A Note on Positive Semi-definiteness of Some Non-Pearsonian Correlation Matrices

SK Mishra
Department of Economics
North-Eastern Hill University
Shillong, Meghalaya (India)
mishrasknehu@yahoo.com

I. Introduction: A correlation matrix, $\Re$, is a real and symmetric $m \times m$ matrix such that $-1 \le r_{ij} \in \Re \le 1;\ i, j = 1,2,\ \ldots\ , m.$ Moreover, $r_{ii} = 1.$ The Pearsonian (or the product moment) correlation coefficient, e.g. r_{12} (between two variates, say x_1 and x_2, each in n observations), is given by the formula:

$$r(x_1, x_2) = \text{cov}(x_1, x_2) / \sqrt{\text{var}(x_1) \cdot \text{var}(x_2)} \quad \ldots \quad (1)$$

where, $\bar{x}_a = \frac{1}{n}\sum_{k=1}^{n} x_{ka};\ \text{cov}(x_1, x_2) = \frac{1}{n}\sum_{k=1}^{n} x_{k1}x_{k2} - \bar{x}_1^2\bar{x}_2^2$ and $\text{var}(x_a) = \text{cov}(x_a, x_a);\ a = 1, 2.$

A little of algebra also gives us the identity:

$$r(x_1, x_2) = (1/4)\left[\text{var}(x_1 + x_2) - \text{var}(x_1 - x_2)\right] \Big/ \sqrt{\text{var}(x_1) \cdot \text{var}(x_2)} \quad \ldots \quad (2).$$

The Pearsonian correlation matrix is necessarily a positive semi-definite matrix (meaning that all its eigenvalues are non-negative) since it is the quadratic form of a real matrix, $X(n, m).$ It also implies that if $\Re$ is not a semi-positive matrix, then $X(n, m)$ is not a real matrix.

II. Robust Measures of Correlation: The Pearsonian coefficient of correlation as a measure of association between two variates is highly prone to the deleterious effects of outlier observations (data). Statisticians have proposed a number of formulas, other than the one that obtains Pearson's coefficient of correlation, that are considered to be less affected by errors of observation, perturbation or presence of outliers in the data. Some of them transform the variables, say x_1 and x_2, into $z_1 = \phi_1(x_1)$ and $z_2 = \phi_2(x_2)$, where $\phi_a(x_a)$ is a linear (or nonlinear) monotonic (order-preserving) rule of transformation or mapping of x_a to z_a. Then, $r(z_1, z_2)$ is obtained by the appropriate formula and it is considered as a robust measure of $r(x_1, x_2)$. Some others use different measures of central tendency, dispersion and co-variation, such as median for mean, mean deviation for standard deviation and so on. In what follows, we present a few formulas of obtaining different types of correlation efficient.

II.1. Spearman's Rank Correlation Coefficient: If x_1 and x_2 are two variables, both in n observations, and $z_1 = \mathbb{R}(x_1)$ and $z_2 = \mathbb{R}(x_2)$ are their rank numerals with $\mathbb{R}(.)$ as the rank-ordering rule, then the Pearson's formula applied on (z_1, z_2) obtains the Spearman's correlation coefficient (Spearman, 1904). There is a simpler (but less general) formula that obtains rank correlation coefficient, given as:

$$\rho(x_1, x_2) = r(z_1, z_2) = 1 - 6\sum_{k=1}^{n}(z_{k1} - z_{k2})^2 / [n(n^2 - 1)] \quad \ldots \quad (3)$$

II.2. Signum Correlation Coefficient: Let c_1 and c_2 be the measures of central tendency or location (such as arithmetic mean or median) of x_1 and x_2 respectively. We transform them to $z_{ka} = (x_{ka} - c_a)/|x_{ka} - c_a|$ if $|x_{ka} - c_a| > 0,\ else\ z_{ka} = 1.$ Then, $r(z_1, z_2)$ is the signum correlation coefficient (Blomqvist, 1950; Shevlyakov, 1997). Due to the special nature of transformation, we have

$$r(z_1, z_2) \cong \mathrm{cov}(z_1, z_2) = (1/n)\sum_{i=1}^{n} z_{i1} z_{i2} \qquad \ldots \qquad (4)$$

In this study we will use median as a measure of central tendency to obtain signum correlation coefficients.

II.3. Kendall's Tau: If x_1 and x_2 are two variables, both in n observations, and $z_1 = \mathbb{R}(x_1)$ and $z_2 = \mathbb{R}(x_2)$ are their rank numerals with $\mathbb{R}(.)$ as the rank-ordering rule, we define

$$c_k = 1\ iff\ (z_{k1} > z_{l1}\ and\ z_{k2} > z_{l2})\ else\ c_k = 0;\ k, l = 1, 2, \ldots, n \qquad \ldots \qquad (5)$$

Then $r(z_1, z_2) = -1 + 4\sum_{k=1}^{n} c_k / (n^2 - n)$ is a measure of association called Kendall's Tau.

II.4. Bradley's Absolute Correlation Coefficient: Bradley (1985) showed that if $(u_k, v_k);\ k = 1, n$ are n pairs of values such that the variables u and v have the same median = 0 and the same mean deviation (from median) or $(1/n)\sum_{k=1}^{n}|u_k| = (1/n)\sum_{k=1}^{n}|v_k| = d \neq 0$, both of which conditions may be met by any pair of variables when suitably transformed, then the absolute correlation may be defined as

$$\rho(u, v) = \sum_{k=1}^{n}\left(|u_k + v_k| - |u_k - v_k|\right) \Big/ \sum_{k=1}^{n}\left(|u_k| + |v_k|\right). \qquad \ldots \qquad (6)$$

II.5. Shevlyakov Correlation Coefficient: Hampel et al. (1986) defined the median of absolute deviations (from median) as a measure of scale, $s_H(x_a) = \underset{k}{median}\,|x_{ka} - \underset{k}{median}(x_{ka})|;\ a = 1, 2$ which is a very robust measure of deviation, and using this measure, Shevlyakov (1997) defined median correlation,

$$r_{med} = \left[med^2|u| - med^2|v|\right] \Big/ \left[med^2|u| + med^2|v|\right] \qquad \ldots \qquad (7)$$

where u and v are given as $u_k = (x_{k1} - med(x_1))/s_H(x_1) + (x_{k2} - med(x_2))/s_H(x_2)$ and $v_k = (x_{k1} - med(x_1))/s_H(x_1) - (x_{k2} - med(x_2))/s_H(x_2);\ k = 1, 2, \ldots, n.$

III. Are Robust Correlation Matrices Positive Semi-definite? In this study we investigate into the question whether the correlation matrix, $\Re$, whose any element r_{ij} is a robust measure of correlation (obtained by the formulas such as Spearman's, Blomqvist's, Kendall's, Bradley's or Shevlyakov's), is positive semi-definite. We use the dataset given in Table-1 as the base data for our experiments.

We have carried out ten thousand experiments for each method (Spearman's ρ, Blomqvist's signum, Kendall's tau, Bradley's absolute r, and Shevlyakov's r_{med}) of computing robust correlation, $r_{ij} \in \Re$. In each experiment, the base data (Table-1) has been perturbed by a small (between -0.25 to 2.5) quantity generated randomly (and distributed uniformly) over n observations on all the eight variables. In each experiment, the correlation matrix, $\Re$, has been computed and its eigenvalues are obtained. If any (at least one) eigenvalue of the correlation matrix has been found to be negative, the occurrence has been counted as a failure (of the matrix being positive semi-definite), else it is counted

as a success. Such three sets of experiments have been carried out with three different seeds for generating random numbers (for perturbation).

Table-1: Base Dataset for Computation of Robust Correlation Matrices by Different Methods

SL. No.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	SL. No.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	18	3	1	2	3	3	2	7	20	8	2	9	4	9	11	7	2
2	17	17	6	6	13	10	16	16	21	13	3	9	1	7	5	3	9
3	6	1	6	1	10	16	5	7	22	15	4	7	5	6	15	12	15
4	5	4	2	1	3	2	6	1	23	17	16	11	5	9	10	10	12
5	21	10	8	3	7	7	12	8	24	2	7	5	1	3	1	1	1
6	14	20	7	3	14	16	19	14	25	3	7	3	1	2	12	4	8
7	3	15	12	1	2	15	10	9	26	20	19	4	4	8	13	14	10
8	4	13	1	1	1	6	8	6	27	19	18	6	2	16	14	19	12
9	18	16	4	1	1	4	5	5	28	3	14	9	3	11	5	10	3
10	4	14	8	4	3	16	12	14	29	8	2	7	1	10	4	2	1
11	17	14	8	9	15	11	20	13	30	16	21	11	9	10	18	18	17
12	3	9	4	6	4	4	4	6	31	5	10	4	3	12	2	11	6
13	7	5	5	2	12	9	13	10	32	21	17	9	8	11	13	15	10
14	12	6	6	3	2	8	9	8	33	14	8	4	3	5	6	10	13
15	1	5	3	4	12	15	12	11	34	9	6	1	1	2	10	8	4
16	11	1	7	1	3	2	3	1	35	19	16	7	2	1	6	7	9
17	9	12	6	8	12	16	20	16	36	19	15	10	7	4	17	17	15
18	16	5	3	1	6	3	3	7	37	22	5	7	1	6	3	4	6
19	10	11	7	10	8	14	13	15	Note: This dataset has been perturbed in our experiments								

.

IV. The results, Discussion and Conclusion: Our findings reveal that while Spearman's rho, Blomqvist's signum, Kendall's tau and Bradley's absolute correlation formulas yield positive semi-definite correlation matrix (without any failure). Of these, positive semi-definiteness of the matrices based on the first three measures (Spearman's rho, Blomqvist's signum and Kendall's tau) is expected since they are Pearsonian correlation matrices of transformed variables. However, the failure rate of Shevlyakov's formula is very high (about 81 percent: more exactly 81.47%, 80.94% and 81.41% for the three random number seeds: 13317, 31921 and 17523, respectively). Two sample correlation matrices (and their eigenvalues) are presented in Table-2.1 and Table-2.2. It is found that the smallest eigenvalue so often turns out to be negative.

Table-2.1. Shevlyakov's Robust Correlation matrix and its Eigenvalues (Sample-1)

Variable	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1.00000	0.59053	0.27080	0.13168	0.20532	0.10974	0.41999	0.40193
x_2	0.59053	1.00000	0.34953	0.65555	0.56049	0.58320	0.85656	0.66425
x_3	0.27080	0.34953	1.00000	0.19778	0.21678	0.22393	0.25129	0.34782
x_4	0.13168	0.65555	0.19778	1.00000	0.47509	0.66117	0.83071	0.60398
x_5	0.20532	0.56049	0.21678	0.47509	1.00000	0.67361	0.67501	0.39314
x_6	0.10974	0.58320	0.22393	0.66117	0.67361	1.00000	0.79762	0.70722
x_7	0.41999	0.85656	0.25129	0.83071	0.67501	0.79762	1.00000	0.74020
x_8	0.40193	0.66425	0.34782	0.60398	0.39314	0.70722	0.74020	1.00000
Eigenvalues (in descending order)								
	4.62370	**1.17033**	**0.79635**	**0.61840**	**0.42434**	**0.16789**	**0.14943**	**0.04956**

Table-2.2. Shevlyakov's Robust Correlation matrix and its Eigenvalues (Sample-2)								
Variable	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1.00000	0.58154	0.36318	0.28763	0.20059	-0.01382	0.43710	0.45526
x_2	0.58154	1.00000	0.48691	0.65733	0.46806	0.50631	0.86147	0.63320
x_3	0.36318	0.48691	1.00000	0.32810	0.17847	0.44367	0.23434	0.43348
x_4	0.28763	0.65733	0.32810	1.00000	0.42870	0.60102	0.85698	0.62314
x_5	0.20059	0.46806	0.17847	0.42870	1.00000	0.61081	0.67040	0.41083
x_6	-0.01382	0.50631	0.44367	0.60102	0.61081	1.00000	0.86103	0.73863
x_7	0.43710	0.86147	0.23434	0.85698	0.67040	0.86103	1.00000	0.69714
x_8	0.45526	0.63320	0.43348	0.62314	0.41083	0.73863	0.69714	1.00000
Eigenvalues (in descending order)								
	4.67015	1.20149	0.83475	0.58967	0.42884	0.25716	0.13572	**-0.11778**

Table-2.3. Nearest PSD Shevlyakov's Correlation matrix and its Eigenvalues (from Table-2.2)								
Variable	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1.0000	0.5915	0.3543	0.2958	0.2016	0.0056	0.4087	0.4499
x_2	0.5915	1.0000	0.4766	0.6668	0.4692	0.5287	0.8286	0.6270
x_3	0.3543	0.4766	1.0000	0.3196	0.1775	0.4236	0.2638	0.4391
x_4	0.2958	0.6668	0.3196	1.0000	0.4296	0.6195	0.8300	0.6180
x_5	0.2016	0.4692	0.1775	0.4296	1.0000	0.6130	0.6672	0.4102
x_6	0.0056	0.5287	0.4236	0.6195	0.6130	1.0000	0.7969	0.7265
x_7	0.4087	0.8286	0.2638	0.8300	0.6672	0.7969	1.0000	0.7150
x_8	0.4499	0.6270	0.4391	0.6180	0.4102	0.7265	0.7150	1.0000
Eigenvalues (in descending order)								
	4.64125994	1.18540099	0.816828682	0.585393293	0.420774306	0.243645189	0.106686193	1.14002254E-005

The observed failure rate of Shevlyakov's correlation matrix raises a question whether it can be used directly for further analysis of correlation matrices - without being approximated by the nearest positive semi-definite matrix (Mishra, 2008). While the product moment correlation coefficient (of Karl Pearson) is so much sensitive to the outliers, the robust nature of Shevlyakov's correlation coefficient is attractive. But, unfortunately, its robustness goes along with its being extremely prone to non-positive semi-definiteness and unsuitability to multivariate analysis. In view of these findings, it is safe to use Spearman's ρ, Blomqvist's signum, Kendall's tau or Bradley's absolute r rather than Shevlyakov's correlation coefficient for constructing correlation matrices for any further analysis.

As an exercise, we obtain the nearest positive semi-definite (PSD) correlation matrix from the matrix presented in Table-2.2 and present it in Table-2.3. This PSD matrix can be used for for further analysis.

References

- Blomqvist, N. (1950) "On a Measure of Dependence between Two Random Variables", *Annals of Mathematical Statistics*, 21(4): 593-600.
- Bradley, C. (1985) “The Absolute Correlation”, *The Mathematical Gazette*, 69(447): 12-17.
- Hampel, F. R., Ronchetti, E.M., Rousseeuw, P.J. and W. A. Stahel, W.A. (1986) *Robust Statistics: The Approach Based on Influence Functions,* Wiley, New York.
- Mishra, S.K. (2008) “A Note on Solution of the Nearest Correlation Matrix Problem by von Neumann Matrix Divergence”, Available at SSRN: http://ssrn.com/abstract=1106882
- Mishra, S.K. (2008) “The Nearest Correlation Matrix Problem: Solution by Differential Evolution Method of Global Optimization”, *Journal of Quantitative Economics*, New Series, 6(1&2): 240-262.
- Shevlyakov, G.L. (1997) “On Robust Estimation of a Correlation Coefficient”, *Journal of Mathematical Sciences*, 83(3): 434-438.
- Spearman, C. (1904) "The Proof and Measurement of Association between Two Things", *American Journal of Psychology,* 15: 88-93.

```
      PROGRAM ROBUSTC ! CHECK IF A ROBUST R MATRIX IS + SEMI-DEFINITE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N=37,M=8,SCALEX=100,SCALEZ=0.7,NITER=10000)
C     SCALEX SCALES GENERATED BASE 1ST VARIABLE DATA BETWEEN (0-SCALEX)
C     SCALEZ GENERATES 2ND TO MTH VARIABLE SCALEX*SCALEZ(+-)RANDOM
C     LARGER SCALEZ,SAY .9 GENERATES HIGHLY CORRELATED DATA & VICE VERSA
C     TO GENERATE HIGLY CORRELATED DATA SET SCALEZ LARGER, SAY 0.9
      DIMENSION Z(N,M),X(N,M),R(M,M),RR(M,M),AV(M),SD(M),XX(N),U(N),V(N)
      DIMENSION VR(M,M),WR(M,M),ZV(N,M)
      COMMON /RNDM/IU,IV
      WRITE(*,*)'FEED RANDOM NUMBER SEED'
      READ(*,*) IU
      WRITE(*,*)'INPUT DATA: WHETHER GENERATE (0) OR READ FROM FILE (1)'
      READ(*,*) IDAT
C     ================================================================
      IF(IDAT.NE.0) THEN
C     READS DATA FROM FILE ------------------------------------------
       OPEN(7,FILE='NDAT1.TXT')
       DO I=1,N
       READ(7,*)(ZV(I,J),J=1,M)
       WRITE(*,*)I,(ZV(I,J),J=1,M)
       ENDDO
       CLOSE(7)
      ELSE
C     GENERATES DATA ------------------------------------------------
       DO I=1,N
       CALL RANDOM(RAND)
       ZV(I,1)=RAND*SCALEX
       DO J=2,M
       CALL RANDOM(RAND)
       ZV(I,J)=ZV(I,1)*SCALEZ+(RAND-0.5)*(1.D0-SCALEZ)*SCALEX*2
       ENDDO
       ENDDO
      ENDIF
C     ================================================================
      OPEN(7,FILE='CORIND.TXT')
      WRITE(*,*)'FEED CHOICE OF CORRELATION.'
      WRITE(*,*)'(0)ZERO IS KARL PEARSON R'
      WRITE(*,*)'(1): RANK CORRELATION'
      WRITE(*,*)'(2): SIGNUM CORRELATION'
      WRITE(*,*)'(3): BRADLEY CORRELATION'
      WRITE(*,*)'(4): SHEVLYAKOV CORRELATION'
      WRITE(*,*)'(5): KENDALL TAU'
      READ(*,*) NTYPE
      ICHK=0 !WILL COUNT THE NUMBER OF FAILURES
      DO ITER=1,NITER !=====================================================
      DO I=1,N
      DO J=1,M
      CALL RANDOM(RAND)
      Z(I,J)=ZV(I,J)+(RAND-0.5D0)*0.5 !PURTURBATION (-0.25 TO 0.25)
      ENDDO
      ENDDO
      DO J=1,M
      DO I=1,N
      X(I,J)=Z(I,J)
      ENDDO
      ENDDO
C     ----------------------------------------------------------------
      IF(NTYPE.EQ.0) THEN
      WRITE(*,*) 'KARL PEARSON CORRELATION MATRIX'
      CALL PROD(X,N,M,AV,SD,R)
      ENDIF
C     ----------------------------------------------------------------
      IF(NTYPE.EQ.1) THEN
      WRITE(*,*) 'SPEARMAN RANK CORRELATION MATRIX'
      DO J=1,M
      DO I=1,N
```

```
      XX(I)=Z(I,J)
      ENDDO
      CALL RANK(XX,N) ! RANK TRANSFORMATION OF X
      DO I=1,N
      X(I,J)=XX(I)
      ENDDO
      ENDDO
      CALL PROD(X,N,M,AV,SD,R)
      WRITE(*,*)'CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(R(J,JJ),JJ=1,M)
      ENDDO
      CALL EIGEN(R,VR,WR)
      WRITE(*,*)'EIGENVALUES OF CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(WR(J,JJ),JJ=1,M)
      ENDDO
      JCHK=0
      DO I=1,M
      IF(WR(I,I).LT.0.AND.JCHK.EQ.0) THEN
      ICHK=ICHK+1
      JCHK=1
      ENDIF
      ENDDO
      ENDIF
C     ------------------------------------------------------------------
      IF(NTYPE.EQ.2) THEN
      WRITE(*,*) 'SIGNUM CORRELATION MATRIX'
      DO J=1,M
      DO I=1,N
      XX(I)=Z(I,J)
      ENDDO
      CALL MEDIAN(XX,N,AMED,AMDEV)
      DO I=1,N
      X(I,J)=1.D0
      DEV=(Z(I,J)-AMED)
      IF(DABS(DEV).GT.1.D-06) X(I,J)=DEV/DABS(DEV)
      ENDDO
      ENDDO
      DO J=1,M
      DO JJ=1,M
      R(J,JJ)=0.D0
      DO I=1,N
      CONC=X(I,J)*X(I,JJ)
      R(J,JJ)=R(J,JJ)+CONC
      ENDDO
      R(J,JJ)=R(J,JJ)/N
      ENDDO
      ENDDO
      DO J=1,M
      WRITE(7,*)(R(J,JJ),JJ=1,M)
      ENDDO
      WRITE(*,*)'------- SIGNUM CORRELATION MATRIX (PROPER) -----------'
      CALL PROD(X,N,M,AV,SD,R)
      WRITE(*,*)'OVER'
      WRITE(*,*)'CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(R(J,JJ),JJ=1,M)
      ENDDO
      CALL EIGEN(R,VR,WR)
      WRITE(*,*)'EIGENVALUES OF CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(WR(J,JJ),JJ=1,M)
      ENDDO
      JCHK=0
      DO I=1,M
      IF(WR(I,I).LT.0.AND.JCHK.EQ.0) THEN
```

```
      ICHK=ICHK+1
      JCHK=1
      ENDIF
      ENDDO
      ENDIF
C     ------------------------------------------------------------------
      IF(NTYPE.EQ.3) THEN
      WRITE(*,*) 'BRADLEY CORRELATION MATRIX'
      DO J=1,M
      DO I=1,N
      XX(I)=Z(I,J)
      ENDDO
      CALL MEDIAN(XX,N,AMED,AMDEV)
      DO I=1,N
      X(I,J)=0.D0
      DEV=(Z(I,J)-AMED)
      X(I,J)=DEV/AMDEV
      ENDDO
      ENDDO
      DO J=1,M
      DO JJ=1,M
      R(J,JJ)=0.D0
      S1=0.D0
      S2=0.D0
      DO I=1,1,N
      S1=S1+DABS(X(I,J)+X(I,JJ)) - DABS(X(I,J)-X(I,JJ))
      S2=S2+DABS(X(I,J))+ DABS(X(I,JJ))
      ENDDO
      R(J,JJ)=S1/S2
      ENDDO
      ENDDO
      DO J=1,M
      WRITE(7,*)(R(J,JJ),JJ=1,M)
      ENDDO
      WRITE(*,*)'CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(R(J,JJ),JJ=1,M)
      ENDDO
      CALL EIGEN(R,VR,WR)
      WRITE(*,*)'EIGENVALUES OF CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(WR(J,JJ),JJ=1,M)
      ENDDO
      WRITE(*,*)'-------------------------------------------------------'
      JCHK=0
      DO I=1,M
      IF(WR(I,I).LT.0.AND.JCHK.EQ.0) THEN
      ICHK=ICHK+1
      JCHK=1
      ENDIF
      ENDDO
      ENDIF
C     ------------------------------------------------------------------
      IF(NTYPE.EQ.4) THEN
      WRITE(*,*) 'SHEVLYAKOV CORRELATION MATRIX'
      DO J=1,M
      DO I=1,N
      XX(I)=Z(I,J)
      ENDDO
      CALL MEDIAN(XX,N,AMED,AMDEV)
      DO I=1,N
      XX(I)=DABS(Z(I,J)-AMED)
      ENDDO
      CALL MEDIAN(XX,N,HMED,HMDEV)
      DO I=1,N
      X(I,J)=(Z(I,J)-AMED)/HMED
      ENDDO
```

```
      ENDDO
      DO J=1,M
      DO JJ=1,M
      R(J,JJ)=0.D0
      DO I=1,N
      U(I)=DABS(X(I,J)+X(I,JJ))
      V(I)=DABS(X(I,J)-X(I,JJ))
      ENDDO
      CALL MEDIAN(U,N,UMED,UMDEV)
      CALL MEDIAN(V,N,VMED,VMDEV)
      R(J,JJ)=(UMED**2-VMED**2)/(UMED**2+VMED**2)
      ENDDO
      ENDDO
      WRITE(*,*)'CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(R(J,JJ),JJ=1,M)
      ENDDO
      CALL EIGEN(R,VR,WR)
      WRITE(*,*)'EIGENVALUES OF CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(WR(J,JJ),JJ=1,M)
      ENDDO
      WRITE(*,*)'------------------------------------------------------'
      JCHK=0
      DO I=1,M
      IF(WR(I,I).LT.0.AND.JCHK.EQ.0) THEN
      ICHK=ICHK+1
      JCHK=1
      ENDIF
      ENDDO
      ENDIF
C     ----------------------------------------------------------------
      IF(NTYPE.EQ.5) THEN
      WRITE(*,*) 'KENDALL TAU - CORRELATION MATRIX'
      DO J=1,M
      DO I=1,N
      XX(I)=Z(I,J)
      ENDDO
      CALL RANK(XX,N) ! RANK TRANSFORMATION OF X
      DO I=1,N
      X(I,J)=XX(I)
      ENDDO
      ENDDO

      TA=0.D0
      DO I=1,M
      DO J=1,M
      TC=0.D0
      DO IK=1,N
      DO JK=1,N
      IF(X(IK,I).GT.X(JK,I).AND.X(IK,J).GT.X(JK,J)) TC=TC+1
      ENDDO
      ENDDO
      R(I,J)=4.D0*TC/(N*N-N)-1.D0
      ENDDO
      ENDDO
      WRITE(*,*)'CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(R(J,JJ),JJ=1,M)
      ENDDO
      CALL EIGEN(R,VR,WR)
      WRITE(*,*)'EIGENVALUES OF CORRELATION MATRIX'
      DO J=1,M
      WRITE(*,21)(WR(J,JJ),JJ=1,M)
      ENDDO
      JCHK=0
      DO I=1,M
```

```
      IF(WR(I,I).LT.0.AND.JCHK.EQ.0) THEN
      ICHK=ICHK+1
      JCHK=1
      ENDIF
      ENDDO
      ENDIF
C     ------------------------------------------------------------------
    1 FORMAT(10F7.3)
   21 FORMAT(10F8.5)
      ENDDO !=============================================================
      WRITE(*,*)'PERCENT OF (-) EIGENVALUE CASES=',ICHK/FLOAT(NITER)*100
      CLOSE(7)
      END
C     ------------------------------------------------------------------
      SUBROUTINE PROD(X,N,M,AV,SD,R)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(N,M),R(M,M),AV(M),SD(M)
      DO J=1,M
      AV(J)=0.D0
      SD(J)=0.D0
      DO I=1,N
      AV(J)=AV(J)+X(I,J)
      SD(J)=SD(J)+X(I,J)**2
      ENDDO
      SD(J)=DSQRT((N*SD(J)-AV(J)*AV(J))/N**2)
      AV(J)=AV(J)/N
      ENDDO
      DO J=1,M
      DO JJ=1,M
      R(J,JJ)=0.D0
      DO I=1,N
      R(J,JJ)=R(J,JJ)+X(I,J)*X(I,JJ)
      ENDDO
      R(J,JJ)=(R(J,JJ)/N-AV(J)*AV(JJ))/(SD(J)*SD(JJ))
      ENDDO
      ENDDO
      DO J=1,M
      WRITE(7,*)(R(J,JJ),JJ=1,M)
      ENDDO
      WRITE(*,1)(AV(J),J=1,M)
      WRITE(*,1)(SD(J),J=1,M)
    1 FORMAT(8F9.5)
      WRITE(*,*)'-------------------------------------------------------'
      RETURN
      END
C     ------------------------------------------------------------------
      SUBROUTINE RANK(X,N)
      PARAMETER (NMAX=1000)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(N), SL(NMAX)
      DO I=1,N
      SL(I)=DFLOAT(I)
      ENDDO
      DO I=1,N-1
      DO II=I+1,N
      IF(X(I).LT.X(II)) THEN
      T=X(I)
      X(I)=X(II)
      X(II)=T
      T=SL(I)
      SL(I)=SL(II)
      SL(II)=T
      ENDIF
      ENDDO
      ENDDO
      DO I=1,N
      X(I)=I
```

```
      ENDDO
      DO I=1,N-1
      DO II=I+1,N
      IF(SL(I).GT.SL(II)) THEN
      T=X(I)
      X(I)=X(II)
      X(II)=T
      T=SL(I)
      SL(I)=SL(II)
      SL(II)=T
      ENDIF
      ENDDO
      ENDDO
      RETURN
      END
C     ------------------------------------------------------------------
      SUBROUTINE MEDIAN(X,N,A,V) ! -------------------------------------
C     SUBROUTINE MEDIAN : FINDS MEDIAN (A) AND MEAN DEVIATION (V) OF A
C     GIVEN VARIATE, VARIATE X(N)
      PARAMETER (NMAX=1000)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(N),Z(NMAX)
C     STORE X IN Z
      DO I=1,N
      Z(I)=X(I)
      ENDDO
C     ARRANGE Z IN AN ASCENDING ORDER
      DO I=1,N-1
      DO J=I+1,N
      IF(Z(I).GT.Z(J)) THEN ! EXCHANGE
      TEMP=Z(I)
      Z(I)=Z(J)
      Z(J)=TEMP
      ENDIF
      ENDDO
      ENDDO
      K=(N+1)/2 ! K IS OBTAINED AS INT((N+1)/2.0D0)
      A=(Z(K)+Z(N+1-K))/2.D0 ! GIVES MEDIAN FOR ODD AS WELL AS EVEN N
C     FIND MEAN DEVIATION
      V=0.D0
      DO I=1,N
      V=V+DABS(Z(I)-A) ! A IS MEDIAN
      ENDDO
      V=V/N ! V IS MEAN DEVIATION FROM MEDIAN
C     WRITE(*,*)'MEDIAN =',A,'  MEAN DEVIATION =',V
      RETURN
      END
C     ------------------------------------------------------------------
      SUBROUTINE EIGEN(A,V,W)
      PARAMETER(N=8)
C     COMPUTES EIGENVALUES AND VECTORS OF A REAL SYMMETRIC MATRIX
C     A(N,N) =GIVEN REAL SYMMETRIC MATRIX WHOSE EIGENVALUES AND VECTORS
C     ARE BE FOUND. ITS ORDER IS N X N
C     W(N,N) CONTAINS EIGENVALUES IN ITS MAIN DIAGONAL. OTHER ELEMENTS=0
C     V(N,N) CONTAINS EIGENVECTORS
C     PROGRAM BY KRISNAMURTHY,EV & SEN (1976) COMPUTER-BASED NUMERICAL
C     ALGORITHMS. AFFILIATED EAST-WEST PRESS, NEW DELHI
      DOUBLE PRECISION A(N,N),V(N,N),W(N,N),P(N)
      DOUBLE PRECISION PMAX,EPLN,TAN,SIN,COS,AI,TT,TA,TB
      DIMENSION MM(N)
C     ------------ INITIALISATION ---------------------------------------
      DO I=1,N
      DO J=1,N
      V(I,J)=0.D0
      W(I,J)=A(I,J)
      ENDDO
      P(I)=0.D0
```

```
      ENDDO
      PMAX=0.D0
      EPLN=0.D0
      TAN=0.D0
      SIN=0.D0
      COS=0.D0
      AI=0.D0
      TT=0
      NN=1
      EPLN=1.0D-100
C     ------------------------------------------------------------------
      IF(NN.NE.0) THEN
      DO I=1,N
      DO J=1,N
      V(I,J)=0.D0
      IF(I.EQ.J) V(I,J)=1.D0
      ENDDO
      ENDDO
      ENDIF
      NR=0
      MI=N-1
      DO I=1,MI
      P(I)=0.D0
      MJ=I+1
      DO J=MJ,N
      IF(P(I).LE.DABS(A(I,J))) THEN
      P(I)=DABS(A(I,J))
      MM(I)=J
      ENDIF
      ENDDO
      ENDDO

    7 DO 8 I=1,MI
      IF(I.LE.1) GOTO 10
      IF(PMAX.GT.P(I)) GOTO 8
   10 PMAX=P(I)
      IP=I
      JP=MM(I)
    8 CONTINUE
C     EPLN=DABS(PMAX)*1.0D-09
      IF (PMAX.LE.EPLN) THEN
C     WRITE(*,*)'PMAX EPLN',PMAX, EPLN
C     PAUSE'CONVERGENCE CRITERION IS MET'
      GO TO 12
      ENDIF
      NR=NR+1
      TA=2.D0*A(IP,JP)
      TB=(DABS(A(IP,IP)-A(JP,JP))+
     1DSQRT((A(IP,IP)-A(JP,JP))**2+4.D0*A(IP,JP)**2))
      TAN=TA/TB
      IF(A(IP,IP).LT.A(JP,JP)) TAN=-TAN
      COS=1.D0/DSQRT(1.D0+TAN**2)
      SIN=TAN*COS
      AI=A(IP,IP)
      A(IP,IP)=(COS**2)*(AI+TAN*(2.D0*A(IP,JP)+TAN*A(JP,JP)))
      A(JP,JP)=(COS**2)*(A(JP,JP)-TAN*(2.D0*A(IP,JP)-TAN*AI))
      A(IP,JP)=0.D0
      IF(A(IP,IP).GE.A(JP,JP)) GO TO 15
      TT=A(IP,IP)
      A(IP,IP)=A(JP,JP)
      A(JP,JP)=TT
      IF(SIN.GE.0.D0) GO TO 16
      TT=COS
      GO TO 17
   16 TT=-COS
   17 COS=DABS(SIN)
      SIN=TT
```

```
470:     15 DO 18 I=1,MI
471:        IF(I-IP) 19, 18, 20
472:     20 IF(I.EQ.JP)GO TO 18
473:     19 IF(MM(I).EQ.IP) GO TO 21
474:        IF(MM(I).NE.JP) GO TO 18
475:     21 K=MM(I)
476:        TT=A(I,K)
477:        A(I,K)=0.D0
478:        MJ=I+1
479:        P(I)=0.D0
480:        DO 22 J=MJ,N
481:        IF(P(I).GT.DABS(A(I,J))) GO TO 22
482:        P(I)=DABS(A(I,J))
483:        MM(I)=J
484:     22 CONTINUE
485:        A(I,K)=TT
486:     18 CONTINUE
487:        P(IP)=0.D0
488:        P(JP)=0.D0
489:        DO 23 I=1,N
490:        IF(I-IP) 24, 23, 25
491:     24 TT=A(I,IP)
492:        A(I,IP)=COS*TT+SIN*A(I,JP)
493:        IF(P(I).GE.DABS(A(I,IP))) GO TO 26
494:        P(I)=DABS(A(I,IP))
495:        MM(I)=IP
496:     26 A(I,JP)=-SIN*TT+COS*A(I,JP)
497:        IF(P(I).GE.DABS(A(I,JP))) GO TO 23
498:     30 P(I)=DABS(A(I,JP))
499:        MM(I)=JP
500:        GO TO 23
501:     25 IF(I.LT.JP) GO TO 27
502:        IF(I.GT.JP) GO TO 28
503:        IF(I.EQ.JP) GO TO 23
504:     27 TT=A(IP,I)
505:        A(IP,I)=COS*TT+SIN*A(I,JP)
506:        IF(P(IP).GE.DABS(A(IP,I))) GO TO 29
507:        P(IP)=DABS(A(IP,I))
508:        MM(IP)=I
509:     29 A(I,JP)=-TT*SIN+COS*A(I,JP)
510:        IF(P(I).GE.DABS(A(I,JP))) GO TO 23
511:        GO TO 30
512:     28 TT=A(IP,I)
513:        A(IP,I)=TT*COS+SIN*A(JP,I)
514:        IF(P(IP).GE.DABS(A(IP,I))) GO TO 31
515:        P(IP)=DABS(A(IP,I))
516:        MM(IP)=I
517:     31 A(JP,I)=-TT*SIN+COS*A(JP,I)
518:        IF(P(JP).GE.DABS(A(JP,I))) GO TO 23
519:        P(JP)=DABS(A(JP,I))
520:        MM(JP)=I
521:     23 CONTINUE
522:        IF(NN.EQ.0) GOTO 7
523:        DO 32 I=1,N
524:        TT=V(I,IP)
525:        V(I,IP)=TT*COS+SIN*V(I,JP)
526:        V(I,JP)=-TT*SIN+COS*V(I,JP)
527:     32 CONTINUE
528:        GO TO 7
529:     12 DO I=1,N
530:        P(I)=A(I,I)
531:        ENDDO
532:        DO I=1,N
533:        DO J=1,N
534:        A(I,J)=W(I,J)
535:        W(I,J)=0.D0
536:        ENDDO
```

```
      W(I,I)=P(I)
      ENDDO
      RETURN
      END
C     ------------------------------------------------------------------
C     RANDOM NUMBER GENERATOR (UNIFORM BETWEEN 0 AND 1 - BOTH EXCLUSIVE)
      SUBROUTINE RANDOM(RAND)
       DOUBLE PRECISION  RAND
       COMMON /RNDM/IU,IV
       IV=IU*65539 ! THIS IS (2^16)+(2^1)+(2^0)
       IF(IV.LT.0) THEN
       IV=IV+2147483647+1  ! THIS IS (2^31-1) +1
       ENDIF
       RAND=IV
       IU=IV
       RAND=RAND*4.656612875245796924105750827168D-10 !THIS = 1/(2^31-1)
       RETURN
       END
```

```
      PROGRAM NCORMAT ! ---------------------------------------------
C     (OBTAINS POSITIVE SEMI-DEFINITE CORRELATION MATRIX (R)
C     FROM A NON-POSITIVE-SEMI-DEFINITE  or PSEUDO-CORRELATION MATRX (Q)
C     Non-Positive-Semi-definite matrix has at least one - BUT NOT ALL -
C     of its eigenvalues negative
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NORM=2)
      DIMENSION  A(200,200),V(200,200),W(200,200),P(200),R(200,200)
      DIMENSION D(200,200),Q(200,200),C(200,200),MM(200),QQ(200,200)
      CHARACTER *80 INFIL,OUTFIL
      COMMON /RNDM/IU,IV
C     ---------------------------------------------------------------------
      WRITE(*,*)'THE ORDER OF Q MATRIX ?'
      READ(*,*) N
      WRITE(*,*)'MATRIX Q TO BE READ OR GENERATED? TO BE READ FEED(0)'
      WRITE(*,*)'IF Q TO BE GENERATED, THEN FEED(ANY NON-ZERO NUMBER)'
      READ(*,*) IGEN
      WRITE(*,*)'OUTPUT FILE NAME (TO STORE THE RESULTS)?'
      READ(*,*) OUTFIL
      WRITE(*,*)'NUMBER OF ITERATIONS(SAY 100) AND ACC (SAY 1.0D-10)?'
      WRITE(*,*)'ACC IS A SMALL NUMBER FOR ACCURACY'
      READ(*,*) NIT,ACC
      WRITE(*,*)'FEED A PERTURBATION VALUE TO REPLACE NONPOSITIVE EIGEN'
      WRITE(*,*)'VALUE. IT WOULD BE A SMALL POSITIVE VALUE, SAY, 1.D-10'
      READ(*,*) EPS
C     ---------------------------------------------------------------------
      IF(IGEN.EQ.0) THEN
      WRITE(*,*)'INPUT FILE NAME (TO READ THE Q MATRIX)?'
      READ(*,*) INFIL
      OPEN(7,FILE=INFIL) ! OPENING THE INUT FILE
      DO I=1,N
      READ(7,*)(Q(I,J),J=1,N)
      ENDDO
      CLOSE(7) ! CLOSING THE INUT FILE

C     -----------------GENERATE MATRIX FOR EXPERIMENTS ----------------
      ELSE
      WRITE(*,*)'FEED SEED'
      READ(*,*) IU
      DO I=1,N
      DO J=1,N
      CALL RANDOM(RAND)
      Q(I,J)=0.2D0+RAND*.7 ! OR BY SOME OTHER SPECIFICATION
      IF(I.EQ.J) Q(I,J)=1.D0
      ENDDO
      ENDDO
      ENDIF
C     ---------------------------------------------------------------------
C     ---------------------------------------------------------------------
      OPEN(7,FILE=OUTFIL) !OPENING THE OUTPUT FILE, WILL BE CLOSED LATER
      WRITE(7,*)'THE ORIGINAL MATRIX (Q) IS GIVEN BELOW'
      DO I=1,N
      WRITE(7,1)(Q(I,J),J=1,N)
      ENDDO
    1 FORMAT(10F8.4)
C     ---------- STORE Q IN A: WORK WITH A, PRESERVE Q ----------------
      DO I=1,N
      DO J=1,N
      A(I,J)=Q(I,J)
      ENDDO
      ENDDO
      CALL EIGEN(A,N,NN,V,W,P,MM) ! FIND EIGENVALUES AND VECTORS OF A
      WRITE(*,*)'THE EIGENVALUES OF THE ORIGINAL MATRIX ARE AS FOLLOWS'
      WRITE(7,*)'THE EIGENVALUES OF THE ORIGINAL MATRIX ARE AS FOLLOWS'
      WRITE(*,*)(A(I,I),I=1,N)
      WRITE(7,*)(A(I,I),I=1,N)
      WRITE(*,*)'-------------------------------------------------------'
```

```
C     A IS DISTURBED DUE TO INVOKING EIGEN. RESTORE IT,INITIALIZE OTHERS
      DO I=1,N
      DO J=1,N
      W(I,J)=0.D0 !INITIALISE AND KEEP NONNEGATIVE EIGENVALUES IN W(I,I)
      IF(A(I,J).GT.0.D0 .AND. I.EQ.J) W(I,J)=A(I,J)
      IF(I.EQ.J.AND.W(I,J).EQ.0.D0) W(I,J)=EPS
      A(I,J)=Q(I,J) ! RESTORE THE ORIGINAL MATRIX IN A
      D(I,J)=0.D0 ! INTIALIZE D MATRIX BY ZERO
      ENDDO
      ENDDO
C     ------------------------------------------------------------------
      S=1.0D30 ! NORM VERY LARGE TO BEGIN WITH

      DO IT=1,NIT ! ITERATION BEGINS
      DO I=1,N
      DO J=1,N
      R(I,J)=A(I,J)-D(I,J)
      A(I,J)=R(I,J)
      ENDDO
      ENDDO

      CALL EIGEN(A,N,NN,V,W,P,MM) ! FIND EIGENVALUES AND VECTORS OF A
      DO I=1,N
      DO J=1,N
      W(I,J)=0.D0
      IF(I.EQ.J .AND. A(I,J).GT.0.D0) W(I,J)=A(I,J)
      IF(I.EQ.J.AND.W(I,J).EQ.0.D0) W(I,J)=EPS
      ENDDO
      ENDDO
C     CONSTRUCTING THE MATRIX FROM EIGENVECTORS AND NON-NEG EIGENVALUES
      DO I=1,N
      DO J=1,N
      C(I,J)=0.D0
      DO K=1,N
      C(I,J)=C(I,J)+V(I,K)*W(K,J)
      ENDDO
      ENDDO
      ENDDO
      DO I=1,N
      DO J=1,N
      A(I,J)=0.D0
      DO K=1,N
      A(I,J)=A(I,J)+C(I,K)*V(J,K)
      ENDDO
      ENDDO
      ENDDO
C     MATRIX HAS BEEN RECONSTRUCTED
      DO I=1,N
      DO J=1,N
      D(I,J)=A(I,J)-R(I,J)
      IF(I.EQ.J) A(I,J)=1.D0
      ENDDO
      ENDDO

C     COMPUTE NORM
      SS=0.D0
      DO I=1,N
      DO J=1,N
      SS=SS+DABS(A(I,J)-Q(I,J))**2
      ENDDO
      ENDDO
      IF(DABS(SS-S).LT.ACC) THEN
      S=SS
      GO TO 10
      ELSE
      S=SS
      ENDIF
```

```
      ENDDO  ! ITERATIONS END

C     ----------------------------------------------------------------
C     FINAL RESULTS
  10  WRITE(*,*)'NO. OF ITERATIONS PERFORMED = ',IT

      WRITE(7,*)'FINAL RESULTS - THE OUTPUT MATRIX'
      DO I=1,N
      WRITE(7,1)(A(I,J),J=1,N)
      DO J=1,N
      R(I,J)=A(I,J)
      ENDDO
      ENDDO
      WRITE(7,*)'------THE FINAL EIGENVALUES ARE --------'
      CALL EIGEN(R,N,NN,V,W,P,MM)
      WRITE(7,*)(R(I,I),I=1,N)
      WRITE(7,*)'NORM=',SS**(1.D0/NORM)
      WRITE(7,*)'NO. OF ITERATIONS PERFORMED = ',IT
      CLOSE(7)! CLOSE THE OUTPUT FILE
C     ----------------------------------------------------------------
      WRITE(*,*)'RESULTS HAVE BEEN STORED IN THE OUTPUT FILE'
      WRITE(*,*)'END'
      END
C     ----------------------------------------------------------------
      SUBROUTINE EIGEN(A,N,NN,V,W,P,MM)
C     THIS SUBROUTINE IS ADAPTED FROM KRISHNAMURTHY AND SEN (1976)
      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)
      DIMENSION A(200,200),V(200,200),W(200,200),P(200)
      DIMENSION MM(200)
C     ------------ INITIALISATION -------------------------
      NN=1
      DO 50 I=1,N
      DO 51 J=1,N
      V(I,J)=0.0
 51   W(I,J)=0.0
      P(I)=0.0
 50   CONTINUE
      PMAX=0
      EPLN=0
      TAN=0
      SIN=0
      COS=0
      AI=0
      TT=0
      EPLN=1.0D-310
C      EPLN=1.0D-99
C     ---------------------------------------------------------
      IF(NN.NE.0) THEN
      DO 3 I=1,N
      DO 3 J=1,N
      V(I,J)=0.0
      IF(I.EQ.J) V(I,J)=1.0
 3    CONTINUE
      ENDIF
 2    NR=0
 5    MI=N-1
      DO 6 I=1,MI
      P(I)=0.0
      MJ=I+1
      DO 6 J=MJ,N
      IF(P(I).GT.DABS(A(I,J))) GO TO 6
      P(I)=DABS(A(I,J))
      MM(I)=J
 6    CONTINUE
 7    DO 8 I=1,MI
      IF(I.LE.1) GOTO 10
      IF(PMAX.GT.P(I)) GOTO 8
```

```
202: 10    PMAX=P(I)
203:       IP=I
204:       JP=MM(I)
205: 8     CONTINUE
206:       IF (PMAX.LE.EPLN) THEN
207:       GO TO 12
208:       ENDIF
209:       NR=NR+1
210: 13    TA=2.0*A(IP,JP)
211:       TB=(DABS(A(IP,IP)-A(JP,JP))+
212:      * DSQRT((A(IP,IP)-A(JP,JP))**2+4.0*A(IP,JP)**2))
213:       TAN=TA/TB
214:       IF(A(IP,IP).LT.A(JP,JP)) TAN=-TAN
215: 14    COS=1.0/DSQRT(1.0+TAN**2)
216:       SIN=TAN*COS
217:       AI=A(IP,IP)
218:       A(IP,IP)=(COS**2)*(AI+TAN*(2.0*A(IP,JP)+TAN*A(JP,JP)))
219:       A(JP,JP)=(COS**2)*(A(JP,JP)-TAN*(2.0*A(IP,JP)-TAN*AI))
220:       A(IP,JP)=0.0
221:       IF(A(IP,IP).GE.A(JP,JP)) GO TO 15
222:       TT=A(IP,IP)
223:       A(IP,IP)=A(JP,JP)
224:       A(JP,JP)=TT
225:       IF(SIN.GE.0) GO TO 16
226:       TT=COS
227:       GO TO 17
228: 16    TT=-COS
229: 17    COS=DABS(SIN)
230:       SIN=TT
231: 15    DO 18 I=1,MI
232:       IF(I-IP) 19, 18, 20
233: 20    IF(I.EQ.JP)GO TO 18
234: 19    IF(MM(I).EQ.IP) GO TO 21
235:       IF(MM(I).NE.JP) GO TO 18
236: 21    K=MM(I)
237:       TT=A(I,K)
238:       A(I,K)=0.0
239:       MJ=I+1
240:       P(I)=0.0
241:       DO 22 J=MJ,N
242:       IF(P(I).GT.DABS(A(I,J))) GO TO 22
243:       P(I)=DABS(A(I,J))
244:       MM(I)=J
245: 22    CONTINUE
246:       A(I,K)=TT
247: 18    CONTINUE
248:       P(IP)=0.0
249:       P(JP)=0.0
250:       DO 23 I=1,N
251:       IF(I-IP) 24, 23, 25
252: 24    TT=A(I,IP)
253:       A(I,IP)=COS*TT+SIN*A(I,JP)
254:       IF(P(I).GE.DABS(A(I,IP))) GO TO 26
255:       P(I)=DABS(A(I,IP))
256:       MM(I)=IP
257: 26    A(I,JP)=-SIN*TT+COS*A(I,JP)
258:       IF(P(I).GE.DABS(A(I,JP))) GO TO 23
259: 30    P(I)=DABS(A(I,JP))
260:       MM(I)=JP
261:       GO TO 23
262: 25    IF(I.LT.JP) GO TO 27
263:       IF(I.GT.JP) GO TO 28
264:       IF(I.EQ.JP) GO TO 23
265: 27    TT=A(IP,I)
266:       A(IP,I)=COS*TT+SIN*A(I,JP)
267:       IF(P(IP).GE.DABS(A(IP,I))) GO TO 29
268:       P(IP)=DABS(A(IP,I))
```

```
269:        MM(IP)=I
270:  29    A(I,JP)=-TT*SIN+COS*A(I,JP)
271:        IF(P(I).GE.DABS(A(I,JP))) GO TO 23
272:        GO TO 30
273:  28    TT=A(IP,I)
274:        A(IP,I)=TT*COS+SIN*A(JP,I)
275:        IF(P(IP).GE.DABS(A(IP,I))) GO TO 31
276:        P(IP)=DABS(A(IP,I))
277:        MM(IP)=I
278:  31    A(JP,I)=-TT*SIN+COS*A(JP,I)
279:        IF(P(JP).GE.DABS(A(JP,I))) GO TO 23
280:        P(JP)=DABS(A(JP,I))
281:        MM(JP)=I
282:  23    CONTINUE
283:        IF(NN.EQ.0) GOTO 7
284:        DO 32 I=1,N
285:        TT=V(I,IP)
286:        V(I,IP)=TT*COS+SIN*V(I,JP)
287:        V(I,JP)=-TT*SIN+COS*V(I,JP)
288:  32    CONTINUE
289:        GO TO 7
290:  12    RETURN
291:        END
292:  C     ---------------------------------------------------------------------
293:  C     RANDOM NUMBER GENERATOR (UNIFORM BETWEEN 0 AND 1 - BOTH EXCLUSIVE)
294:         SUBROUTINE RANDOM(RAND)
295:         IMPLICIT DOUBLE PRECISION  (A-H,O-Z)
296:         COMMON /RNDM/IU,IV
297:         IV=IU*65539
298:         IF(IV.LT.0) THEN
299:         IV=IV+2147483647+1
300:         ENDIF
301:         RAND=DBLE(IV+.0D0)
302:         IU=IV
303:         RAND=RAND*0.4656613E-09
304:         RETURN
305:         END
306:  C     ---------------------------------------------------------------------
307:
308:
```