PROBABILISTIC ANALYSIS
FOR FATIGUE STRENGTH DEGRADATION OF MATERIALS

Prepared by:
Lola Boyce, Ph. D., P. E.

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, Ohio 44135

The Division of Engineering
The University of Texas at San Antonio
San Antonio, TX 78285
January, 1989
PROBABILISTIC ANALYSIS
FOR FATIGUE STRENGTH DEGRADATION OF MATERIALS

Prepared by:
Lola Boyce, Ph. D., P. E.

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, Ohio 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>PREFACE.</td>
<td>i</td>
</tr>
<tr>
<td>ABSTRACT.</td>
<td>ii</td>
</tr>
<tr>
<td>LIST OF FIGURES.</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES.</td>
<td>iv</td>
</tr>
<tr>
<td>1.0 INTRODUCTION.</td>
<td>1</td>
</tr>
<tr>
<td>2.0 FATIGUE CRACK GROWTH MODEL</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Background.</td>
<td>2</td>
</tr>
<tr>
<td>2.2 RANDOM2 Computer Program</td>
<td>3</td>
</tr>
<tr>
<td>3.0 FATIGUE STRENGTH REDUCTION MODEL.</td>
<td>4</td>
</tr>
<tr>
<td>3.1 Background.</td>
<td>4</td>
</tr>
<tr>
<td>3.2 RANDOM3 and RANDOM4 Computer Programs.</td>
<td>5</td>
</tr>
<tr>
<td>3.3 Sensitivity Study.</td>
<td>5</td>
</tr>
<tr>
<td>4.0 REFERENCES.</td>
<td>9</td>
</tr>
<tr>
<td>5.0 APPENDIX 1: FATIGUE CRACK GROWTH MODEL: RANDOM2 USER MANUAL.</td>
<td>10</td>
</tr>
<tr>
<td>6.0 APPENDIX 2: FATIGUE STRENGTH DEGRADATION MODEL: RANDOM3 AND RANDOM4 USER MANUAL.</td>
<td>47</td>
</tr>
</tbody>
</table>
The University of Texas at San Antonio (UTSA) is a relatively new university. It was established in 1969 and opened for classes in 1973. As the only comprehensive public university serving the nation's ninth largest city, it was and is vital to San Antonio and the entire South Texas Region. In 1983, just seven years ago, an undergraduate engineering program was established at UTSA with the support of the community and its leaders. Today, all three undergraduate engineering programs are ABET accredited and serve about 800 students, a significant percentage of whom are Hispanic. The future includes a new engineering building, providing new laboratory facilities and equipment, together with offices and laboratories, planned to open in January, 1991. Furthermore, a graduate program is planned at both M.S. and Ph.D. levels, and it is hoped that the first Master's Degree students will be able to enroll in Fall, 1989.

Naturally, the engineering research environment is just developing at UTSA. Now, thanks in great measure to the UT System support and this ongoing NASA grant, good progress is being made. Specifically, the purchase of a UT System CRAY-XM/P in March, 1986 and a second one in December, 1988 has provided a world-class analytical and numerical research environment not ordinarily available to a new university. As a result the UTSA Supercomputer Network Research Facility (SNRF) was developed by the principal investigator, Dr. Lola Boyce. This has allowed the successful completion of this research project, the first of its kind at UTSA.

This NASA research grant has allowed two Mechanical Engineering students, Thomas Lovelace and Callie Scheidt, to work directly with the principal investigator, Dr. Boyce, providing them with a quality research experience they would otherwise probably not have had. Both students have expressed an interest in continuing their educations at the graduate level.

In conclusion, and in view of the significant accomplishments in fundamental research, enhancement of the engineering research environment at UTSA, and direct support of Mechanical Engineering students, it is hoped that the proposed extension of this grant will receive favorable consideration at NASA. The principal investigator sincerely thanks NASA for funding this first year grant.
ABSTRACT

This report presents the results of the first year of effort of a program of research conducted for NASA-LeRC by The University of Texas at San Antonio (UTSA). The research included development of methodology that provides a probabilistic treatment of lifetime prediction of structural components of aerospace propulsion systems subjected to fatigue. Material strength degradation models, based on primitive variables, include both a fatigue strength reduction model and a fatigue crack growth model. Linear elastic fracture mechanics is utilized in the latter model. Probabilistic analysis is based on simulation, and both maximum entropy and maximum penalized likelihood methods are used for the generation of probability density functions. The resulting constitutive relationships are included in several computer programs, RANDOM2, RANDOM3 AND RANDOM4. These programs determine the random lifetime, of an engine component, in mechanical load cycles, to reach a critical fatigue strength or crack size. The material considered was a cast nickel base-superalloy, one typical of those used in the Space Shuttle Main Engine (SSME).
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>p.d.f. of base line room temperature (RT) problem.</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>c.d.f. of base line room temperature (RT) problem.</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>c.d.f. of base line room temperature (RT) problem compared with c.d.f. of base line high temperature (HT) problem.</td>
</tr>
<tr>
<td>A1.2</td>
<td>20</td>
<td>c.d.f. of log of mechanical cycles for fatigue crack growth model, using maximum penalized likelihood.</td>
</tr>
<tr>
<td>TABLE</td>
<td>Description</td>
<td>PAGE</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Base line room temperature (RT) problem input using the fatigue strength</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>reduction model with maximum entropy p.d.f. generation (RANDOM4).</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sensitivity study cases for fatigue strength reduction model with maximum</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>entropy p.d.f. generation (RANDOM4).</td>
<td></td>
</tr>
<tr>
<td>A1.1</td>
<td>RANDOM2 Sample problem input (SI units).</td>
<td>18</td>
</tr>
<tr>
<td>A1.2</td>
<td>Physical quantities, symbols, and units for fatigue crack growth model for</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>RANDOM3 and RANDOM4.</td>
<td></td>
</tr>
<tr>
<td>A2.1</td>
<td>RANDOM3 and RANDOM4 input (SI units).</td>
<td>56</td>
</tr>
<tr>
<td>A2.2</td>
<td>Physical quantities, symbols, and units for fatigue crack growth model for</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>RANDOM 3 and RANDOM4.</td>
<td></td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

This report presents the results of the first year effort of a research program entitled "Development of Advanced Methodologies for Probabilistic Constitutive Relationships of Material Strength Models." This research is sponsored by the National Aeronautics and Space Administration-Lewis Research Center (NASA-LeRC). The principal investigator is Dr. Lola Boyce, Associate Professor of Mechanical Engineering, The University of Texas at San Antonio (UTSA). The objective of the research program is the development of methodology that provides a probabilistic treatment of lifetime prediction of structural components of aerospace propulsion systems subjected to fatigue.

Two material strength degradation models, based on primitive variables were developed as part of this first year effort: a fatigue crack growth model and a fatigue strength reduction model. The former model utilizes principles of linear elastic fracture mechanics while the latter is, recently developed at NASA-LeRC, quantifies the reduction of strength under cyclic loading, including elevated temperature treatment. Probabilistic analysis is based on simulation, and both maximum entropy and maximum penalized likelihood methods are used for the generation of probability density functions that predict the random lifetime of a material typical of those used in the Space Shuttle Main Engine (SSME), namely a cast nickel base-superalloy.

The resulting constitutive relationships are included in several computer programs, RANDOM2, RANDOM3, and RANDOM4. The programs were developed using both the NASA-LeRC and UTSA Supercomputer Network Research Facility (SNRF) Cray X-MP. New versions of the program accompany this report (see enclosed floppy disk), utilizing the new IMSL Ver. 10 subroutines. Thus, these new versions of the programs will execute on the current NASA-LeRC supercomputer facilities. Also the floppy disk contains sample problems to verify program performance at NASA-LeRC.

Finally, a sensitivity study was carried out for the fatigue strength reduction model for the case of a relatively high mean stress and a relatively low constant amplitude alternating stress at failure. In addition to varying the stresses, the effect of temperature was also considered. A paper was produced documenting much of the effort of this first year research program. This paper is entitled "Probabilistic Constitutive Relationships for Cyclic Material Strength Models", by L. Boyce and C.C. Chamis. It was presented at the 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, VA, April, 1988 and is published in the Proceedings. It has also been submitted to the AIAA Journal of Propulsion and Power.
2.0 FATIGUE CRACK GROWTH MODEL

2.1 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\(^1\)

\[
da/dN = C(\Delta K)^m
\]

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

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

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

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

or,

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

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

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

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.\(^2\) 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 quantative 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,
\[ N = f(X_i), \quad i = 1, \ldots, 6, \] (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.\(^1\)

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.\(^3\)

2.2 RANDOM2 Computer Program

A FORTRAN computer program for the fatigue crack growth model, called RANDOM2, was written using the above-described probabilistic methodology and the constitutive relationship expressed in equation (3). Although the four independent random variables could have any distribution, this initial program provided for normal or lognormal only.

A complete Users Manual for RANDOM2 is contained in Appendix 1. Also, a disk containing a new version of RANDOM2 and a sample problem accompanies this report. The new version of RANDOM2, documented in the Users Manual, uses the new ISML, Ver. 10 subroutines and provides for parameter input from an input file.
3.0 FATIGUE STRENGTH REDUCTION MODEL

3.1 Background

Fatigue strength data are usually presented as cycles to failure for each of several stress amplitudes, the familiar S-N diagram. Results indicate that for lower stress amplitudes the cycles (or time) to failure increases. Thus, a power curve fit through the data yields a monotonically decreasing curve. In general, this curve is represented as

\[ S = \left[\frac{N}{C'}\right]^{-1/m'} \]  

where the primitive variables in this equation are as follows: S is the applied constant amplitude alternating stress at failure or fatigue strength, N is number of cycles, C' is a material parameter that varies from specimen to specimen and m' is a material constant.\(^4\) Equation (6) can be written in terms of "cycles to reach a given fatigue strength" as

\[ N = C'S^{-m'} \]  

Recently another fatigue strength reduction model has been proposed that takes into account the effect of temperature as well as other parameters that affect strength.\(^5\) The general form of the constitutive relationships for this model is applied to the constituents of high temperature composite materials. Specifically, it is applied herein for the case of a single material constituent. The mechanical property of interest is fatigue strength which is expressed in terms of primitive variables, including the general categories of temperature, mechanical cycles and mean stress. For these categories, the relationship becomes

\[ \frac{S}{S_o} = \left[ \frac{T_F - T}{T_F - T_o} \right]^n \left[ \frac{S_F - \sigma}{S_F - \sigma_o} \right]^m \left[ \frac{\log N_{MF} - \log N_M}{\log N_{MF} - \log N_{MO}} \right]^q \]  

where S is the applied constant amplitude alternating stress at failure (fatigue strength) at current (or operating) temperature, T, mean stress, \(\sigma\), and mechanical cycle, \(N_M\). \(S_o\) is fatigue strength at reference temperature, \(T_o\) (usually room temperature), reference mean stress (or residual stress), \(\sigma_o\), and reference mechanical cycle, \(N_{MO}\). Also, \(T_F\) is the final or melting temperature of the material, \(S_F\) is the final or tensile strength of the material, and \(N_{MF}\) is the final mechanical cycle or lifetime. Empirical parameters, n, m, and q, are determined from available experimental data or estimated from anticipated behavior of the particular product term.\(^6\) Note that the term containing mechanical cycles is expressed in terms of the log of cycles rather than cycles. This formulation is attractive when \(N_M\) and \(N_{MO}\) are small compared to \(N_{MF}\). The equation may be solved for \(N_M\), or the "cycles to reach a given fatigue strength." The expression is

\[ N = 10 \exp \left[ \log N_{MF} - \left( \log N_{MF} - \log N_{MO} \right) \left\{ \frac{S}{S_o} \left[ \frac{T_F - T}{T_F - T_o} \right]^n \left[ \frac{S_F - \sigma}{S_F - \sigma_o} \right]^m \right\}^{1/q} \right] \]  

(9)
For values typical of a cast nickel base-superalloy subjected to typical loads and temperatures, equation (9) indicates increasing life for decreasing temperature, decreasing tensile mean stress, and decreasing applied alternating stress. It indicates decreasing life for increasing temperature, decreasing compressive mean stress, and increasing applied alternating stress. Therefore, equation (9) predicts observed trends in general.

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 for RANDOM3. For RANDOM4, a p.d.f. of cycles is generated using the maximum entropy method. Maximum entropy uses Jaynes' principle which says that "the minimally prejudiced distribution is that which maximizes the entropy subjected to the constraints supplied by the given information."7

3.2 RANDOM3 and RANDOM4 Computer Programs

FORTRAN computer programs for the fatigue strength reduction model called RANDOM3 and RANDOM4 were written using the above-described probabilistic methodology and the constitutive relationship expressed in equation (9). Although the thirteen independent random variables could have any distribution, these programs provided for normal or lognormal only.

A complete Users Manual for RANDOM3 and RANDOM4 is contained in Appendix 2. Also, a disk containing new versions of RANDOM3 and RANDOM4 uses the new IMSL, Ver. 10 subroutines and provides for parameter input from an input file.

3.3 Sensitivity Study

The fatigue strength degradation model using the maximum entropy method of p.d.f. generation (RANDOM4) was selected for use in a sensitivity study. A base line problem utilizing a high mean stress (σ = 90 ksi) and a low constant amplitude alternating stress at failure (S = 22.5 ksi) was established. A room temperature (T = 68°F) problem was executed. The input for this problem is given in Table 1 and the output, in the form of a p.d.f. and a c.d.f. is given in Figures 1 and 2. A high temperature (T= 1562°F) base line problem was also selected. Then, for a fixed base line alternating stress at failure, the mean stress was varied above and below the base line value. Both room and high temperatures were selected. Finally, for a fixed base line mean stress, the alternating stress at failure was varied above and below the base line value. Again, room and high temperatures were selected. A summary of the cases studied is given in Table 2.

Conclusions drawn from this sensitivity study are summarized below. Increasing temperature for the same stress conditions reduces lifetime for all cases (see, for example, Figure 3). At room temperature, when mean stress is increased by 10%, lifetime decreases only very slightly. At high temperature, however, when mean stress is increased by 10%, lifetime decreases substantially. Also, at room temperature, when alternating stress is increased by 30%, lifetime decreases only slightly. At high temperatures, however, when alternating stress is increased by 30%, lifetime decreases very substantially. Considering the above points, lifetime is more sensitive to increasing alternating stress, rather than mean stress. This is probably because alternating stress was increased by 30%, whereas mean stress was increased by only 10%.
Table 1: Base line room temperature (RT) problem input, using the fatigue strength reduction model with maximum entropy p.d.f. generation (RANDOM4)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Distribution Type</th>
<th>Mean</th>
<th>Std. Dev. Value</th>
<th>% of Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_F$ (Melting Temp.)</td>
<td>Normal</td>
<td>2732.0 °F</td>
<td>82.0</td>
<td>3</td>
</tr>
<tr>
<td>$S_F$ (Ult. Tensile Str.)</td>
<td>Lognormal</td>
<td>130.0 ksi</td>
<td>6.5</td>
<td>5</td>
</tr>
<tr>
<td>$N_{MF}$ (Log of Final Cycle)</td>
<td>Lognormal</td>
<td>8.0</td>
<td>0.8</td>
<td>10</td>
</tr>
<tr>
<td>$T_0$ (Ref. Temp.)</td>
<td>Normal</td>
<td>68.0 °F</td>
<td>2.0</td>
<td>3</td>
</tr>
<tr>
<td>$\sigma_0$ (Residual Comp. Stress)</td>
<td>Lognormal</td>
<td>-2.9 ksi</td>
<td>-0.145</td>
<td>5</td>
</tr>
<tr>
<td>$N_M$ (Log of Ref. Cycle)</td>
<td>Lognormal</td>
<td>7.0</td>
<td>0.7</td>
<td>10</td>
</tr>
<tr>
<td>$S_O$ (Ref. Fatigue Str.)</td>
<td>Lognormal</td>
<td>72.6 ksi</td>
<td>3.6</td>
<td>5</td>
</tr>
<tr>
<td>$T$ (Current Temp.)</td>
<td>Normal</td>
<td>68.0 °F</td>
<td>2.0</td>
<td>3</td>
</tr>
<tr>
<td>$\sigma$ (Current Mean Stress)</td>
<td>Lognormal</td>
<td>90.0 ksi</td>
<td>4.5</td>
<td>5</td>
</tr>
<tr>
<td>$S$ (Current Fatigue Str.)</td>
<td>Lognormal</td>
<td>22.5 ksi</td>
<td>1.125</td>
<td>5</td>
</tr>
<tr>
<td>$n$ (Temp. Exponent)</td>
<td>Normal</td>
<td>0.5</td>
<td>0.015</td>
<td>0.3</td>
</tr>
<tr>
<td>$m$ (Stress Exponent)</td>
<td>Normal</td>
<td>0.5</td>
<td>0.015</td>
<td>0.3</td>
</tr>
<tr>
<td>$q$ (Cycle Exponent)</td>
<td>Normal</td>
<td>0.5</td>
<td>0.015</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Fig. 1: p.d.f. of base line room temperature (RT) problem
Fig. 2  c.d.f. of base line room temperature (RT) problem

Table 2  Sensitivity study cases for fatigue strength reduction model with maximum entropy p.d.f. generation (RANDOM4)

<table>
<thead>
<tr>
<th>σ (ksi)</th>
<th>S (ksi)</th>
<th>T (°F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Line (RT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>22.5</td>
<td>68</td>
</tr>
<tr>
<td>90</td>
<td>22.5</td>
<td>68</td>
</tr>
<tr>
<td>100</td>
<td>22.5</td>
<td>68</td>
</tr>
<tr>
<td>Base Line (HT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>22.5</td>
<td>1562</td>
</tr>
<tr>
<td>90</td>
<td>22.5</td>
<td>1562</td>
</tr>
<tr>
<td>100</td>
<td>22.5</td>
<td>1562</td>
</tr>
<tr>
<td>Base Line (RT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>15.0</td>
<td>68</td>
</tr>
<tr>
<td>90</td>
<td>22.5</td>
<td>68</td>
</tr>
<tr>
<td>90</td>
<td>30.0</td>
<td>68</td>
</tr>
<tr>
<td>Base Line (HT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>15.0</td>
<td>1562</td>
</tr>
<tr>
<td>90</td>
<td>22.5</td>
<td>1562</td>
</tr>
<tr>
<td>90</td>
<td>30.0</td>
<td>1562</td>
</tr>
</tbody>
</table>
Fig. 3 c.d.f. of base line room temperature (RT) problem compared with c.d.f. of base line high temperature (HT) problem.
4.0 REFERENCES


5.0 APPENDIX 1

FATIGUE CRACK GROWTH MODEL:
RANDOM2 USER MANUAL
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.\(^3\)

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

(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

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

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,

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

(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

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

(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.\(^4\) 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:

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

(4)
or, in general,

\[ N = f(X_i), \ i = 1, \ldots, 6, \quad (5) \]

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:

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:

<table>
<thead>
<tr>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>0</td>
<td>50.0E-01</td>
<td>10.0E-05</td>
</tr>
</tbody>
</table>

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

EXAMPLE:

<table>
<thead>
<tr>
<th>1234567890</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
</tbody>
</table>
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, \( A_f \) and \( Y \), 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>Standard Deviation (% 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.

```
12345678901234567890123456789012345678901
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 Chamis. 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>
<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</td>
<td>m/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 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
C PARTS-EOUFRAN FATIGUE CRACK GROWTH EQUATION
C RANDOMIZED AND APPLIED TO CRACK LENGTH
INTEGER ISEED,NTOT,M,INITMISS,MAXIT,NODE
REAL XM,YS,YM,YS,PS,PRW,RPWSP(9999),ALPHA
COMMON/WORKSP/RWSP
DIMENSION RIN(1000),RAI(1000),RCC(1000)
DIMENSION DELSIC(1000),BMD(1000)
DIMENSION XM2(1000),C(1000)
DIMENSION STAT(9999),BENS(1000),DISTX(1000)
DIMENSION BNE0(1000),BB(1000),FF(1000)
EXTERNAL RNLNL,RNSET,RNAND,DEPL,IVKIN

1001 FORMAT(112,112)
1002 FORMAT(114,14)
1003 FORMAT(114)
1004 FORMAT(114,14)
1005 FORMAT(114)
1006 FORMAT(114,14)
1007 FORMAT(114)
1008 FORMAT(114,14)
1009 FORMAT(114)
1010 FORMAT(114,14)
1011 FORMAT(114)

C NORMAL MATERIAL PROPERTY, M
READ (5,1002) ISEED,NTOT
WRITE(6,1002) ISEED,NTOT
READ (5,1011) YM,YS
YS=9.14
YM=2.8
CALL RNSET(ISEED)
CALL RNAND(NTOT)
DD 100 I=1,NTOT
102 CONTINUE
WRITE(6,1019)

C LOGNORMAL INITIAL CRACK SIZE (PORE DIAMETER), AI
WRITE(6,1001) RIN(I),I=1,NTOT
READ (5,1011) XM, XS
WRITE(6,1011) XM, XS
XM = 300.0E-06
YM = 45.0E-06
YS = 500.0E-06
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNMLN(0.1I,YM,YS,RAI)
WRITE(6,1020)

C LOGNORMAL MATERIAL PROPERTY, C
WRITE(6,1001) RAI(I),I=1,NTOT
READ (5,1011) XM, XS
WRITE(6,1011) XM, XS
XM = 2.0E-06
XS = 0.220E-11
YM = LOG(XM) - 0.5*YS**2
WRITE(6,1010) NMINT

CALCULATE WINDOW WIDTH, HH

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

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

DO 6001 I = 1, NODE - 2
    BNDS(I+1) = BNDS(I) + (I*HH)
  6001 CONTINUE

WRITE(6,983) 983: FORMAT(' LOG OF CURRENT CYCLES, LOG_XNF')
WRITE(6,1001) (BNDS(I), I = 1, NODE)

REORDER BNDS FOR PLOTTING

SAVE1 = BNDS(1)
SAVE2 = BNDS(NODE)
BNDS(NODE) = BNDS(2)
BNDS(1) = BNDS(I+2)

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

WRITE(6,984) 984: FORMAT(' ORDERED LOG OF CURRENT CYCLES, LOG_XNF',
     1X AXIS PDF, CDF PLOT')
WRITE(6,1003) (BNDS(I), I = 1, NODE)
WRITE(6,1001) (BNDS(J), DENS(I), J = 1, NODE)

LD XNF TO PLOT FILES

WRITE(34,990) 990: FORMAT('E12.4,1X,E12.4')
WRITE(34,991) (BNDS(J), DENS(I), J = 1, NODE)
WRITE(34,991) (BNDS(J), DENS(I+1), J = 1, NODE)

CALCULATE CDF OF LOG OF CURRENT CYCLES

READ(5,1010) IOPT
WRITE(6,992) 992: FORMAT('CDF PARAMETERS')
WRITE(6,1010) IOPT
XO = BNDS(1)
DO 6003 I = 1, NODE
    P = GCDF(XO, IOPT, NODE, BNDS, DENS),
    BNDSX(I) = XO
    XO = XO + HH
    DIStX(I) = P
  6003 CONTINUE

WRITE(6,994) 994: FORMAT('CDF OF LOG OF CURRENT CYCLES, LOG XNF',
     1Y AXIS PDF, CDF PLOT')
WRITE(6,1001) (DIStX(I), I = 1, NODE)
WRITE(6,1001) (DIStX(I), I = 1, NODE)

CALCULATE CDF OF LOG OF CURRENT CYCLES

WRITE(6,993) 993: FORMAT(' ORDERED LOG OF CURRENT CYCLES, LOG XNF',
     1X AXIS PDF, CDF PLOT')
WRITE(6,1001) (BNDS(I), I = 1, NODE)
WRITE(6,1001) (BNDSX(I), I = 1, NODE)

WRITE LOG OF CURRENT CYCLES AND CDF OF LOG OF CURRENT CYCLES TO THE PLOT FILES
SUBROUTINE SORT (Y,N)
DIMENSION Y(10000)
C Y IS THE ARRAY TO BE SORTED
C AT COMPLETION Y(1) IS SMALLEST VALUE
C AT COMPLETION Y(N) IS LARGEST VALUE
N1 = N - 1
DO 1 I = 1, N1
J = I + 1
DO 2 X = J, N
IF (Y(J).LT.Y(X)) GO TO 2
2 CONTINUE
1 CONTINUE
RETURN
END

IMSL Name: D3SPL/DD3SPL (Single/Double precision version)

Computer: IBM/Single

Revised: November 1, 1985

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

Usage: CALL D3SPL (NODS, X, NODE, BNDS, INIT, ALPHA, MAXIT, EPS,
DENS, STAT, HESS, LDHESS, ILOHI, DENEST, B)

Arguments:
NODS - Number of observations. (Input)
X - Vector of length NODS 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(1) 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
DENEST - NODE by Matrix containing the gradient vector, among other quantities. (Output)
B - Vector of length NODE containing the NODE values. (Output)
IPUT - Pivot vector of length NODE-2. (Output)
WK2 - Work vector of length NODE-2. (Output)

Chapter: STAT/LIBRARY Density and Hazard Estimation

Copyright: 1985 by IMSL, Inc. All Rights Reserved.

Warranty: IMSL warrants only that IMSL testing has been applied to this code. No other warranty, expressed or implied, is applicable.

SUBROUTINE D3SPL (NOBS, X, NODE, BNDS, INIT, ALPHA, MAXIT, EPS, 
DENS, STAT, HESS, LDHESS, LDHCI, DENEST, B, 
IPUT, WK2)

INTEGER NOBS, NODE, INIT, MAXIT, LDHESS, LDHCI(NODE),
IPUT(*)
REAL ALPHA, EPS, X(*), BNDS(2), DENES(2), STAT(4),
HESS(LDHESS), DENEST(NODE), B(*), WK2(*)

INTEGER I, IMTR, IPTR, IER, K, KM1, KM2, KPI, KP2, M, MOLD, 
NER, NOD1, NSM, NSMALL, CK, CKM1, CKM2, CKCM1, CKPI, CKP2, 
CONS, EPS1, FACTOR, FK, FM1, FM2, FKPI, H, H2, HS,
SUM, TEMP, WK(4), 
DOUBLE PRECISION SUM1, SUM2, SUM3

INTEGER MINCR(8)
SAVE MINCR

INTRINSIC ALOG, AMAX1, AMAX0, MIN0, MOD

REAL ALOG, AMAX1, SORT

EXTERNAL EIMES, EINFO, EIPSX, EISTR, EIR, RABD, SAXPY;
SCOPY, SHPRD, SSCAL, DCSPF, LITRF, LFSRF

EXTERNAL ISMIN, MINCD, SDOT, SNRM2, SSUM

REAL SDOT, SNRM2, SSUM

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

CALL EIPSH ('D3SPL ')

Error checks

IF (NOBS .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 //

   'necessary.'
   END IF

IF (NODE .LE. 4) THEN
   CALL EISTI (1, NODE)
   CALL EIMES (5, 2, 'NODE = %11. The number of mesh //

   'nodes, NODE, must be an odd integer greater //
IF (MOD(NODE/2) .EQ. 0) THEN
  CALL EISTI (1, NODE)
  CALL EIMES (5, 3, 'NODE = %I(1) must be an odd integer."
                 'greater than 4.')
END IF

IF (ALPHA .LE. 0.0) THEN
  CALL EISTR (5, ALPHA).
  CALL EIMES (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 EIMES (5, 5, 'MAXIT = %I(1). The maximum number"
                 'of iterations. MAXIT must be greater than 0.')
END IF

IF (BNDSD1 .GT. BNDSD2) THEN
  CALL EISTR (1, BNDSD1).
  CALL EIMES (5, 6, 'BNDSD(1) = %R(1) and BNDSD(2) ="
                 '%R(2). The minimum value for X, BNDSD(1), must"
                 'be less than or equal to the maximum value for"
                 '%X, BNDSD(2).')
END IF

IF (DENS(1).NE.0 .OR. DENS(NODE).NE.0) THEN
  CALL EISTR (1, DENS(1)).
  CALL EISM (1, NODE).
  CALL EIMES (5, 7, 'DENS(1) = %R(1) and DENS(NODE)=%I(1)"
                 'estimates of the density must be zero.')
END IF

IF (DENS(ISMIN(NODE-DENS(1))).LT. 0) THEN
  CALL EIMES (5, 8, 'The initial estimates of the"
                 'density, DENS, must be greater than or "
                 'equal to 0."
END IF

END IF

NBI = 0
DO 10 I = 1, NBI
  IF (X(I).LT.BNDSD1 .OR. X(I).GT.BNDSD2) THEN
    NBI = NBI + 1
END IF
10 CONTINUE

IF (NBI .EQ. NBI) THEN
  CALL EIMES (5, 9, 'All elements in X lie outside the"
                 'interval BNDSD1 to BNDSD2. 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 (WIRCD(0).NE. 0) GO TO 9000

IMPTTR = 0

IF (INIT .EQ. 0) THEN
  DENS(1) = 2.0/(BNDSD2-BNDSD1)
  DENS(3) = 0.0
M = 3
ELSE
M = NODE
END IF
C 20 IF (INIT ..EQ. 0) THEN
    MOLD = M
    IMPTR = IMPTR + 1
    M = MIN0(NODE, MINC(IMPTR))
END IF
C
H = (BNDS(2) - BNDS(1)) / (M - 1)
H2 = H * H
H3 = H2 * H
C
IF (INIT ..NE. 0) THEN
    CALL SSCAL (NODE, 1.0 / (HSSUM(NODE, DENS, 1)), DENS, 1)
END IF
C
B(1) = BNDS(1)
DO 50 I = 2, M
    B(I) = B(I-1) + H
    IF (I-PTR .LE. NOBS) GO TO 50
C 50 CONTINUE
C
IPTR = 0
IF (X(IPTR) .LT. BNDS(1)) GO TO 40
DO 30 K = 1, M - 1
    ILOMH(K+1) = IPTR
    ILOMH(K+2) = IPTR + 1
    IF (X(IPTR) .LT. B(K+1)) THEN
        ILOMH(K+2) = ILOMH(K+2) + 1
        IF (IPTR .LE. NOBS) GO TO 50
    END IF
C 30 CONTINUE
C
IF (INIT ..EQ. 0) THEN
    CALL D2SPT (H-2, B(2), 1, MOLD, BNDS, DENS, DEBEST, WK, WK)
    TEMP = 1.0 / (HSSUM(H, H, H, M))
    DO 80 I = 2, M - 1
        DENS(I) = AMAX1(TEMP, SQRT(DENS(I-1, I)))
C 80 CONTINUE
C
ELSE
    DO 90 I = 2, M - 1
        DENS(I) = SQRT(DENS(I))
C 90 CONTINUE
C
END IF
C
DENS(M) = 0.0
C
DO 140 ITER = 1, MAXIT
C
C
DO 140 M = 0.0
C
HESS(M, M) = 0.0
C
HESS(M, M) = 0.0
C
DO 140 M = 0.0
C
SUM = 0.0
C
CK** are true estimates = FK**2
C
DO 120 K=2, M - 1
KM1 = K - 1
KM2 = MAXO(1+K-2)
KP1 = K + 1
KP2 = MNO(M+K+2)
FKM1 = DENS(KM1)
FKM2 = DENS(KM2)
CKM2 = FKM2**2
CKM1 = FKM1**2
CKP1 = DENS(KP1)**2
CKP2 = DENS(KP2)**2
BK = B(K)
BKM1 = B(KM1)
SUM = SUM + CK
IF (K.EQ.4) HESS(1,KM1) = 4.0*FK*FKM2*FACTOR
SUM1 = 0.000
SUM2 = 0.000
SUM3 = 0.000
DO 100 I = ILOHI(K,1), ILOHI(K,2)
TEMP = (X(I) - BK)/H
CONS = (1.0 - TEMP)/(CK + (CKP1 + CKM1)*CONS)
SUM1 = SUM1 - CONS
SUM2 = SUM2 + CONS
CONTINUE
100
CKCM2 = CK - CKM1
DO 110 I = ILOHI(KM1,1), ILOHI(KM2,2)
CONS = (X(I) - BKM1)/CONS
SUM3 = SUM3 + CONS
TEMP = CONS
SUM3 = SUM3 + CONS
CONTINUE
110
TEMP = FACTOR*(CKM2 + CKP2 - 4.0)*(CKM1 + CKP1 + 6.0*CONS) + SUM1
BSMALL = BSMALL - 2.0*CONS
HESS(3,KM1) = TEMP + 4.0*CONS + (CONS*CONS - CONS)*CONS
CONTINUE
120
BSMALL = 1.0/H - SUM + BSMALL
CALL SCOPY (M-2, DEVEST(1,3), 1, DEVEST(1,3), 1)
CALL SCOPY (M-2, -BSMALL/(2.0*SUM), HESS(3,1), LDRESS)
CALL SCOPY (M-4, HESS(3,1), LDRESS, HESS(3,1), LDRESS)
HESS(3+M-3) = 0.0
HESS(M+K-2) = 0.0
CALL SCOPY (M-3, HESS(2,2), LDRESS, HESS(4,1), LDRESS)
HESS(4+M-2) = 0.0
CALL LFB (M-2, HESS, LDRESS, 1, 2, IPVT, WK2)
CALL LFS (M-2, HESS, LDRESS, 1, 2, IPVT, DEVEST(1,1), 1)
CALL LFS (M-2, HESS, LDRESS, 1, 2, IPVT, DEVEST(1,1), 1)
IF (MIRC(1), NE. 0) GO TO 9900
CONS = SBOT(M-2, DEVEST(1,3), 1, DEVEST(1,3), 1)
CONS = (1.0/H - SUM - SBOT(M-2, DEVEST(1,3), 1, DEVEST(1,3), 1))/CONS
Update the gradient
CALL SAXPY (M-2, CONS, DENEST(1:2), 1., DENEST(1:1), 1)

Parameter updates

CALL SAXPY (M-2, -1.0, DENEST(1:1), 1., DENEST(2), 1)

Check the convergence criterion

TEMP = SNRM2(M-2, DENEST(2:1))

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

TEMP = TEMP*1.0E-4/SORT(M-2.0)

DO 130 I = 2, M + 1

DEN(S) = AMAX1(TEMP, DEN(S))

130 CONTINUE

140 CONTINUE

CALL EISTI (1, MAXIT)

CALL EIMES (_X** (MAXIT=X(11))) was exceeded,

& The maximum number of iterations //

150 CALL SHFROD (M-2, DENEST(2), 1., DENEST(2), 1., DENEST(2), 1)

IF (M .NE. NODE) GO TO 20

C SUM1 = 0.0

Evaluate log likelihood and penalty

C Penalty

DO 160 K = 1, M

KM1 = MAX0(K-1,1)

KP1 = MIN0(K+1, M)

SUM1 = SUM1 + (DENEST(KM1) - 2.0*DENEST(K)+DENEST(KP1))**2

160 CONTINUE

SUM2 = 0.0

Log-likelihood

DO 170 I = 1, NOBS

IF (X(I).GE.BNDS(1) .AND. X(I).LE.BNDS(2)) THEN

CALL DISPT (1, X(I), 1, NODE, BNDS, DENEST, WK, WK,

W4, W6)

SUM2 = SUM2 + ALOG(DENEST(I))

170 CONTINUE

180 CONTINUE

SUM1 = 0.0

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

SUM2 = 0.0

DO 180 K = 1, M - 1

FKP1 = DENEST(KP1)

BN = BK

CONS = FK + FKP1

TEMP = CONS + FKP1

SUM1 = SUM1 + H2*TEMP/6.0 + 0.5*BN*CONS

SUM2 = SUM2 + H3*(TEMP+FKP1)/12.0 + H2*BN*TEMP/3.0 +

0.5*BN*CONS

180 CONTINUE

SUM1 = SUM1

SUM2 = SUM2 - SUM1*SUM1

C 9000 CALL EIPOP (."DISPL ")

Exit section

RETURN

/EOF
File _DUA01CNORMAL.INP112 (407/157:0), last revised on 29-NOV-1998 11:17, is a 1 block sequential file owned by UIC [DECNET]. The records are variable length with implied (CR) carriage control. The longest record is 39 bytes.

Job NORMAL (1815) queued to TERM1LA120A on 29-NOV-1998 11:17 by user DECNET; UIC [DECNET], user account DECNET at priority 100, started on printer LTA4; on 29-NOV-1998 11:24 from queue TERM1LA120A.
<table>
<thead>
<tr>
<th>No.</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000E+00</td>
<td>0.5000E+01</td>
<td>0.1000E-03</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.5000E+01</td>
<td>0.1000E-03</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.0000E+00</td>
<td>0.5049E-01</td>
<td>0.1325E+00</td>
<td>0.3412E+00</td>
<td>0.3127E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.4706E+00</td>
<td>0.6719E+00</td>
<td>0.2542E+00</td>
<td>0.7132E+00</td>
<td>0.3259E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.7450E+00</td>
<td>0.7193E+00</td>
<td>0.6884E+00</td>
<td>0.5957E+00</td>
<td>0.5005E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.4047E+00</td>
<td>0.3027E+00</td>
<td>0.2019E+00</td>
<td>0.1118E+00</td>
<td>0.4101E-01</td>
</tr>
<tr>
<td>7</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OUTPUT STATISTICS**

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000E+00</td>
<td>0.3434E+02</td>
<td>0.3913E+01</td>
<td>0.2274E+00</td>
<td></td>
</tr>
</tbody>
</table>

**NUMBER OF MISSING VALUES**

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LOG OF CURRENT CYCLES, LOG XMF**

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3274E+01</td>
<td>0.3434E+01</td>
<td>0.2946E+01</td>
<td>0.3089E+01</td>
<td>0.3210E+01</td>
</tr>
<tr>
<td>0.3332E+01</td>
<td>0.3453E+01</td>
<td>0.2574E+01</td>
<td>0.3696E+01</td>
<td>0.3817E+01</td>
</tr>
<tr>
<td>0.3446E+01</td>
<td>0.4467E+01</td>
<td>0.4709E+01</td>
<td>0.4910E+01</td>
<td></td>
</tr>
<tr>
<td>0.3153E+01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ORDERED LOG OF CURRENT CYCLES, LOG XMF**

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000E+00</td>
<td>0.3045E-02</td>
<td>0.1435E-01</td>
<td>0.3725E-01</td>
<td>0.7362E-01</td>
</tr>
<tr>
<td>0.1339E+00</td>
<td>0.1872E+00</td>
<td>0.2616E+00</td>
<td>0.3446E+00</td>
<td>0.4329E+00</td>
</tr>
<tr>
<td>0.2832E+00</td>
<td>0.5385E+00</td>
<td>0.6952E+00</td>
<td>0.8982E+00</td>
<td>0.9758E+00</td>
</tr>
<tr>
<td>0.1000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CDF PARAMETERS**

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7274E+01</td>
<td>0.3446E+01</td>
<td>0.2967E+01</td>
<td>0.3089E+01</td>
<td>0.3210E+01</td>
</tr>
<tr>
<td>0.3939E+01</td>
<td>0.4060E+01</td>
<td>0.4162E+01</td>
<td>0.3935E+01</td>
<td>0.4424E+01</td>
</tr>
<tr>
<td>0.4546E+01</td>
<td>0.4667E+01</td>
<td>0.4789E+01</td>
<td>0.4910E+01</td>
<td>0.5031E+01</td>
</tr>
<tr>
<td>0.3597E+01</td>
<td>0.3434E+01</td>
<td>0.3574E+01</td>
<td>0.3817E+01</td>
<td>0.4424E+01</td>
</tr>
<tr>
<td>0.5153E+01</td>
<td>0.4667E+01</td>
<td>0.4789E+01</td>
<td>0.4910E+01</td>
<td>0.5031E+01</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. RNLN - 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;
6.0 APPENDIX 2

FATIGUE STRENGTH DEGRADATION MODEL:
RANDOM3 AND RANDOM4 USER MANUAL
FATIGUE STRENGTH REDUCTION MODEL:
RANDOM3 and RANDOM4 USER MANUAL

Prepared by:
Lola Boyce, Ph.D., P.E.
Thomas B. Lovelace

APPENDIX 2
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

48
# 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>7</td>
</tr>
<tr>
<td>5.0 References</td>
<td>11</td>
</tr>
<tr>
<td>6.0 Appendix A: Physical Quantities, Symbols, and Units</td>
<td>12</td>
</tr>
<tr>
<td>7.0 Appendix B: RANDOM3 Sample Problem: Source, Input and Output Files</td>
<td>13</td>
</tr>
<tr>
<td>8.0 Appendix C: RANDOM4 Sample Problem Source, Input and Output Files</td>
<td>37</td>
</tr>
<tr>
<td>9.0 Appendix D: IMSL Subroutine Calls from RANDOM3 and RANDOM4</td>
<td>76</td>
</tr>
<tr>
<td>10.0 Appendix E: SAS/GRAPH Program for RANDOM3 and RANDOM4</td>
<td>77</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

This User Manual documents the FORTRAN programs RANDOM3 and RANDOM4. They are based on fatigue strength reduction, using a probabilistic constitutive model. They predict the random lifetime of an engine component to reach a given fatigue strength (see Section 2.0, Theoretical Background).

Included in this Manual are details regarding the theoretical backgrounds of RANDOM3 and RANDOM4, input data instructions and sample problems illustrating the use of RANDOM3 and RANDOM4. Appendix A gives information on the physical quantities, their symbols, FORTRAN names and both SI and U.S. Customary units. Appendix B and C include photocopies of the actual computer printout corresponding to the sample problems. Appendices D and E detail the IMSL, Version 10, subroutines and functions called by RANDOM3 and RANDOM4 and SAS/GRAPH programs that can be used to plot both the probability density functions (p.d.f.) and the cumulative distribution functions (c.d.f.).
2.0 THEORETICAL BACKGROUND

Fatigue strength data are usually presented as cycles to failure for each of several stress amplitudes, the familiar S-N diagram. Results indicate that for lower stress amplitudes the cycles (or time) to failure increases. Thus, a power curve fit through the data yields a monotonically decreasing curve. In general, this curve is represented as

\[ S = \left(\frac{N}{C'}\right)^{-1/m'} \]  

(6)

where the primitive variables in this equation are as follows: \( S \) is the applied constant amplitude alternating stress at failure or fatigue strength, \( N \) is number of cycles, \( C' \) is a material parameter that varies from specimen to specimen and \( m' \) is a material constant.\(^3\)

Equation (6) can be written in terms of "cycles to reach a given fatigue strength" as

\[ N = C'S^{-m'} \]  

(7)

Recently another fatigue strength reduction model has been proposed that takes into account the effect of temperature as well as other parameters that affect strength.\(^4\) The general form of the constitutive relationships for this model is applied to the constituents of high temperature composite materials. Specifically, it is applied herein for the case of a single material constituent. The mechanical property of interest is fatigue strength which is expressed in terms of primitive variables, including the general categories of temperature, mechanical cycles and mean stress. For these categories, the relationship becomes

\[ \frac{S}{S_0} = \left[ \frac{T_F - T}{T_F - T_o} \right]^n \left[ \frac{S_F - \sigma}{S_F - \sigma_o} \right]^m \left[ \frac{\log N_{MF} - \log N_{Mo}}{\log N_{MF} - \log N_{MO}} \right]^q \]  

(8)

where \( S \) is the applied constant amplitude alternating stress at failure (fatigue strength) at current (or operating) temperature, \( T \), mean stress, \( \sigma \), and mechanical cycle, \( N_M \). \( S_0 \) is fatigue strength at reference temperature, \( T_o \) (usually room temperature), reference mean stress (or residual stress), \( \sigma_o \), and reference mechanical cycle, \( N_{MO} \). Also, \( T_F \) is the final or melting temperature of the material, \( S_F \) is the final or tensile strength of the material, and \( N_{MF} \) is the final mechanical cycle or lifetime. Empirical parameters, \( n \), \( m \), and \( q \), are determined from available experimental data or estimated from anticipated behavior of the particular product term.\(^5\) Note that the term containing mechanical cycles is expressed in terms of the log of cycles rather than cycles. This formulation is attractive when \( N_M \) and \( N_{MO} \) are small compared to \( N_{MF} \). The equation may be solved for \( N_M \), or the "cycles to reach a given fatigue strength." The expression is

\[ N = 10 \exp \left[ \log N_{MF} - \left( \log N_{MF} - \log N_{MO} \right) \left( \frac{S}{S_0} \left[ \frac{T_F - T}{T_F - T_o} \right]^n \left[ \frac{S_F - \sigma}{S_F - \sigma_o} \right]^m \right]^{1/q} \right] \]  

(9)
For values typical of a cast nickel base-superalloy subjected to typical loads and temperatures, equation (9) indicates increasing life for decreasing temperature, decreasing tensile mean stress, and decreasing applied alternating stress. It indicates decreasing life for increasing temperature, decreasing compressive mean stress, and increasing applied alternating stress. Therefore, equation (9) predicts observed trends in general.

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 for RANDOM3. For RANDOM4, a p.d.f. of cycles is generated using the maximum entropy method. Maximum entropy uses Jaynes' principle which says that "the minimally prejudiced distribution is that which maximizes the entropy subjected to the constraints supplied by the given information."
3.0 INPUT DATA

Data input for RANDOM3 and RANDOM4 is user friendly and easy to manipulate (see, for example, the file entitled NORMAL.INP, in Section 4.0). The first twelve lines of input have the same format, 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:**

```
123456789012345678901234567890
   1        40
```

2. Ultimate Tensile Strength, SF

**EXAMPLE:**

```
123456789012345678901234567890
900.0000   45.0000
```

3. Log of Final Cycle, NMF

**EXAMPLE:**

```
123456789012345678901234567890
  8.0000     0.8000
```

4. Reference Fatigue Strength, SO

**EXAMPLE:**

```
123456789012345678901234567890
  500.0000  25.0000
```

5. Log of Reference Cycle, NMO

**EXAMPLE:**

```
123456789012345678901234567890
   7.0000    0.7000
```
6. Current Fatigue Strength, S

**EXAMPLE:**

| 123456789012345678901234567890 | 250.0000 | 12.0000 |

7. Residual Compressive Stress, SIGO

**EXAMPLE:**

| 123456789012345678901234567890 | 20.0000 | 1.0000 |

8. Current Mean Stress, SIG

**EXAMPLE:**

| 123456789012345678901234567890 | 150.0000 | 7.5000 |

9. Temperature Exponent, XXN, Stress Exponent, XXM, and Cycle Exponent, XXQ

**EXAMPLE:**

| 123456789012345678901234567890 | 0.5000 | 0.0150 |

10. Melting Temperature, TF

**EXAMPLE:**

| 123456789012345678901234567890 | 1500.0000 | 75.0000 |

11. Reference Temperature, TO

**EXAMPLE:**

| 123456789012345678901234567890 | 20.0000 | 0.6000 |
12. Current Temperature, T

EXAMPLE:

123456789012345678901234567890
850.0000   25.0000

13. The DESPL parameters are NODE, INIT, ALPHA, EPS, and MAXIT and are entered in that order as follows:

EXAMPLE:

123456789012345678901234567890
21   0     20.0000   1.0E-05   30

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

EXAMPLE:

1234567890
2
4.0 SAMPLE PROBLEMS FOR RANDOM3 AND RANDOM4

The objective of these programs is to predict the random lifetime to reach a given fatigue strength for an engine component. The theory is based on fatigue strength reduction, using a probabilistic constitutive model. The only difference between RANDOM3 and RANDOM4 is the method used to generate p.d.f. estimates. RANDOM3 uses maximum penalized likelihood, while RANDOM4 uses maximum entropy (see Section 2.0, Theoretical Background). RANDOM3 and RANDOM4 input parameters are given in Table A2.1.

<table>
<thead>
<tr>
<th>FORTRAN Name</th>
<th>Distribution Type</th>
<th>Mean</th>
<th>Standard Deviation (Value)</th>
<th>Standard Deviation (% of Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>normal</td>
<td>900.0</td>
<td>45.0</td>
<td>(3%)</td>
</tr>
<tr>
<td>NMF</td>
<td>lognormal</td>
<td>8.0</td>
<td>0.8</td>
<td>(10%)</td>
</tr>
<tr>
<td>SO</td>
<td>lognormal</td>
<td>500.0</td>
<td>25.0</td>
<td>(5%)</td>
</tr>
<tr>
<td>NMO</td>
<td>lognormal</td>
<td>7.0</td>
<td>0.7</td>
<td>(10%)</td>
</tr>
<tr>
<td>S</td>
<td>lognormal</td>
<td>250.0</td>
<td>12.5</td>
<td>(5%)</td>
</tr>
<tr>
<td>SIGO</td>
<td>lognormal</td>
<td>-20.0</td>
<td>-1.0</td>
<td>(1%)</td>
</tr>
<tr>
<td>SIG</td>
<td>lognormal</td>
<td>150.0</td>
<td>7.5</td>
<td>(5%)</td>
</tr>
<tr>
<td>XXN</td>
<td>normal</td>
<td>0.5</td>
<td>0.015</td>
<td>(0.3%)</td>
</tr>
<tr>
<td>XXM</td>
<td>normal</td>
<td>0.5</td>
<td>0.015</td>
<td>(0.3%)</td>
</tr>
<tr>
<td>XXQ</td>
<td>normal</td>
<td>0.5</td>
<td>0.015</td>
<td>(0.3%)</td>
</tr>
<tr>
<td>TF</td>
<td>normal</td>
<td>1500.0</td>
<td>45.0</td>
<td>(3%)</td>
</tr>
<tr>
<td>TO</td>
<td>normal</td>
<td>20.0</td>
<td>0.6</td>
<td>(3%)</td>
</tr>
<tr>
<td>T</td>
<td>normal</td>
<td>850.0</td>
<td>25.5</td>
<td>(3%)</td>
</tr>
</tbody>
</table>
The input is entered in the following format in a file entitled NORMAL.INP.

<table>
<thead>
<tr>
<th>1234567890123456789012345678901234567890</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>900.0000</td>
<td>45.0000</td>
</tr>
<tr>
<td>8.0000</td>
<td>0.8000</td>
</tr>
<tr>
<td>500.0000</td>
<td>25.0000</td>
</tr>
<tr>
<td>7.0000</td>
<td>0.7000</td>
</tr>
<tr>
<td>250.0000</td>
<td>12.5000</td>
</tr>
<tr>
<td>20.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>150.0000</td>
<td>7.5000</td>
</tr>
<tr>
<td>0.5000</td>
<td>0.0150</td>
</tr>
<tr>
<td>1500.0000</td>
<td>75.0000</td>
</tr>
<tr>
<td>20.0000</td>
<td>0.6000</td>
</tr>
<tr>
<td>850.0000</td>
<td>25.5000</td>
</tr>
<tr>
<td>21 0</td>
<td>20.00 1.0E-05 30</td>
</tr>
</tbody>
</table>

Execution of RANDOM3 and RANDOM4 (source code entitled NR3.FOR and NR4.FOR, respectively) produces files entitled RANDM33 and RANDM44. These give intermediate results (see Appendices B and C). Execution also produces plotfiles entitled PLOT1 and PLOT2 (see Appendices B and C). 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 RANDOM3 and RANDOM4. The plots are drawn from the plotfiles by the SAS/GRAPH graphing program (see Appendix D). These plots for the sample problem are shown Figures 1, 2, 3, and 4. This same sample problem has been reported in Boyce and Chamin. There, however, it utilized U.S. Customary units and older versions of RANDOM3 and RANDOM4 (using IMSL Version 9.2 subroutines).
Fig. A2.1  p.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum penalized likelihood method of p.d.f. generation.

Fig. A2.2  c.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum penalized likelihood method of p.d.f. generation.
Fig. A2.3  p.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum entropy method of p.d.f. generation.

Fig. A2.4  c.d.f. of log of mechanical cycles for fatigue strength reduction model, using maximum entropy method of p.d.f. generation.
5.0 REFERENCES

1 IMSL, "STAT/LIBRARY, FORTRAN Subroutines for Statistical Analysis", Houston, Texas


### 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 A2.2 Physical quantities, symbols, and units for fatigue crack growth model for RANDOM3 and RANDOM4.**

<table>
<thead>
<tr>
<th>Physical Quantity</th>
<th>Theory Symbol</th>
<th>FORTRAN Name</th>
<th>Units SI</th>
<th>Units U.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultimate Tensile Strength</td>
<td>SF</td>
<td>SF</td>
<td>MPa</td>
<td>ksi</td>
</tr>
<tr>
<td>Final Cycle (lifetime)</td>
<td>N_MF</td>
<td>NMF</td>
<td>dimensionless</td>
<td></td>
</tr>
<tr>
<td>Reference Fatigue Strength</td>
<td>SO</td>
<td>SO</td>
<td>MPa</td>
<td>ksi</td>
</tr>
<tr>
<td>Reference Cycles</td>
<td>N_MO</td>
<td>NMO</td>
<td>dimensionless</td>
<td></td>
</tr>
<tr>
<td>Current Fatigue Strengths</td>
<td>S</td>
<td>S</td>
<td>MPa</td>
<td>ksi</td>
</tr>
<tr>
<td>Residual Compressive Stress</td>
<td>σ₀</td>
<td>SIGO</td>
<td>MPa</td>
<td>ksi</td>
</tr>
<tr>
<td>Current Mean Stress</td>
<td>σ</td>
<td>SIG</td>
<td>MPa</td>
<td>ksi</td>
</tr>
<tr>
<td>Empirical Material Parameters</td>
<td>n</td>
<td>XXN</td>
<td>dimensionless</td>
<td></td>
</tr>
<tr>
<td></td>
<td>m</td>
<td>XXM</td>
<td>dimensionless</td>
<td></td>
</tr>
<tr>
<td></td>
<td>q</td>
<td>XXQ</td>
<td>dimensionless</td>
<td></td>
</tr>
<tr>
<td>Melting Temperature</td>
<td>TF</td>
<td>TF</td>
<td>°C</td>
<td>°F</td>
</tr>
<tr>
<td>Reference Temperature</td>
<td>TO</td>
<td>TO</td>
<td>°C</td>
<td>°F</td>
</tr>
<tr>
<td>Current Temperature</td>
<td>T</td>
<td>T</td>
<td>°C</td>
<td>°F</td>
</tr>
</tbody>
</table>
7.0 APPENDIX B

RANDOM3 SAMPLE PROBLEM: SOURCE, INPUT AND OUTPUT FILES
2025 FORMAT('NORMAL XNM')
WRITE(*,1001)((XNM(I),I=1,NTOT)
WRITE(*,1002)ISeed,NTOT
CALL RNSET(iseed)
CALL RNnor(NTOT,XNM)
DO 203 I=1,NTOT
XNM(I)=SG#*XXO(I)+YM
203 CONTINUE
WRITE(*,1997)
2026 FORMAT('NORMAL XNM')
WRITE(*,1001)((XNM(I),I=1,NTOT)
WRITE(*,1002)ISeed,NTOT
CALL RNSET(iseed)
CALL RNnor(NTOT,XNM)
DO 204 I=1,NTOT
XXO(I)=GS#*XXO(I)+YM
204 CONTINUE
WRITE(*,1997)
2027 FORMAT('NORMAL XXO')
WRITE(*,1001)((XXO(I),I=1,NTOT)
C NORMAL TEMPERATURE, TF, TO, T
C NORMAL FINAL (MELTING) TEMPERATURE, TF
WRITE(*,1002)ISeed,NTOT
READ(3,1011) YM,YS
WRITE(*,1011) YM,YS
CALL RNSET(iseed)
CALL RNnor(NTOT,TF)
DO 205 I=1,NTOT
TF(I)=TS#TF(I)+YM
205 CONTINUE
WRITE(*,1997)
2025 FORMAT('NORMAL TF')
WRITE(*,1001)((TF(I),I=1,NTOT)
C NORMAL REFERENCE TEMPERATURE, TO
READ(3,1011) YM,YS
WRITE(*,1011) YM,YS
CALL RNSET(iseed)
CALL RNnor(NTOT,TO)
DO 206 I=1,NTOT
TO(I)=TS#TO(I)+YM
206 CONTINUE
WRITE(*,1997)
2025 FORMAT('NORMAL TO')
WRITE(*,1001)((TO(I),I=1,NTOT)
C NORMAL CURRENT TEMPERATURE, T
READ(3,1011) YM,YS
WRITE(*,1011) YM,YS
CALL RNSET(iseed)
CALL RNnor(NTOT,T)
DO 207 I=1,NTOT
T(I)=TS#T(I)+YM
207 CONTINUE
WRITE(*,1997)
2025 FORMAT('NORMAL T')
WRITE(*,1001)((T(I),I=1,NTOT)
C CALCULATE LOG OF CURRENT CYCLES, LOG XNM
DO 102 I=1,NTOT
RX=(SF(I)-SIG(I))/(SF(I)-SIG(I))##XXN(I)
RE=(TF(I)-TO(I))/(TF(I)-TO(I))##XXN(I)
XNM#=(SF(I))/((SF(I)-SIG(I))##XXN(I)
XXM#=(SF(I))/((SF(I)-SIG(I))##XXN(I)
IF(XNM#2.0E0)XNM2=0.0
102 CONTINUE
WRITE(6,2028)
2028 FORMAT(' LOG OF CYCLES TO REACH MEAN FATIGUE STR = 
1/"MP A")
WRITE(6,1001)(XNM(I),I=1,NTOT)
C SORT LOG OF CYCLES
CALL SORT(XNM,NTOT)
WRITE(6,2029)
2029 FORMAT(' SORTED LOG OF CYCLES')
WRITE(6,1001)(XNM(I),I=1,NTOT)
C CALCULATE PDF OF LOG CURRENT CYCLES: LOG XNM
READ(3,1009)NODE,INIT,ALPHA,EPS,M0IT
WRITE(6,985)
985 FORMAT(' DESPL PARAMETERS')
WRITE(6,1009)NODE,INIT,ALPHA,EPS,M0IT
BNDS(1)=XNM(1) - 0.05*XNM(1)
BNDS(2)=XNM(NTOT) + 0.05*XNM(NTOT)
WRITE(6,979)BNDS(1),BNDS(2)
979 FORMAT(' BNDS(1),BNDS(2)=',E12.4,E12.4)
CALL DESPL(TOT,XNM,NODE,BNDS,INIT,ALPHA,M0IT,EPS,DENS,STAT
1NMSS)
WRITE(6,980)
980 FORMAT(' PDF OF LOG CURRENT CYCLES: LOG XNM,Y AXI S OF PDF PLOT')
WRITE(6,1001)(DENS(I),I=1,NODE)
WRITE(6,981)
981 FORMAT(' OUTPUT STATISTICS')
WRITE(6,1001)(STAT(I),I=1,4)
WRITE(6,982)
982 FORMAT(' NUMBER OF MISSING VALUES')
WRITE(6,1010)NMSS
C CALClATE WINDOW WIDTH, HH
HH=(BNDS(2)-BNDS(1))/4(NODE-1)
C CALClATE VALUES OF LOG OF CURRENT CYCLES AT WHICH PDF IS ESTIMATED
ALSO CALLED "NODE" VALUES
DO 6001,I=1,NODE-2
BNDS(I+2)=BNDS(I) + (I+HH)
6001 CONTINUE
983 FORMAT(' LOG OF CURRENT CYCLES, LOG XNM')
WRITE(6,1001)(BNDS(I),I=1,NODE)
C REORDER BNDS FOR PLOTTING
SAVE1 = BNDS(2)
SAVE2 = BNDS(NODE)
RNDS=NODE-2,1
DO 6002, I=1,NODE-2
BNDS(I+1)=BNDS(I+2)
6002 CONTINUE
RBNS(NODE-1)=SAVE2
BNDS(NODE)=SAVE1
WRITE(6,984)
984 FORMAT(' ORDERED LOG OF CURRENT CYCLES, LOG,XNM')
WRITE(6,1001)(BNDS(I),I=1,NODE)
C WRITE LOG OF CURRENT CYCLES AND PDF OF LOG OF CURRENT CYCLES
C LOG XNM TO PLOT FILES
WRITE(34,990)
990 FORMAT(' (E12.4,E12.4)')
WRITE(*,991)(BNDST(J),DENG(J),J=1,NODE)

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

END

CALCULATE CDF OF LOG OF CURRENT CYCLES

READ(3,1010)I0PT
WRITE(6,992)
992 FORMAT(20CDF PARAMETERS')
WRITE(6,1010)I0PT
X0=BNDST(1)
DO 5003=1,NODE
F=CDF(X0,POPT,NODE,BNDST,DENS)
BNDSTX(I)=X0
X0=X0+DENS
DISTX(I)=F
5003 CONTINUE
WRITE(6,994)
994 FORMAT(20CDF OF LOG OF CURRENT CYCLES, LOG XHM,
LY AXI0 OF PDF, CDF PLOT')
WRITE(6,1001)(DISTX(I),I=1,NODE)
C
WRITE(6,993)
993 FORMAT(20ORDERED LOG OF CURRENT CYCLES, LOG XHM,
IX AXIS OF PDF, CDF PLOT')
WRITE(6,1001)(BNDST(I),I=1,NODE)
WRITE(6,1001)(BNDSTX(I),I=1,NODE)
WRITE LOG OF CURRENT CYCLES AND CDF OF LOG OF CURRENT
TO THE PLOT FILES
WRITE(35,990)
WRITE(35,991)(BNDST(J),DISTX(J),J=1,NODE)
STOP
END

SUBROUTINE SORT(Y,N)
DIMENSION Y(10000)
N=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
END

IMSL Name: D3SPL/DO3SPL (Single/Double precision version)

Computer: IBM/SINGLE

Revised: November 1, 1985

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

Usage: CALL-D3SPL(NOBS,X,NODE,BNDST,INIT,ALPHA,MAXIT,eps,
DENS,STAT,HESS,LDHESS,ILDFH,DDENS,DENEST,B,
IPUT,WK2)

Arguments:
NOBS - Number of observations. (Input)
INTRINSIC ALGB, AMAX1, MAX0, MIND, MOD, SORT
INTEGER MAX0, MIND, MOD
REAL ALGB, AMAX1, SORT

EXTERNAL EIMES, EIPOP, EIPSH, EISSL, EISTR, SADD, SACY, SACT
SADD, SACY, SACT

EXTERNAL SMIN, SIROD, SDOT, SNRM2, SSUM

INTEGER ISMIN, ISMID, SDOT, SNRM2, SSUM

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

CALL EIPSH ('DISPL ') Error checks
NCR = 1,
IF (NCR .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 (NODS .LE. 4) THEN
CALL EISTR (1, NODS, 'The number of mesh '//'
'nodes, NODS, must be an odd integer greater '//'
'than 4.')
ELSE IF (MOD (NODS, 2) .EQ. 0) THEN
CALL EISTR (1, NODS, 'NODS = %I(II) must be an odd integer '//'
'greater than 4. ')
ELSE IF (ALPHA .LE. 0.0) THEN
CALL EIMES (5, 4, 'ALPHA = %R(I1). The penalty weighting '//'
'factor which controls smoothness, ALPHA, must '//'
'be greater than 0. ')
END IF
ELSE IF (MAXIT .LE. 0.0) THEN
CALL EISTR (1, MAXIT, 'The maximum number '//'
'of iterations, MAXIT, must be greater than 0. ')
ELSE IF (BND(1) .GT. BND(2)) THEN
CALL EISTR (1, BND(1), 'The minimum value for X, BND(1), must '//'
'be less than or equal to the maximum value for '//'
'X, BND(2). ')
ELSE IF (INIT .NE. 0) THEN
IF (DENS(1).NE.0 .OR. DENS(NODS).NE.0) THEN
CALL EISTR (1, DENS(NODS), 'The initial estimates of the '//'
'density, DENS, must be greater than or '//'
'equal to 0. ')
END IF
END IF
I
70 FACTOR = 2.0*ALPHA/H3  Initialize mesh node densities
C   IF (INIT.EQ.0) THEN  Via DESFT
C     CALL D253T (M-2, B(2), 1, MOLD, BNDS, DENS, DENSEST, WK, WK,  
C        TEMP = 1.0/(M*M*M)  
C        DO 80 I=2, M-1  
C        DENS(I) = AMAX1(TEMP,SORT(DENEST(I-1:I)))  
C     END IF
C     ELSE  Via the initial estimates
C       DO 80 I=2, M-1  
C       DENS(I) = SORT(DENS(I))  
C     END IF
C     DENS(M) = 0.0  Maximize
C    DO 140 ITER=1, MAXIT  Get Hessian - Lagrange
C       HESS(1,1) = 0.0  
C       HESS(2,2) = 0.0  
C       BSMALL = 0.0  
C       SUM = 0.0  
C       CK** are true estimates = FKK**2
C       DO 120 K=2, M-1
C          KM1 = K - 1  
C          KM2 = MAX0(1,K-2)  
C          KP1 = K + 1  
C          KP2 = MINO(K+K2)  
C          FK = DENS(K)  
C          FKM1 = DENS(KM1)  
C          FKM2 = DENS(KM2)  
C          CKM1 = FKM1**2  
C          CK = FKK**2  
C          CKP1 = DENS(KP1)**2  
C          CKP2 = DENS(KP2)**2  
C          BK = B(K)  
C          BKM1 = B(KM1)  
C          SUM = SUM + CK  
C          IF (K.GE.4) HESS(1,KM1) = 4.0*FK*FKM2*FACTOR
C         DO 100 I=ILOHI(K,1), ILOHI(K,2)
C            TEMP = (X(I)-BK)/H  
C            CONS = (1.0-TEMP)/(CK+(CKP1-CK)*TEMP)  
C            SUM1 = SUM1 + CONS  
C            SUM2 = SUM2 + CONS*CONS  
C         END DO
C         SUM3 = SUM1 + CONS*CONS  
C     END DO
C     CONTINUE
C     CKCM1 = CK - CKM1  
C     DO 110 I=ILOHI(KM1,1), ILOHI(KM1,2)
C        CONS = (X(I)-BKMK1)/H  
C        TEMP = CKMK1 + CKCM1*CONS  
C        SUM1 = SUM1 + CONS/TEMP  
C        SUM2 = SUM2 + CONS*CONS/TEMP  
C        SUM3 = SUM3 + CONS*CONS/(1.0-CONS)/TEMP  
C     END DO
C     CONTINUE
C     TEMP = FACTOR*(CKM2+CKP2-4.0*(CKM1+CKP1)+6.0*CK) + SUM1  
C     TEMP = 2.0*TEMP  
C     BSMALL = BSMALL + 2.0*CK*TEMP
HESS(3,KM1) = TEMP + 4.0*CK*(5.0*FACTOR+SUM2)

IF (K,NE,2) HESS(2,KM1) = 4.0*FK*FM1*(4.0*FACTOR+SUM3)
DENEST(KM1,1) = FK*TEMP
DENEST(KM1,2) = -2.0*FK

CONTINUE
Bsmall = 1.0/H - SUM + BSALL

CALL SCOPY (M-2, DEHST(1,3); 1, DEHST(1,3); 1)
Save portion of DEHST

CALL SADD (M-2, -BSMALL/(2.0*SUM), HESS(3,1); LDHSS)
Finish with the hessian

CALL SCOPY (M-4, HESS(1,3); LDHSS, HESS(5,1); LDHSS)
Fill out symmetric band structure

HESS(S-M-3) = 0.0
HESS(S-2) = 0.0
HESS(S-2) = 0.0

CALL SCOPY (M-3, HESS(2,2); LDHSS, HESS(4,1); LDHSS)

CALL SCOPY (M-4, HESS(4,2); LDHSS, HESS(4,1); LDHSS)

CALL LTRB (M-2, HESS, LDHSS, 2, 2, HESS, LDHSS, IPT, WK2)
Solve symmetric band linear system

CALL LFSRB (M-2, HESS, LDHSS, 2, 2, HESS, LDHSS, IPT, WK2)

IF (NRC(1), .NE, 0) GO TO 9000

CONS = SDOT(M-2, DEHST(1,3); 1, DEHST(1,3); 1)/CONS
Compute the constant

CONS = (1.0/H-SUM-SDOT(M-2, DEHST(1,3); 1, DEHST(1,3); 1))/CONS

CALL SAVXY (M-2, CONS, DEHST(1,2); 1, DEHST(1,3); 1)
Parameter updates

CALL SAVXY (M-2, -1.0, DEHST(1,3); 1, DEHST(1,3); 1)
Parameter updates

TEMP = SGN(M-2, DENS(1)); 1)
Check the convergence criterion

IF (SRN2(M-2, DENS(1)); 1, LT, EPS1*TEMP) GO TO 150
Ad hoc projection to plus quadrant

TEMP = TEMP*MAX(4/SORT(M-2,0))

DO 130, I=2, M-1

DENS(I) = MAXI(TEMP, DENS(I))

130 CONTINUE

CALL EISTI (1, MAXI)

CALL EIMES (3, 1, 'The maximum number of iterations ' ' (MAXI=1(1)) was exceeded. ')

C 150 CALL SHPRED (M-2, DENS(2); 1, DENS(2); 1, DENS(2); 1)

IF (M, NE. NODE) GO TO 20
Evaluate log likelihood and penalty

SUM1 = 0.0
Penalty

DO 160 K=1, M

KM1 = MAXO(K-1, K)
KPI = MINO(K+1, M)

SUM1 = SUM1 + (DENS(KM1)-2.0*DENS(K)+DENS(KPI))**2

160 CONTINUE

STAT(2) = -0.5*FACTOR*SUM1

SUM2 = 0.0

DO 170 I=1, M, NOS

IF (X(I)>BND(1) AND X(I)<BND(2)) THEN

CALL D2SPT (1, X(I), 1, NODE, BDNS, DENS, DEHST, WK, WK)

SUM2 = SUM2 + ALOG(DEHST(1,1))

170 CONTINUE

STAT(1) = SUM2
Evaluate M.L.P.E. mean and variance
File _DUA0:ENORMAL.INFI41 (252,111,0), last revised on 22-DEC-1988 13:01, is a 1 block sequential file owned by UIC [DECNET]. The records are variable length with implied (CR) carriage control. The longest record is 39 bytes.

Job NORMAL (129) queued to TERMSLAL120A on 22-DEC-1988 13:01 by user DECNET, UIC [DECNET], under account DECNET at priority 100, started on printer LTA41 on 22-DEC-1988 13:03 from queue TERMSLAL120A.
File DBAO:J1PLOT1.CPP:1 (359,209,0), last revised on 23-NOV-1988 11:26, is a 2 block sequential file owned by UIC (11,11). The records are variable length with FORTRAN (FTN) carriage control. The longest record is 21 bytes.

The document contains a printed page with a diagram labeled "NETMONPRIV" and "PLOTZ". The page is marked as of poor quality and contains a file description with details about the file and job details for a job named "PLOTZ". The page also includes a note about the quality of the print.
8.0 APPENDIX C

RANDOM4 SAMPLE PROBLEM: SOURCE, INPUT AND OUTPUT FILES
C LOGNORMAL FATIGUE STRENGTH AT REFERENCE CONDITIONS, S0
WRITE(6,1005) ISEED,NTOT
READ(5,1006) XM,XS
WRITE(6,1006)XM,XS
XM = 500.
XS = 25.
YS = SQRT(LOG(1.0+(XS/XM)**2)) 
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNRLN(NTOT,YM,YS,50)
WRITE(20,1001) (SO(I),I=1,NTOT)
WRITE(6,2022)
2022 FORMAT(35HLOGNORMAL SO',)
WRITE(6,1001) (SO(I),I=1,NTOT)
C LOGNORMAL LOG OF REFERENCE CYCLES, XLNMO
WRITE(6,1005) ISEED,NTOT
READ(5,1006)XM,XS
WRITE(6,1006)XM,XS
XM = 25.
XS = 25.
YS = SQRT(LOG(1.0+(XS/XM)**2)) 
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNRLN(NTOT,YM,YS,XLNMO)
WRITE(21,1001)(XLNMO(I),I=1,NTOT)
WRITE(6,2023)
2023 FORMAT(35HLOGNORMAL XLNMO')
WRITE(6,1005) ISEED,NTOT
READ(5,1006)XM,XS
WRITE(6,1006)XM,XS
XM = 125.
YS = SQRT(LOG(1.0+(XS/XM)**2)) 
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNRLN(NTOT,YM,YS,5)
WRITE(22,1001)(XLNMO(I),I=1,NTOT)
WRITE(6,2024)
2024 FORMAT(35HLOGNORMAL S')
WRITE(6,1001) (SO(I),I=1,NTOT)
C DEFINE RANDOM STRESSES
C LOGNORMAL REFERENCE STRESS, SIGO
WRITE(6,1005) ISEED,NTOT
READ(5,1006)XM,XS
WRITE(6,1006)XM,XS
XM = 20.
XS = 1.
YS = SQRT(LOG(1.0+(XS/XM)**2)) 
YM = LOG(XM) - 0.5*YS**2
CALL RNSET(ISEED)
CALL RNRLN(NTOT,YM,YS,SIGO)
C CHANGE SIGO TO NEGATIVE VALUES FOR COMPRESSIVE
C RESIDUAL STRESSES.
DO 401 I = 1,NTOT
SIGO(I) = -SIGO(I)
401 CONTINUE
WRITE(6,2036)
2036 FORMAT(35HLOGNORMAL SIGO')
WRITE(6,1005) ISEED,NTOT
C LOGNORMAL CURRENT STRESS, SIG.
WRITE(6,1005) ISEED,NTOT

```fortran
READ(S,1006) XM, XS
WRITE(S,1006) XM, XS
YM = 1500.
YM = 1500.
YM = LOG(XM) - 0.5*YS**2.
CALL RNSET(ISEED).
CALL RNMLC(NTOT,YM,YS,SIG).
WRITE(27,1001)(SIG(I),I=1,NTOT)
WRITE(4,2037).
2037 FORMAT('LOGNORMAL SIG')
WRITE(4,1001)(SIG(I),I=1,NTOT)
C NORMAL EXPONENTS XM*XM*XXD
YM = 0.5
YS = 0.015
WRITE(4,1005) ISEED, NTOT
READ(S,1006) YM,YS
WRITE(S,1006) YM,YS
CALL RNSET(ISEED).
CALL RNMLC(NTOT,XXN)
DO 101 I=1,NTOT
XXN(I) = YM*XXN(I) + YM
101 CONTINUE
WRITE(23,1001)(XXN(I),I=1,NTOT)
WRITE(4,2025).
2025 FORMAT('NORMAL XXN')
WRITE(4,1001)(XXN(I),I=1,NTOT)
WRITE(4,1005) ISEED, NTOT
CALL RNMLC(NTot,XXN)
CALL RNSET(ISEED).
CALL RNMLC(NTot,XXN)
DO 201 I=1,NTOT
XXN(I) = YM*XXN(I) + YM
201 CONTINUE
WRITE(24,1001)(XXN(I),I=1,NTOT)
WRITE(4,2026).
2026 FORMAT('NORMAL XXN')
WRITE(4,1001)(XXN(I),I=1,NTOT)
WRITE(4,1005) ISEED, NTOT
CALL RNMLC(NTot,XXN)
CALL RNSET(ISEED).
CALL RNMLC(NTot,XXN)
DO 301 I=1,NTOT
XXQ(I) = YM*XXQ(I) + YM
301 CONTINUE
WRITE(25,1001)(XXQ(I),I=1,NTOT)
WRITE(4,2027).
2027 FORMAT('NORMAL XXQ')
WRITE(4,1001)(XXQ(I),I=1,NTOT)
C DEFINE DETERMINISTIC TEMPERATURES
TF = 1500.
TF = 1500.
DO 20 TF = 1, NTOT
WRITE(5,1008) TF,
WRITE(5,1008) TF,
CALL RNSET(ISEED).
CALL RNMLC(NTot,TF)
DO 405 I=1,NTOT
TF(I) = YM*TF(I) + YM
405 CONTINUE
WRITE(6,2046)
```

ORIGINAL PAGE IS OF POOR QUALITY
2046 FORMAT(16 NORMAL T) WRITE(6,1001)(TF(I),I=1,NTOT)
C NORMAL REFERENCE TEMPERATURE TO
WRITE(6,1005)(SEED,NTOT)
READ(5,1006)YM,YS
TM=20.
YS=0.6
CALL RNSEED(SEED)
CALL RNHOR(NTOT)
DO 406 I=1,NTOT
TM(I)=Y(I)*TM+YS
406 CONTINUE
WRITE(6,2047)
2047 FORMAT(16 NORMAL T)
WRITE(6,1001)(TD(I),I=1,NTOT)
C NORMAL CURRENT TEMPERATURE T
WRITE(6,1005)SEED+NTOT
READ(5,1006)YM,YS
WRITE(6,1006)TM+YS
C YM=850.
YS=42.3
CALL RNSEED(SEED)
CALL RNHOR(NTOT)
TM=YS*TM+YS
407 CONTINUE
WRITE(6,2048)
2048 FORMAT(16 NORMAL T)
WRITE(6,1001)(TI(I),I=1,NTOT)
C CALCULATE CURRENT LOG OF CYCLES: LOG XNM
DO 100 I=1,NTOT
R=(SF(I)-SIGO(I))/SF(I)-SIGO(I)
XNM(I)=R
WRITE(6,876)R
100 CONTINUE
C6876 FORMAT(16 'E12.4')
C TEMP=((TF(I)-TD(I))/TF(I)-TD(I))**XNM(I)
WRITE(6,7876)TEMP
C7876 FORMAT(16 'E12.4')
C SS=(SF(I)-SIGO(I))
XXO=0.5*SF(I)
WRITE(6,1001)SS
WRITE(6,1001)XXO
XNM1=(SF(I)-SIGO(I))**SF(I)/SF(I)
C WRITE(6,8876) XNM1
C8876 FORMAT(16 'E12.4')
XNM2=(XXO(XM1)-XNM(XM1))*XNM(XM1))
WRITE(6,8875)XNM2
C8875 FORMAT(16 'E12.4')
IF(XM2.LT.0.000)XNM2=0.0
XNM(I)=XNM2
XNM(I)=10.**XNM2
C 102 CONTINUE
WRITE(29,1001)(XNM(I),I=1,NTOT)
WRITE(6,2028)
2028 FORMAT(16 LOG OF CYCLES TO REACH MEAN FATIGUE STR = 
350 MPa
WRITE(6,1001)(XNM(I),I=1,NTOT)
C SORT LOG OF CYCLES
CALL SORT(XNM,NTOT)
WRITE(29,1001)(XNM(I),I=1,NTOT)
WRITE (6,2029)
C FORMAT('SORTED LOG OF CYCLES')
WRITE (A,1001)(XNM(I),I=1,NMT)
C CALCULATE PDF OF LOG OF CURRENT CYCLES, LOG XNM
C USING THE MAXIMUM ENTROPY METHOD
C CALCULATE SAMPLE MOMENTS, SM
C NUMBER OF MOMENTS, MM
MM=4
CALL SMXM(XNM,MM,NMT,SM)
WRITE(30,1001)(SM(I),I=1,MM)
WRITE(6,2038)
C FORMAT('SAMPLE MOMENTS')
WRITE(6,1001)(SM(I),I=1,MM)
C OBTAIN MAXIMUM ENTROPY DISTRIBUTION
START=1
DATA=1
C CALCULATE MAX AND MIN ORDINATES FOR PDF (AND CDF)
BNDS(2) = XNM(1) - 0.05*XNM(1) - 1
BNDS(2) = XNM(NMT) + 0.05*XNM(NMT)
WRITE (16,9877) BNDS(1),BNDS(2)
WRITE (6,9877) BNDS(1),BNDS(2)
9877 FORMAT ('BNDS(1),BNDS(2)=',F12.4,1X,F12.4)
CALL RPI(MM,SM,BNDS(1),BNDS(2),0,XP,START,DATA,AL,CUM)
WRITE(31,1001)(AL(I),I=1,MM)
WRITE(6,2039)
C FORMAT('LAGRANGIAN MULTIPLIERS')
WRITE(6,1001)(AL(I),I=1,MM)
C CALCULATE VALUES OF ORDINATES FOR PDF (AND CDF)
C NUMBER OF ORDINATES USED
C CALCULATE WINDOW WIDTH, NH
NH=1
NH=(BNDS(2)-BNDS(1))/(NODE-1)
C CALCULATE VALUES OF LOG OF CURRENT CYCLES AT WHICH PDF IS ESTIMATED;
ALSO CALLED 'NODE'-VALUES
DO 6001,NODE-2
BNDS(NODE) = BNDS(I)
6001 CONTINUE
WRITE(6,983)
983 FORMAT('LOG OF CURRENT CYCLES; LOG XNM')
WRITE(6,1001)(BNDS(I),I=1,NODE)
C REORDER BNDS FOR PLOTTING
SAVE1 = BNDS(2)
BNDS(NODE) = BNDS(2)
BNDS(1) = BNDS(1)
DO 6002, NODE-2
BNDS(NODE-1) = SAVE2
BNDS(NODE) = SAVE1
6002 CONTINUE
WRITE(6,984)
984 FORMAT('ORDERED-LOG OF CURRENT CYCLES; LOG XNM, 1X AXIS PDF, CDF PLOT')
BNDS(NODE) = BNDS(1)
WRITE(6,984)
C CALCULATE VALUES OF THE PDF AT EACH ORDINATE
DO 1001, NODE
C FOR 4 MOMENTS THERE ARE 5 LAGRANGIAN MULTIPLIERS
DENS(I) = EXP(AL(I))*(AL(1)*BNDS(I)+AL(2)*BNDS(I)**2+
AL(3)*BNDS(I)**3+AL(4)*BNDS(I)**4)
1001 CONTINUE
C WRITE LOG OF CURRENT CYCLES AND PDF OF LOG OF CURRENT CYCLES,
C LOG XHM TO PLOT FILES
WRITE(34,990)
990 FORMAT(2E12.4,1X,E12.4)
WRITE(34,991)(BNDS(J),DENS(J),J=1,NODE)
991 FORMAT(E12.4,1X,E12.4)
C CALCULATE CDF OF LOG OF CURRENT CYCLES
IOPT=2
REM(3*1004)IOPT
WRITE(6,992)
992 FORMAT('CDF PARAMETERS')
WRITE(6,1004)IOPT
XX=BNDS(I)
UU=UU3*IOPT+NODE,BNDS,DENS
PN=GCDF(XX+IOPT+NODE,BNDS,DENS)
XX=XX+IOPT+NODE
DISTX(I)=P
6003 CONTINUE
WRITE(6,994)
994 FORMAT('CDF OF LOG OF CURRENT CYCLES, LOG XHM,
LX AXIS OF PDF, CDF PLOT')
WRITE(6,1001)(DISTX(I),I=1,NODE)
C WRITE(6,993)
993 FORMAT('ORDERED LOG OF CURRENT CYCLES, LOG XHM,
LX AXIS OF PDF, CDF PLOT')
WRITE(6,1001)(BNDS(I),I=1,NODE)
WRITE(6,1001)(BNDSX(I),I=1,NODE)
WRITE LOG OF CURRENT CYCLES AND CDF OF LOG OF CURRENT
to the PLOT FILES
WRITE(35,990)
WRITE(35,991)(BNDS(J),DISTX(J),J=1,NODE)
STOP
END
C SUBROUTINE SORT(Y,N)
DIMENSION Y(10000)
C Y IS THE ARRAY TO BE SORTED
C AT COMPLETION Y(N) IS SMALLEST VALUE
N1 = N
DO 1 I=1,N1
J = I+1
DO 2 K=J,N
IF (Y(I),LT,Y(K)) GO TO 2
Y(I) = Y(K)
Y(K) = TEMP
2 CONTINUE
1 CONTINUE
RETURN
END
C SUBROUTINE SMOM(X,M,NSAMP,SM)
C CALCULATES SAMPLE CENTRAL MOMENTS
C X(I) = SAMPLE VALUES, DIMENSION NSAMP
C M = NUMBER OF MOMENTS DESIRED
C NSAMP = SAMPLE SIZE
C SM = VALUE OF MOMENTS, DIMENSION M
DIMENSION X(10000),SM(10)
C. CALCULATE MEAN

\[ \text{SUM} = 0.0 \]

DO 1 I=1,NSAMP
    SM(I) = SUM/FLOAT(NSAMP)
1 CONTINUE

C. CALCULATE VARIANCE

\[ \text{SUM} = 0.0 \]

DO 2 I=1,NSAMP
    SM(I)**2 = SUM+(X(I)**2-SM(I)**2)
2 CONTINUE

C. CALCULATE HIGHER MOMENTS

DO 3 J=1,NSAMP
    SM(J)**3 = SUM+(X(J)**3-SM(J)**3)
3 CONTINUE

SUBROUTINE MEPI(N,C,M,XMIN,XMAX,XNF,XP,KSTART,KDATA,AL,CUM)
IMPLICIT REAL*8 (A-H,O-Z)

DIMENSION AL(*)
COMMON/FAIL/,NFAIL
COMMON/HELP/S(101),XX(14,101),C(8),M

WRITE THE INPUT DATA

IF (KDATA.EQ.0.0) GO TO 1
WRITE (6,25) KDATA
WRITE (6,26) KPRINT
WRITE (6,27) M
WRITE (6,28) XMAX
WRITE (6,29) XMIN
WRITE (6,30) (CM(I),I=1,4)
WRITE (6,31) (CM(I),I=5,N)
WRITE (6,32) TOL
WRITE (6,33) NXP

1 CONTINUE

MFAIL = 0
M=31
X2MIN = 0.0
X2MAX = 1.
C = 100
CC = CM(I)

CALL TRNI(XMAX,XMIN,CC,X2MAX,X2MIN,N)

C. CALCULATE THE MOMENTS ABOUT THE ORIGIN FOR THE MODIFIED LIMITS

STORE THEM IN COMMON IN C
CALL CONVER(CC,N)

GENERATE THE SIMPSON MULTIPLIERS AND STORE THEM IN HELP COMMON

CALL SIMON

GENERATE THE X*S POWER FOR SUBROUTINE FUNCT, STORE THEM IN HELP COMMON ARRAY

CALL MULTI (X2MAX,X2MIN,N)

DEFINE THE INPUT DATA FOR SUBROUTINE MPOPT

ETA(1)=1.0-12
ETA(2)=TOL
ETA(3)=1.0-24
ETA(4)=1.0-24
MODE=1
UMIN=0.0

WRITE THE INTERMEDIATE RESULTS YOU HAVE OBTAINED SO FAR

IF (KPRINT.EQ.0) GO TO 2
WRITE (6,34)
WRITE (6,35) M
WRITE (6,36) X2MAX,X2MIN
WRITE (6,37) (CC(I),I=1,4)
IF (N.GT.4) WRITE (6,22) (CC(I),I=5,N)
WRITE (6,38) (C(I),I=1,4)
IF (N.GT.4) WRITE (6,22) (C(I),I=5,N)
WRITE (6,39) (ETA(I),I=1,4)
CONTINUE

FIND A STARTING POINT FOR SUBROUTINE MPOPT TO START THE OPTIMIZATION ALGORITHM

IF (KSTART.EQ.0) GO TO 16
IF (KSTART.EQ.4) WRITE (6,44)
CALL START (X2MAX,X2MIN,AL,KSTART,CC,N,KPRINT,UMIN,MODE,MAXFN,ETA)
IF (NFAIL.EQ.1) GO TO 9

PRINT THE STARTING VALUES

IF (KPRINT.EQ.0) GO TO 7
GO TO (3,4,5,6) KSTART
WRITE (6,40)
WRITE (6,41) (AL(I),I=1,4)
IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
GO TO 7
WRITE (6,42)
WRITE (6,41) (AL(I),I=1,4)
IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
GO TO 7
WRITE (6,43)
WRITE (6,41) (AL(I),I=1,4)
IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
GO TO 7
WRITE (6,44)
WRITE (6,41) (AL(I),I=1,4)
IF (N.GT.4) WRITE (6,22) (AL(I),I=5,N)
GO TO 7
CONTINUE

CHANGE STARTING VALUES TO 0-1 DOMAIN FOR KSTART=0

range=XMAX-XMIN
THE ALGORITHM IS SIMILAR TO TRN2 BUT APPEARS TO GIVE BETTER NUMERICAL RESULTS.

IF (ABS(XMIN), LT, 1.E-10) GO TO 19
DO 17 I=2, NPL
ALS(I)=0.0
11 I=11
DO 18 J=11, N
ALS(I)=ALS(I)+FACTO(J)*XMIN*(J-I)*RANGE*I1/AL(J+1)*FACTO(J-I-I)
18 CONTINUE
17 CONTINUE
GO TO 50
DO 20 I=2, NPL
20 ALS(I)=RANGE*(I-1)*AL(I)
CONTINUE
50 DO 51 I=1, N
51 AL(I)=ALS(I+1)
CONTINUE
MFAIL=0
IF (KPRINT.EQ.0) GO TO 8
WRITE (6, 45)
8 CONTINUE
AL(N+1)=2.0
CALL MPOPT (AL, N, ETA, UMIN, MAXFN, MODE, KPRINT)
IF (MFAIL.EQ.0) GO TO 10
IF (KSTART.EQ.4) GO TO 9
THE PROGRAM HAS FAILED SO FAR, TRY ANOTHER STARTING POINT AND TRY AGAIN.
KSTART=KSTART+4
IF (KSTART.EQ.4.AND. N.LE.2) GO TO 9
GO TO 2
CONTINUE
WRITE (6, 46)
CALL EXIT
CONTINUE
CALCULATE THE ZERO TH LAGRANGIAN MULTIPLIER
SUM=0.0
DO 12 I=1, N
12 K=I+1
DO 11 K=I+1, N
SZ=SZ+AL(K)*X(K, I)
11 CONTINUE
SUM=SUM+SZ*EXP(SZ)
12 CONTINUE
NPL=N+1
DO 13 I=1, N
K=I+1
13 AL(K)=AL(K-1)
CONTINUE
DELTA=(X2MAX-X2MIN)/FLOAT(N-1)
AL(I)=ALG(SUM*DELTA/3.)
WRITE (6, 101) SUM
101 FORMAT (24H SUM OF RESIDUALS SQUARED=.E12.5)
IF (KPRINT.EQ.0) GO TO 14
WRITE (6, 47) (AL(I), I=1, NPL)
CONTINUE
C.... RESET KSTART TO ZERO
ICALCULATE THE LAGRANGIAN MULTIIPLIERS FOR THE ORIGINAL LIMITS

CALL TRN2 (XMAX, XMIN, AL, XMAX, XMIN, N)

CALCULATE THE CUMULATIVE DISTRIBUTION FUNCTION VALUE AT THE GIVEN POINT

IF (NXP.EQ.0) RETURN

DO 15 I = 1, NXP

CLUDF (XMIN, XMAX, XP(I), AL, NFXP)

15 CONTINUE

RETURN

21 FORMAT (5X, 'INPUT DATA IS PRINTED OUT FOR KDATA = 1 ONLY...KDATA =

22 I185, 'INTERMEDIATE OUTPUT EVERY KPRINT (TH) CYCLE...KPRINT =

23 I185, 'NUMBER OF KNOWN FIRST MOMENTS...

24 N= L185, 'HIGHER LIMIT...

25 XMAX = L185, 'LOWER LIMIT...

26 XMIN = L185, 'FIRST MOMENTS...

27 CC: (1) = L185, 'THE ALLOWED TOLERANCE IN LAGRANGIAN EQUATIONS...TOL =

28 L185, 'THE CUMULATIVE DISTRIBUTION REQUIRED AT NXP POINTS.NXP =

29 L185, 'INTERMEDIATE RESULTS FOR SUBROUTINE...MEP1' '20X33(' ')

30 10X41(' ')/

31 MAX...

32 M =

33 XMAX...XMIN =

34 CC(I) =

35 C(I) =

36 ETA(I) =

37 AL(I) =

38 UNIFORM ASSUMPTION STARTING METHOD...AS35( ' ')/

39 FORMAT ( 'STEP BY STEP STARTING METHOD...AS29( ' ')/

40 RESIDUALS...NO.' '10X...RESIDUALS X(1) X(2)

41 R(3) R(4) R(3) R

42 THE PROGRAM HAS FAILED!

43 WRITE (6, 49)

44 FORMAT (53H WARNING - MEAN IS NEARLY ZERO AND MEP1 WILL NOT WORK/12

45 1'H TRANSFORM X')
SUBROUTINE MPOPT (X,NDIM,ETA,EST,MAX,MODE,IPRINT)
IMPLICIT REAL*8 (A-H,O-Z)
REAL* KTB, IPRINT
COMMON /FAIL/, NFAG
DIMENSION X(NDIM), G1(10), G2(10), ETA(10), H(10), P
1(10), Y1(10), Y2(10), ETA(10), H(10), P
EXTERNAL FUNCT
R1=R0
IFLAG=0
M=0
N2=NDIM+1
N1=NDIM+2
NUMF=0
IER=0
DO 1 I=1,N1
X1(I)=X(I)
1 CONTINUE
CALL FUNCT (NDIM;X1,F,F1,G1,RK)
NUMF=NUMF+1
IF (IER.NE.0) GO TO 30
DO 2 I=1,NDIM
X2(I)=X1(I)
2 CONTINUE
CALL LINES (X2;H,R0,NDIM,F2,G2;NUMF,IER,EPS,EST,RK)
IF (IER.NE.1) RETURN
IF (IFLAG.EQ.0) GO TO 30
DO 4 I=1,N1
R1(I)=X2(I)
4 CONTINUE
ALFA(I)=X2(I)
5 CONTINUE
KTB=KTB+1
IF (IPRINT.EQ.0) GO TO 6
IF (MOD(KTB,IPRINT).NE.0) GO TO 6
CALL OUTP (X2,F2,NDIM,GG,NUMF,RK)
6 CONTINUE
P(I,J)=0.
IF (IPRINT.EQ.0) GO TO 10
IF (MOD(KTB,IPRINT).NE.0) GO TO 10
7 CONTINUE
PRINT*, 'GOT BY A1'
8 CONTINUE
PRINT*, 'GOT BY A1'
9 CONTINUE
PRINT*, 'GOT BY A1'
10 CONTINUE
PRINT*, 'GOT BY A1'
11 CONTINUE
Y(I)=G2(I)
12 CONTINUE
Y(I)=G2(I)
CC

PRINT*, 'GOT BY A1'
GG=0.
DO 22, I=1,NDIM
GG=GG+G2(I)*G2(I)
PRINT*, 'GOT BY AIS'
CONTINUE
GG=SQR(T(GG))
KOUNT=KOUNT+1
H=H+1
IF (IPRINT.EQ.0) GO TO 23
IF (MOD(H,TB*IPRINT).NE.0) GO TO 23
PRINT*, 'GOT BY GB'
CALL OUTP(X2,F2,H,NDIM,GG,NUMF,RR)
PRINT*, 'GOT BY H'
CONTINUE
KB=KB+1
IF (MODE.EQ.2) GO TO 25
PRINT*, 'GOT BY HA'
IF (H+1.GT.MAX) GO TO 30
PRINT*, 'GOT BY H8'
NSOL=0.
DO 24 I=1,NDIM
IF (ABS(RR(I)).GT.ETA(2)) NSOL=1
CONTINUE
PRINT*, 'GOT BY HC'
IF (NSOL.EQ.0) GO TO 26
PRINT*, 'GOT BY HE'
GO TO 29
PRINT*, 'GOT BY HE8'
IF ((GG,1,G2A,ID1,1),(1/G2A,GT.MAX)) GO TO 26.
PRINT*, 'GOT BY H8'
GO TO 29
CONTINUE
PRINT*, 'GOT BY H8'
IF (IPRINT.EQ.0) GO TO 27
PRINT*, 'GOT BY H9'
WRITE (6,43)
PRINT*, 'GOT BY I'
CALL OUTP(X2,F2,H,NDIM,GG,NUMF,RR)
PRINT*, 'GOT BY J'
CONTINUE
X(I)=X2(I)
CONTINUE
EST=F2
RETURN
CONTINUE
PRINT*, 'GOT BY JAC'
IF (KOUNT.LE.HA) GO TO 11
PRINT*, 'GOT BY JB'
GO TO 10
PRINT*, 'GOT BY JC'
CONTINUE
PRINT 34, IER
NFAIL+1
RETURN
KRB=KRBI+1
IF (KRB.GT.10) NFAIL=1
IF (NFAIL.EQ.10) RETURN
DO 32 I=1,NDIM
61(I)=G2(I)
32 CONTINUE
AMBDAA=1
IF (ALFA) .GT. 64
PRINT*, 'GOT BY B5'
ELSE
IF (ALFA-AMBDAA) .GT. 64
PRINT*, 'GOT BY B6'
AMBDAA=ALFA
ALFA=1
DO 8 I=1,N
X(I)=X(I)+AMBDAA*H(I)
PRINT*, 'GOT BY B7'
CONTINUE
FX-FY
DO 3 Y=1,N
PRINT*, 'GOT BY B8'
CALL FUNCT (N*X+F*G+RR)
PRINT*, 'GOT BY B9'
IF (NFAIL.EQ.1) RETURN
PRINT*, 'GOT BY B10'
NUMF=NUMF+1
IF (F.LT.FX) RETURN
PRINT*, 'GOT BY B11'
FY=FY
DO 9 J=1,N
DY=DY+G(J)*H(J)
PRINT*, 'GOT BY B12'
CONTINUE
9 CONTINUE
IF (DY) .GT. 30,13
PRINT*, 'GOT BY B13'
IF (FY) .GT. 11,13
PRINT*, 'GOT BY B14'
AMBDAA=AMBDAA+ALFA
ALFA=AMBDAA
(NMAX-AMBDAA+1.E10) 7,7,12
PRINT*, 'GOT BY B16'
IF (N) .EQ. 0
GO TO 31
PRINT*, 'GOT BY B17'
T=0.
10 IF (AMBDAA) .GT. 15,30,15
PRINT*, 'GOT BY B18'
F3=K*(FX-FY)/AMBDAA+DX+DY
ALFA=MAX1(ABS(Z),ABS(DX),ABS(DY))
DALFA=ALPHA/DALFA+DX/DALFA+DY/DALFA
IF (DALFA) .LE. 0.16
PRINT*, 'GOT BY B19'
W=ALFA*SORT(DALFA)
ALFA=ALFA/W
IF (ALFA) .GT. 1.01
PRINT*, 'GOT BY B20'
ALFA=(DY-Z1)/ALFA
GO TO 19
CONTINUE
CALL FUNCT (N*X+F*G+RR)
PRINT*, 'GOT BY B21'
IF (NFAIL.EQ.1) RETURN
NUMF=NUMF+1
IF (F.LT.FX) GO TO 50
CONTINUE
CONTINUE
CALL FUNCT (N*X+F*G+RR)
GO TO 20
IF (F-Y) < 21.21,22
IF (F-FY) < 30.30,22
DALFA=0.
DO 23 I=1,N
DALFA=DALFA+G(I)*H(I)
23 CONTINUE
IF (DALFA) > 24.27,27
IF (F-FX) > 28.29,27
IF (DX-DALFA) > 26.30,26
FX=F
DX=DALFA
T=ALFA
AMBD=ALFA
DO TO 13
IF (FY-F) > 29,28,29
IF (DY-DALFA) > 27,30,29
FY=F
DY=DALFA
AMBD=AMBD-ALFA
DO TO 13
AMBD=AMBD-ALFA
RETURN
31 CONTINUE
IF (DY. GE. 0.) IER=-2
IF (GNRM.LE.1.E-10) GO TO 32
IF (GNRM/GNRM.LE.EPS) IER=-3
32 CONTINUE
IF (DALFA.LT.0.) IER=-1
NFAL=I
WRITE(*,33)
FORMAT('THE PROGRAM HAS FAILED')
RETURN
33 END
SUBROUTINE FUNCT (N,AL,U,GRAD,RR)
IMPLICIT REAL*8 (A-H,O-Z)
THIS SUBROUTINE IS USED TO CALCULATE THE OPTIMIZATION AND THE
GRADIENT AT ANY GIVEN POINT FOR SUBROUTINE POP7
DIMENSION AL(*), GRAD(*), SUM(17), RR(*)
COMMON/HHELP/S(101),XX(16,101),C(8)*8
C.... ABOVE LINE CHANGED FROM TEXT
C31=2*N+1
ZERO=0.0
DO 1 I=1,N21
PRINT*, 'GOT BY CI'
1 CONTINUE
CONTINUE
DO 4 I=1,M
SZ=ZERO
DO 3 K=1,N
SZ=SZ+AL(K)*XX(K,I)
2 CONTINUE
IF (SZ.GT.74.) GO TO 9
PR=SZ.'GOT BY CZ'
SS=EXP(SZ)*S(I)
SUM(I)=SUM(I)+SS
DO 4 J=2,N2
  SUM(J)=SUM(J)+X(J-1,I)*S
  PRINT*, 'GOT BY C4'
  CONTINUE
4  C4

CONTINUE
5  C5

DO 6 I=2,N2
  SUM(I)=SUM(I)/SUM(1)
  PRINT*, 'GOT BY C5'
  CONTINUE
6  C6

DO 8 I=1,N
  RR(I)=(SUM(I+1)-C(I))/C(I)
  U=U+RR(I)*Rk(I)
  PRINT*, 'GOT BY C6'
  CONTINUE
7  C7

DO 8 K=1,N
  GRAD(K)=0.0
  DO 7 J=1,N
    GRAD(K)=GRAD(K)+(SUM(J+K)-SUM(J+1)*SUM(K+1))*RR(J)/C(J)
    PRINT*, 'GOT BY C7'
  CONTINUE
8  C8

PRINT*, 'GOT BY C8'
CONTINUE
9  C9

PRINT*, 'GOT BY C9'
RETURN
10  C10

AA=SQRT
ZERO=0.0
DO TO 2
PRINT*, 'GOT BY C11'
END

SUBROUTINE START (XMAX,XMIN,ALAMDA,KSTART,CC,NL,IPRINT,UMIN,MODE,M)
15  C15

IMPLICIT REAL*8 (A-H,O-Z)
20  C20

THIS SUBROUTINE IS USED TO FIND A REASONABLE STARTING POINT FOR 25  C25
SUBROUTINE MPDP

DIMENSION R(11)
26  C26

DIMENSION CC(*)
27  C27

COMMON/HELP/XX(16,101),C(8),M
30  C30

END
35  C35

DO 2 I=1,NL
  ALAMDA(I)=0.0
  CONTINUE
2  C2

RETURN
3  C3

CONTINUE
4  C4

DO 4 I=1,NL
  ALAMDA(I)=0.0
  CONTINUE
4  C4

CONTINUE
CONTINUE
DO 23 I=1,NP1
  W(I,1)=R(I)
CONTINUE
DO 24 J=1,NP1
  W(1,J)=W(1,J)*X(J)
  Y(J)=Y(J)+DELTA
Y(J)=C(J)*Y(J)
CONTINUE
CALL SOLVE (W,Y,XID,NP1,10)
GO TO 12
CONTINUE
N=2
ALAMBDA(2)=-5/CC(2)
ALAMBDA(1)=CC(1)/CC(2)
NFAIL=0
CONTINUE
ALAMBDA(N+1)=2.0
ALAMBDA(N+2)=0.0
C PRINT*, 'GOT BY A,'
CALL HPOP (ALAMBDA,N,ETA,UMIN,MAXFN,MODE,IPRINT)
C PRINT*, 'GOT BY B,'
IF (NPFAIL.EQ.1) RETURN
IF (N.EQ.NL) RETURN
ALAMBDA(N+1)=0.0
N=N+1
GO TO 27
END

SUBROUTINE SOLVE (A,X,XID,N,NA)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(NA,N), X(N)
D=0.
DATA DIV/6.93147181/
DO I=1,N
  AA=0.
  DO J=I,N
    AB=ABS(A(J,I))
    IF (AB.LE.AA) GO TO 1
    AA=AB
    CONTINUE
    D=D+ALGB(AB)
  GO TO 9
IF (I.EQ.N) GO TO 7
IF (K.EQ.I) GO TO 3
DO J=1,N
  AB=A(I,J)
  A(I,J)=A(K,J)
  A(K,J)=AB
  CONTINUE
  AB=X(I)
  X(I)=X(K)
  X(K)=AB
  I=I+1
GO TO 4
CONTINUE
DO 5 J=1,N
  AA=ABS(A(J,I))/A(I,I)
  A(J,I)=0.
  DO K=1,N
    A(J,K)=A(J,K)+AA*A(I,K)
  CONTINUE
SUBROUTINE SIMSON
IMPLICIT REAL*8 (A-H,O-Z)

THIS SUBROUTINE IS TO CALCULATE THE SIMPSON MULTIPLIERS

COMMON/HELP/S(101),XX(16,101)*C(8),M

ABOVE LINE CHANGED FROM TEXT

S(1)=1.
S(M)=1.
N=n-1
DO 1 I=2,N+1
S(I)=4.
CONTINUE
N=n-1
DO 2 I=3,N+2
S(I)=2.
CONTINUE
RETURN
END

SUBROUTINE MULTI (XMAX,XMIN,N)
IMPLICIT REAL*8 (A-H,O-Z)

THIS SUBROUTINE IS USED TO GENERATE THE X*S POWER FOR SUBROUTINE

FUNCT

COMMON/HELP/S(101),XX(16,101)*C(8),M

ABOVE LINE CHANGED FROM TEXT

DELTA=(XMAX-XMIN)/FLOAT(N-1)

DO 1 I=1,M
XX(I,I)=XMIN+FLOAT(I-1)*DELTA
N=N+1
DO 1 J=2,N
XX(I,J)=XX(J-1,1)*XX(1,1)
CONTINUE
RETURN
END

SUBROUTINE CONVER (CM,NL)
IMPLICIT REAL*8 (A-H,O-Z)

THIS SUBROUTINE IS TO CALCULATE THE MOMENTS ABOUT THE ORIGIN

DIMENSION CM(*)
COMMON HELP/S(101)*X(16101)-C(0)-

C ABOVE LINE CHANGED FROM TEXT

C(1)=CM(1)
IF (NL.EQ.1) RETURN
DO 2 J=1,N
C(J)=CM(J)-C(I)***J*(-1)**J
N=J-1
DO 1 K=1,N
C(J)=C(J)-(-1)**K*FACTO(J)/(FACTO(K)*FACTO(J-K))*C(1)**(K)*C(J-K)
CONTINUE
CONTINUE
RETURN
END

SUBROUTINE TRN1(X1MAX,X1MIN,J,X2MAX,X2MIN,NL)
IMPLICIT REAL*8 (A-H,O-Z)

DIMENSION C(I)
SCL=X1MAX-X1MIN)/(X2MAX-X2MIN)
C(I)=C(I)/SCL-X1MIN/SCL*X2MIN
IF (NL.EQ.1) RETURN
DO 1 I=2,N
C(I)=C(I)/SCL**(FLOAT(I))
CONTINUE
RETURN
END

SUBROUTINE TRN2(X1MAX,X1MIN,J,X2MAX,X2MIN,N)
IMPLICIT REAL*8 (A-H,O-Z)

THIS SUBROUTINE IS AN ALTERNATIVE TO TRN2 (BELOW)

CALCULATES THE LAGRANGIAN MULTIPLIERS FOR A DIFFERENT INTERVAL

DOUBLE PRECISION VERSION

DIMENSION X(I)

DX1MAX=X1MAX
DXMIN=X1MIN
DX2MAX=X2MAX
DX2MIN=X2MIN
NP1=N+1
DO 10 I=1,NP1
10 DX(I)=X(I)**NP1
DX(1)=DX(1)/DXMAX
A=DX2MIN-DX1MIN/S
DX(I)=DX(I)-ALOG(S)
DO 30 I=1,N
30 DX(I)=DX(I)+DX(I)**A**I
CONTINUE
IF (NL.EQ.1) GO TO 6
DO 2 J=3,N
DO 1 I=J,N
FAC=I
KK=I-J+2
DO 2 I=KK
ACFAC=DBLE(FLOAT(K))
CONTINUE
DX(J)=DX(J)+FAC/DBLE(FACTO(J-1))*A**(I-J+1)*DX(I+1)
30 CONTINUE
CONTINUE
DX(J)=DX(J)/S**(J-1)
SUBROUTINE TRN2 (XMAX, XMIN, X, X2MAX, X2MIN, N)
IMPLICIT REAL*8 (A-H, O-Z)

THIS SUBROUTINE IS USED TO CALCULATE THE LAGRANGIAN MULTIPLIERS
AT THE ORIGINAL LIMITS

DIMENSION X(1)
S=(XMAX-XMIN)/(X2MAX-X2MIN)
A=X2MIN-XMIN/S
X(1)=X(1)-ALOG(S)
DO 1*I=N
X(1)=X(1)+X(I+1)*A**I
CONTINUE
IF (N.EQ.1) GO TO 5
DO 5J=N+1
FAC=1.
DO 2K=2,N
FAC=FAC*FLOAT(K)
CONTINUE
X(J)=X(J)+FAC/FACT(J-1)**(I-J+1)**X(I+1)
CONTINUE
X(J)=X(J)/S**(J-1)
CONTINUE
X(N+1)=X(N+1)/S**N
RETURN
END

FUNCTION CDF (XMIN, XMAX, XP, AL, N)
IMPLICIT REAL*8 (A-H, O-Z)

THIS FUNCTION SUBROUTINE IS TO CALCULATE THE CUMULATIVE DISTRIBUTION FUNCTION AT A GIVEN POINT

INPUT
XMIN = LOWER BOUND
XMAX = UPPER BOUND
XP = SPECIFIED POINT
AL(I) = ARRAY OF PARAMETERS, DIMENSION N
N = NUMBER OF PARAMETERS

DO 3I=1,N
IF (XP.LE.XMIN) GO TO 3
IF (XP.GE.XMAX) GO TO 4
RANGE=XMAX-XMIN
SS=RANGE/51.
JSS=SS
JSS=JSS/2**4
JSM1=JSS-1
DELTA=RANGE/FLOAT(JSM1)
DO 1I=1,JSM1+2
X=XMIN+FLOAT(I-1)*DELTA
AREA=AREA+4.*ENTRPF(AL, N, X)
1 CONTINUE
GOTO 5
3 CONTINUE
4 CONTINUE
5 RETURN
END
FUNCTION ENTRPF (AL,NPL,X)
IMPLICIT REAL*8 (A-H,O-Z)

FUNCTION TO EVALUATE THE ENTROPY DENSITY FUNCTION AT A GIVEN POINT

INPUT
AL(I) = ARRAY CONTAINING PARAMETERS, DIMENSION NPL
NPL = NUMBER OF PARAMETERS
X = GIVEN VALUE

DIMENSION AL(*)
S=AL(I)
DO 1 I=2,NPL
S=S+AL(I)**(I-1)
CONTINUE
END

FUNCTION FACTO (M)
IMPLICIT REAL*8 (A-H,O-Z)

C. CALLS FACTORIAL OF M
FACTO=1.
IF (M.EQ.0) RETURN
DO 1 I=1,M
FACTO=FACTO*FLOAT(I)
1 CONTINUE
RETURN
END
<table>
<thead>
<tr>
<th>Normal T</th>
<th>Po</th>
<th>1.40</th>
</tr>
</thead>
<tbody>
<tr>
<td>.399E+03</td>
<td>.995E+03</td>
<td>.855E+03</td>
</tr>
<tr>
<td>.607E+03</td>
<td>.204E+03</td>
<td>.947E+03</td>
</tr>
<tr>
<td>.912E+03</td>
<td>.961E+03</td>
<td>.961E+03</td>
</tr>
<tr>
<td>.204E+03</td>
<td>.204E+03</td>
<td>.204E+03</td>
</tr>
<tr>
<td>.197E+03</td>
<td>.202E+03</td>
<td>.202E+03</td>
</tr>
</tbody>
</table>

Log of Cycles to Reach Mean Fatigue Str =

<table>
<thead>
<tr>
<th>50 MPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>.432E+01</td>
</tr>
<tr>
<td>.797E+01</td>
</tr>
<tr>
<td>.703E+01</td>
</tr>
<tr>
<td>.917E+01</td>
</tr>
<tr>
<td>.494E+01</td>
</tr>
<tr>
<td>.672E+01</td>
</tr>
<tr>
<td>.642E+01</td>
</tr>
</tbody>
</table>

Sorted Log of Cycles

<table>
<thead>
<tr>
<th>50 MPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>.602E+01</td>
</tr>
<tr>
<td>.604E+01</td>
</tr>
<tr>
<td>.604E+01</td>
</tr>
<tr>
<td>.604E+01</td>
</tr>
<tr>
<td>.604E+01</td>
</tr>
<tr>
<td>.604E+01</td>
</tr>
</tbody>
</table>

Sample Moments

<table>
<thead>
<tr>
<th>50 MPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>.733E+01</td>
</tr>
<tr>
<td>.178E+00</td>
</tr>
<tr>
<td>.762E+00</td>
</tr>
</tbody>
</table>

| BMD(1), BMD(2) = | .572E+01 | .963E+01 |
**INPUT DATA FOR SUBROUTINE** MEPI

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT DATA IS PRINTED OUT FOR KDATA = 1 ONLY</td>
<td>KDATA = 1</td>
</tr>
<tr>
<td>INTERMEDIATE OUTPUT EVERY KPRINT(TH) CYCLE</td>
<td>KPRINT = 1</td>
</tr>
<tr>
<td>NUMBER OF KNOWN FIRST MOMENTS</td>
<td>N = 4</td>
</tr>
<tr>
<td>HIGHER LIMIT</td>
<td>XMAX = 0.963779301E+01</td>
</tr>
<tr>
<td>LOWER LIMIT</td>
<td>XMIN = 0.572349819E+01</td>
</tr>
<tr>
<td>FIRST MOMENTS</td>
<td>CC(1) = 0.735481628E+01</td>
</tr>
<tr>
<td>THE ALLOWED TOLERANCE IN LAGRANGIAN EQUATIONS</td>
<td>TOL = 0.100000000E-05</td>
</tr>
<tr>
<td>THE CUMULATIVE DISTRIBUTION REQUIRED AT NXP POINTS</td>
<td>NXP = 0</td>
</tr>
</tbody>
</table>
File /DBAO:/PLOT1.CPR/1 (359,204,0), last revised on 23-NOV-1988 11:20, is a 2 block sequential file owned by UIC (11.11). The records are variable length with FORTRAN (FTP) carriage control. The longest record is 25 bytes.

Job PLOT1 (483) queued to SYS$BPRT on 23-NOV-1988 11:20 by user NETNONPRIV, UIC (11.11), under account 20100ADD at priority 100.
DUE TO FOCUS ON
OF POOR QUALITY
9.0 APPENDIX D

IMSL SUBROUTINE CALLS FROM RANDOM3 AND RANDOM4

RANDOM3

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. RNNLNL - 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 the ordinates of the density.

RANDOM4

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. RNNLNL - Generates pseudorandom numbers from a lognormal distribution.
APPENDIX E

SAMPLE SAS/GRAPH PROGRAM FOR RANDOM3 AND RANDOM4

data a;
INFILE 'PLOT1.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 'PLOT2.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;