General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

376 NO

MSC-IN-65-ED-21

MSC INTERNAL TECHNICAL NOTE

GENERATION OF RANDOM VECTORS WITH KNOWN MEAN AND COVARIANCE

BY

Fred M. Speed

FACILITY FORM 602

N70-35849
(ACCESSION NUMBER)
(THRU)

26
(PAGES)

1
(CODE)

TMX-64494
(NASA CR OR TMX OR AD NUMBER)

19
(CATEGORY)

AUG 1970
RECEIVED
NASA STI FACILITY
INPUT BRANCH

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

June 8, 1965

MSC INTERNAL NOTE NO. 65-ED-21

GENERATION OF RANDOM VECTORS WITH KNOWN MEAN AND COVARIANCE

Prepared by: Fred M. Speed
Fred M. Speed
AST, Data Systems

Approved: Eugene L. Davis, Jr.
Eugene L. Davis, Jr.
Chief, Theory and
Analysis Office

Approved: Eugene H. Brock
Eugene H. Brock
Chief, Computation and
Analysis Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

June 8, 1965

GENERATION OF RANDOM VECTORS WITH KNOWN MEAN AND COVARIANCE

By

Fred M. Speed
MANNED SPACECRAFT CENTER

SUMMARY

In simulation studies, it is necessary to introduce normally distributed random "errors" to the data in order to simulate the actual conditions. These random "errors" are normally distributed random vectors with known mean and covariance. The algorithms currently being used (ref. 3) to generate these vectors encounter one of the two following problems.

(1) The algorithm can not handle singular covariance matrices (a singular covariance matrix occurs when the data is completely correlated).

(2) The algorithm requires the inversion of a matrix.

The purpose of this paper is to present an algorithm that does not require the inversion of a matrix and can handle both singular and non-singular covariance matrices. The proposed algorithm should be more efficient than the current algorithms.

INTRODUCTION

This paper is divided into four parts. The first part gives some definitions and theorems that will be needed in the development of the algorithm. The second part contains the development of the algorithm. The third part describes a computer program that generates the desired random vectors. The last part contains Appendices A and B. Appendix A contains a listing of the computer program GERN and presents several examples illustrating the use of the program. Appendix B contains listings of the subroutines that make up the N(0, 1) random number generator packet.

SYMBOLS

X	column vector
Y	column vector
R	positive semi-definite symmetric matrix
Other capital letters	matrices, unless otherwise stated
A^T	transpose of A
A^{-1}	inverse of A
Y is distributed according to $N(\mu, R)$	Y is from a normal population with mean μ and covariance matrix R

E(Y)	expected value of Y
COV(Y)	covariance of Y
I	identity matrix
ϕ	null vector
D	diagonal matrix

DEFINITIONS AND THEOREMS

Definition 1. Two n x n real matrices A and B are said to be congruent if there exists a non-singular matrix P such that

$$PAP^T = B.$$

Since congruence is a special case of equivalence, the matrix P can be obtained from elementary row-column operations (ref. 1).

Theorem I. (ref. 1) Every n x n real symmetric matrix A of rank r is congruent to a diagonal matrix, whose diagonal elements consist of r ones and n-r zeros.

Theorem II. (ref. 2) Let X be distributed according to $N(\phi, H)$. If Y = AX, then Y is distributed according to $N(\phi, AHA^T)$.

Theorem III. (ref. 2) If $Y = AX + \mu$, where A is a constant matrix and μ is a constant column vector, then

$$E(Y) = AE(X) + \mu$$

$$COV(Y) = COV(AX)$$

The following theorem is an immediate consequence of matrix theory.

Theorem IV. If D is a diagonal matrix, whose element consists of zeros and ones, then $D = D \cdot D$.

THE DEVELOPMENT OF THE ALGORITHM

Suppose it is necessary to generate an $n \times 1$ random vector Y with mean ϕ and covariance R. The following theorem provides a new method to obtain this Y.

Theorem V. Let X, an $n \times 1$ vector, be distributed according to $N(\phi, I)$. If R is a $n \times n$ given covariance matrix, then there exists an $n \times n$ matrix A, such that if $Y = AX$, then Y is distributed according to $N(\phi, R)$.

Proof: Let the rank of R be $r \leq n$. By definition, R is a real symmetric matrix. Thus, by Theorem 1, there exists a non-singular matrix P such that $PRP^T = D$, where D is a diagonal matrix whose diagonal elements consists of r ones and n-r zeros.

Theorem II implies that a necessary and sufficient condition that $Y = AX$ be $N(\phi, R)$ is that $AA^T = R$. It will be shown that $A = RP^T$ is one matrix such that $AA^T = R$.

Since $PRP^T = D$, then

(i) $RP^T = P^{-1} D$

(ii) $PR = D\,(P^T)^{-1}$

(iii) $R = P^{-1} D\,(P^T)^{-1}$

Thus, if $A = RP^T$, then

$$\begin{aligned} AA^T &= RP^T\, PR^T \\ &= RP^T\, PR \\ &= P^{-1}\, D\, D(P^T)^{-1} \\ &= P^{-1}\, D(P^T)^{-1} \\ &= R \end{aligned}$$

Thus, if $A = RP^T$, then $Y = AX = RP^TX$ is $N(\phi, R)$ and the theorem is proved.

It has been shown (ref. 1) that P can be found by forming the augmented matrix [R, I] and operating on this matrix with elementary row-column operations in such a way that R is diagonalized. The result of these operations on [R, I] is [D, P]. Since X is distributed according to N(ϕ, I), the elements of X are readily obtainable from a N(0, 1) random number generator. Thus, given the covariance matrix R, the random vector Y can be obtained rather easily.

SUBROUTINE GERN

GERN is a FORTRAN IV subroutine used to generate random vectors having a known mean and known covariance matrix. All computations are done in single-precision floating point arithmetic. The method used is described in the previous section.

Calling Sequence

Call GERN (N, R, Y, A)

where:

N	size of R
R	covariance matrix. R is dimensioned R(50, 50)

Y	n x 1 random vector from a N(ϕ, R) population. Y is dimensioned Y(50)
A	a matrix such that Y = AX is distributed according to N(ϕ, R). A is dimensioned A(50, 50).

Error Messages

If R is not positive semi-definite, a negative number will occur on the diagonal of R during the row-column operations. When this occurs, the following message is printed: "THE ORIGINAL MATRIX IS NOT POSITIVE SEMI-DEFINITE."

Restrictions

$N \leq 50$

N(0, 1) Random Number Generator

See Appendix B

Method

Given: R

Construct: [R, I]

Operate on [R, I] with elementary row-column operations to obtain [D, P].

Construct: $X \sim N(0, I)$

Compute: $A = RP^T$

Compute: $Y = AX$

$Y \sim N($?

CONCLUSION

Theorem V provides a very simple method to obtain random vectors that can be used to simulate observations that are highly or even completely correlated as well as observations that are independent.

REFERENCES

1. Ayres Jr., Frank: Theory and Problems of Matrices. New York, Schorem Publishing Co., pp. 40-42, p. 115.

2. Anderson, T. W.: An Introduction to Multivariate Statistical Analysis. New York, John Waley and Sons, Inc., p. 25.

3. Tapley, B. D.: Odell, P. L.: A Study of Optimum Methods for Determining and Predicting Space Vehicle Trajectories and Control Programs. Contract NAS 9-2619.

4. Brunk, H. D.: An Introduction to Mathematical Statistics. Boston, Gin and Co., pp. 86-90.

APPENDIX A

This appendix contains three examples and a listing of the Subroutine GERN.

Example 1: Suppose it is necessary to simulate a 5 x 1 radar observation vector Y, where

$$Y = \begin{bmatrix} \text{range} \\ \text{azimuth} \\ \text{time} \\ \text{elevation} \\ \text{range-rate} \end{bmatrix}$$

In order to simulate the actual conditions, random "observation errors" must be added to the vector Y. Let these "errors" be from a N(ϕ, R) population, where

$$R = \begin{bmatrix} 1.0000 & 0.5576 & 0.4641 & 0.8197 & 0.2333 \\ 0.5576 & 2.0000 & 0.1719 & 0.2516 & 0.2265 \\ 0.4641 & 0.1719 & 3.0000 & 0.0264 & 0.0334 \\ 0.8197 & 0.2516 & 0.0264 & 4.0000 & 0.9608 \\ 0.2333 & 0.2265 & 0.0334 & 0.9608 & 5.0000 \end{bmatrix}$$

Subroutine GERN computes the matrix

$$A = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.5576 & 1.2996 & 0.0000 & 0.0000 & 0.0000 \\ 0.4641 & -0.0668 & 1.6673 & 0.0000 & 0.0000 \\ 0.8197 & -0.1581 & -0.2186 & 1.8042 & 0.0000 \\ 0.2333 & 0.0742 & -0.0419 & 0.4279 & 2.1806 \end{bmatrix}$$

This matrix is then multiplied by X, where X is distributed according to $N(\phi, I)$. The result of this multiplication is the random "error" vector

$$Z = \begin{bmatrix} -1.0675 \\ 1.0053 \\ -1.7816 \\ -1.6026 \\ -4.2417 \end{bmatrix}$$

The vector Z is added to the vector Y to obtain the simulated radar observation vector.

Example 2: Suppose, in Example 1, it is assumed th
can be measured without error. This
the varianc
Under this assumption, the "erro

from a N(ϕ, R) population, where R might be

$$R = \begin{bmatrix} 1.0000 & 0.2248 & 0.0000 & 0.9471 & 0.4625 \\ 0.2248 & 2.0000 & 0.0000 & 0.0865 & 0.6449 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.9471 & 0.0865 & 0.0000 & 4.0000 & 0.2663 \\ 0.4625 & 0.6449 & 0.0000 & 0.2663 & 5.0000 \end{bmatrix}$$

The matrix A is computed to be

$$A = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.2248 & 1.3962 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.9471 & -0.0905 & 0.0000 & 1.7591 & 0.0000 \\ 0.4625 & 0.3873 & 0.0000 & -0.0777 & 2.1517 \end{bmatrix}$$

and the random "error" vector Z is

$$Z = \begin{bmatrix} 0.5879 \\ 0.2784 \\ 0.0000 \\ 1.7551 \\ 0.5159 \end{bmatrix}$$

Example 3: Suppose it is necessary to simulate a 6 x 1 observation vector Y, where

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix}$$

and where $y_6 = \sum_{i=1}^{5} y_i$

This is a case where an element of the observation vector is completely correlated with the other elements. Let the "errors" be from a $N(\phi, R)$ population where

$$R = \begin{bmatrix} 2.000 & 0.411 & 1.334 & -0.097 & 1.612 & 5.259 \\ 0.411 & 4.000 & -0.238 & -0.684 & -0.656 & 2.832 \\ 1.334 & -0.238 & 6.000 & -1.590 & 1.024 & 6.530 \\ -0.097 & -0.684 & -1.590 & 8.000 & -1.226 & 4.401 \\ 1.612 & -0.656 & 1.024 & -1.226 & 10.000 & 10.755 \\ 5.259 & 2.832 & 6.530 & 4.401 & 10.755 & 29.779 \end{bmatrix}$$

Note that the last row of R is the sum of the first five rows of R and hence R is a singular covariance matrix.

Subroutine GERN computes the matrix

$$A = \begin{bmatrix} 1.414 & 0.000 & 0.000 & 0.000 & 0.000 & 0.002 \\ 0.290 & 1.978 & 0.000 & 0.000 & 0.000 & 0.002 \\ 0.943 & -0.258 & 2.245 & 0.000 & 0.000 & 0.004 \\ -0.069 & -0.335 & -0.718 & 2.714 & 0.000 & 0.001 \\ 1.114 & -0.498 & -0.080 & -0.506 & 2.861 & 0.000 \\ 3.719 & 0.886 & 1.447 & 2.208 & 2.861 & 0.005 \end{bmatrix}$$

The random "error" vector Z is calculated as before and the result is

$$Z = \begin{bmatrix} -1.581 \\ -0.051 \\ 3.311 \\ -0.496 \\ .688 \\ 1.874 \end{bmatrix}$$

A listing of Subroutine GERN follows.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

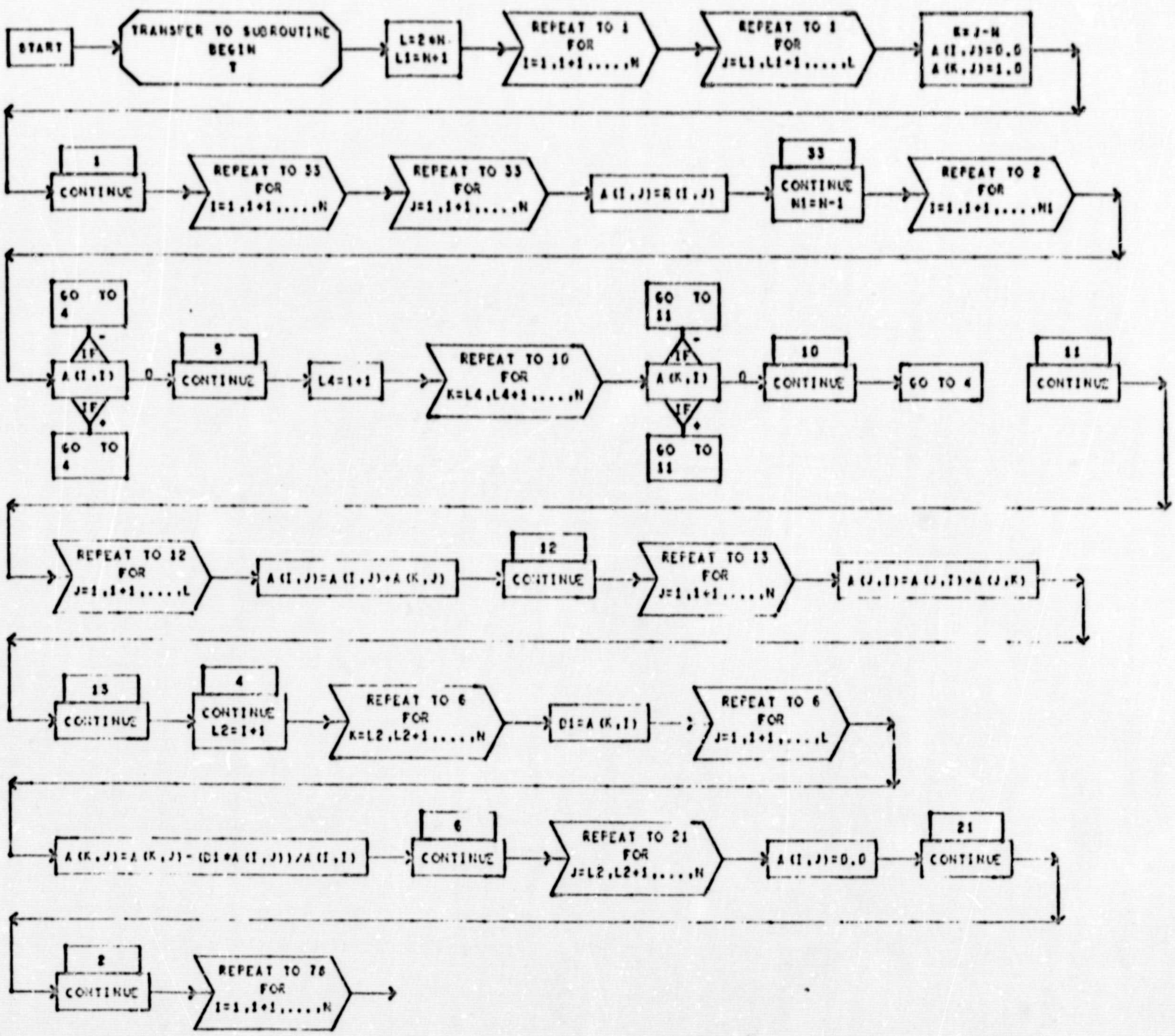
```
$IBFTC GERN
      SUBROUTINE GERN(N,R,Y,H)
C     GERN WAS PROGRAMED BY F. M. SPEED  MSC , APRIL,1965
C     CALLING SEQUENCE
C     CALL GERN(N,R,Y,A)
C     WHERE
C     N  IS THE SIZE OF R
C     R  IS THE COVARIANCE MATRIX,R IS DIMENSIONED  R(50,50)
C     Y  IS RANDOM VECTOR FROM A N(Ø,R) POPULATION . Y IS DIMENSIONE
C     Y(50)
C     A  IS A MATRIX SUCH THAT Y=AX IS DISTRIBUTED ACCORDING TO N(Ø
C     A IS DIMENSIONED A(50,50).
      DIMENSION Y(50),X(50),A(50,100),B(50,50),P(50,50)
     A ,D(50,50),H(50,50)
      CALL BEGIN(T)
      L = 2*N
      L1 = N+1
C   CONSTRUCT  (R,I)
      DO  1 I= 1,N
      DO  1 J=L1,L
      K = J-N
      A(I,J) = 0.0
      A(K,J) = 1.0
    1 CONTINUE
      DO 33 I=1,N
      DO 33 J=1,N
      A(I,J) = R(I,J)
   33 CONTINUE
      N1 = N-1
C   COMPUTE     (D,P)
      DO 2 I = 1,N1
      IF(A(I,I))4,5,4
    5 CONTINUE
      L4 = I + 1
      DO 10 K =L4,N
      IF(A(K,I))11,10,11
   10 CONTINUE
      GO TO 4
   11 CONTINUE
      DO 12 J=1,L
      A(I,J) = A(I,J) + A(K,J)
   12 CONTINUE
      DO 13 J = 1,N
      A(J,I) = A(J,I) + A(J,K)
   13 CONTINUE
    4 CONTINUE
      L2 = I+1
      DO 6 K=L2,N
      D1 = A(K,I)
      DO 6 J= 1,L
      A(K,J) = A(K,J) - (D1*A(I,J))/A(I,I)
    6 CONTINUE
      DO 21 J = L2,N
      A(I,J) = 0.0
   21 CONTINUE
    2 CONTINUE
      DO 78 I=1,N
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

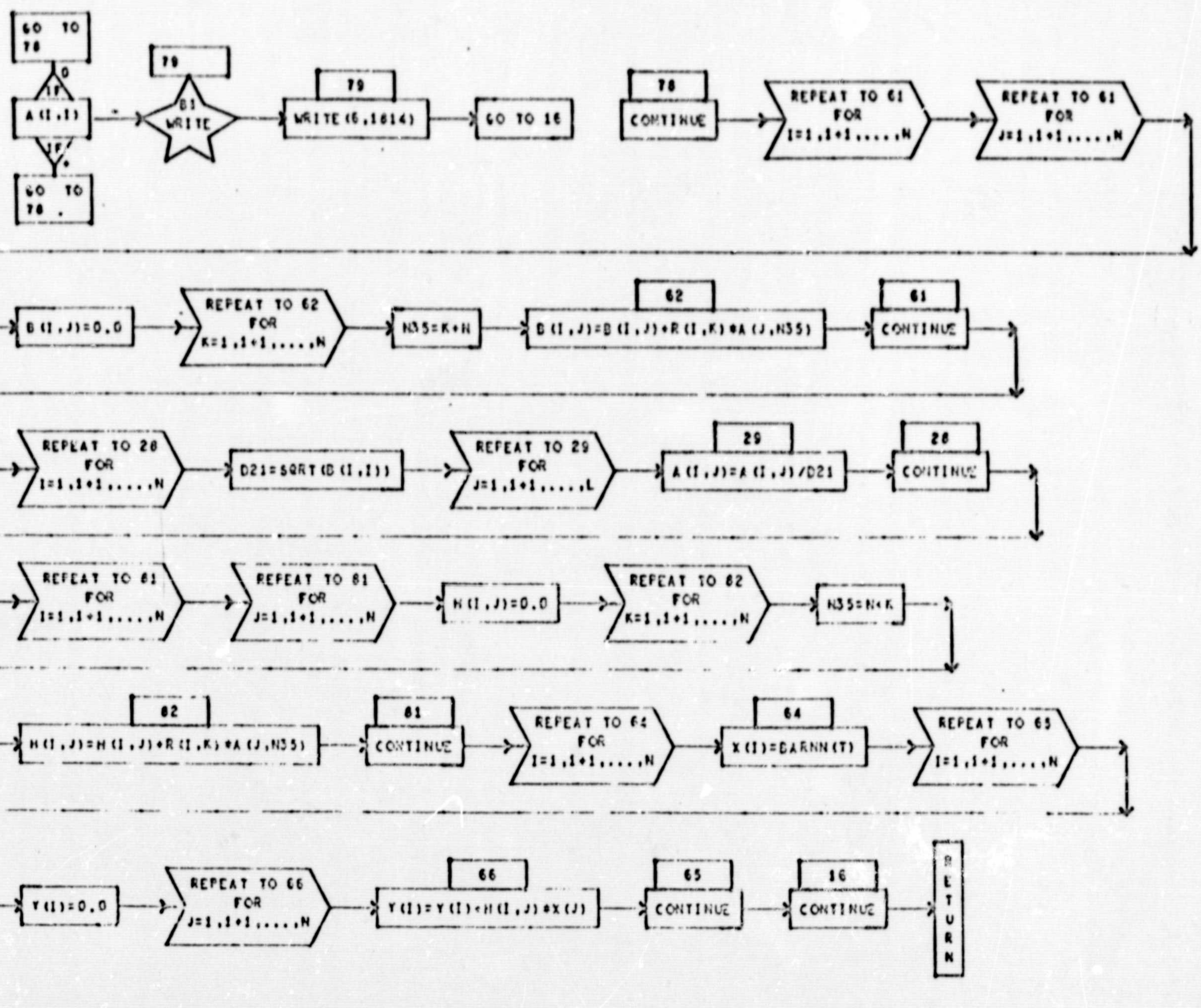
```
      IF(A(I,I)) 79,78,78
   79 WRITE(6,1814)
 1814  FORMAT(49H THE ORGINAL MATRIX IS NOT POSITIVE SEMI-DEFINITE)
      GO TO 16
   78 CONTINUE
      DO 61 I = 1,N
      DO 61 J = 1,N
      B(I,J) = 0.0
      DO 62 K = 1,N
      N35 = K + N
   62 B(I,J) = B(I,J) + R(I,K)*A(J,N35)
   61 CONTINUE
      DO 28 I = 1,N
      D21  = SQRT(B(I,I))
      DO 29 J = 1,L
   29 A(I,J) = A(I,J)/D21
   28 CONTINUE
C  PT  = TRANSPOSE OF P
C  COMPUTE     RPT
      DO 81 I=1,N
      DO 81 J=1,N
      H(I,J)= 0.0
      DO 82 K=1,N
      N35 = N + K
   82 H(I,J) = H(I,J) + R(I,K)*A(J,N35)
   81 CONTINUE
C  COMPUTE      Y =(RPT)*X
      DO 64 I = 1,N
   64 X(I) = RARNV(T)
      DO 65 I = 1,N
      Y(I) = 0.0
      DO 66 J = 1,N
   66 Y(I) = Y(I) + H(I,J)*X(J)
   65 CONTINUE
   16 CONTINUE
      RETURN
      END
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

SUBROUTINE GERN(N,R,Y,N)



REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.



APPENDIX B

N(O, 1) Random Number Generator

The N(O, 1) random number generator packet consist of the following subroutines.

(i)	BEGIN
(ii)	FIXB
(iii)	BARNN
(iv)	BARNA
(v)	LOOKUP
(vi)	BLOCK DATA
(vii)	BARN

All these subroutines are in FORTRAN IV, except BARN, which is in MAP. The first six subroutines in the packet were programed by K. Oney of Wolf Research and Development Corporation, Houston, Texas. The last subroutine, BARN, is a SHARE subroutine.

A listing of each of the subroutines follows.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

```
$IBFTC BEGIN    LIST
      SUBROUTINE BEGIN ( T )
      DIMENSION TABLE(90),TABLE1(135)
      COMMON /ONEY/ TABLE,TABLE1
C
      CALL TIME (I)
      CALL FIXB (I,XT)
      T = XT
      RETURN
      END


$IBFTC FIXB     LIST
      SUBROUTINE FIXB (JKO,TIM7)
C
      TIM = FLOAT(JKO)
      TIM1 = TIM * 1.E-5
      TIM2 = FLOAT(IFIX(TIM1))*1.E+5
      TIM3 = TIM - TIM2
      TIM4 = TIM3 * 1.E-3
      TIM5 = FLOAT(IFIX(TIM4))*1.0E+3
      TIM6 = TIM3 - TIM5
      TIM7 = TIM3*1.E-5 + TIM6*1.E-8
      RETURN
      END


$IBFTC BARNN    LIST
      FUNCTION BARNN ( T )
      XT = BARN(1.) + T
      WRITE(6,2)T,XT
      IF(XT .LT. 0.0)XT = ABS(XT)
    2 FORMAT(2(5X,E14.7) )
      IF(XT.GT. 1.0)GO TO 1
      CALL LOOKUP(XT,XG)
      BARNN = XG
      RETURN
    1 XT = XT - 1.0
      CALL LOOKUP (XT,XG)
      BARNN = XG
      RETURN
      END


$IBFTC LOOKUP   LIST
      SUBROUTINE LOOKUP (X,PHINV)
C
      DIMENSION TABLE(90),TABLE1(135)
      COMMON /ONEY/TABLE,TABLE1
C
      FLAG = 0.0
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

```
      FLAG1 = 0.0
      IF(X .LT. 0.1)GO TO 2
      IF(X .LT. 0.9)GO TO 5
      X = 1.0 - X
      FLAG1 = 1.0
      GO TO 2
    5 II = IFIX(X*100.)
      I = II - 7
      X0 = FLOAT(II) / 100.
      X1 = X0 + 0.01
      P = (X-X0) / (X1-X0)
    4 PS = P*P
      PC = PS*P
      PF = PC*P
      P5 = PF*P
      AM2 = 0.83333334E-2 * (P*6.0-PS*5.0-PC*5.0+PF*5.0-P5)
      AM1 = -0.41666667E-1 * (P*12.0-PS*16.0+PC+PF*4.0-P5)
      A0 = 0.83333334E-1 * (12.0-P*4.0-PS*15.0+PC*5.0+PF*3.0-P5)
      A1 = 0.83333334E-1 * (P*12.0+PS*8.0-PC*7.0-PF*2.0+P5)
      A2 = -0.41666667E-1 * (P*6.0+PS-PC*7.0-PF+P5)
      A3 = 0.83333334E-2 * (P*4.0-PC*5.0+P5)
      IF(FLAG .NE. 0.0)GO TO 3
      PHINV = AM2*TABLE(I-2) + AM1*TABLE(I-1) + A0*TABLE(I)
     1     + A1*TABLE(I+1) + A2*TABLE(I+2) + A3*TABLE(I+3)
      IF(FLAG1 .EQ. 0.0)GO TO 6
      PHINV = -PHINV
      X = 1.0 - X
    6 RETURN
    2 Z = ALOG(X)
      IX = IFIX(Z*10.) / 2
      X0 = (FLOAT(IX)*2.0 - 2.0) * 0.1
      X1 = X0 + 0.2
      P = (Z-X0) / (X1-X0)
      I = IABS(IX) - 7
      FLAG = 1.0
      GO TO 4
    3 PHINV = AM2*TABLE1(I+2) + AM1*TABLE1(I+1) + A0*TABLE1(I) +
     1     A1*TABLE1(I-1) + A2*TABLE1(I-2) + A3*TABLE1(I-3)
      IF(FLAG1 .EQ. 0.0)GO TO 7
      PHINV =-PHINV
      X = 1.0 - X
    7 RETURN
      END


$IBFTC BARNA    LIST
      FUNCTION BARNA(T)
      XT = BARN(1.) + T
      IF(XT .GT. 1.0)GO TO 1
      BARNA = XT
      RETURN
    1 BARNA = XT - 1.0
      RETURN
      END
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

```
$IBFTC BBLOC    LIST
      BLOCK DATA
      COMMON /ONE1/ TABLE,TABLE1
      DIMENSION TABLE(90),TABLE1(135)
      DATA (TABLE1(I),I=1,40) /
     X -0.14445822E 01,-0.15480089E 01,-0.16469216E 01,-0.17419096E 01,
     X -0.18333769E 01,-0.19216686E 01,-0.20070761E 01,-0.20898409E 01 ,
     X -0.97291052E 00,-0.11015135E 01,-0.12222678E 01,-0.13363469E 01,
     X -0.21702071E 01,-0.22463373E 01,-0.23244072E 01,-0.23985641E 01,
     X -0.24709386E 01,-0.25416478E 01,-0.26107965E 01,-0.26784793E 01,
     X -0.27447826E 01,-0.28097820E 01,-0.28735504E 01,-0.29361520E 01,
     X -0.29976459E 01,-0.30580858E 01,-0.31175251E 01,-0.31760073E 01,
     X -0.32335765E 01,-0.32902727E 01,-0.33461332E 01,-0.34011926E 01,
     X -0.34554832E 01,-0.35090356E 01,-0.35618778E 01,-0.36140367E 01,
     X -0.36655375E 01,-0.37164033E 01,-0.37666569E 01,-0.38163197E 01 /
      DATA (TABLE1(I),I=41,80) /
     X -0.38654009E 01,-0.39139462E 01,-0.39619492E 01,-0.40094320E 01 ,
     X -0.40564132E 01,-0.41029076E 01,-0.41489291E 01,-0.41944917E 01,
     X -0.42396085E 01,-0.42842922E 01,-0.43285543E 01,-0.43724068E 01,
     X -0.44158600E 01,-0.44589247E 01,-0.45016109E 01,-0.45439291E 01,
     X -0.45858856E 01,-0.46274920E 01,-0.46687561E 01,-0.47096856E 01,
     X -0.47502837E 01,-0.47905725E 01,-0.48305453E 01,-0.48702127E 01,
     X -0.49095825E 01,-0.49486693E 01,-0.49874531E 01,-0.50259663E 01,
     X -0.50642065E 01,-0.51021785E 01,-0.51398894E 01,-0.51773410E 01,
     X -0.52145419E 01,-0.52514954E 01,-0.52882068E 01,-0.53246804E 01,
     X -0.53609209E 01,-0.53969324E 01,-0.54327192E 01,-0.54682855E 01 /
      DATA (TABLE1(I),I=81,120) /
     X -0.55036353E 01,-0.55387720E 01,-0.55736998E 01,-0.56084222E 01,
     X -0.56429429E 01,-0.56772649E 01,-0.57113916E 01,-0.57453267E 01,
     X -0.57790729E 01,-0.58126335E 01,-0.58460113E 01,-0.58792093E 01,
     X -0.59122305E 01,-0.59450774E 01,-0.59777527E 01,-0.60102589E 01,
     X -0.60425989E 01,-0.60747752E 01,-0.61067899E 01,-0.61386456E 01,
     X -0.61703447E 01,-0.62018891E 01,-0.62332813E 01,-0.62645233E 01,
     X -0.62956174E 01,-0.63265656E 01,-0.63573699E 01,-0.63880320E 01,
     X -0.64185540E 01,-0.64489380E 01,-0.64791855E 01,-0.65092985E 01,
     X -0.65392786E 01,-0.65691276E 01,-0.65988477E 01,-0.66284402E 01,
     X -0.66579051E 01,-0.66872456E 01,-0.67164648E 01,-0.67455636E 01 /
      DATA (TABLE1(I),I=121,135) /
     X -0.67745407E 01,-0.68033974E 01,-0.68321396E 01,-0.68607676E 01,
     X -0.68892795E 01,-0.69176753E 01,-0.69459645E 01,-0.69741438E 01,
     X -0.70022125E 01,-0.70301756E 01,-0.70580284E 01,-0.70857950E 01,
     X -0.71134456E 01,-0.71409942E 01,-0.71684439E 01   /
      DATA (TABLE(I),I=1,40) /
     X -0.14050715E 01,-0.13467549E 01,-0.12815514E 01,-0.12265291E 01,
     X -0.11749867E 01,-0.11263911E 01,-0.10803193E 01,-0.10364334E 01,
     X -0.99445788E 00,-0.95416525E 00,-0.91536509E 00,-0.87789629E 00,
     X -0.84162125E 00,-0.80642124E 00,-0.77219321E 00,-0.73884695E 00,
     X -0.70630256E 00,-0.67449974E 00,-0.64334540E 00,-0.61281229E 00,
     X -0.58284150E 00,-0.55333471E 00,-0.52440051E 00,-0.49585035E-00,
     X -0.46764880E-00,-0.43991316E-00,-0.41246313E-00,-0.38532045E-00,
     X -0.35846879E-00,-0.33155335E-00,-0.30549079E-00,-0.27931993E-00,
     X -0.25334710E-00,-0.22754497E-00,-0.20189348E-00,-0.17637416E-00,
     X -0.15096921E-00,-0.12566134E-00,-0.10043371E-00,-0.75260862E-01 /
      DATA (TABLE(I),I=41,80) /
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

```
X -0.50153583E-01,-0.25064907E-01, 0.            , 0.25064907E-01,
X  0.50153583E-01, 0.75269862E-01, 0.10043371E-00, 0.12566134E-00,
X  0.15096921E-00, 0.17637416E-00, 0.20189348E-00, 0.22754497E-00,
X  0.25334710E-00, 0.27931903E-00, 0.30548079E-00, 0.33185335E-00,
X  0.35845879E-00, 0.38532045E-00, 0.41246313E-00, 0.43991316E-00,
X  0.46769880E-00, 0.49585035E-00, 0.52440051E 00, 0.55338471E 00,
X  0.58284150E 00, 0.61281299E 00, 0.64334540E 00, 0.67448974E 00,
X  0.70630256E 00, 0.73884685E 00, 0.77219321E 00, 0.80642124E 00,
X  0.84162123E 00, 0.87789629E 00, 0.91536509E 00, 0.95416525E 00,
X  0.99445788E 00, 0.10364334E 01, 0.10803193E 01, 0.11263911E 01 /
 DATA (TABLE(I),I=81,90) /
X  0.11749867E 01, 0.12265281E 01, 0.12815514E 01, 0.13407549E 01,
X  0.14050715E 01, 0.14757910E 01, 0.15547734E 01, 0.16448536E 01,
X  0.17506859E 01, 0.18807936E 01                                 /
 END
```

```
$IBMAP RARN
        ORG     4096
RARN    TMI     RDNN
RDUN    LDQ     RDUN+12,0
        MPY     RDUN+13,0
        LLS     4,0
        ALS     4,0
        LRS     4,0
        STQ     RDUN+12,0
        ADD     RDUN+12,0
        STO     RDUN+12,0
        ARS     4,0
        ORA     RDUN+14,0
        FAD     RDUN+14,0
        TRA     1,4
        OCT     2312421537,1737,2200000000,0
RDNN    SXD     IR1,1
        SXD     IR4,4
        LXA     L20,1
        CLA     RDUN+15,0
        STO     C,0
        TSX     RDUN,4
        FAD     C,0
        STO     C,0
        TIX     RDUN+5,1,1
        FDP     L20+1,0
        CLA     L20+4,0
        LLS     35,0
        FSB     L20+2,0
        LRS     35,0
        FMP     L20+3,0
        LXD     IR1,1
        LXD     IR4,4
        TRA     1,4
L20     HTR     20,0
        DEC     20.,.5,15.49193340,0
IR1     HTR     0,0
IR4     HTR     0,0
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

```
C         HTR       0,0
          END
```