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

https://ntrs.nasa.gov/search.jsp?R=19890014519 2020-04-16T08:12:29+00:00Z
https://ntrs.nasa.gov/search.jsp?R=19890014519 2020-04-16T08:12:29+00:00Z
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:\(^3\)

\[
\frac{da}{dN} = C(\Delta K)^m
\]

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

\[
\Delta K = Y\Delta\sigma/\pi a
\]

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

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

or,

\[
\frac{da}{dN} = C Y^m\Delta\sigma^m\pi^{-m/2}a^{-m/2}.
\]

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

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

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.\(^4\) 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:

\[
N = f(C,m,\Delta\sigma,a_i,a_f,Y)
\]
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 I3,2X,I3,2X,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:

\[
\begin{array}{cccc}
123456789012345678901234567890 & 1 & 40 \\
\end{array}
\]

2. Material Property, RMM

EXAMPLE:

\[
\begin{array}{cccc}
123456789012345678901234567890 & 28.0E-01 & 1.4E-01 \\
\end{array}
\]

3. Initial Crack Size (Pore Diameter), RAI

EXAMPLE:

\[
\begin{array}{cccc}
123456789012345678901234567890 & 300.0E-06 & 45.0E-06 \\
\end{array}
\]

4. Material Property, RCC

EXAMPLE:

\[
\begin{array}{cccc}
123456789012345678901234567890 & 2.20E-11 & 0.22E-11 \\
\end{array}
\]

5. Stress Range, DELSIG

EXAMPLE:

\[
\begin{array}{cccc}
123456789012345678901234567890 & 6.2E+02 & 6.2E+01 \\
\end{array}
\]
6. The DESPL parameters are NODE, INIT, ALPHA, EPS, MAXIT and are entered in that order as follows:

EXAMPLE:

1234567890123456789012345678901234567890
21 0 50.0E-01 10.0E-05 30

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

EXAMPLE:

1234567890
2
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 (Value)</th>
<th>(% of Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMM</td>
<td>normal</td>
<td>28.0E-01</td>
<td>1.4E-01</td>
<td>(5%)</td>
</tr>
<tr>
<td>AI</td>
<td>lognormal</td>
<td>300.0E-06</td>
<td>45.0E-06</td>
<td>(15%)</td>
</tr>
<tr>
<td>RCC</td>
<td>lognormal</td>
<td>2.20E-11</td>
<td>0.22E-11</td>
<td>(10%)</td>
</tr>
<tr>
<td>DELSIG</td>
<td>lognormal</td>
<td>6.2E+02</td>
<td>6.2E+01</td>
<td>(10%)</td>
</tr>
<tr>
<td>AF</td>
<td>N/A</td>
<td>2.0E-03</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>YY</td>
<td>N/A</td>
<td>1.0</td>
<td>N/A</td>
<td></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 Chami.6 There, however, it utilized U.S. Customary units and an older version of RANDOM2 (IMSL Version 9.2 subroutines).

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>Ai</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>Af</td>
<td>AF</td>
<td>m</td>
<td>in</td>
</tr>
<tr>
<td>Geometry Dependent Constant</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
WRITE(6,1010)NNISS

1 CALCULATE WINDOW WIDTH, HH
   HH=(BNDS(2)-BNDS(1))/(NODE-1)

2 CALCULATE VALUES OF LOG OF CURRENT CYCLES AT WHICH PDF IS ESTIMATED
   ALSO CALLED 'NODE' VALUES
   DO 6001 I=1,NODE-2
      BNDS(I)=BNDS(1)+(I*HH)
   6001 CONTINUE

3 WRITE(6,982)
   982 FORMAT('LOG OF CURRENT CYCLES, LOG XMF')
   WRITE(6,1001)(BNDS(I),I=1,NODE)

4 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
   BNDS(NODE-1)=SAVE2
   BNDS(NODE)=SAVE1

5 WRITE(6,984)
   984 FORMAT('ORDERED LOG OF CURRENT CYCLES, LOG XMF')
   WRITE(6,1001)(BNDS(I),I=1,NODE)

6 WRITE LOG OF CURRENT CYCLES AND PDF OF LOG OF CURRENT CYCLES
   LOG XMF TO PLOT FILES
   WRITE(34,990)
   990 FORMAT('E12.4,1X,E12.4')
   WRITE(34,991)(BNDS(J),DENS(J),J=1,NODE)
   991 FORMAT('E12.4,1X,E12.4')

7 CALCULATE CDF OF LOG OF CURRENT CYCLES
   READ(5,1010)IPT
   WRITE(6,992)
   992 FORMAT('BCDF PARAMETERS')
   WRITE(6,1010)IPT
   X=BNDS(1)
   DO 6003 J=1,NODE
      P=BCDF(X,J,NODE,BNDS,DENS)
      BNDS(J)=X
      X=X+HH
      DISTX(J)=P
   6003 CONTINUE
   WRITE(6,994)
   994 FORMAT('CDF OF LOG OF CURRENT CYCLES, LOG XMF, 1X AXES OF PDF, CDF PLOT')
   WRITE(6,1001)(DISTX(J),J=1,NODE)

8 WRITE(6,993)
   993 FORMAT('ORDERED LOG OF CURRENT CYCLES, LOG XMF, 1X AXES OF PDF, CDF PLOT')
   WRITE(6,1001)(BNDS(I),I=1,NODE)
   WRITE(6,1001)(BNDS(I),I=1,NODE)

9 WRITE LOG OF CURRENT CYCLES AND CDF OF LOG OF CURRENT
   TO THE PLOT FILES
I, the array to be sorted
C at completion Y(I) is smallest value
C at completion Y(N) is largest value
N = N - 1
DO I = 1, N
IF (Y(I), LT, Y(K)) GO TO 2
Y(I) = Y(K)
Y(K) = TEMP
2 CONTINUE
1 CONTINUE
RETURN

IMSL Name: D3SPL/D03SPL (Single/Double precision version)

Computer: IBM/SINGLE

Revised: November 1, 1985
Purpose: Non-parametric pdf function estimation by the penalized likelihood method
Usage: CALL D3SPL (NOBS, X, NODE, BND, INIT, ALPHA, MAXIT, EPS, DENS, STAT, HESS, LDHES, ILOHI, ILOHI2, H, IPUT, W

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)
BND - Vector of length 2 containing the minimum and maximum values for X(1) in BND(1) and BND(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=F, Output otherwise
STAT - Vector of length 4 containing output statistics. (Output) STAT(1) and STAT(2) contain the log-likelihood and the log-likelihood 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)
LDHES - 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
SUBROUTINE D3SPL (NDBS, X, NODE, RNDIS, INIT, ALPHA, MAXIT, EPS, 
DENS, STAT, HESS, LDHESS, ILOM1, DENEST, B, 
IPVT, WK2)

INTEGER NDBS, NODE, INIT, MAXIT, LDHESS, ILOM1, DENEST, B, 
IPVT, WK2
REAL ALPHA, EPS, X(*), RNDIS(2), DENS(*), STAT(4), 
HESS(LDHESS, *), DENEST(NODE*), B(*), WK2(*)

INTEGER I, IMPTR, IPTR, IER, K, KM1, KM2, KP1, KP2, M, MOLD, 
NER, NOB1
REAL BR, BKMI, BSML, CK, CKM1, CKM2, CKMCM1, CKP1, CKP2, 
CONS, EPS1, FACTOR, FK, FKMI, FKM2, FKP1, H, H2, H3, 
SUM, TEM, WK(4)

DOUBLE PRECISION SUM1, SUM2, SUM3

INTEGER MINCR(8)

SAVE MINCR

INTRINSIC ALAD, AMAX1, MAX0, MIN0, MOD, SORT

EXTERNAL EIMES, EIFDP, EIFPSH, EISTI, EISTR, SADU, SAXF)

REAL SCOPY, SHPRD, SSCL, I3SFM, LSTRK, LFSRK

EXTERNAL ISMIN, NINCR, SDOT, SNRM2, SSUM

DATA MINCR/5, 9, 17, 33, 65, 129, 253, 100001/

CALL EIFPSH ('D3SPL ')

Error checks

IF (NDBS .LT. 1) THEN
  CALL EIMES (5, .1, 'After removing all missing (NA, not a ' 
  1
  'number) values from X there are no valid ' 
  2
  'observations. At least one valid observation ' 
  3
  'is necessary.')
END IF

IF (NODE .LE. 4) THEN
  CALL EISTI (4, NODE)

  CALL EIMES (5, .1, 'NODE = %d to. The number of mesh ' 
  1
  'nodes: HIDE must be an odd integer greater ' 
  2
  'than one.'
ELSE IF (MOD(NODE, 2) .EQ. 0) THEN
  CALL EISTI (1, NODE)
  CALL EINES (5, 3, 'NODE = X(I) must be an odd integer')
END IF

IF (ALPHA .LE. 0.0) THEN
  CALL EISTR (1, ALPHA)
  CALL EINES (5, 4, 'ALPHA = %R(1). The penalty weighting factor which controls smoothness. ALPHA must be greater than 0.')
END IF

IF (MAXIT .LE. 0.0) THEN
  CALL EISTI (1, MAXIT)
  CALL EINES (5, 5, 'MAXIT = %Z(II). The maximum number of iterations. MAXIT must be greater than 0.')
END IF

IF (BNDS(1) .GT. BNDS(2)) THEN
  CALL EISTR (1, BNDS(1))
  CALL EISTR (2, BNDS(2))
  CALL EINES (5, 6, 'BNDS(1) = %R(1) and BNDS(2) = %R(2). The minimum and maximum value for X; BNDS(1) must be less than or equal to the maximum value for X; BNDS(2).')
END IF

IF (INIT .NE. 0) THEN
  IF (DENS(1).NE.0 .OR. DENS(NODE).NE.0) THEN
    CALL EISTR (1, DENS(NODE))
    CALL EISTR (2, DENS(1))
    CALL EINES (5, 7, 'DENS(NODE) = %R(1) and DENS(1) = %R(2). The beginning and ending initial estimates of the density must be zero.')
  END IF
END IF

IF (DENS(1).EQ.0 .OR. DENS(NODE).EQ.0) THEN
  CALL EINES (5, 8, 'The initial estimates of the density must be greater than or equal to 0.')
END IF

END IF

IG = 0
DO IG = 1, NOBS
  IF (X(I).LT.BNDS(1) .OR. X(I).GT.BNDS(2)) THEN
    NOBI = NOBI + 1
  END IF
END DO

CONTINUE

IF (NOBI .EQ. NOBS) THEN
  CALL EINES (5, 9, 'All elements of X must lie in the interval BNDS(1) to BNDS(2). At least one element of X must lie in this interval.')
END IF

IF (EPS .LE. 0.0) THEN
  EPS1 = 1.0E-4
ELSE
  EPS1 = EPS
END IF

IF (MICR .NE. 0) GO TO 9000

IMPR = 0
C Initialization.

C Set initial densities

IF (INIT .EQ. 0) THEN
  DENS(1) = 0.0
  DENS(2) = 0.0
  DENS(3) = 0.0
C
ELSE
*M = NODE
END IF
C
20 IF(INIT .EQ. 0). THEN
   M = MIN(NODE,MINCR(IMPTR))
END IF
C
H = (BNDS(2) - BNDS(1)) / (M - 1)
H2 = H*H
H3 = H2*H
C
IF(INIT .NE. 0). THEN
   CALL SSCALE (NODE, 1.0/(H*SSUM(NODE*(DENS(1))), DENS, 1))
END IF
C
B(1) = BNDS(1)
DO 30 I = 2, M
   B(I) = B(I-1) + H
30 CONTINUE
C
IPTR = 0
DO 40 IPTR = IPTR + 1
   IF(X(IPTR) .LT. BNDS(1)) GO TO 40
   IL = I = IPTR
   IF(IPTR .LE. NOBS) THEN
      IF(X(IPTR) .LT. B(K+1)) THEN
         IL = I = IPTR + 1
      END IF
   END IF
40 CONTINUE
C
FACTOR = 2.0*ALPHA/H3
C
IF(INIT .EQ. 0). THEN
   CALL D2SPT (M-2, B(2), I, M0, BNDS, DENS, DENEST, WK, WN, WK)
   TEMP = 1.0/(M0*M0)
   DO 80 I = 2, M - 1
      DENS(I) = AMAX(DENS(I-1,I + 1)
80 CONTINUE
C
ELSE
   DO 90 I = 2, M - 1
      DENS(I) = SQRT(DENS(I))
90 CONTINUE
C
DENS(M) = 0.0
C
D0 140 ITR = 1, MAXIT
   HESS(I+1) = 0.0
   HESS(I) = 0.0
   BSMALL = 0.0
   SUM = 0.0
C
CK** are true estimates = Fh**2
C
DO 120 K = 2, M - 1
KM1 = K - 1
KM2 = MAX0(1, K - 2)
KP1 = K + 1
KP2 = MIN0(K, M + 2)
CM1 = DENS(KM1)
CM2 = DENS(KM2)
CMM = CKM1 + CKM2
IF (K .GE. 4) HESS(1, KM1) = 4.0*FKM1*FACTOR
SUM2 = 0.0D0
SUM1 = 0.0D0
DO 100 I = ILOHI(KM1), ILOHI(KM2)
   TEMP = (X(I) - B(K))/H
   CONS = (1.0 - TEMP)/CK*(CKP1 - CK)*TEMP
   SUM1 = SUM1 + CONS
   SUM2 = SUM2 + CONS*CONS
100 CONTINUE
CKM1 = CK - CKM1
DO 110 J = ILOHI(KM1), ILOHI(KM1 + 1)
   TEMP = CM1 + CONS/TEMPS
   SUM2 = SUM2 + CONS*CONS/TEMPS
110 CONTINUE
TEMP = FACTOR*(CKM2*CKP2 - 4.0*(CKM1*CKP1) + 6.0*CK) + SUM1
TEMP = 2.0*TEMP
BSMALL = BSMALL + 2.0*CONS/TEMPS
HESS(3, KM1) = TEMP + 4.0*CONS*FACTORSUM2
IF (K .NE. 2) HESS(2, KM1) = 4.0*FKM1*(-4.0*FACTORSUM3)
DENEST(KM1, 1) = FKTEMP
DENEST(KM1, 2) = -2.0*FK
120 CONTINUE
BSMALL = 1.0 - SUM + BSMALL
C CALL SCOPY (M, 2, DENEST(1, 1), DENEST(1, 3), 3)
C Finish with the hessian
C CALL SADD (M, 2, BSMALL/(2.0*SUM), HESS(5:1), LDHESS)
C CALL SCOPY (M, 4, HESS(1, 3), LDHESS, HESS(5:1), LDHESS)
C CALL SCOPY (M, 4, HESS(2, 2), LDHESS, HESS(4:1), LDHESS)
C Solve symmetric band linear system
C CALL L2TRB (M, 2, HESS, LDHESS, 2, 2, HESS, LDHESS, IPVT, WN2)
C CALL LFSB (M, 2, HESS, LDHESS, 2, 2, IPVT, DENEST, 1, DENEST)
C CALL LFSK (M, 2, HESS, LDHESS, 2, 2, IPVT, DENEST(1:2), 1)
C IF (MRI(1:1) .NE. 0) GO TO 9000
C Compute the constant
CONS = SPOT(M, 2, DENEST(1, 1), 1, DENEST(1:2), 1)
CONS = (1.0/H - SUM - SPOT(M, 2, DENEST(1, 1), 1, DENEST(1:2), 1))/CONS
C Update the gradient
CALL SAXPY (M-2, CONS, DENEST(1), 1, DENEST(1), 1)
CALL SAXPY (M-2, -1.0, DENEST(1), 1, DENEST(2), 1)

Parameter updates
Check the convergence criterion

IF (SNRM2(M-2, DENEST(1), 1) .LT. EPSI*TEMP) GO TO 150

Ad hoc projection to plus quadrant

TEMP = TEMP*1.0E-4/SORT(M-2.0)
DO 130 I=2, M-1
DENEST(I) = AMAX1(TEMP, DENEST(I))
130 CONTINUE

CALL EIESTI (1, MAXIT)
CALL EIEMES (3, 1, 'The maximum number of iterations was exceeded.')

CALL SHPRD (M-2, DENEST(2), 1, DENEST(2), 1, DENEST(2), 1)

IF (M .LT. NODE) GO TO 20

evaluate loss likelihood and penalty

C
SUM1 = 0.0
Penalty
DO 160 K=1, M

KP1 = MAX0(K, M)
K1 = M01(K+1, M)
SUM = SUM + (DENEST(K1)-2.0*DENEST(K)+DENEST(KP1))**2
160 CONTINUE

STAT(2) = -0.5*FACTOR*SUM1
STAT(2) = -0.5*FACTOR*SUM1

C
SUM2 = 0.0
Los-likelihood
DO 170 T=1, NODE
IF (X(T).LE.BNDS(1) AND X(T).GE.BNDS(2)) THEN
CALL D2SPF (1, X(T), 1, NODE, BNDS, DENEST, WK, WK)

SUM = SUM + ALOG(DENEST(1, 1))
170 CONTINUE

STAT(1) = SUM2
Evaluate M.L.F.E. mean and variance

C
SUM1 = 0.0
SUM2 = 0.0
DO 180 K=1, M-1
FK = DENEST(K)
FKP1 = DENEST(K+1)
BK = BK
CONS = FK + FKP1
TEMP = CONS + BK
SUM1 = SUM1 + H2*TEMP/(6.0 + 0.5*CONS**2)
SUM2 = SUM2 + H2*(CONS**2 - FK)**2/12.0 + H2*CONS**2
180 CONTINUE

STAT(3) = SUM1
STAT(4) = SUM2 - SUM1*SUM1

C
9000 CALL EIPOP ('D3SPF ')
RETURN
END
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Current Cycles</th>
<th>X Factor</th>
<th>Y Factor</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
<tr>
<td>0.3078E+01</td>
<td>0.1389E+01</td>
<td>0.1000E+03</td>
<td>0.3893E+02</td>
<td>0.3943E+02</td>
<td>0.2246E+00</td>
</tr>
</tbody>
</table>
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;