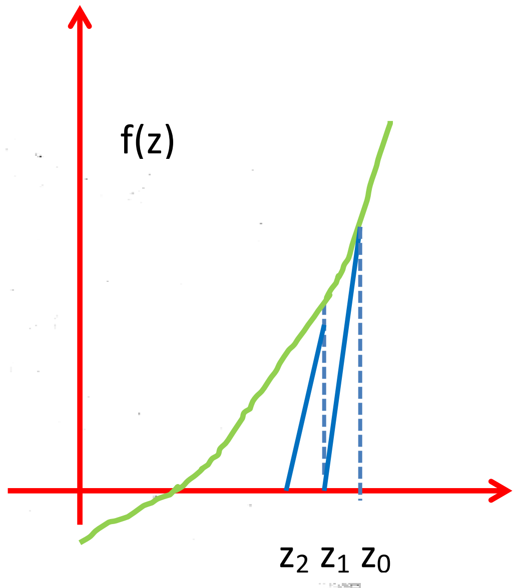
第四章：非线性方程和方程组的求解

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

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

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。

实现此功能的子程序及功能如下：

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，…，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)

六、例

求实系数代数方程

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

的全部根。

```
PROGRAM EXAMPLES05

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
```

以上是方程的 6 个根。

```
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
```

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

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

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 $\sum_{j=1}^{n} z_j = 1$,工作失败。

JS——整型一维数组,长度为 N。本子程序的工作数组。

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

需要,主程序 EXAMPLES04，　　子程序：　FS, DNETN, AGAUS

```
PROGRAM EXAMPLES04
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(*,30) L
 DO 10 I=1,3
10 WRITE(*,20) I,X(I)
20 FORMAT(1X,'X(',I2,' )=',D15.6)
30 FORMAT(1X,'L=',I4)
 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

L=95

X（1）=0.785

X（2）=0.49

X（3）=0.369

L=95 说明迭代了 100-95=5 次就满足精度要求了。

```
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
```

```
SUBROUTINE AGAUS(A,B,N,X,L,JS)      ！全选主元高斯消去法。
    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
```