FATIGUE CRACK GROWTH MODEL
RANDOM2 USER MANUAL

Prepared by:

Lola Boyce, Ph.D., P.E.
Thomas B. Lovelace

APPENDIX 1
of Annual Report
of Project Entitled
Development of Advanced Methodologies
for Probabilistic Constitutive Relationships
of Material Strength Models

NASA Grant No. NAG 3-867

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Cleveland, OH 44135

The Division of Engineering
The University of Texas at San Antonio
San Antonio, TX 78285
January, 1989
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2.0 Theoretical Background</td>
<td>2</td>
</tr>
<tr>
<td>3.0 Input Data</td>
<td>4</td>
</tr>
<tr>
<td>4.0 Sample Problem</td>
<td>6</td>
</tr>
<tr>
<td>5.0 References</td>
<td>9</td>
</tr>
<tr>
<td>6.0 Appendix A: Physical Quantities, Symbols, and Units</td>
<td>10</td>
</tr>
<tr>
<td>7.0 Appendix B: RANDOM2 Sample Problem: Source, Input and Output Files</td>
<td>11</td>
</tr>
<tr>
<td>8.0 Appendix C: IMSL Subroutine Calls from RANDOM2</td>
<td>33</td>
</tr>
<tr>
<td>9.0 Appendix D: SAS/GRAPH Program for RANDOM2</td>
<td>34</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

This User Manual documents the FORTRAN program RANDOM2. RANDOM2 is based on fracture mechanics using a probabilistic fatigue crack growth model. It predicts the random lifetime of an engine component to reach a given crack size (see Section 2.0, Theoretical Background).

Included in this Manual are details regarding the theoretical background of RANDOM2, input data instructions and a sample problem illustrating the use of RANDOM2. Appendix A gives information on the physical quantities, their symbols, FORTRAN names, and both SI and U.S. Customary units. Appendix B includes photocopies of the actual computer printout corresponding to the sample problem. Appendices C and D detail the IMSL, Ver. 10 1, subroutines and functions called by RANDOM2 and a SAS/GRAPH 2 program that can be used to plot both the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.).
2.0 THEORETICAL BACKGROUND

Fatigue crack growth data are usually presented as cycles, N, to reach a particular crack length, a. The initial crack size is a_i. It is generally accepted that under constant amplitude alternating stress, fatigue crack growth can be related to stress intensity through a first order differential equation: \[ \frac{d}{dN} = C(\Delta K)^m \] (1)

where C is a material parameter, m is a material property (often a constant) and \( \Delta K \) is the stress intensity range. Stress intensity range is given by

\[ \Delta K = Y \Delta \sigma \sqrt{a} \]

where Y is a constant dependent upon component and crack geometry and \( \Delta \sigma \) is the constant amplitude alternating stress. Therefore, equation (1) can be written as

\[ \frac{d}{dN} = C(Y \Delta \sigma \sqrt{a})^m \]

or,

\[ \frac{d}{dN} = C Y^m \Delta \sigma^m \pi^{m/2} a^{-m/2} \] (2)

Equation (2) can be integrated, from the initial crack length, \( a_i \), to the final crack length, \( a_f \), to yield N, the number of cycles. The result is

\[ N = \frac{1}{CY^m \pi^{m/2} \Delta \sigma^m} \left[ \frac{a_f^{m/2} + 1 - a_i^{m/2}}{-m/2 + 1} \right] \] (3)

Thus, equation (3) gives the "cycles to reach a given crack length."

Metallurgical evidence indicates that casting pores play a significant role in the high-cycle fatigue life of cast nickel base-superalloys, especially at high temperatures. The location and size of these fatigue crack-initiating pores vary greatly from one aerospace propulsion system component to another. This accounts for the large variability in fatigue life and leads to consideration of fatigue crack growth as a random phenomenon.

Fatigue life directly relates to casting pore size, and pore size can be used to determine initial crack size, \( a_i \). Thus, utilizing principles of both probabilistic analysis and fatigue crack growth, a quantitative probabilistic constitutive relationship between fatigue life and fracture mechanics parameters can be developed. Using the "randomized equation" approach, the fatigue crack growth model, given by equation (3) has the following form:

\[ N = f(C,m,\Delta \sigma,a_i,a_f,Y) \] (4)
or, in general,

\[ N = f(X_i), \ i = 1, \ldots, 6, \]  

(5)

where the \( X_i \) are the six independent variables in equations (3) and (4). Equation (3) is "randomized" by assuming the first four variables in equation (4) to be random. Assuming a small crack in a relatively large component leads to assuming \( Y = 1.0 \), a deterministic value. A deterministic final crack size was chosen since experimental evidence indicated that it was relatively unimportant.\(^3\)

Probabilistic analysis, via simulation, yields the distribution of the dependent random variable, cycles, \( N \). A probability density function (p.d.f.) of cycles is generated using the maximum penalized likelihood method. Maximum penalized likelihood generates the p.d.f. estimate using the method of maximum likelihood together with a penalty function to smooth it.\(^5\)
3.0 INPUT DATA

Data input for RANDOM2 is user friendly and easy to manipulate (see, for example, the file entitled NORMAL.INP, in Section 4.0). The first five lines of input have the same format, namely 2E12.4, and the last two lines differ. The last two lines of input have the formats 13,2X,13,2X,2E12.4,2X,I3 and I3, respectively. A brief line by line description is given along with an example for each line (Note: the ruler is to aid the user in formatting and is not a part of the input). A table listing the physical quantities, their units and symbols is given in Appendix A.

1. Random Number Generator Seed, ISEED, and Sample Size, NTOT

EXAMPLE:

```
123456789012345678901234567890
1   40
```

2. Material Property, RMM

EXAMPLE:

```
123456789012345678901234567890
28 0E-01  1.4E-01
```

3. Initial Crack Size (Pore Diameter), RAI

EXAMPLE:

```
123456789012345678901234567890
300 0E-06  45.0E-06
```

4. Material Property, RCC

EXAMPLE:

```
123456789012345678901234567890
2.20E-11 0.22E-11
```

5. Stress Range, DELSIG

EXAMPLE:

```
123456789012345678901234567890
6.2E+02  6.2E+01
```
6. The DESPL parameters are NODE, INIT, ALPHA, EPS, MAXIT and are entered in that order as follows:

**EXAMPLE:**

\[
\begin{array}{cccccc}
1234567890123456789012345678901234567890 & \\
21 & 0 & 50.0E-01 & 10.0E-05 & 30 & \\
\end{array}
\]

7. The DESPL parameter, IOPT, is entered as follows:

**EXAMPLE:**

\[
\begin{array}{ccc}
1234567890 & \\
2 & \\
\end{array}
\]
4.0 SAMPLE PROBLEM FOR RANDOM2

The objective of this program is to predict the random lifetime, to reach a given crack size for an engine component. The theory is based on fracture mechanics, using a probabilistic fatigue crack growth model (see Section 2.0, Theoretical Background). RANDOM2 input parameters are given in Table A1.1. Note that the first four parameters are random. Their means and standard deviations are input by the user. The last two parameters, AF and YY, are deterministic and are fixed internally by the program. They are equal to the values shown in Table A1.1.

Table A1.1 RANDOM2 sample problem input (SI units)

<table>
<thead>
<tr>
<th>FORTRAN Name</th>
<th>Distribution Type</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMM</td>
<td>normal</td>
<td>28.0E-01</td>
<td>1.4E-01 (5%)</td>
</tr>
<tr>
<td>AI</td>
<td>lognormal</td>
<td>300.0E-06</td>
<td>45.0E-06 (15%)</td>
</tr>
<tr>
<td>RCC</td>
<td>lognormal</td>
<td>2.20E-11</td>
<td>0.22E-11 (10%)</td>
</tr>
<tr>
<td>DELSIG</td>
<td>lognormal</td>
<td>6.2E+02</td>
<td>6.2E+01 (10%)</td>
</tr>
<tr>
<td>AF</td>
<td>N/A</td>
<td>2.0E-03</td>
<td>N/A</td>
</tr>
<tr>
<td>YY</td>
<td>N/A</td>
<td>1.0</td>
<td>N/A</td>
</tr>
</tbody>
</table>

The input is entered in the following format in a file entitled NORMAL.INP.

```
1234567890123456789012345678901234567890
1 40
28.0E-01 1.4E-01
300.0E-06 45.0E-06
2.20E-11 0.22E-11
6.2E+02 6.2E+01
21 0 50.0E-01 10.0E-05 30
2
```
Execution of RANDOM2 (source code entitled NR2.FOR) produces an output file entitled RANDM22 giving intermediate results (see Appendix B). Execution also produces the plotfiles OUT1 and OUT2 (see Appendix B). These files are used to plot the X and Y axes of the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.), respectively, generated by RANDOM2. The plots are drawn from the plotfiles by the SAS/GRAPH graphing program (see Appendix C). These plots for the sample problem are shown in Figures A1.1 and A1.2.

This same sample problem has been reported in Boyce and Charnis.6 There, however, it utilized U.S. Customary units and an older version of RANDOM2 (IMSL Version 9.2 subroutines).

![Probability Density Function](image)

**Fig. A1.1** p.d.f. of log of mechanical cycles for fatigue crack growth model, using maximum penalized likelihood.
Fig. A1.2  c.d.f. of log of mechanical cycles for fatigue crack growth model, using maximum penalized likelihood.
5.0 REFERENCES


6.0 APPENDIX A

PHYSICAL QUANTITIES, SYMBOLS, AND UNITS

The physical quantities, their symbols, and units for the fatigue crack growth model are given in the following table.

Table A1.2 Physical quantities, symbols, and units for fatigue crack growth model for RANDOM2

<table>
<thead>
<tr>
<th>Physical Quantity</th>
<th>Theory Symbol</th>
<th>FORTRAN Name</th>
<th>SI Units</th>
<th>U.S. Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material Property</td>
<td>m</td>
<td>RMM</td>
<td>m/cycle/M Pa m</td>
<td>in/cycle/ksi in</td>
</tr>
<tr>
<td>Initial Crack Size</td>
<td>A_i</td>
<td>RAI</td>
<td>m</td>
<td>in</td>
</tr>
<tr>
<td>Material Property</td>
<td>C</td>
<td>RCC</td>
<td>m/cycle</td>
<td>in/cycle</td>
</tr>
<tr>
<td>Alternating Stress</td>
<td>Δσ</td>
<td>DELSIG</td>
<td>M Pa</td>
<td>ksi</td>
</tr>
<tr>
<td>Final Crack Size</td>
<td>A_f</td>
<td>AF</td>
<td>m</td>
<td>in</td>
</tr>
<tr>
<td>Geometry Dependent</td>
<td>Y</td>
<td>YY</td>
<td>(dimensionless)</td>
<td></td>
</tr>
</tbody>
</table>
7.0 APPENDIX B

SAMPLE PROBLEM: SOURCE, INPUT AND OUTPUT FILES
CALL RNSET(ISET)
CALL RNLNL(NTOT, YM, YS, RCC)
WRITE(6,201)
201 FORMAT(' MATERIAL PROPERTY, C', C)
WRITE(6,1001) (RCC(I), I=1, NTOT)
C LOGNORMAL STRESS RANGE, DELSIG
WRITE(6,1002) ISEED, NTOT
READ (5,1011) XM, XS
WRITE(6,1011) XM, XS
XS = 0.02E0
YM = 0.02E0
YS = LOG10((1.0+(XS/XM)**2))
YM = LOG10(XM) - 0.5*YS**2
CALL RNSET(ISET)
CALL RNLNL(NTOT, YM, YS, DELSIG)
WRITE(6,2022)
2022 FORMAT(' STRESS RANGE, DELSIG')
WRITE(6,1001) DELSIG(I), I=1, NTOT)
C DEFINE DETERMINISTIC PARAMETERS
C PI
PI = 3.1415926535897932364626433
C COMPONENT AND CRACK SHAPE PARAMETER, Y1
Y1 = 1.0
C FINAL CRACK SIZE: AF
AF = 2.0E-03
C CALCULATE CYCLES TO REACH CRACK SIZE 2.0E-03M
DO 101 I=1, NTOT
XNF1 = 1.0/(RCC(I)**Y1/RMM(I)**2)*DEL5(I)**2
RMM(I) = ((AF**1.*RMM(I)/2.)*R5(I)**2*(1.0-RMM(I)/2.))/
1*(1.0-RMM(I)/2.))
XNF1 = XM**XNF2
C CALCULATE LOG OF CYCLES TO REACH CRACK SIZE 2.0E-03M
XNF(I) = LOG10(XNF1(I))
101 CONTINUE
WRITE(6,2023)
2023 FORMAT(' LOG OF CYCLES TO REACH CRACK SIZE=2.0E-03M,','/
1* GIVN STRESS MEAN AMPLITUDE=6.2E+02MPA')
WRITE(6,1001) XNF(I), I=1, NTOT)
C SORT LOG OF CYCLES
CALL SORT(XNF, NTOT)
WRITE(6,2024)
2024 FORMAT(' SORTED LOG OF CYCLES')
WRITE(6,1001) XNF(I), I=1, NTOT)
C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C CALCULATE PDF OF LOG OF CURRENT CYCLES; LOG XNF
READ(5,1009) NODE, INIT, ALPHA, EPS, MAXIT
WRITE(6,985)
985 FORMAT(' DESPL PARAMETERS')
WRITE(6,990) NODE, INIT, ALPHA, EPS, MAXIT
BNDS(I) = XNF(I) - 0.05*XNF(1)
BNDS(2) = XNF(NTOT) + 0.05*XNF(NTOT)
WRITE(6,979) BNDS(I), BNDS(2)
979 FORMAT('BNDS(I), BNDS(2)')
E12 = 4.0, 12
CALL DESPL(NTOT, XNF, NODE, BNDS, INIT, ALPHA, MAXIT, EPS, DENS, STAT
1NNIS)
1NNIS)
980 FORMAT('PDF OF LOG OF CURRENT CYCLES; LOG XNF, Y AXIS OF PDF PLOT')
WRITE(6,990) (DENS(I), I=1, NODE)
981 FORMAT(' OUTPUT STATISTICS')
WRITE(6,981) (STAT(I), I=1, 4)
982 FORMAT(' NUMBER OF MISSING VALUES')
WRITE(6,1010) NMISS

CALCULATE WINDOW WIDTH, HH

HH=(BNDS(2)-BNDS(1))/(NODE-1)

CALCULATE VALUES OF LOG OF CURRENT CYCLES AT WHICH PDF IS ESTIMATED;

ALSO CALLED 'NODE' VALUES

DO 6001 I=1,NODE-2

6001 CONTINUE

983 FORMAT(' LOG OF CURRENT CYCLES, LOG XNF')

WRITE(6,1001)(BNDS(I),I=1,NODE)

REORDER BNDS FOR PLOTTING

SAVE1 = BNDS(2)
SAVE2 = BNDS(NODE)
BNDS(NODE)=BNDS(2)
DO 6002 I=1,NODE-2
BNDS(I)=BNDS(I+2)

6002 CONTINUE

984 FORMAT(' ORDERED LOG OF CURRENT CYCLES, LOG XNF,
1X AXIS PDF, CDF PLOT')

WRITE(6,1001)(BNDS(I),I=1,NODE)

WRITE LOG OF CURRENT CYCLES AND PDF OF LOG OF CURRENT CYCLES,

LOG XNF TO PLOT FILES

WRITE(34,990)

990 FORMAT(E12.4,1X,E12.4)

WRITE(34,991)(BNDS(J),DENS(J),J=1,NODE)

CALCULATE CDF OF LOG OF CURRENT CYCLES

READ(5,1010) I0FT

WRITE(6,992)

NODS(1) = I0FT

DO 6003 I=1,NODE

6003 CONTINUE

WRITE(6,993)

993 FORMAT(' ORDERED LOG OF CURRENT CYCLES, LOG XNF,
1X AXIS OF PDF, CDF PLOT')

WRITE(6,1001)(BNDS(I),I=1,NODE)

WRITE(6,1001)(BNDS(X),I=1,NODE)

WRITE LOG OF CURRENT CYCLES AND CDF OF LOG OF CURRENT

TO THE PLOT FILES

OF POOR QUALITY
WRITE(35,990)
WRITE(35,991)(BNDS(J),DISTX(J),J=1,NODE)
STOP
END

DIMENSION Y(10000)

Y IS THE ARRAY TO BE SORTED
C AT COMPLETION Y(I) IS SMALLEST VALUE
C AT COMPLETION Y(N) IS LARGEST VALUE
I = N - 1
DO 1 I=1,N
J = I + 1
DO 2 K=J,N
IF (Y(I),LT,Y(K)) GO TO 2
TEMP = Y(I)
Y(I) = Y(K)
Y(K) = TEMP
2 CONTINUE
1 CONTINUE
RETURN

IMSL Name: DISPL/DDISP (Single/Double precision version)

Computer: IBM/SINGLE

Revised: November 1, 1985

Purpose: Nonparametric probability density function estimation by the penalized likelihood method.

Usage: CALL DISPL (NOBS, X, NODE, BNDS, INIT, ALPHA, MAXIT, EPS,
DENS, STAT, HESS, LDHESS, ILOHI, ILOEN, B, IFUP, WA2)

Arguments:
NOBS - Number of observations. (Input)
X - Vector of length NOBS containing the random sample of
   responses. (Input)
NODE - Number of mesh nodes for the discrete pdf estimate.
   (Input)
BNDS - Vector of length 2 containing the minimum and maximum
   values for X(i) in BNDS(1) and BNDS(2), respectively.
   (Input)
INIT - Initialization option. (Input)
ALPHA - Positive penalty weighting factor which controls the
   smoothness of the estimate. (Input)
MAXIT - Maximum number of iterations allowed in the iterative
   procedure. (Input)
EPS - Convergence criterion. (Input)
DENS - Vector of length NODE containing the estimated values of
   the discrete pdf at the NODE equally spaced mesh nodes.
   (Input/output if INIT=1; Output otherwise)
STAT - Vector of length 4 containing output statistics. (Output)
   STAT(1) and STAT(2) contain the log-likelihood and the
   log-penalty terms, respectively. STAT(3) and STAT(4)
   contain the estimated mean and variance for the
   estimated density.
HESS - Seven by NODE-2 hessian matrix (and its factorization).
   (Output)
LDHESS - Leading dimension of HESS exactly as specified in the
   dimension statement in the calling program. (Input)
ILOHI - NODE by 2 matrix containing the indices for the risk set

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE DIPSL (NODS, X, NODE, BNDS, INIT, ALPHA, MAXIT, EPS,
    DENS, STAT, HESS, LDHESS, ILOHI, DENEST, B,
    IPUT, WK2)
C
INTEGER NODS, NODE, INIT, MAXIT, LDHESS, ILOHI(NODE),*
C
REAL ALPHA, EPS, X(*), BNDS(2), DENS(*), STAT(4),
    HESS(LDHESS), DENEST(NODE), B(*), WK2(*)
C
SPECIFICATIONS FOR LOCAL VARIABLES
C
INTEGER I, IMTR, IPT, ITER, K, KM1, K2, KP1, KP2, M, M0LD,
C
REAL BK, BM1, BMALL, CK, CM1, CM2, CCMCM, CKP1, CKP2,
    CMS, EPS, FACTOR, FK, FMI1, FMI2, FK1, H, H2, H3,
    SUM, TEMP, WK(4)
C
DOUBLE PRECISION SUM1, SUM2, SUM3
C
INTEGER MINTC(I)
C
SAVE MINTC
C
SPECIFICATIONS FOR INTRINSICS
C
EXTERNAL EIMES, EIPOP, EIPSH, EIPT, EISTF, SABD, SAXP
    SCOP, SHPRD, SSDL, CSCF, LSTR, LFPRD
C
SPECIFICATIONS FOR FUNCTIONS
C
EXTERNAL ISMIN, MIRCD, S0D7, SNNM2, SNUM
C
INTEGER S0D7, SNNM2, SNUM
C
DATA MINTC/5, 9, 17, 33, 65, 129, 253, 100001/
C
CALL EIPSH ("DIPSL")
C
Error checks

NER = 7
IF (NODS .LT. 1) THEN
    CALL EIMES (5, 1, 'After removing all missing (NaN) not a
        number) values from X there are no valid
        observations. At least one valid observation
        is necessary.')
END IF
IF (NODE .LE. 4) THEN
    CALL EISTF (1, NODE)
    CALL EIMES (5, 2, 'NODE = %I(1). The number of mesh
        nodes, NODE, must be an odd integer greater
        ');
M = 3
ELSE
    M = NODE
END IF
C
20. IF (INIT .EQ. 0) THEN
    MODL = M
    IMPTR = IMPTR + 1
    M = MIN(MODEL, MINCR(IMPTR))
END IF
C
30. IF (M < 2) THEN
    M = 2
END IF
C
H = (BNDS(2) - BNDS(1))/(M - 1)
H2 = H**2
H3 = H2**3
C
IF (INIT .NE. 0) THEN
    CALL SQUERY (NODE, 1.0/(H*SUM(NODE, DENS, 1)), DENS, 1)
C
50. IF (INIT .EQ. 0) THEN
    CALL D3S3PT (M, B(2), 1, MOLD, BNDS, DENS, DENEST, Wk, Wk)
    TEMP = 1.0/(H**M)
    DO 60 I=2, M
        DENS(I) = AMAX1(TEMP, SQRT(DENEST(I-1, 1)))
        CONTINUE
    END IF
C
IF (INIT .EQ. 0) THEN
    CALL D3S3PT (M, B(2), 1, MOLD, BNDS, DENS, DENEST, Wk, Wk)
    TEMP = 1.0/(H**M)
    DO 60 I=2, M
        DENS(I) = AMAX1(TEMP, SQRT(DENEST(I-1, 1)))
        CONTINUE
    ELSE
        DO 90 I=2, M-1
            DENS(I) = SORT(DENS(I))
            CONTINUE
    END IF
C
DO 140 ITER = 1, MAXIT
    HESS(1, 1) = 0.0
    HESS(2, 2) = 0.0
    HESS(3, 3) = 0.0
    PSML = 0.0
    SUM = 0.0
    CH** are true estimates = F**2

C
END
CALL SAXPY (M-2, CONS, DENS(1), 1, DENS(1), 1)

CALL SAXPY (M-2 -1.0, DENS(1), 1, DENS(2), 1)

Check the convergence criterion

IF (SRM2(M-2, DENS(2)), 1)

TEMP = TEMP1.0E-4/SORT(M-2.0)

IF (130 1=2, M - 1)

DENS(I) = AMAX1(TEMP, DENS(I))

CONTINUE

CALL ESUM (1, MAXIT)

IF (ESUM = MAXIT) 'The maximum number of iterations exceeded.'

CALL SHPREP (M-2, DENS(2), 1, DENS(2), 1)

Replace DENS(2) with squares

IF (M .NE. NODE) GO TO 20

Evaluate log likelihood and penalty

SUM1 = 0.0

Penalty

DO 160 K = 2, M-

KPK = MAX(K, M)

SUM1 = SUM1 + (DENS(KP1) - 2.0*DENS(K) + DENS(KP1))**2

160 CONTINUE

STAT(2) = -0.5*FACT0*SUM1

SUM2 = 0.0

Loss-likelihood

DO 170 I = 1, NODE

IF (X(I) > BNDS(2)) THEN

CALL D5SFT (1, X(I), 1, NODE, BNDS, DENS, DENS, WK, WK)

170 CONTINUE

SUM2 = SUM2 + ALOG(DENS1(1))

END IF

Evaluate M.L.F.E. mean and variance

STAT(1) = SUM2

SUM1 = 0.0

SUM2 = 0.0

DO 180 K = 1, K - 1

F = DENS(K)

FPK = DENS(KP1)

B = B(K)

CONS = F + FPK

SUM1 = SUM1 + H1*CONS/6.0 + 0.5*CONS**2

SUM2 = SUM2 + H2*CONS/12.0 + H2*CONS/3.0 +

0.5*CONS

180 CONTINUE

STAT(3) = SUM1

STAT(4) = SUM2 - SUM1*SUM1

Exit section

9000 CALL EIPOP ('D3SFL '

RETURN

/EDF
8.0 APPENDIX C
IMSL SUBROUTINE CALLS FROM RANDOM2

1. RNSET - Initializes a random seed for use in the IMSL random number generators.
2. RNNOR - Generates pseudorandom numbers from a standard normal distribution using an inverse CDF method.
3. RNLNL - Generates pseudorandom numbers from a lognormal distribution.
4. DESPL - Performs nonparametric probability density function estimation by the penalized likelihood method.
5. GCDF - Evaluates a general continuous cumulative distribution function given ordinates of the density.
9.0 APPENDIX D
SAMPLE SAS/GRAPH (VER. 5.16) PROGRAM FOR RANDOM2

data a;
INFILE 'OUT1.CPR' FIRSTOBS=2; input x y;
GOPTIONS DEVICE=HP7470;
proc gplot;
   axis1 label=(h=1 f=simplex 'LOG OF CYCLES')
          value=(h=1 f=simplex);
   axis2 value=(h=1 f=simplex) label=none;
   plot y*x / haxis=axis1 vaxis=axis2;
TITLE H=1 A=90 F=SIMPLEX 'PROBABILITY DENSITY FUNCTION';
symbol i=spline v=square;
data B;
INFILE 'OUT2.CPR' FIRSTOBS=2; input x y;
proc gplot;
   axis1 label=(h=1 f=simplex 'LOG OF CYCLES')
          value=(h=1 f=simplex);
   axis2 value=(h=1 f=simplex) label=none;
   plot y*x / haxis=axis1 vaxis=axis2;
TITLE H=1 A=90 F=SIMPLEX 'CUMULATIVE DISTRIBUTION FUNCTION';
symbol i=spline v=square;