*,100) X,Y
10    CONTINUE
100   FORMAT(1X,'X=',F6.3,5X,'gamma(X)=',D13.6)
      Y=MGAM1(1.5D0)*MGAM1(2.5D0)/MGAM1(4.0D0)
      WRITE(*,200) Y
200   FORMAT(1X,'B(1.5,2.5)=',D13.6)
      END
```

运行结果为

```
X=  .500     gamma(X)=     .177245D+01
X= 1.000     gamma(X)=     .100001D+01
X= 1.500     gamma(X)=     .886227D+00
X= 2.000     gamma(X)=     .100001D+01
X= 2.500     gamma(X)=     .132934D+01
X= 3.000     gamma(X)=     .200002D+01
X= 3.500     gamma(X)=     .332335D+01
X= 4.000     gamma(X)=     .600006D+01
X= 4.500     gamma(X)=     .116317D+02
X= 5.000     gamma(X)=     .240002D+02
B(1.5,2.5)=  .196348D+00
```

12.2 不完全伽马函数

一、功能

计算不完全伽马函数(Incomplete Gamma function)的值。

二、方法说明

不完全伽马函数的定义为

$$\Gamma(\alpha,x)=\frac{P(\alpha,x)}{\Gamma(\alpha)}\quad,\quad \alpha>0,\ x>0 \tag{12.2.1}$$

其中

$$P(\alpha,x)=\int_0^x e^{-t}t^{\alpha-1}dt$$

对于不完全伽马函数有

$$\Gamma(a,0)=0\ ,\ \Gamma(a,\infty)=1$$

不完全伽马函数也可表示为

$$\Gamma(a,x)=1-\frac{Q(a,x)}{\Gamma(a)}\quad ,\quad a>0,\ x>0 \tag{12.2.2}$$

其中

$$Q(a,x)=\int_x^{\infty}e^{-t}t^{a-1}dt$$

而 $Q(a,x)/\Gamma(a)$ 称为余不完全伽马函数。

当 $x<a+1$ 时，用(12.2.1)式计算。其中 $P(a,x)$ 用如下级数计算

$$P(a,x)=e^{-x}x^{a}\sum_{k=1}^{\infty}\frac{\Gamma(a)}{\Gamma(a+1+k)}x^{k}$$

当 $x\geqslant a+1$ 时，用(12.2.2)式计算。其中 $Q(a,x)$ 用如下连分式计算：

$$Q(a,x)=e^{-x}x^{a}\varphi(a,x)$$

$$\varphi(a,x)=\cfrac{1}{x+\cfrac{1-a}{1+\cfrac{1}{x+\cfrac{2-a}{1+\cfrac{2}{x+\cdots+\cfrac{n-a}{1+\cfrac{n}{x+\cdots}}}}}}}$$

三、子程序语句

FUNCTION MGAM2(A,X)

四、形参说明

A——双精度实型变量，输入参数。自变量 a 的值。要求 A>0，当 A≤0 时，本子程序返回函数值 -1.0，且在子程序中输出信息"ERR * * A<=0!"。

X——双精度实型变量，输入参数。自变量 x 的值。要求 X>0，当 X≤0 时，本子程序返回函数值 -1.0，且在子程序中输出信息"ERR * * X<=0!"。

函数名 MGAM2 返回双精度实型不完全伽马函数值。

五、子程序（文件名：MGAM2.FOR）

```
      FUNCTION MGAM2(A,X)
      DOUBLE PRECISION MGAM2,A,X
      DOUBLE PRECISION MGAM1,P,Q,D,S,S1,P0,Q0,P1,Q1,QQ
      IF ((A.LE.0.0).OR.(X.LT.0.0)) THEN
        IF (A.LE.0.0) THEN
          WRITE(*,*)' ERR * * A<=0!'
        END IF
        IF (X.LT.0.0) THEN
          WRITE(*,*)' ERR * * X<0!'
        END IF
        MGAM2=-1.0
      END IF
      IF (X+1.0.EQ.1.0) THEN
```

```
      MGAM2=0.0
      RETURN
    END IF
    IF (X.GT.1.0D+35) THEN
      MGAM2=1.0
      RETURN
    END IF
    Q=LOG(X)
    Q=A*Q
    QQ=EXP(Q)
    IF (X.LT.1.0+A) THEN
      P=A
      D=1.0/A
      S=D
      DO 10 N=1,100
        P=1.0+P
        D=D*X/P
        S=S+D
        IF (ABS(D).LT.ABS(S)*1.0D-07) THEN
          S=S*EXP(-X)*QQ/MGAM1(A)
          MGAM2=S
          RETURN
        END IF
10    CONTINUE
    ELSE
      S=1.0/X
      P0=0.0
      P1=1.0
      Q0=1.0
      Q1=X
      DO 20 N=1,100
        P0=P1+(N-A)*P0
        Q0=Q1+(N-A)*Q0
        P=X*P0+N*P1
        Q=X*Q0+N*Q1
        IF (ABS(Q)+1.0.NE.1.0) THEN
          S1=P/Q
          P1=P
          Q1=Q
          IF (ABS((S1-S)/S1).LT.1.0D-07) THEN
            S=S1*EXP(-X)*QQ/MGAM1(A)
            MGAM2=1.0-S
```

```
          RETURN
        END IF
        S=S1
      END IF
      P1=P
      Q1=Q
20    CONTINUE
    END IF
    WRITE(*,*) ' A too large !'
    S=1.0-S*EXP(-X)*QQ/MGAM1(A)
    MGAM2=S
    RETURN
    END
```

六、例

计算当 $a=0.5,5.0,50.0$ 时，x 分别取 0.1，1.0，10.0 时的不完全伽马函数值 $\Gamma(a,x)$。

主程序(文件名:MGAM20.FOR)为

```
      DOUBLE PRECISION MGAM2,Y,S,T,A(3),X(3)
      DATA A/0.5,5.0,50.0/
      DATA X/0.1,1.0,10.0/
      WRITE(*,*)
      DO 10 I=0,2
      DO 10 J=0,2
        S=A(I+1)
        T=X(J+1)
        Y=MGAM2(S,T)
        WRITE(*,100) A(I+1),X(J+1),Y
10    CONTINUE
      WRITE(*,*)
100   FORMAT(1X,'gamma(',F4.1,' , ',F4.1,')=',D13.6)
      END
```

运行结果为

```
gamma(  .5 ,   .1)=    .345279D+00
gamma(  .5 ,  1.0)=    .842701D+00
gamma(  .5 , 10.0)=    .999992D+00
gamma( 5.0 ,   .1)=    .766773D-07
gamma( 5.0 ,  1.0)=    .365981D-02
gamma( 5.0 , 10.0)=    .970748D+00
gamma(50.0 ,   .1)=    .298088-114
gamma(50.0 ,  1.0)=    .123374D-64
gamma(50.0 , 10.0)=    .185471D-18
```

七、附注

本子程序需要调用计算伽马函数值的子程序 MGAM1(参看 12.1 节)。

12.3 误差函数

一、功能

计算实变量 x 的误差函数值

$$\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$$

二、方法说明

误差函数 erf(x)具有下列极限值与对称性:

$$\mathrm{erf}(0)=0,\mathrm{erf}(\infty)=1,\mathrm{erf}(-x)=-\mathrm{erf}(x)$$

当 $x\geqslant 0$ 时,可以用不完全伽马函数计算误差函数,即

$$\mathrm{erf}(x)=\Gamma(0.5,x^2)$$

利用误差函数可以计算余误差函数,即

$$\mathrm{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}dt=1-\mathrm{erf}(x)$$

也可以计算正态概率积分,即

$$\Phi(x)=\frac{2}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}dt=\frac{1}{2}+\frac{1}{2}\mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)$$

三、子程序语句

FUNCTION MERRF(X)

四、形参说明

X——双精度实型变量,输入参数。自变量值。

函数名 MERRF 返回双精度实型函数值 erf(X)。

五、子程序(文件名:MERRF.FOR)

```
FUNCTION MERRF(X)
DOUBLE PRECISION MERRF,X,MGAM2
IF (X.GE.0.0) THEN
  MERRF=MGAM2(0.5D0,X*X)
ELSE
  MERRF=-MGAM2(0.5D0,X*X)
END IF
RETURN
END
```

六、例

计算 0 到 2 之间、间隔为 0.05 的误差函数值。

主程序(文件名:MERRF0.FOR)为

```
      DOUBLE PRECISION MERRF,X(4),Y(4)
```

```
      WRITE(*,*)
      DO 10 I=0,9
        DO 20 J=0,3
          X(J+1)=0.05*(4.0*I+J)
          Y(J+1)=MERRF(X(J+1))
20      CONTINUE
        WRITE(*,100) (X(K),Y(K),K=1,4)
10    CONTINUE
100   FORMAT(1X,4('erf(',F4.2,')=',F8.6,1X))
      X(1)=2.0
      Y(1)=MERRF(X(1))
      WRITE(*,200) X(1),Y(1)
200   FORMAT(1X,'erf(',F4.2,')=',F8.6)
      WRITE(*,*)
      END
```

运行结果为

```
erf( .00)=  .000000 erf( .05)=  .056372 erf( .10)=  .112463 erf( .15)=  .167996
erf( .20)=  .222703 erf( .25)=  .276326 erf( .30)=  .328627 erf( .35)=  .379382
erf( .40)=  .428392 erf( .45)=  .475482 erf( .50)=  .520500 erf( .55)=  .563323
erf( .60)=  .603856 erf( .65)=  .642029 erf( .70)=  .677801 erf( .75)=  .711156
erf( .80)=  .742101 erf( .85)=  .770668 erf( .90)=  .796908 erf( .95)=  .820891
erf(1.00)=  .842701 erf(1.05)=  .862436 erf(1.10)=  .880205 erf(1.15)=  .896124
erf(1.20)=  .910314 erf(1.25)=  .922900 erf(1.30)=  .934008 erf(1.35)=  .943762
erf(1.40)=  .952285 erf(1.45)=  .959695 erf(1.50)=  .966105 erf(1.55)=  .971623
erf(1.60)=  .976348 erf(1.65)=  .980376 erf(1.70)=  .983790 erf(1.75)=  .986672
erf(1.80)=  .989091 erf(1.85)=  .991111 erf(1.90)=  .992790 erf(1.95)=  .994179
erf(2.00)=  .995322
```

七、附注

本子程序需要调用计算不完全伽马函数值的子程序 MGAM2(参看 12.2 节)，而不完全伽马函数子程序 MGAM2 又要调用伽马函数子程序 MGAM1(参看 12.1 节)。

12.4 第一类整数阶贝塞耳函数

一、功能

计算实变量 x 的第一类整数阶贝塞耳(Bessel)函数值 $J_n(x)$。

二、方法说明

第一类贝塞耳函数的定义为

$$J_n(x)=\left(\frac{x}{2}\right)^n\sum_{k=0}^{\infty}\frac{(-1)^k}{k!\ (n+k)!}\cdot\left(\frac{x}{2}\right)^{2k}$$

其中 n 为非负整数。其积分表达式为

$$J_n(x)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\cos(nt-x\sin t)\mathrm{d}t$$

第一类整数阶贝塞耳函数具有下列递推关系

$$J_{n+1}(x)=\frac{2n}{x}J_n(x)-J_{n-1}(x) \tag{12.4.1}$$

$J_0(x)$与$J_1(x)$的计算公式如下：

当$|x|<8.0$时，$J_0(x)=A(y)/B(y)$，其中$y=x^2$，且

$$A(y)=a_0+a_1y+a_2y^2+a_3y^3+a_4y^4+a_5y^5$$

$$B(y)=b_0+b_1y+b_2y^2+b_3y^3+b_4y^4+b_5y^5$$

其系数分别为

$a_0=5.7568490574\times10^{10}$，　$a_1=-1.3362590354\times10^{10}$

$a_2=6.516196407\times10^{8}$，　$a_3=-1.121442418\times10^{7}$

$a_4=7.739233017\times10^{4}$，　$a_5=-1.849052456\times10^{2}$

$b_0=5.7568490411\times10^{10}$，　$b_1=1.029532985\times10^{9}$

$b_2=9.494680718\times10^{6}$，　$b_3=5.927264853\times10^{4}$

$b_4=2.678532712\times10^{2}$，　$b_5=1.0$

$$J_1(x)=x\frac{C(y)}{D(y)}$$

其中　$y=x^2$，且

$$C(y)=c_0+c_1y+c_2y^2+c_3y^3+c_4y^4+c_5y^5$$

$$D(y)=d_0+d_1y+d_2y^2+d_3y^3+d_4y^4+d_5y^5$$

其系数分别为

$c_0=7.2362614232\times10^{10}$，　$c_1=-7.895059235\times10^{9}$

$c_2=2.423968531\times10^{8}$，　$c_3=-2.972611439\times10^{6}$

$c_4=1.570448260\times10^{4}$　$c_5=-3.016036606\times10$

$d_0=1.44725228443\times10^{11}$，　$d_1=2.300535178\times10^{9}$

$d_2=1.858330474\times10^{7}$，　$d_3=9.944743394\times10^{4}$

$d_4=3.769991397\times10^{2}$，　$d_5=1.0$

当$|x|\geq8.0$时，令$z=8.0/|x|$，$y=z^2$，则

$$J_0(x)=\sqrt{\frac{2}{\pi|x|}}[E(y)\cos\theta-zF(y)\sin\theta]$$

其中　$\theta=|x|-\frac{\pi}{4}$，且

$$E(y)=e_0+e_1y+e_2y^2+e_3y^3+e_4y^4$$

$$F(y)=f_0+f_1y+f_2y^2+f_3y^3+f_4y^4$$

其系数分别为

$e_0=1.0$，　$e_1=-0.1098628627\times10^{-2}$

$e_2=0.2734510407\times10^{-4}$，　$e_3=-0.2073370639\times10^{-5}$

$e_4=0.2093887211\times10^{-6}$

$f_0=-0.1562499995\times10^{-1}$，　　$f_1=0.1430488765\times10^{-3}$

$f_2=-0.6911147651\times10^{-5}$，　　$f_3=0.7621095161\times10^{-6}$

$f_4=-0.934935152\times10^{-7}$

而且

$$J_1(x)=\sqrt{\frac{2}{\pi|x|}}[G(y)\cos\theta-zH(y)\sin\theta]\;;\;x>0$$

$$J_1(-x)=-J_1(x)$$

其中　$\theta=|x|-\frac{3\pi}{4}$，且

$$G(y)=g_0+g_1y+g_2y^2+g_3y^3+g_4y^4$$

$$H(y)=h_0+h_1y+h_2y^2+h_3y^3+h_4y^4$$

其系数分别为

$g_0=1.0$，　　$g_1=0.183105\times10^{-2}$

$g_2=-0.3516396496\times10^{-4}$，　　$g_3=0.2457520174\times10^{-5}$

$g_4=-0.240337019\times10^{-6}$

$h_0=0.4687499995\times10^{-1}$，　　$h_1=-0.2002690873\times10^{-3}$

$h_2=0.8449199096\times10^{-5}$，　　$h_3=-0.88228987\times10^{-6}$

$h_4=0.105787412\times10^{-6}$

当 $n\geqslant2$ 时，如果 $|x|>n$，则可用递推公式(12.4.1)进行递推计算；否则用下列递推关系计算：

$$J_{n-1}(x)=\frac{2n}{x}J_n(x)-J_{n+1}(x) \qquad (12.4.2)$$

三、子程序语句

FUNCTION MBSL1(N,X)

四、形参说明

N——整型变量，输入参数。第一类贝塞耳函数的阶数，要求 N≥0。当 N<0 时，本子程序中按|N|处理。

X——双精度实型变量，输入参数。自变量值。

函数名 MBSL1 返回双精度实型函数值 $J_N(X)$

五、子程序(文件名：MBSL1.FOR)

```
      FUNCTION MBSL1(N,X)
      DOUBLE PRECISION MBSL1,X
      DOUBLE PRECISION T,Y,Z,P,Q,S,B0,B1
      DOUBLE PRECISION A(6),B(6),C(6),D(6),E(5),F(5),G(5),H(5)
      DATA A/57568490574.0,-13362590354.0,651619640.7,
     *          -11214424.18,77392.33017,-184.9052456/
      DATA B/57568490411.0,1029532985.0,9494680.718,
     *          59272.64853,267.8532712,1.0/
```

```
      DATA C/72362614232.0,-7895059235.0,242396853.1,
     *          -2972611.439,15704.4826,-30.16036606/
      DATA D/144725228443.0,2300535178.0,18583304.74,
     *          99447.43394,376.9991397,1.0/
      DATA E/1.0,-0.1098628627D-02,0.2734510407D-04,
     *          -0.2073370639D-05,0.2093887211D-06/
      DATA F/-0.1562499995D-01,0.1430488765D-03,-0.6911147651D-05,
     *          0.7621095161D-06,-0.934935152D-07/
      DATA G/1.0,0.183105D-02,-0.3516396496D-04,
     *          0.2457520174D-05,-0.240337019D-06/
      DATA H/0.4687499995D-01,-0.2002690873D-03,0.8449199096D-05,
     *          -0.88228987D-06,0.105787412D-06/
      T=ABS(X)
      IF (N.LT.0) N=-N
      IF (N.NE.1) THEN
        IF (T.LT.8.0) THEN
          Y=T*T
          P=A(6)
          Q=B(6)
            DO 10 I=5,1,-1
              P=P*Y+A(I)
              Q=Q*Y+B(I)
10          CONTINUE
            P=P/Q
        ELSE
          Z=8.0/T
          Y=Z*Z
          P=E(5)
          Q=F(5)
          DO 20 I=4,1,-1
            P=P*Y+E(I)
            Q=Q*Y+F(I)
20        CONTINUE
          S=T-0.785398164
          P=P*COS(S)-Z*Q*SIN(S)
          P=P*SQRT(0.636619772/T)
          END IF
      END IF
      IF (N.EQ.0) THEN
        MBSL1=P
        RETURN
      END IF
```

```
      B0=P
      IF (T.LT.8.0) THEN
        Y=T*T
        P=C(6)
        Q=D(6)
        DO 30 I=5,1,-1
          P=P*Y+C(I)
          Q=Q*Y+D(I)
30      CONTINUE
        P=X*P/Q
      ELSE
        Z=8.0/T
        Y=Z*Z
        P=G(5)
        Q=H(5)
        DO 40 I=4,1,-1
          P=P*Y+G(I)
          Q=Q*Y+H(I)
40      CONTINUE
        S=T-2.356194491
        P=P*COS(S)-Z*Q*SIN(S)
        P=P*X*SQRT(0.636619772/T)/T
      END IF
      IF (N.EQ.1) THEN
        MBSL1=P
        RETURN
      END IF
      B1=P
      IF (X.EQ.0.0) THEN
        MBSL1=0.0
        RETURN
      END IF
      S=2.0/T
      IF (T.GT.1.0*N) THEN
        IF (X.LT.0.0) B1=-B1
        DO 50 I=1,N-1
          P=S*I*B1-B0
          B0=B1
          B1=P
50      CONTINUE
      ELSE
        M=SQRT(40.0*N)
```

```
      M=(N+M)/2
      M=M+M
      P=0.0
      Q=0.0
      B0=1.0
      B1=0.0
      DO 60 I=M-1,0,-1
        T=S*(I+1)*B0-B1
        B1=B0
        B0=T
        IF (ABS(B0).GT.1.0D+10) THEN
          B0=B0*1.0D-10
          B1=B1*1.0D-10
          P=P*1.0D-10
        Q=Q*1.0D-10
        END IF
        IF (MOD(I+2,2).EQ.0) Q=Q+B0
        IF (I+1.EQ.N) P=B1
60      CONTINUE
      Q=2.0*Q-B0
      P=P/Q
    END IF
    IF ((X.LT.0.0).AND.(MOD(N,2).EQ.1)) P=-P
    MBSL1=P
    RETURN
    END
```

六、例

计算阶数 $n=0,1,2,3,4,5$ 时，自变量 x 的值分别取 0.05，0.5，5.0，50.0 时的第一类贝塞耳函数值 $J_n(x)$。

主程序(文件名：MBSL10.FOR)为

```
      DOUBLE PRECISION X,Y,MBSL1
      WRITE(*,*)
      DO 10 N=0,5
        X=0.05
        DO 20 I=1,4
          Y=MBSL1(N,X)
          WRITE(*,100) N,X,Y
          X=X*10.0
20      CONTINUE
10    CONTINUE
100   FORMAT(1X,'N=',I2,2X,'X=',F7.3,3X,'J(n,x)=',D13.6)
```

```
      WRITE(*,*)
      END
```

运行结果为

```
N= 0   X=   .050    J(n,x)=   .999375D+00
N= 0   X=   .500    J(n,x)=   .938470D+00
N= 0   X=  5.000    J(n,x)= -.177597D+00
N= 0   X= 50.000    J(n,x)=   .558124D-01
N= 1   X=   .050    J(n,x)=   .249922D-01
N= 1   X=   .500    J(n,x)=   .242268D+00
N= 1   X=  5.000    J(n,x)= -.327579D+00
N= 1   X= 50.000    J(n,x)= -.975118D-01
N= 2   X=   .050    J(n,x)=   .312435D-03
N= 2   X=   .500    J(n,x)=   .306040D-01
N= 2   X=  5.000    J(n,x)=   .465651D-01
N= 2   X= 50.000    J(n,x)= -.597129D-01
N= 3   X=   .050    J(n,x)=   .260376D-05
N= 3   X=   .500    J(n,x)=   .256373D-02
N= 3   X=  5.000    J(n,x)=   .364831D+00
N= 3   X= 50.000    J(n,x)=   .927348D-01
N= 4   X=   .050    J(n,x)=   .162740D-07
N= 4   X=   .500    J(n,x)=   .160736D-03
N= 4   X=  5.000    J(n,x)=   .391233D+00
N= 4   X= 50.000    J(n,x)=   .708410D-01
N= 5   X=   .050    J(n,x)=   .813717D-10
N= 5   X=   .500    J(n,x)=   .805363D-05
N= 5   X=  5.000    J(n,x)=   .261141D+00
N= 5   X= 50.000    J(n,x)= -.814002D-01
```

12.5 第二类整数阶贝塞耳函数

一、功能

计算实变量 x 的第二类整数阶贝塞耳(Bessel)函数值 $Y_n(x)$。

二、方法说明

第二类整数阶贝塞耳函数具有如下递推关系

$$Y_{n+1}(x)=\frac{2n}{x}Y_n(x)-Y_{n-1}(x)\ ,\ x\geqslant 0$$

这个递推公式是稳定的。

当 $x<8.0$ 时，

$$Y_0(x)=\frac{A(y)}{B(y)}+\frac{2}{\pi}J_0(x)\ln x$$

其中　$y=x^2$，且

$$A(y)=a_0+a_1y+a_2y^2+a_3y^3+a_4y^4+a_5y^5$$

$$B(y)=b_0+b_1y+b_2y^2+b_3y^3+b_4y^4+b_5y^5$$

其系数分别为

$a_0=-2.957821389\times10^{9}$，　　$a_1=7.062834065\times10^{9}$

$a_2=-5.123598036\times10^{8}$，　　$a_3=1.087988129\times10^{7}$

$a_4=-8.632792757\times10^{4}$，　　$a_5=2.284622733\times10^{2}$

$b_0=4.0076544269\times10^{10}$，　　$b_1=7.452499648\times10^{8}$

$b_2=7.189466438\times10^{6}$，　　$b_3=4.744726470\times10^{4}$

$b_4=2.261030244\times10^{2}$，　　$b_5=1.0$

而

$$\mathrm{Y}_1(x)=x\frac{C(y)}{D(y)}+\frac{2}{\pi}\left[\mathrm{J}_1(x)\ln x-\frac{1}{x}\right]$$

其中 $y=x^2$，且

$$C(y)=c_0+c_1y+c_2y^2+c_3y^3+c_4y^4+c_5y^5$$

$$D(y)=d_0+d_1y+d_2y^2+d_3y^3+d_4y^4+d_5y^5+d_6y^6$$

其系数分别为

$c_0=-4.900604943\times10^{12}$，　　$c_1=1.275274390\times10^{12}$

$c_2=-5.153438139\times10^{10}$，　　$c_3=7.349264551\times10^{8}$

$c_4=-4.237922726\times10^{6}$，　　$c_5=8.511937935\times10^{3}$

$d_0=2.499580570\times10^{13}$，　　$d_1=4.244419664\times10^{11}$

$d_2=3.733650367\times10^{9}$，　　$d_3=2.245904002\times10^{7}$

$d_4=1.020426050\times10^{5}$，　　$d_5=3.549632885\times10^{2}$

$d_6=1.0$

当 $x\geqslant8.0$ 时，令 $z=8.0/x$，$y=z^2$，则

$$\mathrm{Y}_1(x)=\sqrt{\frac{2}{\pi x}}\left[E(y)\sin\theta+zF(y)\cos\theta\right]$$

其中　$\theta=x-\frac{\pi}{4}$，且

$$E(y)=e_0+e_1y+e_2y^2+e_3y^3+e_4y^4$$

$$F(y)=f_0+f_1y+f_2y^2+f_3y^3+f_4y^4$$

其系数分别为

$e_0=1.0$，　　$e_1=-0.1098628627\times10^{-2}$

$e_2=0.2734510407\times10^{-4}$，　　$e_3=-0.2073370639\times10^{-5}$

$e_4=0.2093887211\times10^{-6}$，

$f_0=-0.1562499995\times10^{-1}$，　　$f_1=0.1430488765\times10^{-3}$

$f_2=-0.6911147651\times10^{-5}$，　　$f_3=0.7621095161\times10^{-6}$

$f_4=-0.934945152\times10^{-7}$

而且

$$Y_1(x)=\sqrt{\frac{2}{\pi x}}[G(y)\sin\theta+yH(y)\cos\theta]$$

其中 $\theta=x-\frac{3\pi}{4}$，且

$$G(y)=g_0+g_1y+g_2y^2+g_3y^3+g_4y^4$$
$$H(y)=h_0+h_1y+h_2y^2+h_3y^3+h_4y^4$$

其系数分别为

$g_0=1.0$，　　$g_1=0.183105\times10^{-2}$

$g_2=-0.3516396496\times10^{-1}$，　　$g_3=0.2457520174\times10^{-5}$

$g_4=-0.240337019\times10^{-5}$，

$h_0=0.4687499995\times10^{-1}$，　　$h_1=-0.2002690873\times10^{-3}$

$h_2=0.8449199096\times10^{-5}$，　　$h_3=-0.88228987\times10^{-6}$

$h_4=0.105787412\times10^{-4}$

三、子程序语句

FUNCTION MBSL2(N,X)

四、形参说明

N——整型变量，输入参数。第二类贝塞耳函数的阶数，要求 N≥0。当 N<0 时，本子程序按|N|处理。

X——双精度实型变量，输入参数，自变量值，要求 X≥0，当 X<0 时，本子程序按|X|处理。

函数名 MBSL2 返回双精度实型函数值 $Y_n(X)$。

五、子程序（文件名：MBSL2.FOR）

```
      FUNCTION MBSL2(N,X)
      DOUBLE PRECISION MBSL2,X,MBSL1
      DOUBLE PRECISION Y,Z,P,Q,S,B0,B1
      DOUBLE PRECISION A(6),B(6),C(6),D(7),E(5),F(5),G(5),H(5)
      DATA A/-2.957821389D+09,7.062834065D+09,-5.123598036D+08,
     *       1.087988129D+07,-8.632792757D+04,2.284622733D+02/
      DATA B/4.0076544269D+10,7.452499648D+08,7.189466438D+06,
     *       4.74472647D+04,2.261030244D+02,1.0/
      DATA C/4.900604943D+12,1.27527439D+12,-5.153438139D+10,
     *       7.349264551D+08,-4.237922726D+06,8.511937935D+03/
      DATA D/2.49958057D+13,4.244419664D+11,3.733650367D+09,
     *       2.245904002D+07,1.02042605D+05,3.549632885D+02,1.0/
      DATA E/1.0,-0.1098628627D-02,0.2734510407D-04,
     *       -0.2073370639D-05,0.2093887211D-06/
      DATA F/-0.1562499995D-01,0.1430488765D-03,-0.6911147651D-05,
     *       0.7621095161D-06,-0.934935152D-07/
```

```
      DATA G/1.0,0.183105D-02,-0.3516396496D-04,
     *       0.2457520174D-05,-0.240337019D-06/
      DATA H/0.4687499995D-01,-0.2002690873D-03,0.8449199096D-05,
     *       -0.88228987D-06,0.105787412D-06/
      IF (N.LT.0) N=-N
      IF (X.LT.0.0) X=-X
      IF (X+1.0.EQ.1.0) THEN
        MBSL2=-1.0D+35
        RETURN
      END IF
      IF (N.NE.1) THEN
        IF (X.LT.8.0) THEN
          Y=X*X
          P=A(6)
          Q=B(6)
          DO 10 I=5,1,-1
            P=P*Y+A(I)
            Q=Q*Y+B(I)
10        CONTINUE
          P=P/Q+0.636619772*MBSL1(0,X)*LOG(X)
        ELSE
          Z=8.0/X
          Y=Z*Z
          P=E(5)
          Q=F(5)
          DO 20 I=4,1,-1
          P=P*Y+E(I)
            Q=Q*Y+F(I)
20        CONTINUE
          S=X-0.785398164
          P=P*SIN(S)+Z*Q*COS(S)
          P=P*SQRT(0.636619772/X)
        END IF
      END IF
      IF (N.EQ.0) THEN
        MBSL2=P
        RETURN
      END IF
      B0=P
      IF (X.LT.8.0) THEN
        Y=X*X
        P=C(6)
```

```
            Q=D(7)
            DO 30 I=5,1,-1
              P=P*Y+C(I)
              Q=Q*Y+D(I+1)
30          CONTINUE
            Q=Q*Y+D(1)
            P=X*P/Q+0.636619772*(MBSL1(1,X)*LOG(X)-1.0/X)
          ELSE
            Z=8.0/X
            Y=Z*Z
            P=G(5)
            Q=H(5)
            DO 40 I=4,1,-1
              P=P*Y+G(I)
              Q=Q*Y+H(I)
40          CONTINUE
            S=X-2.356194491
            P=P*SIN(S)+Z*Q*COS(S)
            P=P*SQRT(0.636619772/X)
          END IF
          IF (N.EQ.1) THEN
            MBSL2=P
            RETURN
          END IF
          B1=P
          S=2.0/X
          DO 50 I=1,N-1
            P=S*I*B1-B0
            B0=B1
            B1=P
50        CONTINUE
          MBSL2=P
          RETURN
          END
```

六、例

计算阶数 $n=0,1,2,3,4,5$ 时,自变量 x 的值分别取 0.05,0.5,5.0,50.0 时的第二类贝塞耳函数值 $Y_n(x)$。

主程序(文件名:MBSL20.FOR)为

```
        DOUBLE PRECISION X,Y,MBSL2
        WRITE(*,*)
        DO 10 N=0,5
```

```
        X=0.05
        DO 20 I=1,4
          Y=MBSL2(N,X)
          WRITE(*,100) N,X,Y
          X=X*10.0
20      CONTINUE
10    CONTINUE
100   FORMAT(1X,'N=',I2,2X,'X=',F7.3,3X,'Y(n,x)=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
N=  0   X=    .050   Y(n,x)=  -.197931D+01
N=  0   X=    .500   Y(n,x)=  -.444519D+00
N=  0   X=   5.000   Y(n,x)=  -.308518D+00
N=  0   X=  50.000   Y(n,x)=  -.980650D-01
N=  1   X=    .050   Y(n,x)=  -.127899D+02
N=  1   X=    .500   Y(n,x)=  -.147147D+01
N=  1   X=   5.000   Y(n,x)=   .147863D+00
N=  1   X=  50.000   Y(n,x)=  -.567957D-01
N=  2   X=    .050   Y(n,x)=  -.509615D+03
N=  2   X=    .500   Y(n,x)=  -.544137D+01
N=  2   X=   5.000   Y(n,x)=   .367663D+00
N=  2   X=  50.000   Y(n,x)=   .957931D-01
N=  3   X=    .050   Y(n,x)=  -.407564D+05
N=  3   X=    .500   Y(n,x)=  -.420595D+02
N=  3   X=   5.000   Y(n,x)=   .146267D+00
N=  3   X=  50.000   Y(n,x)=   .644592D-01
N=  4   X=    .050   Y(n,x)=  -.489026D+07
N=  4   X=    .500   Y(n,x)=  -.499273D+03
N=  4   X=   5.000   Y(n,x)=  -.192142D+00
N=  4   X=  50.000   Y(n,x)=  -.880580D-01
N=  5   X=    .050   Y(n,x)=  -.782401D+09
N=  5   X=    .500   Y(n,x)=  -.794630D+04
N=  5   X=   5.000   Y(n,x)=  -.453695D+00
N=  5   X=  50.000   Y(n,x)=  -.785485D-01
```

七、附注

本子程序需要调用计算第一类贝塞耳函数值 $J_0(x)$ 与 $J_1(x)$ 的子程序 MBSL1(参看12.4节)。

12.6 变型第一类整数阶贝塞耳函数

一、功能

计算实变量 x 的变型第一类整数阶贝塞耳(Bessel)函数值 $I_n(x)$。

二、方法说明

变型第一类整数阶贝塞耳函数表示为

$$I_n(x)=(-j)^n J_n(jx)$$

其中 n 为非负整数；$j=\sqrt{-1}$；$J_n(jx)$为纯虚变量(jx)的第一类贝塞耳函数。

$I_0(x)$与$I_1(x)$的计算公式如下：

当$|x|<3.75$时，令 $y=(x/3.75)^2$，则

$$I_0(x)=a_0+a_1y+a_2y^2+a_3y^3+a_4y^4+a_5y^5+a_6y^6$$

$$I_1(x)=x(b_0+b_1y+b_2y^2+b_3y^3+b_4y^4+b_5y^5+b_6y^6)$$

其系数分别为

$a_0=1.0$， $a_1=3.5156229$

$a_2=3.0899424$， $a_3=1.2067492$

$a_4=2.659732\times10^{-1}$， $a_5=3.60768\times10^{-2}$

$a_6=4.5813\times10^{-3}$

$b_0=5.0\times10^{-1}$， $b_1=8.7890594\times10^{-1}$

$b_2=5.1498869\times10^{-1}$， $b_3=1.5084934\times10^{-1}$

$b_4=2.658773\times10^{-2}$， $b_5=3.01532\times10^{-3}$

$b_6=3.2411\times10^{-4}$

当$|x|\geqslant3.75$时，令 $y=3.75/|x|$，则

$$I_0(x)=\frac{e^{|x|}}{\sqrt{|x|}}C(y)$$

$$I_1(|x|)=\frac{e^{|x|}}{\sqrt{|x|}}D(y),\ I_1(-|x|)=-I_1(|x|)$$

其中

$$C(y)=c_0+c_1y+c_2y^2+c_3y^3+\cdots+c_8y^8$$

$$D(y)=d_0+d_1y+d_2y^2+d_3y^3+\cdots+d_8y^8$$

其系数分别为

$c_0=3.9894228\times10^{-1}$， $c_1=1.328592\times10^{-2}$

$c_2=2.25319\times10^{-3}$， $c_3=-1.57565\times10^{-3}$

$c_4=9.16281\times10^{-3}$， $c_5=-2.057706\times10^{-2}$

$c_6=2.635537\times10^{-2}$， $c_7=-1.647633\times10^{-2}$

$c_8=3.92377\times10^{-3}$

$d_0=3.9894228\times10^{-1}$， $d_1=-3.988024\times10^{-2}$

$d_2=-3.62018\times10^{-3}$,　　$d_3=1.63801\times10^{-3}$

$d_4=-1.031555\times10^{-2}$,　　$d_5=2.282967\times10^{-2}$

$d_6=-2.895312\times10^{-2}$,　　$d_7=1.787654\times10^{-2}$

$d_8=-4.20059\times10^{-3}$

当 $n\geqslant2$ 时，变型第一类贝塞耳函数具有下列递推关系：

$$I_{n+1}(x)=-\frac{2n}{x}I_n(x)+I_{n-1}(x)$$

但这个递推公式是不稳定的。实际计算时用如下递推关系：

$$I_{n-1}(x)=\frac{2n}{x}I_n(x)+I_{n+1}(x)$$

三、子程序语句

FUNCTION MBSL3(N,X)

四、形参说明

N——整型变量，输入参数。变型第一类贝塞耳函数的阶数，要求 N≥0。当 N<0 时，本子程序按|N|处理。

X——双精度实型变量，输入参数。自变量值。

函数名 MBSL3 返回双精度实型函数值 $I_N(X)$。

五、子程序(文件名：MBSL3.FOR)

```
      FUNCTION MBSL3(N,X)
      DOUBLE PRECISION MBSL3,X
      DOUBLE PRECISION T,Y,P,B0,B1,Q,A(7),B(7),C(9),D(9)
      DATA A/1.0,3.5156229,3.0899424,1.2067492,
     *          0.2659732,0.0360768,0.0045813/
      DATA B/0.5,0.87890594,0.51498869,0.15084934,
     *          0.02658773,0.00301532,0.00032411/
      DATA C/0.39894228,0.01328592,0.00225319,
     *          -0.00157565,0.00916281,-0.02057706,
     *          0.02635537,-0.01647663,0.00392377/
      DATA D/0.39894228,-0.03988024,-0.00362018,
     *          0.00163801,-0.01031555,0.02282967,
     *          -0.02895312,0.01787654,-0.00420059/
      IF (N.LT.0) N=-N
      T=ABS(X)
      IF (N.NE.1) THEN
        IF (T.LT.3.75) THEN
          Y=(X/3.75)*(X/3.75)
          P=A(7)
          DO 10 I=6,1,-1
10        P=P*Y+A(I)
        ELSE
```

```
        Y=3.75/T
        P=C(9)
        DO 20 I=8,1,-1
20      P=P*Y+C(I)
        P=P*EXP(T)/SQRT(T)
      END IF
    END IF
    IF (N.EQ.0) THEN
      MBSL3=P
      RETURN
    END IF
    Q=P
    IF (T.LT.3.75) THEN
      Y=(X/3.75)*(X/3.75)
      P=B(7)
      DO 30 I=6,1,-1
30    P=P*Y+B(I)
      P=P*T
    ELSE
      Y=3.75/T
      P=D(9)
      DO 40 I=8,1,-1
40    P=P*Y+D(I)
      P=P*EXP(T)/SQRT(T)
    END IF
    IF (X.LT.0.0) P=-P
    IF (N.EQ.1) THEN
      MBSL3=P
      RETURN
    END IF
    IF (X+1.0.EQ.1.0) THEN
      MBSL3=0.0
      RETURN
    END IF
    Y=2.0/T
    T=0.0
    B1=1.0
    B0=0.0
    M=SQRT(40.0*N)
    M=N+M
    M=M+M
    DO 50 I=M,1,-1
```

```
        P=B0+I*Y*B1
        B0=B1
        B1=P
        IF (ABS(B1).GT.1.0D+10) THEN
          T=T*1.0D-10
          B0=B0*1.0D-10
          B1=B1*1.0D-10
        END IF
        IF (I.EQ.N) T=B0
50    CONTINUE
      P=T*Q/B1
      IF ((X.LT.0.0).AND.(MOD(N,2).EQ.1)) P=-P
      MBSL3=P
      RETURN
      END
```

六、例

计算阶数 $n=0,1,2,3,4,5$ 时，自变量 x 的值分别取 0.05，0.5，5.0，50.0 时变型第一类贝塞耳函数值 $I_n(x)$。

主程序（文件名：MBSL30.FOR）为

```
      DOUBLE PRECISION X,Y,MBSL3
      WRITE(*,*)
      DO 10 N=0,5
        X=0.05
        DO 20 I=1,4
          Y=MBSL3(N,X)
          WRITE(*,100) N,X,Y
          X=X*10.0
20      CONTINUE
10    CONTINUE
100   FORMAT(1X,'N=',I2,2X,'X=',F7.3,3X,'I(n,x)=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
N=  0   X=    .050   I(n,x)=   .100063D+01
N=  0   X=    .500   I(n,x)=   .106348D+01
N=  0   X=   5.000   I(n,x)=   .272399D+02
N=  0   X=  50.000   I(n,x)=   .293256D+21
N=  1   X=    .050   I(n,x)=   .250078D-01
N=  1   X=    .500   I(n,x)=   .257894D+00
N=  1   X=   5.000   I(n,x)=   .243356D+02
```

N=	1	X=	50.000	I(n,x)=	.290308D+21
N=	2	X=	.050	I(n,x)=	.312565D−03
N=	2	X=	.500	I(n,x)=	.319061D−01
N=	2	X=	5.000	I(n,x)=	.175056D+02
N=	2	X=	50.000	I(n,x)=	.281647D+21
N=	3	X=	.050	I(n,x)=	.260457D−05
N=	3	X=	.500	I(n,x)=	.264511D−02
N=	3	X=	5.000	I(n,x)=	.103312D+02
N=	3	X=	50.000	I(n,x)=	.267776D+21
N=	4	X=	.050	I(n,x)=	.162781D−07
N=	4	X=	.500	I(n,x)=	.164806D−03
N=	4	X=	5.000	I(n,x)=	.510823D+01
N=	4	X=	50.000	I(n,x)=	.249510D+21
N=	5	X=	.050	I(n,x)=	.813887D−10
N=	5	X=	.500	I(n,x)=	.822317D−05
N=	5	X=	5.000	I(n,x)=	.215797D+01
N=	5	X=	50.000	I(n,x)=	.227855D+21

12.7 变型第二类整数阶贝塞耳函数

一、功能

计算实变量 x 的变型第二类整数阶贝塞耳(Bessel)函数值 $K_n(x)$。

二、方法说明

变型第二类整数阶贝塞耳函数表示为

$$K_n(x)=\frac{\pi}{2}j^{n+1}[J_n(jx)+jY_n(jx)],x>0$$

其中 n 为非负整数,$j=\sqrt{-1}$,$J_n(jx)$ 与 $Y_n(jx)$ 分别为纯虚变量 (jx) 的第一类与第二类贝塞耳函数。

$K_0(x)$ 与 $K_1(x)$ 的计算公式如下:

当 $x\leqslant 2.0$ 时,令 $y=x^2/4.0$,则

$$K_0(x)=A(y)-I_0(x)\ln\left(\frac{x}{2}\right)$$

$$K_1(x)=\frac{1}{x}B(y)+I_1(x)\ln\left(\frac{x}{2}\right)$$

其中 $I_0(x)$,$I_1(x)$ 分别为变型第一类零阶、一阶贝塞耳函数,而

$$A(y)=a_0+a_1y+a_2y^2+a_3y^3+a_4y^4+a_5y^5+a_6y^6$$
$$B(y)=b_0+b_1y+b_2y^2+b_3y^3+b_4y^4+b_5y^5+b_6y^6$$

其系数分别为

$a_0=-0.57721566$, $a_1=0.42278420$

$a_2=0.23069756$, $a_3=0.3488590\times 10^{-1}$

$a_4=0.262698\times10^{-2}$，　　$a_5=0.10750\times10^{-3}$

$a_6=0.74\times10^{-5}$

$b_0=1.0$，　　$b_1=0.15443144$

$b_2=-0.67278579$，　　$b_3=-0.18156897$

$b_4=-0.1919402\times10^{-1}$，　　$b_5=-0.110404\times10^{-2}$

$b_6=-0.4686\times10^{-4}$

当 $x>2.0$ 时，令 $y=2.0/x$，则

$$\mathrm{K}_0(x)=\frac{e^{-x}}{\sqrt{x}}C(y)$$

$$\mathrm{K}_1(x)=\frac{e^{-x}}{\sqrt{x}}D(y)$$

其中

$$C(y)=c_0+c_1y+c_2y^2+c_3y^3+c_4y^4+c_5y^5+c_6y^6$$
$$D(y)=d_0+d_1y+d_2y^2+d_3y^3+d_4y^4+d_5y^5+d_6y^6$$

其系数分别为

$c_0=0.125331414\times10^{1}$，　　$c_1=-0.7832358\times10^{-1}$

$c_2=0.2189568\times10^{-1}$，　　$c_3=-0.1062446\times10^{-1}$

$c_4=0.587872\times10^{-2}$，　　$c_5=-0.251540\times10^{-2}$

$c_6=0.53208\times10^{-3}$

$d_0=0.125331414\times10^{1}$，　　$d_1=0.23498619$

$d_2=-0.3655620\times10^{-1}$，　　$d_3=0.1504268\times10^{-1}$

$d_4=-0.780353\times10^{-2}$，　　$d_5=0.325614\times10^{-2}$

$d_6=-0.68245\times10^{-3}$

当 $n\geqslant2$ 时，用下列递推关系进行计算：

$$\mathrm{K}_{n+1}(x)=\frac{2n}{x}\mathrm{K}_n(x)+\mathrm{K}_{n-1}(x)$$

三、子程序语句

FUNCTION MBSL4(N,X)

四、形参说明

N——整型变量，输入参数。变型第二类贝塞耳函数的阶数，要求 N≥0。当 N<0 时，本子程序按|N|处理。

X——双精度实型变量，输入参数。自变量值，要求 X>0.0。当 X<0.0 时，本子程序按|X|处理。当 X=0.0 时，本子程序返回函数值 10^{35}。

函数名 MBSL4 返回双精度实型函数值 $\mathrm{K}_N(X)$。

五、子程序（文件名：MBSL4.FOR）

```
      FUNCTION MBSL4(N,X)
      DOUBLE PRECISION MBSL4,X,MBSL3
      DOUBLE PRECISION Y,P,B0,B1,A(7),B(7),C(7),D(7)
```

```
      DATA A/-0.57721566,0.4227842,0.23069756,0.0348859,
     *       0.00262698,0.0001075,0.0000074/
      DATA B/1.0,0.15443144,-0.67278579,-0.18156897,
     *       -0.01919402,-0.00110404,-0.00004686/
      DATA C/1.25331414,-0.07832358,0.02189568,-0.01062446,
     *       0.00587872,-0.0025154,0.00053208/
      DATA D/1.25331414,0.23498619,-0.0365562,0.01504268,
     *       -0.00780353,0.00325614,-0.00068245/
      IF (N.LT.0) N=-N
      IF (X.LT.0.0) X=-X
      IF (X+1.0.EQ.1.0) THEN
        MBSL4=1.0D+35
        RETURN
      END IF
      IF (N.NE.1) THEN
        IF (X.LE.2.0) THEN
          Y=X*X/4.0
          P=A(7)
          DO 10 I=6,1,-1
10        P=P*Y+A(I)
          P=P-MBSL3(0,X)*LOG(X/2.0)
        ELSE
          Y=2.0/X
          P=C(7)
          DO 20 I=6,1,-1
20        P=P*Y+C(I)
          P=P*EXP(-X)/SQRT(X)
        END IF
      END IF
      IF (N.EQ.0) THEN
        MBSL4=P
        RETURN
      END IF
      B0=P
      IF (X.LE.2.0) THEN
        Y=X*X/4.0
        P=B(7)
        DO 30 I=6,1,-1
30      P=P*Y+B(I)
        P=P/X+MBSL3(1,X)*LOG(X/2.0)
      ELSE
        Y=2.0/X
```

```
      P=D(7)
      DO 40 I=6,1,-1
40    P=P*Y+D(I)
      P=P*EXP(-X)/SQRT(X)
    END IF
    IF (N.EQ.1) THEN
      MBSL4=P
      RETURN
    END IF
    B1=P
    Y=2.0/X
    DO 50 I=1,N-1
      P=B0+I*Y*B1
      B0=B1
      B1=P
50  CONTINUE
    MBSL4=P
    RETURN
    END
```

六、例

计算阶数 n=0,1,2,3,4,5 时,自变量 x 分别取 0.05,0.5,5.0,50.0 时变型第二类贝塞耳函数值 $K_n(x)$。

主程序(文件名:MBSL40.FOR)为

```
      DOUBLE PRECISION X,Y,MBSL4
      WRITE(*,*)
      DO 10 N=0,5
        X=0.05
        DO 20 I=1,4
          Y=MBSL4(N,X)
          WRITE(*,100) N,X,Y
          X=X*10.0
20      CONTINUE
10    CONTINUE
100   FORMAT(1X,'N=',I2,2X,'X=',F7.3,3X,'K(n,x)=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
N=  0  X=   .050  K(n,x)=   .311423D+01
N=  0  X=   .500  K(n,x)=   .924419D+00
N=  0  X=  5.000  K(n,x)=   .369110D-02
```

```
N=    0   X=   50.000   K(n,x)=    .341017D-22
N=    1   X=     .050   K(n,x)=    .199097D+02
N=    1   X=     .500   K(n,x)=    .165644D+01
N=    1   X=    5.000   K(n,x)=    .404461D-02
N=    1   X=   50.000   K(n,x)=    .344410D-22
N=    2   X=     .050   K(n,x)=    .799501D+03
N=    2   X=     .500   K(n,x)=    .755018D+01
N=    2   X=    5.000   K(n,x)=    .530894D-02
N=    2   X=   50.000   K(n,x)=    .354793D-22
N=    3   X=     .050   K(n,x)=    .639800D+05
N=    3   X=     .500   K(n,x)=    .620579D+02
N=    3   X=    5.000   K(n,x)=    .829177D-02
N=    3   X=   50.000   K(n,x)=    .372793D-22
N=    4   X=     .050   K(n,x)=    .767840D+07
N=    4   X=     .500   K(n,x)=    .752245D+03
N=    4   X=    5.000   K(n,x)=    .152591D-01
N=    4   X=   50.000   K(n,x)=    .399528D-22
N=    5   X=     .050   K(n,x)=    .122861D+10
N=    5   X=     .500   K(n,x)=    .120980D+05
N=    5   X=    5.000   K(n,x)=    .327063D-01
N=    5   X=   50.000   K(n,x)=    .436718D-22
```

七、附注

本子程序需要调用计算变型第一类贝塞耳函数值 $I_0(x)$ 与 $I_1(x)$ 的子程序 MBSL3(参看12.6节)。

12.8 不完全贝塔函数

一、功能

计算不完全贝塔(Beta)函数值 $B_x(a,b)$

二、方法说明

不完全贝塔函数的定义为

$$B_x(a,b)=\frac{1}{B(a,b)}\int_0^x t^{a-1}(1-t)^{b-1}dt$$

其中 $a,b>0$，$0\leqslant x\leqslant 1$，$B(a,b)$ 为贝塔函数，即

$$B(a,b)=\frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

不完全贝塔函数具有下列对称关系与极限值：

$$B_x(a,b)=1-B_{1-x}(b,a)$$

$$B_0(a,b)=0,\quad B_1(a,b)=1$$

不完全贝塔函数还可以用下列连分式表示：

$$B_x(a,b)=\frac{x^a(1-x)^b}{a\,B(a,b)}\varphi(x)$$

$$\varphi(x)=\frac{1}{1+}\frac{d_1}{1+}\frac{d_2}{1+}\cdots$$

其中

$$d_{2k+1}=-\frac{(a+k)(a+b+k)x}{(a+2k)(a+2k+1)}\quad,\quad k=1,2,\cdots$$

$$d_{2k}=\frac{k(b-k)x}{(a+2k-1)(a+2k)}\quad,\quad k=1,2,\cdots$$

当 $x<\frac{a+1}{a+b+2}$ 时，连分式 $\varphi(x)$ 的收敛速度很快。而当 $x>\frac{a+1}{a+b+2}$ 时，可以用对称关系来计算。

三、子程序语句

FUNCTION MBETA(A,B,X)

四、形参说明

A——双精度实型变量，输入参数。不完全贝塔函数的参数，要求 A>0。当 A≤0 时，本子程序输出信息"ERR * *   A<=0!"，并返回函数值−1.0。

B——双精度实型变量，输入参数。不完全贝塔函数的参数，要求 B>0。当 B≤0 时，本子程序输出信息"ERR * *   B<=0!"，并返回函数值−1.0。

X——双精度实型变量，输入参数。不完全贝塔函数的自变量值，要求 0≤X≤1。当 X<0 或 X>1 时，本子程序输出信息"ERR * *   X<0 or X>1!"，并返回函数值 10^{35}。

函数名 MBETA 返回双精度实型函数值 $B_X(A,B)$。

五、子程序(文件名:MBETA.FOR)

```
      FUNCTION MBETA(A,B,X)
      DOUBLE PRECISION MBETA,A,B,X
      DOUBLE PRECISION Y,BETA,MGAM1
      IF (A.LE.0.0) THEN
        WRITE(*,*)'ERR * * A<=0 !'
        MBETA=-1.0
        RETURN
      END IF
      IF (B.LE.0.0) THEN
        WRITE(*,*)'ERR * * B<=0 !'
        MBETA=-1.0
        RETURN
      END IF
      IF ((X.LT.0.0).OR.(X.GT.1.0)) THEN
        WRITE(*,*)'ERR * * X<0 or X>1 !'
        MBETA=1.0D+35
        RETURN
      END IF
```

```
      IF ((X+1.0.EQ.1.0).OR.(X+1.0.EQ.2.0)) THEN
         Y=0.0
      ELSE
        Y=A*LOG(X)+B*LOG(1.0-X)
        Y=EXP(Y)
        Y=Y*MGAM1(A+B)/(MGAM1(A)*MGAM1(B))
      END IF
      IF (X.LT.(A+1.0)/(A+B+2.0)) THEN
        Y=Y*BETA(A,B,X)/A
      ELSE
        Y=1.0-Y*BETA(B,A,1.0-X)/B
      END IF
      MBETA=Y
      RETURN
      END

      FUNCTION BETA(A,B,X)
      DOUBLE PRECISION BETA,A,B,X
      DOUBLE PRECISION D,P0,Q0,P1,Q1,S0,S1
      P0=0.0
      Q0=1.0
      P1=1.0
      Q1=1.0
      DO 10 K=1,100
        D=(A+K)*(A+B+K)*X
        D=-D/((A+K+K)*(A+K+K+1.0))
        P0=P1+D*P0
        Q0=Q1+D*Q0
        S0=P0/Q0
        D=K*(B-K)*X
        D=D/((A+K+K-1.0)*(A+K+K))
        P1=P0+D*P1
        Q1=Q0+D*Q1
        S1=P1/Q1
        IF (ABS(S1-S0).LT.ABS(S1)*1.0D-07) THEN
          BETA=S1
          RETURN
        END IF
10    CONTINUE
      WRITE(*,*)' A or B too big !'
      BETA=S1
      RETURN
```

```
      END
```

六、例

计算(a,b)为(0.5,0.5),(0.5,5.0),(1.0,3.0),(5.0,0.5),(8.0,10.0)时,x取值为0.0,0.2,0.4,0.6,0.8,1.0时的不完全贝塔函数值 $B_x(a,b)$。

主程序(文件名:MBETA0.FOR)为

```
      DOUBLE PRECISION MBETA,X,A0,B0,Y,A(5),B(5)
      DATA A/0.5,0.5,1.0,5.0,8.0/
      DATA B/0.5,5.0,3.0,0.5,10.0/
      WRITE(*,*)
      DO 20 J=0,5
        X=0.2D0*J
        WRITE(*,100) X
        DO 10 I=1,5
          A0=A(I)
          B0=B(I)
          Y=MBETA(A0,B0,X)
          WRITE(*,200) A0,B0,Y
10      CONTINUE
20    CONTINUE
100   FORMAT(1X,'X=',F4.2)
200   FORMAT(1X,'     B(',F5.2,'  ,  ',F5.2,')=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
 X=  .00
       B(    .50  ,    .50)=    .000000D+00
       B(    .50  ,   5.00)=    .000000D+00
       B(   1.00  ,   3.00)=    .000000D+00
       B(   5.00  ,    .50)=    .000000D+00
       B(   8.00  ,  10.00)=    .000000D+00
 X=  .20
       B(    .50  ,    .50)=    .273989D+00
       B(    .50  ,   5.00)=    .916541D+00
       B(   1.00  ,   3.00)=    .363530D+00
       B(   5.00  ,    .50)=    .818696D-04
       B(   8.00  ,  10.00)=    .958500D-02
 X=  .40
       B(    .50  ,    .50)=    .365257D+00
       B(    .50  ,   5.00)=    .979039D+00
       B(   1.00  ,   3.00)=    .856001D+00
```

B(	5.00 ,	.50)=	.271228D−02
B(	8.00 ,	10.00)=	.228213D+00

X= .60

B(	.50 ,	.50)=	.634743D+00
B(	.50 ,	5.00)=	.997288D+00
B(	1.00 ,	3.00)=	.947637D+00
B(	5.00 ,	.50)=	.209607D−01
B(	8.00 ,	10.00)=	.929174D+00

X= .80

B(	.50 ,	.50)=	.726011D+00
B(	.50 ,	5.00)=	.999916D+00
B(	1.00 ,	3.00)=	.992615D+00
B(	5.00 ,	.50)=	.560966D+00
B(	8.00 ,	10.00)=	.999546D+00

X= 1.00

B(	.50 ,	.50)=	.100000D+01
B(	.50 ,	5.00)=	.100000D+01
B(	1.00 ,	3.00)=	.100000D+01
B(	5.00 ,	.50)=	.100000D+01
B(	8.00 ,	10.00)=	.100000D+01

七、附注

本子程序要调用计算连分式函数 $\varphi(x)$ 的子程序 BETA，由本子程序自带。还要调用计算伽马函数的子程序 MGAM1（参看 12.1 节）。

12.9 正态分布函数

一、功能

计算随机变量 x 的正态分布函数 $P(a,\sigma,x)$ 的值。

二、方法说明

正态分布函数的定义为

$$P(a,\sigma,x)=\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x}e^{-\frac{(t-a)^2}{2\sigma^2}}dt$$

其中 a 为随机变量 x 的数学期望(即平均值)，$\sigma>0$，σ^2 为随机变量 x 的方差。

正态分布函数可以用误差函数来计算，即

$$P(a,\sigma,x)=\frac{1}{2}+\frac{1}{2}\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\sigma}\right)$$

三、子程序语句

FUNCTION MGASS(A,D,X)

四、形参说明

A——双精度实型变量，输入参数。数学期望值。

D——双精度实型变量，输入参数。D^2 为方差值，要求 D>0。

X——双精度实型变量，输入参数。随机变量值。

函数名 MGASS 返回双精度实型函数值 P(A,D,X)。

五、子程序(文件名:MGASS.FOR)

```
      FUNCTION MGASS(A,D,X)
      DOUBLE PRECISION MGASS,A,D,X,MERRF
      IF (D.LE.0.0) D=1.0D-10
      MGASS=0.5+0.5*MERRF((X-A)/(SQRT(2.0)*D))
      RETURN
      END
```

六、例

计算当(a,σ)分别为(-1.0,0.5),(3.0,15.0)时，x 取值为-10.0,-5.0,0.0,5.0,10.0 时的正态分布函数值 $P(a,\sigma,x)$。

主程序(文件名:MGASS0.FOR)为

```
      DOUBLE PRECISION MGASS,A0,D0,X,Y,A(2),D(2)
      DATA A/-1.0,3.0/
      DATA D/0.5,15.0/
      WRITE(*,*)
      DO 30 I=1,2
        A0=A(I)
        D0=D(I)
        X=-10.0
        DO 10 J=0,4
          Y=MGASS(A0,D0,X)
          WRITE(*,100) A0,D0,X,Y
          X=X+5.0
10      CONTINUE
30    CONTINUE
100   FORMAT(1X,'P(',F6.2,' , ',F6.2,' , ',F6.2,')=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
P( -1.00 ,   .50 ,  -10.00)=    .000000D+00
P( -1.00 ,   .50 ,   -5.00)=    .610623D-15
P( -1.00 ,   .50 ,     .00)=    .977250D+00
P( -1.00 ,   .50 ,    5.00)=    .100000D+01
P( -1.00 ,   .50 ,   10.00)=    .100000D+01
P(  3.00 ,  15.00 , -10.00)=    .193062D+00
P(  3.00 ,  15.00 ,  -5.00)=    .296901D+00
```

```
P(   3.00  ,   15.00  ,        .00)=      .420740D+00
P(   3.00  ,   15.00  ,       5.00)=      .553035D+00
P(   3.00  ,   15.00  ,      10.00)=      .679631D+00
```

七、附注

本子程序需要调用误差函数子程序 MERRF(参看 12.3 节)。而误差函数子程序又需要调用不完全伽马函数子程序 MGAM2(参看 12.2 节)。不完全伽马函数子程序又要调用伽马函数子程序 MGAM(参看 12.1 节)。

12.10 t-分布函数

一、功能

计算 t-分布函数值 $P(t,n)$。

二、方法说明

t-分布又称为 Student-分布，它的定义为

$$P(t,n)=\frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)}\int_{-t}^{t}\left(1+\frac{x^2}{n}\right)^{-\frac{n+1}{2}}dx$$

其中 t 为随机变量，n 为自由度，且 $t\geqslant 0$。它的极限值为

$$P(0,n)=0\ ,\ P(\infty,n)=1$$

t-分布函数可以用不完全贝塔函数表示，即

$$P(t,n)=1-B_{\frac{n}{n+t^2}}\left(\frac{n}{2},0.5\right)$$

三、子程序语句

FUNCTION MSTDT(T,N)

四、形参说明

T——双精度实型变量，输入参数。随机变量值，要求 T≥0。

N——整型变量，输入参数。自由度。

函数名 MSTDT 返回双精度实型函数值 P(T,N)。

五、子程序(文件名:MSTDT.FOR)

```
FUNCTION MSTDT(T,N)
DOUBLE PRECISION MSTDT,T,MBETA
IF (T.LT.0.0) T=-T
MSTDT=1.0-MBETA(N/2.0D0,0.5D0,N/(N+T*T))
RETURN
END
```

六、例

计算当 n 分别为 1,2,3,4,5 时，t 的取值为 0.5 与 5.0 时的 t-分布函数值 $P(t,n)$。

主程序(文件名:MSTDT0.FOR)为

```
      DOUBLE PRECISION MSTDT,T,Y
      WRITE(*,*)
      DO 10 N=1,5
        T=0.5
        Y=MSTDT(T,N)
        WRITE(*,100) T,N,Y
        T=5.0
        Y=MSTDT(T,N)
        WRITE(*,100) T,N,Y
10    CONTINUE
100   FORMAT(1X,'P(',F4.2,' , ',I2,')=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
P( .50  ,  1)=   .273939D+00
P(5.00  ,  1)=   .875927D+00
P( .50  ,  2)=   .311108D+00
P(5.00  ,  2)=   .963219D+00
P( .50  ,  3)=   .325943D+00
P(5.00  ,  3)=   .985174D+00
P( .50  ,  4)=   .333855D+00
P(5.00  ,  4)=   .992855D+00
P( .50  ,  5)=   .338772D+00
P(5.00  ,  5)=   .996117D+00
```

七、附注

本子程序需要调用不完全贝塔函数子程序 MBETA(参看 12.8 节)，而不完全贝塔函数子程序又要调用伽马函数子程序 MGAM1(参看 12.1 节)。

12.11 χ^2-分布函数

一、功能

计算 χ^2 变量的分布函数值 $P(\chi^2,n)$。

二、方法说明

χ^2-分布函数的定义为

$$P(\chi^2,n)=\frac{1}{2^{n/2}\Gamma\left(\frac{n}{2}\right)}\int_0^{\chi^2} t^{(n/2)-1}e^{-t/2}dt$$

其中 n 为自由度，且 $\chi^2\geqslant 0$。它的极限值为

$$P(0,n)=0\ ,\ P(\infty,n)=1$$

χ^2-分布函数可以用不完全伽马函数来计算，即

$$P(\chi^2, n) = \Gamma\left(\frac{n}{2}, \frac{\chi^2}{2}\right)$$

三、子程序语句

FUNCTION MCHII(X,N)

四、形参说明

X——双精度实型变量，输入参数。χ^2 变量值。

N——整型变量，输入参数。自由度。

函数名 MCHII 返回双精度实型函数值 $P(\chi^2, N)$。

五、子程序(文件名:MCHII.FOR)

```
      FUNCTION MCHII(X,N)
      DOUBLE PRECISION MCHII,X,MGAM2
      IF (X.LT.0.0) X=-X
      MCHII=MGAM2(N/2.0D0,X/2.0)
      RETURN
      END
```

六、例

计算 n 分别为 1,2,3,4,5 时，χ^2 变量取值为 0.5 与 5.0 时的 χ^2-分布函数值 $P(\chi^2, N)$。

主程序(文件名:MCHII0.FOR)为

```
      DOUBLE PRECISION MCHII,T,Y
      WRITE(*,*)
      DO 10 N=1,5
        T=0.5
        Y=MCHII(T,N)
        WRITE(*,100) T,N,Y
        T=5.0
        Y=MCHII(T,N)
        WRITE(*,100) T,N,Y
10    CONTINUE
100   FORMAT(1X,'P(',F4.2,' , ',I2,')=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
P( .50  ,  1)=    .520500D+00
P(5.00  ,  1)=    .974653D+00
P( .50  ,  2)=    .221197D+00
P(5.00  ,  2)=    .917916D+00
P( .50  ,  3)=    .811086D-01
P(5.00  ,  3)=    .828203D+00
```

P(.50 , 4)= .264968D−01

P(5.00 , 4)= .712695D+00

P(.50 , 5)= .787671D−02

P(5.00 , 5)= .584120D+00

七、附注

本子程序需要调用不完全伽马函数子程序 MGAM2(参看 12.2 节)。而不完全伽马函数子程序又要调用伽马函数子程序 MGAM1(参看 12.1 节)。

12.12 F-分布函数

一、功能

计算随机变量 F 的 F-分布函数值 $P(F,n_1,n_2)$。

二、方法说明

随机变量 F 的 F-分布函数定义为

$$P(F,n_1,n_2)=\frac{\Gamma\left(\frac{n_1+n_2}{2}\right)}{\Gamma\left(\frac{n_1}{2}\right)\Gamma\left(\frac{n_2}{2}\right)}n_1^{\frac{n_1}{2}}n_2^{\frac{n_2}{2}}\int_F^{\infty}\frac{t^{\frac{n_1}{2}-1}}{(n_2+n_1t)^{(n_1+n_2)/2}}\mathrm{d}t$$

其中随机变量 $F\geqslant 0$,n_1 与 n_2 为自由度。它的极限值为

$$P(0,n_1,n_2)=1,P(\infty,n_1,n_2)=0$$

F-分布函数可以用不完全贝塔函数计算,即

$$P(F,n_1,n_2)=B_{n_2/(n_2+n_1F)}\left(\frac{n_2}{2},\frac{n_1}{2}\right)$$

三、子程序语句

FUNCTION MFFFF(F,N1,N2)

四、形参说明

F——双精度实型变量,输入参数。随机变量 F 值,要求 F≥0。

N1,N2——均为整型变量,输入参数。自由度。

函数名 MFFFF 返回双精度实型函数值 P(F,N1,N2)。

五、子程序(文件名:MFFFF.FOR)

```
FUNCTION MFFFF(F,N1,N2)
DOUBLE PRECISION MFFFF,F,MBETA
IF (F.LT.0.0) F=-F
MFFFF=MBETA(N2/2.0D0,N1/2.0D0,N2/(N2+N1*F))
RETURN
END
```

六、例

计算自由度(n_1,n_2)分别为(2,3),(5,10)时,随机变量 F 的取值为 3.5,9.0 时的 F-分布函数值 $P(F,n_1,n_2)$。

主程序(文件名:MFFFF0.FOR)为

```
      DOUBLE PRECISION MFFFF,F,Y
      DIMENSION N(2),M(2)
      DATA N/2,5/
      DATA M/3,10/
      WRITE(*,*)
      DO 10 I=1,2
        N1=N(I)
        N2=M(I)
        F=3.5
        Y=MFFFF(F,N1,N2)
        WRITE(*,100) F,N1,N2,Y
        F=9.0
        Y=MFFFF(F,N1,N2)
        WRITE(*,100) F,N1,N2,Y
10    CONTINUE
100   FORMAT(1X,'P(',F5.2,' , ',I2,' , ',I2,')=',D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
P( 3.50 , 2 ,  3)=  .138025D+00
P( 9.00 , 2 ,  3)=  .502706D-01
P( 3.50 , 5 , 10)=  .357164D-01
P( 9.00 , 5 , 10)=  .168888D-02
```

七、附注

本子程序需要调用不完全贝塔函数子程序 MBETA(参看 12.8 节)。而不完全贝塔函数子程序又要调用伽马函数子程序 MGAM1(参看 12.1 节)。

12.13 正弦积分

一、功能

计算正弦积分值。

二、方法说明

正弦积分的定义为

$$\mathrm{Si}(x)=\int_0^x \frac{\sin t}{t}dt, x>0$$

本子程序采用变步长勒让德-高斯(Legendre-Gauss)求积法。

三、子程序语句

FUNCTION MSINN(X)

四、形参说明

X——双精度实型变量，输入参数，自变量值。

函数名 MSINN 返回双精度实型函数值 Si(X)。

五、子程序(文件名:MSINN.FOR)

```
      FUNCTION MSINN(X)
      DOUBLE PRECISION MSINN,X,FLRGS
      EXTERNAL SI
      DOUBLE PRECISION SI
      IF (X.LE.0.0) THEN
        MSINN=0.0
        RETURN
      END IF
      MSINN=FLRGS(0.0D0,X,SI,1.0E-07)
      RETURN
      END

      FUNCTION SI(X)
      DOUBLE PRECISION SI,X
      SI=SIN(X)/X
      RETURN
      END
```

六、例

计算自变量 x 从 0.5 开始、每隔 2.0 的 10 个正弦积分值 Si(x)。

主程序(文件名:MSINN.FOR)为

```
      DOUBLE PRECISION MSINN,X,Y
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(8X,'X',10X,' SI(X)')
      DO 20 I=0,20
        X=0.5+I
        Y=MSINN(X)
        WRITE(*,30) X,Y
20    CONTINUE
30    FORMAT(1X,F10.2,5X,D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
    X            SI(X)
   .50       .493107D+00
  1.50       .132468D+01
  2.50       .177852D+01
```

3.50	.183313D+01
4.50	.165414D+01
5.50	.146072D+01
6.50	.142179D+01
7.50	.151068D+01
8.50	.162960D+01
9.50	.167446D+01
10.50	.162294D+01
11.50	.153571D+01
12.50	.149234D+01
13.50	.152291D+01
14.50	.159072D+01
15.50	.163258D+01
16.50	.161563D+01
17.50	.156146D+01
18.50	.152128D+01
19.50	.152863D+01
20.50	.157232D+01

七、附注

本子程序需要调用勒让德-高斯求积法子程序 FLRGS(参看 6.7 节)。

12.14 余弦积分

一、功能

计算余弦积分值。

二、方法说明

余弦积分的定义为

$$\mathrm{Ci}(x) = -\int_x^{\infty}\frac{\cos t}{t}\mathrm{d}t, x > 0$$

计算余弦积分的公式为

$$\mathrm{Ci}(x) = \gamma + \ln x - \int_0^x \frac{1-\cos t}{t}\mathrm{d}t$$

其中

$$\gamma = 0.57721566490153286060651$$

为欧拉(Euler)常数。

本子程序采用勒让德-高斯(Legendre-Gauss)求积法计算积分

$$\int_0^x \frac{1-\cos t}{t}\mathrm{d}t$$

三、子程序语句

FUNCTION MCOSS(X)

四、形参说明

X——双精度实型变量，输入参数。自变量值。

函数名 MCOSS 返回双精度实型余弦积分函数值 Ci(X)。

五、子程序(文件名:MCOSS.FOR)

```
      FUNCTION MCOSS(X)
      DOUBLE PRECISION MCOSS,X,FLRGS,Y,R
      EXTERNAL CI
      DOUBLE PRECISION CI
      IF (X.LE.0.0) THEN
        MCOSS=-1.0D+35
        RETURN
      END IF
      Y=X
      R=0.57721566490153286060651
      MCOSS=FLRGS(0.0D0,X,CI,1.0E-07)
      MCOSS=R+LOG(Y)-MCOSS
      RETURN
      END

      FUNCTION CI(X)
      DOUBLE PRECISION CI,X
      CI=(1.0-COS(X))/X
      RETURN
      END
```

六、例

计算自变量 x 从 0.5 开始、每隔 2.0 的 10 个余弦积分值。

主程序(文件名:MCOSS0.FOR)为

```
      DOUBLE PRECISION MCOSS,X,Y
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(9X,'X',10X,' CI(X)')
      DO 20 I=0,20
        X=0.5+I
        Y=MCOSS(X)
        WRITE(*,30) X,Y
20    CONTINUE
30    FORMAT(1X,F10.2,5X,D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
    X             CI(X)
```

.50	-.177784D+00
1.50	.470356D+00
2.50	.285871D+00
3.50	-.321285D-01
4.50	-.193491D+00
5.50	-.142053D+00
6.50	.111015D-01
7.50	.115633D+00
8.50	.994314D-01
9.50	.267806D-02
10.50	-.782840D-01
11.50	-.785705D-01
12.50	-.114083D-01
13.50	.557570D-01
14.50	.655370D-01
15.50	.171928D-01
16.50	-.403075D-01
17.50	-.561024D-01
18.50	-.211074D-01
19.50	.288333D-01
20.50	.485874D-01

七、附注

本子程序需要调用勒让德-高斯求积法子程序 FLRGS(参看 6.7 节)。

12.15 指数积分

一、功能

计算指数积分值。

二、方法说明

指数积分的定义为

$$\mathrm{Ei}(x)=-\int_{x}^{\infty}\frac{e^{-t}}{t}\mathrm{d}t, x>0$$

或

$$\mathrm{Ei}(x)=\int_{-\infty}^{x}\frac{e^{t}}{t}\mathrm{d}t, x<0$$

计算指数积分的公式为

$$\mathrm{Ei}(x)=-\gamma+\ln x+\int_{0}^{x}\frac{e^{-t}-1}{t}\mathrm{d}t, x>0$$

其中

$$\gamma=0.57721566490153286060651$$

为欧拉(Euler)常数。

本子程序采用变步长勒让德-高斯(Legendre-Gauss)求积法计算积分

$$\int_0^x \frac{e^{-t}-1}{t}dt$$

三、子程序语句

FUNCTION MEXPP(X)

四、形参说明

X——双精度实型变量，输入参数。自变量值。

函数名 MEXPP 返回双精度实型指数积分函数值 Ei(X)。

五、子程序(文件名:MEXPP.FOR)

```
      FUNCTION MEXPP(X)
      DOUBLE PRECISION MEXPP,X,FLRGS,Y,R
      EXTERNAL EI
      DOUBLE PRECISION EI
      IF (X.LE.0.0) THEN
        MEXPP=-1.0D+35
        RETURN
      END IF
      Y=X
      R=0.57721566490153286060651
      MEXPP=FLRGS(0.0D0,X,EI,1.0E-07)
      MEXPP=R+LOG(Y)+MEXPP
      RETURN
      END

      FUNCTION EI(X)
      DOUBLE PRECISION EI,X
      EI=(EXP(-X)-1.0)/X
      RETURN
      END
```

六、例

计算自变量 x 从 0.5 开始,每隔 0.2 的 10 个指数积分值 Ei (x)。

主程序(文件名:MEXPP0.FOR)为

```
      DOUBLE PRECISION MEXPP,X,Y
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(9X,'X',10X,' EI(X)')
      DO 20 I=0,20
        X=0.05+I*0.1
        Y=MEXPP(X)
        WRITE(*,30) X,Y
20    CONTINUE
```

```
30    FORMAT(1X,F10.2,5X,D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
   X          E1(X)
  .05      -.246790D+01
  .15      -.146446D+01
  .25      -.104428D+01
  .35      -.794215D+00
  .45      -.625331D+00
  .55      -.503364D+00
  .65      -.411517D+00
  .75      -.340341D+00
  .85      -.284019D+00
  .95      -.238738D+00
 1.05      -.201873D+00
 1.15      -.171555D+00
 1.25      -.146413D+00
 1.35      -.125417D+00
 1.45      -.107777D+00
 1.55      -.928821D-01
 1.65      -.802476D-01
 1.75      -.694887D-01
 1.85      -.602950D-01
 1.95      -.524144D-01
 2.05      -.456406D-01
```

七、附注

本子程序需要调用勒让德-高斯求积法子程序 FLRGS(参看 6.7 节)。

12.16 第一类椭圆积分

一、功能

计算第一类椭圆积分。

二、方法说明

第一类椭圆积分的定义为

$$F(k,\varphi)=\int_0^{\varphi}\frac{1}{\sqrt{1-k^2\sin^2\theta}}\,d\theta,\ 0\leqslant k\leqslant 1$$

当 $\varphi=\frac{\pi}{2}$ 时，$F\left(k,\frac{\pi}{2}\right)$ 称为第一类完全椭圆积分。

当 $|\varphi|>\frac{\pi}{2}$ 时，第一类椭圆积分有如下关系：

$$F(k, n\pi \pm \varphi) = 2nF\left(k, \frac{\pi}{2}\right) \pm F(k, \varphi)$$

本子程序采用变步长勒让德-高斯(Legendre-Gauss)求积法计算上述积分(本子程序中自带)。

三、子程序语句

FUNCTION MELP1(K,F)

四、形参说明

K——双精度实型变量,输入参数。要求 $0 \leqslant K \leqslant 1$。

F——双精度实型变量,输入参数。椭圆积分中的 φ。

函数名 MELP1 返回双精度实型第一类椭圆积分值 F(K,F)。

五、子程序(文件名:MELP1.FOR)

```
      FUNCTION MELP1(K,F)
      DOUBLE PRECISION K,F,MELP1,PI,Y,FF,FFF,E,FK1
      IF ((K.LT.0.0).OR.(K.GT.1.0)) THEN
        MLEP1=-1.0E+37
        RETURN
      END IF
      PI=3.1415926
      Y=ABS(F)
      N=0
10    IF (Y.GE.PI) THEN
        N=N+1
        Y=Y-PI
        GOTO 10
      END IF
      E=1.0
      IF (Y.GE.PI/2) THEN
        N=N+1
        E=-1.0
        Y=PI-Y
      END IF
      IF (N.EQ.0) THEN
        FF=FK1(K,Y)
        MELP1=FF
        RETURN
      ELSE
        FF=FK1(K,PI/2.0)
        FFF=FK1(K,Y)
        MELP1=2.0*N*FF+E*FFF
        RETURN
      END IF
```

```
      END

      FUNCTION FK1(K,Y)
      DIMENSION T(5),C(5)
      DOUBLE PRECISION FK1,K,Y,A,B,F,G,T,C,S,P,H,AA,BB,W,X,Q
      DATA T/-0.9061798459,-0.5384693101,0.0,
     *       0.5384693101,0.9061798459/
      DATA C/0.2369268851,0.4786286705,0.5688888889,
     *       0.4786286705,0.2369268851/
      A=0.0
      B=Y
      M=1
      S=(B-A)*0.02
      P=0.0
10    H=(B-A)/M
      G=0.0
      DO 30 I=1,M
        AA=A+(I-1)*H
        BB=A+I*H
        W=0.0
        DO 20 J=1,5
          X=((BB-AA)*T(J)+(BB+AA))/2.0
          FK1=1.0/SQRT(1.0-K*K*SIN(X)*SIN(X))
          W=W+FK1*C(J)
20      CONTINUE
        G=G+W
30    CONTINUE
      G=G*H/2.0
      Q=ABS(G-P)/(1.0+ABS(G))
      IF ((Q.GE.1.0D-10).AND.(ABS(H).GT.ABS(S))) THEN
        P=G
        M=M+1
        GOTO 10
      END IF
      FK1=G
      RETURN
      END
```

六、例

取 $k=0.5$，φ 取

$$\varphi_i = \frac{\pi}{18}i,\ i = 0,1,\cdots,10$$

计算第一类椭圆积分值。

主程序(文件名:MELP10.FOR)为

```
      DOUBLE PRECISION K,F,MELP1,Y
      K=0.5
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(9X,'F',10X,'FK(K,F)')
      DO 20 I=0,10
        F=0.0+I*3.1415926/18.0
        Y=MELP1(K,F)
        WRITE(*,30) F,Y
20    CONTINUE
30    FORMAT(1X,F10.2,5X,D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
 F        FK(K,F)
 .00      .000000D+00
 .17      .174754D+00
 .35      .350819D+00
 .52      .529429D+00
 .70      .711647D+00
 .87      .898245D+00
1.05      .108955D+01
1.22      .128530D+01
1.40      .148455D+01
1.57      .168575D+01
1.75      .188695D+01
```

12.17 第二类椭圆积分

一、功能

计算第二类椭圆积分。

二、方法说明

第二类椭圆积分的定义为

$$E(k,\varphi)=\int_0^\varphi\sqrt{1-k^2\sin^2\theta}\,d\theta,0\leqslant k\leqslant 1$$

当 $\varphi=\frac{\pi}{2}$时,$E\left(k,\frac{\pi}{2}\right)$称为第二类完全椭圆积分。

当$|\varphi|>\frac{\pi}{2}$时,第二类椭圆积分有如下关系:

$$E(k,n\pi\pm\varphi)=2nE(k,\frac{\pi}{2})\pm E(k,\varphi)$$

本子程序采用变步长勒让德-高斯(Legendre-Gauss)求积法(由本程序自带)计算上述积分。

三、子程序语句

FUNCTION MELP2(K,F)

四、形参说明

K——双精度实型变量,输入参数。要求 0≤K≤1.0。

F——双精度实型变量,输入参数,第二类椭圆积分中的 φ。

五、子程序(文件名:MELP2.FOR)

```
      FUNCTION MELP2(K,F)
      DOUBLE PRECISION MELP2,K,F,Y,PI,E,FF,FFF,EK1
      IF ((K.LT.0.0).OR.(K.GT.1.0)) THEN
        MELP2=-1.0E+37
        RETURN
      END IF
      PI=3.1415926
      Y=ABS(F)
      N=0
10    IF (Y.GE.PI) THEN
        N=N+1
        Y=Y-PI
        GOTO 10
      END IF
      E=1.0
      IF (Y.GE.PI/2) THEN
        N=N+1
        E=-1.0
        Y=PI-Y
      END IF
      IF (N.EQ.0) THEN
        FF=EK1(K,Y)
        MELP2=FF
        RETURN
      ELSE
        FF=EK1(K,PI/2.0)
        FFF=EK1(K,Y)
        MELP2=2.0*N*FF+E*FFF
        RETURN
      END IF
      END

      FUNCTION EK1(K,Y)
```

```
      DIMENSION T(5),C(5)
      DOUBLE PRECISION EK1,K,Y,A,B,F,G,T,C,S,P,H,AA,BB,W,X,Q
      DATA T/-0.9061798459,-0.5384693101,0.0,
     *       0.5384693101,0.9061798459/
      DATA C/0.2369268851,0.4786286705,0.5688888889,
     *       0.4786286705,0.2369268851/
      A=0.0
      B=Y
      M=1
      S=(B-A)*0.02
      P=0.0
10    H=(B-A)/M
      G=0.0
      DO 30 I=1,M
        AA=A+(I-1)*H
        BB=A+I*H
        W=0.0
        DO 20 J=1,5
          X=((BB-AA)*T(J)+(BB+AA))/2.0
          EK1=SQRT(1.0-K*K*SIN(X)*SIN(X))
          W=W+EK1*C(J)
20      CONTINUE
        G=G+W
30    CONTINUE
      G=G*H/2.0
      Q=ABS(G-P)/(1.0+ABS(G))
      IF ((Q.GE.1.0D-10).AND.(ABS(H).GT.ABS(S))) THEN
        P=G
        M=M+1
        GOTO 10
      END IF
      EK1=G
      RETURN
      END
```

六、例

取 $k=0.5$，φ 分别为

$$\varphi_i=\frac{\pi}{18}i,\ i=0,1,\cdots,10$$

计算第二类椭圆积分值 $E(k,\varphi)$。

主程序(文件名:MELP20.FOR)为

```
      DOUBLE PRECISION K,F,Y,MELP2
```

```
      K=0.5
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(9X,'F',10X,'EK(K,F)')
      DO 20 I=0,10
        F=0.0+I*3.1415926/18.0
        Y=MELP2(K,F)
        WRITE(*,30) F,Y
20    CONTINUE
30    FORMAT(1X,F10.2,5X,D13.6)
      WRITE(*,*)
      END
```

运行结果为

```
  F          EK(K,F)
 .00        .000000D+00
 .17        .174312D+00
 .35        .347329D+00
 .52        .517882D+00
 .70        .685060D+00
 .87        .848317D+00
1.05        .100756D+01
1.22        .116318D+01
1.40        .131606D+01
1.57        .146746D+01
1.75        .161887D+01
```

第13章 随机数的产生

13.1 0到1之间均匀分布的一个随机数

一、功能

产生0到1之间均匀分布的一个随机数。

二、方法说明

设 $m=2^{16}$,则产生0到1之间均匀分布随机数的计算公式如下:

$$\begin{cases} R_i = \mathrm{mod}(2053R_{i-1} + 13849,\ m) \\ RND_i = R_i/m \end{cases}$$

三、子程序语句

REAL FUNCTION NRND1(R)

四、形参说明

R——双精度实型变量,输入兼输出参数,随机数的种子。主程序中对应的实元必须为双精度实型变量名,不能是表达式。当主程序中改变了 R 的实参值,则以后产生的随机数将变成一个新的随机数序列。

函数名NRND1返回一个0到1之间均匀分布的随机数。

五、子程序(文件名:NRND1.FOR)

```
REAL FUNCTION NRND1(R)
DOUBLE PRECISION S,U,V,R
S=65536.0
U=2053.0
V=13849.0
M=R/S
R=R-M*S
R=U*R+V
M=R/S
R=R-M*S
NRND1=R/S
RETURN
END
```

六、例

连续产生10个0到1之间均匀分布的随机数。R的初值取5.0。

主程序(文件名:NRND10.FOR)为

```
DOUBLE PRECISION R
```

```
      REAL NRND1
      R=5.0
      WRITE(*,*)
      DO 10 I=1,10
        Y=NRND1(R)
        WRITE(*,20) Y
10    CONTINUE
      WRITE(*,*)
20    FORMAT(1X,F10.6)
      END
```

运行结果为

```
.367950
.613571
.672925
.325943
.372284
.510239
.731262
.492630
.580719
.427414
```

七、附注

(1) 利用本子程序最多可以连续产生2^{16}个0到1之间的随机数。若在不改变种子的情况下,连续产生的随机数超过2^{16}个,则将造成循环。

(2) 在主程序中对随机数种子R赋初值后,每调用一次本子程序,其R值由子程序带回进行循环使用,以保证产生的随机数属同一序列。一旦在主程序中改变了R值,以后产生的随机数将属于一个新的序列。

13.2 0到1之间均匀分布的随机数序列

一、功能

产生n个0到1之间均匀分布的随机数的序列。

二、方法说明

同13.1节。

三、子程序语句

SUBROUTINE NRNDS (N,A,R)

四、形参说明

N——整型变量,输入参数。随机数序列的长度。

A——实型一维数组,长度为N,输出参数。存放0到1之间均匀分布的N个随机数。

R——双精度实型变量,输入兼输出参数。随机数序列的种子,不同的R初值将产生

不同的随机数序列。主程序中对应的实元必须为双精度实型变量名，不能为表达式。

五、子程序(文件名:NRNDS.FOR)

```
      SUBROUTINE NRNDS(N,A,R)
      DIMENSION A(N)
      DOUBLE PRECISION R,S,U,V
      S=65536.0
      U=2053.0
      V=13849.0
      DO 20 I=1,N
        R=U*R+V
        M=R/S
        R=R-M*S
        A(I)=R/S
20    CONTINUE
      RETURN
      END
```

六、例

产生 50 个 0 到 1 之间均匀分布的随机数的序列。R 初值取 1.0。

主程序(文件名:NRNDS0.FOR)为

```
      DOUBLE PRECISION R
      DIMENSION A(50)
      R=1.0
      CALL NRNDS(50,A,R)
      WRITE(*,*)
      WRITE(*,10) (A(I),I=1,50)
      WRITE(*,*)
10    FORMAT(1X,5F10.6)
      END
```

运行结果为

```
 .242645   .362045   .490295   .787796   .556549
 .806564   .087891   .650772   .246429   .130966
 .084900   .510818   .921661   .382126   .715698
 .539810   .441620   .856918   .464661   .159622
 .915680   .102219   .066162   .042130   .703217
 .914902   .504578   .109207   .413605   .341843
 .014282   .532730   .906219   .679916   .079651
 .734573   .290436   .475998   .435059   .386612
 .925629   .526962   .064880   .410721   .421173
 .879684   .203491   .978775   .636444   .831039
```

七、附注

(1) 对于不同的随机数种子 R，将产生不同的 0 到 1 之间均匀分布的随机数序列。

(2) 对于某一个随机数种子 R，最多能产生 2^{16} 个 0 到 1 之间均匀分布的随机数。若 N $>2^{16}$，则会造成循环。

13.3 任意区间内均匀分布的一个随机整数

一、功能

产生给定区间$[a, b]$内均匀分布的一个随机整数。

二、方法说明

首先产生$[0, S]$内均匀分布的随机整数。其计算公式如下：

$$\begin{cases} R_i = \mathrm{mod}(5 * R_{i-1}, 4M) \\ RND_i = \mathrm{INT}(R_i/4) \end{cases}$$

其中 $R_0 \geqslant 1.0$ 的奇数为初值（随机数种子），$S=b-a+1$，$M=2^k$，而 $k=[\log_2 S]+1$。

然后将每个随机数加 a，即实际需要的随机数为 $a+RND_i$。

三、子程序语句

FUNCTION NRAB1 (R,A,B)

四、形参说明

R——实型变量，输入兼输出参数。随机数的种子，要求 $R\geqslant 1.0$，且为奇数。主程序中对应的实元必须为实型变量名，不能是表达式。当主程序中改变了 R 的实参值，则以后产生的随机数将变成一个新的随机数序列。

A，B——均为整型变量，输入参数。随机数所在的区间为[A，B]。

函数名 NRAB1 返回一个在区间[A，B]内的随机整数。

五、子程序（文件名：NRAB1.FOR）

```
      FUNCTION NRAB1(R,A,B)
      INTEGER A,B
      S=B-A+1.0
      K=LOG(S-0.5)/LOG(2.0)+1
      L=1
      DO 10 I=1,K
10    L=2*L
      K=1
      S=4.0*L
20    IF (K.LE.1) THEN
        R=R+R+R+R+R
        M=R/S
        R=R-M*S
        J=A+R/4.0
        IF (J.LE.B) THEN
          NRAB1=J
          K=K+1
```

```
      END IF
      GOTO 20
    END IF
    RETURN
    END
```

六、例

产生 100 个 101 到 200 之间均匀分布的随机整数。取随机数种子为 5.0。

主程序(文件名:NRAB10.FOR)为

```
      INTEGER B,C,P(100)
      B=101
      C=200
      R=5.0
      DO 10 I=1,100
10    P(I)=NRAB1(R,B,C)
      WRITE(*,*)
      WRITE(*,100)
100   FORMAT(1X,'R:')
      WRITE(*,200) (P(I),I=1,100)
200   FORMAT(1X,10I6)
      WRITE(*,*)
      END
```

运行结果为

```
R:
   107   132   129   114   167   176   190   163   156   121
   200   150   180   113   162   151   110   147   105   122
   120   197   198   135   144   189   158   131   124   170
   191   168   181   118   187   148   130   119   192   173
   115   172   175   165   166   171   196   193   178   103
   112   157   126   185   138   159   136   149   155   116
   177   160   141   174   140   169   186   143   184   133
   134   139   164   161   146   199   125   195   188   153
   106   127   104   117   182   123   145   194   183   128
   109   142   179   108   137   154   111   152   101   102
```

七、附注

本子程序产生区间[A,B]上均匀分布的一个随机整数。在同一种子 R 下,如果产生的随机数个数超过 k,则将造成循环,其中 $k=B-A+1$。

13.4 任意区间内均匀分布的随机整数序列

一、功能

产生给定区间[a, b]内均匀分布的随机整数序列。

二、方法说明

同13.3节。

三、子程序语句

SUBROUTINE NRABS (R,A,B,N,P)

四、形参说明

R——实型变量，输入兼输出参数。随机数的种子，要求R≥1.0，且为奇数。主程序中对应的实元必须为实型变量名，不能是表达式。当主程序中改变了R的实参值，则以后产生的随机数将变成一个新的随机数序列。

A，B——均为整型变量，输入参数。随机数序列所在的区间为[A，B]。

N——整型变量，输入参数。随机数序列的长度。

P——整型一维数组，长度为N，输出参数。存放[A，B]区间内均匀分布的N个随机整数。

五、子程序(文件名：NRABS.FOR)

```
      SUBROUTINE NRABS(R,A,B,N,P)
      INTEGER P(N)
      INTEGER A,B
      S=B-A+1.0
      K=LOG(S-0.5)/LOG(2.0)+1
      L=1
      DO 10 I=1,K
10    L=2*L
      K=1
      S=4.0*L
      I=1
20    IF ((I.LE.L).AND.(K.LE.N)) THEN
        R=R+R+R+R+R
        M=R/S
        R=R-M*S
        J=A+R/4.0
        IF (J.LE.B) THEN
          P(K)=J
          K=K+1
        END IF
        I=I+1
        GOTO 20
      END IF
      RETURN
      END
```

六、例

产生100个101到200之间均匀分布的随机整数。取随机数种子为R=1.0。

主程序(文件名:NRABS0.FOR)为

```
      INTEGER P(100),B,C
      N=100
      B=101
      C=200
      R=1.0
      CALL NRABS(R,B,C,N,P)
      WRITE(*,*)
      WRITE(*,10)
10    FORMAT(1X,'R:')
      WRITE(*,20) (P(I),I=1,N)
20    FORMAT(1X,10I6)
      WRITE(*,*)
      END
```

运行结果为

R:

102	107	132	129	114	167	176	190	163	156
121	200	150	180	113	162	151	110	147	105
122	120	197	198	135	144	189	158	131	124
170	191	168	181	118	187	148	130	119	192
173	115	172	175	165	166	171	196	193	178
103	112	157	126	185	138	159	136	149	155
116	177	160	141	174	140	169	186	143	184
133	134	139	164	161	146	199	125	195	188
153	106	127	104	117	182	123	145	194	183
128	109	142	179	108	137	154	111	152	101

七、附注

本子程序产生区间[A,B]上均匀分布的 N 个随机整数。如果 $N>k$,则后 $N-k$ 个随机整数为 0,其中 $k=B-A+1$。

13.5 任意均值与方差的一个正态分布随机数。

一、功能

产生给定均值与方差的正态分布的一个随机数。

二、方法说明

产生均值为 μ、方差为 σ^2 的正态分布随机数 y 的计算公式为:

$$y=\mu+\sigma\frac{\left(\sum_{i=1}^{n}RN_i\right)-\frac{n}{2}}{\sqrt{n/12}}$$

其中 n 足够大。通常取 $n=12$,其近似程度已是相当好了,此时有

$$y = \mu + \sigma(\sum_{i=1}^{12} RN_i - 6)$$

其中 RN_i 为 0 到 1 之间均匀分布的随机数。

三、子程序语句

REAL FUNCTION NGRN1 (U,G,R)

四、形参说明

U——实型变量，输入参数。正态分布的均值 μ。

G——实型变量，输入参数。正态分布的方差 $\sigma^2=G^2$。

R——双精度实型变量，输入兼输出参数。随机数的种子。主程序中对应的实元必须为双精度实型变量名，不能为表达式。当主程序中改变了 R 的实参值，以后产生的随机数将变成一个新的序列。

函数名 NGRN1 返回一个具有均值 U、方差 G^2 的正态分布随机数。

五、子程序(文件名:NGRN1.FOR)

```
      REAL FUNCTION NGRN1(U,G,R)
      DOUBLE PRECISION R,S,W,V,T
      S=65536.0
      W=2053.0
      V=13849.0
      T=0.0
      DO 10 I=1,12
        R=W*R+V
        M=R/S
        R=R-M*S
        T=T+R/S
10    CONTINUE
      NGRN1=U+G*(T-6.0)
      RETURN
      END
```

六、例

产生 50 个均值为 1.0、方差为 1.5^2 的正态分布随机数。随机数种子取 R=5.0。

主程序(文件名:NGRN10.FOR)为

```
      DOUBLE PRECISION R
      REAL NGRN1,A(50)
      R=5.0
      U=1.0
      G=1.5
      N=50
      DO 10 I=1,N
10    A(I)=NGRN1(U,G,R)
      WRITE(*,*)
```

```
      WRITE(*,20) (A(I),I=1,50)
20    FORMAT(1X,5F10.6)
      WRITE(*,*)
      END
```

运行结果为

1.238632	−1.177994	.512802	1.904770	1.591660
1.167221	−.774796	1.859360	1.663437	1.731186
.656357	−1.467309	−.046036	−.986099	.306259
3.924789	−.536758	3.515366	−.325089	.035629
.191269	.235580	.262314	.365219	3.638046
1.174545	2.068466	.413559	2.303574	4.832260
−.906631	.180649	2.187851	2.208725	1.837021
−.333511	3.290878	.803940	1.299423	−1.128922
4.112656	.617905	.480576	.794418	−1.346817
1.650620	2.380478	2.436508	1.912460	.902084

七、附注

在主程序中对随机数种子 R 赋初值后，以后均由本子程序带回。如果在主程序中改变了 R 值，则以后产生的随机数属一个新的序列。

13.6 任意均值与方差的正态分布随机数序列

一、功能

产生给定均值 μ 与方差 σ^2 的正态分布的 n 个随机数。

二、方法说明

同 13.5 节。

三、子程序语句

SUBROUTINE NGRNS (U,G,R,N,A)

四、形参说明

U——实型变量，输入参数。正态分布的均值 μ。

G——实型变量，输入参数。正态分布的方差 $\sigma^2=G^2$。

R——双精度实型变量，输入兼输出参数。随机数的种子，不同的 R 将产生不同的随机数序列。主程序中对应的实元必须为双精度实型变量名，不能为表达式。

N——整型变量，输入参数。随机数序列的长度。

A——实型一维数组，长度为 N，输出参数。存放给定均值 U、方差 G^2 与随机数种子 R 的正态分布的 N 个随机数。

五、子程序（文件名：NGRNS.FOR）

```
      SUBROUTINE NGRNS(U,G,R,N,A)
      DOUBLE PRECISION R,S,W,V,T
```

```
      DIMENSION A(N)
      S=65536.0
      W=2053.0
      V=13849.0
      DO 20 J=1,N
        T=0.0
        DO 10 I=1,12
          R=W*R+V
          M=R/S
          R=R-M*S
          T=T+R/S
10      CONTINUE
        A(J)=U+G*(T-6.0)
20    CONTINUE
      RETURN
      END
```

六、例

产生 50 个均值为 1.0、方差为 1.5^2 的正态分布随机数。随机数种子取 R=3.0。

主程序(文件名:NGRNS0.FOR)为

```
      DOUBLE PRECISION R
      DIMENSION A(50)
      R=3.0
      U=1.0
      G=1.5
      N=50
      WRITE(*,*)
      CALL NGRNS(U,G,R,N,A)
      WRITE(*,20) (A(I),I=1,50)
20    FORMAT(1X,5F10.6)
      WRITE(*,*)
      END
```

运行结果为

```
 -.412430   .367233  -.042557  3.701950  1.944504
 1.528854 -1.201248  2.097946 -2.729813   .159225
 -.391190  2.452692   .564621  -.741653  2.387619
 1.796188  1.327805   .326218  2.635178  1.598434
 1.059738  3.362839 -1.148514  3.369431  1.260422
 2.368210  1.536545  1.109177   .429855  3.342331
  .190353  2.317673  1.568039  1.785202   .812912
```

.994919	.174973	2.196823	1.904221	1.640915
.750656	3.077194	.464279	2.755461	.295090
−1.573685	.993088	2.839157	1.808273	3.244186

七、附注

对于不同的均值 μ 与方差 σ^2,取不同的随机数种子R,产生的正态分布随机数序列也将是不同的。

第14章 多项式与一般函数的计算

14.1 一维多项式求值

一、功能

计算多项式

$$P(x)=a_n x^{n-1}+a_{n-1}x^{n-2}+\cdots+a_2 x+a_1$$

在指定点 x 处的函数值。

二、方法说明

首先将多项式表述成如下嵌套形式：

$$P(x)=(\cdots((a_n x+a_{n-1})x+a_{n-2})x+\cdots+a_2)x+a_1$$

然后从里往外一层一层地进行计算，其递推计算的公式如下：

$$u_n=a_n$$

$$u_k=u_{k+1}x+a_k,\ k=n-1,\cdots,2,1$$

最后得到的 u_1 即是多项式值 $P(x)$。

三、子程序语句

FUNCTION OPLYV(A,N,X)

四、形参说明

A——双精度实型一维数组，长度为 N，输入参数。存放 N−1 次多项式的 N 个系数 $a_n,a_{n-1},\cdots,a_1$。

N——整型变量，输入参数。多项式的项数，其次数为 N−1。

X——双精度实型变量，输入参数。自变量值。

函数名 OPLYV 返回双精度实型函数值。

五、子程序(文件名:OPLYV.FOR)

```
      FUNCTION OPLYV(A,N,X)
      DOUBLE PRECISION OPLYV,A(N),X
      OPLYV=A(N)
      DO 10 I=N-1,1,-1
10    OPLYV=OPLYV*X+A(I)
      RETURN
      END
```

六、例

计算多项式

$$P(x)=2x^6-5x^5+3x^4+x^3-7x^2+7x-20$$

在 $x=\pm0.9,\pm1.1,\pm1.3$ 处的函数值。

主程序(文件名:OPLYV0.FOR)为

```
      DOUBLE PRECISION A(7),X(6),OPLYV,Y
      DATA A/-20.0,7.0,-7.0,1.0,3.0,-5.0,2.0/
      DATA X/0.9,-0.9,1.1,-1.1,1.3,-1.3/
      DO 10 I=1,6
        Y=OPLYV(A,7,X(I))
        WRITE(*,100) I,X(I),I,Y
10    CONTINUE
100   FORMAT(1X,'X(',I2,')=',F5.2,5X,'P(',I2,')=',D13.6)
      END
```

运行结果为

```
X( 1)=   .90     P( 1)= -.185623D+02
X( 2)=  -.90     P( 2)= -.267154D+02
X( 3)=  1.10     P( 3)= -.195561D+02
X( 4)= -1.10     P( 4)= -.216130D+02
X( 5)=  1.30     P( 5)= -.208757D+02
X( 6)= -1.30     P( 6)= -.634044D+01
```

14.2 一维多项式多组求值

一、功能

利用系数预处理法对多项式

$$P(x)=a_nx^{n-1}+a_{n-1}x^{n-2}+\cdots+a_2x+a_1$$

进行多组求值。其中 $n=2^k(k\geqslant1)$

二、方法说明

该多项式为

$$P(x)=a_nx^{n-1}+a_{n-1}x^{n-2}+\cdots+a_2x+a_1$$

其中 $n=2^k(k\geqslant1)$。

首先将多项式变为首一多项式,即令

$$T(x)=P(x)/a_n$$
$$=x^{n-1}+t_{n-1}x^{n-2}+\cdots+t_2x+t_1$$

其中

$$t_k=a_k/a_n,\ k=n-1,\cdots,2,1$$

然后对首一多项式 $T(x)$ 中的系数进行预处理。其预处理过程如下。

首先,将 $T(x)$ 分解为如下形式:

$$T(x)=(x^j+b)q(x)+r(x)$$

其中 $j=2^{k-1}$,$q(x)$ 与 $r(x)$ 均为 $2^{k-1}-1$ 次的首一多项式,b,$q(x)$ 与 $r(x)$ 按以下规则确定:

多项式中的中项系数减 1 为 b，即

$$b = t_{2^{k-1}} - 1$$

多项式左半部分除以 x^s 为 $q(x)$，即

$$q(x) = x^{s-1} + q_{s-1}x^{s-2} + \cdots + q_2x + q_1$$

其中

$$s = 2^{k-1}$$

$$q_i = t_{i+s},\ i = s-1,\cdots,2,1$$

多项式右半部分减去 b 与 $q(x)$ 中各系数的乘积，即

$$r(x) = x^{s-1} + r_{s-1}x^{s-2} + \cdots + r_2x + r_1$$

其中

$$r_i = t_i - bq_i,\ i = s-1,\cdots,2,1$$

由于 $q(x)$ 与 $r(x)$ 还是首一多项式，也可以按照上述方法分别进行分解。这个过程一直作到 $q(x)$ 与 $r(x)$ 的次数为 1 为止，并且，每次分解后的系数仍放在 $T(x)$ 的各系数的存储单元中。最后就可以得到 $T(x)$ 经分解处理后的系数。

首一多项式 $T(x)$ 的系数经预处理后，就可以用这些预处理后的系数对不同的 x 求函数值。

这种方法特别适用于对多个 x 进行求值，减少求值中的乘法次数。

如果原来的多项式不满足 $n = 2^k$ 这个条件时，可以添加系数为 0 的各项及系数为 1 的最高次项。

三、子程序语句

SUBROUTINE OPLYS(A,N,B,M,X,L,P)

四、形参说明

A——双精度实型一维数组，长度为 N，输入参数。存放 N−1 次多项式的系数：

$$P(x) = a_Nx^{N-1} + a_{N-1}x^{N-2} + \cdots + a_2x + a_1$$

返回时该数组将被破坏。

N——整型变量，输入参数。多项式 $P(x)$ 的项数，即该多项式的最高次为 N−1。

B——双精度实型一维数组，长度为 M=2*N，工作数组。

M——整型变量，输入参数。工作数组 B 的长度，M=2*N。

X——双精度实型一维数组，长度为 L，输入参数。存放 L 组自变量值。

L——整型变量，输入参数。给定自变量的个数。

P——双精度实型一维数组，长度为 L，输出参数。存放给定自变量值的多项式值，即

$$P(k) = P(X(k)),\ k = 1,2,\cdots,L$$

五、子程序(文件名：OPLYS.FOR)

```
      SUBROUTINE OPLYS(A,N,B,M,X,L,P)
      DOUBLE PRECISION A(N),B(M),X(L),P(L)
      DOUBLE PRECISION Y,Z
      INTEGER T,S
      Y=A(N)
```

```
      DO 10 I=1,N
10    B(I)=A(I)/Y
      K=LOG(N-0.5)/LOG(2.0)+1
      NN=1
      DO 20 I=1,K
20    NN=2*NN
      DO 40 I=N+1,NN-1
40    B(I)=0.0
      B(NN)=1.0
      T=NN
      S=1
      DO 70 I=1,K-1
        T=T/2
        MM=-T
        DO 60 J=1,S
          MM=MM+T+T
          B(MM)=B(MM)-1.0
          DO 50 KK=2,T
50        B(MM-KK+1)=B(MM-KK+1)-B(MM)*B(MM+T-KK+1)
60      CONTINUE
        S=S+S
70    CONTINUE
      DO 200 KK=1,L
        DO 90 I=0,(NN-2)/2
90      A(I+1)=X(KK)+B(2*I+1)
        MM=1
        Z=X(KK)
        DO 110 I=1,K-1
          MM=MM+MM
          LL=MM+MM
          Z=Z*Z
          DO 100 J=0,NN-1,LL
100       A(J/2+1)=A(J/2+1)+A((J+MM)/2+1)*(Z+B(J+MM))
110     CONTINUE
        Z=Z*Z/X(KK)
        IF (NN.NE.N) A(1)=A(1)-Z
        P(KK)=A(1)*Y
200   CONTINUE
      RETURN
      END
```

六、例

(1) 利用系数预处理法计算多项式

$$P(x) = 2x^6 + 5x^5 + 3x^4 + x^3 - 7x^2 + 7x - 20$$

在 $x = \pm 0.9, \pm 1.1, \pm 1.3$ 处的函数值。

主程序(文件名:OPLYS0.FOR)为

```
      DOUBLE PRECISION A(7),X(6),B(14),P(6)
      DATA X/0.9,-0.9,1.1,-1.1,1.3,-1.3/
      DATA A/-20.0,7.0,-7.0,1.0,3.0,-5.0,2.0/
      CALL OPLYS(A,7,B,14,X,6,P)
      DO 20 I=1,6
20    WRITE(*,100) I,X(I),I,P(I)
100   FORMAT(1X,'X(',I2,')=',F5.2,5X,'P(',I2,')=',D13.6)
      END
```

运行结果为

```
X( 1)=   .90    P( 1)= -.185623D+02
X( 2)=  -.90    P( 2)= -.267154D+02
X( 3)=  1.10    P( 3)= -.195561D+02
X( 4)= -1.10    P( 4)= -.215130D+02
X( 5)=  1.30    P( 5)= -.208757D+02
X( 6)= -1.30    P( 6)= -.634044D+01
```

(2) 利用系数预处理法计算多项式

$$P(x) = \sum_{k=1}^{16} kx^{k-1}$$

在 $x = \pm 0.9, \pm 1.1, \pm 1.3$ 处的函数值。

主程序(文件名:OPLYS1.FOR)为

```
      DOUBLE PRECISION A(16),X(6),B(32),P(6)
      DATA X/0.9,-0.9,1.1,-1.1,1.3,-1.3/
      DATA A/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,
     *       11.0,12.0,13.0,14.0,15.0,16.0/
      CALL OPLYS(A,16,B,32,X,6,P)
      DO 20 I=1,6
20    WRITE(*,100) I,X(I),I,P(I)
100   FORMAT(1X,'X(',I2,')=',F5.2,5X,'P(',I2,')=',D13.6)
      END
```

运行结果为

```
X( 1)=   .90    P( 1)=  .518215D+02
X( 2)=  -.90    P( 2)= -.133476D+01
X( 3)=  1.10    P( 3)=  .375698D+03
X( 4)= -1.10    P( 4)= -.358245D+02
X( 5)=  1.30    P( 5)=  .282065D+04
X( 6)= -1.30    P( 6)= -.475288D+03
```

14.3 二维多项式求值

一、功能

计算二维多项式

$$P(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}x^{i-1}y^{j-1}$$

在给定点 (x,y) 处的函数值。

二、方法说明

将二维多项式变形如下：

$$P(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}x^{i-1}y^{j-1}$$
$$=\sum_{i=1}^{m}\Big[\sum_{j=1}^{n}(a_{ij}x^{i-1})y^{j-1}\Big]$$

令

$$S_i=\sum_{j=1}^{n}(a_{ij}x^{i-1})y^{j-1}$$

则计算 S_i 的递推公式如下：

$$\begin{cases}u_n=a_{in}x^{i-1}\\ u_j=u_{j+1}y+a_{ij}x^{i-1},\quad j=n-1,\cdots,2,1\end{cases}$$

其中最后的 u_1 即为 S_i。

最后将所有 S_i 累加，即

$$P(x,y)=\sum_{i=1}^{m}S_i$$

三、子程序语句

FUNCTION OBPLY(A,M,N,X,Y)

四、形参说明

A——双精度实型二维数组，体积为M×N，输入参数。二维多项式

$$P(x,y)=\sum_{i=1}^{M}\sum_{j=1}^{N}a_{ij}x^{i-1}y^{j-1}$$

的系数矩阵 a_{ij}。

M——整型变量，输入参数。变量 x 的最高次数为M－1。

N——整型变量，输入参数。变量 y 的最高次数为N－1。

X,Y——均为双精度实型变量，输入参数。给定的自变量一对值。

函数名OBPLY返回双精度实型函数值。

五、子程序(文件名:OBPLY.FOR)

```
      FUNCTION OBPLY(A,M,N,X,Y)
      DOUBLE PRECISION A(M,N),X,Y,OBPLY
      DOUBLE PRECISION XX,S
```

```
      OBPLY=0.0
      XX=1.0
      DO 100 I=1,M
        S=A(I,N)*XX
        DO 50 J=N-1,1,-1
50      S=S*Y+A(I,J)*XX
        OBPLY=OBPLY+S
        XX=XX*X
100   CONTINUE
      RETURN
      END
```

六、例

计算二维多项式

$$P(x,y)=\sum_{i=1}^{4}\sum_{j=1}^{5}a_{ij}x^{i-1}y^{j-1}$$

在(0.6,—1.3)处的函数值。其中系数矩阵为

$$A=\begin{bmatrix}1.0 & 2.0 & 3.0 & 4.0 & 5.0\\ 6.0 & 7.0 & 8.0 & 9.0 & 10.0\\ 11.0 & 12.0 & 13.0 & 14.0 & 15.0\\ 16.0 & 17.0 & 18.0 & 19.0 & 20.0\end{bmatrix}$$

主程序(文件名:OBPLY0.FOR)为

```
      DOUBLE PRECISION A(4,5),X,Y,OBPLY
      DATA A/1.0,6.0,11.0,16.0,
     *       2.0,7.0,12.0,17.0,
     *       3.0,8.0,13.0,18.0,
     *       4.0,9.0,14.0,19.0,
     *       5.0,10.0,15.0,20.0/
      X=0.6
      Y=-1.3
      WRITE(*,100) X,Y,OBPLY(A,4,5,X,Y)
100   FORMAT(1X,'P(',F4.2,' , ',F5.2,')=',D13.6)
      END
```

运行结果为

```
P( .60 , -1.30)=  .396655D+02
```

14.4 复系数多项式求值

一、功能

计算复系数多项式

$$P(z)=a_nz^{n-1}+a_{n-1}z^{n-2}+\cdots+a_2z+a_1$$

在给定复数 z 时的函数值。

二、方法说明

同 14.1 节。只是改成复数运算。

本子程序中要调用复数乘法的子程序 PCMUL(见 15.1 节)。

三、子程序语句

SUBROUTINE OCPLY(AR,AI,N,X,Y,U,V)

四、形参说明

AR,AI——均为双精度实型一维数组,长度均为N,输入参数。其中 AR(i) 为系数 a_i 的实部,AI(i) 为 a_i 的虚部。

N——整型变量。输入参数。多项式的项数,即多项式的次数为 N−1。

X,Y——均为双精度实型变量,输入参数。分别为给定复数的实部与虚部,即给定复数为

$$z = X + jY$$

U,V——均为双精度实型变量,输出参数。多项式值的实部与虚部,即

$$U + jV = P(X + jY)$$

五、子程序(文件名:OCPLY.FOR)

```
      SUBROUTINE OCPLY(AR,AI,N,X,Y,U,V)
      DOUBLE PRECISION AR(N),AI(N),X,Y,U,V
      U=AR(N)
      V=AI(N)
      DO 10 I=N-1,1,-1
        CALL PCMUL(U,V,X,Y,U,V)
        U=U+AR(I)
        V=V+AI(I)
10    CONTINUE
      RETURN
      END
```

六、例

计算复数多项式

$$P(z) = (2 + j2)z^3 + (1 + j)z^2 + (2 + j)z + (2 + j)$$

当 $z = 1 + j$ 时的函数值。

主程序(文件名:OCPLY0.FOR)为

```
      DOUBLE PRECISION AR(4),AI(4),X,Y,U,V
      DATA AR/2.0,2.0,1.0,2.0/
      DATA AI/1.0,1.0,1.0,2.0/
      X=1.0
      Y=1.0
      CALL OCPLY(AR,AI,4,X,Y,U,V)
      WRITE(*,100) U,V
100   FORMAT(1X,D13.6,' +j',D13.6)
```

```
      END
```

运行结果为

-.700000D+01 +j .600000D+01

七、附注

本子程序需要调用复数乘法子程序 PCMUL(参见15.1节)。

14.5 多项式相乘

一、功能

求两个多项式

$$P_m(x) = p_m x^{m-1} + p_{m-1} x^{m-2} + \cdots + p_2 x + p_1$$

$$Q_n(x) = q_n x^{n-1} + q_{n-1} x^{n-2} + \cdots + q_2 x + q_1$$

的乘积多项式

$$S_{m+n-1}(x) = P_m(x) Q_n(x)$$
$$= s_{m+n-1} x^{m+n-2} + s_{m+n-2} x^{m+n-3} + \cdots + s_2 x + s_1$$

二、方法说明

乘积多项式 $S_{m+n-1}(x)$ 中的各系数按如下公式计算:

$s_k = 0,\ k = 1,2,\cdots,m+n-1$

$s_{i+j-1} = s_{i+j-1} + p_i q_j,\ i = 1,2,\cdots,m;\ j = 1,2,\cdots,n$

三、子程序语句

SUBROUTINE OPMUL(P,M,Q,N,S,L)

四、形参说明

P——双精度实型一维数组,长度为M,输入参数。M-1次多项式 $P_m(x)$ 的系数。

M——整型变量,输入参数。多项式 $P_m(x)$ 的项数。

Q——双精度实型一维数组,长度为N,输入参数。N-1次多项式 $Q_n(x)$ 的系数。

N——整型变量,输入参数。多项式 $Q_n(x)$ 的项数。

S——双精度实型一维数组,长度为L,输出参数。乘积多项式 $P_m(x)Q_n(x)$ 的系数。其中L=M+N-1,即乘积多项式 $S_{m+n-1}(x)$ 的次数为M+N-2。

L——整型变量,输入参数。乘积多项式 $S_{m+n-1}(x)$ 的项数,其中L=M+N-1。

五、子程序(文件名:OPMUL.FOR)

```
      SUBROUTINE OPMUL(P,M,Q,N,S,L)
      DOUBLE PRECISION P(M),Q(N),S(L)
      DO 10 I=1,L
10    S(I)=0.0
      DO 50 I=1,M
      DO 50 J=1,N
        K=I+J-1
        S(K)=S(K)+P(I)*Q(J)
50    CONTINUE
```

```
      RETURN
      END
```

六、例

计算下列两个多项式

$$P_6(x) = 3x^5 - x^4 + 2x^3 + 5x^2 - 6x + 4$$

$$Q_4(x) = 2x^3 - 6x^2 + 3x + 2$$

的乘积多项式 $S_9(x) = P_6(x)Q_4(x)$ 。

主程序(文件名:OPMUL0.FOR)为

```
      DOUBLE PRECISION P(6),Q(4),S(9)
      DATA P/4.0,-6.0,5.0,2.0,-1.0,3.0/
      DATA Q/2.0,3.0,-6.0,2.0/
      CALL OPMUL(P,6,Q,4,S,9)
      DO 10 I=1,9
10    WRITE(*,100) I,S(I)
100   FORMAT(1X,'S(',I2,')=',D13.6)
      END
```

运行结果为

```
S( 1)=   .800000D+01
S( 2)=   .000000D+00
S( 3)= -.320000D+02
S( 4)=   .630000D+02
S( 5)= -.380000D+02
S( 6)=   .100000D+01
S( 7)=   .190000D+02
S( 8)= -.200000D+02
S( 9)=   .600000D+01
```

14.6 多项式相除

一、功能

求多项式

$$P_m(x) = p_m x^{m-1} + p_{m-1} x^{m-2} + \cdots + p_2 x + p_1$$

被多项式

$$Q_n(x) = q_n x^{n-1} + q_{n-1} x^{n-2} + \cdots + q_2 x + q_1$$

除以后的商多项式 $S_{m-n+1}(x)$ 与余多项式 $R_{n-1}(x)$ 。

二、方法说明

本子程序采用综合除法求商多项式中的各系数。

设商多项式最高次为 $k = m - n$，则 $S_{k+1}(x)$ 的系数由如下过程来确定($i = 0,1,\cdots,k$)：

$$s_{k+1-i} = p_{m-i}/q_n$$

$$p_j = p_j - s_{k+1-i}q_{j+i-k},\quad j = m-i, m-i-1, \cdots, k+1-i$$

最后的 $p_1, p_2, \cdots, p_{n-1}$ 即为余多项式的系数 $r_1, r_2, \cdots, r_{n-1}$。

三、子程序语句

SUBROUTINE OPDIV(P,M,Q,N,S,K,R,L)

四、形参说明

P——双精度实型一维数组，长度为 M，输入参数。多项式 $P_m(x)$ 的系数。

M——整型变量，输入参数。多项式 $P_m(x)$ 的项数。

Q——双精度实型一维数组，长度为 N，输入参数。多项式 $Q_n(x)$ 的系数。

N——整型变量，输入参数。多项式 $Q_n(x)$ 的项数。

S——双精度实型一维数组，长度为 K，输出参数。商多项式的系数。其中 K=M−N+1。

K——整型变量，输入参数。商多项式的项数。要求 K=M−N+1。

R——双精度实型一维数组，长度为 L，输出参数。余多项式的系数。其中 L=N−1。

L——整型变量，输入参数。余多项式的项数。要求 L=N−1。

五、子程序(文件名:OPDIV.FOR)

```
      SUBROUTINE OPDIV(P,M,Q,N,S,K,R,L)
      DOUBLE PRECISION P(M),Q(N),S(K),R(L)
      DO 10 I=1,K
10    S(I)=0.0
      IF (Q(N)+1.0.EQ.1.0) RETURN
      LL=M-1
      DO 50 I=K,1,-1
        S(I)=P(LL+1)/Q(N)
        MM=LL
        DO 30 J=1,N-1
          P(MM)=P(MM)-S(I)*Q(N-J)
          MM=MM-1
30      CONTINUE
        LL=LL-1
50    CONTINUE
      DO 80 I=1,L
80    R(I)=P(I)
      RETURN
      END
```

六、例

求多项式

$$P_5(x) = 3x^4 + 6x^3 - 3x^2 - 5x + 8$$

被多项式

$$Q_3(x) = 2x^2 - x + 1$$

除以后的商多项式 $S_3(x)$ 与余多项式 $R_2(x)$ 。

主程序(文件名:OPDIV0.FOR)为

```
      DOUBLE PRECISION P(5),Q(3),S(3),R(2)
      DATA P/8.0,-5.0,-3.0,6.0,3.0/
      DATA Q/1.0,-1.0,2.0/
      CALL OPDIV(P,5,Q,3,S,3,R,2)
      DO 10 I=1,3
10    WRITE(*,100) I,S(I)
      WRITE(*,*)
      DO 20 I=1,2
20    WRITE(*,200) I,R(I)
100   FORMAT(1X,'S(',I2,')=',D13.6)
200   FORMAT(1X,'R(',I2,')=',D13.6)
      END
```

运行结果为

```
S( 1)= -.375000D+00
S( 2)=  .375000D+01
S( 3)=  .150000D+01

R( 1)=  .837500D+01
R( 2)= -.912500D+01
```

14.7 复系数多项式相乘

一、功能

求两个复系数多项式

$$P_m(x) = p_m x^{m-1} + p_{m-1}x^{m-2} + \cdots + p_2 x + p_1$$

$$Q_n(x) = q_n x^{n-1} + q_{n-1}x^{n-2} + \cdots + q_2 x + q_1$$

的乘积多项式

$$S_{m+n-1}(x) = s_{m+n-1}x^{m+n-2} + \cdots + s_2 x + s_1$$

其中所有的系数 $p_i(i=1,2,\cdots,m)$,$q_i(i=1,2,\cdots,n)$,$s_i(i=1,2,\cdots,m+n-1)$ 均为复数。

二、方法说明

同 14.5 节。只是所有运算为复数运算。

三、子程序语句

SUBROUTINE OCMUL(PR,PI,M,QR,QI,N,SR,SI,L)

四、形参说明

PR,PI——均为双精度实型一维数组,长度均为M,输入参数。多项式 $P_m(x)$ 的系数,

其中 PR 为实部,PI 为虚部。

M——整型变量,输入参数。多项式 $P_m(z)$ 的项数。

QR, QI——均为双精度实型一维数组,长度均为 N,输入参数,多项式 $Q_n(z)$ 的系数,其中 QR 为实部,QI 为虚部。

N——整型变量。输入参数。多项式 $Q_n(z)$ 的项数。

SR,SI——均为双精度实型一维数组,长度均为 L,输出参数。$P_m(z)$ 与 $Q_n(z)$ 乘积多项式的系数,其中 SR 为实部,SI 为虚部,L=M+N−1。

L——整型变量,输入参数。乘积多项式的项数。要求 L=M+N−1。

五、子程序(文件名:OCMUL.FOR)

```
      SUBROUTINE OCMUL(PR,PI,M,QR,QI,N,SR,SI,L)
      DOUBLE PRECISION PR(M),QR(N),SR(L),PI(M),QI(N),SI(L)
      DOUBLE PRECISION U,V
      DO 10 I=1,L
        SR(I)=0.0
        SI(I)=0.0
10    CONTINUE
      DO 50 I=1,M
      DO 50 J=1,N
        K=I+J-1
        CALL PCMUL(PR(I),PI(I),QR(J),QI(J),U,V)
        SR(K)=SR(K)+U
        SI(K)=SI(K)+V
50    CONTINUE
      RETURN
      END
```

六、例

求下列两个多项式

$$P_6(z) = (3 + j2)z^5 + (-1 - j)z^4 + (2 + j)z^3 + (5 - j4)z^2 + (-6 + j3)z + (4 + j2)$$

$$Q_4(z) = (2 + j)z^3 + (-6 - j4)z^2 + (3 + j2)z + (2 + j)$$

的乘积多项式 $S_9(z)$

主程序(文件名:OCMUL0.FOR)为

```
      DOUBLE PRECISION PR(6),PI(6),QR(4),QI(4),SR(9),SI(9)
      DATA PR/4.0,-6.0,5.0,2.0,-1.0,3.0/
      DATA PI/2.0,3.0,-4.0,1.0,-1.0,2.0/
      DATA QR/2.0,3.0,-6.0,2.0/
      DATA QI/1.0,2.0,-4.0,1.0/
      CALL OCMUL(PR,PI,6,QR,QI,4,SR,SI,9)
      DO 10 I=1,9
10    WRITE(*,100) I,SR(I),SI(I)
```

```
100   FORMAT(1X,'S(',I2,')=',D13.6,' +j',D13.6)
      END
```

运行结果为

```
S( 1)=   .600000D+01 +j   .800000D+01
S( 2)= -.700000D+01 +j   .140000D+02
S( 3)= -.260000D+02 +j -.340000D+02
S( 4)=   .800000D+02 +j   .160000D+02
S( 5)=   .580000D+02 +j   .800000D+01
S( 6)=   .900000D+01 +j   .150000D+02
S( 7)=   .100000D+02 +j   .260000D+02
S( 8)= -.110000D+02 +j -.270000D+02
S( 9)=   .400000D+01 +j   .700000D+01
```

七、附注

本子程序需要调用复数乘法子程序 PCMUL(参见 15.1 节)。

14.8 复系数多项式相除

一、功能

求复系数多项式

$$P_m(z)=p_m z^{m-1}+p_{m-1}z^{m-2}+\cdots+p_2 z+p_1$$

被复系数多项式

$$Q_n(z)=q_n z^{n-1}+q_{n-1}z^{n-2}+\cdots+q_2 z+q_1$$

除以后的商多项式 $S_{m-n+1}(z)$ 与余多项式 $R_{n-1}(z)$。

二、方法说明

同 14.6 节。只是所有的运算为复数运算。

三、子程序语句

SUBROUTINE OCDIV(PR,PI,M,QR,QI,N,SR,SI,K,RR,RI,L)

四、形参说明

PR,PI——均为双精度实型一维数组,长度均为 M,输入参数。多项式 $P_m(z)$ 的系数,其中 PR 为实部,PI 为虚部。

M——整型变量,输入参数。多项式 $P_m(z)$ 的项数。

QR,QI——均为双精度实型一维数组,长度均为 N,输入参数。多项式 $Q_n(z)$ 的系数,其中 QR 为实部,QI 为虚部。

N——整型变量,输入参数。多项式 $Q_n(z)$ 的项数。

SR,SI——均为双精度实型一维数组,长度均为 K,输出参数。商多项式的系数,其中 SR 为实部,SI 为虚部。K=M−N+1。

K——整型变量,输入参数。商多项式的项数。要求 K=M−N+1。

RR,RI——均为双精度实型一维数组,长度均为 L,输出参数。余多项式的系数,其中 RR 为实部,RI 为虚部。L=N−1。

L——整型变量，输入参数。余多项式的项数。要求 L=N−1。

五、子程序(文件名:OCDIV.FOR)

```
      SUBROUTINE OCDIV(PR,PI,M,QR,QI,N,SR,SI,K,RR,RI,L)
      DOUBLE PRECISION PR(M),PI(M),QR(N),QI(N)
      DOUBLE PRECISION SR(K),SI(K),RR(L),RI(L)
      DOUBLE PRECISION U,V,D
      DO 10 I=1,K
        SR(I)=0.0
        SI(I)=0.0
10    CONTINUE
      D=QR(N)*QR(N)+QI(N)*QI(N)
      IF (D+1.0.EQ.1.0) RETURN
      LL=M-1
      DO 50 I=K,1,-1
        CALL PCDIV(PR(LL+1),PI(LL+1),QR(N),QI(N),U,V)
        SR(I)=U
        SI(I)=V
        MM=LL
        DO 30 J=1,N-1
          CALL PCMUL(SR(I),SI(I),QR(N-J),QI(N-J),U,V)
          PR(MM)=PR(MM)-U
          PI(MM)=PI(MM)-V
          MM=MM-1
30      CONTINUE
        LL=LL-1
50    CONTINUE
      DO 80 I=1,L
        RR(I)=PR(I)
        RI(I)=PI(I)
80    CONTINUE
      RETURN
      END
```

六、例

求复系数多项式

$P_5(x) = (3-j)x^4 + (6-j5)x^3 + (-3+j4)x^2 + (-5+j4)x + (8+j3)$

被复系数多项式

$$Q_3(x) = (2+j2)x^2 + (-1-j3)x + (1+j2)$$

除以后的商多项式 $S_3(x)$ 与余多项式 $R_2(x)$ 。

主程序(文件名:OCDIV0.FOR)为

```
      DOUBLE PRECISION PR(5),PI(5),QR(3),QI(3)
      DOUBLE PRECISION SR(3),SI(3),RR(2),RI(2)
```

```
      DATA PR/8.0,-5.0,-3.0,6.0,3.0/
      DATA PI/3.0,4.0,4.0,-5.0,-1.0/
      DATA QR/1.0,-1.0,2.0/
      DATA QI/2.0,-3.0,2.0/
      CALL OCDIV(PR,PI,5,QR,QI,3,SR,SI,3,RR,RI,2)
      DO 10 I=1,3
10    WRITE(*,100) I,SR(I),SI(I)
      WRITE(*,*)
      DO 20 I=1,2
20    WRITE(*,200) I,RR(I),RI(I)
100   FORMAT(1X,'S(',I2,')=',D13.6,' +j',D13.6)
200   FORMAT(1X,'R(',I2,')=',D13.6,' +j',D13.6)
      END
```

运行结果为

```
S( 1)=   .262500D+01 +j -.500000D+00
S( 2)=   .125000D+01 +j -.350000D+01
S( 3)=   .500000D+00 +j -.100000D+01

R( 1)=   .437500D+01 +j -.175000D+01
R( 2)= -.912500D+01 +j  .123750D+02
```

七、附注

本子程序需要调用复数乘法子程序 PCMUL(参见 15.1 节)及复数除法子程序 PCDIV(参见 15.2 节)

14.9 函数连分式的计算

一、功能

计算函数连分式

$$\varphi(x) = b_1 + \frac{x - x_1}{b_2 +} \frac{x - x_2}{b_3 +} \cdots + \frac{x - x_{n-1}}{b_n}$$

在指定点处的函数值。

二、方法说明

采用从最后一节开始,逐节往前递推计算的方法。即

$$\begin{cases} u = b_n \\ u = b_k + (x - x_k)/u, \ k = n-1, \cdots, 2, 1 \end{cases}$$

最后结果 u 就是连分式的值。

三、子程序语句

FUNCTION OFPQV(X,B,N,T)

四、形参说明

X——双精度实型一维数组,长度为 N,输入参数。函数连分式中的各结点值 $x_1, x_2,$

$\cdots, x_{n-1}$。其中最后一个元素为任意。

B——双精度实型一维数组，长度为N，输入参数。函数连分式中的常数项 b_1 及部分分母 $b_2, b_3, \cdots, b_n$。

N——整型变量，输入参数。函数连分式的节数。

T——双精度实型变量，输入参数。自变量值。

函数名 OFPQV 返回双精度函数值。

五、子程序(文件名:OFPQV.FOR)

```
      FUNCTION OFPQV(X,B,N,T)
      DOUBLE PRECISION OFPQV,X(N),B(N),T
      OFPQV=B(N)
      DO 10 K=N-1,1,-1
        IF (ABS(OFPQV)+1.0D0.EQ.1.0D0) THEN
          OFPQV=SIGN(1.0D+35,OFPQV)
          OFPQV=OFPQV*SIGN(1.0D0,T-X(K))
        ELSE
          OFPQV=B(K)+(T-X(K))/OFPQV
        END IF
10    CONTINUE
      RETURN
      END
```

六、例

计算并输出函数连分式

$$\varphi(x) = 1 + \frac{x-1}{3+}\,\frac{x-2}{-1+}\,\frac{x-3}{2+}\,\frac{x-4}{5+}\,\frac{x-5}{-8+}\,\frac{x-6}{11}$$

当 $x = 0.0$ 及 $x = 3.5$ 时的函数值。

主程序(文件名:OFPQV0.FOR)为

```
      DOUBLE PRECISION B(7),X(7),T,OFPQV
      DATA B/1.0,3.0,-1.0,2.0,5.0,-8.0,11.0/
      DATA X/1.0,2.0,3.0,4.0,5.0,6.0,0.0/
      T=0.0
      WRITE(*,100) T,OFPQV(X,B,7,T)
      WRITE(*,*)
      T=3.5
      WRITE(*,100) T,OFPQV(X,B,7,T)
100   FORMAT(1X,'T=',F5.2,5X,'Y=',D13.6)
      END
```

运行结果为

```
T= .00      Y= .722174D+00
```

T= 3.50　　Y= .358899D+01

14.10 函数曲线的输出

一、功能

打印输出连续函数的曲线。

二、方法说明

打印函数曲线的格式如下：

(1) 走纸方向为 X 轴方向，Y 方向的宽度为 60。X 轴的位置可由用户指定。

(2) 在打印函数曲线之前，首先打印输出自变量的取值范围，X 方向走纸一行时自变量的增量值，Y 方向一格所表示的函数值增量。

(3) 在 X 轴的左侧印出对应的函数值。

三、子程序语句

SUBROUTINE OFCUR(A,B,H,D,YC,XC,K,N,F)

四、形参说明

A,B——均为实型变量，输入参数。函数曲线的自变量取值范围[A,B]。

H——实型变量，输入参数。X 方向走纸一行所对应的自变量增量，简称步长。

D——实型变量，输入参数。Y 方向一格所对应的函数值增量。

YC——字符型变量，输入参数。打印函数曲线所用的字符。

XC——字符型变量，输入参数。打印 X 轴所用的字符。

K——整型变量，输入参数。X 轴的位置，即 X 轴在打印纸上所占的列号。要求 1≤K≤60。若 K<1，则置 K=1；若 K>60，则置 K=60。

N——整型变量，输入参数。打印函数值的间隔，即每隔 N 步打印一次函数值。

F——实型函数段名，输入参数。用于计算函数值。在主程序中要用外部语句对相应的实函数名进行说明。此函数子程序由用户自编，其语句形式为

FUNCTION F(X)

五、子程序(文件名:OFCUR.FOR)

```
      SUBROUTINE OFCUR(A,B,H,D,YC,XC,K,N,F)
      CHARACTER C(60)
      CHARACTER YC,XC
      WRITE(*,*)
      WRITE(*,10) A,B
10    FORMAT(1X,'BEGIN=',E13.6,10X,'END=',E13.6)
      WRITE(*,20) H
20    FORMAT(1X,'STEP=',E13.6)
      WRITE(*,30) D
30    FORMAT(1X,'SCALE=',E13.6)
      WRITE(*,*)
      IF (K.LT.1) K=1
```

```
      IF (K.GT.60) K=60
      C(K)=XC
      L=0
      X=A
40    IF (X.LE.B) THEN
        Y=F(X)
        YY=Y/D
        IF (YY.GE.0.0) M=YY+0.5
        IF (YY.LT.0.0) THEN
          M=ABS(YY)+0.5
          M=-M
        END IF
        M=K+M
        IF (M.LT.1) M=1
        IF (M.GT.60) M=60
        DO 50 I=1,60
          IF (I.NE.K) C(I)=' '
50      CONTINUE
        IF (M.NE.K) C(M)=YC
        IF (L.EQ.0) THEN
          WRITE(*,60) Y,(C(I),I=1,60)
        ELSE
          WRITE(*,70) (C(I),I=1,60)
        END IF
        L=L+1
        IF (L.EQ.N) L=0
        X=X+H
        GOTO 40
      END IF
60    FORMAT(1X,E13.6,60A1)
70    FORMAT(14X,60A1)
      WRITE(*,*)
      RETURN
      END
```

六、例

(1) 打印函数

$$f_1(x) = \frac{1}{1+25x^2}$$

的曲线。其中 A=−1,B=1,打印步长 H=0.1,函数值增量 D=0.02,X 轴在打印纸的第 1 列(即 K=1),每隔 2 步打印一次函数值(即 N=2),X 轴打印字符为"I",函数曲线的打印字符为"*"。

(2) 打印函数

$$f_2(x) = \sin 2x$$

的曲线。其中 A=0,B=π,打印步长 H=0.1,函数值增量 D=0.04,X 轴在打印纸的第 30 列(即 K=30),每隔 5 步打印一次函数值(即 N=5),X 轴打印字符为“*”,函数曲线的打印字符为“+”。

主程序及计算函数值的子程序(文件名:OFCUR0.FOR)为

```
EXTERNAL F1,F2
CALL OFCUR(-1.0,1.0,0.1,0.02,'*','I',1,2,F1)
WRITE(*,*)
CALL OFCUR(0.0,3.1415926,0.1,0.04,'+','*',30,5,F2)
END

FUNCTION F1(X)
F1=1.0/(1.0+25.0*X*X)
RETURN
END

FUNCTION F2(X)
F2=SIN(2.0*X)
RETURN
END
```

运行结果为

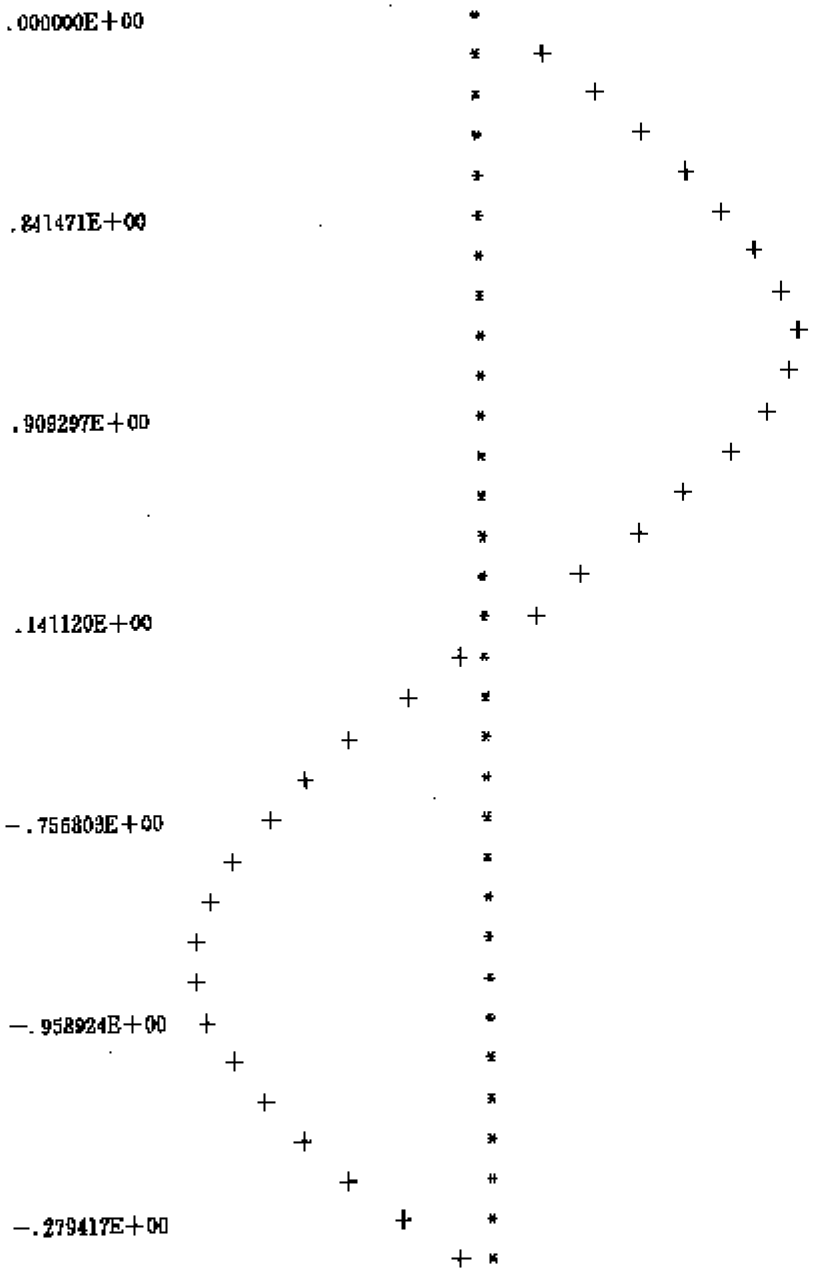
```
BEGIN= -.100000E+01            END=  .100000E+01
STEP=    .100000E+00
SCALE=   .200000E-01

.384615E-01 I *
            I *
.588235E-01I  *
            I  *
.100000E+00 I   *
            I     *
.200000E+00 I         *
            I              *
.500000E+00 I                        *
            I                                       *
.100000E+01 I                                                  *
            I                                       *
.500000E+00 I                        *
            I              *
.200000E+00 I         *
            I     *
.100000E+00 I   *
            I  *
.588235E-01 I  *
            I *
```

BEGIN= .000000E+00 END= .314159E+01

STEP= .100000E+00

SCALE= .400000E−01



第15章 复数运算

15.1 复数乘法

一、功能

计算两个复数的乘积

$$e + jf = (a + jb)(c + jd)$$

其中 $j = \sqrt{-1}$。

二、方法说明

通常，计算两个复数的乘积需要四次实数乘法，本子程序只需要三次乘法，其算法如下。

设

$$e + jf = (a + jb)(c + jd)$$

令

$$p = ac, q = bd, s = (a + b)(c + d)$$

则有

$$e = p - q, f = s - p - q$$

三、子程序语句

SUBROUTINE PCMUL(A,B,C,D,E,F)

四、形参说明

A,B,C,D —均为双精度实型变量，输入参数。分别表示两个复数，A+jB 与 C+jD。

E,F——均为双精度实型变量，输出参数。分别表示两个复数乘积的实部与虚部，即

$$E + jF = (A + jB)(C + jD)$$

五、子程序（文件名：PCMUL.FOR）

```
SUBROUTINE PCMUL(A,B,C,D,E,F)
DOUBLE PRECISION A,B,C,D,E,F
DOUBLE PRECISION P,Q,S
P=A*C
Q=B*D
S=(A+B)*(C+D)
E=P-Q
F=S-P-Q
END
```

六、例

计算并输出下列两个复数的乘积

$$(-1.3+j4.5)(7.6+j3.6)$$

主程序(文件名:PCMULD.FOR)为

```
      DOUBLE PRECISION A,B,C,D,E,F
      A=-1.3
      B=4.5
      C=7.6
      D=3.6
      CALL PCMUL(A,B,C,D,E,F)
      WRITE(*,10) E,F
10    FORMAT(1X,D13.6,'+j',D13.6)
      END
```

运行结果为

```
-.260800D+02+j  .295200D+02
```

15.2 复数除法

一、功能

计算两个复数的商

$$e+jf=(a+jb)/(c+jd)$$

其中 $j=\sqrt{-1}$。

二、方法说明

本子程序采用如下算法:

设

$$e+jf=(a+jb)/(c+jd)$$

令

$$p=ac, q=-bd, s=(a+b)(c-d), w=c^2+d^2$$

则有

$$e=(p-q)/w, f=(s-p-q)/w$$

三、子程序语句

SUBROUTINE PCDIV(A,B,C,D,E,F)

四、形参说明

A,B,C,D——均为双精度实型变量,输入参数,分别表示两个复数:A+jB 与 C+jD。

E,F——均为双精度实型变量,输出参数。分别表示两个复数商的实部与虚部,即

$$E+jF=(A+jB)/(C+jD)$$

五、子程序(文件名:PCDIV.FOR)

```
      SUBROUTINE PCDIV(A,B,C,D,E,F)
      DOUBLE PRECISION A,B,C,D,E,F
      DOUBLE PRECISION P,Q,S,W
      P=A*C
```

```
      Q=-B*D
      S=(A+B)*(C-D)
      W=C*C+D*D
      IF (W+1.0.EQ.1.0) THEN
        E=SIGN(1.0D+35,A)
        F=SIGN(1.0D+35,B)
      ELSE
        E=(P-Q)/W
        F=(S-P-Q)/W
      END IF
      END
```

六、例

计算并输出下列两个复数的商

$$(-1.3+j4.5)/(7.6-j3.6)$$

主程序(文件名:PCDIV0.FOR)为

```
      DOUBLE PRECISION A,B,C,D,E,F
      A=-1.3
      B=4.5
      C=7.6
      D=-3.6
      CALL PCDIV(A,B,C,D,E,F)
      WRITE(*,10) E,F
10    FORMAT(1X,D13.6,'+j',D13.6)
      END
```

运行结果为

```
 -.368778D+00+j  .417421D+00
```

15.3 复数乘幂

一、功能

计算复变量的整数次幂,即

$$(x+jy)^n$$

其中 $j=\sqrt{-1}$,n 为整数。

二、方法说明

设给定复数为

$$z=x+jy$$

化为

$$z=r(\cos\theta+j\sin\theta)$$

其中

$$r = \sqrt{x^2 + y^2}, \theta = \mathrm{Arctg}\,\frac{y}{x}$$

则

$$\begin{aligned} z^n &= (x + jy)^n \\ &= r^n(\cos n\theta + j\sin n\theta) \\ &= u + jv \end{aligned}$$

其中

$$u = r^n\cos n\theta, v = r^n\sin n\theta$$

三、子程序语句

SUBROUTINE PPOWR(X,Y,N,U,V)

四、形参说明

X,Y——均为双精度实型变量，输入参数。表示复数X+jY。

N——整型变量，输入参数。幂的次数。

U,V——均为双精度实型变量，输出参数。分别表示复数(X+jY)N次幂的实部与虚部，即

$$U + jV = (X + jY)^N$$

五、子程序(文件名:PPOWR.FOR)

```
      SUBROUTINE PPOWR(X,Y,N,U,V)
      DOUBLE PRECISION X,Y,U,V
      DOUBLE PRECISION R,Q
      Q=ATAN2(Y,X)
      R=SQRT(X*X+Y*Y)
      IF (R+1.0.NE.1.0) THEN
        R=N*LOG(R)
        R=EXP(R)
      END IF
      U=R*COS(N*Q)
      V=R*SIN(N*Q)
      END
```

六、例

计算并输出复数(2.0+j2.0)的−3,0,3次幂。

主程序(文件名:PPOWR0.FOR)为

```
      DOUBLE PRECISION X,Y,U,V
      X=2.0
      Y=2.0
      DO 100 N=-3,3,3
        CALL PPOWR(X,Y,N,U,V)
        WRITE(*,10) N,U,V
100   CONTINUE
10    FORMAT(1X,'N=',I2,5X,D13.6,'+j',D13.6)
```

```
      END
```

运行结果为

```
N=-3        -.312500D-01+j -.312500D-01
N= 0         .100000D+01+j  .000000D+00
N= 3        -.160000D+02+j  .160000D+02
```

七、附注

形参 U 与 X,V 与 Y 对应的实参分别可以是同名变量.

15.4 复数的 N 次方根

一、功能

计算复变量的 n 次方根,即

$$(x+\mathrm{j}y)^{\frac{1}{n}}$$

其中 $\mathrm{j}=\sqrt{-1}$, n 为正整数.

二、方法说明

设给定复数为

$$z=x+\mathrm{j}y$$

化为

$$z=r(\cos\theta+\mathrm{j}\sin\theta)$$

其中

$$r=\sqrt{x^2+y^2},\theta=\mathrm{Arctg}\,\frac{y}{x}$$

则

$$\begin{aligned}z^{\frac{1}{n}}&=(x+\mathrm{j}y)^{\frac{1}{n}}\\&=r^{\frac{1}{n}}\left[\cos\frac{2(k-1)\pi+\theta}{n}+\mathrm{j}\sin\frac{2(k-1)\pi+\theta}{n}\right]\\&=u_k+\mathrm{j}v_k,\ k=1,2,\cdots,n\end{aligned}$$

其中

$$u_k=r^{\frac{1}{n}}\cos\frac{2(k-1)\pi+\theta}{n},v_k=r^{\frac{1}{n}}\sin\frac{2(k-1)\pi+\theta}{n}$$

三、子程序语句

SUBROUTINE PNTRT(X,Y,N,U,V)

四、形参说明

X,Y——均为双精度实型变量,输入参数。表示复数 X+jY。

N——整型变量,输入参数。表示对 X+jY 求 N 次方根。要求 N 为正整数,当 N<1 时,输出的 N 个 N 次方根均为 0。

U,V——均为双精度实型一维数组,长度均为 N,输出参数。存放(X+jY)的 N 个 N

次方根。

五、子程序(文件名:PNTRT.FOR)

```
      SUBROUTINE PNTRT(X,Y,N,U,V)
      DOUBLE PRECISION X,Y,U(N),V(N)
      DOUBLE PRECISION R,Q,T
      IF (N.LT.1) THEN
        DO 10 K=1,N
          U(K)=0.0
          V(K)=0.0
10      CONTINUE
        RETURN
      END IF
      Q=ATAN2(Y,X)
      R=SQRT(X*X+Y*Y)
      IF (R+1.0.NE.1.0) THEN
        R=(1.0/N)*LOG(R)
        R=EXP(R)
      END IF
      DO 20 K=1,N
        T=(2.0*(K-1.0)*3.1415926+Q)/(1.0*N)
        U(K)=R*COS(T)
        V(K)=R*SIN(T)
20    CONTINUE
      END
```

六、例

计算并输出复数 1+j 的 5 次方根。

主程序(文件名:PNTRT0.FOR)为

```
      DOUBLE PRECISION X,Y,U(5),V(5)
      X=1.0
      Y=1.0
      CALL PNTRT(X,Y,5,U,V)
      DO 10 K=1,5
10    WRITE(*,100) U(K),V(K)
100   FORMAT(1X,5X,D13.6,'+j',D13.6)
      END
```

运行结果为

```
.105858D+01+j   .167662D+00
.167662D+00+j   .105858D+01
-.954957D+00+j  .486575D+00
-.757858D+00+j -.757858D+00
.486575D+00+j  -.954957D+00
```

15.5 复数指数

一、功能

计算复变量的指数，即

$$e^{x+jy}$$

其中 $j=\sqrt{-1}$。

二、方法说明

设复数为

$$z = x + jy$$

则

$$\begin{aligned} e^z &= e^{x+jy} = e^x(\cos y + j\sin y) \\ &= u + jv \end{aligned}$$

其中

$$u = e^x\cos y, v = e^x\sin y$$

三、子程序语句

SUBROUTINE PCEXP(X,Y,U,V)

四、形参说明

X,Y——均为双精度实型变量，输入参数。表示复数 X+jY。

U,V——均为双精度实型变量，输出参数。存放复数(X+jY)指数的实部与虚部，即

$$U+jV=e^{X+jY}$$

五、子程序(文件名:PCEXP.FOR)

```
      SUBROUTINE PCEXP(X,Y,U,V)
      DOUBLE PRECISION X,Y,U,V
      DOUBLE PRECISION P
      P=EXP(X)
      U=P*COS(Y)
      V=P*SIN(Y)
      END
```

六、例

计算并输出复数 $z=1.0+j4.0$ 的指数 e^z。

主程序(文件名:PCEXP0.FOR)为

```
      DOUBLE PRECISION X,Y,U,V
      X=1.0
      Y=4.0
      CALL PCEXP(X,Y,U,V)
      WRITE(*,10) U,V
10    FORMAT(1X,D13.6,'+j',D13.6)
      END
```

运行结果为

−.177679D+01+j −.205720D+01

七、附注

形参 U 与 X、V 与 Y 对应的实参分别可以是同名变量。

15.6 复数对数

一、功能

计算复变量的自然对数，即

$$\ln(x+jy)$$

其中 $j=\sqrt{-1}$。

二、方法说明

设给定的复数为

$$z=x+jy$$

则

$$\ln z=\ln(x+jy)=\ln\sqrt{x^2+y^2}+j\mathrm{Arctg}\frac{y}{x}$$
$$=u+jv$$

其中

$$u=\ln\sqrt{x^2+y^2},\ v=\mathrm{Arctg}\frac{y}{x}$$

三、子程序语句

SUBROUTINE PCLOG(X,Y,U,V)

四、形参说明

X,Y——均为双精度实型变量，输入参数。表示复数 X+jY。

U,V——均为双精度实型变量，输出参数。存放复数(X+jY)自然对数的实部与虚部，即

$$U+jV=\ln(X+jY)$$

五、子程序（文件名：PCLOG.FOR）

```
SUBROUTINE PCLOG(X,Y,U,V)
DOUBLE PRECISION X,Y,U,V
DOUBLE PRECISION P
P=LOG(SQRT(X*X+Y*Y))
V=ATAN2(Y,X)
U=P
END
```

六、例

计算并输出复数 1.0+j4.0 的自然对数。

主程序（文件名：PCLOG0.FOR）为

```
      DOUBLE PRECISION X,Y,U,V
      X=1.0
      Y=4.0
      CALL PCLOG(X,Y,U,V)
      WRITE(*,10) U,V
10    FORMAT(1X,D13.6,'+j',D13.6)
      END
```

运行结果为

.141661D+01+j .132582D+01

七、附注

形参 U 与 X,V 与 Y 对应的实参分别可以是同名变量。

15.7 复数正弦

一、功能

计算复变量的正弦值,即

$$\sin(x+\mathrm{j}y)$$

其中 $\mathrm{j}=\sqrt{-1}$。

二、方法说明

设复数为

$$z=x+\mathrm{j}y$$

则

$$\begin{aligned}\sin z&=\sin(x+\mathrm{j}y)=\sin x\cos\mathrm{j}y+\cos x\sin\mathrm{j}y\\&=\sin x\frac{e^{y}+e^{-y}}{2}+\mathrm{j}\cos x\frac{e^{y}-e^{-y}}{2}\\&=u+\mathrm{j}v\end{aligned}$$

其中

$$u=\sin x\frac{e^{y}+e^{-y}}{2},\quad v=\cos x\frac{e^{y}-e^{-y}}{2}$$

三、子程序语句

SUBROUTINE PCSIN(X,Y,U,V)

四、形参说明

X,Y——均为双精度实型变量,输入参数。表示复数 X+jY。

U,V——均为双精度实型变量,输出参数。存放复数(X+jY)的正弦值的实部与虚部,即

$$U+\mathrm{j}V=\sin(X+\mathrm{j}Y)$$

五、子程序(文件名:PCSIN.FOR)

```
      SUBROUTINE PCSIN(X,Y,U,V)
      DOUBLE PRECISION X,Y,U,V
```

```
      DOUBLE PRECISION P,Q
      P=EXP(Y)
      Q=EXP(-Y)
      V=COS(X)*(P-Q)/2.0
      U=SIN(X)*(P+Q)/2.0
      END
```

六、例

计算并输出复数 1.0+j4.0 的正弦值。

主程序(文件名:PCSIN0.FOR)为

```
      DOUBLE PRECISION X,Y
      X=1.0
      Y=4.0
      CALL PCSIN(X,Y,X,Y)
      WRITE(*,10) X,Y
10    FORMAT(1X,D13.6,'+j',D13.6)
      END
```

运行结果为

```
.229791D+02+j  .147448D+02
```

七、附注

形参 U 与 X,V 与 Y 对应的实参分别可以是同名变量。

15.8 复数余弦

一、功能

计算复变量的余弦值,即

$$\cos(x+\mathrm{j}y)$$

其中 $\mathrm{j}=\sqrt{-1}$。

二、方法说明

设给定的复数为

$$z=x+\mathrm{j}y$$

则

$$\begin{aligned}\cos z&=\cos(x+\mathrm{j}y)=\cos x\cos\mathrm{j}y-\sin x\sin\mathrm{j}y\\&=\cos x\frac{e^{y}+e^{-y}}{2}-\mathrm{j}\sin x\frac{e^{y}-e^{-y}}{2}\\&=u+\mathrm{j}v\end{aligned}$$

其中

$$u=\cos x\frac{e^{y}+e^{-y}}{2},v=-\sin x\frac{e^{y}-e^{-y}}{2}$$

三、子程序语句

SUBROUTINE PCCOS(X,Y,U,V)

四、形参说明

X,Y——均为双精度实型变量,输入参数。表示复数 X+jY。

U,V——均为双精度实型变量,输出参数。存放复数(X+jY)余弦值的实部与虚部,即

$$U+jV=\cos(X+jY)$$

五、子程序(文件名:PCCOS.FOR)

```
      SUBROUTINE PCCOS(X,Y,U,V)
      DOUBLE PRECISION X,Y,U,V
      DOUBLE PRECISION P,Q
      P=EXP(Y)
      Q=EXP(-Y)
      V=-SIN(X)*(P-Q)/2.0
      U=COS(X)*(P+Q)/2.0
      END
```

六、例

计算并输出复数 1.0+j4.0 的余弦值。

主程序(文件名:PCCOS0.FOR)为

```
      DOUBLE PRECISION X,Y
      X=1.0
      Y=4.0
      CALL PCCOS(X,Y,X,Y)
      WRITE(*,10) X,Y
10    FORMAT(1X,D13.6,'+j',D13.6)
      END
```

运行结果为

```
.147547D+02+j -.229637D+02
```

七、附注

形参 U 与 X,V 与 Y 对应的实参分别可以是同名变量。

15.9 复数作图

一、功能

打印输出给定的复数点集。

二、方法说明

设给定 n 个复数

$$x_k+jy_k,\ k=1,2,\cdots,n$$

其中 $j=\sqrt{-1}$。

在打印纸上建立一个直角坐标系XOY，其X值与Y值的范围均为[－35，35]，即所有复数点均打印在宽为71列、长为71行的打印纸上，坐标原点在其中心。为了充分利用71行，71列的打印幅面，在本子程序中，选取所有复数点中X绝对值最大的点打印在第1列(负值)或71列(正值)上，Y绝对值最大的点打印在第1行(正值)或第71行(负值)上，其余各值将按同样比例进行放大或缩小，并在子程序中返回X值的比例与Y值的比例。

三、子程序语句

SUBROUTINE PPLOT(N,A,XC,YC,C,XD,YD)

四、形参说明

N——整型变量，输入参数。给定的复数点个数。

A——实型二维数组，体积为N×2，输入参数。存放各复数点，其中A(k,1)为第k个复数的实部(即X值)，A(k,2)为第k个复数的虚部(即Y值)，k＝1,2,…,N。

XC——字符型变量，输入参数。打印X轴所用的字符。

YC——字符型变量，输入参数。打印Y轴所用的字符。

C——字符型变量，输入参数。打印复数点所用的字符。

XD——实型变量，输出参数。打印复数点时X值的比例系数。该比例系数在子程序中也打印输出。

YD——实型变量，输出参数。打印复数点时Y值的比例系数。该比例系数在子程序中也打印输出。

五、子程序(文件名:PPLOT.FOR)

```
      SUBROUTINE PPLOT(N,A,XC,YC,C,XD,YD)
      DIMENSION A(N,2)
      CHARACTER XY(72,72)
      CHARACTER XC,YC,C
      DO 10 I=1,72
      DO 10 J=1,72
10    XY(I,J)=' '
      XY(1,36)='Y'
      DO 20 I=2,72
20    XY(I,36)=YC
      XY(37,72)='X'
      DO 30 J=1,71
30    XY(37,J)=XC
      XD=0.0
      YD=0.0
      DO 40 K=1,N
        IF (ABS(A(K,1)).GT.XD) XD=ABS(A(K,1))
        IF (ABS(A(K,2)).GT.YD) YD=ABS(A(K,2))
```

```
40    CONTINUE
      XD=35/XD
      YD=35/YD
      DO 50 K=1,N
        I=37.0-YD*A(K,2)+0.5
        J=XD*A(K,1)+36.0+0.5
        XY(I,J)=C
50    CONTINUE
      WRITE(*,*)
      DO 60 I=1,72
        WRITE(*,70) (XY(I,J),J=1,72)
60    CONTINUE
70    FORMAT(1X,72A1)
      WRITE(*,*)
      WRITE(*,80) XD
80    FORMAT(1X,'X-SCALE IS: ',E15.6)
      WRITE(*,90) YD
90    FORMAT(1X,'Y-SCALE IS: ',E15.6)
      WRITE(*,*)
      RETURN
      END
```

六、例

打印输出下列 50 个复数点：

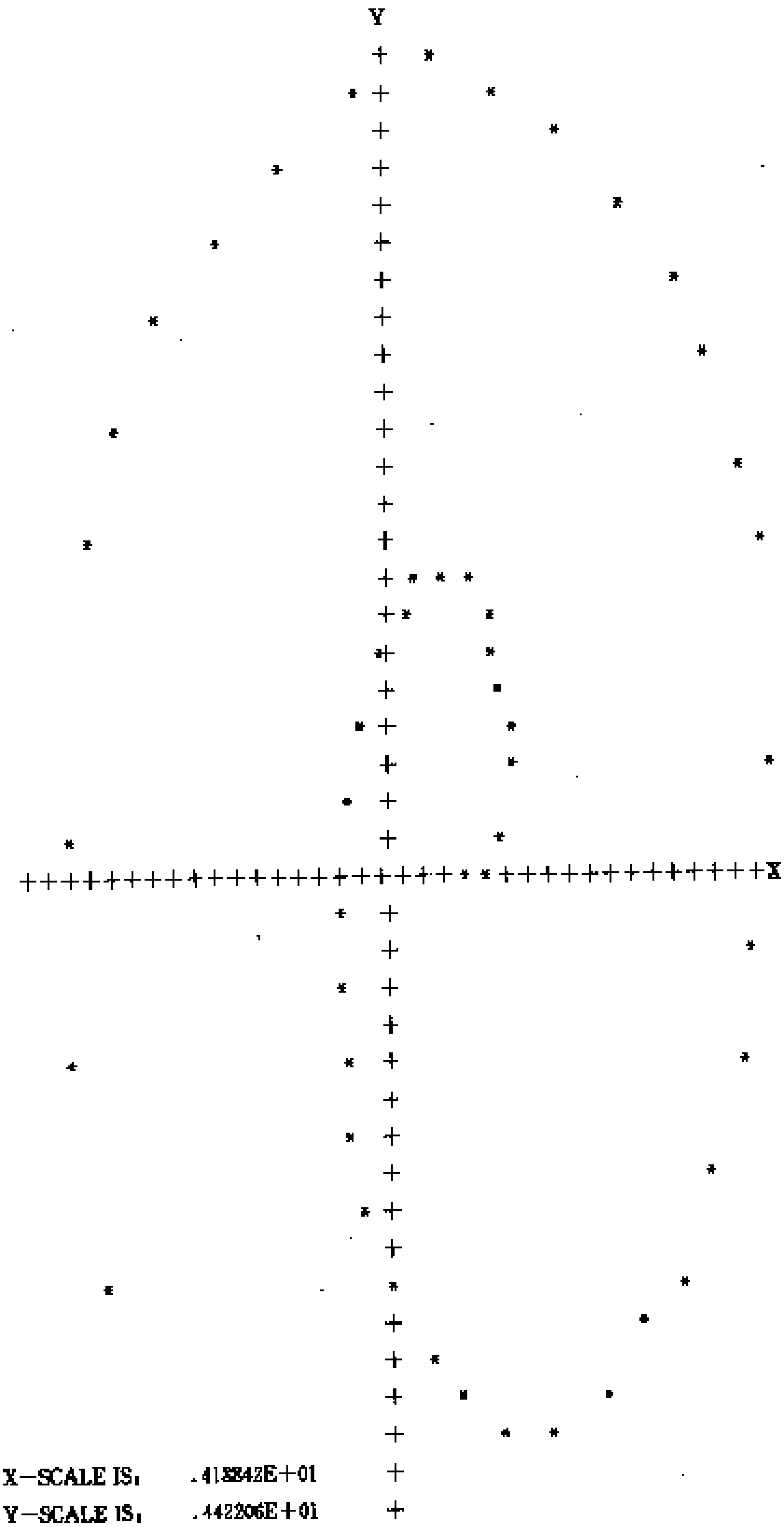
$$2.0+\varphi e^{i\varphi}$$

其中 $\varphi=0.2(k-1)$，$k=1,2,\cdots,50$。X 轴与 Y 轴用字符'+'，复数点用字符'*'。

主程序(文件名:PPLOT0.FOR)为

```
      DIMENSION A(50,2)
      CHARACTER XC,YC,C
      XC='+'
      YC='+'
      C='*'
      N=50
      DO 10 K=1,N
        FI=0.2*(K-1)
        A(K,1)=2.0+FI*COS(FI)
        A(K,2)=FI*SIN(FI)
10    CONTINUE
      CALL PPLOT(N,A,XC,YC,C,XD,YD)
      END
```

运行结果为



X-SCALE IS, .418842E+01
Y-SCALE IS, .442206E+01

附录1　FORTRAN77 库管理程序的使用

如果希望将 FORTRAN 子程序集变成 FORTRAN 程序库，则应首先将所有的子程序进行编译，形成扩展名为.OBJ 的文件；然后使用库管理程序 LIB.EXE，将所有扩展名为.OBJ 的文件处理成一个 FORTRAN 程序库。

LIB.EXE 的使用方法如下。

首先键入命令

LIB ↙

则提示如下：

library file：此时应回答欲处理的库名(.LIB 可省略)。此时，如果回答的库名是新库，则又提示：

Create?　此时按回车键"↙"即可。

接着又提示如下：

Operations：　此时将命令字以及随后的模块名或目标文件名列出(.OBJ 可省略)。如果直接按回车键，则表示无修改。

list file：　回答交叉列表文件名。一般直接按回车键。

out put library：　输入处理后的库名。若库名不变，则直接按回车键即可。

下面介绍几个命令字的使用。

+：　加号后紧跟一个目标文件名，就把这个目标文件加到库中作为最后一个模块。

-：　减号后紧跟一个模块名，就把库文件中的这个模块删除，并将删除模块后剩下的空间消除，这样就使库文件不会无限制地增大。

*：　星号后紧跟一个模块名，就把该模块从库文件中提取出来建立一个单独的目标文件(扩展名为.OBJ)，但这个模块将仍保留在库中。

;：　分号后紧跟回车，使后面的提示符后的回答都选为缺省值，这样就不必逐个回答各个提问，从而节省操作时间。

&：　使用 & 可以实现续行；这个命令字只用于回答"Operation："这一提问。

Ctrl-C：　中断 lib 过程。

附录 2　关于《FORTRAN 常用算法程序集》(第二版)配套软盘的说明

本书有两种类型的配套软盘。

第一种类型的配套软盘中包括了本书中的所有 FORTRAN77 源程序，其文件名与书中的完全相同，它们按章存放在各子目录中，子目录名分别为 CH1，CH2，…，CH15。

第二种类型的配套软盘中只有一个库文件，文件名为 XUF.LIB。在这个库文件中，包括了本书中所有的子程序模块。需要调用本书中的子程序时，与调用 FORTRAN 标准函数一样方便，只要在链接时给出库文件名 XUF 即可。

如果读者需要上述的配套软盘，可以直接与清华大学出版社软件部联系。

参考文献

[1] 徐士良.计算机常用算法.第二版.北京:清华大学出版社,1995

[2] 李庆扬等.数值分析.武汉:华中理工大学出版社,1986

[3] 徐士良.C常用算法程序集.北京:清华大学出版社,1994

[4] [美]W.H.普雷斯等著.王璞等译.数值方法大全——科学计算的艺术.兰州:兰州大学出版社,1991

[5] 邹海.最优设计中的新计算法.北京:新时代出版社,1982

[6] 蔡大用.数值代数.北京:清华大学出版社,1987

[7] [美]STEVEN A.TRETTER著.王平孙译.离散时间信号处理系统.北京:高等教育出版社,1982

[8] 许志宏等.TQ-16机FORTRAN语言常用算法程序集.北京:化学工业出版社,1982

[9] 刘德贵等.FORTRAN算法汇编.北京:国防工业出版社,1980

[10] 《数学手册》编写组.数学手册.北京:高等教育出版社,1979

[11] 陆大絟.随机过程及其应用.北京:清华大学出版社,1986