第二章　矩阵运算

矩阵的运算主要有：矩阵相乘，矩阵求逆，**矩阵分解（三角分解，奇异值分解，QR 分解）**
矩阵求逆可以采用第一章的高斯消去法计算，因为求 AX=B 实际上是 $X=A^{-1}B$，所以可以采用化为三角阵进行矩阵求逆。

三角分解： A=LU, L 是下三角矩阵，对角线元素为 1，U 是上三角矩阵

奇异值分解： 对于一个 m×n 的矩阵（m≥n）

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

这里 U 是 m×m 正交矩阵，　V 是 n×n 的正交矩阵，Σ是对角矩阵

2.1 QR 分解：

我们这一章介绍矩阵的一种分解方式 QR 分解。

A=QR

Q 是一个正交矩阵，R 是一个上三角矩阵

QR 分解法是目前求"一般矩阵"全部特征值的最有效并广泛应用的方法。

一般矩阵先经过正交相似变换成为上海森堡矩阵，

$$
H = \begin{bmatrix}
h_{11} & h_{12} & \cdots & h_{1n-1} & h_{1n} \\
h_{21} & h_{22} & \cdots & h_{2n-1} & h_{2n} \\
 & h_{32} & h_{33} & \cdots & h_{3n} \\
 & & \ddots & \ddots & \vdots \\
 & & & h_{nn-1} & h_{nn}
\end{bmatrix}
$$

然后再应用 QR 方法求特征值和特征向量。

QR方法的基本思想是利用矩阵的QR分解通过迭代格式

$$\begin{cases} A_k = Q_k R_k \\ A_{k+1} = R_k Q_k \end{cases} \quad (k = 1, 2, \cdots).$$

将$A = A_1$化成相似的上三角阵（或分块上三角阵），从而求出矩阵A的全部特征值与特征向量。

该算法生成一个矩阵序列{A_k}，如果矩阵是对称的，Q_k 的乘积收敛到矩阵 A 的特征向量，A 收敛到对角矩阵。

它是将矩阵分解成一个正规**正交矩阵 Q** 与**上三角形矩阵 R**，所以称为 **QR 分解法**。

对一个矩阵的各个列向量逐一进行相应的豪斯霍尔德变换，可以将这个矩阵变换为上海森伯矩阵、上三角矩阵等形式。后者就是 QR 分解的豪斯霍尔德算法。

有线性变换将向量 ξ 映射成与向量 v 正交的 n-1 维子空间对称的向量η，且有

$$\eta = (\mathrm{I} - \frac{2}{\langle \mathrm{v} \cdot \mathrm{v} \rangle} \mathrm{v}\mathrm{v}^{\mathrm{T}})\xi = H\xi$$

这种变换叫豪斯霍尔德（Householder）变换。

矩阵的 QR 分解有重要的应用，除了可以计算特征值以外还是求解线性代数方程组的可靠方法。

$$Ax=b$$

A 是 m×n 的矩阵，

如果 m=n 且 A=QR，自然等价于

$$Rx=Q^Tb$$

R 是三角矩阵，该方程自然容易求解

如果 m>n，则上式即为最小二乘问题

$$Ax\approx b$$

或者更加确切的写为

$$\min_x \|r\|_2^2 = \min_x \|b-Ax\|_2^2 = \arg\min(\|b-Ax\|_2^2)$$

通过 QR 分解，方程两边同乘以 Q^T，则最小二乘的问题等价为函数的极小化

$$\|b-Ax\|_2 = \left\|b - Q\begin{pmatrix} R \\ 0 \end{pmatrix} x\right\|_2 = \left\|Q^Tb - \begin{pmatrix} R \\ 0 \end{pmatrix} x\right\|_2$$

因此 QR 分解可以有效求解最小二乘问题。

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。

按照如上的结论，我们编写 QR 分解的算法，步骤如下：

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

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

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：

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

（3）用 Q_k 左乘 A，即

$$Q_k A \Rightarrow A$$

其计算公式如下：

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

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。

```
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
```

```
PROGRAM EXAMPLE02
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 PROGRAM
```

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

正交矩阵

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

上三角矩阵

2.2 求广义逆的奇异值分解法

1. **功能**:利用奇异值分解求一般 m*n 阶实矩阵 A 的广义逆 A^+

2. 奇异值分解的**方法说明**：

设 m*n 矩阵 A 的实矩阵，则存在一个 m*m 阶的列正交矩阵 U 和 n*n 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

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

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

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

其中

$$\widetilde{U}=U_1U_2\cdots U_k, k=\min\{n,m-1\}$$

$$\widetilde{V}=V_1V_2\cdots V_l, l=\min\{m,n-2\}$$

$\tilde{U}$ 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 $\tilde{V}$ 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：

$$I-\rho V_j V_j^T$$

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

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

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

$$B' = U^T_{p-1,p}\cdots U^T_{23}U^T_{12}BV_{12}V_{23}\cdots V_{m-1,m}$$

其中变换 $U^T_{j,j+1}$ 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：

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

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

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

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

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 。

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

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。

3. 利用奇异值分解求解一般实矩阵 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。

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

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。

例子：

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

```
PROGRAM BGINV0
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 PROGRAM
```

运行结果：

```
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-05   .110000D+02
 .160000D+02   .170000D+02   .799999D+01   .900000D+01   .130000D+02
 .200000D+01   .400000D+01   .300000D+01   .400000D+01   .600000D+01
```

```
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
```

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(I)=CS*E(I)
        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
```