An Improved Approach for Flight Readiness Certification—Probabilistic Models for Flaw Propagation and Turbine Blade Failure

Volume II: Software Documentation

N. R. Moore
D. H. Ebbeler
L. E. Newlin
S. Sutharshana
M. Creager

December 30, 1992

NASA
National Aeronautics and Space Administration
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California
The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.
Preface

This report presents a methodology for managing failure risk cost-effectively and evaluating flight readiness of such aerospace systems as launch vehicles and planetary spacecraft. The methodology was developed by the Jet Propulsion Laboratory (JPL) under NASA RTOP 553-02-01 sponsored by the Office of Space Flight (OSF), NASA Headquarters. This work was performed as a part of the Certification Process Assessment task initiated by OSF due to concern about criteria for certifying flight readiness of the Space Shuttle propulsion system. The methodology is not only applicable to flight readiness evaluation, but also to design definition and to the identification of risk control measures during the design, development, or operational phases of a project.

An early phase of this work included an extensive review of certification and failure risk assessment approaches used by the aerospace industry and government agencies. Based on the findings of this review, further work was focused on defining, developing, and demonstrating an improved technical approach for failure risk assessment that can incorporate information from both test experience and analytical modeling to obtain a quantitative failure risk estimate. This approach, called Probabilistic Failure Assessment (PFA), is of particular value when information relevant to failure prediction, including test experience and knowledge of parameters used in analytical modeling of failure phenomena, is expensive or difficult to acquire. Under such constraints, a quantitative evaluation of failure risk based on the information available from both analytical modeling and operating experience is needed to make effective risk management decisions that utilize financial resources efficiently.

The PFA methodology is applicable to failure modes that can be characterized by analytical or empirical modeling of failure phenomena, including those of structural, electro-optical, propulsion, power, and thermal control systems, and is especially useful when models or information used in analysis are uncertain or approximate. PFA can be applied at any time in the design, development, or operational phases of a program to quantitatively estimate failure risk based on the information available at the time of the risk assessment and can be used to evaluate and rank alternative measures to control risk, thereby enabling the more effective allocation of limited financial resources.

1 See [14] of Section 1.0 references.
The work documented in this report was carried out by a multidisciplinary team of JPL technical personnel, which was managed by N. R. Moore. This team was composed of individuals with expertise in statistics, systems modeling, and engineering analysis. D. H. Ebbeler formulated and structured the statistical methodology and directed its implementation. L. E. Newlin formulated and implemented probabilistic engineering models and implemented the statistical methodology. S. Sutharshana formulated and implemented probabilistic analytical methods and models. M. Creager\(^2\) made major contributions to defining and formulating the probabilistic modeling approach and analytical modeling procedures used in this work. D. Goode typeset the manuscript, including graphics, using computerized desktop publishing methods, and E. Reinig edited the manuscript.

In developing the PFA methodology, the JPL team interacted with aerospace system manufacturers, the Marshall Space Flight Center, and the Lewis Research Center. Individuals of these organizations generously shared information and spent significant amounts of time with the JPL team. In particular, Rocketdyne, Canoga Park, California; Aerojet TechSystems, Nimbus, California; and Pratt & Whitney, West Palm Beach, Florida, collaborated in performing the application examples given herein. In addition, technical comments on certification approaches and failure modeling were provided by personnel from the above-listed organizations and General Electric, Cincinnati, Ohio; the Federal Aviation Administration; and the Wright-Patterson Air Force Base.

The PFA methodology, examples of its application to spaceflight components, and computer software used to implement PFA are documented in the two volumes of this report. Volume I documents the PFA methodology and the application examples, including the rationale for PFA and the analysis procedures used in the examples. Volume II contains detailed documentation of the computer software used to implement PFA for the application examples, including user's guides, code execution examples, flowcharts, and listings of the computer programs.

\(^2\) Currently of Structural Integrity Engineering, Chatsworth, CA.
Acknowledgments

The authors wish to acknowledge the guidance and encouragement provided by D. L. Winterhalter of the Office of Space Flight, NASA Headquarters, whose support made this work possible. The authors appreciate the encouragement of R. Bardos, J. Mulcahy, and other individuals of OSF. The review and comments on this work provided by R. Weinstock of Vitro, Washington, D.C., contributed substantially to its successful completion.

The application examples of this report were performed in collaboration with Rocketdyne, Canoga Park, California; Aerojet TechSystems, Nimbus, California; and Pratt & Whitney, West Palm Beach, Florida. Several individuals at each organization contributed generously to this work, including E. P. Fox, C. G. Annis, and D. Paulus of Pratt & Whitney; K. J. O'Hara, D. O'Connor, K. J. Chang, and D. Russell of Rocketdyne; and B. Boehm of Aerojet. The authors worked particularly closely with E. P. Fox of Pratt & Whitney and K. J. O'Hara of Rocketdyne; their considerable contributions are gratefully acknowledged. The contributions of C. G. Annis, D. Russell, and K. J. Chang to the crack growth analysis are also gratefully acknowledged.

Present or former personnel of the NASA Marshall Space Flight Center, whose cooperation was instrumental in successfully performing this work, include J. Lombardo, J. S. Richards, G. W. Smith, L. D. Salter, H. P. Stinson, and J. Townsend.

Throughout the course of this work constructive guidance was provided by the Liquid Rocket Engine Certification Subcommittee of Aerospace Division Committee G-11, Society of Automotive Engineers. The membership of this subcommittee included: W. E. Campbell, Aerojet; K. J. O'Hara, Rocketdyne; E. P. Fox, Pratt & Whitney; J. S. Richards and H. P. Stinson, NASA-MSFC; R. L. Doebler, Aerospace Corp.; and N. R. Moore, JPL.

Finally, the authors wish to acknowledge the review of the technical approach of this work provided by the late R. P. Feynman of the California Institute of Technology.

The authors express their gratitude to all those individuals who contributed to this work and regret that a complete listing is not feasible.
Abstract

An improved methodology for quantitatively evaluating failure risk of spaceflight systems to assess flight readiness and identify risk control measures is presented. This methodology, called Probabilistic Failure Assessment (PFA), combines operating experience from tests and flights with analytical modeling of failure phenomena to estimate failure risk. The PFA methodology is of particular value when information on which to base an assessment of failure risk, including test experience and knowledge of parameters used in analytical modeling, is expensive or difficult to acquire.

The PFA methodology is a prescribed statistical structure in which analytical models that characterize failure phenomena are used conjointly with uncertainties about analysis parameters and/or modeling accuracy to estimate failure probability distributions for specific failure modes. These distributions can then be modified, by means of statistical procedures of the PFA methodology, to reflect any test or flight experience. State-of-the-art analytical models currently employed for design, failure prediction, or performance analysis are used in this methodology.

The PFA methodology can be applied at any time in the design, development, or operational phases of a program to quantitatively estimate failure risk based on the information available at the time failure risk is assessed. Sensitivity analyses conducted as a part of PFA can be used to evaluate and rank such alternative measures to control risk as design changes, testing, or inspections, thereby enabling limited program resources to be allocated more effectively.

PFA is generally applicable to failure modes that can be characterized by analytical or empirical models of failure phenomena and is especially valuable when models or information used in analysis are uncertain or approximate. Such failure modes include, but are not limited to, fatigue, flaw propagation, erosion, malfunctions of mechanical or electrical systems, and shortfalls with respect to performance or life goals for thermal control, electro-optical, power, or propulsion systems.

It is often not feasible to acquire enough test experience to establish high reliability at high confidence for spaceflight systems. Moreover, the results of conventionally performed analytical modeling of failure modes can be subject to serious misinterpretation when uncertain or approximate information is used to establish analysis parameters and calibrate the accuracy of analysis models. Under these conditions, a quantitative evaluation of failure risk based on the information...
available from both test or flight experience and analytical modeling is needed to make effective risk management decisions.

This report discusses the rationale for the statistical approach taken in the PFA methodology, describes the PFA methodology, and presents examples of its application to structural failure modes. The engineering models and computer software used in fatigue crack growth and fatigue crack initiation applications are thoroughly documented.
3.1 Introduction .................................................. 3-3

3.2 Turbine Blade LCF Analysis .................................... 3-7
  3.2.1 Component Description .................................. 3-7
  3.2.2 Modeling Approach ...................................... 3-7
  3.2.3 Mission Strain History for the Blade ................... 3-7
  3.2.4 Driver Transformation .................................. 3-9
  3.2.5 Mean Strain Effects .................................... 3-13
  3.2.6 Damage Calculations ................................... 3-14
  3.2.7 Alternative Characterizations of the ATD Blade LCF Materials Model 3-14
    3.2.7.1 Parameter Uncertainty .............................. 3-14
    3.2.7.2 A Procedure for Bootstrapping the Impact of Limited Stress/Life Data 3-16
    3.2.7.3 Spatial Symmetry Effects ......................... 3-20
  3.2.8 Modeling Multiple Critical Locations .................... 3-21
  3.2.9 Probabilistic Failure Model Implementation ............ 3-22

3.3 Turbine Blade Case Study ..................................... 3-27
  3.3.1 Driver Description ..................................... 3-27
  3.3.2 Materials Characterization ............................. 3-30
  3.3.3 Analysis ............................................. 3-31
  3.3.4 Results .............................................. 3-31

3.4 Analysis Procedure ......................................... 3-35
  3.4.1 Selecting the Component, Failure Mode, and Critical Location 3-35
  3.4.2 Preliminary Deterministic Analysis ........................ 3-35
  3.4.3 Driver Characterization ................................ 3-36
  3.4.4 Materials Characterization ................................ 3-36
  3.4.5 Time History Definition ................................ 3-38
  3.4.6 Probability of Failure Curve Parameter Estimation .......... 3-38
  3.4.7 Driver Sensitivity Analysis ............................. 3-40
  3.4.8 Probability of Failure Curve Standardization ............ 3-41

References ................................................................ 3-42

Appendix 3.A List of Symbols ...................................... 3-43

Appendix 3.B Input And Output Files ............................ 3-47
  Input File - BLDLDC .......................................... 3-47
  Output File - BLDLCO ........................................... 3-48
  Output File - LOWLIF ........................................... 3-52
  Output File - DUMP ............................................ 3-56

4.0 Probabilistic Modeling of Turbine Blade High Cycle Fatigue Failure ....... 4-1
  4.1 Introduction ................................................... 4-3
4.2 Turbine Blade HCF Methodology ........................................ 4-5
  4.2.1 Component Description .............................................. 4-5
  4.2.2 Modeling Approach ................................................... 4-5
  4.2.3 Driver Transformation ............................................... 4-6
  4.2.4 Mean Stress Effects .................................................. 4-7
  4.2.5 Modeling Multiple Critical Locations .............................. 4-10
  4.2.6 Probabilistic Failure Model Implementation ...................... 4-10

References ................................................................. 4-14

Appendix A List of Symbols .............................................. 4-15
7.3.4 Program BLDHCF Listing ........................................ 7-269
  Program BLDHCF Listing Temporal Order, Uniform
  Distribution .................................................. 7-270
  Program BLDHCF Listing Temporal Order, Truncated
  Normal Distribution ................................. 7-271

Reference ......................................................... 7-341
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2-1</td>
<td>Information Sources for Failure Risk Assessment</td>
<td>1-7</td>
</tr>
<tr>
<td>1.2-2</td>
<td>The Probabilistic Failure Assessment Methodology</td>
<td>1-8</td>
</tr>
<tr>
<td>1.2-3</td>
<td>The Probabilistic Failure Modeling Procedure</td>
<td>1-10</td>
</tr>
<tr>
<td>1.3-1</td>
<td>Integration of PFA into the Design and Development Process</td>
<td>1-15</td>
</tr>
<tr>
<td>2.1-1</td>
<td>Crack Growth Failure Modeling Approach</td>
<td>2-3</td>
</tr>
<tr>
<td>2.2-1</td>
<td>Crack Growth Failure Simulation Structure</td>
<td>2-6</td>
</tr>
<tr>
<td>2.2-2</td>
<td>Geometry of Duct</td>
<td>2-8</td>
</tr>
<tr>
<td>2.2-3</td>
<td>Description of the Stochastic Crack Growth Equation in Log–Log Space</td>
<td>2-12</td>
</tr>
<tr>
<td>2.2-4</td>
<td>Flowchart for the Crack Growth Calculations</td>
<td>2-16</td>
</tr>
<tr>
<td>2.3-1</td>
<td>HPOTP Heat Exchanger</td>
<td>2-17</td>
</tr>
<tr>
<td>2.3-2</td>
<td>Detail of the HPOTP Heat Exchanger Coil Near Weld 3</td>
<td>2-18</td>
</tr>
<tr>
<td>2.3-3</td>
<td>Schematic of Approximating the SIF Solution for Duct with a Finite Width Plate Solution</td>
<td>2-19</td>
</tr>
<tr>
<td>2.3-4</td>
<td>The Mean Crack Growth Rate Curves for Welded 316L Data</td>
<td>2-19</td>
</tr>
<tr>
<td>2.3-5</td>
<td>Failure Life Distributions and Driver Sensitivities for HEX Coll</td>
<td>2-24</td>
</tr>
<tr>
<td>2.3-6</td>
<td>Impact of Initial Crack Size $a_i$ on Failure Life Distributions for HEX Coll</td>
<td>2-24</td>
</tr>
<tr>
<td>2.3-7</td>
<td>Effect of Initial Crack Size and Crack Growth Threshold on B.1 Life for HEX Coll</td>
<td>2-25</td>
</tr>
<tr>
<td>2.4-1</td>
<td>Proposed External Heat Exchanger Block II Design</td>
<td>2-28</td>
</tr>
<tr>
<td>2.4-2</td>
<td>Arrangement of the Channels in the EXHEX</td>
<td>2-28</td>
</tr>
<tr>
<td>2.4-3</td>
<td>Crack Configuration Used for EXHEX Channel</td>
<td>2-29</td>
</tr>
<tr>
<td>2.4-4</td>
<td>Crack Growth Data for C10100 Copper and Mean Curve with $\lambda_{Kp} = 0$</td>
<td>2-29</td>
</tr>
<tr>
<td>2.4-5</td>
<td>EXHEX Failure Life Distribution and Driver Sensitivities</td>
<td>2-32</td>
</tr>
<tr>
<td>2.5-1</td>
<td>Overall Procedure for Crack Growth Failure Mode</td>
<td>2-34</td>
</tr>
<tr>
<td>2.5-2</td>
<td>Steps of the Probability of Failure Curve Parameter Estimation</td>
<td>2-48</td>
</tr>
<tr>
<td>3.1-1</td>
<td>Low Cycle Fatigue Failure Modeling Approach</td>
<td>3-3</td>
</tr>
<tr>
<td>3.1-2</td>
<td>Calculation Procedure Used in the Low Cycle Fatigue Model</td>
<td>3-4</td>
</tr>
<tr>
<td>3.2-1</td>
<td>Axial Cross Section of the ATD-HPFTP Turbine Showing the Monolithic Disk and Both Stages of Blades</td>
<td>3-8</td>
</tr>
<tr>
<td>3.2-2</td>
<td>ATD-HPFTP First Stage Turbine Blade</td>
<td>3-8</td>
</tr>
<tr>
<td>3.2-3</td>
<td>Illustration of the Strain-Time History for the ATD Blade</td>
<td>3-9</td>
</tr>
<tr>
<td>3.2-4</td>
<td>Schematic of Temperature Profile During Shutdown</td>
<td>3-10</td>
</tr>
<tr>
<td>3.2-5</td>
<td>The Simulation Structure for the Parametric Representation of Parameter Uncertainty</td>
<td>3-17</td>
</tr>
<tr>
<td>3.2-6</td>
<td>The Simulation Structure for the Bootstrapping Representation of Parameter Uncertainty</td>
<td>3-18</td>
</tr>
<tr>
<td>3.2-7</td>
<td>Plot of Eight Specimen Failure Points</td>
<td>3-19</td>
</tr>
<tr>
<td>3.2-8</td>
<td>Structure of the Probabilistic Failure Model</td>
<td>3-23</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.1-4</td>
<td>Detail of the HPOTP Heat Exchanger Coll Small Tube Outlet Near Weld 3</td>
<td>6-32</td>
</tr>
<tr>
<td>6.2-1</td>
<td>Format for File BDLCD</td>
<td>6-49</td>
</tr>
<tr>
<td>6.2-2</td>
<td>Format for File RELATD</td>
<td>6-52</td>
</tr>
<tr>
<td>6.2-3</td>
<td>Data Blocks for Input File</td>
<td>6-52</td>
</tr>
<tr>
<td>7.1-1</td>
<td>Tree Structure for Program PROCRK</td>
<td>7-4</td>
</tr>
<tr>
<td>7.2-1</td>
<td>Tree Structure for Program BDLDCF for the Uniform Variation in Materials</td>
<td>7-78</td>
</tr>
<tr>
<td></td>
<td>Shape Parameter $m$</td>
<td></td>
</tr>
<tr>
<td>7.2-2</td>
<td>Tree Structure for Program BDLDCF for the Truncated Normal Variation in</td>
<td>7-79</td>
</tr>
<tr>
<td></td>
<td>Materials Shape Parameter $m$</td>
<td></td>
</tr>
<tr>
<td>7.2-3</td>
<td>Tree Structure for Program BDLDCF V3.4B1.3 for the Bootstrapping of the</td>
<td>7-80</td>
</tr>
<tr>
<td></td>
<td>Materials Shape Parameter $m$</td>
<td></td>
</tr>
<tr>
<td>7.3-1</td>
<td>Tree Structure for Program BLDHCF for the Uniform Variation in Materials</td>
<td>7-256</td>
</tr>
<tr>
<td></td>
<td>Shape Parameter $m$</td>
<td></td>
</tr>
<tr>
<td>7.3-2</td>
<td>Tree Structure for Program BLDHCF for the Truncated Normal Variation in</td>
<td>7-257</td>
</tr>
<tr>
<td></td>
<td>Materials Shape Parameter $m$</td>
<td></td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4-1</td>
<td>Index of Topics Contained in the Report</td>
<td>1-18</td>
</tr>
<tr>
<td>1.4-2</td>
<td>Index of Software Documentation Contained in the Report</td>
<td>1-19</td>
</tr>
<tr>
<td>2.3-1</td>
<td>Description of Drivers Used in the HEX Coil Analysis</td>
<td>2-21</td>
</tr>
<tr>
<td>2.4-1</td>
<td>Driver Distributions for EXHEX</td>
<td>2-30</td>
</tr>
<tr>
<td>2.B-1</td>
<td>HPOTP Heat Exchanger Coil Beam-End Forces Near Weld 3</td>
<td>2-43</td>
</tr>
<tr>
<td>2.B-2</td>
<td>Scanning Circumference for Critical Angle Causing Minimum Life</td>
<td>2-45</td>
</tr>
<tr>
<td>2.B-3</td>
<td>Lives for Different Random Number Seeds and History Lengths</td>
<td>2-47</td>
</tr>
<tr>
<td>2.B-4</td>
<td>Probability of Failure Curve Parameter Estimates for Different Initial Crack Sizes</td>
<td>2-50</td>
</tr>
<tr>
<td>2.B-5</td>
<td>Driver Sensitivity Analysis for 0.005 in. Initial Crack Size</td>
<td>2-51</td>
</tr>
<tr>
<td>3.2-1</td>
<td>Nominal History for the ATD-HPFTP First Stage Turbine Blade</td>
<td>3-13</td>
</tr>
<tr>
<td>3.2-2</td>
<td>Driver Distributions and Influences for ATD-HPFTP First Stage Turbine Blade LCF</td>
<td>3-25</td>
</tr>
<tr>
<td>3.3-1</td>
<td>Driver Distributions for ATD-HPFTP First Stage Turbine Blade LCF</td>
<td>3-28</td>
</tr>
<tr>
<td>3.3-2</td>
<td>B.1 Life (missions) at 95% Assurance for All Driver Variation</td>
<td>3-34</td>
</tr>
<tr>
<td>3.4-1</td>
<td>Matrix of Thermal Strains as a Function of Gas Film Coefficient and Gas Temperature During Acceleration</td>
<td>3-36</td>
</tr>
<tr>
<td>3.4-2</td>
<td>Values of Thermal Strain for Varying Values of Temperature and Slope During Deceleration</td>
<td>3-36</td>
</tr>
<tr>
<td>3.4-3</td>
<td>PWA 1480 S/N Data</td>
<td>3-37</td>
</tr>
<tr>
<td>3.4-4</td>
<td>Summary of Materials Characterization Study of PWA 1480 Data</td>
<td>3-37</td>
</tr>
<tr>
<td>3.4-5</td>
<td>Driver Sensitivity Analysis for the Turbine Blade for Weibull Materials Variation</td>
<td>3-41</td>
</tr>
<tr>
<td>7.1-1</td>
<td>List of Subprograms For Program PROCRK</td>
<td>7-5</td>
</tr>
<tr>
<td>7.1-2</td>
<td>List of Variables For Program PROCRK</td>
<td>7-7</td>
</tr>
<tr>
<td>7.2-1</td>
<td>List of Subprograms For Program BLDLCF</td>
<td>7-81</td>
</tr>
<tr>
<td>7.2-2</td>
<td>List of Variables For Program BLDLCF</td>
<td>7-85</td>
</tr>
<tr>
<td>7.3-1</td>
<td>List of Subprograms For Program BLDHCF</td>
<td>7-258</td>
</tr>
<tr>
<td>7.3-2</td>
<td>List of Variables For Program BLDHCF</td>
<td>7-261</td>
</tr>
</tbody>
</table>
5.0 Analysis Software
Section 5.1
Crack Growth Analysis Software

5.1.1 Introduction
This section presents a description of the computer program PROCRK which implements the crack growth analysis discussed in Section 2. The code PROCRK was used to analyze the HPOTP Heat Exchanger (HEX) Coil and the proposed External Heat Exchanger (EXHEX). The program PROCRK is modular and hence can be easily modified for crack growth analysis of different components. Different modules were provided for stress analysis and stress intensity factor calculations for the HEX coil and EXHEX analyses. The overall layout of the program is described by using a main flowchart that refers to other flowcharts which describe subprograms and key portions of the main program in greater detail. The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the crack growth analysis code PROCRK are given in Section 7.1. The relevant user's guide for running this code is given in Section 6.1. A glossary of standard flowchart symbols is given in Appendix 5.A.

5.1.2 PROCRK Program
The crack growth methodology is implemented as the FORTRAN program PROCRK. This section provides the description and flowcharts for program PROCRK.

5.1.2.1 Main Routine
The master flowchart for the PROCRK program is given in Figure 5.1-1. The program starts by opening the following input and output files:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRKDAT</td>
<td>Input</td>
<td>Analysis data</td>
</tr>
<tr>
<td>CRKRES</td>
<td>Output</td>
<td>Input data echo, results</td>
</tr>
<tr>
<td>IOUTPR</td>
<td>Output</td>
<td>Run information and user-requested information</td>
</tr>
<tr>
<td>LOWLIF</td>
<td>Output</td>
<td>First one percent of sorted crack growth lives</td>
</tr>
</tbody>
</table>

The arrays and variables are set to their default or initial values in the SETDEF routine described in Section 5.1.2.2. The input data is then read from the CRKDAT file in the INPUT routine described in Section 5.1.2.3 and an echo of the input data is written onto the CRKRES file. The materials data including the da/dN vs. ΔK crack growth data is read and processed in the GRODAT routine described below in Section 5.1.2.4.
Figure 5.1-1  Main Flowchart for Crack Growth Analysis Program PROCRK

START

SETDEF
INITIALIZE ARRAYS AND SET DEFAULTS
(See Section 5.1.2.2)

INPUT
READ AND ECHO INPUT DATA
(See Section 5.1.2.3)

GRODAT
READ MATERIAL DATA AND PERFORM
REGRESSION TO DERIVE MEAN CURVE
(See Section 5.1.2.4)

DO K ← 1
TO NHYPER BY 1
OUTER LOOP

HYPDRW
SELECT HYPERPARAMETERS FOR DRIVERS
(See Section 5.1.2.5)

DO I ← 1
TO NLIFE BY 1
INNER LOOP

PARDRW
SELECT DRIVER VALUES
(See Section 5.1.2.6)

LIFCAL
PERFORM DRIVER TRANSFORMATION AND
CALCULATE CRACK GROWTH
(See Section 5.1.2.7)

INSORT
SAVE AND SORT LOWEST ONE PERCENT OF LIVES
(See [1], Appendix 5.B)

WRITE RESULTS TO:
CRKRES, LOWLIF

STOP
The selection of hyperparameters\(^1\) is performed in the outer DO loop of the simulation by calling the HYPDRW routine described in Section 5.1.2.5. The driver draws are performed within the inner DO loop of the simulation by calling the PARDRW routine described in Section 5.1.2.6. The routine LIFCAL performs driver transformation and calculates the crack growth life within the inner DO loop. The LIFCAL routine is described below in Section 5.1.2.7.

The crack growth lives are arranged in ascending order in a list containing the lowest one percent of the lives. The INSORT routine performs an insertion sort with each new life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIF. Routine INSORT is described in Appendix 5.B of [1].

5.1.2.2 SETDEF Routine

The arrays and variables are set to their default or initial values in this routine. Most of the arrays and variables are initialized to zero. The array LIFE( ), which is used to store and sort the lowest one percent of the crack growth lives, is initialized to a large value and the number of crack lengths NCRL used for block growth calculations is initialized to fifty. Also, the logical variable FAIL which flags unstable crack growth (i.e., when \( K > K_c \)) is initialized to 'FALSE'.

5.1.2.3 INPUT Routine

The input data is read from the CRKDAT file in this routine. First the analysis control variables including the simulation size are read and echoed in the IOUTPR file. Then, the driver distribution information is read. Next, the load/stress history information is read. Finally, some miscellaneous information, such as the Willenborg retardation model parameter, is read. An echo of the input data is written onto the CRKRES file.

5.1.2.4 GRODAT Routine

The flowchart for the GRODAT routine is given in Figure 5.1-2. First the \( da/dN \) vs. \( \Delta K \) crack growth data and material properties, such as fracture toughness, threshold stress intensity factor (SIF) range, and tensile strength, are read from the CRKDAT file. Then regression of the crack growth data is performed to fit the generalized Forman Equation 2-7. Four options are available to derive the equation parameters \( C, n, m, p, \) and \( q \), as follows:

\(^1\) Hyperparameters are discussed in Section 2.1.1 of [1].
READ CRACK GROWTH RATE DATA

IREGOP = 1

FALSE

FIX p REGRESS FOR C, m, n, q

IREGOP = 2

FALSE

FIX m, p REGRESS FOR C, n, q

IREGOP = 3

FALSE

FIX m, p, q REGRESS FOR C, n,

IREGOP = 4

FALSE

REGRESS FOR C, m, n, p, q

RETURN

Figure 5.1-2  Flowchart for Subprogram GRODAT
<table>
<thead>
<tr>
<th>OPTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fix ( p ) regress for ( C, m, n, q )</td>
</tr>
<tr>
<td>2</td>
<td>Fix ( m, p ) regress for ( C, n, q )</td>
</tr>
<tr>
<td>3</td>
<td>Fix ( m, p, q ) regress for ( C, n )</td>
</tr>
<tr>
<td>4</td>
<td>Regress for ( C, m, n, p, q )</td>
</tr>
</tbody>
</table>

An external function DETER4 is employed to calculate the determinant of a 4x4 matrix for the \( \text{IREGOP} = 4 \) case.

### 5.1.2.5 HYPDRW Routine
The selection of hyperparameters is performed in the outer DO loop of the simulation by calling the HYPDRW routine. This includes calling the RANDOM and PRYRV subroutines to obtain the \( \rho \) and \( \theta \) parameters for drivers with Beta distributions, and \( \mu \) and \( \sigma \) parameters for drivers with Normal distributions.

### 5.1.2.6 PARDRW Routine
The driver draws are performed within the inner DO loop of the simulation by calling the PARDRW routine. Drivers are selected by calling BETAGN, NORMGN, and PRYRV, which draw from Beta, Normal, and Uniform distributions, respectively. The general-purpose probability distribution subroutines RANDOM, BETAGN, NORMGN, and PRYRV are described in Sections 4.4 and 7.6 of [1].

### 5.1.2.7 LIFCAL Routine
The flowchart for the LIFCAL routine is given in Figure 5.1-3. First, the stress history is derived in one of the following routines.

<table>
<thead>
<tr>
<th>ROUTINE</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRAN1</td>
<td>HEX coil stress calculation</td>
</tr>
<tr>
<td>STRAN2</td>
<td>EXHEX stress calculation</td>
</tr>
</tbody>
</table>

STRAN1 and STRAN2 routines are described in Section 5.1.2.8. A rainflow cycle count is performed and a stress-level vs. number-of-cycles table is generated in the CYCOUN routine described in Section 5.1.2.9. The life integration DO loop calculates block growth rates at \( \text{NCRL} \) number of crack lengths by calling the BLKGRO routine described in Section 5.1.2.10.

### 5.1.2.8 STRAN1 and STRAN2 Routines
The flowchart for the STRAN1 routine is given in Figure 5.1-4. The maximum principal stress was assumed to be the axial stress component for the HEX coil. The composite principal stress history, which is due to static, random, sinusoidal, and aerodynamic loads, is derived in this routine. First, the static stresses are assigned to the stress history. Then, the reference time histories for each load component are scaled by the non-time-varying dynamic stress magnitudes and added.
START

KPROB = 1

VALUE OF KPROB

KPROB = 2

STRAN1
STRESS CALCULATION FOR HEX COIL
(See Section 5.1.2.8)

STRAN2
STRESS CALCULATION FOR EXHEX
(See Section 5.1.2.8)

CYCOUN
PERFORM RAINFLOW CYCLE COUNTING
(See Section 5.1.2.9)

DO J ← 1
TO NCRL BY 1
LOOP FOR EVERY CRACK DIVISION

BLKGRO
CALCULATE BLOCK GROWTH RATES \( da/dB \)
(See Section 5.1.2.10)

CALCULATE TOTAL LIFE BASED ON BLOCK GROWTH RATES
\[
L = \lambda_{gro} T \sum_{j=1}^{N_{CRL}} \left( \frac{\Delta a}{da/dB} \right) (Eq. 2-19)
\]

RETURN

Figure 5.1-3  Flowchart for Subprogram LIFCAL
START

CALCULATE STATIC LOAD SCALE FACTORS

DO II ← 1 TO NLOAD BY 1 LOOP OVER ALL LOAD SOURCES

CALCULATE TIME VARYING LOAD SCALE FACTORS

LOCATION IS EXTERIOR SURFACE

FALSE

M4L1

STRESS TERMS $\sigma_{Di}$, $\sigma_{ST}$ CALCULATIONS FOR LOCATION 1
(See Section 5.1.3.3 of [1])

LOCATION IS INTERIOR SURFACE

FALSE

M4L2

STRESS TERMS $\sigma_{Di}$, $\sigma_{ST}$ CALCULATION FOR LOCATION 2
(Similar to M4L1)

A

Figure 5.1-4  Flowchart for Subprogram STRAN1
ASSIGN STATIC STRESSES
\[ \sigma(t) = \sigma_{ST} \]

DO I = 1 TO NLOAD BY 1
LOOP OVER ALL LOAD SOURCES

SCALE TIME VARYING STRESSES
\[ \sigma(t) = \sigma(t) + \lambda \frac{d\sigma}{dt} \sigma_j(t) \]  (Eq. 2-5)

RETURN

Figure 5.1-4  Flowchart for Subprogram STRAN1 (Cont'd)
to the principal stress time history. The stress magnitudes are calculated by calling the following routines.

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>POSITION</th>
<th>ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Exterior Surface</td>
<td>M4L1</td>
</tr>
<tr>
<td>2</td>
<td>Interior Surface</td>
<td>M4L2</td>
</tr>
</tbody>
</table>

M4L1 and M4L2 routines are described in Section 5.1.2.3 in [1].

The flowchart for the STRAN2 routine is given in Figure 5.1-5. This routine is similar to STRAN1 except that the stress magnitudes rather than the load magnitudes are provided as input and hence additional routines are not called for the stress magnitude calculations. Due to the nature of the loading in the EXHEX the maximum principal stress was assumed to be equal to the $\sigma_z$ component.

5.1.2.9 CYCOUN Routine

The flowchart for CYCOUN is given in Figure 5.1-6. This routine is similar to the rainflow cycle counting routine described in Section 5.1.3.5 in [1].

First, the principal stress history is scanned to identify the largest stress and its location. The history is resequenced such that the largest stress is placed at the beginning and end of the stress array. Then, the intermediate points in the history are filtered, leaving only the peaks and troughs. This is done by testing for a sign change in the gradients of adjacent segments. Next, the counting of the cycles begins. Consecutive peaks and troughs are added to a holding array, each time checking whether the new peak-trough segment is greater than the previous one; if so, then a cycle has been closed. Then, the peak and trough corresponding to the closed cycle are removed from the holding array. The cycle is saved if it is large enough, i.e., larger than a user-specified threshold. The procedure is repeated by adding new peaks and troughs to the holding array until another cycle is identified.

Once all the cycles have been identified, the alternating and mean values of each stress cycle are calculated. The stress range of the biggest cycle is divided into one hundred equal stress ranges (or bins) and each stress cycle is assigned to a bin based on its magnitude. This reduces the results of the cycle counting to a number-of-cycles vs. stress-level table. An equivalent mean stress may be calculated for the entire history based on the mean of the biggest cycle. The routine NEUBER, described in Section 5.1.3.6 in [1], is called to estimate the equivalent mean stress.
ASSIGN STATIC STRESSES
\[ \sigma(t) = \lambda_{ST} \lambda_{ST}^* \sigma_{ST} \]

DO I ---- NLOAD
TO NLOAD BY 1
LOOP OVER ALL LOAD SOURCES

SCALE TIME VARYING STRESSES
\[ \sigma(t) = \sigma(t) + \lambda_D \lambda_{DY} \sigma_D(t) \] (Eq. 2-5)

RETURN

Figure 5.1-5 Flowchart for Subprogram STRAN2
START

DO I ← 1
TO NUMBER OF STRESS HISTORY POINTS M BY 1

\( \sigma_{eff_{\text{max}}} > \sigma_{eff_i} \)

TRUE

RECORD THE LARGEST \( \sigma_{eff} \) AND ITS LOCATION
\( \sigma_{eff_{\text{max}}} = \sigma_{eff_i} \)

\( J_{\text{max}} = I \)

DO I ← 1
TO \((M - J_{\text{max}} + 1)\) BY 1

\( J = J_{\text{max}} - 1 + I \)

\( \sigma_{F_i} = \sigma_{eff_i} \)

DO I ← \((M - J_{\text{max}} + 2)\)
TO M BY 1

Figure 5.1-6 Flowchart for Subprogram CYCOUN
Figure 5.1-6  Flowchart for Subprogram CYCOUN (Cont'd)
Figure 5.1-6  Flowchart for Subprogram CYCOUN (Cont'd)
COPY STRESS POINTS TO A HOLDING ARRAY
\[ E_k = \sigma_j \]

\[ K < 3 \]

\[ |E_k - E_{k-1}| < |E_{k-1} - E_{k-2}| \]

\[ |E_{k-1} - E_{k-2}| > \text{TRUNC} \]

\[ I = I + 1 \]
\[ \sigma_{eff1, i} = \max [E_{k-1}, E_{k-2}] \]
\[ \sigma_{eff2, i} = \min [E_{k-1}, E_{k-2}] \]

DISCARD POINTS K-1 AND K-2 AND DECREMENT THE COUNTER
\[ E_{k-2} = E_k \]
\[ K = K - 2 \]

Figure 5.1-6 Flowchart for Subprogram CYCOUN (Cont'd)
RECORD THE FINAL NUMBER OF CYCLES FOUND $N = 1$

**DO** $I = 1$

TO NUMBER OF BINS BY 1

ASSIGN THE STRESS CYCLES TO THE APPROPRIATE BIN

$\text{INEUB} = 1$

**FALSE**

$\text{SM} = \text{SMEANF}$

**TRUE**

$\text{SM} = \text{NEUBER}$

CALCULATE EQUIVALENT MEAN STRESS
(See Section 5.1.3.6 of [1])

RETURN

---

**Figure 5.1-6** Flowchart for Subprogram CYCOUN (Cont'd)
5.1.2.10 BLKGRO Routine

The flowchart for BLKGRO is given in Figure 5.1-7. First, the stress-intensity factor coefficients are calculated in the following routines:

<table>
<thead>
<tr>
<th>ROUTINE</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRIF1</td>
<td>Stress intensity factor coefficients for HEX coil</td>
</tr>
<tr>
<td>STRIF2</td>
<td>Stress intensity factor coefficients for EXHEX</td>
</tr>
</tbody>
</table>

The stress intensity factor routines STRIF1 and STRIF2 are described in Section 5.1.2.11.

The crack growth in a block is calculated as given by Equation 2-17, by summing the growth due to the cycles at each stress level, for each direction (a and c) of crack growth. If growth retardation is considered (IRET = 1), an effective SIF range, $\Delta K_{\text{eff}}$, and stress ratio, $R_{\text{eff}}$, are calculated as per the Willenborg model given by Equations 2-12 through 2-16. Growth calculations are performed after checking for $\Delta K_{\text{eff}} < \Delta K_{\text{th}}$ and $K_{\text{max}} > K_c$ conditions, which are the no-growth and the unstable crack cases, respectively.

5.1.2.11 STRIF1 and STRIF2 Routines

The STRIF1 routine calculates stress intensity factor coefficients for the HEX coil crack configuration. As described in Section 2.1 the standard solution, for an elliptic crack in a finite width plate, given in NASA/FLAGRO [2] is employed.

The STRIF2 routine calculates SIF coefficients for the EXHEX crack configuration. The expressions given in [3] for a crack in a plate are employed.
Figure 5.1-7  Flowchart for Subprogram BLKGRO
CALCULATE BLOCK GROWTH BY SUMMING CYCLE GROWTH RATE

\[
\frac{da}{dB} = \sum_{i=1}^{100} \left( \frac{da}{dN} \right)_i \text{ (Eq. 2-17)}
\]

Figure 5.1-7  Flowchart for Subprogram BLKGRO (Cont’d)
Section 5.2
Low Cycle Fatigue Analysis Software

5.2.1 Introduction

This section presents a description of the computer program which implements the LCF analysis discussed in Section 3.2. The code for analyzing the ATD-HPFTP first stage turbine blade is described below in Section 5.2.2. The overall layout of the program is described by using a main flowchart that refers to other flowcharts, which describe subprograms and key portions of the main program in greater detail. The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the LCF analysis code BLDLCF are given in Section 7.2. The materials characterization subprograms and those subprograms that are of a generic nature, such as the random variate generators, are described in [1], Section 4.1 and Section 4.4 respectively. The relevant user’s guide for running this code is given in Section 6.2. A glossary of standard flowchart symbols is given for the reader’s benefit in Appendix 5.A.

5.2.2 BLDLCF Program

The LCF analysis of the ATD-HPFTP first stage turbine blade is implemented as the FORTRAN program BLDLCF. Figure 5.2-1 shows the structure of the Probabilistic Failure Model (PFM) for the Blade. This section provides the description and flowcharts for program BLDLCF.

5.2.2.1 Main Routine

The master flowchart for the BLDLCF program is given in Figure 5.2-2. The program starts by opening the following input and output files:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDLCD</td>
<td>Input</td>
<td>Analysis data</td>
</tr>
<tr>
<td>BLDLCO</td>
<td>Output</td>
<td>Input data echo, results</td>
</tr>
<tr>
<td>RELATD</td>
<td>Input</td>
<td>Related material data input</td>
</tr>
<tr>
<td>RELATO</td>
<td>Output</td>
<td>Echo of information in RELATD</td>
</tr>
<tr>
<td>DUMP</td>
<td>Output</td>
<td>Results of materials characterization calculations</td>
</tr>
<tr>
<td>IOUTPR</td>
<td>Output</td>
<td>Run information and user-requested information</td>
</tr>
<tr>
<td>LOWLIF</td>
<td>Output</td>
<td>First one percent of sorted fatigue lives</td>
</tr>
</tbody>
</table>

The arrays and variables are then set to their default or initial values. The input data is read from the BLDLCD file. An echo of the input data is written onto BLDLCO. The related material S/N information is read from the file RELATD and

2 Files RELATD and RELATO are opened in INFAGG.
Figure 5.2-1 Structure of the Turbine Blade LCF Probabilistic Failure Model
READ AND ECHO INPUT DATA

PERFORM MATERIALS INFORMATION AGGREGATION
(See [1], Section 4.1.3)

INITIALIZE ARRAYS AND SET DEFAULTS

DO J = 1 TO NHYPER BY 1 OUTER LOOP

CHOOSE $h_{gas}, T_{gas}, \lambda_G, m$ DISTRIBUTIONS
CHOOSE $\lambda_{MA}, \lambda_{TH}$
(See [1], Section 7.6.6)

PERFORM MATERIALS PARAMETER ESTIMATION
(See [1], Section 4.1.5)

SELECT MATERIALS HEAT
(See [1], Section 4.1.5)

DO I = 1 TO NLIFE BY 1 INNER LOOP

DRIVER SELECTION FOR
$h_{gas}, T_{gas}, m, \lambda_G, \omega(t_5), \epsilon_A, T_s,$
$\epsilon_D, \epsilon_B, \lambda_p, \lambda_u, \lambda_{dam}, \lambda_{TMF}$
(See [1], Sections 4.4.3, 4.4.5 & 7.6.6)

Figure 5.2-2 Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF
Figure 5.2-2  Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF (Cont'd)
processed in the INFAGG routine. INFAGG controls the materials information aggregation and is described in [1], Section 4.1.3.

The selection of hyperparameters\(^3\) is performed in the outer DO loop of the simulation by calling the PRYRV routine to obtain the Beta distribution parameters \( \rho \) and \( \theta \) for \( h_{\text{gas}}, T_{\text{gas}}, m, \) and \( \lambda_G \), whose probability distributions are described by Beta distributions. The selection of values for \( \lambda_{MA} \) and \( \lambda_{TH} \) is also performed. The PAREST routine controls the calculations for estimating the parameters for the S/N model. Routine PAREST is described in [1], Section 4.1.5. If materials process variation is included, the materials parameter \( Z \) in [1], Equation 2-48 is selected by calling the NORMGN routine and then transforming the resulting Normal variate to a Lognormal variate.

The inner DO loop for the simulation performs the driver selection. The drivers \( h_{\text{gas}}, T_{\text{gas}}, m, \lambda_G, \omega(t), e_A, T_s, e_D, \varepsilon_B, \lambda_p, \lambda_{\text{dam}}, \text{ and } \lambda_{TMF} \) are selected by calling BETAGN, NORMGN, and PRYRV which draw from Beta, Normal, and Uniform distributions, respectively. The random variate routines BETAGN, NORMGN, and PRYRV are described in [1], Sections 4.4, and 7.6.

In the symmetry DO loop, the materials model parameter \( \varphi \) is found from the minimum of 50 draws of a Weibull distribution. Calls to WEIBGN provide the 50 values of \( \varphi \). Subroutine WEIBGN is described in [1], Section 4.4.6.

When all the S/N model parameters have been selected for the region with S/N data, the S/N curve is tied to the tensile point \( S_o \) by routine KOMO. The routine BLDLIF performs driver transformation and calculates the fatigue life. The flowchart for BLDLIF is given in Figure 5.2-3 and the routine is described below. Subprogram KOMO is described in [1], Sections 4.1.6.

The fatigue lives are arranged in ascending order in a list containing the lowest fifty percent of the lives. The INSORT routine performs an insertion sort with each new fatigue life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIE. Subprogram INSORT is described in [1], Appendix 5.B.

If a truncated Normal distribution was used for the materials shape parameter \( m \), the empirical median S/N curve will be calculated upon user request. The routine SORTM is called to sort the values of \( m \) and the routine EXPCTD calculates the

---

\(^3\) Hyperparameters are discussed in [1], Section 2.1.1.
median S/N curve. Sections 4.1.10 and 4.1.3.12 of [1], describe the routines SORTM and EXPCTD, respectively.

5.2.2.2 BLDLIF Routine

The flowchart for the BLDLIF routine is given in Figure 5.2-3. First, the thermal strain during acceleration is calculated using the acceleration model of Equation 3-2. Next, the deceleration model calculations are performed, Equations 3-3, 3-6, and 3-7, the deceleration slope, thermal strain, and rotor speed are obtained. The total mechanical and total thermal strain-time histories are calculated using Equations 3-5 and 3-4, respectively. Then, the composite strain-time history is obtained by combining the thermal and mechanical strains according to Equation 3-1. Finally, the RAINF3 routine is called. This routine performs a rainflow cycle count and derives the fatigue life.

5.2.2.3 RAINF3 Routine

The flowchart for RAINF3 is given in Figure 5.2-4. First, the equivalent strain history is scanned to identify the largest strain and its location. The history is resequenced such that the largest strain is placed at the beginning and end of the strain array. Then, the intermediate points in the history are filtered leaving only the peaks and troughs. This is done by testing for a sign change in the gradients of adjacent segments. Next, the counting of the cycles begins. Consecutive peaks and troughs are added to a holding array, each time checking if the new peak-trough segment is greater than the previous one; if so, then a cycle has been closed. Then, the peak and trough corresponding to the closed cycle are removed from the holding array. The cycle is saved if it is large enough, i.e., larger than a user-specified threshold. The procedure is repeated by adding new peaks and troughs to the holding array until another cycle is identified.

Once all the cycles have been identified, an equivalent zero-mean strain range is calculated for each cycle using the Walker relation given by Equation 3-8. The life corresponding to each strain cycle is obtained from the S/N curve by calling the GTLIFE routine. The GTLIFE routine is described under materials characterization in [1], Section 4.1.8. Miner's rule is used to accumulate the damage due to each cycle. There are three separate DO loops over the number of cycles in the last three steps, starting with the Walker transformation. This was done to enable vectorization of the DO loops. For running on a scalar machine, these three steps may be embedded within a single DO loop.

5.2.3 BLDLCF Program, Nonparametric Materials Model

The LCF analysis of the ATD-HPFTP first stage turbine blade using the nonparametric materials model is implemented as the FORTRAN program BLDLCF V3.4B1.3. Figure 5.2-5 shows the structure of the PFM for the Blade using the non-
SELECTED VALUES FOR $\varepsilon_B, \lambda_p, \omega(t_s), \lambda_{\alpha}$, $h_{gas}, T_{gas}, T_s, e_A, \lambda_G, m, e_D$

- NOMINAL STRAINS AND ENVIRONMENT
- PARAMETRIC SENSITIVITIES
- REFERENCE STRAIN-TIME HISTORY

\[ \varepsilon_M(t_i) = \lambda_p \lambda_{MA} C_{MS}(t_i) \varepsilon_{Mnom}(\omega_0) \]
\[ \varepsilon_{TH}(t_1) = \lambda_{\alpha} \lambda_{TH}[f(h_{gas}, T_{gas}) + e_A] \]
\[ \varepsilon_{TH}(t_i) = \lambda_{\alpha} \lambda_{TH} \lambda_G \varepsilon_{THnom}(t_i), \quad i = 2, 3, 4 \]
\[ \varepsilon_{TH}(t_5) = \lambda_{\alpha} \lambda_{TH} \varepsilon_{THnom}(t_5) \]
\[ \varepsilon_{TH}(t_6) = \lambda_{\alpha} \lambda_{TH}[f(m, T_s) + e_D] \]
\[ \varepsilon_{T}(t_i) = \varepsilon_M(t_i) + \varepsilon_{TH}(t_i) + \varepsilon_{\varphi}(t_i) \]

Figure 5.2-3 Flowchart for Subprogram BLDLIF
1
FALSE
DO I ← 1
TO NUMBER OF STRESS HISTORY POINTS M BY 1

\[ \sigma_{\text{effmax}} > \sigma_{\text{effi}} \]

TRUE
RECORD THE LARGEST \( \sigma_{\text{eff}} \) AND ITS LOCATION
\[ \sigma_{\text{effmax}} = \sigma_{\text{effi}} \]
\[ J_{\text{MAX}} = I \]

DO I ← 1
TO \( (M - J_{\text{MAX}} + 1) \) BY 1

\[ J = J_{\text{MAX}} - 1 + 1 \]
\[ \sigma_{p_i} = \sigma_{\text{effi}} \]

DO I ← \( (M - J_{\text{MAX}} + 2) \)
TO M BY 1

A

Figure 5.2-4 Flowchart for Subprogram RAINF3
Figure 5.2-4  Flowchart for Subprogram RAINF3 (Cont'd)
TOTAL NUMBER OF POINTS IN ARRAY
NEWTOT = K + 1
INDEX_{NEWTOT} = M + 1

DO I ← 1 TO NEWTOT BY 1

SET UP THE PEAK-TROUGH ARRAY
K = INDEX_I
σ_i = σ_{\text{Pa}}

INITIALIZE COUNTERS
I = 0, J = 0, K = 0

INCREMENT COUNTERS
J = J + 1
K = K + 1

J > NEWTOT

Figure 5.2-4  Flowchart for Subprogram RAINF3 (Cont'd)
COPY STRESS POINTS TO A HOLDING ARRAY

\[ E_k = \sigma_j \]

\[ K < 3 \]

\[ |E_k - E_{k-1}| < |E_{k-1} - E_{k-2}| \]

\[ |E_{k-1} - E_{k-2}| > \text{TRUNC} \]

SINCE CYCLE IS LARGE ENOUGH TO SAVE

\[ l = l + 1 \]

\[ \sigma_{eff \ l} = \max [E_{k-1}, E_{k-2}] \]

\[ \sigma_{eff \ l'} = \min [E_{k-1}, E_{k-2}] \]

DISCARD POINTS K-1 AND K-2 AND DECREMENT THE COUNTER

\[ E_{k-2} = E_k \]

\[ K = K - 2 \]

Figure 5.2-4 Flowchart for Subprogram RAINF3 (Cont'd)
RECORD THE FINAL NUMBER OF CYCLES FOUND

\[ N = 1 \]

\[ \text{DO I -- 1} \]

TO NUMBER OF CYCLES N BY 1

CALCULATE THE EQUIVALENT STRAIN USING WALKER RELATION

\[ \Delta \varepsilon_{EQ} = \left( \frac{\Delta \varepsilon}{2 \varepsilon_{max}} \right)^{w-1} \Delta \varepsilon \]  
(Eq. 3-8)

\[ \text{DO I -- 1} \]

TO NUMBER OF CYCLES N BY 1

Figure 5.2-4  Flowchart for Subprogram RAINF3 (Cont'd)
LIFE \_\_ = GTLIFE

CALCULATE THE FATIGUE LIFE FROM THE S/N CURVE (See [1], Section 4.1.8)

DO I \_\_ 1
TO NUMBER OF CYCLES N BY 1

INVERT THE LIFE

INVLIF \_\_ = \_1 / LIFE \_\_

DO I \_\_ 1
TO NUMBER OF CYCLES N BY 1

SUM THE DAMAGE FRACTIONS

SUMDAM = SUMDAM + INVLIF \_\_

RAINF3 = PERIOD/SUMDAM

RETURN

Figure 5.2-4  Flowchart for Subprogram RAINF3 (Cont'd)
Figure 5.2-5 Structure of the Turbine Blade LCF Probabilistic Failure Model Using the Nonparametric Materials Model
parametric materials model. This section provides the description and flowcharts for program BLDLCF V3.4B1.3 and its routines which differ from Section 5.2.2 above and Section 4.1 of [1].

5.2.3.1 Main Routine

The master flowchart for the BLDLCF V3.4B1.3 program is given in Figure 5.2-6. The program starts by opening the following input and output files:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDLCD</td>
<td>Input</td>
<td>Analysis data</td>
</tr>
<tr>
<td>BDLDCO</td>
<td>Output</td>
<td>Input data echo, results</td>
</tr>
<tr>
<td>RELATD</td>
<td>Input</td>
<td>Related material data input</td>
</tr>
<tr>
<td>RELATO</td>
<td>Output</td>
<td>Echo of information in RELATD</td>
</tr>
<tr>
<td>DUMP</td>
<td>Output</td>
<td>Results of materials characterization calculations</td>
</tr>
<tr>
<td>IOUTPR</td>
<td>Output</td>
<td>Run information and user-requested information</td>
</tr>
<tr>
<td>LOWLIF</td>
<td>Output</td>
<td>First one percent of sorted fatigue lives</td>
</tr>
</tbody>
</table>

The arrays and variables are then set to their default or initial values. The input data is read from the BLDLCD file. An echo of the input data is written onto BDLDCO. The related material S/N information is read from the file RELATD and processed in the INFAGG routine. INFAGG controls the materials information aggregation and is described in Section 5.2.3.2.

The selection of hyperparameters\(^5\) is performed in the outer DO loop of the simulation by calling the PRYRV routine to obtain the Beta distribution parameters \(p\) and \(\theta\) for \(h_{gas}, T_{gas}, m,\) and \(\lambda_G,\) whose probability distributions are described by Beta distributions. The selection of values for \(\lambda_{MA}\) and \(\lambda_{TH}\) is also performed. The PEB routine controls the calculations for bootstrapping the residuals, generating the pseudo S/N data, and then calculating the structural parameters. Routine PEB is described in Section 5.2.3.5.\(^6\) If materials process variation is included, the materials parameter \(Z\) in [1], Equation 2-48 is selected by calling the NORMGN routine and then transforming the resulting Normal variate to a Lognormal variate.

The inner DO loop for the simulation performs the driver selection. The drivers \(h_{gas}, T_{gas}, m, \lambda_G, \omega(t_s), e_A, T_s, e_D, \varepsilon_B, \lambda_P, \lambda_\alpha, \lambda_{dam},\) and \(\lambda_{TMF}\) are selected by calling BETAGN, NORMGN, and PRYRV, which draw from Beta, Normal, and Uniform distributions, respectively. The random variate routines BETAGN, NORMGN, and PRYRV are described in [1], Sections 4.4, and 7.6.

---

\(^4\) Files RELATD and RELATO are opened in INFAGG.

\(^5\) Hyperparameters are discussed in [1], Section 2.1.1.

\(^6\) The bootstrapping calculations are discussed in Section 3.2.7.
Figure 5.2-6  Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF Using the Nonparametric Materials Model
WORSTN
SELECT $\min \{ \varepsilon_i \}_{NSYM}$ FOR BOTH WEIBULL AND LOGNORMAL DISTRIBUTIONS
(See Section 5.2.3.8)

KOMO
TIE S/N CURVE TO TENSILE POINT
(See [1], Section 4.1.6)

BLDLIF
DRIVER TRANSFORMATION CALCULATE FATIGUE LIFE
(See Figure 5.2-3)

INSORT
SAVE AND SORT LOWEST FIFTY PERCENT OF LIVES
(See [1], Appendix 5.B)

WRITE RESULTS TO: BDLCO LOWLIF

SORTM, EXPCTD
CALCULATE EMPIRICAL MEDIAN S/N CURVE
(See [1], Sections 4.1.10 & 4.1.3.12)

STOP

Figure 5.2-6  Main Flowchart for the ATD Blade LCF Analysis Program BDLCF Using the Nonparametric Materials Model (Cont'd)
A call to WORSTN provides the “worst of 50” materials intrinsic variability $\varepsilon$ for both Weibull and Lognormal distributions. The routine WORSTN is described in Section 5.2.3.8.

When all the S/N model parameters have been selected for the region with S/N data, the S/N curve is tied to the tensile point $S_0$ by routine KOMO. The routine BDLIF performs driver transformation and calculates the fatigue life. The flowchart for BDLIF is given in Figure 5.2-3 and the routine is described below. Subprogram KOMO is described in [1], Sections 4.1.6.

The fatigue lives are arranged in ascending order in a list containing the lowest fifty percent of the lives. The INSORT routine performs an insertion sort with each new fatigue life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIF. Subprogram INSORT is described in [1], Appendix 5.B.

The empirical median S/N curve is calculated next. The routine SORTM is called to sort the values of $m$ and the routine EXPCTD calculates the median S/N curve. Sections 4.1.10 and 4.1.3.12 of [1] describe the routines SORTM and EXPCTD, respectively.

5.2.3.2 INFAGG Routine

The flowchart for the INFAGG routine is given in Figure 5.2-7. The routine controls the calls to the data input and information aggregation calculation routines. INFAGG starts by opening the following input and output files: 7

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELATD</td>
<td>Input</td>
<td>Related material data input</td>
</tr>
<tr>
<td>RELATO</td>
<td>Output</td>
<td>Related material data echo</td>
</tr>
</tbody>
</table>

The arrays are then set to their default or initial values by routine INIT. Routine RCE reads the data from files SPECFD and RELATD, transforms (or converts) the stresses to an equivalent stress ratio of $R = -1.0$, and echoes the data to files SPECFO and RELATO. Routines INIT and RCE are described in [1], Sections 4.1.3.1 and 4.1.3.2.

The information aggregation begins with linear regression calculations performed by routine SW2SU2 on the combined specific and related data. Then the constraints on the shape parameters $\{m_i\}$ implied by the user-provided $C_0$ constraint are calculated by FINDMC. The routines SW2SU2 and FINDMC are described in [1], Sections 4.1.3.12 and 4.1.3.13.

---

7 The nonparametric model does not have the capability to utilize related data at this time.
Figure 5.2-7  Flowchart for Subprogram INFAGG
Figure 5.2-7  Flowchart for Subprogram INFAGG (Cont'd)
CONCAV
IMPOSES S/N CURVE CONCAVITY CONSTRAINT
(See [1], Section 4.1.3.10)

WRITE TO DUMP PRIOR AND POSTERIOR CREDIBILITY RANGES
\( \hat{C}, \hat{m}, \sigma^2 \)

MEDIAN
CALCULATE MEDIAN \( m \) VALUES
(See [1], Section 4.1.3.11)

EXPCTD
CALCULATE PARAMETER VALUES FOR MEDIAN S/N CURVE
(See [1], Section 4.1.3.12)

\( \pi(m_j) \)
LIFE REGION BOUNDARIES SPECIFIC S/N DATA

RETURN

Figure 5.2-7 Flowchart for Subprogram INFAGG (Cont'd)
Figure 5.2-7  Flowchart for Subprogram INFAGG (Cont'd)
WRITE TO DUMP
\( \hat{m}, \pi(m), \sigma^2 \)

\( \pi(m) \)
LIFE REGION BOUNDARIES
SPECIFIC S/N DATA

RETURN

EXPB
CALCULATE PARAMETER VALUES
FOR MEDIAN S/N CURVE
(See Section 5.2.3.4)

RESIDUALS
S VALUES

RETURN

Figure 5.2-7  Flowchart for Subprogram INFAGG (Cont'd)
described in Section 5.2.3.3 and in [1], Section 4.1.3.5, respectively. The remaining routine calls depend upon the choice of distribution for the shape parameters.

The Uniform distribution case begins with the confidence interval calculations performed by INTRVL. By definition, the prior credibility ranges are the confidence intervals. If materials processes variation is specified, GTPVAR calculates $\sigma^2$, Equation 2-49 of [1], the extent of departures from the multiple heat median S/N curve warranted by the available information. The credibility ranges, $C$ constraint, and the user-provided range information are combined by routine FNSRNG to obtain posterior credibility ranges on the shape parameters $\pi(m_j)$.\(^8\) The user-supplied $m$ ranges for the non-data life regions to the right of those with data are added to the array containing the $\pi(m_j)$ by routine ADDREG.\(^9\) Concavity constraints are applied within subprogram CONCAV. The results of the calculations above are written to file DUMP. Finally, the median S/N curve is calculated. The median $m$'s are found by MEDIAN and then used by EXPCTD to obtain the median curve parameters which are written to file DUMP. Routines INTRVL, GTPVAR, FNSRNG, ADDREG, CONCAV, MEDIAN, and EXPCTD are described in [1], Sections 4.1.3.6, 4.1.3.7, 4.1.3.8, 4.1.3.9, 4.1.3.10, 4.1.3.11, and 4.1.3.12, respectively.

The truncated Normal distribution case begins with the Bayesian analysis performed by MUSIG to find the Normal distribution parameters for the $m$'s. If materials process variation is requested, GTPVAR calculates $\sigma^2$, the extent of departures from the multiple heat median S/N curve warranted by the available information, by using Equation 2-49 of [1]. The $C$ constraint and the user-provided range information are combined by routine NORRNG to obtain posterior credibility ranges on the shape parameters $\pi(m_j)$.\(^8\) The user-supplied $m$ ranges and Normal distribution parameters for the non-data life regions to the right of those with data are added to the arrays containing the $\pi(m_j)$, $m^*$, and $\sigma^2$ by routine ADDRGN.\(^9\) Concavity constraints are applied within subprogram CONCAV. Then results of the calculations above are written to file DUMP. Routines MUSIG, GTPVAR, NORRNG, ADDRGN, and CONCAV are described in [1], Sections 4.1.3.13, 4.1.3.7, 4.1.3.14, 4.1.3.15, and 4.1.3.10.

The bootstrapping option uses $m$ and $K$ estimates to obtain the median curve parameters using EXPB, which are then written to file DUMP. Routine EXPB is described in Section 5.2.3.4.

\(^8\) Combining information to obtain the posterior credibility ranges on $m$ is discussed in [1], Page 2-13.

\(^9\) No data regions to the right are discussed in [1], Page 2-17.
5.2.3.3 SW2SU2 Routine

The flowchart for the SW2SU2 routine is given in Figure 5.2-8. The routine performs the $y$ on $x$ and $x$ on $y$ regressions to obtain the sample variances $S_x^2$, $S_y^2$, and $S_{xy}$, and the residual variances $S_w^2$ and $S_0^2$ for each life region. For the calculations, $x$ is equal to In $S$ and $y$ is equal to In $N$. The routine SW2SU2 starts by initializing the arrays required for the calculations.

Within the outer region DO loop are two sets of nested DO loops, where the region counter $L = 1, ..., R$, and $R$ is the number of life regions with S/N data. In each set of DO loops, the outer loop is for each S/N data set, $j = 0, ..., P$, and the inner DO loop is for each data point in each data set, $k = 1, ..., N_j$. The first step is to calculate the sample means $\bar{x}_j$ and $\bar{y}_j$ for each data set in each region. Then the sample variances and degrees of freedom for each region in each data set are calculated as follows:

\[
N S_x^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (x_{jk} - \bar{x}_j)^2
\]

\[
N S_y^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (y_{jk} - \bar{y}_j)^2
\]

\[
N S_{xy} = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (x_{jk} - \bar{x}_j)(y_{jk} - \bar{y}_j)
\]

\[
N = \sum_{j=0}^{P} (N_j - 1) - 1
\]

where $S_x^2$, $S_y^2$, and $S_{xy}$ are the sample variance of $x$, sample variance of $y$, and sample covariance of $x$ and $y$, and $N$ is the number of degrees of freedom for each life region, respectively. If $S_{xy}$ is non-negative, the data does not support the analysis assumptions and the program run will be terminated. The sample variances are used to calculate the regression parameters $d$ and $b$ of Equations 2-20 and 2-21 of [1].

---

10 $R$ is equal to one for the strain formulation.
Figure 5.2-8  Flowchart for Subprogram SW2SU2
SUM OVER ALL DATA POINTS IN EACH DATA SET \( J \) FOR REGION \( L \)

\[ N S_x^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (x_{jk} - \bar{x}_j)^2 \]

\[ N S_y^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (y_{jk} - \bar{y}_j)^2 \]

\[ N S_{xy} = \sum_{j=0}^{P} \sum_{k=1}^{N_j} (x_{jk} - \bar{x}_j)(y_{jk} - \bar{y}_j) \]

SUM OVER EACH DATA SET \( J \) FOR REGION \( L \)

\[ N = \sum_{j=0}^{P} (N_j - 1) - 1 \]

\[ NS_{xy} \geq 0 \]

TRUE

TRMNAT

STOP PROGRAM

FALSE

CALCULATE SAMPLE VARIANCES

\( S_x^2, S_y^2, S_{xy} \)

STOP

Figure 5.2-8 Flowchart for Subprogram SW2SU2 (Cont'd)
CALCULATE $d$ AND $b$
(Equations 2-20 and 2-21 of [1])

\[ d = \frac{S_{xy}}{S_x^2} \]
\[ b = \frac{S_{xy}}{S_y^2} \]

CALCULATE $\hat{m}$ AND $\hat{K}$

\[ \hat{m} = -d \]
\[ \hat{K} = \exp \left[ \frac{\hat{m} \bar{y}_0 - \bar{y}_0}{\hat{m}} \right] \]

DO $J \rightarrow 0$

TO $P$ BY 1

DO $K \rightarrow 1$

TO $N_j$ BY 1

SUM OVER ALL DATA POINTS IN EACH DATA SET $J$ FOR REGION $P$

\[ N S_{\hat{w}}^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} \hat{w}_{jk}^2 \]
\[ N S_{\hat{u}}^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_j} \hat{u}_{jk}^2 \]

Figure 5.2-8  Flowchart for Subprogram SW2SU2 (Cont'd)
CALCULATE RESIDUAL VARIANCES $S_w^2$ and $S_u^2$

CALCULATE RESIDUALS

$$e_k = \hat{w}_{ok} \sqrt{\frac{N_o}{N_o - 2}}$$

Figure 5.2-8  Flowchart for Subprogram SW2SU2 (Cont'd)
\[ d = \frac{S_{xy}}{S_x^2} \quad \text{and} \quad b = \frac{S_{xy}}{S_y^2} \]

\[ \hat{m} = -d \quad \text{and} \quad \hat{K} = \exp \left( \frac{\hat{m} \bar{x}_o - \bar{y}_o}{\hat{m}} \right) \]

The second set of DO loops calculates the residuals \( e \) and the residual variances \( S_{\hat{w}}^2 \) and \( S_{\hat{u}}^2 \) for each life region given by

\[ e_k = \hat{w}_{ok} \sqrt{\frac{N_0}{N_0 - 2}} \]

\[ N S_{\hat{w}}^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_i} \hat{w}_{jk}^2 \]

\[ N S_{\hat{u}}^2 = \sum_{j=0}^{P} \sum_{k=1}^{N_i} \hat{u}_{jk}^2 \]

where

\[ \hat{w}_{jk} = (y_{jk} - \bar{y}_j) - d (x_{jk} - \bar{x}_j) \]

\[ \hat{u}_{jk} = (x_{jk} - \bar{x}_j) - b (y_{jk} - \bar{y}_j) \]

from Equations 2-20 and 2-21 of [1].

### 5.2.3.4 EXPB Routine

The flowchart for the EXPB routine is given in Figure 5.2-9. The routine controls the calls to the median curve calculations for the bootstrap option. The routine uses the point estimates for the \( m \) and \( K \) to calculate the remainder of the parameters consistent with \( m \), \( K \), and the specific material data set. The stress values corresponding to the life region boundaries are obtained from FINDSB. If the tensile point \( S_o \) for the stress formulation is being used, then the S/N curve can be tied to \( S_o \) by routine KOMO.\(^{11}\) The results of the calculations are written to file DUMP. Routines FINDSB and KOMO are described in [1], Sections 4.1.5.7 and 4.1.6, respectively.

\(^{11}\) The tensile point calculations are included in routine EXPB in anticipation of future work on the bootstrap option.
FINDSB

CALCULATE $S_{t,i+1}^*$ FOR USE IN LIFE CALCULATION
(See [1], Section 4.1.5.7)

USE $S_o$ ?

NO

YES

KOMO

CALCULATE $m_o$ AND $K_o$
(See [1], Section 4.1.6)

WRITE TO DUMP
$\beta_o, \hat{k}, [K_i, m_i], S_{t,i+1}^*$

RETURN

Figure 5.2-9  Flowchart for Subprogram EXPB
5.2.3.5 PEB Routine

The flowchart for the PEB routine is given in Figure 5.2-10. The routine controls the calls to the bootstrapping calculations. The calculations begin by the call to routine PICRES which performs the bootstrapping on the residuals and then generates the pseudo S/N data. Routine MREGR performs the regression to obtain a value of \( m \) for the pseudo S/N data. The routines PICRES and MREGR are described in Sections 5.2.3.6 and 5.2.3.7, respectively.

The remaining calculations find the \( \{K_j\} \) and \( \beta_o \) parameters consistent with the pseudo S/N data and the calculated \( m \). The calculations begin by routine TRNSFM transforming the specific material S/N data.\(^{12}\) The transformation produces the \( \{Z_i\} \) as a function of the S/N data, the \( m \), and the life region boundary. Then, the sample mean and variance of \( Z \) are calculated by routine SMNVAR. KBETA computes the estimates of \( k \) and \( \beta_o \). Then, the \( K \) is calculated by routine FINDK using Equations 2-37 through 2-41 of [1]. Finally, the stress value corresponding to the life region boundary is obtained from FINDSB. Routines TRNSFM, SMNVAR, KBETA, FINDK, and FINDSB are described in [1], Sections 4.1.5.3 through 4.1.5.7.

5.2.3.6 PICRES Routine

Routine PICRES bootstraps the residuals and generates the pseudo S/N data. The bootstrapping is performed by sampling with replacement on the set of residuals \( e \) for each stress value \( S_i \). Then the pseudo S/N data is generated by calculating a new life value \( N_i^* \) for each stress value and selected residual \( e_i^* \) according to

\[
N_i^* = \hat{a} S_i^{-\hat{m}} e_i^*
\]

The inflation of the residuals by \( \sqrt{\frac{N_o}{N_o - 2}} \) was performed in routine SW2SU2.

5.2.3.7 MREGR Routine

The flowchart for the MREGR routine is given in Figure 5.2-11. The routine performs the \( y \) on \( x \) and \( x \) on \( y \) regressions to obtain the estimate of the shape parameter \( m \). For the calculations, \( x \) is equal to \( \ln S \) and \( y \) is equal to \( \ln N \). MREGR starts by initializing the arrays required for the calculations.

---

\(^{12}\) The S/N data transformation is discussed in [1], Page 2-16.
PICRES
BOOTSTRAP RESIDUALS AND GENERATE PSEUDO S/N DATA
(See Section 5.2.3.6)

MREGR
REGRESS TO OBTAIN $m$
(See Section 5.2.3.7)

TRNSFM
TRANSFORM S/N DATA INTO $\{Z_i\}$
(See [1], Section 4.1.5.3)

SMNVAR
COMPUTE $\bar{Z}$ AND $S_Z^2$
(See [1], Section 4.1.5.4)

KBETA
COMPUTE ESTIMATES FOR $k$ AND $\beta_0$
(See [1], Section 4.1.5.5)

Figure 5.2-10 Flowchart for Subprogram PEB
FINDK

COMPUTE \( \{K_j\} \)
(See [1], Section 4.1.5.6)

FINDSB

CALCULATE \( S^*_j, i + 1 \) FOR USE IN LIFE CALCULATION
(See [1], Section 4.1.5.7)

\[ \beta_{o}, \{K_j, m_j\}, S^*_j, i + 1 \]

RETURN

Figure 5.2-10  Flowchart for Subprogram PEB (Cont’d)
INITIALIZE ARRAYS

DO \( L \rightarrow 1 \)
TO \( R \) BY 1

CALCULATE \( \bar{x} \) AND \( \bar{y} \)
IN REGION \( L \)

DO \( K \rightarrow 1 \)
TO \( N_o \) BY 1

SUM OVER ALL DATA POINTS FOR REGION \( L \)

\[ NS_x^2 = \sum_{k=1}^{N_o} (x_k - \bar{x})^2 \]
\[ NS_{xy} = \sum_{k=1}^{N_o} (x_k - \bar{x})(y_k - \bar{y}) \]

\( NS_{xy} \geq 0 \)

TRUE

TRMNAT
STOP PROGRAM

FALSE

CALCULATE SAMPLE VARIANCES

\( S_x^2, S_{xy} \)

STOP

Figure 5.2-11  Flowchart for Subprogram MREGR
Figure 5.2-11 Flowchart for Subprogram MREGR (Cont'd)

Calculate $d$
(Equation 2-20 of [1])

$$d = \frac{S_{xy}}{S_x^2}$$

Calculate $\hat{m}$

$$\hat{m} = -d$$

RETURN
Within the outer region DO loop are two inner DO loops, where the region counter \( L = 1, \ldots, R \), is the number of life regions with S/N data.\(^{13}\) Each inner DO loop is for each data point, \( k = 1, \ldots, N_0 \). The first step is to calculate the sample means \( \bar{x} \) and \( \bar{y} \) in each region. Then the sample variances for each region are calculated as follows:

\[
NS_x^2 = \sum_{j=1}^{N_0} (x_j - \bar{x})^2
\]

\[
NS_{xy} = \sum_{j=1}^{N_0} (x_j - \bar{x})(y_j - \bar{y})
\]

where \( S_x^2 \) and \( S_{xy} \) are the sample variance of \( x \) and the sample covariance of \( x \) and \( y \) for each region, respectively. If \( S_{xy} \) is non-negative, the data does not support the analysis assumptions and the program run will be terminated. The sample variances are used to calculate the regression parameter \( d \) of [1], Equation 2-20,

\[
d = S_{xy} / S_x^2.
\]

Then the shape parameter \( m \) is given by

\[
m = -d.
\]

5.2.3.8 WORSTN Routine

The following routine can be used to provide an analytic solution to the problem of selecting the smallest of \( N \) lives for either the parametric or bootstrapping characterization of materials model specification error.

Routine WORSTN performs the worst of \( N \) selection of the materials intrinsic variation parameter \( \varepsilon \) described in Section 3.2.7.3. The first step is to obtain a Uniform(0,1) random variate for \( F \). Then the Weibull worst of \( N \) variate is given by

\[
\varepsilon = \exp \left[ \left( \ln \left( \frac{-\ln(1-F)}{N} \right) - \ln(\ln 2) \right) \frac{m}{\beta_0} \right]
\]

Finally the Lognormal worst of \( N \) variate is obtained using the algorithm given in 26.2.23 of [4].

\(^{13}\) \( R \) is currently equal to 1 for the bootstrapping option, but the region DO loop has been included in anticipation of future work on the bootstrap option.
Section 5.3
High Cycle Fatigue Analysis Software

5.3.1 Introduction

This section presents a description of the computer program which implements the HCF analysis discussed in Section 4. The code for analyzing the ATD-HPOTP first and third stage turbine blades is described below in Section 5.3.2. The overall layout of the program is described by using a main flowchart that refers to other flowcharts, which describe subprograms and key portions of the main program in greater detail. The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the HCF analysis code BLDHCF are given in Section 7.3. The materials characterization subprograms and those subprograms that are of a generic nature, such as the random variate generators, are described in [1], Section 4.1 and Section 4.4 respectively. A glossary of standard flowchart symbols is given for the reader’s benefit in Appendix 5.A.

5.3.2 BLDHCF Program

The HCF analysis of the ATD-HPOTP first and third stage turbine blades is implemented as the FORTRAN program BLDHCF. Figure 5.3-1 shows the structure of the PFM for the Blade. This section provides the description and flowcharts for program BLDHCF.

5.3.2.1 Main Routine

The master flowchart for the BLDHCF program is given in Figure 5.3-2. The program starts by opening the following input and output files:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDHCD</td>
<td>Input</td>
<td>Analysis data</td>
</tr>
<tr>
<td>BLDHCO</td>
<td>Output</td>
<td>input data echo, results</td>
</tr>
<tr>
<td>RELATD</td>
<td>Input</td>
<td>Related material data input</td>
</tr>
<tr>
<td>RELATO</td>
<td>Output</td>
<td>Echo of information in RELATD</td>
</tr>
<tr>
<td>DUMP</td>
<td>Output</td>
<td>Results of materials characterization calculations</td>
</tr>
<tr>
<td>IOUTPR</td>
<td>Output</td>
<td>Run information and user-requested information</td>
</tr>
<tr>
<td>LOWLIF</td>
<td>Output</td>
<td>First one percent of sorted fatigue lives</td>
</tr>
</tbody>
</table>

Routine DRVRIN is called to read the input data from the BLDHCD file. An echo of the input data is written onto BLDHCO. DRVRIN is described in Section 5.3.2.2. The related material S/N information is read from the file RELATD and processed in the INFAGG routine. INFAGG controls the materials information aggregation and

14 Files RELATD and RELATO are opened in INFAGG.
Figure 5.3-1 Structure of the Turbine Blade HCF Probabilistic Failure Model
START

DRVRIN
READ AND ECHO
INPUT DATA
(See Section 5.3.2.2)

INFAGG
PERFORM MATERIALS
INFORMATION AGGREGATION
(See [1], Section 4.1.3)

INITIALIZE ARRAYS
AND SET DEFAULTS

DO J = 1 TO NHYPER BY 1
OUTER LOOP

PAREST
PERFORM MATERIALS
PARAMETER ESTIMATION
(See [1], Section 4.1.5)

OPTIONAL
IF MATERIALS
PROCESS VARIATION
WAS USED

SELECT MATERIALS HEAT
(See [1], Section 4.1.5)

DO I = 1 TO NLIFE BY 1
INNER LOOP

SELECT
DRIVER SELECTION
(See Section 5.3.2.3)

Figure 5.3-2 Main Flowchart for the ATD Blade HCF Analysis Program BLDHCF
DO M ← 1
TO NSYM BY 1
SYMMETRY LOOP

WEIBGN
SELECT MATERIALS SCATTER PARAMETER \( \varphi \)
(See [1], Section 4.4.6)

\[ \varphi = \min \{ \varphi_1, \ldots, \varphi_{NSYM} \} \]

KOMO
TIE S/N CURVE TO TENSILE POINT
(See [1], Section 4.1.6)

BLDHLF
DRIVER TRANSFORMATION CALCULATE FATIGUE LIFE
(See Section 5.3.2.4)

INSORT
SAVE AND SORT LOWEST FIFTY PERCENT OF LIVES
(See [1], Appendix 5.B)

WRITE RESULTS TO: BLDHCO LOWLIF

OPTIONAL
IF TRUNCATED NORMAL VARIATION ON \( m \) WAS USED

SORTM, EXPCTD
CALCULATE EMPIRICAL MEDIAN S/N CURVE
(See [1], Sections 4.1.10 & 4.1.3.12)

STOP

Figure 5.3-2 Main Flowchart for the ATD Blade HCF Analysis Program BLDHCF (Cont'd)
is described in [1], Section 4.1.3. The arrays and variables are then set to their default or initial values.

In the outer DO loop of the simulation, the PAREST routine controls the calculations for estimating the parameters for the S/N model. Routine PAREST is described in [1], Section 4.1.5. If materials process variation is included, the materials parameter Z in [1], Equation 2-48 is selected by calling the NORMGN routine and then transforming the resulting Normal variate to a Lognormal variate.

The inner DO loop for the simulation performs the driver selection. The SELECT routine controls the driver selection and is described in Section 5.3.2.3.

In the symmetry DO loop, the materials model parameter \( \varphi \) is found from the minimum of 54 draws of a Weibull distribution. Calls to WEIBGN provide the 54 values of \( \varphi \). Subroutine WEIBGN is described in [1], Section 4.4.6.

When all the S/N model parameters have been selected for the region with S/N data, the S/N curve is tied to the tensile point \( S_o \) by routine KOMO. The routine BLDHLF performs driver transformation and calculates the fatigue life. The BLDHLF routine is described in Section 5.3.2.4. Subprogram KOMO is described in [1], Sections 4.1.6.

The fatigue lives are arranged in ascending order in a list containing the lowest fifty percent of the lives. The INSORT routine performs an insertion sort with each new fatigue life. When the outer DO loop is completed, the list of lives representing the left-hand tail of the failure distribution is written to file LOWLIF. Subprogram INSORT is described in [1], Appendix 5.B.

If a truncated Normal distribution was used for the materials shape parameter \( m \), the empirical median S/N curve will be calculated upon user request. The routine SORTM is called to sort the values of \( m \) and the routine EXPCTD calculates the median S/N curve. Sections 4.1.10 and 4.1.3.12 of [1] describe the routines SORTM and EXPCTD, respectively.

5.3.2.2 DRVRIN Routine
The DRVRIN routine controls the input/output of the driver distributions and the structural and geometric parameters. The input data is read from file BLDHCD and the data is written to file BLDHCO.

5.3.2.3 SELECT Routine
The SELECT routine controls the driver selection. The drivers \( \omega \), \( R_{root} \), \( R_{avg} \), \( C \), \( r_d \), \( \lambda_B \), \( \lambda_D \), and \( m_w \) are selected by calling NORMGN and PRYRV which draw from
Normal and Uniform distributions respectively. The random variate routines NORMGN and PRYRV are described in [1], Sections 4.4 and 7.6.

5.3.2.4 BLDHLF Routine

BLDHLF performs the driver transformation and fatigue life calculation. The flowchart for the driver transformation is given in Figure 5.3-3. First, the mass flow rate \( \dot{m} \) and the change in enthalpy \( \Delta h \) are calculated using the performance balance characterization. Next, the blade root mean stress \( \bar{\sigma}_{BR} \) calculation is performed, Equation 4-1. The blade undamped vibratory stress \( \sigma_{UD} \) is calculated based on the empirical characterization as a function of \( \bar{\sigma}_{BR} \). The blade damper effectiveness characterization model is used to obtain the ratio of damped vibratory stress to undamped vibratory stress \( \sigma_D / \sigma_{UD} \) as a function of the centrifugal force produced by the blade damper. Then the mean and alternating stresses, the maximum and minimum stresses, and the stress ratio are calculated using Equations 4-2 through 4-6.

The flowchart for the fatigue life calculation is given in Figure 5.3-4. First, the equivalent zero mean stress is calculated using the Walker relation of Equation 4-7. The life in cycles \( N_f \) corresponding to the equivalent stress cycle is obtained from the S/N curve by calling the GTLIFE routine. The GTLIFE routine is described under materials characterization in [1], Section 4.1.8. The failure life in seconds \( L \) is calculated as a function of \( N_f \), the rotor speed \( \omega \), and the number of stators \( N_s \).

References


Figure 5.3-3  Structure of the Driver Transformation for the Turbine Blade HCF Model

\[ \sigma_{\text{MAX}} = \sigma_{\text{BR}}, \quad \sigma_{\text{ALT}} = \sigma_{\text{UD}} (\sigma_D / \sigma_{\text{UD}}) \]

\[ \sigma_{\text{MAX}} = \sigma_{\text{MIN}} + \sigma_{\text{ALT}} \]

\[ \sigma_{\text{MIN}} = \sigma_{\text{MIN}} - \sigma_{\text{ALT}} \]

\[ R = \sigma_{\text{MIN}} / \sigma_{\text{MAX}} \]
SELECTED VALUES FOR $\omega, R_{\text{root}}, R_{\text{avg}}, C, r_d$

PARAMETRIC SENSITIVITIES

**SELECTED VALUES FOR $\lambda_B, \lambda_D$**

DRIVER TRANSFORMATION

$\sigma_{\text{MAX}}, R$

ZERO MEAN STRESS TRANSFORMATION VIA WALKER RELATION

$\sigma_{\text{EQ}} = \left(\frac{1 - R}{2}\right)^{m_w} \sigma_{\text{MAX}}$

**SELECTED VALUE FOR $m_w$**

MATERIALS S/N MODEL

$N_f = K_f m_j \sigma_{\text{EQ}}^m \varphi^m$

CYCLES TO FAILURE $N_f$

CALCULATE LIFE

$L = \frac{60 N_f}{\omega N_s}$

**NUMBER OF STATORS $N_s$ & ROTOR SPEED $\omega$**

PREDICTED FAILURE TIME

*Figure 5.3-4* Structure of the Failure Life Calculation for Turbine Blade HCF
Appendix 5.A
Program Flowchart Symbols

The symbols employed in the flowcharts are given in Figure 5.A-1.

![Flowchart Symbols]

Figure 5.A-1  Program Flowchart Symbols
6.0 Software User’s Documentation
Section 6.1
Crack Growth Analysis User's Guide

6.1.1 PROCRK Program

A user's guide for running the crack growth analysis code PROCRK is given here. The crack growth analysis methodology is discussed in Section 2.2, the program description and flowcharts are presented in Section 5.1, and the code structure and listing are provided in Section 7.1.

The PROCRK program was used to analyze the crack growth failure of the HPOTP heat exchanger (HEX) coil and the proposed external heat exchanger (EXHEX). The dynamic load input for the program consists of narrow-band and sinusoidal reference time histories. These reference time histories are generated using the program NBSIN. The output of PROCRK includes the simulated B-lives and a list of the lowest one percent of lives. The list of lives may be used as input to the regression programs of Section 4.2 in [1] in order to compute the parameters of the Bayesian prior failure distribution. This prior distribution and the success/failure data are used as input to the Bayesian updating program BAYES to obtain a posterior failure distribution. Programs NBSIN and BAYES are described in Sections 4.5 and 4.3, respectively, of [1].

6.1.2 How to Use Program PROCRK

The program PROCRK is intended to be run in batch (i.e., background) mode. PROCRK requires the input file CRKDAT and a set of load data files containing the reference time histories. The names of the load data files must be defined by the user. The file CRKDAT contains the analysis control parameters, driver distributions, engineering analysis parameters, and materials information. A complete description of the input data for the CRKDAT data file is given in Section 6.1.3.1.

The results from the PROCRK program are written to three output files: CRKRES, IOUTPR, and LOWLIF. CRKRES contains the echo of the information in CRKDAT and the results of the simulation. File IOUTPR contains an echo of the analysis parameters and, if requested, a dump of intermediate calculations. If the program terminates prematurely, an error message will be printed in the IOUTPR file. A list of error messages and possible remedies for the problems is given in Section 6.1.6. LOWLIF contains the first one percent of the lives of the simulated failure distribution.
6.1.3 Description of Input Data Files

Annotated examples of the complete CRKDAT data file format structure for the HEX coil and EXHEX problems are presented in Figures 6.1-1 and 6.1-2, respectively. The data lines of the input files are given in boxes, with a description of each data line located above or adjacent to each box. The specific input parameters of Figures 6.1-1 and 6.1-2 are individually defined in Section 6.1.3.1. Input parameter values given in Figures 6.1-1 and 6.1-2 are not necessarily those used in the application case studies of Section 2.

The input data is read by free format statements from file CRKDAT. Thus, the numbers may be provided sequentially on a line up to 80 characters in length, with each number separated by a blank character or comma. Each number may also be on a separate line in the file. However, it is recommended that the input format suggested in Figures 6.1-1 and 6.1-2 be followed whenever possible.

6.1.3.1 Input File CRKDAT

The required data for the CRKDAT file is divided into the four blocks shown in Figure 6.1-3: analysis parameters, driver information, load and stress, and materials information. The analysis parameters block contains the analysis parameters and the keys to select the program options. The driver information block contains the parameters that define the driver distributions. The number of dynamic loads, the magnitudes of the dynamic loads/stresses, the load file names, the static loads/stresses, and geometry are given in the load and stress block. The materials information block contains the \( \frac{da}{dN} \) vs. \( \Delta K \) data, the stress ratio, and the yield strength.

The input parameters are described below by using the following convention: the input variable names are indicated by **BOLD UPPERCASE** letters; the variable types are specified as character [CHR], integer [INT], real [RE], and double precision real [DRE]; the function of the variable is **underlined** and followed by a description and a list of options, when appropriate; the program and file names are indicated by **UPPERCASE** letters. A consistent set of units is given in parentheses for specifying dimension, load, stress, and stress intensity factor input parameters. All character strings must be enclosed by ‘single quotes’. The user is reminded about the difference between the number “0” and the letter “O” when preparing the input files.
Problem type
Crack growth model type
Random number seed
Output dump controller
Inner loop size
Outer loop size
Growth retardation switch (on)
Neuber's rule switch (on)
Number of B-lives

Decimal equivalent of percentages for B-lives

| 0.0001 | 0.0005 | 0.001 | 0.005 | 0.01 |

Two Beta distributions on weld offset information

| 0.06  | 0.06  | 0.00  | 0.00  | 0.0  | 0.0  |
| 0.00  | 0.00  | 0.00  | 0.00  | 0.0  | 0.0  |
| 1.00  |       |       |       |      |      |

Beta distribution on duct inside diameter information

| 0.1885 | 0.1915 | 0.50  | 0.50  | 0.5  | 20.  |

Beta distribution on wall thickness information

| 0.0113 | 0.0157 | 0.27273 | 0.27273 | 0.5  | 20.  |

Beta distribution on initial crack aspect ratio information

| 0.20  | 1.00  | 0.50  | 0.50  | 0.0  | 0.0  |

Beta distribution on initial crack size information

| 0.005 | 0.005 | 0.0 | 0.0 | 0.0 | 0.0 |

| 2.00  | 2.00  | 0.15 | 1.00 |
| 2.00  | 2.00  | 0.20 | 1.00 |
| 486.  | 666.  | 29.  | 56.5 |
| 799.  | 908.  | 49.5 | 48.  |
| 3808. | 4177. | 69.  | 69.  |

Narrow-band random load scale factor
Sinusoidal load scale factor
Normal distribution on inner wall temperature information
Normal distribution on outer wall temperature information
Normal distribution on internal pressure information

| 0.80  | 1.20  |
| 0.50  | 1.50  |
| 0.80  | 1.20  |
| 0.90  | 1.10  |

Uniform distribution bounds for weld offset accuracy factor
Uniform distribution bounds for aerodynamic load scale factor
Uniform distribution bounds for aerostatic load scale factor
Uniform distribution bounds for aeroloids stress analysis accuracy factor

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem
Uniform distribution bounds for dynamic stress analysis accuracy factor
Uniform distribution bounds for Neuber's rule accuracy factor
Uniform distribution bounds for threshold stress intensity factor uncertainty
Uniform distribution bounds for critical stress intensity factor uncertainty
Uniform distribution bounds for stress intensity factor calculation accuracy
Uniform distribution bounds for crack growth calculation accuracy factor

Number of dynamic loads

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerostatic load: $P, M_x, M_y, M_z, V_y, V_z$</td>
<td></td>
</tr>
<tr>
<td>Dynamic loads: file name, load type, $P, M_x, M_y, M_z, V_y, V_z$</td>
<td></td>
</tr>
<tr>
<td>'NBM3'</td>
<td>1</td>
</tr>
<tr>
<td>'SIN1'</td>
<td>2</td>
</tr>
<tr>
<td>'AERO1'</td>
<td>3</td>
</tr>
<tr>
<td>External pressure, $P_o$</td>
<td>3640</td>
</tr>
<tr>
<td>Critical duct location</td>
<td>2</td>
</tr>
<tr>
<td>Angular position about the duct circumference, $\phi$</td>
<td>10</td>
</tr>
<tr>
<td>Willenborg retardation model constant</td>
<td>2.3</td>
</tr>
<tr>
<td>Reference time history period, $T$</td>
<td>1.0</td>
</tr>
<tr>
<td>Noise filter</td>
<td>0.0</td>
</tr>
<tr>
<td>Number of points in reference time histories</td>
<td>20001</td>
</tr>
<tr>
<td>10 points of the piecewise linear $F_k$ vs. $R/t$ curve</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.1-1 Format for File CRKDAT for HEX Coll Problem (Cont'd)
Description of material data

<table>
<thead>
<tr>
<th>6</th>
<th>Number of segments in σε vs. ε curve</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>σ₁ ε₁,  ε₁</td>
</tr>
<tr>
<td></td>
<td>σ₂ ε₂,  ε₂</td>
</tr>
<tr>
<td></td>
<td>σ₃ ε₃,  ε₃</td>
</tr>
<tr>
<td></td>
<td>σ₄ ε₄,  ε₄</td>
</tr>
<tr>
<td></td>
<td>σ₅ ε₅,  ε₅</td>
</tr>
<tr>
<td></td>
<td>σ₆ ε₆,  ε₆</td>
</tr>
<tr>
<td>21.95</td>
<td>0.001</td>
</tr>
<tr>
<td>55.77</td>
<td>0.002</td>
</tr>
<tr>
<td>144.85</td>
<td>0.005</td>
</tr>
<tr>
<td>322.73</td>
<td>0.010</td>
</tr>
<tr>
<td>1945.90</td>
<td>0.050</td>
</tr>
<tr>
<td>50688.0</td>
<td>0.660</td>
</tr>
</tbody>
</table>

'400 F 316L WELDED FROM RkD'

Materials information: yield strength, critical stress intensity factor, number of data divisions, and regression option

| 27000 | 80.0 | 2 | 4 |

Threshold stress intensity factor range model parameters: ΔK<sub>TH0</sub>, C₀, d

| 4.0317 | 1.070 | 0.16327 |

Materials information for first data division: number of points in data division and stress ratio

| 16 | 0.90 |

da/dN vs. ΔK data for division 1

| 9.183E-10 | 2.56 |
| 1.138E-8 | 2.69 |
| 3.362E-8 | 2.82 |
| 8.473E-8 | 3.00 |
| 4.408E-8 | 3.33 |
| 5.838E-8 | 3.53 |
| 5.679E-8 | 3.74 |
| 7.220E-8 | 3.95 |
| 8.202E-8 | 4.18 |
| 7.440E-8 | 4.42 |
| 9.028E-8 | 4.67 |
| 1.133E-7 | 4.94 |
| 1.533E-7 | 5.22 |
| 1.629E-7 | 5.51 |
| 1.727E-7 | 5.81 |
| 2.321E-7 | 5.99 |

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem (Cont'd)
Materials information for second data division: number of points in data division and stress ratio

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>0.70</td>
</tr>
</tbody>
</table>

da/dN vs. ΔK data for division 2

<table>
<thead>
<tr>
<th>da/dN (m/m)</th>
<th>ΔK (m/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.661E-9</td>
<td>3.58</td>
</tr>
<tr>
<td>2.469E-8</td>
<td>3.80</td>
</tr>
<tr>
<td>1.387E-7</td>
<td>4.49</td>
</tr>
<tr>
<td>1.162E-7</td>
<td>4.88</td>
</tr>
<tr>
<td>1.631E-7</td>
<td>5.28</td>
</tr>
<tr>
<td>1.539E-7</td>
<td>5.74</td>
</tr>
<tr>
<td>1.562E-7</td>
<td>6.24</td>
</tr>
<tr>
<td>1.839E-7</td>
<td>6.77</td>
</tr>
<tr>
<td>2.089E-7</td>
<td>7.35</td>
</tr>
<tr>
<td>3.497E-7</td>
<td>7.99</td>
</tr>
<tr>
<td>2.949E-7</td>
<td>9.37</td>
</tr>
<tr>
<td>3.848E-7</td>
<td>10.15</td>
</tr>
<tr>
<td>6.968E-7</td>
<td>11.91</td>
</tr>
<tr>
<td>8.980E-7</td>
<td>12.87</td>
</tr>
<tr>
<td>1.111E-6</td>
<td>13.89</td>
</tr>
<tr>
<td>1.380E-6</td>
<td>15.00</td>
</tr>
<tr>
<td>2.790E-6</td>
<td>17.49</td>
</tr>
<tr>
<td>3.901E-6</td>
<td>18.17</td>
</tr>
</tbody>
</table>

Figure 6.1-1 Format for File CRKDAT for HEX Coil
Problem (Cont’d)
<table>
<thead>
<tr>
<th>Problem type</th>
<th>Crack growth model type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Random number seed</th>
<th>Output dump controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>675</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inner loop size</th>
<th>Outer loop size</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Growth retardation switch (on)</th>
<th>Neuber's rule switch (off)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of B-lives</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

**Decimal equivalent of percentages for B-lives**

| 0.0001 | 0.0005 | 0.001 | 0.005 | 0.01   |

**Beta distribution information for plate width**

| 0.027  | 0.033  | 0.50   | 0.50   | 0.0    | 0.0    |

**Beta distribution information for initial crack size**

| 0.009  | 0.011  | 0.5    | 0.5    | 0.0    | 0.0    |

<table>
<thead>
<tr>
<th>Narrow-band random load scale factor</th>
<th>Sinusoidal load scale factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00 2.00 0.15 1.00</td>
<td>2.00 2.00 0.20 1.00</td>
</tr>
</tbody>
</table>

**Uniform distribution bounds for static stress analysis accuracy factor**

| 0.90 | 1.10 |

**Uniform distribution bounds for dynamic stress analysis accuracy factor**

| 0.80 | 1.20 |

**Uniform distribution bounds for threshold stress intensity factor uncertainty**

| 0.00 | 1.00 |

**Uniform distribution bounds for critical stress intensity factor uncertainty**

| 0.90 | 1.10 |

<table>
<thead>
<tr>
<th>-1.38629</th>
<th>0.95166</th>
</tr>
</thead>
</table>

**Uniform distribution bounds for stress intensity factor calculation accuracy**

| -1.50 | -2.50 |

**Uniform distribution bounds for crack growth calculation accuracy factor**

| 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  |

**Uniform distribution bounds for Forman equation m variation**

| 2.00  | 2.00  | 2.00  | 2.00  | 0.15  | 2.00  | 0.20  |

**Number of dynamic load sources**

| 2      |

**Static stresses: \( \sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{xz}, \sigma_{yz} \)**

| 0.00  | 0.00  | 5000.0 | 0.00  | 0.00  | 0.00  |

**Dynamic stresses: file name, load type, \( \sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{xz}, \sigma_{yz} \)**

| 'NBSZ'  | 1  | 0.00  | 0.00  | 552.34 | 0.00  | 0.00  | 0.00  |
| 'SIN2'  | 2  | 0.00  | 0.00  | 495.86 | 0.00  | 0.00  | 0.00  |

**Figure 6.1-2** Format for File CRKDAT for EXHEX

Problem
2.3 Willenborg retardation model constant
1.0 Reference time history period, T
0.0 Noise filter
6001 Number of points in reference time histories

Description of material data

'C10100 COPPER FROM NASA/JSC'

Yield strength, $K_c$, number of data divisions, and regression option

| 6100 | 100.0 | 1 | 3 |

Threshold stress intensity factor range model parameters: $\Delta K_{THo}$, $C_o$, $d$

| 2.2642 | -2.6912 | -0.55288 |

Regression constraints: $m$, $p$, and $q$

| -2.000 | 0.00 | 0.00 |

Materials information for first data division: number of points in data division and stress ratio

| 8 | 0.20 |

$da/dN$ vs. $\Delta K$ data for division 1

| 5.017E-8 | 3.037 |
| 5.900E-8 | 3.191 |
| 9.798E-8 | 3.607 |
| 1.127E-7 | 3.649 |
| 2.397E-7 | 4.223 |
| 4.069E-7 | 4.864 |
| 5.334E-7 | 5.473 |
| 8.762E-7 | 6.109 |

Figure 6.1-2 Format for File CRKDAT for EXHEX
Problem (Cont'd)
Analysis Parameters Block

KPROB

[INT]

Problem type
PROCPRK has the ability to analyze the HEX Coil and the EXHEX. The following integer values control the type of problem.

- \( KPROB = 1 \) analyze the HEX coil problem
- \( KPROB = 2 \) analyze EXHEX problem

KGROW

[INT]

Crack growth type
The parameter \( m \) in the Forman equation may be fixed at the mean value from the regression or may vary between MVARA and MVARB. Controls the type of \( m \) parameter variation to be included in the Forman crack growth Equation 2-7.

- \( KGROW = 1 \) no \( m \) variation will be included
- \( KGROW = 2 \) allows Uniform variation in \( m \)

RAND

[DRE]

Random number seed
Needed by PROCPRK's built-in random number generator.
**IOUT**

[INT]

Output dump controller

PROCRK has the ability to write intermediate calculations to file IOUTPR. The following integer values control the "dump" of PROCRK's calculations.

- **IOUT = 0**  no intermediate calculation output
- **IOUT = 15**  driver sampling and driver transformation calculations
- **IOUT = 20**  crack growth calculations
- **IOUT = 25**  stress calculations
- **IOUT = 30**  rainflow cycle counting

**NLIFE**

[INT]

Inner loop number

Size of the inner loop of the Monte Carlo (MC) simulation. A positive value is required.

**NHYPER**

[INT]

Outer loop number

Size of the outer loop of the MC simulation. The program requires a positive value.

**IRET**

[INT]

Crack growth retardation switch

Switch to invoke the Willenborg retardation model in the crack growth calculations. The following integer values control the retardation option.

- **IRET = 0**  no growth retardation
- **IRET = 1**  include growth retardation

**INEUB**

[INT]

Neuber's stress calculation switch

Switch to use the Neuber's rule to calculate an equivalent mean stress. The following integer values control the Neuber's rule option.
INEUB = 0  no Neuber's equivalent mean stress calculation  
INEUB = 1  include Neuber's equivalent mean stress calculation

NBLIFE
[INT]

Number of B-lives
The number of B-lives to be provided from the simulated distribution of life. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percentage; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%. NBLIFE must be non-negative and cannot exceed 10.

BLFPER(1)  BLFPER(2) ...  BLFPER(NBLIFE)
[RE]  [RE]  [RE]

B-life percentages
The decimal equivalent of the percentages at which the B-lives are required; e.g., if the B.1 life is desired, then BLFPER = 0.001. A total of NBLIFE percentages must be provided. The percentage cannot exceed 1% (BLFPER ≤ 0.01).

Driver Information Block

WOFFA  WOFFB  WOFFR1  WOFFR2  WOFFT1  WOFFT2
[RE]  [RE]  [RE]  [RE]  [RE]  [RE]

WOFFC  WOFFD  WOFFR3  WOFFR4  WOFFT3  WOFFT4
[RE]  [RE]  [RE]  [RE]  [RE]  [RE]

WOFFE
[RE]

Beta distribution on weld offset information
$W_{\text{OFF}}$ in Equation 2-3 is the weld offset. It is required for the HEX coil problem (KPROB = 1). It may be characterized by two Beta probability distributions. The first two lines are the two Beta distributions, one per line. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for $W_{\text{OFF}}$. The next two parameters are the lower and upper bounds for a Uniform distribution on $p$. Similarly, the last two parameters
The first line is to describe a Uniform distribution on $\theta$. The third line is the decimal equivalent percentage weight for the first Beta distribution, and it must be between 0.00 and 1.00.

<table>
<thead>
<tr>
<th>WOFFA</th>
<th>lower bound of the first Beta distribution on $W_{OFF}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WOFFB</td>
<td>upper bound of the first Beta distribution on $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFR1</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the first Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFR2</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the first Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFT1</td>
<td>Uniform distribution lower bound of parameter $\theta$ in the first Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFT2</td>
<td>Uniform distribution upper bound of parameter $\theta$ in the first Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFC</td>
<td>lower bound of the second Beta distribution on $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFD</td>
<td>upper bound of the second Beta distribution on $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFR3</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the second Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFR4</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the second Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFT3</td>
<td>Uniform distribution lower bound of parameter $\theta$ in the second Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFT4</td>
<td>Uniform distribution upper bound of parameter $\theta$ in the second Beta distribution of $W_{OFF}$</td>
</tr>
<tr>
<td>WOFFE</td>
<td>decimal equivalent percentage weight occurring in the first Beta distribution of the weld offset, $W_{OFF}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDIAA</th>
<th>INDIAB</th>
<th>INDIR1</th>
<th>INDIR2</th>
<th>INDIT1</th>
<th>INDIT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
</tr>
</tbody>
</table>

Beta distribution on duct inside diameter information $D_i$ (in.), the duct inside diameter. It is required for the HEX coil problem ($KPROB = 1$). It is used to calculate $R_i$ in Equation 2-1 and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the duct inside diameter. The next two parameters are the lower and upper bounds for a Uniform distribution on $\rho$. Similarly, the last two parameters describe a Uniform distribution on $\theta$.

| INDIAA | lower bound of the Beta distribution on $D_i$ |
INDIAB upper bound of the Beta distribution on $D_i$  
INDIR1 Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $D_i$  
INDIR2 Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $D_i$  
INDIT1 Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $D_i$  
INDIT2 Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $D_i$  

THICA  THICB  THICR1  THICR2  THICT1  THICT2  
[RE]  [RE]  [RE]  [RE]  [RE]  [RE]

Beta distribution on wall thickness information  
$t$ (in.), the duct wall thickness. It is required for the HEX coil problem ($KPROB = 1$). It is used to calculate the area and calculate $R_o$ in Equation 2-1 and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the wall thickness. The next two parameters are the lower and upper bounds for a Uniform distribution on $\rho$. Similarly, the last two parameters describe a Uniform distribution on $\theta$.  

THICA lower bound of the Beta distribution on $t$  
THICB upper bound of the Beta distribution on $t$  
THICR1 Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $t$  
THICR2 Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $t$  
THICT1 Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $t$  
THICT2 Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $t$  

AOCA  AOCB  AOCR1  AOCR2  AOCT1  AOCT2  
[RE]  [RE]  [RE]  [RE]  [RE]  [RE]

Beta distribution on initial crack aspect ratio information  
a/c, the initial aspect ratio of an elliptic crack. It is required for the HEX coil problem ($KPROB = 1$). It is used to calculate the initial half crack length $c_i$ given the initial
crack depth \( a \), and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the aspect ratio. The next two parameters are the lower and upper bounds for a Uniform distribution on \( \rho \). Similarly, the last two parameters describe a Uniform distribution on \( \theta \).

\[
\begin{align*}
\text{AOCA} & \quad \text{lower bound of Beta distribution on } a/c \\
\text{AOCB} & \quad \text{upper bound of Beta distribution on } a/c \\
\text{AOCR1} & \quad \text{Uniform distribution lower bound of parameter } \rho \text{ in the Beta distribution of } a/c \\
\text{AOCR2} & \quad \text{Uniform distribution upper bound of parameter } \rho \text{ in the Beta distribution of } a/c \\
\text{AOCT1} & \quad \text{Uniform distribution lower bound of parameter } \theta \text{ in the Beta distribution of } a/c \\
\text{AOCT2} & \quad \text{Uniform distribution upper bound of parameter } \theta \text{ in the Beta distribution of } a/c
\end{align*}
\]

\[
\begin{align*}
\text{WITHA} & \quad \text{lower bound of the Beta distribution on } W \\
\text{WITHB} & \quad \text{upper bound of the Beta distribution on } W \\
\text{WITHR1} & \quad \text{Uniform distribution lower bound of parameter } \rho \text{ in the Beta distribution of } W \\
\text{WITHR2} & \quad \text{Uniform distribution upper bound of parameter } \rho \text{ in the Beta distribution of } W \\
\text{WITHT1} & \quad \text{Uniform distribution lower bound of parameter } \theta \text{ in the Beta distribution of } W \\
\text{WITHT2} & \quad \text{Uniform distribution upper bound of parameter } \theta \text{ in the Beta distribution of } W
\end{align*}
\]

Beta distribution on plate width information

\( W \) (in.), the plate width. It is required for the EXHEX problem (\( \text{KPROB} = 2 \)). It is used to calculate the stress intensity factor coefficients and is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the width. The next two parameters are the lower and upper bounds for a Uniform distribution on \( \rho \). Similarly, the last two parameters describe a Uniform distribution on \( \theta \).
**Beta distribution on initial crack size information**

\( a_i \) (in.), the initial crack depth for an elliptic crack in the HEX coil problem (\( KPROB = 1 \)) or half the crack length for the EXHEX problem (\( KPROB = 2 \)). It is characterized by a Beta probability distribution. See Section 2.1.3.1 in [1] for specifying parameters to define a Beta driver distribution. The first two parameters are the lower and upper bounds, respectively, for the initial crack size. The next two parameters are the lower and upper bounds for a Uniform distribution on \( p \). Similarly, the last two parameters describe a Uniform distribution on \( \theta \).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIA</td>
<td>lower bound of the Beta distribution on ( a_i )</td>
</tr>
<tr>
<td>AIB</td>
<td>upper bound of the Beta distribution on ( a_i )</td>
</tr>
<tr>
<td>AIR1</td>
<td>Uniform distribution lower bound of parameter ( p ) in the Beta distribution of ( a_i )</td>
</tr>
<tr>
<td>AIR2</td>
<td>Uniform distribution upper bound of parameter ( p ) in the Beta distribution of ( a_i )</td>
</tr>
<tr>
<td>AIT1</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the Beta distribution of ( a_i )</td>
</tr>
<tr>
<td>AIT2</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the Beta distribution of ( a_i )</td>
</tr>
</tbody>
</table>

**Distribution on narrow-band random load scale factor information**

This line contains the parameters to define the narrow-band random load scale factor, \( \lambda_{D\, RANDOM} \), in Equation 2-5. See Section 2.1.3.2 in [1] on load scale factors for a detailed description of the parameters \( k \), coefficient of variation \( C \), and strain gage factor \( d \).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAMNA</td>
<td>lower bound of Uniform distribution of ( k ) for the narrow-band random load scale factor</td>
</tr>
<tr>
<td>LAMNB</td>
<td>upper bound of Uniform distribution of ( k ) for the narrow-band random load scale factor</td>
</tr>
<tr>
<td>LAMNC</td>
<td>coefficient of variation ( C ) for the narrow-band random load scale factor</td>
</tr>
<tr>
<td>LAMND</td>
<td>strain gage factor ( d ) for the narrow-band random load scale factor</td>
</tr>
</tbody>
</table>
Distribution on sinusoidal load scale factor information

This line contains the parameters to define the sinusoidal load scale factor, $\lambda_{D_{\text{sinusoidal}}}$ in Equation 2-5. See Section 2.1.3.2 in [1] on load scale factors for a detailed description of the parameters $k$, coefficient of variation $C$, and strain gage factor $d$.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAMSA</td>
<td>Lower bound of Uniform distribution of $k$ for the sinusoidal load scale factor</td>
</tr>
<tr>
<td>LAMSB</td>
<td>Upper bound of Uniform distribution of $k$ for the sinusoidal load scale factor</td>
</tr>
<tr>
<td>LAMSC</td>
<td>Coefficient of variation $C$ for the sinusoidal load scale factor</td>
</tr>
<tr>
<td>LAMSD</td>
<td>Strain gage factor $d$ for the sinusoidal load scale factor</td>
</tr>
</tbody>
</table>

Normal distribution on inner wall temperature information

$T_i$ (°R), the inner wall temperature. It is required for the HEX coil problem ($KPROB$ = 1). It is used to calculate the temperature difference across the wall of the duct, $\Delta T$ (°R) in Equation 2-2, and is characterized by a Normal distribution.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMUA</td>
<td>Uniform distribution lower bound of parameter $\mu$ in the Normal distribution of $T_i$</td>
</tr>
<tr>
<td>TIMUB</td>
<td>Uniform distribution upper bound of parameter $\mu$ in the Normal distribution of $T_i$</td>
</tr>
<tr>
<td>TISIGA</td>
<td>Uniform distribution lower bound of parameter $\sigma$ in the Normal distribution of $T_i$</td>
</tr>
<tr>
<td>TISIGB</td>
<td>Uniform distribution upper bound of parameter $\sigma$ in the Normal distribution of $T_i$</td>
</tr>
</tbody>
</table>

6 - 18
Normal distribution on outer wall temperature information
$T_o \, (^\circ R)$, the outer wall temperature. It is required for the HEX coil problem ($KPROB = 1$). It is used to calculate the temperature difference across the wall of the duct, $\Delta T \, (^\circ R)$ in Equation 2-2, and is characterized by a Normal distribution.

- **TOMUA**: Uniform distribution lower bound of parameter $\mu$ in the Normal distribution of $T_o$
- **TOMUB**: Uniform distribution upper bound of parameter $\mu$ in the Normal distribution of $T_o$
- **TOSIGA**: Uniform distribution lower bound of parameter $\sigma$ in the Normal distribution of $T_o$
- **TOSIGB**: Uniform distribution upper bound of parameter $\sigma$ in the Normal distribution of $T_o$

Normal distribution on internal pressure information
$P_i \, (psi)$ in Equation 2-1. It is required for the HEX coil problem ($KPROB = 1$). This is the inner wall pressure, and it is characterized by a Normal distribution.

- **PCMUA**: Uniform distribution lower bound of parameter $\mu$ in the Normal distribution of $P_i$
- **PCMUB**: Uniform distribution upper bound of parameter $\mu$ in the Normal distribution of $P_i$
- **PCSlGA**: Uniform distribution lower bound of parameter $\sigma$ in the Normal distribution of $P_i$
- **PCSlGB**: Uniform distribution upper bound of parameter $\sigma$ in the Normal distribution of $P_i$

Weld offset stress accuracy factor Uniform distribution information
$\lambda_{OFF}$ in Equation 2-3. It is required for the HEX coil problem ($KPROB = 1$). This is the weld offset stress concentration accuracy factor, and it is characterized by a Uniform distribution.

- **LAMWA**: Uniform distribution lower bound of $\lambda_{OFF}$
- **LAMWB**: Uniform distribution upper bound of $\lambda_{OFF}$
Aerodynamic load scale factor distribution information: $\lambda_{D_{aero}}$ in Equation 2-5. It is required for the HEX coil problem ($\text{KPROB} = 1$). This is the aerodynamic load scale factor, and it is characterized by a Uniform distribution.

- AERDA: Uniform distribution lower bound of aerodynamic load scale factor
- AERDB: Uniform distribution upper bound of aerodynamic load scale factor

Aerostatic load scale factor distribution information: $\lambda_{S_{aero}}$ in Equation 2-5. It is required for the HEX coil problem ($\text{KPROB} = 1$). This is the aerostatic load scale factor, and it is characterized by a Uniform distribution.

- AERSA: Uniform distribution lower bound of aerostatic load scale factor
- AERSB: Uniform distribution upper bound of aerostatic load scale factor

Aeroloads stress analysis accuracy factor Uniform distribution information: $\lambda_{A_{E_{ERO}}}$ in Equation 2-5. It is required for the HEX coil problem ($\text{KPROB} = 1$). This is the aeroloads stress analysis accuracy factor, and it is characterized by a Uniform distribution.

- ASTRA: Uniform distribution lower bound of aeroloads stress analysis accuracy factor
- ASTRB: Uniform distribution upper bound of aeroloads stress analysis accuracy factor

Static stress analysis accuracy factor Uniform distribution information: $\lambda_{ST_{str}}$ in Equation 2-5. This is the static stress analysis accuracy factor, and it is characterized by a Uniform distribution. It is required for the EXHEX channel problem ($\text{KPROB} = 2$).
Dynamic stress analysis accuracy factor Uniform distribution information
\( \lambda_{\text{DYN}} \) in Equation 2-5. This is the dynamic stress analysis accuracy factor, and it is characterized by a Uniform distribution.

Neuber's Rule accuracy factor Uniform distribution information
It is required for the HEX Coil problem (\( K\text{PROB} = 1 \)) when \( \text{INEUB} = 1 \). This is the Neuber's Rule accuracy factor, \( \lambda_{\text{neu}} \), and it is characterized by a Uniform distribution. Neuber's Rule is described in Section 2.2.1.4 of [1].

Threshold stress intensity factor uncertainty Uniform distribution information
\( \lambda_{\text{Kth}} \) in Equation 2-8. This is the threshold stress intensity factor range accuracy factor, and it is characterized by a Uniform distribution.
Critical stress intensity factor uncertainty Uniform distribution information
\( \lambda_{Kc} \) in Equation 2-8. This is the critical stress intensity factor uncertainty, and it is characterized by a Uniform distribution.

- **LAMKCA**: Uniform distribution lower bound of critical stress intensity factor uncertainty
- **LAMKCB**: Uniform distribution upper bound of critical stress intensity factor uncertainty

Stress intensity factor calculation accuracy factor Uniform distribution information
This line contains the Uniform distribution bounds for the stress intensity factor calculation accuracy factor, \( \lambda_{sir} \).

- **KLAMA**: Uniform distribution lower bound on stress intensity factor calculation accuracy factor
- **KLAMB**: Uniform distribution upper bound on stress intensity factor calculation accuracy factor

Crack growth calculation accuracy factor distribution information
This line contains the Uniform distribution bounds in \( \log_{e} \) space for the crack growth calculation accuracy factor, \( \lambda_{gro} \), in Equation 2-18.

- **LAMGRA**: lower bound on crack growth calculation accuracy factor
- **LAMGRB**: upper bound on crack growth calculation accuracy factor

Forman equation parameter \( m \) Uniform distribution information
This line contains the Uniform distribution bounds for the Forman equation parameter \( m \) in Equation 2-7. This is required if \( KGROW = 2 \).
Load and Stress Block
The input for loads and stresses for the HEX coil problem \((KPROB = 1)\) and EXHEX problem \((KPROB = 2)\) are different. For the HEX coil problem the beam-end forces (axial force, moments, and shear forces) from a node in a beam finite element mesh were used. For the EXHEX channel the stress components \((\sigma_x, \sigma_y, \text{etc.})\) from a node in a three-dimensional finite element mesh were used.

**NLOAD**

[INT]

Number of dynamic loads
Total number of dynamic or time-varying loads. **NLOAD** cannot exceed 16.

<table>
<thead>
<tr>
<th>PSTAT</th>
<th>TSTAT</th>
<th>MSTAT(1)</th>
<th>MSTAT(2)</th>
<th>VSTAT(1)</th>
<th>VSTAT(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
<td>[RE]</td>
</tr>
</tbody>
</table>

Aerostatic loads
This line contains the six beam-end force components due to aerostatic loads. It is required for the HEX coil problem \((KPROB = 1)\). \(^1\)

- **PSTAT** \(P\) (lbs) in Equation 2-1, the static axial load component
- **TSTAT** \(M_x\) (in.-lbs), the static torsional load component
- **MSTAT(1)** \(M_y\) (in.-lbs) in Equation 2-1, the static moment load component about the y-axis
- **MSTAT(2)** \(M_z\) (in.-lbs) in Equation 2-1, the static moment load component about the z-axis
- **VSTAT(1)** \(V_y\) (lbs), the static shear load component along the y-axis
- **VSTAT(2)** \(V_z\) (lbs), the static shear load component along the z-axis

---

\(^1\) PROCRK does not require \(M_y, V_y, \text{and} V_z\). Nevertheless, placeholders for these parameters must be included as the crack growth model uses routines M4L1 and M4L2 without modifications.
LDNAME(I)   TYPE(I)   P(I)   T(I)   M(1,I)   M(2,I)   V(1,I)   V(2,I)  
[CHR]   [INT]   [RE]   [RE]   [RE]   [RE]   [RE]   [RE]   

Dynamic loads
This line contains the dynamic load file names, load types, and the six components of the beam-end force magnitudes. It is required for the HEX coil problem (KPROB = 1). A total of NLOAD lines must be specified (i.e., the value of I goes from 1 to NLOAD).²

LDNAME(I) File name containing the reference time history for load I. The file name cannot be more than six characters long and must be enclosed by single quotes.

TYPE(I) Load-type of load I, used to assign the appropriate load scale factor
TYPE(I) = 1 Narrow-band random load
TYPE(I) = 2 Sinusoidal load
TYPE(I) = 3 Aerodynamic load

P(I) $P$ (lbs) in Equation 2-1, the dynamic axial load magnitude for load I

T(I) $M_x$ (in.-lbs), the dynamic torsional load magnitude for load I

M(1,I) $M_y$ (in.-lbs) in Equation 2-1, the dynamic moment load magnitude about the y-axis for load I

M(2,I) $M_z$ (in.-lbs) in Equation 2-1, the dynamic moment load magnitude about the z-axis for load I

V(1,I) $V_y$ (lbs), the dynamic shear load magnitude along the y-axis for load I

V(2,I) $V_z$ (lbs), the dynamic shear load magnitude along the z-axis for load I

SXST   SYST   SZST   SXST   SXYZT   SXZST   SYZST  
[RE]   [RE]   [RE]   [RE]   [RE]   [RE]   [RE]   

Static stresses
This line contains the six stress components due to static loads. It is required for the EXHEX problem (KPROB = 2).

SXST $\sigma_x$ (psi), due to static loads

SYST $\sigma_y$ (psi), due to static loads

² PROCKR does not require $M_x$, $V_y$, and $V_z$. Nevertheless, placeholders for these parameters must be included as the crack growth model uses routines M4L1 and M4L2 without modifications.
Dynamic stresses
This line contains the dynamic load file names, load types, and the six stress component magnitudes. It is required for the EXHEX problem (KPROB = 2). A total of NLOAD lines must be specified (i.e., the value of I goes from 1 to NLOAD).

LDNAME(I)  TYPE(I)  SX(I)  SY(I)  SZ(I)  SXY(I)  SXZ(I)  SYZ(I)
[CHR]  [INT]  [RE]  [RE]  [RE]  [RE]  [RE]  [RE]

File name containing the reference time history for load source I. The file name cannot be more than six characters long and must be enclosed by single quotes.

Load-type of load I, used to assign the appropriate load scale factor
TYPE(I) = 1 Narrow-band random load
TYPE(I) = 2 Sinusoidal load

sx(I)  σx (psi), due to dynamic load source I
sy(I)  σy (psi), due to dynamic load source I
sz(I)  σz (psi), due to dynamic load source I
sxy(I)  σxy (psi), due to dynamic load source I
sxz(I)  σxz (psi), due to dynamic load source I
syz(I)  σyz (psi), due to dynamic load source I

External pressure
p_o (psi) in Equation 2-1. This is the outer wall pressure. It is required for the HEX coil problem (KPROB = 1).

Critical location
Critical location of interest on the duct wall. It is required for the HEX coil problem (KPROB = 1).
LOCAT = 1   outer wall
LOCAT = 2   inner wall

ANGLE
(RE)

Critical angle
\( \phi \) (degrees) in Equation 2-1. This is the angle measured counterclockwise from the Z-direction to the critical circumferential location of the duct. It is required for the HEX coil problem (KPROB = 1).

RSO
(RE)

Willenborg retardation model constant
\( RSO \) in Equation 2-13. This is the Willenborg retardation model constant.

PERIOD
(RE)

Period
\( T \) (sec) in Equation 2-18. This is the period of the reference time histories, and it is required so that life may be provided in seconds.

TRUNC
(RE)

Noise filter
Value (psi) used to filter out the insignificant cycles in the composite stress-time history during rainflow cycle counting.

NRAN
(INT)

Number of history points
Number of points in the reference time history files for the dynamic loads. NRAN cannot exceed 20,000.
EM  COEXP  NU
[RE]  [RE]  [RE]

Materials information
This line contains the elastic modulus, coefficient of thermal expansion, and Poisson's ratio. This line is required for the HEX coil problem (KPROB = 1).

EM  \(E \text{ (psi)}\) in Equation 2-2, Young's modulus of elasticity
COEXP  \(\alpha\ (°R)\) in Equation 2-2, the coefficient of thermal expansion
NU  \(\nu\) in Equation 2-2, the materials Poisson's ratio

FK(I)  RT(I)
[RE]  [RE]

\(F_k\) versus \(R/t\) curve
\(F_k\) versus \(R/t\) points for each segment of the curve are used by Equation 2-3 in the weld offset eccentricity stress concentration calculations. It is required for the HEX coil problem (KPROB = 1). A block of 10 segments must be provided (i.e., the value of I goes from 1 to 10). Both FK and RT must be positive and increase with increasing I (i.e., I = 1 is the lower bound of the first segment, and I = 10 is the upper bound of the last segment).

FK(I)  \(F_k(R/t)\) value
RT(I)  \(R/t\) value

NUMSEG
[INT]

Number of segments
The number of piecewise linear segments in the stress-strain versus strain curve provided. It is required for the HEX coil problem (KPROB = 1) when INEUB = 1.

SE(J)  E(J)
[RE]  [RE]

Stress-strain versus strain curve
\(\sigma\varepsilon\) versus \(\varepsilon\) points for each segment of the \(\sigma\) vs. \(\varepsilon\) curve are used in the Neuber's Rule calculations. It is required for the HEX coil problem (KPROB = 1) when INEUB = 1. A block of NUMSEG lines must be provided (i.e., the value of J goes
from 1 to \text{NUMSEG}). Both \text{SE} and \text{E} must be positive and increase with increasing \text{J} as \text{PROCRK} assumes that the \text{J} = 0 point is at the origin.

\begin{align*}
\text{SE}(\text{J}) & \quad \text{value of the product of stress and strain, } \sigma \epsilon, \text{ at the upper end of the } \\
\text{Jth segment of the stress-strain versus strain curve} \\
\text{E}(\text{J}) & \quad \text{value of the strain } \epsilon \text{ at the upper end of the Jth segment of the stress-strain versus strain curve}
\end{align*}

\textbf{Materials Information Block}

\textbf{DESCRP} [CHR]

Description of material data
Name and test environment for the material data. This is a character string no more than 40 characters long, enclosed by single quotes.

\begin{tabular}{|c|c|c|c|}
\hline
\text{FTY} & \text{KC} & \text{NDIV} & \text{IREGOP} \\
\hline
\text{RE} & \text{RE} & \text{INT} & \text{INT} \\
\hline
\end{tabular}

\textbf{Materials information}
Yield strength, critical stress intensity factor, number of divisions of data, and regression option. The data in each division must have the same stress ratio but data with the same stress ratios may be assigned to different divisions if desired (e.g., from different tests). \text{NDIV} cannot exceed ten.

\begin{align*}
\text{FTY} & \quad \text{yield strength (psi)} \\
\text{KC} & \quad \text{critical stress intensity factor (ksi V m \text{\textsuperscript{-0.5}})} \\
\text{NDIV} & \quad \text{number of data divisions for the material data} \\
\text{IREGOP} & \quad \text{regression option to fit the generalized Forman Equation 2.7}\textsuperscript{\text{3}} \\
\text{IREGOP} & = 0 \quad \text{fix } p \text{ regress for } C, n, m, q \\
\text{IREGOP} & = 1 \quad \text{fix } m, p \text{ regress for } C, n, q \\
\text{IREGOP} & = 2 \quad \text{fix } q, p \text{ regress for } C, n, m \\
\text{IREGOP} & = 3 \quad \text{fix } m, q, p \text{ regress for } C, n \\
\text{IREGOP} & = 4 \quad \text{regress for } C, n, m, q, p \\
\end{align*}

\textsuperscript{3} When \text{KGROW} = 1, the selected value of \text{m} will supercede the value obtained from the regression.
Threshold stress intensity factor range model information
The parameters for the threshold stress intensity factor range vs. stress ratio model given by Equation 2-10.

\[ \Delta K_{THO} \]

DKTHO stress intensity factor range, \( \Delta K_{THO} \) at \( R = 0 \) (ksi√in.)

CO empirical model constant \( C_o \)

DEE empirical model exponent \( d \)

\( p \) constraint for IREGOP = 0
Parameter \( p \) in the generalized Forman Equation 2-7. This is required when IREGOP = 0.

\( m, p \) constraint for IREGOP = 1
Parameters \( m \) and \( p \) in the generalized Forman Equation 2-7. This is required when IREGOP = 1.

\( q, p \) constraint for IREGOP = 2
Parameters \( q \) and \( p \) in the generalized Forman Equation 2-7. This is required when IREGOP = 2.

\( m, q, p \) constraint for IREGOP = 3
Parameters \( m \), \( q \), and \( p \) in the generalized Forman Equation 2-7. This is required when IREGOP = 3.
**NP(I)**  **RDATA(I)**  
[INT]  [RE]

Information for each crack growth rate data division.

Number of points and stress ratio for each data division. This line must be provided for each data division I. **NP(I)** for each division cannot exceed two hundred.

- **NP(I)**  number of data points in the division
- **RDATA(I)**  stress ratio for the data in the division

**DADN(I,J)**  **DELK(I,J)**  
[RE]  [RE]

Crack growth $da/dN$ vs. $\Delta K$ data

Crack growth rate versus stress intensity factor range data points for each data division. A block of **NP(I)** lines must be specified (i.e., the value of J goes from 1 to **NP(I)**). This block must be provided for each data division (i.e., the value of I goes from 1 to **NDIV**).

- **DADN(I,J)**  crack growth rate $da/dN$ (in./cycle)
- **DELK(I,J)**  stress intensity factor range $\Delta K$ (ksi$\cdot$$\sqrt{in.}$)

### 6.1.3.2 Reference Time History Files

The data format for the reference time history files is given below. There must be **NLOAD** files with the same names, as specified by **LDNAME(I)** in file **CRKDAT**. Reference time histories are typically generated by program **NBSIN** described in Sections 4.5, 6.6, and 7.7 of [1].

**STRHIS(I,J)**  
[RE]

The points of the Ith reference history

The points of the reference time history specified by **LDNAME(I)**. The data is entered one point per line for J = 1, ..., **NRAN**.

### 6.1.4 Options and Capabilities

**PROCRRK** is a Monte Carlo simulation program which generates a sequence of component lives for a particular failure mode, where life is defined as the accumulated operating time at failure. The simulation has a double-loop structure with **NHYPER** outer loops and **NLIFE** inner loops. The simulation size is dependent on
the failure probability at which a life estimate is desired and the precision desired. For the HEX coil and EXHEX applications, single-loop runs with \texttt{NHYPER = 10,000} and \texttt{NLIFE = 1} were used to characterize component reliability, and single-loop runs with \texttt{NHYPER = 1000} and \texttt{NLIFE = 1} were used for the marginal analysis to assess the importance of drivers.

During a run, it may be desirable to "hold" a driver at a fixed value. This may be the nominal or median value of the driver. This is done for drivers with a Beta or a Uniform distribution by merely specifying both the upper and lower bounds to be the desired value. For drivers with a Normal distribution, the standard deviation $\sigma$, or coefficient of variation $C$, is set at zero, and the mean, $\mu$, is set at the desired value.

The procedure of holding certain drivers at fixed values while letting the other drivers vary according to their probability distributions may be used for driver variation sensitivity studies. That is, the effect on life of driver variation may be evaluated by letting it vary while holding other drivers at fixed values.

A printout of intermediate calculations in various parts of the program may be obtained via the \texttt{IOUT} option. This output will be printed in the IOUTPR file. It is recommended that such output not be requested when the simulation size is large since the information will be dumped during every simulation loop.

\subsection*{6.1.5 Code Execution Example}

The following example run of the crack growth analysis code PROCRK was carried out with random variation of all drivers for the HPOTP heat exchanger coil small tube outlet (\texttt{KPROB = 1}). In this example run, 1000 lives were simulated (\texttt{NLIFE = 1 times NHYPER = 1000}) with no variation in the Forman constant $m$, \texttt{KGROW = 1}. The Willenborg retardation model and the Neuber's rule to calculate the mean stress are switched on (\texttt{IRET = 1}, \texttt{INEUB = 1}). The B-lives\footnote{A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.} to be provided are B.1, B.5, and B1 (\texttt{NBLIFE = 3}, \texttt{BLFPER(1) = 0.001}, \texttt{BLFPER(2) = 0.005}, \texttt{BLFPER(3) = 0.01}). The user may refer to Section 2.2 for additional information on the engineering analysis and to Section 2.3 for the results of the case study for this component.
Figure 6.1-4 shows the component in detail and the location of the critical weld, designated as \( \Delta \). The external pressure \( P_{CO} \) is 3640 psi. The elastic modulus \( EM \) is \( 2.9 \times 10^7 \), the coefficient of thermal expansion \( COEXP \) is \( 8.8 \times 10^{-6} \), and Poisson's ratio \( NU \) is 0.30 for the material.

The drivers for the crack growth failure of weld 3 are as follows:

<table>
<thead>
<tr>
<th>DRIVER</th>
<th>DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weld Offset</td>
<td>Beta</td>
</tr>
<tr>
<td>Inner Diameter</td>
<td>Beta</td>
</tr>
<tr>
<td>Wall Thickness</td>
<td>Beta</td>
</tr>
<tr>
<td>Initial Crack Aspect Ratio</td>
<td>Beta</td>
</tr>
<tr>
<td>Initial Crack Size</td>
<td>Beta</td>
</tr>
<tr>
<td>Random &amp; Sine Load Scale Factors</td>
<td>Normal</td>
</tr>
<tr>
<td>Flow Conditions ( (T_i, T_o, P_i) )</td>
<td>Normal</td>
</tr>
<tr>
<td>Weld Offset Stress Concentration Accuracy</td>
<td>Uniform</td>
</tr>
<tr>
<td>Aerodynamic &amp; Static Load Scale Factors</td>
<td>Uniform</td>
</tr>
<tr>
<td>Aeroloads &amp; Dynamic Stress Analysis Accuracy Factors</td>
<td>Uniform</td>
</tr>
<tr>
<td>Neuber's Rule Accuracy</td>
<td>Uniform</td>
</tr>
<tr>
<td>Threshold Stress Intensity Uncertainty Factor</td>
<td>Uniform</td>
</tr>
<tr>
<td>Critical Stress Intensity Uncertainty Factor</td>
<td>Uniform</td>
</tr>
<tr>
<td>Stress Intensity Factor Calculation Accuracy</td>
<td>Uniform</td>
</tr>
<tr>
<td>Growth Calculation Accuracy</td>
<td>Uniform</td>
</tr>
</tbody>
</table>
The rationale for the specification of the driver distributions is given in Section 2.3. The initial crack size was held at 0.005" by fixing the upper and lower bounds of the distribution at \( AIA = AIB = 0.005 \). Also, the weld offset was held at 6% by fixing \( WOFFA = WOFFB = 0.06 \). The threshold stress intensity factor range accuracy was set to \( LAMKHA = LAMKHB = 0 \) resulting in a zero threshold for the crack growth analysis. The critical stress intensity factor accuracy was set to \( LAMKCA = LAMKCB = 1 \).

In addition to the static loads, there were one narrow-band random load, one sinusoidal load, and one aerodynamic load. The three dynamic loads (NLOAD = 3) used here are a subset of the loads for this component. The three reference time histories are in the files named NBM3, SIN1, and AERO1, and the contents of these input files are given below. The reference time histories have five points (NRAN = 5) and represent 0.00025 seconds (PERIOD = 0.00025) of the loading. The reference time histories used for the case studies of the HEX coil small tube outlet given in Section 2.3 consisted of 20,000 points. Shorter histories are used here to permit their inclusion in this example. The critical location is the inner wall (LOCAT = 2) at a circumferential position of ANGLE = 10\(^\circ\).

The material properties used are for welded 316L stainless steel. The yield strength \( FTY = 27,000 \) psi, and the critical stress intensity factor \( KC = 80.0 \) ksi\(\sqrt{in}\). The Willenborg retardation model parameter \( RSO = 2.3 \). The threshold SIF model parameters \( DKTHO = 4.0317 \), \( CO = 1.070 \), and \( DEE = 0.16327 \). Three divisions (NDIV = 3) of \( da/dN \) vs. \( \Delta K \) data, which is a subset of the data used in the case study of this component described in Section 2.3, are provided. The first division has 16 data points generated at a stress ratio \( R = 0.90 \), the second division has 18 data points at a stress ratio \( R = 0.70 \), and the third division has 17 data points at a stress ratio of 0.16. The regression option (IREGOP = 4), which derives all the Forman constants \( C, n, m, p, \) and \( q \), was used. If further explanation of file CRKDAT is required, refer to Section 6.1.3.1 and Figure 6.1-1.

The echo of the input data is in the output file CRKRES. The simulated B-lives are also given for the component. For instance, the B.1 life is \( 1.1 \times 10^5 \) seconds. This value is different from the B.1 life obtained in the case study of this component given in Section 2.3 because the number and size of the reference time histories, crack growth rate data points, and the number of simulation trials have been reduced to facilitate the example run. There are only three time histories with just five points each used here, and therefore they do not properly represent the loads for the HEX coil problem. Also, the \( F_k \) versus \( R/t \) curve is only an example curve.
The IOUTPR file gives an echo of the analysis parameters. The dump parameter IOUT is zero; therefore, no other output is in this file. The LOWLIF file contains the lowest one percent of the 1000 simulation lives.

Input File - CRKDAT

1
1
675
0
1
1000
1
1
3
0.001 0.005 0.010
0.06 0.06 0.00 0.00 0.0 0.0
0.00 0.00 0.00 0.00 0.0 0.0
1.00
0.1885 0.1915 0.50 0.50 0.5  20.
0.0113 0.0157 0.27273 0.27273 0.5 20.
0.200 1.000 0.50 0.50 0.0 0.0
0.005 0.005 0.00 0.00 0.0 0.0
2.00 2.00 0.15 1.00
2.00 2.00 0.20 1.00
486. 666. 29. 56.5
799. 908. 49.5 48.
3808. 4177. 69. 69.
0.80 1.20
0.50 1.50
0.80 1.20
0.90 1.10
0.80 1.20
0.60 1.40
0.00 0.00
1.00 1.00
0.90 1.10
-0.6931 0.557
3
0.00 0.00 -0.07214 0.00 0.00 0.00
'NBH3' 1 0.00 0.00 0.00 0.355475 0.00 0.00
'SIN1' 2 0.027374 0.000451 0.001621 0.082116 0.205288 0.005789
'AERO1' 3 0.00 0.00 0.00 0.07179 0.00 0.00
3640.
2
85.
2.30
0.00025
50.
| 5 | 29000000. | 8.8E-06 | 0.30 |
| 0.615 | 2.00 |
| 0.693 | 4.80 |
| 0.753 | 7.20 |
| 0.813 | 9.60 |
| 0.873 | 12.50 |
| 0.933 | 15.80 |
| 0.993 | 20.00 |
| 1.029 | 24.00 |
| 1.053 | 30.00 |
| 1.053 | 200.00 |
| 6 | 21.95 | 0.001 |
| 55.77 | 0.002 |
| 144.85 | 0.005 |
| 322.73 | 0.010 |
| 1945.90 | 0.050 |
| 50688.0 | 0.660 |

'400F 316L WELDED, FROM Rkd'
27000 80.0 3 4
4.0317 1.070 0.16327
16 0.90
9.183E-10 2.56
1.138E-8 2.69
3.362E-8 2.82
8.473E-8 3.00
4.408E-8 3.33
5.838E-8 3.53
5.679E-8 3.74
7.220E-8 3.95
8.202E-8 4.18
7.440E-8 4.42
9.028E-8 4.67
1.133E-7 4.94
1.533E-7 5.22
1.629E-7 5.51
1.727E-7 5.81
2.321E-7 5.99
18 0.70
4.661E-9 3.58
2.469E-8 3.80
1.387E-7 4.49
1.162E-7 4.88
1.631E-7 5.28
1.539E-7 5.74
1.562E-7 6.24
1.839E-7 6.77
2.089E-7 7.35
3.497E-7 7.99

6 - 35
<table>
<thead>
<tr>
<th></th>
<th>2.949E-7</th>
<th>9.37</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.848E-7</td>
<td>10.15</td>
</tr>
<tr>
<td></td>
<td>6.968E-7</td>
<td>11.91</td>
</tr>
<tr>
<td></td>
<td>8.980E-7</td>
<td>12.87</td>
</tr>
<tr>
<td></td>
<td>1.111E-6</td>
<td>13.89</td>
</tr>
<tr>
<td></td>
<td>1.380E-6</td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td>2.790E-6</td>
<td>17.49</td>
</tr>
<tr>
<td></td>
<td>3.901E-6</td>
<td>18.17</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>1.775E-7</td>
<td>9.10</td>
</tr>
<tr>
<td></td>
<td>1.969E-7</td>
<td>9.91</td>
</tr>
<tr>
<td></td>
<td>2.454E-7</td>
<td>10.79</td>
</tr>
<tr>
<td></td>
<td>2.543E-7</td>
<td>11.78</td>
</tr>
<tr>
<td></td>
<td>4.050E-7</td>
<td>12.83</td>
</tr>
<tr>
<td></td>
<td>5.355E-7</td>
<td>13.96</td>
</tr>
<tr>
<td></td>
<td>7.369E-7</td>
<td>15.20</td>
</tr>
<tr>
<td></td>
<td>1.058E-6</td>
<td>16.53</td>
</tr>
<tr>
<td></td>
<td>2.008E-6</td>
<td>18.00</td>
</tr>
<tr>
<td></td>
<td>2.650E-6</td>
<td>19.56</td>
</tr>
<tr>
<td></td>
<td>4.238E-6</td>
<td>21.24</td>
</tr>
<tr>
<td></td>
<td>5.679E-6</td>
<td>23.11</td>
</tr>
<tr>
<td></td>
<td>8.308E-6</td>
<td>25.07</td>
</tr>
<tr>
<td></td>
<td>9.687E-6</td>
<td>27.33</td>
</tr>
<tr>
<td></td>
<td>1.649E-5</td>
<td>32.96</td>
</tr>
<tr>
<td></td>
<td>2.335E-5</td>
<td>38.56</td>
</tr>
<tr>
<td></td>
<td>3.304E-5</td>
<td>45.07</td>
</tr>
</tbody>
</table>

**Input File - NBM3**

-0.862955457680720
0.981515081918201
1.03346865031769
1.10476309499562
1.32048639932450

**Input File - SIN1**

-0.976676043502130
-0.931062212127054
-0.862522537797772
-0.772744694860142
-0.663939311885647

**Input File - AERO1**

-0.870754448952271
-0.953457959513392

6 - 36
Output File - CRKRES

Copyright (C) 1991, California Institute of Technology. U.S. Government
Sponsorship under NASA Contract NAS7-918 is acknowledged.

**PROCRRK**

**INPUT DATA**

<table>
<thead>
<tr>
<th>DRIVERS</th>
<th>PARAMETER DISTRIBUTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>RHO</strong></td>
</tr>
<tr>
<td>WELD OFFSET (%)</td>
<td>Be(0.06, 0.06) U(0.00000, 0.00000) U(0.0, 0.0)</td>
</tr>
<tr>
<td></td>
<td>Be(0.00, 0.00) U(0.00000, 0.00000) U(0.0, 0.0)</td>
</tr>
<tr>
<td></td>
<td>TEST = 1.00</td>
</tr>
<tr>
<td>INNER DIAMETER</td>
<td>Be(0.1885, 0.1915) U(0.50000, 0.50000) U(0.5, 20.0)</td>
</tr>
<tr>
<td>WALL THICKNESS</td>
<td>Be(0.0113, 0.0157) U(0.27273, 0.27273) U(0.5, 20.0)</td>
</tr>
<tr>
<td>CRACK SHAPE A/C</td>
<td>Be(0.2000, 1.0000) U(0.50000, 0.50000) U(0.0, 0.0)</td>
</tr>
<tr>
<td>CRACK SIZE A</td>
<td>Be(0.0050, 0.0050) U(0.00000, 0.00000) U(0.0, 0.0)</td>
</tr>
<tr>
<td>LAMBDA RANDOM</td>
<td>k: U(2.00000, 2.00000)</td>
</tr>
<tr>
<td></td>
<td>COEFFICIENT OF VARIATION: 0.150</td>
</tr>
<tr>
<td></td>
<td>STRAIN GAGE FACTOR: 1.00000000</td>
</tr>
<tr>
<td>LAMBDA SINE</td>
<td>k: U(2.00000, 2.00000)</td>
</tr>
<tr>
<td></td>
<td>COEFFICIENT OF VARIATION: 0.200</td>
</tr>
<tr>
<td></td>
<td>STRAIN GAGE FACTOR: 1.00000000</td>
</tr>
<tr>
<td>INNER TEMPERATURE</td>
<td>NORMAL: MU(486.0, 666.0) SIGMA(29.0, 56.5)</td>
</tr>
<tr>
<td>OUTER TEMPERATURE</td>
<td>NORMAL: MU(799.0, 908.0) SIGMA(49.5, 48.0)</td>
</tr>
<tr>
<td>INNER PRESSURE</td>
<td>NORMAL: MU(3808.0, 4177.0) SIGMA(69.0, 69.0)</td>
</tr>
</tbody>
</table>
WELD OFFSET K FAC U( 0.80000, 1.20000)
DYN AERO LOAD FAC U( 0.50000, 1.50000)
STAT AERO LOAD FAC U( 0.80000, 1.20000)
AERO STR ANAL FAC U( 0.90000, 1.10000)
DYN STR ANAL FAC U( 0.80000, 1.20000)
NEUBERS RULE U( 0.60000, 1.40000)
LAMBDA Kth U( 0.00000, 0.00000)
LAMBDA Kc U( 1.00000, 1.00000)
K CALC FAC U( 0.90000, 1.10000)
GROWTH CALC FAC U(-0.69310, 0.55700)

LOADS INPUT

<table>
<thead>
<tr>
<th></th>
<th>LOADS</th>
<th>T LOADS</th>
<th>M2 LOADS</th>
<th>M3 LOADS</th>
<th>V2 LOADS</th>
<th>V3 LOADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATIC AERO</td>
<td>0.000000</td>
<td>0.000E+00</td>
<td>-721E-01</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>NBM3</td>
<td>0.000000</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.355E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>SIN1</td>
<td>0.027374</td>
<td>0.451E-03</td>
<td>0.162E-02</td>
<td>0.821E-01</td>
<td>0.205E+00</td>
<td>0.579E-02</td>
</tr>
<tr>
<td>AERO1</td>
<td>0.000000</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.718E-01</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>

MISCELLANEOUS INPUT

EXTERNAL PRESSURE 3640.
ANALYSIS LOCATION 2
ANGLE THETA (DEGREES) 85.0
WILLENBORG OVERLOAD FACTOR 0.23000E+01
STRESS-TIME HISTORY PERIOD 0.00025
STRESS-TIME HISTORY NOISE FILTER 50.0
NUMBER OF TIME-VARYING LOADS 3
NUMBER OF POINTS IN HISTORIES 5

ELASTIC MODULUS 0.290E+08
COEFF OF THERMAL EXPANSION 0.87999997E-05
POISSONS RATIO 0.300

Fk VS. Rt CURVE INPUT

<table>
<thead>
<tr>
<th>Fk</th>
<th>Rt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.62</td>
<td>2.00</td>
</tr>
<tr>
<td>0.69</td>
<td>4.80</td>
</tr>
<tr>
<td>0.75</td>
<td>7.20</td>
</tr>
<tr>
<td>0.81</td>
<td>9.60</td>
</tr>
<tr>
<td>0.87</td>
<td>12.50</td>
</tr>
<tr>
<td>0.93</td>
<td>15.80</td>
</tr>
<tr>
<td>0.99</td>
<td>20.00</td>
</tr>
<tr>
<td>1.03</td>
<td>24.00</td>
</tr>
<tr>
<td>1.05</td>
<td>30.00</td>
</tr>
<tr>
<td>1.05</td>
<td>200.00</td>
</tr>
</tbody>
</table>

STRESS-STRAIN CURVE INPUT

MAXIMUM NUMBER OF SEGMENTS 6

6 - 39
### Stress-Strain Product

<table>
<thead>
<tr>
<th>Stress-Strain Product</th>
<th>Strain Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.95</td>
<td>0.00100</td>
</tr>
<tr>
<td>55.77</td>
<td>0.00200</td>
</tr>
<tr>
<td>144.85</td>
<td>0.00500</td>
</tr>
<tr>
<td>322.73</td>
<td>0.01000</td>
</tr>
<tr>
<td>1945.90</td>
<td>0.05000</td>
</tr>
<tr>
<td>50688.00</td>
<td>0.66000</td>
</tr>
</tbody>
</table>

### Material Input

**Description:** 400F 316L WELDED, FROM Rkd

**Yield Strength:** 27000.

**Critical SIF:** 80.

**Number of Divisions:** 3

**Regression Option:** 4

### Threshold Model Description

- **DKTHO:** 0.40317E+01
- **Co:** 0.10700E+01
- **d:** 0.16327E+00

**Stress Ratio R:** 0.90

<table>
<thead>
<tr>
<th>da/dN</th>
<th>DELK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.91830E-09</td>
<td>0.25600E+01</td>
</tr>
<tr>
<td>0.11380E-07</td>
<td>0.26900E+01</td>
</tr>
<tr>
<td>0.33620E-07</td>
<td>0.28200E+01</td>
</tr>
<tr>
<td>0.84730E-07</td>
<td>0.30000E+01</td>
</tr>
<tr>
<td>0.44080E-07</td>
<td>0.33300E+01</td>
</tr>
<tr>
<td>0.58380E-07</td>
<td>0.35300E+01</td>
</tr>
<tr>
<td>0.56790E-07</td>
<td>0.37400E+01</td>
</tr>
<tr>
<td>Stress Ratio R = 0.70</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>da/dN</td>
<td>DELK</td>
</tr>
<tr>
<td>0.46610E-08</td>
<td>0.35800E+01</td>
</tr>
<tr>
<td>0.24690E-07</td>
<td>0.38000E+01</td>
</tr>
<tr>
<td>0.13870E-06</td>
<td>0.44900E+01</td>
</tr>
<tr>
<td>0.11620E-06</td>
<td>0.48800E+01</td>
</tr>
<tr>
<td>0.16310E-06</td>
<td>0.52800E+01</td>
</tr>
<tr>
<td>0.15390E-06</td>
<td>0.57400E+01</td>
</tr>
<tr>
<td>0.15620E-06</td>
<td>0.62400E+01</td>
</tr>
<tr>
<td>0.18390E-06</td>
<td>0.67700E+01</td>
</tr>
<tr>
<td>0.20890E-06</td>
<td>0.73500E+01</td>
</tr>
<tr>
<td>0.34970E-06</td>
<td>0.79900E+01</td>
</tr>
<tr>
<td>0.29490E-06</td>
<td>0.93700E+01</td>
</tr>
<tr>
<td>0.38480E-06</td>
<td>0.10150E+02</td>
</tr>
<tr>
<td>0.69680E-06</td>
<td>0.11910E+02</td>
</tr>
<tr>
<td>0.89800E-06</td>
<td>0.12870E+02</td>
</tr>
<tr>
<td>0.11110E-05</td>
<td>0.13890E+02</td>
</tr>
<tr>
<td>0.13800E-05</td>
<td>0.15000E+02</td>
</tr>
<tr>
<td>0.27900E-05</td>
<td>0.17490E+02</td>
</tr>
<tr>
<td>0.39010E-05</td>
<td>0.18170E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stress Ratio R = 0.16</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>da/dN</td>
<td>DELK</td>
<td></td>
</tr>
<tr>
<td>0.17750E-06</td>
<td>0.91000E+01</td>
<td></td>
</tr>
<tr>
<td>0.19690E-06</td>
<td>0.99100E+01</td>
<td></td>
</tr>
<tr>
<td>0.24540E-06</td>
<td>0.10790E+02</td>
<td></td>
</tr>
<tr>
<td>0.25430E-06</td>
<td>0.11780E+02</td>
<td></td>
</tr>
<tr>
<td>0.40500E-06</td>
<td>0.12830E+02</td>
<td></td>
</tr>
<tr>
<td>0.53550E-06</td>
<td>0.13960E+02</td>
<td></td>
</tr>
<tr>
<td>0.73690E-06</td>
<td>0.15200E+02</td>
<td></td>
</tr>
<tr>
<td>0.10580E-05</td>
<td>0.16530E+02</td>
<td></td>
</tr>
<tr>
<td>0.20080E-05</td>
<td>0.18000E+02</td>
<td></td>
</tr>
<tr>
<td>0.26500E-05</td>
<td>0.19560E+02</td>
<td></td>
</tr>
<tr>
<td>0.42380E-05</td>
<td>0.21240E+02</td>
<td></td>
</tr>
<tr>
<td>0.56790E-05</td>
<td>0.23110E+02</td>
<td></td>
</tr>
<tr>
<td>0.83080E-05</td>
<td>0.25070E+02</td>
<td></td>
</tr>
</tbody>
</table>
0.96870E-05  0.27330E+02  
0.16490E-04  0.32960E+02  
0.23350E-04  0.38560E+02  
0.33040E-04  0.45070E+02  

REGRESSION OUTCOME

\[
\begin{array}{cccc}
 C & n & m & p & q \\
 0.56708E-11 & 0.25314E+01 & -0.19413E+01 & 0.71522E+00 & -0.81965E+00 \\
\end{array}
\]

SIMULATION OUTPUT

SHORTEST 1% OF CRACK GROWTH LIVES

LIFE

0.30110E+06  
0.37117E+06  
0.42265E+06  
0.44193E+06  
0.44601E+06  
0.49042E+06  
0.49447E+06  
0.49949E+06  
0.50079E+06  
0.50608E+06  

B LIVES: EMPIRICAL

0.00100  0.30110E+06  
0.00500  0.44601E+06  
0.01000  0.50608E+06  

Output File - IOUTPR

PROBLEM TYPE (HEX COIL = 1, EXHEX = 2) = 1
FORMAN EQUATION WITH m (CONST = 1, VARY = 2) = 1
RANDOM NUMBER SEED = 675.000000000000
IOUT - OUTPUT CONTROL VARIABLE = 0

6 - 42
INNER LOOP SIZE = 1
OUTER LOOP SIZE = 1000
RETARDATION SWITCH (0 - NO, 1 - YES) = 1
NEUBER SWITCH (0 - NO, 1 - YES) = 1

Output File - LOWLIF

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.100000E-02</td>
<td>301098.</td>
</tr>
<tr>
<td>2</td>
<td>0.200000E-02</td>
<td>371174.</td>
</tr>
<tr>
<td>3</td>
<td>0.300000E-02</td>
<td>422653.</td>
</tr>
<tr>
<td>4</td>
<td>0.400000E-02</td>
<td>441930.</td>
</tr>
<tr>
<td>5</td>
<td>0.500000E-02</td>
<td>446013.</td>
</tr>
<tr>
<td>6</td>
<td>0.600000E-02</td>
<td>490422.</td>
</tr>
<tr>
<td>7</td>
<td>0.700000E-02</td>
<td>494468.</td>
</tr>
<tr>
<td>8</td>
<td>0.800000E-02</td>
<td>499490.</td>
</tr>
<tr>
<td>9</td>
<td>0.900000E-02</td>
<td>500786.</td>
</tr>
<tr>
<td>10</td>
<td>0.100000E-01</td>
<td>506082.</td>
</tr>
</tbody>
</table>

6.1.6 Error Messages and Possible Remedies

The following messages, when applicable, will appear in file IOUTPR. An error message stating that a limit has been exceeded will require that the user increase those limits, as directed, and reviewing or consulting Section 7.1 is desirable. The messages are listed in alphabetical order for the convenience of the user.

ERROR: CANNOT OPEN FILE, ‘filename’ DOES NOT EXIST
  Fatal  PROCRK attempted to open the indicated file, however the file did not exist. Check the directory for existence of the file and also check file CRKDAT for correct spelling of the filename.

ERROR: INVALID FORMAN EQUATION SPECIFICATION
  Fatal  KGROW can only have integer values of 1 or 2. Check file CRKDAT for the value used.

ERROR: INVALID LOCATION SPECIFICATION
  Fatal  LOCAT can only have the integer value of 1 or 2. Check file CRKDAT for the value used.

ERROR: INVALID NEUBER’S RULE SWITCH SPECIFICATION
  Fatal  INEUB can only have integer values of 0 or 1. Check file CRKDAT for the value used.
ERROR: INVALID PROBLEM TYPE SPECIFICATION
  Fatal  KPROB can only have integer values of 0 or 1. Check file CRKDAT for the value used.

ERROR: INVALID REGRESSION OPTION SPECIFICATION
  Fatal  IREGOP can only have integer values of 0, 1, 2, 3, or 4. Check file CRKDAT for the value used.

ERROR: INVALID RETARDATION SWITCH SPECIFICATION
  Fatal  IRET can only have integer values of 0 or 1. Check file CRKDAT for the value used.

ERROR: LOAD INCORRECTLY TYPED
  Fatal  TYPE(I) can only have the integer value of 1, 2, or 3. Check file CRKDAT for the value used.

ERROR: NUMBER OF GROWTH RATE DATA POINTS PER DIVISION EXCEEDED
  Fatal  The materials characterization model cannot accept more than 200 da/dN vs. ΔK points in any data division. It is suggested that the number of data points in each division be recounted. If more than 200 points is desired, the parameter MAXDAT must be increased. Refer to Section 7.1 for the routines involved.

ERROR: STRESS-TIME HISTORY TOO LARGE
  Fatal  No more than 20,000 points is allowed for a reference time history, and an attempt has been made to use a larger history. Check file CRKDAT for a value of NRAN larger than 20,000.

K GT Kcr AT A = 'A(1)'
  Warning  This is information to the user that the stress intensity factor K exceeded the critical value Kc at crack length A(1), during block growth calculation, for a draw in the simulation.

NO GROWTH AT 'J'th CRACK LENGTH
  Warning  This is information to the user that there was no growth at the Jth crack length, during block growth calculation, for a draw in the simulation.

NO GROWTH IN A DIRECTION AT 'J'th CRACK LENGTH
  Warning  This is information to the user that there was no growth in the 'a' direction at the Jth crack length, during block growth calculation, for a draw in the simulation.
PROGRAM EXECUTION TERMINATED

Fatal  This message is produced by routine TRMNAT and follows all other fatal messages.

6.1.7 Summary of Input/Output Files

Input Files

CRKDAT
This file is opened in PROCRK. It contains all parameters for the run options; driver distributions; engineering analysis parameters; and the materials input, including \(\frac{da}{dN}\) vs. \(\Delta K\) data points.

User Specified
These are the reference time history files and are opened in PROCRK. They contain the time histories generated by program NBSIN.

Output Files

CRKRES
This file is opened in PROCRK. It contains the echo of the information contained in CRKDAT, and provides the simulated failure distribution B-life information.\(^5\)

IOUTPR
This file is opened in PROCRK. It contains information on the particular run that is not echoed to CRKRES and the data dump provided when the variable IOUT is equal to 15 (Monte Carlo simulation and driver transformation calculations), 20 (crack growth calculations), 25 (stress analysis calculations), or 30 (rainflow cycle counting).

LOWLIF
This file is opened in PROCRK. It contains the lowest one percent of the calculated lives used by the software described in Section 4.2 of [1] to calculate \(\alpha, \beta,\) and \(\theta\), the parameters of the Bayesian prior failure distribution.

\(^5\) A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.
Section 6.2
Low Cycle Fatigue Analysis User's Guide

6.2.1 BLDLCF Program

A user's guide for running the low cycle fatigue (LCF) analysis code BLDLCF is given here. The LCF analysis for the blade is discussed in Section 3, the program description and flowcharts are presented in Section 5.2, and the code structure and listing are provided in Section 7.2.

The BLDLCF program was used to analyze the low cycle fatigue failure of the ATD-HPFTP first stage turbine blade. The output of BLDLCF includes the simulated B-lives and a list of the lowest one percent of lives. The list of lives may be used as input to the regression programs of [1], Section 4.2, to compute the parameters of the Bayesian prior failure distribution. This prior distribution and the success/failure data are used as input to the Bayesian updating program BAYES to obtain a posterior failure distribution. Program Bayes is described in Section 4.3 of [1].

6.2.2 How to Use Program BLDLCF

The program BLDLCF is intended to be run in batch (i.e., background) mode. BLDLCF requires two input data files: BLDLCD and RELATD. The materials characterization model portion of the program requires both files for all runs, even when no related S/N data is used. The file BLDLCD contains the analysis control parameters, driver distributions, engineering analysis parameters, and specific and exogenous materials information. The file RELATD contains the related materials information. A complete description of the input data for the BLDLCD and RELATD data files is given in Section 6.2.3.

The results from the BLDLCF program are written to five output files: BLDLCO, RELATO, DUMP, IOUTPR, and LOWLIF. BLDLCO contains the echo of the information in BLDLCD and the results of the simulation. RELATO contains the echo of the information in RELATD on the related materials data. The results of the materials characterization calculations are primarily given in DUMP. These calculations include point and interval estimates for S/N curve parameters m and C, posterior credibility ranges for m, and an estimate of the median S/N curve. File IOUTPR contains an echo of the analysis parameters and, if requested, a dump of intermediate calculations. If the program terminates prematurely, an error message will be printed in the IOUTPR file. A list of error messages and possible remedies for the problems is given in Section 6.2.6. LOWLIF contains the first one percent of the lives of the simulated failure distribution.
6.2.3 Description of Input Data Files

Annotated examples of the complete data file format structure for BLDLCD and RELATD are presented in Figures 6.2-1 and 6.2-2, respectively. The data lines of the input files are given in boxes, with a description of each data line located adjacent to each box. The specific input parameters of Figures 6.2-1 and 6.2-2 are individually defined in Sections 6.2.3.1 and 6.2.3.2. Input parameter values given in Figures 6.2-1 and 6.2-2 are not necessarily those used in the application case study of Section 3.3.

The input data is read by free format statements from files BLDLCD and RELATD. Thus, the numbers may be provided sequentially on a line up to 80 characters in length, with each number separated by a blank character or comma. Each number may also be on a separate line in the file. However, it is recommended that the input format suggested in Figures 6.2-1 and 6.2-2 be followed whenever possible.

6.2.3.1 Input File BLDLCD

The required data for the BLDLCD file is divided into the four blocks shown in Figure 6.2-3: analysis parameters, driver information, load and geometry, and materials information. The analysis parameters block contains the analysis parameters and the keys to select the program options. The driver information block contains the parameters that define the driver distributions. The parametric sensitivity information, the nominal strains, and reference time history are given in the load and geometry block. The materials information block contains the specific material S/N data including the yield and ultimate strengths, strain ratio, the S/N data points, life region boundaries, and materials characterization model parameter constraints.

The input parameters are described below by using the following convention: the input variable names are indicated by **BOLD UPPERCASE** letters; the variable types are specified as character [CHR], integer [INT], real [RE], and double precision real [DRE]; the function of the variable is **underlined** and followed by a description and a list of options, when appropriate; the program and file names are indicated by **UPPERCASE** letters. A consistent set of units is given in parentheses for specifying dimension, load, and strain input parameters. All character strings must be enclosed by 'single quotes'. The user is reminded about the difference between the number "0" and the letter "O" when preparing the input files.
Random number seed
Value of output dump controller
Inner loop size
Outer loop size
Symmetry number
Type of S/N variation
Request for truncated Normal median S/N curve
Controls materials process variation
Type of materials intrinsic variation
Number of B-lives

Decimal equivalent of percentages for B-lives

| 0.0001 | 0.0005 | 0.001 | 0.005 | 0.01 |

Beta distribution on $h_{gas}$ information

| 676. | 2730. | 0.50 | 0.50 | 0.0 | 0.0 |

Beta distribution on $T_{gas}$ information

| 782. | 1982. | 0.50 | 0.50 | 0.0 | 0.0 |

Beta distribution on $m$ information

| 2730. | 2730. | 0.50 | 0.50 | 0.0 | 0.0 |

Beta distribution on $\lambda_G$ information

| 0.5 | 1.5 | 0.50 | 0.50 | 0.0 | 0.0 |

Time indices of strain time history defining steady state conditions with stochastic rotor speed given by the included Normal distribution information

| 5 | 37592. | 507. |

Normal distribution on $e_A$ information
Normal distribution on $T_s$ information
Normal distribution on $e_D$ information
Uniform distribution bounds for $\epsilon_B$
Uniform distribution bounds for $\lambda_P$
Uniform distribution bounds for $\lambda_{MA}$
Uniform distribution bounds for $\lambda_\alpha$
Uniform distribution bounds for $\lambda_{TH}$
Uniform distribution bounds for $\lambda_{dam}$
Uniform distribution bounds for $\lambda_{TMF}$

**Figure 6.2-1** Format for File BDLCD
### Nominal mechanical strain $\varepsilon_{\text{Mnon}}$ and corresponding rotor speed $\omega_0$

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.295</td>
</tr>
<tr>
<td>3848.2</td>
</tr>
</tbody>
</table>

- Period of reference time history (missions)
- Noise filter (%)
- Number of points in nominal time history
- Walker exponent $w$

#### Coefficients for the start transient response surface function $f_A$

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.00727362$</td>
<td>$0.000067442$</td>
</tr>
<tr>
<td>$-0.000059109$</td>
<td>$-3.52929\times10^{-8}$</td>
</tr>
<tr>
<td>$1.07611\times10^{-8}$</td>
<td>$-2.74419\times10^{-8}$</td>
</tr>
</tbody>
</table>

#### Coefficients for the shutdown transient response surface functions $f_D1$, $f_D2$, and $f_D3$

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-0.132623$</td>
<td>$0.000227427$</td>
</tr>
<tr>
<td>$-0.000059290$</td>
<td>$0.0$</td>
</tr>
<tr>
<td>$0.0$</td>
<td>$4.71714\times10^{-8}$</td>
</tr>
<tr>
<td>$0.20$</td>
<td>$950.0$</td>
</tr>
<tr>
<td>$30523.07$</td>
<td>$-21846.15$</td>
</tr>
</tbody>
</table>

#### Nominal time history

<table>
<thead>
<tr>
<th>Time</th>
<th>$\omega(t)$</th>
<th>$\varepsilon_{\text{TTH}}(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>225.8</td>
<td>$0.0$</td>
<td>$\varepsilon_{\text{TTH}}(t_1)$</td>
</tr>
<tr>
<td>3025.1</td>
<td>$-0.196921$</td>
<td>$\varepsilon_{\text{TTH}}(t_2)$</td>
</tr>
<tr>
<td>6138.8</td>
<td>$0.146025$</td>
<td>$\varepsilon_{\text{TTH}}(t_3)$</td>
</tr>
<tr>
<td>8309.0</td>
<td>$-0.200128$</td>
<td>$\varepsilon_{\text{TTH}}(t_4)$</td>
</tr>
<tr>
<td>0.0</td>
<td>$0.007393$</td>
<td>$\varepsilon_{\text{TTH}}(t_5)$</td>
</tr>
</tbody>
</table>

#### Description of specific material S/N data set

'RT, PWA 1480, 001 DIRECTION'

- Specific materials information: yield and ultimate strengths, number of data divisions, and total number of points in data set

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.54</td>
</tr>
<tr>
<td>1.57</td>
</tr>
<tr>
<td>1.9</td>
</tr>
</tbody>
</table>

- Specific materials information for each data division: number of points in data division, strain ratio, and life region

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
</tr>
<tr>
<td>-1.0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

---

*Figure 6.2-1 Format for File BLDLCD (Cont'd)*
<table>
<thead>
<tr>
<th>$\epsilon_1, N_1$</th>
<th>0.89</th>
<th>6800.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_2, N_2$</td>
<td>0.89</td>
<td>15000.</td>
</tr>
<tr>
<td>$\epsilon_3, N_3$</td>
<td>0.67</td>
<td>27000.</td>
</tr>
<tr>
<td>$\epsilon_4, N_4$</td>
<td>0.67</td>
<td>43200.</td>
</tr>
<tr>
<td>$\epsilon_5, N_5$</td>
<td>0.56</td>
<td>139300.</td>
</tr>
<tr>
<td>$\epsilon_6, N_6$</td>
<td>0.56</td>
<td>545200.</td>
</tr>
<tr>
<td>$\epsilon_7, N_7$</td>
<td>0.56</td>
<td>147000.</td>
</tr>
<tr>
<td>$\epsilon_8, N_8$</td>
<td>0.39</td>
<td>4344800.</td>
</tr>
</tbody>
</table>

1.57 Strain tensile point

1 0 Number of life regions with and without data

5000. Life boundary of region 0

1.0E + 36 Life boundary of region 1

0.00 C constraint

0 0.000 0.000 Prior information on $m$

| 0.00 0.00 0.00 | Bayesian prior distribution information
|-----------------|----------------------------------|
| 0.00 0.00 | Materials process variation information

**Figure 6.2-1** Format for File BLDLCD (Cont'd)
Number of related data sets

Description of related material S/N data set

Related materials information: yield and ultimate strengths, number of data divisions, and total number of points in data set

20 1700. 21 5300. 2 10

Related materials information for data division 1: number of points in data division, stress ratio, and life region

4 0.10 1

S1, N1
S2, N2
S3, N3
S4, N4
6 0.10 2
120000. 72000.
110000. 3224000.
120000. 665000.
110000. 56000.

Figure 6.2-2 Format for File RELATD

Figure 6.2-3 Data Blocks for Input File
Analysis Parameters Block

**RAND**
[DRE]

Random number seed
Needed by BLDLCF's built-in random number generator.

**IOUT**
[INT]

Output dump controller
BLDLCF has the ability to write intermediate calculations to file IOUTPR. The following integer values control the "dump" of BLDLCF's calculations.

- \( IOUT = 0 \) no intermediate calculation output
- \( IOUT = 10 \) materials characterization model calculations
- \( IOUT = 15 \) driver sampling and driver transformation calculations
- \( IOUT = 20 \) rainflow cycle counting and damage accumulation

**NLIFE**
[INT]

Inner loop number
Size of the inner loop of the Monte Carlo (MC) simulation. A positive value is required.

**NHYPER**
[INT]

Outer loop number
Size of the outer loop of the MC simulation. The program requires a positive value.

**NSYM**
[INT]

Symmetry number
The number of modeling units in the component. A positive value is required.

**VARY**
[INT]
Type of S/N variation

Controls the type of stochastic variation to be included in the materials characterization model S/N curve.

VARY = 0 no variation will be included
VARY = 1 allows only intrinsic materials variation
VARY = 2 allows Uniform variation of the materials model shape parameter $m$ and intrinsic materials variation
VARY = 3 allows truncated Normal variation of the materials model shape parameter $m$ and intrinsic materials variation
VARY = 4 allows the variation in the materials model shape parameter $m$ to be "bootstrapped".

NMED [INT]

Request for truncated Normal median S/N curve
If VARY = 3, then NMED controls the calculation of the empirical median S/N curve.

NMED = 0 no median curve calculation is required
NMED = 1 median calculation is required

MPROC [INT]

Controls materials process variation
Controls the inclusion of materials process variation (heat-to-heat variation).
Process variation in materials is discussed in [1], Section 2.1.2.3.

MPROC = 0 no variation to be included
MPROC = 1 variation is to be included

VARPHI [INT]

6 A discussion of the possible stochastic specifications of the materials model shape parameter $m$ is given in [1], Pages 2-13 through 2-14.

7 This option is only available with program BLDLCF V3.4B1.2.

8 The median S/N curve for the truncated Normal distribution is discussed in [1], Page 2-15.
Type of intrinsic materials variation
Controls the type of intrinsic materials variation to be included in the materials characterization model S/N curve. VARPHI is not required if running program BLDLCF V3.4B1.2.

VARPHI = 1 Weibull intrinsic materials variation will be included
VARPHI = 2 Lognormal intrinsic materials variation will be included

NBLIFE
[INT]

Number of B-lives
The number of B-lives to be provided from the simulated distribution of life. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percentage; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%. NBLIFE must be non-negative and cannot exceed 10.

BLFPER(1) BLFPER(2) ... BLFPER(NBLIFE)
[RE] [RE] [RE]

B-life percentages
The decimal equivalent of the percentages at which the B-lives are required; e.g., if the B.1 life is desired, then BLFPER = 0.001. A total of NBLIFE percentages must be provided. The percentage cannot exceed 50% (BLFPER \leq 0.50).

Driver Information Block

HGASA HGASB HGASR1 HGASR2 HGAST1 HGAST2
[RE] [RE] [RE] [RE] [RE] [RE]

Beta distribution on \( h_{gas} \) information
\( h_{gas} \) in Equation 3-2, is the hot gas film coefficient at the start, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for \( h_{gas} \). The next two parameters are the lower and upper bounds for a Uniform distribution on \( \rho \). Similarly, the last two parameters describe a Uniform distribution on \( \theta \).

HGASA lower bound of the Beta distribution on \( h_{gas} \)
HGASB upper bound of the Beta distribution on \( h_{gas} \)
HGASR1 Uniform distribution lower bound of parameter \( \rho \) in the Beta distribution of \( h_{gas} \)
HGASR2 Uniform distribution upper bound of parameter \( \rho \) in the Beta distribution of \( h_{gas} \)
HGAST1 Uniform distribution lower bound of parameter \( \theta \) in the Beta distribution of \( h_{gas} \)
HGAST2 Uniform distribution upper bound of parameter \( \theta \) in the Beta distribution of \( h_{gas} \)

**TGASA** TGASB TGASR1 TGASR2 TGAST1 TGAST2
[RE] [RE] [RE] [RE] [RE] [RE]

Beta distribution on \( T_{gas} \) information

\( T_{gas} \) in Equation 3-2, is the hot gas temperature at the start, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for \( T_{gas} \). The next two parameters are the lower and upper bounds for a Uniform distribution on \( \rho \). Similarly, the last two parameters describe a Uniform distribution on \( \theta \).

- **TGASA** lower bound of the Beta distribution on \( T_{gas} \)
- **TGASB** upper bound of the Beta distribution on \( T_{gas} \)
- **TGASR1** Uniform distribution lower bound of parameter \( \rho \) in the Beta distribution of \( T_{gas} \)
- **TGASR2** Uniform distribution upper bound of parameter \( \rho \) in the Beta distribution of \( T_{gas} \)
- **TGAST1** Uniform distribution lower bound of parameter \( \theta \) in the Beta distribution of \( T_{gas} \)
- **TGAST2** Uniform distribution upper bound of parameter \( \theta \) in the Beta distribution of \( T_{gas} \)

**SLOPEA** SLOPEB SLOPR1 SLOPR2 SLOPT1 SLOPT2
[RE] [RE] [RE] [RE] [RE] [RE]

Beta distribution on \( m \) information

\( m \) (°R/sec) in Equation 3-3, is the deceleration slope at shutdown, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta dis-
distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for \( m \). The next two parameters are the lower and upper bounds for a Uniform distribution on \( \rho \). Similarly, the last two parameters describe a Uniform distribution on \( \theta \).

- **SLOPEA**: lower bound of the Beta distribution on \( m \)
- **SLOPEB**: upper bound of the Beta distribution on \( m \)
- **SLOPR1**: Uniform distribution lower bound of parameter \( \rho \) in the Beta distribution of \( m \)
- **SLOPR2**: Uniform distribution upper bound of parameter \( \rho \) in the Beta distribution of \( m \)
- **SLOPT1**: Uniform distribution lower bound of parameter \( \theta \) in the Beta distribution of \( m \)
- **SLOPT2**: Uniform distribution upper bound of parameter \( \theta \) in the Beta distribution of \( m \)

\( \lambda_G \) in Equation 3-4, is the thermal strain uncertainty factor due to gas temperature variation during the start transient, and it is characterized by a Beta probability distribution. See [1], Section 2.1.3.1 and Equation 2-54, for specifying parameters to define a Beta driver distribution. The Beta distribution format consists of six parameters. The first two parameters are the lower and upper bounds, respectively, for \( \lambda_G \). The next two parameters are the lower and upper bounds for a Uniform distribution on \( \rho \). Similarly, the last two parameters describe a Uniform distribution on \( \theta \).

- **LAMGA**: lower bound of Beta distribution on \( \lambda_G \)
- **LAMGB**: upper bound of Beta distribution on \( \lambda_G \)
- **LAMGR1**: Uniform distribution lower bound of parameter \( \rho \) in the Beta distribution of \( \lambda_G \)
- **LAMGR2**: Uniform distribution upper bound of parameter \( \rho \) in the Beta distribution of \( \lambda_G \)
- **LAMGT1**: Uniform distribution lower bound of parameter \( \theta \) in the Beta distribution of \( \lambda_G \)
- **LAMGT2**: Uniform distribution upper bound of parameter \( \theta \) in the Beta distribution of \( \lambda_G \)
Rotational speed Normal distribution information
The steady state rotational speed is characterized by a Normal($\mu$, $\sigma^2$) distribution. The mean, $\mu$, is equal to the expected operating speed of the turbopump, and the standard deviation, $\sigma$, is obtained from the engine performance balance. Both the mean and standard deviation are in rpm.

- **TSUBI**: time index for strain time history for which distribution on steady state speed is valid
- **SPDMU**: mean, $\mu$, of Normally distributed steady state speed
- **SPDSIG**: standard deviation, $\sigma$, of Normally distributed steady state speed

Modeling uncertainty for the start transient Normal distribution information
$e_A$ is the additive modeling uncertainty characterizing the goodness of fit for the start transient response surface given by Equation 3-2. It is characterized by a Normal($\mu$, $\sigma^2$) distribution. The mean, $\mu$, is equal to the expected modeling uncertainty, usually zero, and the standard deviation, $\sigma$, is obtained from the curve fitting procedure.

- **FAERRM**: mean, $\mu$, of Normally distributed $e_A$
- **FAERRS**: standard deviation, $\sigma$, of Normally distributed $e_A$

Shutdown transient starting temperature Normal distribution information
$T_s$ (°R) in Equation 3-3. It is the gas temperature at the start of the shutdown transient and is characterized by a Normal($\mu$, $\sigma^2$) distribution. The mean, $\mu$, is equal to the expected $T_s$, and the standard deviation, $\sigma$, is obtained from the engine performance balance.

- **TSTMU**: mean, $\mu$, of Normally distributed $T_s$
- **TSTSIG**: standard deviation, $\sigma$, of Normally distributed $T_s
Modeling uncertainty for the shutdown transient Normal distribution information $\theta_D$ is the additive modeling uncertainty characterizing the goodness of fit for the shutdown transient response surface given by Equation 3-3. It is characterized by a Normal($\mu$, $\sigma^2$) distribution. The mean, $\mu$, is equal to the expected modeling uncertainty, usually zero, and the standard deviation, $\sigma$, is obtained from the curve fitting procedure.

- **FDERRM**: mean, $\mu$, of Normally distributed $\theta_D$
- **FDERRS**: standard deviation, $\sigma$, of Normally distributed $\theta_D$

Bending strain Uniform distribution information $\varepsilon_B$ ($\%$) in Equation 3-1. This is the strain due to gas bending and blade tilt, and it is characterized by a Uniform distribution.

- **EBENDA**: Uniform distribution lower bound of $\varepsilon_B$
- **EBENDB**: Uniform distribution upper bound of $\varepsilon_B$

Deviation in blade pull load factor Uniform distribution information $\lambda_P$ in Equation 3-5. This is the deviation in blade pull load due to uncertainty in blade mass factor, and it is characterized by a Uniform distribution.

- **LAMPA**: variation factor Uniform distribution lower bound of $\lambda_P$
- **LAMPB**: variation factor Uniform distribution upper bound of $\lambda_P$

Mechanical strain analysis accuracy factor Uniform distribution information $\lambda_{MA}$ in Equation 3-5. This is the mechanical strain analysis accuracy factor, and it is characterized by a Uniform distribution.
Coefficient of thermal expansion variation factor. Uniform distribution information $\lambda_\alpha$ in Equation 3-4. This is the variation factor for the coefficient of thermal expansion, $\alpha$, and it is characterized by a Uniform distribution.

- **LAMAA**: Uniform distribution lower bound of $\lambda_\alpha$
- **LAMAB**: Uniform distribution upper bound of $\lambda_\alpha$

Thermal strain analysis accuracy factor. Uniform distribution information $\lambda_{TH}$ in Equation 3-4. This is the thermal strain analysis accuracy factor, and it is characterized by a Uniform distribution.

- **TANALA**: Uniform distribution lower bound of $\lambda_{TH}$
- **TANALB**: Uniform distribution upper bound of $\lambda_{TH}$

Damage accumulation model accuracy factor distribution information

This line contains the Uniform distribution bounds in $\log_e$ space for the damage accumulation model accuracy factor, $\lambda_{dam}$, in Equation 2-91 of [1]. See [1], Section 2.2.1.4 for a discussion of the damage accumulation calculations.

- **LAMDAA**: lower bound of damage accumulation accuracy factor
- **LAMDAB**: upper bound of damage accumulation accuracy factor
Thermal Mechanical Fatigue model accuracy factor distribution information
This line contains the Uniform distribution bounds in log_e space for the thermal mechanical fatigue model accuracy factor, \( \lambda_{TMF} \), in Section 3.2.6.

- **LAMTMA**: lower bound of thermal mechanical fatigue accuracy factor
- **LAMTMB**: upper bound of thermal mechanical fatigue accuracy factor

Load and Geometry Block

**EMNOM**  **NOMSPD**

Nominal mechanical strain and rotor speed
The line contains the nominal mechanical strain, \( \varepsilon_{Mnom} \) (%), in Equation 3-5, and the nominal or reference rotor speed, \( \omega_o \) (rpm), corresponding to the nominal mechanical strain value.

**PERIOD**

Period \( T \) (missions) in Equation 2-91 of [1]. This is the period of the nominal strain-time history, and it is required so that life may be provided in missions.

**TRUNC**

Noise Filter
Value (%) used to filter out insignificant cycles in the composite strain-time history during rainflow cycle counting.

**NTIME**

Number of history points
Number of points in the nominal time history. **NTIME** cannot exceed 50.
Walker exponent \( w \) in Equation 3-8. This is the exponent in the Walker relation used in the equivalent zero mean strain range calculation.

Coefficients of the start transient response surface function
The coefficients \( a, b, c, d, e, \) and \( f \) of the start transient response surface function, Equation 3-2.

\[
FA(T_{gas}, h_{gas}) = a + b T_{gas} + c h_{gas} + d T_{gas}^2 + e h_{gas}^2 + f T_{gas} h_{gas}
\]

Coefficients of the shutdown transient response surface functions
The coefficients \( d_{1A}, d_{1B}, d_{1C}, d_{1D}, d_{1E}, d_{2A}, d_{2B}, \) and \( d_{3A}, d_{3B} \) of the shutdown transient response surface functions, Equations 3-3, 3-6, and 3-7.

\[
f_{D1}(m, T_s) = d_{1A} + d_{1B} T_s + d_{1C} m + d_{1D} T_s^2 + d_{1E} m^2 + d_{1F} T_s m \\
t_d = f_{D2}(m, T_s) = d_{2A} + (T_s - d_{2B})/m \\
\omega(t_d) = f_{D3}(t_d) = d_{3A} + d_{3B} t_d
\]
The points of the nominal time history.
The data is entered as rotor speed, thermal strain pairs, one pair per line for \( I = 1, \ldots, \text{NTIME} \).

**Materials Information Block**

**DESCRIPT(0)**

*CHR*

Description of specific material S/N data set
Name and test environment for the specific material S/N data. This is a character string no more than 40 characters long, enclosed by single quotes.

<table>
<thead>
<tr>
<th>FTY</th>
<th>FTU</th>
<th>NDIV</th>
<th>NPTS(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RE]</td>
<td>[RE]</td>
<td>[INT]</td>
<td>[INT]</td>
</tr>
</tbody>
</table>

Specific materials information
Yield strength, ultimate strength, number of divisions of data, number of points in S/N data sets. The data may be divided when they are assigned to a different life region or have different strain ratios. If all data has a strain ratio of \(-1.0\), then the yield and ultimate strengths are not required, but zero values must be specified as placeholders. **NPTS(0)** cannot exceed fifty. The next two data sets have to be provided for each data division.

<table>
<thead>
<tr>
<th>FTY</th>
<th>FTU</th>
<th>NDIV</th>
<th>NPTS(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>yield strength corresponding to the specific material data set (%)</td>
<td>ultimate strength corresponding to the specific material data set (%)</td>
<td>number of data divisions for the specific material data set</td>
<td>total number of points in the specific material S/N data set</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NUM</th>
<th>RATIO</th>
<th>REG</th>
</tr>
</thead>
<tbody>
<tr>
<td>[INT]</td>
<td>[RE]</td>
<td>[INT]</td>
</tr>
</tbody>
</table>

Materials information for each data division of the specific S/N data set
Number of points, strain ratio, and the life region of interest for each data division. This line must be provided for each data division.

<table>
<thead>
<tr>
<th>NUM</th>
<th>RATIO</th>
<th>REG</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of S/N data points in the data division</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RATIO strain ratio for the data in the data division
REG life region number to be assigned to the data in the data division

RAWSTR(I,0) RAWNF(I,0) [RE] [RE]

Specific material S/N data points
Strain versus fatigue life data points for each data division. A block of NUM lines must be specified (i.e., the value of I goes from 1 to NUM). This block must be provided for each data division.

RAWSTR(I,0) strain value (%)
RAWNF(I,0) fatigue life value (cycles)

SZERO [RE]

Tensile point
Strain tensile point, S_o (%). Must be non-negative. A value of zero indicates no tensile point.

NUMREG NNODAT [INT] [INT]

Data regions
Number of life regions that are data-determined and not data-determined. NUMREG + NNODAT cannot exceed three. NUMREG must be 1, 2, or 3, and NNODAT must be non-negative, and should be 0 or 1.

NUMREG number of life regions determined by data
NNODAT number of life regions (to the right) not determined by data

---

9 Extension of the S/N curve to the left is discussed in [1], Page 2-17. This option is not available with the “bootstrapping” model.

10 Extension of the S/N curve to the right is discussed in [1], Page 2-17. This option is not available with the “bootstrapping” model.
NBND(L)  
[RE]

**Life Boundaries**\textsuperscript{11}
The upper boundaries of the life regions are specified (cycles). The value of $L$ goes from \texttt{ZROREG} to the total number of regions (equal to \texttt{NUMREG} + \texttt{NNODAT}). If a non-zero tensile point is specified, then \texttt{ZROREG} = 0 else \texttt{ZROREG} = 1. The program expects the upper bound of the last life region to be $10^{36}$, a proxy for $\infty$.

**CZERO**  
[RE]

Prior information on coefficient of variation of fatigue strength\textsuperscript{12}
Information in the form of a constraint on the coefficient of variation of fatigue strength $C$ for the specific material S/N data set. Value must be non-negative and a value of zero indicates \texttt{CZERO} is not in use.

**MPNT(L) MZERO(1,L) MZERO(2,L)**  
[INT] [RE] [RE]

Prior information on the materials shape parameter $m$\textsuperscript{13}
The number of \texttt{MZERO} values in each life region, and the lower and upper bound for the range of $m$. The value of $L$ goes from 1 to (\texttt{NUMREG} + \texttt{NNODAT}). If \texttt{VARY} = 3 is specified (truncated Normal distribution on $m$), then a prior range of $m$ must be specified for each region.

**MPNT(L)** The number of points, 0, 1, or 2 (no prior on $m$, a point prior on $m$, or a prior over a range of $m$, respectively), in \texttt{MZERO( )} for each region.

**MZERO(1,L)** The lower bound on the range of $m$ or the value of the point prior for $m$.

\textsuperscript{11} Life region boundaries are discussed in [1], Page 2-15.

\textsuperscript{12} The implicit constraint on the materials shape parameter provided by prior information on the coefficient of variation of fatigue strength is discussed in [1], Pages 2-12 through 2-13. This option is not available with the “bootstrapping” model.

\textsuperscript{13} The explicit constraint on the materials shape parameter provided by prior information on the materials shape parameter is discussed in [1], Page 2-12. This option is not available with the “bootstrapping” model.
MZERO(2,L) The upper bound on the prior range of $m$. Program requires that the value be zero if a point prior for $m$ is specified.

DELTA(L) MO(L) SIGMA2(L)
[RE] [RE] [RE]

Information on the Bayesian prior distribution for the truncated Normal distribution.

If VARY = 3, then the materials model uses the truncated Normal distribution. The truncated Normal distribution requires some prior information on the Normal distribution parameters because a Bayesian analysis is performed. The information is required for each life region. The value of L goes from 1 to (NUMREG + NNODAT).

DELTA(L) The shape parameter, $\delta$, of the Bayesian prior distribution is used to compute the Bayesian posterior distribution parameters. Value must be non-negative. A value of zero indicates a diffuse prior distribution.

MO(L) Location parameter, $m_o$, of the Bayesian prior distribution of the shape parameter $m$. Must be positive. Required when DELTA(L) is non-zero.

SIGMA2(L) $\sigma^2$, the known variance of $\ln(fatigue\ life)$, $V(\ln N|\ln S)$. Must be non-negative.

KRATIO LAMN
[RE] [RE]

Materials process variation information

If MPROC = 1, then specification of KRATIO and LAMN is required. KRATIO is $\lambda^*_K$, the ratio $MED\ K'/MED\ K$ where $MED\ K'$ is the median value over all heats for the strain (%) at a life of one cycle, and $MED\ K$ is the median value for the specific S/N data for the strain (%) at a life of one cycle. LAMN is the ratio of the variance of $\ln(life)$ conditional on strain over all heats to the intrinsic materials variation for the given S/N data conditional on strain. Process variation in materials is discussed in [1], Section 2.1.2.3.

---

14 Specification of the Bayesian prior distribution for the truncated Normal case is discussed in [1], Page 2-14.
6.2.3.2 Input File RELATD

The input data for file RELATD, which contains the related materials information,\(^{15}\) is given below. The data format is similar to that used to specify the S/N data in the specific materials information block in the BLDLCD file.

**NSETS**

[Int]

Number of related data sets

Number of related material S/N data sets. The following data groups have to be repeated as a block for each data set. The value of \(J\) varies from 1 to \(\text{NSETS}\). If there is no related data, then file RELATD will only contain the number “0”. \(\text{NSETS}\) cannot exceed five.

**DESCRP(\(J\))**

[CHR]

Description of related material S/N data set

Name and test environment for related material S/N data set \(J\). This is a character string no more than 40 characters long enclosed by single quotes.

**FTY**  **FTU**  **NDIV**  **NPTS(\(J\))**

[RE]  [RE]  [INT]  [INT]

Related materials information

Yield strength, ultimate strength, number of divisions of data, number of points in S/N data set. The data may be divided when they are assigned to a different life region or have different strain ratios. If all data has a strain ratio of \(-1.0\), then the yield and ultimate strengths are not required, but zero values must be specified as placeholders. \(\text{NPTS}(J)\) cannot exceed fifty. The next two data sets have to be provided for each data division.

- **FTY**  yield strength corresponding to related material data set \(J\) (%)
- **FTU**  ultimate strength corresponding to related material data set \(J\) (%)
- **NDIV**  number of data divisions for related material data set \(J\)
- **NPTS(\(J\))**  total number of points in related material S/N data set \(J\)

\(^{15}\) Related S/N data is discussed in [1], Page 2-7. This option is not available with the “bootstrapping” model.
Materials information for each data division of the related S/N data set
Number of points, strain ratio, and the life region of interest for each data division. This line must be provided for each data division.

NUM  number of S/N data points in the data division
RATIO strain ratio for the data in the data division
REG  life region number to be assigned to the data in the data division

Related material S/N data points
Strain versus fatigue life data points for each data division. A block of NUM lines must be specified (i.e., the value of I goes from 1 to NUM). This block must be provided for each data division.

RAWSTR(I,J) strain value (%)
RAWNF(I,J) fatigue life value (cycles)

6.2.4 Options and Capabilities
BLDLCF is a Monte Carlo simulation program which generates a sequence of component lives for a particular failure mode, where life is defined as the accumulated operating time at failure. The simulation has a double-loop structure with NHYPER outer loops and NLIFE inner loops. The simulation size is dependent on the failure probability at which a life estimate is desired and the precision desired. For the blade application, single-loop runs with NHYPER = 20,000 and NLIFE = 1 were used to characterize component reliability, and single-loop runs with NHYPER = 1000 and NLIFE = 1 were used for the marginal analysis to assess the importance of drivers.

During a run it may be desirable to "hold" a driver at a fixed value. This may be the nominal or median value of the driver. This is done for drivers with a Beta or a Uniform distribution by merely specifying the upper and lower bounds to be the desired value. For drivers with a Normal distribution, the standard deviation, \( \sigma \), is set at zero, and the mean, \( \mu \), is set at the desired value.
The procedure of holding certain drivers at fixed values while letting the other drivers vary according to their probability distributions may be used for driver variation sensitivity studies. That is, the effect on life of driver variation may be evaluated by letting it vary while holding other drivers at fixed values. Each driver variation sensitivity was determined in the case studies of this report with the intrinsic variation of the fatigue life of the material included (VARY = 1).

A printout of intermediate calculations in various parts of the program may be obtained via the IOUT option. This output will be printed in the IOUTPR file. It is recommended that such output not be requested when the simulation size is large since the information will be dumped during every simulation loop. The NMED option provides for calculation of an empirical median S/N curve if the truncated Normal distribution or "bootstrapping" is employed. In this case, the median S/N curve is based on the empirical median m from all the shape parameters used in the simulation. The MPROC option activates the calculations for the process variation feature of the materials characterization model, as discussed in [1], Section 2.1.2.3.

6.2.5 Code Execution Example

The following example run of the LCF analysis code for the ATD-HPFTP first stage turbine blade was carried out with random variation of all drivers. In this example run, 20,000 lives were simulated (NLIFE = 1 times NHYPER = 20,000) using Uniform shape parameter variation, VARY = 2 and NMED = 0; Weibull intrinsic materials variation, VARPHI = 1; and no materials process variation, MPROC = 0. The turbine disk has fifty blades about its circumference, so NSYM = 50. The B-lives\(^{17}\) to be provided are B.1, B.2, B.3, B.4, B.5, B.6, B.7, B.8, B.9, and B.1 (NBLIFE = 10, BLFPER(1) = 0.001, BLFPER(2) = 0.002, BLFPER(3) = 0.003, BLFPER(4) = 0.004, BLFPER(5) = 0.005, BLFPER(6) = 0.006, BLFPER(7) = 0.007, BLFPER(8) = 0.008, BLFPER(9) = 0.009, BLFPER(10) = 0.010). The user may refer to Section 3.2 for additional information on the engineering analysis and to Section 3.3 for the results of the case study for this component.

The drivers for LCF failure of the blade are as follows:

---

\(^{16}\) The truncated Normal distribution for the materials model shape parameter m is discussed in [1], Page 2-14.

\(^{17}\) A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.
The rationale for the specification of the driver distributions is given in Section 3.3.1.

The material is for PWA 1480 tested in the [001] orientation, DESCRP = 'RT, PWA 1480, 001 DIRECTION'. The data set includes eight S/N data points, NUM = 8, with a strain ratio of -1.0, RATIO = -1.0. No strain tensile point is used, SZERO = 0, so only one life region upper boundary must be defined, NBND(0) = 1.0E36. The number of regions with data, NUMREG, is 1, and there are no regions to the right without data, NNODAT = 0.\(^{18}\) The data is in one division, NDIV = 1, and the total number of points is eight, NPTS(0) = 8. No constraint on the coefficient of variation of fatigue strength is provided, CZERO = 0. No explicit range on \(m\) is included (MPNT(1) = MZERO(1,L) = MZERO(2,L) = 0). No related data is provided. Thus, the RELATD file is empty, except for a single entry to indicate NSETS = 0.\(^{19}\) If further explanation of files BDLLCD and RELATD is required, refer to Sections 6.2.3.1 and 6.2.3.2, and Figures 6.2-1 and 6.2-2, respectively.

---

\(^{18}\) The nonparametric option is one region only.

\(^{19}\) The nonparametric option does not use a constraint on the coefficient of variation, an explicit range on \(m\), or related data. Nevertheless, placeholders for these parameters must be included because the nonparametric model uses routine RCE without modifications.
The echo of the input data is in the output file BLDLCO. The simulated B-lives are also given for the component. For instance, the B.1 life is 69 missions. The IOUTPR file gives an echo of the analysis parameters. The dump parameter IOUT is zero; therefore, no other output is in this file. The LOWLIF file contains the lowest one percent of the 20,000 simulation lives. Finally, the DUMP file contains the results of the materials characterization model information aggregation calculations.20

Input File - BLDLCD

675
0
1
20000
50
2
0
0
1
10
0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 0.010
676. 2730. 0.50 0.50 0.00 0.00
782. 1982. 0.50 0.50 0.00 0.00
2730. 2730. 0.50 0.50 0.00 0.00
1.5 1.5 0.50 0.50 0.00 0.00
5 37592. 507.
0.0 0.020
1640.0 40.67
0.0 0.003
0.00 0.00
0.96 1.04
0.80 1.20
0.975 1.025
0.70 1.30
0.00 0.00
0.00 0.00
0.295
38482.
1.0
0.0
6
0.50
0.00727362 0.000067442 -0.000059109
-3.52929E-08 1.07611E-08 -2.74419E-08

20 The information aggregation calculations are discussed in [1], Pages 2-6 through 2-14.
Input File - RELATO

Output File - BLDLCO

Copyright (C) 1990, California Institute of Technology. U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

<table>
<thead>
<tr>
<th>RHO</th>
<th>THETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hgas</td>
<td>Be(676., 2730.)</td>
</tr>
<tr>
<td>Tgas (deg R)</td>
<td>Be(782., 1982.)</td>
</tr>
<tr>
<td>DECEL SLOPE</td>
<td>Be(2730., 2730.)</td>
</tr>
</tbody>
</table>
\begin{itemize}
  \item \textit{Tgas UNCERT. Be( 0.50, 1.50) U(0.50000, 0.50000) U( 0.0, 0.0)}
  \item \textit{ROTOR SPEED VARIATION (rpm) AT TIME T5}
  \item \textit{Faccel MODELING ERROR}
  \item \textit{STARTING DECEL TEMPERATURE (deg R)}
  \item \textit{Fdecel MODELING ERROR}
  \item \textit{STRAIN DUE TO GAS BENDING (%)}
  \item \textit{LAMBDA BLADE PULL}
  \item \textit{MECHANICAL ANALYSIS FACTOR}
  \item \textit{COEFFICIENT OF THERMAL EXPANSION FACTOR}
  \item \textit{THERMAL ANALYSIS FACTOR}
  \item \textit{DAMAGE MODEL ACCURACY}
  \item \textit{TMF MODEL ACCURACY}
\end{itemize}

\begin{itemize}
  \item \textit{OTHER STRAIN HISTORY INPUT}
  \item \textit{NOMINAL MECHANICAL STRAIN (%)} \quad 0.2950
  \item \textit{NOMINAL ROTOR SPEED (rpm)} \quad 38482.
  \item \textit{STRAIN-TIME HISTORY PERIOD (missions)} \quad 1.00
  \item \textit{STRAIN-TIME HISTORY NOISE FILTER (%)} \quad 0.00000
  \item \textit{NUMBER OF POINTS IN HISTORIES} \quad 6
  \item \textit{WALKER EXponent} \quad 0.50
\end{itemize}
COEFFICIENTS OF ACCELERATION AND DECELERATION FUNCTIONS

THERMAL STRAIN AT STARTUP (%):

\[ \text{Faccel}(T_{\text{gas}}, H_{\text{gas}}) = 0.727362 \times 10^{-2} + 0.674420 \times 10^{-4} \times T_{\text{gas}} + \\
-0.591090 \times 10^{-4} \times H_{\text{gas}} + -0.352929 \times 10^{-7} \times T_{\text{gas}}^2 + \\
0.107611 \times 10^{-7} \times H_{\text{gas}}^2 + -0.274419 \times 10^{-7} \times T_{\text{gas}} \times H_{\text{gas}} \]

THERMAL STRAIN AT SHUTDOWN (%):

\[ \text{Fdecel1}(m, T_{\text{start}}) = -0.132623 \times 10^0 + 0.227427 \times 10^{-3} \times T_{\text{start}} + \\
-0.592900 \times 10^{-4} \times m + 0.000000 \times 10^0 \times T_{\text{start}}^2 + \\
0.000000 \times 10^0 \times m^2 + 0.471714 \times 10^{-7} \times T_{\text{start}} \times m \]

TIME AT SHUTDOWN (sec):

\[ \text{Fdecel2}(m, T_{\text{start}}) = 0.200000 \times 10^0 + (T_{\text{start}} + 0.950000 \times 10^3) / m \]

ROTOR SPEED AT SHUTDOWN (rpm):

\[ \text{Fdecel3}(t) = 0.305231 \times 10^5 + -0.218462 \times 10^5 \times t \]

STRAIN HISTORY INFORMATION

<table>
<thead>
<tr>
<th>Rotor Speed (rpm)</th>
<th>Thermal Strain (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>225.8</td>
<td>0.000000</td>
</tr>
<tr>
<td>3025.1</td>
<td>-0.196921</td>
</tr>
<tr>
<td>6138.8</td>
<td>0.146025</td>
</tr>
<tr>
<td>8309.0</td>
<td>-0.200128</td>
</tr>
<tr>
<td>0.0</td>
<td>0.007393</td>
</tr>
</tbody>
</table>

MATERIAL INPUT

DESCRIPTION: RT, PWA 1480, 001 DIRECTION

YIELD STRENGTH: 0.154000E+01
ULTIMATE STRENGTH: 0.157000E+01
NUMBER OF POINTS: 8

ORIGINAL S/N | STRESS | TRANSFORMED S/N
--------------|--------|-----------------
STRESS  LIFE  RATIO  REGION  STRESS  LIFE

<table>
<thead>
<tr>
<th>STRESS</th>
<th>LIFE</th>
<th>RATIO</th>
<th>REGION</th>
<th>STRESS</th>
<th>LIFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.89000E+00</td>
<td>6800.</td>
<td>-1.00</td>
<td>1</td>
<td>0.89000E+00</td>
<td>6800.</td>
</tr>
<tr>
<td>0.89000E+00</td>
<td>15000.</td>
<td>-1.00</td>
<td>1</td>
<td>0.89000E+00</td>
<td>15000.</td>
</tr>
<tr>
<td>0.67000E+00</td>
<td>27000.</td>
<td>-1.00</td>
<td>1</td>
<td>0.67000E+00</td>
<td>27000.</td>
</tr>
<tr>
<td>0.67000E+00</td>
<td>43200.</td>
<td>-1.00</td>
<td>1</td>
<td>0.67000E+00</td>
<td>43200.</td>
</tr>
<tr>
<td>0.56000E+00</td>
<td>139300.</td>
<td>-1.00</td>
<td>1</td>
<td>0.56000E+00</td>
<td>139300.</td>
</tr>
<tr>
<td>0.56000E+00</td>
<td>545200.</td>
<td>-1.00</td>
<td>1</td>
<td>0.56000E+00</td>
<td>545200.</td>
</tr>
<tr>
<td>0.56000E+00</td>
<td>147000.</td>
<td>-1.00</td>
<td>1</td>
<td>0.56000E+00</td>
<td>147000.</td>
</tr>
<tr>
<td>0.39000E+00</td>
<td>4344800</td>
<td>-1.00</td>
<td>1</td>
<td>0.39000E+00</td>
<td>4344800</td>
</tr>
</tbody>
</table>

THERE IS 1 REGION(S) WITH DATA 
AND 0 REGION(S) TO THE RIGHT WITHOUT DATA 
THE UPPER BOUND(S) OF THE REGION(S) ARE (CYCLES):

0.100E+37

EXOGENOUS INFORMATION

CONSTRAINT ON COEFFICIENT OF VARIATION, C: 0.0000

EXPLICIT CONSTRAINT ON m FOR EACH REGION:

REGION  # OF POINTS  LOWER BOUND  UPPER BOUND

<table>
<thead>
<tr>
<th>REGION</th>
<th># OF POINTS</th>
<th>LOWER BOUND</th>
<th>UPPER BOUND</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

WEIBULL VARIATION

B LIVES: EMPIRICAL

<table>
<thead>
<tr>
<th>B LIVES</th>
<th>EMPIRICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00100</td>
<td>0.693627E+02</td>
</tr>
<tr>
<td>0.00200</td>
<td>0.104496E+03</td>
</tr>
<tr>
<td>0.00300</td>
<td>0.141498E+03</td>
</tr>
<tr>
<td>0.00400</td>
<td>0.171753E+03</td>
</tr>
<tr>
<td>0.00500</td>
<td>0.203323E+03</td>
</tr>
<tr>
<td>0.00600</td>
<td>0.223266E+03</td>
</tr>
<tr>
<td>0.00700</td>
<td>0.244718E+03</td>
</tr>
<tr>
<td>0.00800</td>
<td>0.266518E+03</td>
</tr>
<tr>
<td>0.00900</td>
<td>0.286573E+03</td>
</tr>
<tr>
<td>0.01000</td>
<td>0.314683E+03</td>
</tr>
<tr>
<td>0.50000</td>
<td>0.900345E+04</td>
</tr>
</tbody>
</table>
RESULTS OF INFORMATION AGGREGATION CALCULATIONS

95% CONFIDENCE INTERVALS ON C AND m FOR EACH REGION

REGION: 1

\( I_o = (0.054422790, 0.185977300) \)

\( J_o = (5.152009000, 9.564463000) \)

POINT ESTIMATES OF C AND m FOR EACH REGION

<table>
<thead>
<tr>
<th>REGION</th>
<th>E(C)</th>
<th>E(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.084455910</td>
<td>7.358236</td>
</tr>
</tbody>
</table>

POSTERIOR CREDIBILITY RANGE ON m FOR EACH REGION

<table>
<thead>
<tr>
<th>REGION</th>
<th>LOWER Bound</th>
<th>UPPER Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1520</td>
<td>9.5645</td>
</tr>
</tbody>
</table>

PARAMETER VALUES FOR MEDIAN S/N CURVE

6 - 76
NUMBER OF REGIONS: 1  \( \beta = 15.7104 \)  \( \beta(k) = 1.0909 \)

<table>
<thead>
<tr>
<th>REGION</th>
<th>( m )</th>
<th>( K )</th>
<th>LIFE BOUND</th>
<th>STRESS BOUND</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.35824</td>
<td>0.30172E+01</td>
<td>0.100E+37</td>
<td>0.000000E+00</td>
</tr>
</tbody>
</table>

**Output File - IOUTPR**

- RANDOM NUMBER SEED = 675.000000000000
- IOUT (MATCHR = 10, BLDLCF = 15, RAINF3 = 20) = 0
- INNER LOOP SIZE = 1
- OUTER LOOP SIZE = 20000
- SYMMETRY NUMBER = 50
- TYPE OF S/N VARIATION DESIRED (0-NONE; 1-INTRINSIC; 2-UNIFORM; 3-NORMAL) = 2
- NORMAL MEDIAN CURVE (0 - NO, 1 - YES) = 0
- MATERIALS PROCESS VARIATION DESIRED (0 - NO, 1 - YES) = 0
- TYPE OF INTRINSIC VARIATION DESIRED (1 - WEIBULL; 2 - LOGNORMAL) = 1

**Output File - LOWLIF**

```
1  0.500000E-04  11.4674
2  0.100000E-03  20.5764
3  0.150000E-03  20.9020
4  0.200000E-03  23.3439
5  0.250000E-03  28.7136
6  0.300000E-03  33.3230
7  0.350000E-03  35.4286
8  0.400000E-03  37.5925
9  0.450000E-03  45.9977
10 0.500000E-03  50.0363
11 0.550000E-03  50.1602
12 0.600000E-03  50.6590
13 0.650000E-03  54.5432
14 0.700000E-03  54.9887
15 0.750000E-03  56.3990
16 0.800000E-03  57.8591
17 0.850000E-03  62.6331
18 0.900000E-03  65.5875
19 0.950000E-03  68.4943
20 1.000000E-02  69.3627
21 1.050000E-02  73.8416
22 1.100000E-02  74.9508
23 1.150000E-02  75.4585
24 1.200000E-02  78.1945
25 1.250000E-02  82.3033
```
<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>0.130000E-02</td>
<td>84.9180</td>
</tr>
<tr>
<td>27</td>
<td>0.135000E-02</td>
<td>85.5436</td>
</tr>
<tr>
<td>28</td>
<td>0.140000E-02</td>
<td>87.7353</td>
</tr>
<tr>
<td>29</td>
<td>0.145000E-02</td>
<td>88.8890</td>
</tr>
<tr>
<td>30</td>
<td>0.150000E-02</td>
<td>93.2934</td>
</tr>
<tr>
<td>31</td>
<td>0.155000E-02</td>
<td>93.3853</td>
</tr>
<tr>
<td>32</td>
<td>0.160000E-02</td>
<td>96.0268</td>
</tr>
<tr>
<td>33</td>
<td>0.165000E-02</td>
<td>96.0511</td>
</tr>
<tr>
<td>34</td>
<td>0.170000E-02</td>
<td>96.3106</td>
</tr>
<tr>
<td>35</td>
<td>0.175000E-02</td>
<td>98.0476</td>
</tr>
<tr>
<td>36</td>
<td>0.180000E-02</td>
<td>99.5991</td>
</tr>
<tr>
<td>37</td>
<td>0.185000E-02</td>
<td>101.824</td>
</tr>
<tr>
<td>38</td>
<td>0.190000E-02</td>
<td>102.286</td>
</tr>
<tr>
<td>39</td>
<td>0.195000E-02</td>
<td>103.012</td>
</tr>
<tr>
<td>40</td>
<td>0.200000E-02</td>
<td>104.496</td>
</tr>
<tr>
<td>41</td>
<td>0.205000E-02</td>
<td>104.946</td>
</tr>
<tr>
<td>42</td>
<td>0.210000E-02</td>
<td>106.325</td>
</tr>
<tr>
<td>43</td>
<td>0.215000E-02</td>
<td>110.003</td>
</tr>
<tr>
<td>44</td>
<td>0.220000E-02</td>
<td>111.212</td>
</tr>
<tr>
<td>45</td>
<td>0.225000E-02</td>
<td>111.670</td>
</tr>
<tr>
<td>46</td>
<td>0.230000E-02</td>
<td>113.510</td>
</tr>
<tr>
<td>47</td>
<td>0.235000E-02</td>
<td>113.610</td>
</tr>
<tr>
<td>48</td>
<td>0.240000E-02</td>
<td>114.501</td>
</tr>
<tr>
<td>49</td>
<td>0.245000E-02</td>
<td>116.168</td>
</tr>
<tr>
<td>50</td>
<td>0.250000E-02</td>
<td>119.642</td>
</tr>
<tr>
<td>51</td>
<td>0.255000E-02</td>
<td>121.653</td>
</tr>
<tr>
<td>52</td>
<td>0.260000E-02</td>
<td>126.945</td>
</tr>
<tr>
<td>53</td>
<td>0.265000E-02</td>
<td>129.652</td>
</tr>
<tr>
<td>54</td>
<td>0.270000E-02</td>
<td>132.441</td>
</tr>
<tr>
<td>55</td>
<td>0.275000E-02</td>
<td>132.713</td>
</tr>
<tr>
<td>56</td>
<td>0.280000E-02</td>
<td>132.853</td>
</tr>
<tr>
<td>57</td>
<td>0.285000E-02</td>
<td>134.850</td>
</tr>
<tr>
<td>58</td>
<td>0.290000E-02</td>
<td>136.655</td>
</tr>
<tr>
<td>59</td>
<td>0.295000E-02</td>
<td>136.710</td>
</tr>
<tr>
<td>60</td>
<td>0.300000E-02</td>
<td>141.498</td>
</tr>
<tr>
<td>61</td>
<td>0.305000E-02</td>
<td>146.554</td>
</tr>
<tr>
<td>62</td>
<td>0.310000E-02</td>
<td>146.987</td>
</tr>
<tr>
<td>63</td>
<td>0.315000E-02</td>
<td>147.589</td>
</tr>
<tr>
<td>64</td>
<td>0.320000E-02</td>
<td>154.347</td>
</tr>
<tr>
<td>65</td>
<td>0.325000E-02</td>
<td>156.143</td>
</tr>
<tr>
<td>66</td>
<td>0.330000E-02</td>
<td>158.882</td>
</tr>
<tr>
<td>67</td>
<td>0.335000E-02</td>
<td>159.672</td>
</tr>
<tr>
<td>68</td>
<td>0.340000E-02</td>
<td>160.197</td>
</tr>
<tr>
<td>69</td>
<td>0.345000E-02</td>
<td>161.686</td>
</tr>
<tr>
<td>70</td>
<td>0.350000E-02</td>
<td>164.602</td>
</tr>
<tr>
<td>71</td>
<td>0.355000E-02</td>
<td>165.648</td>
</tr>
<tr>
<td>72</td>
<td>0.360000E-02</td>
<td>165.831</td>
</tr>
<tr>
<td>73</td>
<td>0.365000E-02</td>
<td>165.867</td>
</tr>
<tr>
<td>74</td>
<td>0.370000E-02</td>
<td>167.298</td>
</tr>
<tr>
<td>75</td>
<td>0.375000E-02</td>
<td>167.348</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>76</td>
<td>0.380000E-02</td>
<td>169.175</td>
</tr>
<tr>
<td>77</td>
<td>0.385000E-02</td>
<td>169.208</td>
</tr>
<tr>
<td>78</td>
<td>0.390000E-02</td>
<td>169.766</td>
</tr>
<tr>
<td>79</td>
<td>0.395000E-02</td>
<td>169.787</td>
</tr>
<tr>
<td>80</td>
<td>0.400000E-02</td>
<td>171.753</td>
</tr>
<tr>
<td>81</td>
<td>0.405000E-02</td>
<td>175.717</td>
</tr>
<tr>
<td>82</td>
<td>0.410000E-02</td>
<td>176.525</td>
</tr>
<tr>
<td>83</td>
<td>0.415000E-02</td>
<td>180.021</td>
</tr>
<tr>
<td>84</td>
<td>0.420000E-02</td>
<td>180.784</td>
</tr>
<tr>
<td>85</td>
<td>0.425000E-02</td>
<td>181.151</td>
</tr>
<tr>
<td>86</td>
<td>0.430000E-02</td>
<td>182.652</td>
</tr>
<tr>
<td>87</td>
<td>0.435000E-02</td>
<td>182.757</td>
</tr>
<tr>
<td>88</td>
<td>0.440000E-02</td>
<td>183.970</td>
</tr>
<tr>
<td>89</td>
<td>0.445000E-02</td>
<td>184.185</td>
</tr>
<tr>
<td>90</td>
<td>0.450000E-02</td>
<td>185.089</td>
</tr>
<tr>
<td>91</td>
<td>0.455000E-02</td>
<td>186.951</td>
</tr>
<tr>
<td>92</td>
<td>0.460000E-02</td>
<td>187.950</td>
</tr>
<tr>
<td>93</td>
<td>0.465000E-02</td>
<td>188.068</td>
</tr>
<tr>
<td>94</td>
<td>0.470000E-02</td>
<td>191.095</td>
</tr>
<tr>
<td>95</td>
<td>0.475000E-02</td>
<td>193.211</td>
</tr>
<tr>
<td>96</td>
<td>0.480000E-02</td>
<td>199.758</td>
</tr>
<tr>
<td>97</td>
<td>0.485000E-02</td>
<td>200.120</td>
</tr>
<tr>
<td>98</td>
<td>0.490000E-02</td>
<td>200.299</td>
</tr>
<tr>
<td>99</td>
<td>0.495000E-02</td>
<td>200.820</td>
</tr>
<tr>
<td>100</td>
<td>0.500000E-02</td>
<td>203.323</td>
</tr>
<tr>
<td>101</td>
<td>0.505000E-02</td>
<td>204.715</td>
</tr>
<tr>
<td>102</td>
<td>0.510000E-02</td>
<td>206.620</td>
</tr>
<tr>
<td>103</td>
<td>0.515000E-02</td>
<td>208.139</td>
</tr>
<tr>
<td>104</td>
<td>0.520000E-02</td>
<td>208.957</td>
</tr>
<tr>
<td>105</td>
<td>0.525000E-02</td>
<td>209.029</td>
</tr>
<tr>
<td>106</td>
<td>0.530000E-02</td>
<td>209.388</td>
</tr>
<tr>
<td>107</td>
<td>0.535000E-02</td>
<td>209.562</td>
</tr>
<tr>
<td>108</td>
<td>0.540000E-02</td>
<td>211.436</td>
</tr>
<tr>
<td>109</td>
<td>0.545000E-02</td>
<td>212.186</td>
</tr>
<tr>
<td>110</td>
<td>0.550000E-02</td>
<td>213.019</td>
</tr>
<tr>
<td>111</td>
<td>0.555000E-02</td>
<td>213.384</td>
</tr>
<tr>
<td>112</td>
<td>0.560000E-02</td>
<td>215.517</td>
</tr>
<tr>
<td>113</td>
<td>0.565000E-02</td>
<td>216.541</td>
</tr>
<tr>
<td>114</td>
<td>0.570000E-02</td>
<td>217.368</td>
</tr>
<tr>
<td>115</td>
<td>0.575000E-02</td>
<td>219.029</td>
</tr>
<tr>
<td>116</td>
<td>0.580000E-02</td>
<td>219.229</td>
</tr>
<tr>
<td>117</td>
<td>0.585000E-02</td>
<td>220.573</td>
</tr>
<tr>
<td>118</td>
<td>0.590000E-02</td>
<td>221.352</td>
</tr>
<tr>
<td>119</td>
<td>0.595000E-02</td>
<td>223.254</td>
</tr>
<tr>
<td>120</td>
<td>0.600000E-02</td>
<td>223.266</td>
</tr>
<tr>
<td>121</td>
<td>0.605000E-02</td>
<td>224.214</td>
</tr>
<tr>
<td>122</td>
<td>0.610000E-02</td>
<td>224.821</td>
</tr>
<tr>
<td>123</td>
<td>0.615000E-02</td>
<td>224.941</td>
</tr>
<tr>
<td>124</td>
<td>0.620000E-02</td>
<td>225.630</td>
</tr>
<tr>
<td>125</td>
<td>0.625000E-02</td>
<td>230.432</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>126</td>
<td>0.630000E-02</td>
<td>230.894</td>
</tr>
<tr>
<td>127</td>
<td>0.635000E-02</td>
<td>232.867</td>
</tr>
<tr>
<td>128</td>
<td>0.640000E-02</td>
<td>233.193</td>
</tr>
<tr>
<td>129</td>
<td>0.645000E-02</td>
<td>235.233</td>
</tr>
<tr>
<td>130</td>
<td>0.650000E-02</td>
<td>235.455</td>
</tr>
<tr>
<td>131</td>
<td>0.655000E-02</td>
<td>235.684</td>
</tr>
<tr>
<td>132</td>
<td>0.660000E-02</td>
<td>237.135</td>
</tr>
<tr>
<td>133</td>
<td>0.665000E-02</td>
<td>239.252</td>
</tr>
<tr>
<td>134</td>
<td>0.670000E-02</td>
<td>240.345</td>
</tr>
<tr>
<td>135</td>
<td>0.675000E-02</td>
<td>241.842</td>
</tr>
<tr>
<td>136</td>
<td>0.680000E-02</td>
<td>242.162</td>
</tr>
<tr>
<td>137</td>
<td>0.685000E-02</td>
<td>242.815</td>
</tr>
<tr>
<td>138</td>
<td>0.690000E-02</td>
<td>244.131</td>
</tr>
<tr>
<td>139</td>
<td>0.695000E-02</td>
<td>244.176</td>
</tr>
<tr>
<td>140</td>
<td>0.700000E-02</td>
<td>244.718</td>
</tr>
<tr>
<td>141</td>
<td>0.705000E-02</td>
<td>245.149</td>
</tr>
<tr>
<td>142</td>
<td>0.710000E-02</td>
<td>248.546</td>
</tr>
<tr>
<td>143</td>
<td>0.715000E-02</td>
<td>251.099</td>
</tr>
<tr>
<td>144</td>
<td>0.720000E-02</td>
<td>251.607</td>
</tr>
<tr>
<td>145</td>
<td>0.725000E-02</td>
<td>251.614</td>
</tr>
<tr>
<td>146</td>
<td>0.730000E-02</td>
<td>252.298</td>
</tr>
<tr>
<td>147</td>
<td>0.735000E-02</td>
<td>253.937</td>
</tr>
<tr>
<td>148</td>
<td>0.740000E-02</td>
<td>255.248</td>
</tr>
<tr>
<td>149</td>
<td>0.745000E-02</td>
<td>259.308</td>
</tr>
<tr>
<td>150</td>
<td>0.750000E-02</td>
<td>259.677</td>
</tr>
<tr>
<td>151</td>
<td>0.755000E-02</td>
<td>260.639</td>
</tr>
<tr>
<td>152</td>
<td>0.760000E-02</td>
<td>261.692</td>
</tr>
<tr>
<td>153</td>
<td>0.765000E-02</td>
<td>262.321</td>
</tr>
<tr>
<td>154</td>
<td>0.770000E-02</td>
<td>263.077</td>
</tr>
<tr>
<td>155</td>
<td>0.775000E-02</td>
<td>263.105</td>
</tr>
<tr>
<td>156</td>
<td>0.780000E-02</td>
<td>263.857</td>
</tr>
<tr>
<td>157</td>
<td>0.785000E-02</td>
<td>265.718</td>
</tr>
<tr>
<td>158</td>
<td>0.790000E-02</td>
<td>265.802</td>
</tr>
<tr>
<td>159</td>
<td>0.795000E-02</td>
<td>266.451</td>
</tr>
<tr>
<td>160</td>
<td>0.800000E-02</td>
<td>266.518</td>
</tr>
<tr>
<td>161</td>
<td>0.805000E-02</td>
<td>266.648</td>
</tr>
<tr>
<td>162</td>
<td>0.810000E-02</td>
<td>268.302</td>
</tr>
<tr>
<td>163</td>
<td>0.815000E-02</td>
<td>268.492</td>
</tr>
<tr>
<td>164</td>
<td>0.820000E-02</td>
<td>268.948</td>
</tr>
<tr>
<td>165</td>
<td>0.825000E-02</td>
<td>268.991</td>
</tr>
<tr>
<td>166</td>
<td>0.830000E-02</td>
<td>269.684</td>
</tr>
<tr>
<td>167</td>
<td>0.835000E-02</td>
<td>272.396</td>
</tr>
<tr>
<td>168</td>
<td>0.840000E-02</td>
<td>272.490</td>
</tr>
<tr>
<td>169</td>
<td>0.845000E-02</td>
<td>273.289</td>
</tr>
<tr>
<td>170</td>
<td>0.850000E-02</td>
<td>273.440</td>
</tr>
<tr>
<td>171</td>
<td>0.855000E-02</td>
<td>273.690</td>
</tr>
<tr>
<td>172</td>
<td>0.860000E-02</td>
<td>275.113</td>
</tr>
<tr>
<td>173</td>
<td>0.865000E-02</td>
<td>277.709</td>
</tr>
<tr>
<td>174</td>
<td>0.870000E-02</td>
<td>278.107</td>
</tr>
<tr>
<td>175</td>
<td>0.875000E-02</td>
<td>279.670</td>
</tr>
</tbody>
</table>
6.2.6 Error Messages and Possible Remedies

The following messages, when applicable, will appear in file IOUTPR. These messages are primarily generated by the materials characterization model (MATCHR) portion of BLDLCF. An error message stating that a limit has been exceeded will require that the user increase those limits, as directed, and reviewing or consulting [1], Section 7.3.1.3, is desirable. The messages are listed in alphabetical order for the convenience of the user.

ERROR: BAD VALUE FOR DELTA OR VALUE OF MO INCONSISTENT WITH DELTA IN REGION 'L'

Fatal This error can occur during the use of the truncated Normal variation option of the materials characterization model for two reasons. First, the value of $\delta$ may be negative. Second, a value of $\delta$ was specified, but the value of $m_o$ is not positive. Check file BLDLC.

ERROR: CO TOO LOW

Fatal The constraint, $C_o$, imposed on the coefficient of variation of fatigue strength is inconsistent with the observed S/N data.
ERROR: EXCEEDED LIMIT ON DEGREES OF FREEDOM IN CHI-SQUARE TABLE, IN REGION 'L'

Fatal As implemented, the credibility interval calculations can handle no more than 150 degrees of freedom, and the amount of data in the region indicated requires more. The $\chi^2$ tables of routine INTRVL must be increased. See [1], Sections 4.1.3.6 and 7.3.1.3, for more information.

ERROR: EXCEEDED LIMIT ON NUMBER OF REGIONS

Fatal The materials characterization model can handle no more than 3 life regions. Check file BLDLCD because the sum of the number of regions with data and the number of regions without data is greater than 3.

ERROR: INVALID RESPONSE TO NORMAL MEDIAN CURVE QUESTION

Fatal NMED can only have the integer value 0 or 1. Check file IOUTPR for the value used.

ERROR: INVALID TYPE OF MATERIALS PROCESS VARIATION DESIRED

Fatal MPROC can only have the integer value 0 or 1. Check file IOUTPR for the value used.

ERROR: INVALID TYPE OF S/N VARIATION DESIRED

Fatal VARY can only have the integer value 0, 1, 2, or 3. Check file IOUTPR for the value used.

ERROR: INVALID VALUE FOR RATIO: 'RATIO'

Fatal An invalid value for the strain ratio has been declared for the specific material data set. Only values between $-1.0$ and $+1.0$ inclusive, are possible. Check file BLDLCD.

ERROR: INVALID VALUE OF RATIO: 'RATIO'

Fatal An invalid value for the strain ratio has been declared for a related material data set. Only values between $-1.0$ and $+1.0$ inclusive, are possible. Check file RELATD.

ERROR: NO INTERSECTION BETWEEN Jo AND Mc
ERROR: NO INTERSECTION BETWEEN Jo AND Mo

21 VARY can also have the integer value of 4 if program BLDLCF V3.4B1.2 is being used.
ERROR: NO INTERSECTION BETWEEN Jo, Mo, AND Mc

ERROR: NO INTERSECTION BETWEEN Mo AND Mc

Fatal These errors indicate that the specified C constraint and/or prior credibility range on m do not agree with each other and/or the observed S/N data.

ERROR: NORMAL VARIATION REQUIRES A PRIOR RANGE ON M

Fatal The truncated Normal variation option of the materials characterization model requires a prior range on m. The number of points for the prior range on m has been incorrectly specified. Check file BLDLCD to verify that the number of points indicated for each range has an integer value of 1 or 2.

ERROR: NUMBER OF POINTS PER DIVISION INCORRECTLY SPECIFIED IN SET 'J'

Fatal The materials characterization model has been given conflicting information about the number of points in one of the related S/N data sets. Check file RELATD for each related data set to compare the total number of points declared with the sum of the numbers of points in each data division.

ERROR: NUMBER OF POINTS PER DIVISION INCORRECTLY SPECIFIED IN SPECIFIC DATA SET

Fatal The materials characterization model has been given conflicting information about the number of points in the specific S/N data set. Check file BLDLCD, since the total number of points in the specific data set declared and the sum of the numbers of points in each data division do not agree.

ERROR: OVERALL PRIOR RANGE INCORRECTLY SPECIFIED IN REGION WITHOUT DATA

Fatal The prior credibility range on m in one of the regions without data has been incorrectly specified. Check file BLDLCD to verify that either more regions without data have been indicated than intended or that the number of points in the prior on m in a region without data has been incorrectly specified. Only the integer value 0, 1, or 2 is acceptable.

ERROR: OVER LIMIT ON NUMBER OF POINTS IN SET 'J'

Fatal The materials characterization model cannot accept more than 50 S/N points in any related material data set. Check file RELATD for the total number of points in each related data set declared, or there may be more than 50 S/N points with an incorrect total declaration. It is suggested that the number of S/N data points in each related set be recounted. If more
than 50 points are desired, the parameter MAXDAT must be increased. Refer to [1], Section 7.3.1.3, for the routines involved.

ERROR: OVER LIMIT ON NUMBER OF RELATED DATA SETS
Fatal The materials characterization model allows up to 5 related data sets. Check file RELATD to determine if more than 5 related data sets were specified. The parameter MAXSET must be increased. Refer to [1], Section 7.3.1.3, for the routines involved.

ERROR: OVER NUMBER OF POINTS LIMIT IN SPECIFIC MATERIAL
Fatal The materials characterization model cannot accept more than 50 S/N points in the specific material data set. Check file BLDLCD for the total number of points in the specific data set declared, or there may be more than 50 S/N points with an incorrect total declaration. If more than 50 points are desired, the parameter MAXDAT must be increased. Refer to [1], Section 7.3.1.3, for the routines involved.

ERROR: OVER REGION LIMIT IN RELATED MATERIAL 'J'
Fatal No more than 3 life regions are allowed, and an attempt has been made to place some S/N data in a region number greater than 3. Check file RELATD for an invalid region number immediately following the strain ratio value in the data set indicated.

ERROR: OVER REGION LIMIT IN SPECIFIC DATA SET
Fatal No more than 3 life regions are allowed, and an attempt has been made to place some S/N data in a region number greater than 3. Check file BLDLCD for an invalid region number immediately following the strain ratio value.

ERROR: POSTERIOR INTERVAL IN REGION 'L' IS INCONSISTENT WITH POINT POSTERIOR IN REGION 'L-1'
Fatal Check file DUMP to verify that the point posterior value of m in region 'L-1' is greater than the upper bound of the posterior credibility range in region 'L'. This error indicates a violation of the concavity assumption.

ERROR: POSTERIOR INTERVAL IN REGION 'L' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN REGION 'L-1'
Fatal Check file DUMP to verify that the lower bound of the posterior credibility range of m in region 'L-1' is greater than the upper bound of the posterior credibility range of m in region 'L'. The data should be checked for consistency.
ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN 'L'

Fatal The number of points for the specified prior range of m in the indicated region has been incorrectly provided. Check file BLDLCD to verify that the number of points indicated for each range has an integer value of 0, 1, or 2.

ERROR: STRAIN-TIME HISTORY TOO LARGE

Fatal No more than 50 points is allowed for the nominal time history and an attempt has been made to use a larger history. Check file BLDLCD for a value of NTIME larger than 50.

ERROR: SXY >= 0 IN REGION 'L'

Fatal During the linear regression calculations, for the region indicated, the resulting value of the sample covariance S_xy was found to be non-negative. This suggests that the data is specified erroneously or is inadequate for analysis, since life increasing with increasing strain contradicts the true fatigue behavior of materials.

ERROR: TOO FEW POINTS FOR REGRESSION IN REGION 'L'

Fatal The materials characterization model does not have the required minimum number of points in the region indicated to perform a linear regression. If there are no related data sets, then there must be at least 3 points in each region. If there are N related data sets, then the total number of points in each region (specific and related combined) must be at least N + 3.

IMPOSSIBLE M RANGE IN REGION 'L'

Fatal Concavity constraints during the random m selection have required an impossible range on m for the region indicated. Take note of all input parameters for this run, and consult [1], Sections 4.1.5.1, 4.1.5.2, and 7.3, to aid in identification of the cause of this error.

NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ON m IN REGION 'L'

Warning This means that the estimate of m based on the S/N data only, in the region indicated, is outside the range indicated by the specified constraints on m and C.

PROCESS EXECUTION TERMINATED

Fatal This message is produced by routine TRMNAT and follows all other fatal messages.
6.2.7 Summary of Input/Output Files

Input Files

BLDLCD
This file is opened in BLDLCE. BLDLCD has the following elements: parameters for the run options; driver distributions; values for nominal strains and their associated parametric sensitivity coefficients; and the specific and exogenous materials input, including yield and ultimate strengths (%), strain ratio, S/N data points, life (cycles) boundaries, region information, coefficient of variation constraint, $C$, and prior ranges on the materials shape parameter $m$ for each region.

RELATD
This file is opened in subroutine INFAGG. It contains the related material data input, including yield and ultimate strengths (%), strain ratio, S/N data points, and region information.

Output Files

BLDLCO
This file is opened in BLDLCE. It contains the echo of the information contained in BLDLCD and provides the simulated failure distribution B-life information.\(^{22}\)

RELATO
This file is opened in subroutine INFAGG. It contains the echo of the information contained in RELATD.

DUMP
This file is opened in BLDLCE. It contains the results of the information aggregation portion of the materials model calculations, such as $I_0$ and $J_0$; the point estimates of $m$ and $C$; posterior credibility ranges for $m$; and a list of the estimated values for all S/N curve parameters. See [1], Section 4.1.

IOUTPR
This file is opened in BLDLCE. It contains information on the particular run that is not echoed to BLDLCO and the data dump provided when the variable IOUT is equal to 10 (materials characterization calculations), 15 (Monte Carlo simulation

\(^{22}\) A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent; e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.
and driver transformation calculations), or 20 (cycle counting and damage accumulation).

LOWLIF
This file is opened in BDLCF. It contains the first one percent of the calculated lives used by the software described in [1], Section 4.2, to calculate $\alpha$, $\beta$, and $\theta$, the parameters of the Bayesian prior failure distribution.

Reference
7.0 Structure and Listing of Programs
Section 7.1
Crack Growth Analysis Software PROCRK

The program tree structures, list of subprograms, descriptions of the key variables, and the FORTRAN source listing for the crack growth analysis code PROCRK are given here. The pertinent crack growth methodology is given in Section 2.2. The overall description of the program and the flowcharts are given in Section 5.1. The user's guide for running PROCRK is given in Section 6.1.

7.1.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for PROCRK is given in Figure 7.1-1. The program, subprogram, and file names are indicated by UPPERCASE letters.

7.1.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7.1-1. The section numbers where the subprograms are described by means of flowcharts are given next to the names.
Figure 7.1-1   Tree Structure for Program PROCRK
Table 7.1-1  List of Subprograms For Program PROCRK  
(Footnotes are at the end of the table)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETAGN</td>
<td>4.4.5*</td>
<td>Generates Beta(a, b, p, θ) random variates.</td>
</tr>
<tr>
<td>BLKGRO</td>
<td>5.1.2.10</td>
<td>Calculates the crack growth rate per load block.</td>
</tr>
<tr>
<td>CYCOUN</td>
<td>5.1.2.9</td>
<td>Calculates the number of cycles by rainflow counting, creates a stress vs. cycles table, and determines the equivalent mean stress.</td>
</tr>
<tr>
<td>DETER4</td>
<td>5.1.2.4</td>
<td>Calculates the determinant of a 4x4 matrix.</td>
</tr>
<tr>
<td>GAM</td>
<td>4.4.4*</td>
<td>Generates Gamma(α, 1) random variates.</td>
</tr>
<tr>
<td>GRODAT</td>
<td>5.1.2.4</td>
<td>Reads material properties and performs regression on crack growth data.</td>
</tr>
<tr>
<td>HYPDRW</td>
<td>5.1.2.5</td>
<td>Performs hyperparameter draws in the outer loop.</td>
</tr>
<tr>
<td>INPUT</td>
<td>5.1.2.3</td>
<td>Reads the data from file CRKDAT and echoes the data to file CRKRES.</td>
</tr>
<tr>
<td>INSORT</td>
<td>5.1.2.7</td>
<td>Performs an insertion sort for the lowest one percent of the lives calculated.</td>
</tr>
<tr>
<td>LIFCAL</td>
<td>5.1.2.7</td>
<td>Calculates the crack growth life.</td>
</tr>
<tr>
<td>M4L1</td>
<td>5.1.3.3</td>
<td>Performs the driver transformation, for location 1, the exterior surface of the duct.</td>
</tr>
<tr>
<td>M4L2</td>
<td>5.1.3.3</td>
<td>Performs the driver transformation, for location 2, the interior surface of the duct.</td>
</tr>
<tr>
<td>NEUBER</td>
<td>5.1.3.6</td>
<td>Calculates the equivalent mean stress from the maximum stress based on Neuber's rule. See Section 2.2.1.4 of [1].</td>
</tr>
<tr>
<td>NORMGN</td>
<td>4.4.3*</td>
<td>Generates Normal(μ, σ²) random variates.</td>
</tr>
<tr>
<td>PARDRW</td>
<td>5.1.2.6</td>
<td>Performs the random life driver parameter draws in the inner loop.</td>
</tr>
<tr>
<td>PROCRK</td>
<td>5.1.2.1</td>
<td>The main routine that controls the logical flow of the probabilistic crack growth analysis.</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7.6.6*</td>
<td>Generates the Uniform(a, b) and Uniform(c, d) pair of independent random variates.</td>
</tr>
<tr>
<td>RANDOM</td>
<td>4.4.2*</td>
<td>Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.</td>
</tr>
<tr>
<td>SETDEF</td>
<td>5.1.2.2</td>
<td>Initializes arrays and variables and sets them to default values.</td>
</tr>
<tr>
<td>STRAN1</td>
<td>5.1.2.8</td>
<td>Derives the composite principal stress history for the HEX coil.</td>
</tr>
<tr>
<td>STRAN2</td>
<td>5.1.2.8</td>
<td>Derives the composite principal stress history for the EXHEX.</td>
</tr>
<tr>
<td>STRIF1</td>
<td>5.1.2.11</td>
<td>Calculates the stress intensity factor coefficients for the HEX coil crack configuration.</td>
</tr>
</tbody>
</table>

Footnotes:

1. Footnotes are at the end of the table.
2. See Section 2.2.1.4 of [1].
Table 7.1-1  List of Subprograms For Program PROCRK  
(Footnotes are at the end of the table)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRIF2</td>
<td>5.1.2.11</td>
<td>Calculates the stress intensity factor coefficients for the EXHEX crack configuration.</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>4.1.11*</td>
<td>Performs premature program termination, when required.</td>
</tr>
</tbody>
</table>

* See [1].

1 The Beta distribution is discussed in [1], Page 2-25.
2 The Normal distribution is discussed in [1], Page 2-23.
3 The Uniform distribution is discussed in [1], Page 2-23.
7.1.3 Description of Variables

A list of variables used in crack growth analysis code, PROCRK, is given in Table 7.1-2. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" can be interpreted as follows: CH6 is a character variable, six characters long; INT is a standard integer variable; LOG is a standard logical variable; RE is a standard real variable; and DRE is a double precision variable. The various array dimensions are defined by using the following parameters: **MAXBLF, MAXDAT, MAXDIV, MAXLD, MAXLIF, MAXM, and MAXSEG**.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AERD</td>
<td>RE</td>
<td>$\lambda_{DAERO}$ in Equation 2-5, the randomly selected load scale factor for the AERoDynamic load components.</td>
</tr>
<tr>
<td>AERDA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the aerodynamic load scale factor.</td>
</tr>
<tr>
<td>AERDB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the aerodynamic load scale factor.</td>
</tr>
<tr>
<td>AERS</td>
<td>RE</td>
<td>$\lambda_{STAERO}$ in Equation 2-5, the randomly selected load scale factor for the AERoStatic load components.</td>
</tr>
<tr>
<td>AERSA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the aerostatic load scale factor.</td>
</tr>
<tr>
<td>AERSB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the aerostatic load scale factor.</td>
</tr>
<tr>
<td>AI</td>
<td>RE</td>
<td>$a_i$ (in.), randomly selected initial crack dimension.</td>
</tr>
<tr>
<td>AIA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on $a_i$.</td>
</tr>
<tr>
<td>AIB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on $a_i$.</td>
</tr>
<tr>
<td>AIR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for $a_i$.</td>
</tr>
<tr>
<td>AIR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the Beta distribution for $a_i$.</td>
</tr>
<tr>
<td>AIR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the Beta distribution for $a_i$.</td>
</tr>
<tr>
<td>AIT</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\theta$ for $a_i$.</td>
</tr>
</tbody>
</table>
### Table 7.1-2 List of Variables For Program PROCRK (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the Beta distribution for ( a_i ).</td>
</tr>
<tr>
<td>AIT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the Beta distribution for ( a_i ).</td>
</tr>
<tr>
<td>ANGLE</td>
<td>RE</td>
<td>( \phi ) (rad) in Equation 2-1, the angle measured counterclockwise from Z-direction to the critical circumferential location.</td>
</tr>
<tr>
<td>AOC</td>
<td>RE</td>
<td>( a/c ), the randomly selected initial crack aspect ratio.</td>
</tr>
<tr>
<td>AOCA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on ( a/c ).</td>
</tr>
<tr>
<td>AOCB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on ( a/c ).</td>
</tr>
<tr>
<td>AOCR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for ( a/c ).</td>
</tr>
<tr>
<td>AOCR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \rho ) in the Beta distribution for ( a/c ).</td>
</tr>
<tr>
<td>AOCR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \rho ) in the Beta distribution for ( a/c ).</td>
</tr>
<tr>
<td>AOCT</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \theta ) for ( a/c ).</td>
</tr>
<tr>
<td>AOCT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the Beta distribution for ( a/c ).</td>
</tr>
<tr>
<td>AOCT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the Beta distribution for ( a/c ).</td>
</tr>
<tr>
<td>ASTR</td>
<td>RE</td>
<td>( \lambda_{AERO_{stat}} ) in Equation 2-5, the randomly selected aerodynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>ASTRA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the aerodynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>ASTRB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the aerodynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>BLFPER(MAXBFL)</td>
<td>RE</td>
<td>1-D array containing user-specified B-lives which are obtained from the simulated failure distribution. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent: e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.</td>
</tr>
<tr>
<td>BLFPOS</td>
<td>INT</td>
<td>The index for the array variable LIFE( ) corresponding to the user-requested simulated failure distribution B-lives contained in variable BLFPER().</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>CEE</td>
<td>RE</td>
<td>C in Equation 2-7, the Generalized Forman model parameter.</td>
</tr>
<tr>
<td>CI</td>
<td>RE</td>
<td>Initial crack size (c_i) (in.) for the elliptic surface flaw.</td>
</tr>
<tr>
<td>CO</td>
<td>RE</td>
<td>(C_o) in Equation 2-10, threshold stress intensity factor (SIF) (\Delta K_{TH}) model parameter.</td>
</tr>
<tr>
<td>COEXP</td>
<td>RE</td>
<td>(\alpha) (°R) in Equation 2-3, the COefficient of thermal EXPansion.</td>
</tr>
<tr>
<td>DADB(2)</td>
<td>RE</td>
<td>Block growth rate (da/dB) in the &quot;a&quot; and &quot;c&quot; directions.</td>
</tr>
<tr>
<td>DADN(MAXDIV, MAXDAT)</td>
<td>RE</td>
<td>2-D array containing the crack growth rate (in./cycle) in the (da/dN) vs. (\Delta K) data.</td>
</tr>
<tr>
<td>DEE</td>
<td>RE</td>
<td>(d) in Equation 2-10, threshold SIF (\Delta K_{TH}) model parameter.</td>
</tr>
<tr>
<td>DELK(MAXDIV, MAXDAT)</td>
<td>RE</td>
<td>2-D array containing the SIF range (ksi √in.) in the (da/dN) vs. (\Delta K) data.</td>
</tr>
<tr>
<td>DESCRP</td>
<td>CH40</td>
<td>Description of the material.</td>
</tr>
<tr>
<td>DK</td>
<td>RE</td>
<td>SIF range (\Delta K(\text{ksi} \sqrt{\text{in.}})).</td>
</tr>
<tr>
<td>DKEFF</td>
<td>RE</td>
<td>Effective SIF range (\Delta K_{eff}) after retardation given in Equation 2-16.</td>
</tr>
<tr>
<td>DKTH</td>
<td>RE</td>
<td>Threshold SIF range (\Delta K_{TH}(\text{ksi} \sqrt{\text{in.}})).</td>
</tr>
<tr>
<td>DKTHO</td>
<td>RE</td>
<td>Threshold SIF range (ksi √in.) at (R = 0) used in Equation 2-10.</td>
</tr>
<tr>
<td>DLTAT</td>
<td>RE</td>
<td>Delta T. (\Delta T) (°R) in Equation 2-2, the temperature difference across the wall of the duct.</td>
</tr>
<tr>
<td>DPCMU</td>
<td>RE</td>
<td>Value of (PCMUB − PCMUA).</td>
</tr>
<tr>
<td>DPCSIG</td>
<td>RE</td>
<td>Value of (PCSIGB − PCSIGA).</td>
</tr>
<tr>
<td>DSALT</td>
<td>RE</td>
<td>Bin stress interval for the stress level vs. number of cycles table from rainflow counting.</td>
</tr>
<tr>
<td>DSTR</td>
<td>RE</td>
<td>(\lambda_{DYNstr}) in Equation 2-5, the randomly selected dynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>DSTRA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the dynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>DSTRB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the dynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>DTIMU</td>
<td>RE</td>
<td>Value of (TIMUB − TIMUA).</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>DTISIG</td>
<td>RE</td>
<td>Value of (TISIGB - TISIGA).</td>
</tr>
<tr>
<td>DTOMU</td>
<td>RE</td>
<td>Value of (TOMUB - TOMUA).</td>
</tr>
<tr>
<td>DTOSIG</td>
<td>RE</td>
<td>Value of (TOSIGB - TOSIGA).</td>
</tr>
<tr>
<td>E(MAXSEG)</td>
<td>RE</td>
<td>1-D array containing the strain ( \varepsilon ) values for the stress/strain versus strain curve.</td>
</tr>
<tr>
<td>EM</td>
<td>RE</td>
<td>( E ) (psi) in Equation 2-2, Young's modulus of elasticity for the material.</td>
</tr>
<tr>
<td>EMM</td>
<td>RE</td>
<td>( m ) in Equation 2-7, the Generalized Forman model parameter.</td>
</tr>
<tr>
<td>ENN</td>
<td>RE</td>
<td>( n ) in Equation 2-7, the Generalized Forman model parameter.</td>
</tr>
<tr>
<td>FAIL</td>
<td>LOG</td>
<td>Unstable crack growth indicator when ( K &gt; K_{cr} ).</td>
</tr>
<tr>
<td>FILNUM(MAXLD)</td>
<td>INT</td>
<td>1-D array containing the file unit numbers for the reference time history files.</td>
</tr>
<tr>
<td>FK(10)</td>
<td>RE</td>
<td>1-D array containing values of ( F_k ), Equation 2-3, used to find stress concentration due to weld eccentricity, ( K_{OFF} ).</td>
</tr>
<tr>
<td>FTEST</td>
<td>LOG</td>
<td>Used to test for existence of files.</td>
</tr>
<tr>
<td>FTY</td>
<td>RE</td>
<td>Material yield strength (psi).</td>
</tr>
<tr>
<td>INDIA</td>
<td>RE</td>
<td>( D_i ) (in.), the randomly selected inner diameter.</td>
</tr>
<tr>
<td>INDIABA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on ( D_i ).</td>
</tr>
<tr>
<td>INDIAB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on ( D_i ).</td>
</tr>
<tr>
<td>INDIR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for ( D_i ).</td>
</tr>
<tr>
<td>INDIR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \rho ) in the Beta distribution for ( D_i ).</td>
</tr>
<tr>
<td>INDIR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \rho ) in the Beta distribution for ( D_i ).</td>
</tr>
<tr>
<td>INDIT</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \theta ) for ( D_i ).</td>
</tr>
<tr>
<td>INDIT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the Beta distribution for ( D_i ).</td>
</tr>
<tr>
<td>INDIT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the Beta distribution for ( D_i ).</td>
</tr>
</tbody>
</table>
### Table 7.1-2
List of Variables For Program PROCRK (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>INEUB</td>
<td>INT</td>
<td>Neuber's rule controller. INEUB = 0, no Neuber's equivalent mean stress calculation; INEUB = 1, include Neuber's equivalent mean stress calculation.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller. IOUT = 0, no intermediate calculation output; IOUT = 15, driver sampling and driver transformation calculations; IOUT = 20, crack growth calculations; IOUT = 25, stress calculations; IOUT = 30, rainflow cycle counting.</td>
</tr>
<tr>
<td>IREGOP</td>
<td>INT</td>
<td>Regression options for Forman growth rate Equation 2-7. IREGOP = 0, fix p regress for C, n, m, q; IREGOP = 1, fix m, p regress for C, n, q; IREGOP = 2, fix q, p regress for C, n, m; IREGOP = 3, fix m, q, p regress for C, n; IREGOP = 4, regress for C, n, m, q, p.</td>
</tr>
<tr>
<td>IRET</td>
<td>INT</td>
<td>Willenborg's retardation model controller. IRET = 0, no growth retardation; IRET = 1, include growth retardation.</td>
</tr>
<tr>
<td>KC</td>
<td>RE</td>
<td>Critical stress intensity factor $K_C$ (ksi).</td>
</tr>
<tr>
<td>KGROW</td>
<td>INT</td>
<td>Generalized Forman coefficient $m$ controller. KGROW = 1, no $m$ variation will be included; KGROW = 2, allows Uniform variation in $m$.</td>
</tr>
<tr>
<td>KLAM</td>
<td>RE</td>
<td>Randomly selected stress intensity factor calculation accuracy $\lambda_{sif}$.</td>
</tr>
<tr>
<td>KLAMA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the stress intensity factor calculation accuracy.</td>
</tr>
<tr>
<td>KLAMB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the stress intensity factor calculation accuracy.</td>
</tr>
<tr>
<td>KMAX(2)</td>
<td>RE</td>
<td>Maximum stress intensity factor $K_{\text{max}}$ (ksi).</td>
</tr>
<tr>
<td>KMAXEF</td>
<td>RE</td>
<td>Effective maximum stress intensity factor $K_{\text{max,eff}}$ after retardation given in Equation 2-12.</td>
</tr>
<tr>
<td>KMIN(2)</td>
<td>RE</td>
<td>Minimum stress intensity factor $K_{\text{min}}$ (ksi).</td>
</tr>
<tr>
<td>KMINEF</td>
<td>RE</td>
<td>Effective minimum stress intensity factor $K_{\text{min,eff}}$ after retardation given in Equation 2-12.</td>
</tr>
<tr>
<td>KOFF</td>
<td>RE</td>
<td>$K_{\text{OFF}}$ in Equation 2-3, the stress concentration factor due to eccentricity of the weld.</td>
</tr>
<tr>
<td>KPROB</td>
<td>INT</td>
<td>Type of crack growth problem. KPROB = 1, analyze the HEX coil problem; KPROB = 2, analyze EXHEX problem.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>LAMGR</td>
<td>RE</td>
<td>$\lambda_{gr}$ in Equation 2-18, the randomly selected crack growth accuracy factor. See Section 2.2.4 for a discussion of crack growth calculations.</td>
</tr>
<tr>
<td>LAMGRA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the crack growth accuracy factor.</td>
</tr>
<tr>
<td>LAMGRB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the crack growth accuracy factor.</td>
</tr>
<tr>
<td>LAMKC</td>
<td>RE</td>
<td>$\lambda_{Kc}$ in Equation 2-8, the randomly selected critical stress intensity factor uncertainty.</td>
</tr>
<tr>
<td>LAMKCA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the critical stress intensity factor uncertainty.</td>
</tr>
<tr>
<td>LAMKCB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the critical stress intensity factor uncertainty.</td>
</tr>
<tr>
<td>LAMKH</td>
<td>RE</td>
<td>$\lambda_{KH}$ in Equation 2-8, the randomly selected threshold stress intensity factor range uncertainty.</td>
</tr>
<tr>
<td>LAMKHA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the threshold stress intensity factor range uncertainty.</td>
</tr>
<tr>
<td>LAMKHB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the threshold stress intensity factor range uncertainty.</td>
</tr>
<tr>
<td>LAMN</td>
<td>RE</td>
<td>$\lambda_{DRANDOM}$ in Equation 2-5, the randomly selected load scale factor for the narrow-band random loads. See Section 2.1.3.2 of [1] for a description of the parameters $k$, coefficient of variation $C$, and strain gage factor $d$.</td>
</tr>
<tr>
<td>LAMNA</td>
<td>RE</td>
<td>Lower bound of the Uniform distribution of $k$ for the narrow-band random load scale factor.</td>
</tr>
<tr>
<td>LAMNB</td>
<td>RE</td>
<td>Upper bound of the Uniform distribution of $k$ for the narrow-band random load scale factor.</td>
</tr>
<tr>
<td>LAMNC</td>
<td>RE</td>
<td>Coefficient of variation $C$ for the narrow-band random load scale factor.</td>
</tr>
<tr>
<td>LAMND</td>
<td>RE</td>
<td>Strain gage correction factor $d$ for the narrow-band random load scale factor.</td>
</tr>
<tr>
<td>LAMNK</td>
<td>RE</td>
<td>Randomly selected $k$ for the narrow-band random load scale factor.</td>
</tr>
<tr>
<td>LAMNNU</td>
<td>RE</td>
<td>The resulting mean, $\mu$, of the Normal distribution for the narrow-band random load scale factor, where $\mu = d/(1 + kC)$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>LAMNSG</td>
<td>RE</td>
<td>The resulting standard deviation, $\sigma$, of the Normal distribution for the narrow-band random load scale factor, where $\sigma = C/(1 + kC)$.</td>
</tr>
<tr>
<td>LAMS</td>
<td>RE</td>
<td>$\hat{\lambda}_{\text{DSINUSOIDAL}}$ in Equation 2-5, the randomly selected load scale factor for the superimposed sinusoidal loads. See Section 2.1.3.2 of [1] for a description of the parameters $k$, coefficient of variation $C$, and strain gage factor $d$.</td>
</tr>
<tr>
<td>LAMSA</td>
<td>RE</td>
<td>Lower bound of the Uniform distribution of $k$ for the superimposed sinusoidal load scale factor.</td>
</tr>
<tr>
<td>LAMSB</td>
<td>RE</td>
<td>Upper bound of the Uniform distribution of $k$ for the superimposed sinusoidal load scale factor.</td>
</tr>
<tr>
<td>LAMSC</td>
<td>RE</td>
<td>Coefficient of variation $C$ for the superimposed sinusoidal load scale factor.</td>
</tr>
<tr>
<td>LAMSD</td>
<td>RE</td>
<td>Strain gage correction factor $d$ for the superimposed sinusoidal load scale factor.</td>
</tr>
<tr>
<td>LAMSK</td>
<td>RE</td>
<td>Randomly selected $k$ for the superimposed sinusoidal load scale factor.</td>
</tr>
<tr>
<td>LAMSMU</td>
<td>RE</td>
<td>The resulting mean, $\mu$, of the Normal distribution for the superimposed sinusoidal load scale factor, where $\mu = d/(1 + kC)$.</td>
</tr>
<tr>
<td>LAMSSG</td>
<td>RE</td>
<td>The resulting standard deviation, $\sigma$, of the Normal distribution for the superimposed sinusoidal load scale factor, where $\sigma = C/(1 + kC)$.</td>
</tr>
<tr>
<td>LAMW</td>
<td>RE</td>
<td>LAMBda Weld offset, the randomly selected $\lambda_{\text{OFF}}$ in Equation 2-3, the accuracy factor for the weld offset eccentricity stress concentration factor, $K_{\text{OFF}}$.</td>
</tr>
<tr>
<td>LAMWA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda_{\text{OFF}}$.</td>
</tr>
<tr>
<td>LAMWB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\lambda_{\text{OFF}}$.</td>
</tr>
<tr>
<td>LDNAME(MAXLD)</td>
<td>CH6</td>
<td>1-D array containing Load NAMEs for the dynamic or time-varying loads. These are the names of the reference time history files.</td>
</tr>
<tr>
<td>LIFE(MAXLIF)</td>
<td>RE</td>
<td>1-D array containing values of the lives generated by program PROCRK. The lives are sorted values for the left-hand tail simulated failure distribution.</td>
</tr>
<tr>
<td>LOCAT</td>
<td>INT</td>
<td>Critical location of interest on the HEX coil wall where 1 is the exterior surface of the duct, and 2 is the interior surface of the duct.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>M(2, MAXLD)</td>
<td>RE</td>
<td>2-D array containing the dynamic or time-varying moment load components. ( M(1,\ast) ) is ( M_y ) (in.-lbs) in Equation 2-1, the moment load components about the y-axis; and ( M(2,\ast) ) is ( M_z ) (in.-lbs) in Equation 2-1, the moment load components about the z-axis.</td>
</tr>
<tr>
<td>MAXBLF</td>
<td>INT</td>
<td>Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.</td>
</tr>
<tr>
<td>MAXDAT</td>
<td>INT</td>
<td>Maximum number of points per data division allowed for ( da/dN ) vs. ( \Delta K ) curve. The maximum number of data points per division allowed is 200.</td>
</tr>
<tr>
<td>MAXDIV</td>
<td>INT</td>
<td>Maximum number of data divisions allowed for ( da/dN ) vs. ( \Delta K ) curve. The maximum number of data divisions allowed is 10.</td>
</tr>
<tr>
<td>MAXLD</td>
<td>INT</td>
<td>Maximum number of dynamic or time-varying loads allowed. The maximum number of loads is 16.</td>
</tr>
<tr>
<td>MAXLIF</td>
<td>INT</td>
<td>Maximum number of crack growth lives allowed for the simulated failure distribution. The maximum number of crack growth lives to be saved is 1000.</td>
</tr>
<tr>
<td>MAXM</td>
<td>INT</td>
<td>Maximum number of points allowed in the time history arrays. The maximum number of points is 20,000.</td>
</tr>
<tr>
<td>MAXSEG</td>
<td>INT</td>
<td>Maximum number of segments allowed in the stress-strain versus strain curve. The maximum number of segments is 10.</td>
</tr>
<tr>
<td>MI</td>
<td>RE</td>
<td>( I ) (in.(^4)) in Equation 2-1, the cross-sectional Moment of Inertia.</td>
</tr>
<tr>
<td>MLAM(2, MAXLD)</td>
<td>RE</td>
<td>2-D array containing the dynamic or time-varying moment load components scaled by DSTR or ASTR and LAMS, LAMN, or AERD, as appropriate, according to variable TYPE(). ( MLAM(1,\ast) ) is ( M_y ) (in.-lbs) in Equation 2-1, the moment load components about the y-axis; and ( MLAM(2,\ast) ) is ( M_z ) (in.-lbs) in Equation 2-1, the moment load components about the z-axis.</td>
</tr>
</tbody>
</table>
| MSLAM(2)     | RE   | 1-D array containing the static moment load components scaled by ASTR, and AERS, or SSTR as appropriate. \( MSLAM(1) \) is \( M_y \) (in.-lbs) in Equation 2-1, the moment load component about the y-axis;
Table 7.1-2  List of Variables For Program PROCRK (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTAT(2)</td>
<td>RE</td>
<td>1-D array containing the static moment load components. MSTAT(1) is $M_y$ (in.-lbs) in Equation 2-1, the moment load component about the y-axis; and MSTAT(2) is $M_z$ (in.-lbs) in Equation 2-1, the moment load component about the z-axis.</td>
</tr>
<tr>
<td>MVAR</td>
<td>RE</td>
<td>Randomly selected Forman coefficient $m$.</td>
</tr>
<tr>
<td>MVARA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the Forman coefficient $m$.</td>
</tr>
<tr>
<td>MVARB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the Forman coefficient $m$.</td>
</tr>
<tr>
<td>NBIN(100)</td>
<td>INT</td>
<td>1-D array containing the number of cycles for the stress level vs. number of cycles table from rainflow counting.</td>
</tr>
<tr>
<td>NBLIFE</td>
<td>INT</td>
<td>Number of B-lives to be obtained from the simulated failure distribution.</td>
</tr>
<tr>
<td>NCRL</td>
<td>INT</td>
<td>Number of crack lengths for life calculations.</td>
</tr>
<tr>
<td>NDIR</td>
<td>INT</td>
<td>Number of directions to grow the crack in.</td>
</tr>
<tr>
<td>NDIV</td>
<td>INT</td>
<td>Number of crack growth data dimensions.</td>
</tr>
<tr>
<td>NEUB</td>
<td>RE</td>
<td>Randomly selected Neuber's rule model accuracy factor $\lambda_{\text{neu}}$.</td>
</tr>
<tr>
<td>NEUBA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the Neuber's rule model accuracy factor.</td>
</tr>
<tr>
<td>NEUBBB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the Neuber's rule model accuracy factor.</td>
</tr>
<tr>
<td>NEWLIF</td>
<td>INT</td>
<td>Crack growth life value returned from call to LIFCAL.</td>
</tr>
<tr>
<td>NHYPER</td>
<td>INT</td>
<td>The outer loop size.</td>
</tr>
<tr>
<td>NLIFE</td>
<td>INT</td>
<td>The inner loop size.</td>
</tr>
<tr>
<td>NLIIFET</td>
<td>INT</td>
<td>Total number of lives calculated by program PROCRK. Value of NHYPER * NLIFE.</td>
</tr>
<tr>
<td>NLOAD</td>
<td>INT</td>
<td>NLOAD in Equation 2-5, the number of dynamic or time-varying loads.</td>
</tr>
<tr>
<td>NP(MAXDIV)</td>
<td>INT</td>
<td>1-D array containing the number of points per data division for the material $da/dN$ data set.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>NRAN</td>
<td>INT</td>
<td>Number of RANdom points. Number of points in the reference time history.</td>
</tr>
<tr>
<td>NU</td>
<td>RE</td>
<td>ν in Equation 2-2, the materials Poisson's ratio.</td>
</tr>
<tr>
<td>NUMSEG</td>
<td>INT</td>
<td>Number of segments of interest in stress-strain versus strain curve.</td>
</tr>
<tr>
<td>P(MAXLD)</td>
<td>RE</td>
<td>1-D array containing $P$ (lbs) in Equation 2-1, the dynamic or time-varying axial load components.</td>
</tr>
<tr>
<td>PC</td>
<td>RE</td>
<td>$p_i$ (psi) in Equation 2-1, the randomly selected internal pressure.</td>
</tr>
<tr>
<td>PCMU</td>
<td>RE</td>
<td>Randomly selected Normal distribution parameter $\mu$ for the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCMUA</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\mu$ in the Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCMUB</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\mu$ in the Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCO</td>
<td>RE</td>
<td>$p_o$ (psi) in Equation 2-1, the external pressure.</td>
</tr>
<tr>
<td>PCSIG</td>
<td>RE</td>
<td>Randomly selected Normal distribution parameter $\sigma$ for the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCSIGA</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\sigma$ in the Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCSIGB</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\sigma$ in the Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PEE</td>
<td>RE</td>
<td>$p$ in Equation 2-7, the Generalized Forman model parameter.</td>
</tr>
<tr>
<td>PERIOD</td>
<td>RE</td>
<td>$T$ (sec) in Equation 2-18, the length of time in seconds of the reference time history.</td>
</tr>
<tr>
<td>PI</td>
<td>RE</td>
<td>$\pi$, constant equal to 3.1415926536...</td>
</tr>
<tr>
<td>PLAM(MAXLD)</td>
<td>RE</td>
<td>1-D array containing $P$ (lbs) in Equation 2-1, the dynamic or time-varying axial load components scaled by DSTR or ASTR and LAMN, LAMS, or AERD, as appropriate, according to variable TYPE().</td>
</tr>
<tr>
<td>PSLAM</td>
<td>RE</td>
<td>$P$ (lbs) in Equation 2-1, the static axial load component scaled by ASTR, and AERS, or SSTR as appropriate.</td>
</tr>
</tbody>
</table>
**Table 7.1-2** List of Variables For Program PROCRK (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSTAT</td>
<td>RE</td>
<td>$P$ (lbs) in Equation 2-1, the static axial load component.</td>
</tr>
<tr>
<td>QUE</td>
<td>RE</td>
<td>$q$ in Equation 2-7, the Generalized Forman model parameter.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>RDATA(MAXDIV)</td>
<td>RE</td>
<td>1-D array containing the stress ratio for growth rate data for each division.</td>
</tr>
<tr>
<td>REFF</td>
<td>RE</td>
<td>Effective stress ratio $K_{\text{min,eff}}/K_{\text{max,eff}}$ after retardation given by Equation 2-16.</td>
</tr>
<tr>
<td>RI</td>
<td>RE</td>
<td>$R_i$ (in.) in Equation 2-1, the duct inner radius.</td>
</tr>
<tr>
<td>RO</td>
<td>RE</td>
<td>$R_o$ (in.) in Equation 2-1, the duct outer radius.</td>
</tr>
<tr>
<td>ROT</td>
<td>RE</td>
<td>$R$ Over $T$, the value of the ratio $R/t$.</td>
</tr>
<tr>
<td>RSO</td>
<td>RE</td>
<td>Willenborg retardation model parameter as given in Equation 2-13.</td>
</tr>
<tr>
<td>RT(IO)</td>
<td>RE</td>
<td>1-D array containing values of $R/t$ used in conjunction with $F_k$, Equation 2-3, to find stress concentration due to weld eccentricity, $K_{\text{OFF}}$.</td>
</tr>
<tr>
<td>SE(MAXSEG)</td>
<td>RE</td>
<td>1-D array containing values of the product of stress and strain $\sigma\epsilon$ for each segment of the stress-strain versus strain curve.</td>
</tr>
<tr>
<td>SPR(MAXM)</td>
<td>RE</td>
<td>1-D array containing the principal stress-time history $\sigma(t)$ (psi), Equation 2-5, resulting from the combination of stresses from static, narrow-band random, superimposed sinusoidal, and aerodynamic load sources.</td>
</tr>
<tr>
<td>SSTR</td>
<td>RE</td>
<td>$\lambda ST_{st}$ in Equation 2-5, the randomly selected static stress analysis accuracy factor.</td>
</tr>
<tr>
<td>SSTRA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the static stress analysis accuracy factor.</td>
</tr>
<tr>
<td>SSTRTB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the static stress analysis accuracy factor.</td>
</tr>
<tr>
<td>STATIC(4)</td>
<td>RE</td>
<td>1-D array containing values of the static stresses $\sigma_{ST}$ (psi), Equation 2-5. STATIC(1) is the axial stress $\sigma_{ST}$. STATIC(2), STATIC(3), and STATIC(4) are not used in the HEX coil or EXHEX analyses.</td>
</tr>
<tr>
<td>STRAMP(4, MAXLD)</td>
<td>RE</td>
<td>2-D array containing values of the amplitudes of the dynamic or time-varying stresses $\sigma_{Df}$ (psi).</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>STRHIS(MAXLD, MAXM)</td>
<td>RE</td>
<td>2-D array containing $\sigma(t)$, Equation 2-5, the reference time histories for the dynamic or time-varying load components.</td>
</tr>
<tr>
<td>SX(MAXLD)</td>
<td>RE</td>
<td>1-D array containing the time-varying magnitude of $\sigma_x$ (psi) stress component.</td>
</tr>
<tr>
<td>SXY(MAXLD)</td>
<td>RE</td>
<td>1-D array containing the time-varying magnitude of $\sigma_{xy}$ (psi) stress component.</td>
</tr>
<tr>
<td>SXZ(MAXLD)</td>
<td>RE</td>
<td>1-D array containing the time-varying magnitude of $\sigma_{xz}$ (psi) stress component.</td>
</tr>
<tr>
<td>SXST</td>
<td>RE</td>
<td>Static $\sigma_x$ (psi) stress component.</td>
</tr>
<tr>
<td>SXYST</td>
<td>RE</td>
<td>Static $\sigma_{xy}$ (psi) stress component.</td>
</tr>
<tr>
<td>SXZST</td>
<td>RE</td>
<td>Static $\sigma_{xz}$ (psi) stress component.</td>
</tr>
<tr>
<td>SY(MAXLD)</td>
<td>RE</td>
<td>1-D array containing the time-varying magnitude of $\sigma_y$ (psi) stress component.</td>
</tr>
<tr>
<td>SYZ(MAXLD)</td>
<td>RE</td>
<td>1-D array containing the time-varying magnitude of $\sigma_{yz}$ (psi) stress component.</td>
</tr>
<tr>
<td>SZ(MAXLD)</td>
<td>RE</td>
<td>1-D array containing the time-varying magnitude of $\sigma_z$ (psi) stress component.</td>
</tr>
<tr>
<td>SYST</td>
<td>RE</td>
<td>Static $\sigma_y$ (psi) stress component.</td>
</tr>
<tr>
<td>SYZST</td>
<td>RE</td>
<td>Static $\sigma_{yz}$ (psi) stress component.</td>
</tr>
<tr>
<td>SZST</td>
<td>RE</td>
<td>Static $\sigma_z$ (psi) stress component.</td>
</tr>
<tr>
<td>T(MAXLD)</td>
<td>RE</td>
<td>1-D array containing $M_x$ (in.-lbs) the dynamic or time-varying torsional load components. Not used in the HEX coil or EXHEX analysis.</td>
</tr>
<tr>
<td>THIC</td>
<td>RE</td>
<td>$t$ (in.), the randomly selected wall thickness at the weld used to calculate the area $A$ and outer radius $R_o$ in Equation 2-1.</td>
</tr>
<tr>
<td>THICA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on $t$.</td>
</tr>
<tr>
<td>THICB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on $t$.</td>
</tr>
<tr>
<td>THICR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for the wall thickness $t$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>THICR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $t$.</td>
</tr>
<tr>
<td>THICR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $t$.</td>
</tr>
<tr>
<td>THICT</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\theta$ for the wall thickness $t$.</td>
</tr>
<tr>
<td>THICT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $t$.</td>
</tr>
<tr>
<td>THICT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $t$.</td>
</tr>
<tr>
<td>TIN</td>
<td>RE</td>
<td>$T_i$ (°R), the randomly selected inner wall surface temperature, used to calculate $\Delta T$ (°R), the temperature difference across the wall of the duct, given in Equation 2-2.</td>
</tr>
<tr>
<td>TIMU</td>
<td>RE</td>
<td>Randomly selected Normal distribution parameter $\mu$ for the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TIMUA</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\mu$ in the Normal distribution of the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TIMUB</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\mu$ in the Normal distribution of the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TISIG</td>
<td>RE</td>
<td>Randomly selected Normal distribution parameter $\sigma$ for the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TISIGA</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\sigma$ in the Normal distribution of the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TISIGB</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\sigma$ in the Normal distribution of the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TOUT</td>
<td>RE</td>
<td>$T_o$ (°R), the randomly selected outer wall surface temperature, used to calculate $\Delta T$ (°R), the temperature difference across the wall of the duct, given in Equation 2-2.</td>
</tr>
<tr>
<td>TOMU</td>
<td>RE</td>
<td>Randomly selected Normal distribution parameter $\mu$ for the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TOMUA</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\mu$ in the Normal distribution of the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TOMUB</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\mu$ in the Normal distribution of the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TOSIG</td>
<td>RE</td>
<td>Randomly selected Normal distribution parameter $\sigma$ for the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TOSIGA</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\sigma$ in the Normal distribution of the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TOSIGB</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\sigma$ in the Normal distribution of the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TRUNC</td>
<td>RE</td>
<td>Value used to filter out noise in the principal stress-time history during rainflow cycle counting. See Section 2.2.1.4 of [1] for a discussion of rainflow cycle counting.</td>
</tr>
<tr>
<td>TSTAT</td>
<td>RE</td>
<td>$M_x$ (in.-lbf), the static torsional load component. Not used in the HEX coil or EXHEX analysis.</td>
</tr>
<tr>
<td>TYPE(MAXLD)</td>
<td>INT</td>
<td>1-D array containing the type of dynamic or time-varying load, used to assign the appropriate load scale factors. $\text{TYPE(<em>)} = 1$, use the narrow-band random load scale factor; $\text{TYPE(</em>)} = 2$, use the superimposed sinusoidal load scale factor; and $\text{TYPE(*)} = 3$, use the aerodynamic load factor.</td>
</tr>
<tr>
<td>V(2, MAXLD)</td>
<td>RE</td>
<td>2-D array containing the time-varying shear load components $V_y$ and $V_z$ (lbf). Not used in the HEX coil or EXHEX analysis.</td>
</tr>
<tr>
<td>VSTAT(2)</td>
<td>RE</td>
<td>1-D array containing the static shear load components $V_y$ and $V_z$ (lbf). Not used in the HEX coil or EXHEX analysis.</td>
</tr>
<tr>
<td>WIDTH</td>
<td>RE</td>
<td>$W$ (in.), the randomly selected plate width used to calculate the SIF for the EXHEX crack configuration.</td>
</tr>
<tr>
<td>WITHA</td>
<td>RE</td>
<td>Lower bound of Beta distribution for $W$.</td>
</tr>
<tr>
<td>WITHB</td>
<td>RE</td>
<td>Upper bound of Beta distribution for $W$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WITHR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for the width ( W ).</td>
</tr>
<tr>
<td>WITHR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \rho ) in the Beta distribution of ( W ).</td>
</tr>
<tr>
<td>WITHR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \rho ) in the Beta distribution of ( W ).</td>
</tr>
<tr>
<td>WITHT</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \theta ) for the width ( W ).</td>
</tr>
<tr>
<td>WITHT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the Beta distribution of ( W ).</td>
</tr>
<tr>
<td>WITHT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the Beta distribution of ( W ).</td>
</tr>
<tr>
<td>WOFF</td>
<td>RE</td>
<td>( W_{OFF} ) in Equation 2-3, the randomly selected Weld Offset (%).</td>
</tr>
<tr>
<td>WOFFA</td>
<td>RE</td>
<td>Lower bound of the first Beta distribution on ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFB</td>
<td>RE</td>
<td>Upper bound of the first Beta distribution on ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFC</td>
<td>RE</td>
<td>Lower bound of the second Beta distribution on ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFD</td>
<td>RE</td>
<td>Upper bound of the second Beta distribution on ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFE</td>
<td>RE</td>
<td>Decimal equivalent percentage weight occurring in the first Beta distribution of the weld offset ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFHI</td>
<td>RE</td>
<td>Upper bound of the randomly selected Beta distribution for the weld offset ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFLO</td>
<td>RE</td>
<td>Lower bound of the randomly selected Beta distribution for the weld offset ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for the weld offset ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \rho ) in the first Beta distribution of ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \rho ) in the first Beta distribution of ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFR3</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \rho ) in the second Beta distribution of ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFR4</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \rho ) in the second Beta distribution of ( W_{OFF} ).</td>
</tr>
</tbody>
</table>
Table 7.1-2  List of Variables For Program PROCRK (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>WOFFT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter ( \theta ) for the weld offset ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the first Beta distribution of ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the first Beta distribution of ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFT3</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the second Beta distribution of ( W_{OFF} ).</td>
</tr>
<tr>
<td>WOFFT4</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the second Beta distribution of ( W_{OFF} ).</td>
</tr>
</tbody>
</table>
### 7.1.4 Program PROCRK Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program PROCRK Listing Temporal Order</td>
<td>7-24</td>
</tr>
<tr>
<td>PROCRK</td>
<td>7-25</td>
</tr>
<tr>
<td>LIFCAL</td>
<td>7-31</td>
</tr>
<tr>
<td>BLKGRO</td>
<td>7-34</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-36</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-37</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-38</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-39</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-40</td>
</tr>
<tr>
<td>M4L1</td>
<td>7-40</td>
</tr>
<tr>
<td>M4L2</td>
<td>7-43</td>
</tr>
<tr>
<td>CYCOUN</td>
<td>7-47</td>
</tr>
<tr>
<td>NEUBER</td>
<td>7-51</td>
</tr>
<tr>
<td>HYPDRW</td>
<td>7-52</td>
</tr>
<tr>
<td>PARDRW</td>
<td>7-54</td>
</tr>
<tr>
<td>STRIF1</td>
<td>7-55</td>
</tr>
<tr>
<td>STRIF2</td>
<td>7-57</td>
</tr>
<tr>
<td>GRODAT</td>
<td>7-58</td>
</tr>
<tr>
<td>DETER4</td>
<td>7-64</td>
</tr>
<tr>
<td>INPUT</td>
<td>7-65</td>
</tr>
<tr>
<td>SETDEF</td>
<td>7-71</td>
</tr>
<tr>
<td>STRAN1</td>
<td>7-71</td>
</tr>
<tr>
<td>STRAN2</td>
<td>7-73</td>
</tr>
<tr>
<td>BETAQN</td>
<td>7-74</td>
</tr>
<tr>
<td>GAM</td>
<td>7-75</td>
</tr>
</tbody>
</table>

PROCRK Version 92.5
## Program PROCRK Listing Temporal Order

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCRK</td>
<td>7-25</td>
</tr>
<tr>
<td>SETDEF</td>
<td>7-71</td>
</tr>
<tr>
<td>INPUT</td>
<td>7-65</td>
</tr>
<tr>
<td>GRODAT</td>
<td>7-58</td>
</tr>
<tr>
<td>DETER4</td>
<td>7-64</td>
</tr>
<tr>
<td>HYPDRW</td>
<td>7-52</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-37</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-38</td>
</tr>
<tr>
<td>PARDRW</td>
<td>7-54</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-74</td>
</tr>
<tr>
<td>GAM</td>
<td>7-75</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-38</td>
</tr>
<tr>
<td>NORMAN</td>
<td>7-39</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-38</td>
</tr>
<tr>
<td>LIFCAL</td>
<td>7-31</td>
</tr>
<tr>
<td>STRAN1</td>
<td>7-71</td>
</tr>
<tr>
<td>M4L1</td>
<td>7-40</td>
</tr>
<tr>
<td>M4L2</td>
<td>7-43</td>
</tr>
<tr>
<td>STRAN2</td>
<td>7-73</td>
</tr>
<tr>
<td>CYCCOUN</td>
<td>7-47</td>
</tr>
<tr>
<td>NEUBER</td>
<td>7-51</td>
</tr>
<tr>
<td>BLKGRO</td>
<td>7-34</td>
</tr>
<tr>
<td>STRIF1</td>
<td>7-55</td>
</tr>
<tr>
<td>STRIF2</td>
<td>7-57</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-36</td>
</tr>
</tbody>
</table>
C PROCRK IS THE MAIN MODULE OF THE PROBABILISTIC CRACK GROWTH PROGRAM
C PROGRAMMER: S. SUTHAR
C
C THIS PROGRAM DRAWS MANY ROUTINES FROM PROGRAM HEXHCF (JPL PUB 92-15)
C
C DATE: DECEMBER 1992
C VERSION: 92.5
C
C Copyright (C) 1991, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.
C***************************************************************************

PROGRAM PROCRK
C=========================================================================
C SUBPROGRAMS: SETDEF, INPUT, GRODAT, HYPDRW, PARDRW, LIFCAL, INSORT
C
C FILES: 1:CRKDAT-OLD; 3:CRKRES-NEW; 8:IOUTPR-NEW;
9:LOWLIF-NEW; 11-26: user named-OLD
C
C IMPLICIT NONE
C
INTEGER MAXBLF, MAXLD, MAXLIF, MAXM, MAXSEG
C
PARAMETER (MAXBLF = 10, MAXLD = 16, MAXLIF = 1000, 
& MAXM = 20000, MAXSEG = 10)
C
INTEGER BLFPOS, I, IOUT, J, K, LOCAT, NBLIFE, NDIR, 
& NCRL, NHYPER, NLIFE, NLIFET, NLOAD, NRAN, 
& NUM, NUMSEG, TYPE(MAXLD)
C
INTEGER INEUB, IRET, KGROW, KPROB
C
DOUBLE PRECISION RAND
C
REAL AERD, AERDA, AERDB, AERS, AERSA, AERSB, AI, AIA, 
& AIT, AIT2, ANGLE, AOCB, AOCA, AOCR, AOCR1, 
& AOCR2, AOC'T, AOC'T2, ASTR, ASTR'A, ASTRB, 
& BLFPER(MAXBLF), CEE, CI, CO, 
& COEXP, DDKTHO, DLTAT, DPCMU, DPCSIG, DSTR, DSTRA, DSTRB, 
& DTIMU, DTISIG, DTOMU, DTOSIG, E(MAXSEG), EM, 
& EMM, ENN, EY, 
& INDIA, INDIAA, INDIAB, INDIR, INDIR1, INDIR2, 
& INDIT, INDIT1, INDIT2, K, 
& KLAM, KLAMA, KLAMB, LAMKH, LAMKA, LAMKB, 
& LAMKH', LAMKCA, LAMKCB, LAMGR, LAMG', LAMGRB, 
& LAMN, LAMNA, LAMB, LAMND, LAMMU, LAMNSG, 
& LAMS, LAMS', LAMS, LAMSD, LAMSN, LAMSSG, 
& LAMW, LAMWA, LAMWB, LIFE(MAXLIF)
C
REAL M(2, MAXLD), MSTAT(2), 
& MVAR, MVARA, MVARB, NEUB, NEUBA, NEUBB, NEWLIF, NU, 
& P(MAXLD), PC, PCMU, PCMUA, PCMU'B, PCO, PCSIG, PCSIGA, 
& PCSIGB, PEE, PERIOD, PSTAT, QUE, RSO, 
& SE(MAXSEG), SSTR, SSTR'A, 
& SSTRA, STHIS(MAXLD, MAXM), SX(MAXLD), SXST, 
& SXY(MAXLD), SXYST, SXST', SY(MAXLD), 
& SYST, SY2(MAXLD), SY2ST, SZ(MAXLD), SZST, 
& T(MAXLD), THIC, THICA, THICB, THICR, THICR1, 
& THICR2, THIC'T, THICT, THICT1, THICT2, TIMU, TIMUA, 
& TIMUB, TISIG, TISIGA, TISIGB, TOMU, 
& TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB, 
& TRUNC, TSTAT, V(2, MAXLD), 
& VSTAT(2), WIDTH, WITHA, 
& WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT, 
& WOFF, WOFFA, WOFFB, WOFFC, WOFFD, 
& WOFFE, WOFFH, WOFFL, WOFFR, WOFFR1, WOFFR2, WOFFR3, 
& WOFFR4, WOFFT, WOFFT1, WOFFT2, WOFFT3, WOFFT4
C
CHARACTER*6 LDNAME(MAXLIF)

7 - 25
COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
& P, T, M, V, PCO, SXST, SYST, SZST, SXST, SYST, SXZST, SYZST
& SXST, SYST, SX, SY, SZ, SK, SXZ, SYZ
COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA,
& AIT, AIT2, AIR, AIT,
& AOC, AOCB, AOCR, AOCR2, AOCR1, AOCR2, AOCR2, AOCR2, AOCR2,
& ASTRA, ASTRB,
& DPCM, DPCSIG, DSTR, DSTRB, DTMU, DTSIG, DTOM, DTOSIG,
& INDIA, INDIA, INDIA, INDIA, INDIA, INDIA, INDIA, INDIA,
& KLMA, KLMA, KLMA, KLMA, KLMA, KLMA, KLMA, KLMA,
& LAMN, LAMN, LAMN, LAMN, LAMN, LAMN, LAMN, LAMN,
& LAMSA, LAMSA, LAMSA, LAMSA, LAMSA, LAMSA, LAMSA, LAMSA,
& LAMWA, LAMWA, MVARA, MVARB, NEUB, NEUB,
& PCO, PCO, PCO, PCO, PCO, PCO, PCO, PCO,
& PCMU, PCMU, PCMU, PCMU, PCMU, PCMU, PCMU, PCMU,
& RAND,
& STRA, STRB,
& THICA, THICB, THICIC, THIC2, THIC3, THIC4, THICT, THICT2,
& TIMU, TIMUB, TIMU, TIMUB, TIMU, TIMUB, TIMU, TIMUB,
& TOMA, TOMUB, TOMA, TOMUB, TOMA, TOMUB, TOMA, TOMUB,
& WITZA, WITZB, WITZA, WITZB, WITZA, WITZB, WITZA, WITZB,
& WOFFA, WOFFB, WOFFA, WOFFB, WOFFA, WOFFB, WOFFA, WOFFB,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
THRESHOLD MODEL COEFFICIENT "d"

SIF RANGE FOR GROWTH RATE DATA

DESCRIPTION OF THE MATERIAL DATA

SIF RANGE

EFFECTIVE SIF RANGE AFTER RETARDATION

THRESHOLD SIF RANGE

THRESHOLD MODEL COEFFICIENT "DKtho"

SELECTED TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES -- \( \Delta T \) ( Ti - To)

EQUA TO PCMUB - PCMUA

EQUA TO PCSIGB - PCSIGN

EQUA TO TIMUB - TIMUA

EQUA TO TISIGB - TISIGA

EQUA TO TOMUB - TOMUA

EQUA TO TOSIGB - TOSIGA

1-D ARRAY WHICH CONTAINS THE STRAIN VALUES

EM COEFFICIENT "m" IN THE GENERALIZED FORMAN MODEL

ENN COEFFICIENT "n" IN THE GENERALIZED FORMAN MODEL

FAIL LOGICAL VARIABLE TO INDICATE UNSTABLE CRACK, K GREATER THAN Kcr

I-D ARRAY CONTAINING UNIT NUMBER FOR STRESS-TIME HISTORIES FILES

1-D ARRAY WITH \( F_k \) VALUES OF THE \( F_k \) VS. RT CURVE

FILE TEST -- USED TO TEST EXISTENCE OF FILE

FTY MATERIAL YIELD STRENGTH

INDIA SELECTED INTERIOR DIAMETER

INDIAB SELECTED INTERIOR DIAMETER LOWER BOUND

INDIR SELECTED RHO FOR INTERIOR DIAMETER

INDIR1 INTERIOR DIAMETER -- RH LOWER BOUND

INDIR2 INTERIOR DIAMETER -- RH UPPER BOUND

BETA HYPER-DISTRIBUTION PARAMETERS

INDIT SELECTED THETA FOR INTERIOR DIAMETER

INDIT1 INTERIOR DIAMETER - THETA LOWER BOUND

INDIT2 INTERIOR DIAMETER - THETA UPPER BOUND

NEUB NEUBER'S RULE CONTROL (1=INCLUDE, 0=EXCLUDE)

IOUT CONTROLS DUMP TO SCREEN/PRINTER

IRET WILLENBORG'S RETARDATION MODEL CONTROLLER (1=INCLUDE, 0=EXCLUDE)

IREGOP FORMAN EQUATION REGRESSION OPTION

KC CRITICAL STRESS INTENSITY FACTOR KC

KgROW GROWTH MODEL, GENERALIZED FORMAN COEFFICIENT m (CONST=1, VARY=2)

Klam STRESS INTENSITY FACTOR CALCULATION ACCURACY

KlamA SIF CALCULATION ACCURACY UNIFORM DISTRIBUTION LOWER BOUND

KlamB SIF CALCULATION ACCURACY UNIFORM DISTRIBUTION UPPER BOUND

Kmax MAXIMUM SIF

Kmaxef EFFECTIVE MAXIMUM SIF AFTER RETARDATION

Kmin MINIMUM SIF

Kminef EFFECTIVE MINIMUM SIF AFTER RETARDATION

Koff STRESS CONCENTRATION DUE TO WELD OFFSET

Kyab TYPE OF PROBLEM (HEX COIL = 1, EXHEX = 2)

Lamgr GROWTH CALCULATION ACCURACY FACTOR

Lamgra GROWTH CALCULATION ACCURACY UNIFORM DISTRIBUTION LOWER BOUND

Lamgrb GROWTH CALCULATION ACCURACY UNIFORM DISTRIBUTION UPPER BOUND

Lamkac CRITICAL SIF Kc UNCERTAINTY UNIFORM DISTRIBUTION LOWER BOUND

Lamkcb CRITICAL SIF Kc UNCERTAINTY UNIFORM DISTRIBUTION UPPER BOUND

Lamkha THRESHOLD SIF Kth UNCERTAINTY

Lamkhaa THRESHOLD SIF Kth UNCERTAINTY UNIFORM DISTRIBUTION LOWER BOUND

Lamkhaa SELECTED LAMBD FOR ONE SIGMA NARROW-BAND RANDOM LOADS

Lamkhaa LAMBDA FOR NARROW-BAND RANDOM LOADS -- LOWER BOUND OF k

Lamkhaa LAMBDA FOR NARROW-BAND RANDOM LOADS -- UPPER BOUND OF k

Lamkn LAMBDA FOR NARROW-BAND RANDOM LOADS COEFFICIENT OF VARIATION

Lamkdm NARROW-BAND RANDOM LOADS STRAIN GAGE ACCURACY FACTOR

Lamkdk LAMBDA FOR NARROW-BAND RANDOM LOADS k -- INDICATES VARIATION DUE TO SAMPLE SIZE

Lamkmu MEAN OF LAMBDA FOR NARROW-BAND RANDOM LOADS (MU, NORMAL DISTRIBUTION)

Lamksd STANDARD DEVIATION OF LAMBDA FOR NARROW-BAND RANDOM LOADS
C LAMSA SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS
C LAMSB LAMBDA FOR SUPERIMPOSED SINE LOADS -- LOWER BOUND OF k
C LAMSC LAMBDA FOR SUPERIMPOSED SINE LOADS -- UPPER BOUND OF k
C LAMSD LAMBDA FOR SUPERIMPOSED SINE LOADS COEFFICIENT OF VARIATION
C LAMSE SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS
C LAMSF LAMBDA FOR SUPERIMPOSED SINE LOADS STRAIN GAGE ACCURACY FACTOR
C LAMSG SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS k -- INDICATES VARIATION DUE TO SAMPLE SIZE
C LAMSMU MEAN OF LAMBDA FOR SUPERIMPOSED SINE LOADS (MU, NORMAL DISTRIBUTION)
C LAMSSG STANDARD DEVIATION OF LAMBDA FOR SUPERIMPOSED SINE LOADS (SIGMA, NORMAL DISTRIBUTION)
C LAMU SELECTED ACCURACY FACTOR FOR WELD ECCENTRICITY STRESS CONCENTRATION FACTOR, Koff
C LAMA LAMU LOWER BOUND
C LAMW LAMW UPPER BOUND
C LDNAME() 1-D ARRAY CONTAINING LOAD NAMES FOR THE TIME-VARYING LOADS
C LIFE() 1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED
C LOCAT -- SORTED VALUES OF THE LEFT-HAND TAIL
C M() 2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C M(1) ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C MAXBLF MAXIMUM NUMBER OF PERCENTAGE PROBABILITY LEVELS
C MAXDAT MAXIMUM NUMBER OF POINTS PER DATA DIVISION ALLOWED
C MAXDIV MAXIMUM NUMBER OF DATA DIVISIONS ALLOWED
C MAXLD MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C MAXLIF MAXIMUM NUMBER OF CRACK GROWTH LIVES ALLOWED
C MAXM MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C MAXSEG MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN CURVE)
C MI MOMENT OF INERTIA FOR DUCT
C MLAM() 2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS SCALED
C BY DSTR OR ASTR AND LAMSA, LAMSB, OR AERD AS APPROPRIATE
C (INDICATED BY TYPE()) -- MLAM(1,*) ARE THE M2 LOADS;
C MLAM(2,*) ARE THE M3 LOADS
C MSLAM() 1-D ARRAY CONTAINING THE STATIC LOADS SCALED BY ASTR
C AND AERS OR SSTR AS APPROPRIATE -- MSLAM(1) IS THE M2 LOAD;
C MSLAM(2) IS THE M3 LOAD
C MSTAT() 1-D ARRAY CONTAINING THE STATIC LOADS -- MSTAT(1) IS THE M2 LOAD;
C MSTAT(2) IS THE M3 LOAD
C MVAR SELECTED FORMAN COEFFICIENT m
C MVARA FORMAN COEFFICIENT m UNIFORM DISTRIBUTION LOWER BOUND
C MVARB FORMAN COEFFICIENT m UNIFORM DISTRIBUTION UPPER BOUND
C NBIN(100) 1-D ARRAY CONTAINING THE NUMBER OF CYCLES AFTER RF COUNTING
C NBLIFE NUMBER OF BLIVES TO BE CALCULATED
C NCRRL NUMBER OF CRACK LENGTHS FROM A1 TO A5 TO DO GROWTH INTEGRATION
C NDIR NUMBER OF DEGREES OF FREEDOM FOR CRACK GROWTH (1 OR 2)
C NDIV NUMBER OF DIVISIONS OF GROWTH RATE DATA
C NEUB SELECTED NEUBER'S RULE MODEL ACCURACY FACTOR
C NEUBA NEUB UNIFORM DISTRIBUTION LOWER BOUND
C NEUBB NEUB UNIFORM DISTRIBUTION UPPER BOUND
C NEWLIF LIFE VALUE RETURNED FROM CALL TO LIFCAL
C NHYPY NUMBER OF SETS OF HYPERPARAMETER DISTRIBUTIONS TO BE
C SAMPLED FROM
C NLIFE NUMBER OF DUCT FAILURE LIVES TO BE CALCULATED
C NLIFET TOTAL NUMBER OF LIVES CALCULATED
C NLOAD NUMBER OF TIME-VARYING LOADS
C NORM RANDOM VARIABLE (SOMETIMES UNIFORM, SOMETIMES NORMAL) USED
C TO OBTAIN SELECTED TEMPERATURES AND PRESSURE
C NP() 1-D ARRAY CONTAINING NUMBER OF POINTS PER DATA DIVISION
C FOR CRACK GROWTH RATE DATA
C NRAN NUMBER OF POINTS IN STRESS-TIME HISTORY
C NU POISSON'S RATIO
C NUMSEG NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
C P() 1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C P1 SELECTED INTERNAL PRESSURE, PSI
C PC SELECTED INTERNAL PRESSURE, PSI
C PCMU SELECTED MEAN OF NORMALLY DISTRIBUTED INTERNAL PRESSURE
C PCMU1 MEAN OF INTERNAL PRESSURE LOWER BOUND
C PCMUB MEAN OF INTERNAL PRESSURE UPPER BOUND
C PCO EXTERNAL PRESSURE, PSI
C PCS1G SELECTED STANDARD DEVIATION OF NORMALLY DISTRIBUTED
C INTERNAL PRESSURE
C PCS1GA STANDARD DEVIATION OF INTERNAL PRESSURE LOWER BOUND
C PCS1GB STANDARD DEVIATION OF INTERNAL PRESSURE UPPER BOUND
C PEE COEFFICIENT "p" IN THE GENERALIZED FORMAN MODEL

7 - 28
LENGTH OF TIME IN SECONDS OF STRESS-TIME HISTORY


1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS SCALED

BY DSTR OR ASTR AND LAMS, LAMN, OR AERD, AS APPROPRIATE (INDICATED BY TYPE(*))

STATIC AXIAL LOAD SCALED BY ASTR AND AERS OR SSTR AS

APPROPRIATE

STATIC AXIAL LOAD

COEFFICIENT "g" IN THE GENERALIZED FORMAN MODEL

RANDOM NUMBER SEED

STRESS RATIO R FOR GROWTH RATE DATA

EFFECTIVE STRESS RATIO AFTER RETARDATION

INNER RADIUS FOR DUCT

OUTSIDE RADIUS FOR DUCT

RATIO R/t

WILLENBOURG RETARDATION MODEL CONSTANT

THE STRESS-STRAIN VS STRAIN CURVE

PRINCIPAL STRESS HISTORY (MAXM)

SELECTED STATIC STRESS ANALYSIS ACCURACY

UNIFORM DISTRIBUTION LOWER BOUND

UNIFORM DISTRIBUTION UPPER BOUND

1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES

1-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES

STRESS-TIME HISTORIES

1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAX STRESS COMPONENT

1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAXY STRESS COMPONENT

1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAXZ STRESS COMPONENT

1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAY STRESS COMPONENT

1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAYZ STRESS COMPONENT

1-D ARRAY FOR TIME-VARYING MAGNITUDE OF SIGMAZ STRESS COMPONENT

I-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS

I-D ARRAY CONTAINING THE TYPE OF TIME-VARYING LOAD, USED FOR LOAD FACTORS -- TYPE(*) = 1 INDICATES NARROW-BAND RANDOM;
TYPE(*) = 2 INDICATES SUPERIMPOSED SINUSOID; TYPE(*) = 3 INDICATES AERO DYNAMIC

V() 2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*) ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS

VSTAT() 1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- VSTAT(1) IS THE V2 LOAD; VSTAT(2) IS THE V3 LOAD

WIDTH SELECTED PLATE WIDTH, IN

WIDTHB WIDTH UPPER BOUND

WIDTHR SELECTED RHO FOR WIDTH

WIDTHR1 WIDTH - RHO LOWER BOUND PARAMETERS

WIDTHR2 WIDTH - RHO UPPER BOUND

WIDTHT SELECTED THETA FOR WIDTH

WIDTHT1 WIDTH - THETA LOWER BOUND

WIDTHT2 WIDTH - THETA UPPER BOUND

WOFF SELECTED WELD OFFSET (%)

WOFFA WELD OFFSET LOWER BOUND - HYPERDISTRIBUTION 1

WOFFT WELD OFFSET - THETA LOWER BOUND - HYPERDISTRIBUTION 1

WOFFR SELECTED RHO FOR WELD OFFSET

WOFFR1 WELD OFFSET - RHO LOWER BOUND - HYPERDISTRIBUTION 1

WOFFR2 WELD OFFSET - RHO UPPER BOUND - HYPERDISTRIBUTION 2

WOFFR3 WELD OFFSET - RHO LOWER BOUND - HYPERDISTRIBUTION 2

WOFFR4 WELD OFFSET - RHO UPPER BOUND - HYPERDISTRIBUTION 2

WOFFT SELECTED THETA FOR WELD OFFSET

WOFFT1 WELD OFFSET - THETA LOWER BOUND - HYPERDISTRIBUTION 1

WOFFT2 WELD OFFSET - THETA UPPER BOUND - HYPERDISTRIBUTION 1

WOFFT3 WELD OFFSET - THETA LOWER BOUND - HYPERDISTRIBUTION 2

WOFFT4 WELD OFFSET - THETA UPPER BOUND - HYPERDISTRIBUTION 2

--- OPEN THE INPUT AND OUTPUT FILES ---

OPEN (1, FILE = 'CRKDAT', STATUS = 'OLD')
OPEN (3, FILE = 'CRKRES', STATUS = 'NEW')
OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')

--- SET DEFAULT VALUES ---

CALL SETDEF (LIFE, NCRL)

--- READ AND ECHO GENERAL DATA ---

CALL INPUT (ANGLE, BLFPER, COEXP, E, EM, & LOCAT, NBLIFE, NHYPER, NLIFE, NLIFET, NRAN, & NU, NUMSEG, PERIOD, RSO, SE, STRHIS, TRUNC)

--- READ MATERIAL PROPERTIES AND PERFORM REGRESSION ON CRACK GROWTH DATA ---

CALL GRODAT (CEE, CO, DEE, DKTHO, EMM, ENN, FTY, KC, PEE, QUE)

--- FOR HEX COIL GROW CRACK IN TWO DIRECTIONS BUT FOR EXHEX ONLY ONE ---

IF(KPROB .EQ. 1) THEN
  NDIR = 2
ELSE
  NDIR = 1
ENDIF

--- THIS LOOP SAMPLES HYPERPARAMETER SETS ---

DO 300 K = 1, NHYPER
  CALL HYPDRW (AERD, AERS, ASTR, DSTR, KLAM, LAMGR, LAMKC, & LAMKH, LAMW, NEUB, SSTR, MVAR)
C IF COEFFICIENT m IS VARYING
IF(KGROW.EQ.2) THEN  
EMM = MVAR  
ENDIF

C >>>> THIS LOOP GENERATES CRACK GROWTH LIVES <<<<

DO 200 I = 1, NLIFE

C PERFORM DRIVER DRAWS

CALL PARDRW (AI, CI, DLTAT, INDIA, LAMN, LAMS, PC,  
& THIC, WIDTH, WOFF)

C PERFORM CRACK GROWTH LIFE CALCULATION

CALL LIFCAL (AERD, AERS, ASTR, AI, ANGLE, CI, CEE,  
& CO, COEXP, DEE, DKTHO, DLTAT, DSTR, E, EM,  
& EMM, ENN, FTY, INDIA, KC, KLAM, LAMGR, LAMKC,  
& LAMKH, LAMN, LAMS, LAMW, LOCAT, NEUB, NEWLIF,  
& NDIR, NCRL, NDIK, NNI, NUMSEG, PC, PEE, PERIOD,  
& QUE, RSO, SE, SSTR, STRHIS, THIC, TRUNC, WIDTH, WOFF)

C SAVE AND SORT THE SHORTEST 1% OF LIVES AFTER SORTING

IF (NLIFET .GT. i) THEN  
CALL INSORT (NEWLIF, LIFE, NLIFET)ENDIF

200 CONTINUE

300 CONTINUE

C WRITE OUT THE LIVES AND BLIVES

WRITE(3,1000)

IF (NLIFET .GT. 1) THEN
  NUM = NLIFET/100
  WRITE (3,1200)
  DO 400 I = 1, NUM
    WRITE(3,1100) LIFE(I)
  WRITE(3,1i00) LIFE(I)
  400 CONTINUE
  WRITE(3,1300)
  DO 500 J = 1, NBLIFE
    BLFPOS = NINT (BLFPER(J) * FLOAT(NLIFET))
    WRITE(3,1400) BLFPER(J), LIFE(BLFPOS)
  500 CONTINUE
ELSE
  WRITE(3,1500) NEWLIF
ENDIF

STOP

C---------------------------- FORMATS -----------------------------

1000 FORMAT(///,30X,'SIMULATION OUTPUT',///)
1100 FORMAT(20X,'E12.5')
1200 FORMAT(13X,'SHORTEST 1% OF CRACK GROWTH LIVES ',///,  
& ' E12.5')
1300 FORMAT(///,2X,'B LIVES: EMPirical',/)
1400 FORMAT(2X,F7.5,5X,E12.5)
1500 FORMAT(13X,'CRACK GROWTH LIFE = ',E12.5)
END

C***********************************************************************
C SUBROUTINE LIFCAL CALCULATES CRACK GROWTH LIFE
C***********************************************************************

C PROGRAMMER: S. SUTHARSHANA
C DATE: DECEMBER 1992
C VERSION: 92.5
C
C Copyright (C) 1991, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

C***********************************************************************
C SUBROUTINE LIFCAL (AERD, AERS, ASTR, AI, ANGLE, CI, CEE,  
& CO, COEXP, DEE, DKTHO, DLTAT, DSTR, E, EM,  
& EMM, ENN, FTY, INDIA, KC, KLAM, LAMGR, LAMKC,  
& LAMKH, LAMN, LAMS, LAMW, LOCAT, NEUB, NEWLIF,  
& NDIR, NCRL, NDIK, NNI, NUMSEG, PC, PEE, PERIOD,  
& QUE, RSO, SE, SSTR, STRHIS, THIC, TRUNC, WIDTH, WOFF)
C SUBPROGRAMS: STRAN1, STRAN2, CYCOUN, BLKGRO
C
IMPLICIT NONE

INTEGER J, JLAST, NBIN(100), MAXLD, MAXM, MAXSEG
PARAMETER (MAXLD = 16, MAXM = 20000, MAXSEG = 10)

C-------------------------- LOCAL VARIABLES --------------------------
REAL A(2), AF, AOC, DADB(2), DELA, DELC, DSALT,
& NEWA(101), PDADB, PC, PEE, PERIOD,
& QUE, RSO, SE, SSTR, STRHIS, THIC, TRUNC, WIDTH, WOFF
LOGICAL FAIL
C
A() CRACK LENGTH IN THE "a" AND "c" DIRECTIONS
C DELA CRACK LENGTH INCREMENT IN THE "a" DIRECTION
C DELC CRACK LENGTH INCREMENT IN THE "c" DIRECTION
C NEWA() ARRAY OF CRACK LENGTHS TO PERFORM BLOCK GROWTH CALCULATIONS AT
C PDADB PREVIOUS da/dB

C-------------------------- EXTERNAL VARIABLES INPUT AND OUTPUT --------------------------
INTEGER INEUB, IRET, KGROW, KPROB
INTEGER IOUT, LOCAT, NDIR, NCRL, NLOAD, NRAN,
& NUMSEG, TYPE(MAXLD)
REAL AERD, AERS, ASTR, AI, ANGLE, CI, CEE, CO,
& COEXP, DLTAT, DSTR, E(MAXSEG), EM,
& EMM, ENN, FTY, INDIA, KC, KLAM,
& LAMKR, LAMKC, LAMKH, LAMN, LAMS, LAMW, M(2,MAXLD),
& MSTAT(2), NEUB, NEWLIF, NU, PC, PCO, PEE,
& PERIOD, PSTAT, QUE, RSO, SE(MAXSEG), SSTR,
& STRHIS(MAXLD,MAXM),
& SX(MAXLD), SXST, SY(MAXLD), SYST, SZ(MAXLD),
& SXST, SYST, SZST, SXY(MAXLD), SXYST, SXZ(MAXLD),
& SXZST, SYZ(MAXLD), SYZST, SZ(MAXLD),
& T(MAXLD), THIC, TRUNC, TSTAT, V(2,MAXLD), VSTAT(2),
& WIDTH, WOFF
CHARACTER*6 LDNAME(MAXLD)
COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
COMMON/NAMES/LDNAME
COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
& P, T, M, V, PCO,
& SXST, SYST, SZST, SXYST, SXZST, SYZST,
& SX, SY, SZ, SKY, SXZ, SYZ
COMMON IOUT

C PERFORM LOAD TO STRESS TRANSFORMATION
IF (KPROB .EQ. 1) THEN
  AF = THIC
  CALL STRAN1 (AERD, AERS, ASTR, ANGLE, COEXP, DLTAT, DSTR, EM,
& INDIA, LAMK, LAMN, LAMS, LAMW, LOCAT, NRAN, NU, PC,
& SPR, STRHIS, THIC, WOFF)
ELSEIF (KPROB .EQ. 2) THEN
  AF = WIDTH/2.0
  CALL STRAN2 (DSTR, LAMK, LAMN, LAMS, NRAN, SPR, SSTR, STRHIS)
ENDIF

C PERFORM CYCLE COUNTING
CALL CYCOUN (DSALT, E, EM, NBIN, NEUB, NUMSEG, NRAN, SE,
& SPR, SM, TRUNC)

C ESTABLISH CRACK LENGTHS AT WHICH BLOCK GROWTH CALCULATIONS ARE PERFORMED
NEWA(1) = AI
DELA = EXP( LOG(AF/AI)/FLOAT(NCRL) )
DO 50 J = 1, NCRL
   NEWA(J+1) = NEWA(J) * DELA
CONTINUE
A(1) = NEWA(1)
TOTTIF = 0.0
FAIL = .FALSE.
JLAST = 1
IF(NDIR .EQ. 2) THEN
   A(2) = CI
ENDIF
PDADB = 0.0
PDCDB = 0.0
C CALCULATE CRACK-GROWTH LIFE FOR THE LOAD BLOCK
C >>>> THIS LOOP IS FOR EVERY CRACK LENGTH
DO 100 J = I, NCRL
   DADB(1) = 0.0
   DADB(2) = 0.0
   CALL BLKGRO (A, CEE, CO, DADB, DEE, DKTHO, DGALT, EMM, ENN, FAIL, FTY, INDIA, KC, KLAM, & LAMKC, LAMKH, NBIN, NDIR, PEE, & QUE, RSO, SM, THIC, WIDTH)
   IF(IOUT .EQ. 20) THEN
      WRITE(8,*)(A(1), A(2), DADB(1), DADB(2)
      IF(NDIR .EQ. 2) THEN
         AOC = A(1)/A(2)
         WRITE(8,*) AOC
      ENDIF
   ENDIF
   IF(PDADB .GT. 0.0) THEN
      DELA = NEWA(J) - NEWA(J-1)
      TOTTIF = 2.0*DELA/(DADB(1) + PDADB) + TOTTIF
   ELSEIF(PDCDB .GT. 0.0) THEN
      TOTTIF = 2.0*DELC/(DADB(2) + PDCDB) + TOTTIF
   ENDIF
   IF (DADB(1) .GT. 0.0) THEN
      A(1) = NEWA(J+1)
      IF(NDIR .EQ. 2) THEN
         RATIO = DADB(2)/DADB(1)
         DELC = (NEWA(J+1) - NEWA(J)) * RATIO
         A(2) = A(2) + DELC
      ENDIF
      IF (FAIL) THEN
         FAIL = .FALSE.
         WRITE(8,*),'K GT Kcr AT A = ',A(1)
         GO TO ii0
      ENDIF
      JLAST = J
   ELSE
      IF(FAIL) THEN
         FAIL = .FALSE.
         WRITE(8,*),'K GT Kcr AT A = ',A(1)
         GO TO ii0
      ELSE
         IF(NDIR .EQ. 2 .OR. A(2) .GT. WIDTH/2.0 .OR. DADB(2) .EQ. 0.0) THEN
            TOTTIF = 1.0E+37
            WRITE(8,*),'NO GROWTH AT', J, 'th CRACK LENGTH'
            GO TO ii0
         ENDIF
      ENDIF
      JLAST = J
   ELSE
      IF(NDIR .EQ. 2) THEN
         DELC = A(2)*EXP(LOG(WIDTH/(2.0*A(2))))/
               FLOAT(NCRL-J+1)) - 1.0)
         A(2) = A(2) + DELC
         A(1) = NEWA(JLAST)
      ENDIF
ENDIF
C 7 - 33
SUBROUTINE BLKGRO (A, CEE, CO, DADB, DEE, DKTHO, DSALT, & EMM, ENN, FAIL, FTY, INDIA, KC, KLAM, & LAMKC, LAMKH, NBIN, NDIR, PEE, QUE, RSO, SM, & THIC, WIDTH)
C SUBPROGRAMS: STRIF1, STRIF2
C
C PROGRAMMER: S. SUTHARSHA
C
C DATE: DECEMBER 1992
C VERSION: 92.5
C
C Copyright (C) 1991, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
C*************************************************************************

AB(1) = A(1)
AB(2) = A(2)
AO(1) = 0.0
AO(2) = 0.0
RPO(1) = 0.0
RPO(2) = 0.0

IF(IOUT .EQ. 20) THEN
  WRITE(*,*) 'INSIDE BLKGRO ROUTINE'
  WRITE(*,*) 'A, C', A(1), A(2)
ENDIF

C CALCULATE THE STRESS INTENSITY FACTOR COEFFICIENTS

CONST = SQRT(PI*A(1)) * KLAM

IF(KPROB .EQ. 1) THEN
  CALL STRIF1(A(I), A(2), F0, F2, INDIA, THIC)
ELSEIF(KPROB .EQ. 2) THEN
  CALL STRIF2(A(I), F0, F2, WIDTH)
ENDIF

C LOOP FOR EVERY STRESS CYCLE IN HISTORY
C AND LOOP FOR 'a' DIRECTION (=1) AND 'c' DIRECTION (=2)

SALMAX = DSALT*I01.0

DO 200 I = 1, 100
  IF (NBIN(I) .GT. 0) THEN
    SALT = SALMAX - FLOAT(I)*DSALT
  DO 100 IDIR = I, NDIR
    KMAX(IDIR) = CONST*(F0(IDIR)*SALT + F2(IDIR)*SM)
    KMIN(IDIR) = CONST*(-F0(IDIR)*SALT + F2(IDIR)*SM)
  ENDIF
  IF(IOUT .EQ. 20) THEN
    WRITE(*,*) 'DIRECTION = ', IDIR
    WRITE(*,*) 'DIR, KMAX1,2, KMIN1,2', IDIR, KMAX(IDIR), KMIN(IDIR)
  ENDIF
  IF(KMAX(IDIR).LE.0.0) THEN
    GO TO 95
  ENDIF
  KMAXE(I) = KMAX(IDIR) - KMIN(IDIR)
  KEFF = KMIN(IDIR)/KMAX(IDIR)
  IF (IRET .EQ. 1) THEN
    AOP = AO(IDIR) + RPO(IDIR)
    RPI = PLSR(IDIR) * (KMAX(IDIR)/FTY)**2
    ARI = AB(IDIR) + RPI
    IF(ARI .GT. AOP) THEN
      RPO(IDIR) = RPI
      AO(IDIR) = AB(IDIR)
    ELSE
      AORPA = AORPA - AB(IDIR)
      KMAX = FTY*(AORPA/PLSR(IDIR))**(0.5)
      KMINEF = KMIN(IDIR) - (KMAX - KMAX(IDIR))/((R0-1.0)
      IF(KMAXE.GT.0.0) THEN
        IF(KMINEF.GT.0.0) THEN
          KEFF = KMAX - KMINEF
          KEFF = KMINEF/KMAXE
        ELSE
          DKEFF = KMAXE
          KEFF = 0.0
        ENDIF
      ELSE
        DKEFF = KMAXE
        KEFF = 0.0
      ENDIF
    ENDIF
  ENDIF
ENDIF

7 - 35
C CALCULATE BLOCK CRACK GROWTH RATE AND NEW CRACK LENGTH
IF(IOUT .EQ. 20) THEN
WRITE(8,*) 'CYC, DIR, DKEFF, REFF ',I,IDIR,DKEFF,REFF
ENDIF
C CHANGE FROM PSI TO KSI
DK = DKEFF/1000.0
C CALCULATE CRACK GROWTH
IF( REFF .LE. 0.9 ) THEN
DKTH=LAMKH*DKTHO*(1.0 - CO*REFF)**DEE
ELSE
DKTH=LAMKH*(DKTHO*(1.0 - 0.90*CO)**DEE)*10.0*(1.0 - REFF)
ENDIF
KCR = KC * LAMKC
KMAXEF = KMAXEF/1000.0
IF ( (DK .GT. DKTH) .AND. (KMAXEF .LE. KCR) ) THEN
DA = FLOAT(NBIN(I)) * CEE * ((1. - REFF)**EMM) * (DK)**ENN
& / ( (1. - REFF) * KCR - DK )**QUE
DADB(IDIR) = DADB(IDIR) + DA
IF(IOUT .EQ. 20) THEN
WRITE(8,*)'DIR, DA, DADB, AB ',IDIR, DA,
& DADB(IDIR),AB(IDIR)
ENDIF
ELSEIF (KMAXEF .GT. KCR ) THEN
FAIL = .TRUE.
RETURN
ENDIF
95 CONTINUE
100 CONTINUE
END

C SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C PROGRAMMER: L. NEWLIN
C DATE: 12MAY88
C VERSION: 2
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)
C INPUTS: NEWLIF, LIFE, NLIFET
C OUTPUTS: LIFE
C IMPLICIT NONE
COMMON IOUT

C====================================================================
INTEGER I, IOUT, MAXLIF, NLIFET, NUM, PLACE
PARAMETER (MAXLIF = 1000)
REAL LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

LIST OF VARIABLES
I CONTROLS DO LOOP FOR INSERTION
IOUT OUTPUT DUMP CONTROLLER
LIFE() 1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
PFM TO BE SORTED
MAXLIF MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
CALCULATION
NEWLIF LIFE VALUE TO BE INSERTED INTO LIFE()
NLIFET TOTAL NUMBER OF LIVES CALCULATED BY PFM
NUM NUMBER OF LIFE VALUES IN LIFE()
PLACE POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()

NUM = NLIFET / 100
C FIND POSITION IN LIFE() FOR NEWLIF
IF (NEWLIF .GT. LIFE(NUM)) GOTO 400
DO 100 I = 1, NUM
   IF (NEWLIF .LT. LIFE(I)) THEN
      PLACE = I
      GOTO 110
   ENDIF
100 CONTINUE
110 CONTINUE
C STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()
DO 200 I = (PLACE + 1), NUM
   TEMP(I) = LIFE(I-1)
200 CONTINUE
C INSERT NEWLIF
LIFE(PLACE) = NEWLIF
C SHIFT VALUES OF LIFE() FOLLOWING NEWLIF
DO 300 I = (PLACE + 1), NUM
   LIFE(I) = TEMP(I)
300 CONTINUE
C IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN
400 CONTINUE
RETURN
END

*******************************************************************************
SUBROUTINE PRYRV(RAND, RHO1, RHO2, THE1, THE2, X, Y)
COMMON IOUT 7 - 37
DOUBLE PRECISION RAND

REAL FRAC, RHO1, RHO2, THE1, THE2, X, Y
INTEGER IOUT

CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE (*,*) 'FRAC = ', FRAC
X = FRAC * (RHO2 - RHO1) + RHO1

CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE (*,*) 'FRAC = ', FRAC
Y = FRAC * (THE2 - THE1) + THE1

IF (IOUT .EQ. 15) WRITE (*,*) 'RHO1 = ', RHO1, ' RHO1 = ', RHO1, ' RHO2 = ', RHO2,

RETURN
END

C SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C Random Number Generator, JPL Publication 85-98, JPL Document
C
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: IDEC 87
C VERSION: MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C V4.3, V4.4, V4.5
C***********************************************************************

SUBROUTINE RANDOM (FRAC, RAND)
C IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL FRAC
DOUBLE PRECISION RANA, RANC, RANDIV, RANM, RANSUB,
& RANT, RANX

LIST OF VARIABLES
UNIFORM (0,1) RANDOM VARIATE
OUTPUT DUMP CONTROLLER
CONSTANT FOR LCG
CONSTANT FOR LCG
RANDOM NUMBER SEED
INTERNAL CALCULATION
CONSTANT FOR LCG
INTERNAL CALCULATION
INTERNAL CALCULATION
INTERNAL CALCULATION

USING LCG RANDOM # GENERATOR

RANA = 671093.0
RANC = 7090885.0
RANM = 33554432.0
10 RANX = RANA * RAND + RANC
RANDIV = RANX / RANM
RANT = DINT(RANDIV)
RANSUB = RANT * RANM
RAND = RANX - RANSUB
FRAC = SNGL(RAND / RANM)

IF (((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0))) GOTO 10

7 - 38
IF (IOUT .EQ. 2) WRITE(8,*)'RANX =,, RANX, ' RANDIV =', RANDIV, & RANT =', RANT, ' RANSUB =,, RANSUB, ' RAND =', RAND, & ' FRAC =,, FRAC
RETURN
END

C NOTES: IOUT=2 DUMPS TO SCREEN

***********************************************************************
C SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE : 3FEB 88
C The random variates are generated using the "Direct Method"
C Abramowitz, M., and Stegun, I. A., editors, Handbook of
C Mathematical Functions, National Bureau of Standards, Applied
C Mathematics Series 55, Issued June 1964, Ninth Printing, November
C 1970 with corrections, pg. 953.

C**********************************************************************
SUBROUTINE NORMGN (RAND, MU, SIGMA, X)
C SUBPROGRAM: RANDOM
C IMPLICIT NONE
COMMON IOUT
DOUBLE PRECISION RAND
REAL FRAC, MU, PI, SIGMA, X, UI, U2, Z1, Z2
PARAMETER (PI = 3.1415926536)
INTEGER IOUT

LIST OF VARIABLES
FRAC UNIFORM(0,1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
MU MEAN OF NORMAL DISTRIBUTION
RAND RANDOM NUMBER SEED
SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION
X NORMAL RANDOM VARIATE
U1 UNIFORM RANDOM NUMBER U(0,1)
U2 UNIFORM RANDOM NUMBER U(0,1)
Z1 NORMAL RANDOM NUMBER ON N(0,1)
Z2 NORMAL RANDOM NUMBER ON N(0,1)

IF (IOUT .EQ. 15) & WRITE(8,*') 'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA
CALL RANDOM (FRAC, RAND)
U1 = FRAC
CALL RANDOM (FRAC, RAND)
U2 = FRAC
IF (IOUT .EQ. 15) & WRITE(8,*') 'U1 =', U1, ' U2 =', U2
Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)
X = SIGMA * Z1 + MU
IF (IOUT .EQ. 15) & WRITE(8,*') 'Z1 =', Z1, ' Z2 =', Z2, ' X =', X
RETURN
END
C SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C PROGRAMMER: L. NEWLIN
C DATE: 5OCT87
C**********************************************************************
SUBROUTINE TRMNAT

WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'
STOP
END
C**********************************************************************

C SUBROUTINE M4L1 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
C FOR LOCATION 1 (PLAIN WELD, EXTERIOR SURFACE OF THE DUCT) UNDER THERMAL
C LOADING
C PROGRAMMER: L. NEWLIN
C DATE: JUL92
C VERSION: 92.4
C**********************************************************************

SUBROUTINE M4L1 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT,
& NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP,
& T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT)
C
C INPUTS: ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT, NLOAD, NU, P,
C PC, PCO, PSTAT, STATIC, T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT
C
C OUTPUTS: STATIC, STRAMP
C
C IMPLICIT NONE
C
COMMON IOUT

INTEGER I, IOUT, J, MAXLD, NLOAD
REAL PI
PARAMETER (MAXLD = 16, PI = 3.1415926536)
REAL ALPHA, ANGLE, AREA, DI, DLTAT, EM, FK(10), GEOM, IFK,
& K(2, 2), KOFF, LAMW, M(2, MAXLD), NI, MSTAT(2), NU,
& P(MAXLD), PC, PCO, PSTAT, RDIFF, R1, R12, RO, RO2,
& ROT, RT(10), SIGIA(MAXLD), SIGIB(MAXLD), STAT, STR2,
& STATIC(4), STRA, STR1, STR2A, STR1B, STR2B, STRIC,
& STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD),
& VSTAT(2), WOFF

LIST OF VARIABLES

ALPHA COEFFICIENT OF THERMAL EXPANSION
ANGLE ANGLE THETA IN RADIANS
AREA CROSS SECTION AREA OF DUCT WALL
DI INTERIOR DIAMETER
DLTAT TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
EM YOUNG'S MODULUS PRIOR TO YIELD
FK() 1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
GEOM INTERMEDIATE THERMAL STRESS CALCULATION VARIABLE
I CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND
AERODYNAMIC LOADS
IFK INTERPOLATED VALUE OF Fk CORRESPONDING TO THE VALUE OF r/t
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING
INTERPOLATION
K() FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT
EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR
FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL
KOFF, THE STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF THE WELD

RI = DI / 2.0
ROT = (DI + THIC) / (2.0 * THIC)

DO 50 J = 2, 10
  INTERPOLATE TO FIND FACTOR FK CORRESPONDING TO VALUE OF r/t
  IF (((ROT .LE. RT(J)) .AND. (ROT .GE. RT(J-1))) THEN
    IFK = ((FK(J) - FK(J-1)) * (ROT - RT(J-1)))
      / (RT(J) - RT(J-1)) + FK(J-1)
  ENDIF
50 CONTINUE

KOFF = LAMW * (1.0 + 3.0 * IFK * WOFF)

IF (IOUT .GE. 25) THEN
  WRITE(8,'(A10,2X,5F10.4)') 'DI = ', DI, ' RI = ', RI
  WRITE(8,'(A10,2X,5F10.4)') 'THIC = ', THIC, ' WOFF = ', WOFF
END
WRITE(8,*) 'LAMW = ', LAMW, ' KOFF = ', KOFF
ENDIF

C CALCULATE THE CROSS-SECTIONAL AREA AND MOMENT OF INERTIA

AREA = PI * ((RI + THIC) ** 2 - RI ** 2)
MI = PI * ((RI + THIC) ** 4 - RI ** 4) / 4.0

C OBTAIN STRESS CONCENTRATION FACTORS AND RADIUS APPROPRIATE TO LOCATION

SKT1 = K(1,1)
SKT2 = K(2,1)
RO = RI + THIC

IF (IOUT.EQ.25) THEN
  WRITE(8,*) 'AREA = ', AREA, ' MI = ', MI
  WRITE(8,*) 'K(1,1) = ', K(1,1), ' SKT1 = ', SKT1
  WRITE(8,*) 'K(2,1) = ', K(2,1), ' SKT2 = ', SKT2
  WRITE(8,*) 'THIC = ', THIC, ' RO = ', RO
  WRITE(8,*) 'ALPHA = ', ALPHA, ' NU = ', NU
  WRITE(8,*) 'DLTAT = ', DLTAT, ' EM = ', EM
  WRITE(8,*)
ENDIF

RI2 = RI ** 2
RO2 = RO ** 2
RDIFF = RO2 - RI2

GEOM = 1.00 - 2.00 * LOG (RO / RI) * RI2 / RDIFF

C TEMPERATURE STRESS

STHMA = ((EM * ALPHA * DLTAT) / (2.00 * (1.00 - NU) * LOG (RO / RI))) * GEOM

C AXIAL STRESS CALCULATIONS

STRIA = PSTAT / AREA
STRIB = (MSTAT(1) * COS (ANGLE) + MSTAT(2) * SIN (ANGLE)) * RO
& / MI
STRIC = (PC - PCO) * RI2 / RDIFF

STATIC(1) = (STRIA + STRIB + STRIC) * SKT1 * KOFF + STHMA

C HOOP (2) AND RADIAL (3) STRESS CALCULATIONS

STR2A = 2.0 * PC * RI2 / RDIFF
STR2B = - PCO * (RO2 + RI2) / RDIFF

STATIC(2) = (STR2A + STR2B) * SKT2 + STHMA

STATIC(3) = - PCO

C SHEAR STRESS

STATIC(4) = TSTAT * RO / (2.0 * MI) - (2.0 / AREA)
& * (VSTAT(1) * COS (ANGLE) + VSTAT(2) * SIN (ANGLE)))

IF (IOUT.EQ.25) THEN
  WRITE(8,*) 'RO2 = ', RO2, ' RI2 = ', RI2
  WRITE(8,*) 'RDIFF = ', RDIFF, ' GEOM = ', GEOM
  WRITE(8,*) 'STATIC STRESS VALUES'
  WRITE(8,*) 'AXIAL STRESSES'
  WRITE(8,*) 'STRIA = ', STRIA, ' STRIB = ', STRIB
  WRITE(8,*) 'STRIC = ', STRIC, ' STHMA = ', STHMA
  WRITE(8,*) 'STATIC(1) = ', STATIC(1)
  WRITE(8,*) 'HOOP STRESSES'
  WRITE(8,*) 'STRA2 = ', STR2A, ' STR2B = ', STR2B
  & STHMA = ', STHMA
  WRITE(8,*) 'STATIC(2) = ', STATIC(2)
  WRITE(8,*) 'RADIAL STRESS', STATIC(3) = ', STATIC(3)
  WRITE(8,*) 'SHEAR STRESS', STATIC(4) = ', STATIC(4)
ENDIF
 Routine M4L2 performs the calculations necessary to find the stress for location 2 (plain weld, interior surface of the duct), under thermal loading.

**INPUTS:**
- ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT
- NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP
- T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT

**OUTPUTS:**
- STATIC, STRAMP

---

Routine M4L2

```plaintext
C
C AXIAL STRESS CALCULATIONS
C
SIGA(I) = P(I) / AREA
SIGIB(I) = (M(I,1) * COS (ANGLE) + M(2,I) * SIN (ANGLE)) * RO / MI
STAMP(1,I) = (SIGA(I) + SIGIB(I)) * STI1 * KOFF
C
C HOOP (2) AND RADIAL (3) STRESSES ARE ZERO
C BECAUSE PRESSURES ARE CONSTANT
STAMP(2,I) = 0.0
STAMP(3,I) = 0.0
C
C SHEAR STRESS
STAMP(4,I) = T(I) * RO / (2.0 * MI) - (2.0 / AREA * (V(I,1) * COS (ANGLE) + V(2,I) * SIN (ANGLE)))
C
IF (IOUT.EQ.25) THEN
WRITE(8,*), 'STRESS VALUES FOR I = ',I
WRITE(8,*), 'AXIAL STRESSES'
WRITE(8,*), 'SIGA = ', SIGA(I), ' SIGIB = ', SIGIB(I)
WRITE(8,*), 'STAMP(1,I) = ', STAMP(1,I)
WRITE(8,*), 'HOOP STRESSES', 'STAMP(2,I) = ',STAMP(2,I)
WRITE(8,*), 'RADIAL STRESS', 'STAMP(3,I) = ',STAMP(3,I)
WRITE(8,*), 'SHEAR STRESS', 'STAMP(4,I) = ',STAMP(4,I)
ENDIF
END
```

---

**SUBROUTINE M4L2**

- Performs the calculations necessary to find the stress for location 2 (plain weld, interior surface of the duct), under thermal loading.
- **INPUTS:**
  - ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT
  - NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP
  - T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT
- **OUTPUTS:**
  - STATIC, STRAMP

**Version:** 92.4

---

**Date:** Jul92

---

**Programmer:** L. Newlin

---

**Version:** 92.4

---

**Inputs:**
- ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT
- NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP
- T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT

**Outputs:**
- STATIC, STRAMP

---

**Notes:**
- Routine M4L2 calculates the stresses at location 2 for a plain weld on the interior surface of the duct under thermal loading.
IMPLICIT NONE

COMMON IOUT

INTEGER I, IOUT, J, MAXLD, NLOAD

REAL PI

PARAMETER (AXLD = 16, PI = 3.1415926536)

REAL ALPHA, ANGLE, AREA, DLTAT, EM, FK(10), GEOM, IFK, DI,
  K(2, 2), KOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU,
  P(MAXLD), PC, PCO, PSTAT, RDIFF, RI, R12, RO, RO2,
  RT, RT(10), SIG1A(MAXLD), SIG1B(MAXLD), SKT1, SKT2,
  STRA, STRB, STR2A, STR2B, STRIC,
  STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD),
  VSTAT(2), WOFF

LIST OF VARIABLES

ALPHA COEFFICIENT OF THERMAL EXPANSION
ANGLE ANGLE THETA IN RADIANS
AREA CROSS SECTION AREA OF DUCT WALL
DI INTERIOR DIAMETER
DLTAT TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
EM YOUNG'S MODULUS PRIOR TO YIELD
FK() 1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
       CONCENTRATION DUE TO WELD ECCENTRICITY
GEOM INTERMEDIATE THERMAL STRESS CALCULATION VARIABLE
I CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND
       AERODYNAMIC LOADS
IFK INTERPOLATED VALUE OF Fk CORRESPONDING TO THE VALUE OF r/t
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING
       INTERPOLATION
K() FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT
       EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR
       FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL
       DIRECTION; K(2,2) IS FOR DUCT INTERIOR FOR HOOP DIRECTION
KOFF STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
LAMW ACCURACY FACTOR OF Fk - r/t CURVE
M() 2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
       ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
MAXLD MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
MI MOMENT OF INERTIA
MSTAT() 1-D ARRAY CONTAINING THE STATIC TIME-VARYING LOADS -- M(1,*)
       ARE THE M2 LOAD; M(2,*) IS THE M3 LOAD
NLOAD NUMBER OF TIME-VARYING LOADS
NU POISSON'S RATIO
P() 1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
PC LIMIT PRESSURE ON INSIDE OF THE VESSEL
PCO LIMIT PRESSURE ON OUTSIDE OF THE VESSEL
PI SELF EXPLANATORY CONSTANT
PSTAT STATIC AXIAL LOAD
RDIFF EQUAL TO RO2 - R12
RI INTERIOR RADIUS
R12 INNER RADIUS SQUARED
RO OUTER RADIUS
RO2 OUTER RADIUS SQUARED
RT EQUAL TO r / t (R Over T)
RT() 1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
       CONCENTRATION DUE TO WELD ECCENTRICITY
SIG1A() 1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO FORCE
       FOR THE TIME-VARYING LOADS
SIG1B() 1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO BENDING
       FOR THE TIME-VARYING LOADS
SKT1 STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
SKT2 STRESS CONCENTRATION FACTOR FOR HOOP STRESS
STATIC() 1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES --
       STATIC(1) IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS;
       STATIC(3) IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
STHMA THE STATIC AXIAL STRESS DUE TO THERMAL GRADIENT
STRIA THE STATIC AXIAL STRESS DUE TO FORCE
THE STATIC AXIAL STRESS DUE TO BENDING
THE STATIC AXIAL STRESS DUE TO MOMENTUM CHANGE (FLUID)
THE STATIC Hoop STRESS AT OUTER SURFACE DUE TO INTERNAL PRESSURE
THE STATIC Hoop STRESS AT OUTER SURFACE DUE TO EXTERNAL PRESSURE
2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
-- STRAMP(I,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
STRAMP(4,*) ARE THE SHEAR STRESSES
1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
WALL THICKNESS AT DUCT OUTER RADIUS
1-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*) ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2 LOAD; V(2) IS THE V3 LOAD
WELD OFFSET
CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF THE WELD
RI = DI / 2.0
ROT = (DI + THIC) / (2.0 * THIC)
DO 50 J = 2, 10
IF (ROT .LE. RT(J)) .AND. (ROT .GE. RT(J-I)) THEN
IFK = (FK(J) - FK(J-I)) * (ROT - RT(J-I)) / (RT(J) - RT(J-I)) + FK(J-I)
ENDIF
KOFF = LAMW * (1.0 + 3.0 * IFK * WOFF)
IF (IOUT .EQ. 25) THEN
WRITE(8,*) 'DI = ', DI, ' RI = ', RI
WRITE(8,*) 'THIC = ', THIC, ' ROT = ', ROT
WRITE(8,*) 'IFK = ', IFK, ' WOFF = ', WOFF
WRITE(8,*) 'LAMW = ', LAMW, ' KOFF = ', KOFF
ENDIF
CALCULATE THE CROSS-SECTIONAL AREA AND MOMENT OF INERTIA
AREA = PI * ((RI + THIC) ** 2 - RI ** 2)
MI = PI * ((RI + THIC) ** 4 - RI ** 4) / 4.0
OBTAIN STRESS CONCENTRATION FACTORS AND RADIUSES APPROPRIATE TO LOCATION
THIS IS THE INTERIOR SURFACE
SKT1 = K(1,2)
SKT2 = K(2,2)
RO = RI + THIC
IF (IOUT .EQ. 25) THEN
WRITE(8,*) 'AREA = ', AREA, ' MI = ', MI
WRITE(8,*) 'K(1,2) = ', K(1,2), ' SKT1 = ', SKT1
WRITE(8,*) 'K(2,2) = ', K(2,2), ' SKT2 = ', SKT2
WRITE(8,*) 'THIC = ', THIC, ' RO = ', RO
WRITE(8,*) 'ALPHA = ', ALPHA, ' NU = ', NU
WRITE(8,*) 'DLTAT = ', DLTAT, ' EM = ', EM
ENDIF
RI2 = RI ** 2
RO2 = RO ** 2
RDIFF = RO2 - RI2
GEOM = 1.00 - 2.00 * LOG (RO / RI) * RO2 / RDIFF
TEMPERATURE STRESS
STHMA = ((EM * ALPHA * DLTAT) / (2.00 * (1.00 - NU)

C
AXIAL STRESS CALCULATIONS

\[
\begin{align*}
\text{STRIA} &= \frac{\text{PSTAT}}{\text{AREA}} \\
\text{STRIB} &= (\text{MSTAT}(1) \times \cos(\text{ANGLE}) + \text{MSTAT}(2) \times \sin(\text{ANGLE})) \times \frac{\text{RI}}{\text{MI}} \\
\text{STRIC} &= (\text{PC} - \text{PCO}) \times \frac{\text{RI}2}{\text{RDIFF}} \\
\text{STATIC}(1) &= (\text{STRIA} + \text{STRIB} + \text{STRIC}) \times \text{SKTI} \times \text{KOFF} + \text{STHMA}
\end{align*}
\]

HOOP (2) AND RADIAL (3) STRESS CALCULATIONS

\[
\begin{align*}
\text{STR2A} &= \text{PC} \times (\text{RI}2 + \text{RO2}) / \text{RDIFF} \\
\text{STR2B} &= -2.0 \times \text{PCO} \times \text{RO2} / \text{RDIFF} \\
\text{STATIC}(2) &= (\text{STR2A} + \text{STR2B}) \times \text{SKT2} + \text{STHMA}
\end{align*}
\]

SHEAR STRESS

\[
\begin{align*}
\text{STATIC}(4) &= \text{TSTAT} \times \text{RI} / (2.0 \times \text{MI}) - (2.0 / \text{AREA}) \times (\text{VSTAT}(1) \times \cos(\text{ANGLE}) + \text{VSTAT}(2) \times \sin(\text{ANGLE}))
\end{align*}
\]

IF (IOUT.EQ.25) THEN

\[
\begin{align*}
&\text{WRITE}(8,*) 'RO2 = ', \text{RO2}, ' RI2 = ', \text{RI2} \\
&\text{WRITE}(8,*) 'RDIFF = ', \text{RDIF}, ' GEOM = ', \text{GEOM} \\
&\text{WRITE}(8,*) 'AXIAL STRESSES' \\
&\text{WRITE}(8,*) 'STRIA = ', \text{STRIA}, ' STRIB = ', \text{STRIB} \\
&\text{WRITE}(8,*) 'STRIC = ', \text{STRIC}, ' STHMA = ', \text{STHMA} \\
&\text{WRITE}(8,*) 'HOOP STRESSES' \\
&\text{WRITE}(8,*) 'STR2A = ', \text{STR2A}, ' STR2B = ', \text{STR2B} \\
&\text{WRITE}(8,*) 'STHMA = ', \text{STHMA} \\
&\text{WRITE}(8,*) 'RADIAL STRESS' \\
&\text{WRITE}(8,*) 'STRAMP(1,I) = ', \text{STRAMP}(1,I) \\
&\text{WRITE}(8,*) 'STRAMP(2,I) = ', \text{STRAMP}(2,I) \\
&\text{WRITE}(8,*) 'STRAMP(3,I) = ', \text{STRAMP}(3,I) \\
&\text{WRITE}(8,*) 'STRAMP(4,I) = ', \text{STRAMP}(4,I)
\end{align*}
\]
ENDIF

DO 100 I = 1, NLOAD

AXIAL STRESS CALCULATIONS

\[
\begin{align*}
\text{SIGIA}(I) &= \frac{\text{P}(I)}{\text{AREA}} \\
\text{SIGIB}(I) &= (\text{M}(1,I) \times \cos(\text{ANGLE}) + \text{M}(2,I) \times \sin(\text{ANGLE})) \times \frac{\text{RI}}{\text{MI}} \\
\text{STRAMP}(1,I) &= (\text{SIGIA}(I) + \text{SIGIB}(I)) \times \text{SKTI} \times \text{KOFF}
\end{align*}
\]

HOOP (2) AND RADIAL (3) STRESSES ARE ZERO

BECAUSE PRESSURES ARE CONSTANT

\[
\begin{align*}
\text{STRAMP}(2,I) &= 0.0 \\
\text{STRAMP}(3,I) &= 0.0
\end{align*}
\]

SHEAR STRESS

\[
\begin{align*}
\text{STRAMP}(4,I) &= \text{T}(I) \times \text{RI} / (2.0 \times \text{MI}) - (2.0 / \text{AREA}) \times (\text{V}(1,I) \times \cos(\text{ANGLE}) + \text{V}(2,I) \times \sin(\text{ANGLE}))
\end{align*}
\]

IF (IOUT.EQ.25) THEN

\[
\begin{align*}
&\text{WRITE}(8,*) 'STRESS VALUES FOR I = ', I \\
&\text{WRITE}(8,*) 'AXIAL STRESSES' \\
&\text{WRITE}(8,*) 'SIGIA = ', \text{SIGIA}(I), ' SIGIB = ', \text{SIGIB}(I) \\
&\text{WRITE}(8,*) 'STRAMP(1,I) = ', \text{STRAMP}(1,I) \\
&\text{WRITE}(8,*) 'HOOP STRESSES' \\
&\text{WRITE}(8,*) 'STRAMP(2,I) = ', \text{STRAMP}(2,I) \\
&\text{WRITE}(8,*) 'RADIAL STRESS' \\
&\text{WRITE}(8,*) 'STRAMP(3,I) = ', \text{STRAMP}(3,I) \\
&\text{WRITE}(8,*) 'SHEAR STRESS', ' STRAMP(4,I) = ', \text{STRAMP}(4,I)
\end{align*}
\]

WRITE(8,*)

7 - 46
ENDIF
100 CONTINUE

IF (IOUT .EQ. 25) THEN
  WRITE(8,*) 'I AXIAL HOOP RADIAL SHEAR'
  WRITE(8,*) STATIC(1), STATIC(2), STATIC(3), STATIC(4)
  DO 300 I = 1, NLOAD
      WRITE(8,*) I, STRAMP(1,I), STRAMP(2,I), STRAMP(3,I),
             STRAMP(4,I)
  300 CONTINUE
ENDIF

RETURN
END

C******************************************************************************

SUBROUTINE CYCOUN (DSALT, E, EM, NBIN, NEUB, NUMSEG, M, & SE, SPR, SM, TRUNC)
C INPUTS: E, EM, M, NEUB, NUMSEG, SE, SPR, SM, TRUNC
C
C OUTPUTS: SPR, DSALT, NBIN, SM
C SUBPROGRAM: NEUBER
C IMPLICIT NONE
COMMON IOUT
INTEGER MAXM, MAXSEG
PARAMETER (MAXM = 20000, MAXSEG = 10)
INTEGER I, INDEX(MAXM), IOUT, J, JMAX, K, M, N, & NBIN(100), NEWTOT, NUMSEG, OVER
INTEGER INEUB, IRET, KGROW, KPROB
REAL DSALT, E(MAXSEG), EE(MAXM), EM, HIGH, & LOW, NEUB, NEUBER, S(MAXM), SALTF, & SE(MAXSEG), SPR(MAXM), SEFMAX, & SM, SMEANF, SMAX, SMIN, SP(MAXM), & TEST1(MAXM), TEST2(MAXM), TRUNC
COMMON/CNTRL/INEUB, IRET, KGROW, KPROB

C LIST OF VARIABLES
C input variables:
C E() 1-D ARRAY CONTAINING THE STRAIN VALUES
C EM() YOUNG'S MODULUS BEFORE YIELD
C IOUT OUTPUT DUMP CONTROLLER
C M TOTAL NUMBER OF STRESS DATA POINTS PER PERIOD
C MAXM MAXIMUM NUMBER OF POINTS IN STRESS TIME HISTORY
C MAXSEG MAXIMUM NUMBER OF SEGMENTS ALLOWED
C NUMREG NUMBER OF REGIONS OF INTEREST
C NUMSEG NUMBER OF SEGMENTS OF INTEREST
C SE() 1-D ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS

7 - 47
SPR(M) PRINCIPAL STRESS HISTORY

TRUNC VALUE USED TO FILTER OUT NOISE

intermediate variables:

EE() HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
INDEX() COUNTER FOR EFFECTIVE STRESSES
I,J,K COUNTERS FOR VARIOUS DO LOOPS
JMAX INDEX (LOCATION) OF SEFMAX IN SPR()
N NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
NEUB NEUBER’S RULE MODEL ACCURACY FACTOR
NEUBER FUNCTION TO CALCULATE EQUIVALENT MEAN STRESS
NEWTOT TOTAL NUMBER OF EFFECTIVE STRESS VALUES AFTER FILTERING OVER
OVER FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
S(NEWTOT) FILTERED EFFECTIVE STRESSES
SALTF ALTERNATING STRESS FOR LARGEST STRESS RANGE CYCLE
SEFMAX LARGEST EFFECTIVE STRESS
SM SM = EQUIVALENT MEAN STRESS
SMEANF MEAN STRESS FOR LARGEST STRESS RANGE CYCLE
SP(M+1) RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
TEST1() 1-D ARRAY USED IN FILTERING THE STRESSES
TEST2() 1-D ARRAY USED IN FINDING CYCLES DURING RAINFLOW ANALYSIS

dump input data
if (iout.eq.30) then
write(8,'(9(1x,' cycoun inputs'
write(8,*','m :','m
WRITE(8,*)'EM :','EM,' TRUNC :','TRUNC,' NEUB :','NEUB
WRITE(8,*)'E():', E
WRITE(8,*)'SE():', SE
WRITE(8,*)'NUMSEG :', NUMSEG
write(8,*','
endif

C INITIALIZE ARRAYS
DO 50 I = 1, MAXM
SP(I) = 0.0
S(I) = 0.0
EE(I) = 0.0
INDEX(I) = 0
TEST1(I) = 0.0
TEST2(I) = 0.0
50 CONTINUE

SM = 0.0

C******* BEGIN RESEQUENCE ***************
C RESEQUENCE effective stresses (needed for rainflow analysis);
C largest effective stress is placed at beginning and end of SP(M+1)

C find SEFMAX, the largest sigma,eff, and JMAX, its location within SPR(M)

SEFMAX = -1.0E+20
DO 200 I=1,M
IF (SPR(I) .GT. SEFMAX ) THEN
SEFMAX = SPR(I)
JMAX = I
ENDIF
200 CONTINUE

C assign all points from JMAX out, to the beginning of SP()
DO 210 I = 1, M-JMAX+1
J = JMAX-1 + I
SP(I) = SPR(J)
210 CONTINUE

C assign points before JMAX to the end of SP()
J = 0
DO 220 I = M-JMAX+2, M
J = J + 1
SP(I) = SPR(J)
220 CONTINUE

SP(M+1) = SPR(JMAX)
if (iout.eq.30) then
write(8,'(sefmax:',sefmax,', jmax:',jmax
write(8,')' 'sp(m+l):(',(sp(i),i=1,m+l)

C****************************** E N D R E S Q

U

C****************************** B E G I N F I L T E R

DO 300 I = 2, M
    TEST1(I) = SP(I-1) - SP(I)
    TEST2(I) = TEST1(I) * (SP(I) - SP(I+1))
300 CONTINUE

if (iout .eq. 30) then
    do 305 i = 2, M
        write(8,'(test1 = ', test1(i), ' test2 = ', test2(i)
    305 continue
endif

K = 1
INDEX(1) = 1
DO 310 I = 2, M
    IF ( (TEST1(I) .NE. 0).AND. (TEST2(I) .LT. 0)) THEN
        K = K + 1
        INDEX(K) = I
   ENDIF
310 CONTINUE

NEWTOT = K + 1
INDEX(NEWTOT) = M + 1
DO 320 I = 1, NEWTOT
    S(I) = SP(K)
320 CONTINUE

if (iout.eq.30) then
    write ( 8, * ) ' newtot :,, newtot
    write(8,*), s(newtot) :,, (s (i), i=1,newtot)
endif

C************************************************************************** END FILTER**************************************************************************

C************************************************************************** B E G I N R A I N F L O W**************************************************************************

C RAINFLOW ANALYSIS to identify cycles within effective stress data,
C S(NEWTOT); places each cycle's max and min values into SPR(N)
C counters: I counts # of cycles found, J counts how many S()'s counted,
C K accumulates unmatched points

I = 0
J = 0
K = 0
400 CONTINUE

J = J+1
K = K+1

C check J to avoid reading beyond end of filtered stress data
IF ( J .GT. NEWTOT ) GOTO 499

C read stress point into a holding array to be checked for cycles
EE(K) = S(J)

410 IF ( K .LT. 3 ) GOTO 400
    IF (ABS (EE(K) - EE(K-1)) .LT. ABS (EE(K-1) - EE(K-2))) GOTO 400
C if not, then a cycle has been found, but we need to check for truncation

7 - 49
IF (ABS (EE(K-1) - EE(K-2)) .GT. TRUNC) THEN

C cycle is large enough to save

I = I+1
SMAX = AMAX1( EE(K-1), EE(K-2) )
SMIN = AMIN1( EE(K-1), EE(K-2) )
SMEANF = (SMAX + SMIN) / 2.0
SPR(I) = SMAX - SMEANF

ENDIF

C discard points K-1 and K-2, and decrement the counter of unmatched points

EE(K-2) = EE(K)  
K = K-2
C return for more counting
GOTO 410

499 CONTINUE

C N equals the final number of cycles found
N = I
if (iout.eq.25) then
write(8,*)'N :',n
write(8,*)'spr(n):'
do 12 i=1,n
write(8,*) spr(i)
12 continue
endif

IF (N .EQ. 0) THEN
C TRUNCATION FILTER TOO LARGE -- NO CYCLES LEFT
GOTO 710
ENDIF

C********************** E N D R A I N F L O W **********************

C calculate alternating and mean effective stresses for the largest
C stress cycle

SALTF = SPR(N)
C Assign the stress cycles to bins

DSALT = SALTF/100.0
LOW = SALTF + DSALT/2.0
DO 510 I= 1, 100
HIGH = LOW
LOW = LOW - DSALT
NBIN(I) = 0
DO 500 J=1, N
IF( (SPR(J) .GT. LOW) .AND. (SPR(J) .LT. HIGH) ) THEN
NBIN(I) = NBIN(I) + 1
ENDIF
500 CONTINUE
510 CONTINUE
if (iout.eq.25) then
write(8,*)'saltf :',saltf
write(8,*)'smeanf:',smeanf
endif

C******* Determine Equivalent Mean Stress, SM(N) for the largest cycle ****
C OVER = 0
C We are calculating the equivalent mean stress using neuber’s rule
C SM is the equivalent mean stress

7 - 50
IF (INEUB .EQ. i) THEN
    SM = NEUBER (EM, SALT, SMEAN, NUMSEG, E, SE, NEUB, OVER)
ELSE
    SM = SMEAN
ENDIF
710 CONTINUE
RETURN
END

C NEUBER USES NEUBER'S RULE AND THE STRESS-STRAIN CURVE TO CALCULATE THE
C THE MEAN STRESS. PROGRAM ASSUMES THAT THE STRESS STRAIN CURVE IS
C PIECEWISE LINEAR WITH AT MOST FIVE SECTIONS.
C
C PROGRAMER: L. NEWLIN
C DATE: 13SEP88
C VERSION: 92.1, 92.2, 92.3, 92.4, 92.5
C
FUNCTION NEUBER (EM, SALT, SMEAN, NUMSEG, E, SE, NEUB, OVER)
C INPUTS: EM, SALT, SMEAN, NUMSEG, SE, E, NEUB, OVER
C OUTPUTS: NEUBER
C
C IMPLICIT NONE
COMMON IOUT
INTEGER I, IOUT, MAXSEG, NUMSEG, OVER
PARAMETER (MAXSEG = 10)
REAL E(MAXSEG), EM, EPSLON, NEUB, NEUBER, PRODCT, SALT,
& SE(MAXSEG), SMEAN, ST,
& TEMP
LIST OF VARIABLES
E() STRAIN VALUES FOR EACH SEGMENT
EM YOUNG'S MODULUS BEFORE YIELD
EPSLON CALCULATED STRAIN (WHERE PLASTIC=ELASTIC DEFORMATION)
I CONTROLS DO LOOP FOR EACH SEGMENT
IOUT OUTPUT DUMP CONTROLLER
MAXSEG MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN)
NEUB NEUBER'S RULE MODEL ACCURACY FACTOR
NEUBER TOTAL EQUIVALENT MEAN STRESS
NUMSEG NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
OVER FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
PRODCT STRESS STRAIN PRODUCT (WHERE PLASTIC=ELASTIC DEFORMATION)
SALT TOTAL ALTERNATING STRESS
SE() 1-DIMENSIONAL ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
FOR A MULTI-SEGMENT CURVE
SMEAN MEAN STRESS
ST UNI-AXIAL TOTAL STRESS
TEMP TEMPORARY VARIABLE FOR NEUBER

TEMP = 0.00
ST = SALT * SMEAN / (ABS (SMEAN)) + SMEAN
PRODCT = NEUB * (ST ** 2) / EM

IF (PRODCT .LE. SE(1)) THEN
    TEMP = SMEAN
ELSE
    DO 800 I = 1, (NUMSEG - 1)
        IF (PRODCT .LT. SE(I)) THEN
            EPSLON = E(I) + ((E(I+1) - E(I)) / (SE(I+1) - SE(I))) * (PRODCT - SE(I))
        ENDIF
    ENDIF
    TEMP = PRODCT / EPSLON - SALT
800 CONTINUE
RETURN
END
ENDIF
ENDIF
CONTINUE
ENDIF
IF (ABS(TEMP) .LT. 1.0E-04) THEN
OVER = 1
WRITE(8,*) 'THE VALUE PRODCT EXCEEDED STRESS-STRAIN CURVE'
ENDIF
TEMP = TEMP * ABS(ST) / ST
NEUBER = TEMP
IF (IOUT .EQ. 25) THEN
WRITE(8,*) 'VALUES FROM NEUBER'
WRITE(8,* ' INPUT VALUES'
WRITE(8,* ' EM = ', EM, ' NEUB = ', NEUB, ' OVER = ', OVER
WRITE(8,* ' SALT = ', SALT, ' SMEAN = ', SMEAN
WRITE(8,* ' CALCULATED VALUES'
WRITE(8,* ' ST = ', ST, ' PRODUCT = ', PRODCT
WRITE(8,* ' EPSLON = ', EPSLON, ' NEUBER = ', TEMP
ENDIF
RETURN
END
C SUBROUTINE HYPDRW PERFORMS THE RANDOM VARIABLE DRAWS IN THE OUTER
C HYPERPARAMETER LOOP
C
C PROGRAMMER : S. SUTHARSHANA
C DATE : 25 JAN 1989
C VERSION : 92.1, 92.2, 92.3, 92.4, 92.5
C
C Copyright (C) 1991, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
C*****************************************************************************
SUBROUTINE HYPDRW (AERD, AERS, ASTR, DSTR, KLAM, LAMGR, LAMKC, & LAMKH, LAMW, NEUB, SSTR, MVAR)
SUBPROGRAMS: PRYRV, RANDOM
C
C IMPLICIT NONE
INTEGER INEUB, IRET, KGROW, KPROB
INTEGER IOUT
DOUBLE PRECISION &
AERD, AERDA, AERDB, AERS, AERSA, AERSB, AIA, AIB,
& AIR1, AIR2, AIT1, AIT2, AIR, AIT,
& AOCCA, AOCCB, AOCCR1, AOCCR2, AOCT1, AOCT2, AOCCR, AOCT,
& ASTR, ASTR, ASTRB, 
& DPCMU, DPCSIG, DSTR, DSTRA, DSTRB, DTMU, DTSIG,
& & DTMU, DTSIG, DUM, INDIAA, INDIA,
& & INDII, INDII, INDII, INDII, INDII, INDII,
& & KLAM, KLAMA, KLAMB, LAMKH, LAMKH, LAMKH,
& & LAMKC, LAMKCA, LAMKCB, LAMGR, LAMGRA, LAMGRB,
& & LAMRA, LAMNB, LAMNC, LAMND,
& & LAMNK, LAMMNU, LAMNSG, LAMSA, LAMSB, LAMSC, LAMSD,
& & LAMSK, LAMSMU, LAMSSG,
& & LAMW, LAMWA, LAMNB, MVAR, MVARA, MVARB
& REAL NEUB, NEUBA, NEUBB, NORM, PCMU, PCMU, PCMUB, PCSIM, &
& PCSIGA, PCSIGB, SSTRA, SSTRA, 
& & SSTRB, TEST, THICA, THICB,
& & THICK, THICK, THICK, THICT1, THICT2, TIMU,
& & TIMUB, TIMUB, TIMUB, TIMUB, TIMUB, TIMUB,
& & TOSIG, TOSIG, TOSIG, TOSIGB, WITHA,
& & WITBA, WITBA, WITBA, WITBA, WITBA, WITBA,
7 - 52
C START MAKING THE RANDOM DRAWS

CALL PRYRV (RAND, AIR1, AIR2, AIT1, AIT2, AIR, AIT)
IF (KPROB .EQ. 1) THEN
   CALL RANDOM (TEST, RAND)
   IF (TEST .LE. WOFFE) THEN
      CALL PRYRV (RAND, WOFFR1, WOFFR2, WOFFT1, WOFFT2, WOFFR, WOFFT)
      WOFFLO = WOFFA
      WOFFHI = WOFFB
   ELSE
      CALL PRYRV (RAND, WOFFR3, WOFFR4, WOFFT3, WOFFT4, WOFFR, WOFFT)
      WOFFLO = WOFFC
      WOFFHI = WOFFD
   ENDIF
ELSEIF (KPROB .EQ. 2) THEN
   CALL PRYRV (RAND, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT)
ENDIF

CALL PRYRV (RAND, IJTMNA, LAMNB, LAMSA, LAMSB, LAMSK)
LAMMU = LAMND / (1.0 + LAMNK * LAMNC)
LAMNSG = LAMNC / (1.0 + LAMSK * LAMSC)
LAMSSG = LAMSC / (1.0 + LAMSK * LAMSC)
IF (IOUT .EQ. 15) THEN
   WRITE(8,*) 'LAMNK = ', LAMNK, ' LAMNMU = ', LAMNMU, ' LAMSSG = ', LAMSSG
ENDIF

IF (KPROB .EQ. 1) THEN
   CALL RANDOM (NORM, RAND)
   TIMU = TIMUA + NORM * DTIMU
TISIG = TISIGA + NORM * DTISIG
TOMU = TOMUA + NORM * DTOMU
TOSIG = TOSIGA + NORM * DTOSIG
PCMU = PCMUA + NORM * DPCMU
PCSIG = PCSIGA + NORM * DPCSIG
ENDIF

IF (IOUT .EQ. 15) THEN
  WRITE (8,*) 'NORM = ', NORM
  WRITE (8,*) 'TIMU = ', TIMU, ' TISIG = ', TISIG
  WRITE (8,*) 'TOMU = ', TOMU, ' TOSIG = ', TOSIG
  WRITE (8,*) 'PCMU = ', PCMU, ' PCSIG = ', PCSIG
ENDIF

CALL PRYRV (RAND, LAMKHA, LAMKHB, LAMKCA, LAMKCB, LAMKH, LAMKC)
CALL PRYRV (RAND, LAMWA, LAMWB, NEUBA, NEUBB, LAMW, NEUB)
CALL PRYRV (RAND, DSTRA, DSTRB, KLAMA, KLAMB, DSTR, KLAM)
CALL PRYRV (RAND, SSTRA, SSTRB, DUM, DUM, SSTR, DUM)
CALL PRYRV (RAND, LAMGRA, LAMGRB, MVARA, MVARB, LAMGR, MVAR)
LAMGR = EXP (LAMGR)
RETURN
END

C*****************************************************************************
C SUBROUTINE PARDRW PERFORMS THE RANDOM LIFE DRIVER PARAMETER DRAWS IN
C THE INNER LOOP
C
C PROGRAMMER : S. SUTHARSHANA
C DATE : DECEMBER 1992
C VERSION : 92.5
C
C Copyright (C) 1991, California Institute of Technology.
C U.S. Government Sponsorship under NASA contract NAS7-918
C is acknowledged.
C*****************************************************************************

SUBROUTINE PARDRW (AI, CI, DLTAT, INDIA, LAMN, LAM, PC, & THIC, WIDTH, WOFF)

C SUBPROGRAMS : BETAGN, NORMGN

C IMPLICIT NONE

INTEGER INEUB, IRET, KGROW, KPROB

INTEGER IOUT

DOUBLE PRECISION RAND

REAL AERDA, AERDB, AERSA, AERSB, AI, AIA, AIB, AIR1, AIR2, & AIT1, AIT2, AIR, AIT, AOC, AOCB, AOCR1, AOCR2, & AOCR3, AOCR4, AOCR5, AOCR6, AOCR7, AOCR8, AOCR9, & CI, DLTAT, DPCMU, DPCSIG, DDTIMU, DTOSIG, & DTOMU, DTOSIG, DTMU, DTISIG, & DTMU, DTOSIG, INDIA, INDIAA, & INDIB, INDIR, INDIR1, INDIR2, INDIT, INDIT1, INDIT2, & KLAMA, KLAMB, LAMKHA, LAMKHB, & LAMKCA, LAMKCB, LAMGR, LAMGRB, & LAMN, LAMMA, LAMB, LAMNC, LAMND, & LAMNU, LAMS, LAMS, LAMSC, LAMSD, & LAMSW, LAMSS, LAMWA, LAMWB, MVARA, MVARB

REAL NEUBA, NEUBB, PC, PCMUA, PCMU, PCSIG, & PCSIGA, PCSIGB, SSTR, SSTRB, & THIC, THICA, THICB, THICR, THICRI, THICR1, THICR2, THICR3, THICT, THICT1, & THICT2, TIMU, TIMUA, TIMUB, TIN, TISIG, TISIGA, TISIGB, & TOMU, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB, TOUT, & WIDTH, WITHA, WITHB, WITHR1, WITHR2, & WITHT1, WITHT2, WITHT, WITHT, WOFF, & WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFH, WOFFL, & WOFFR, WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR5, WOFFT, WOFFT1, & WOFFT2, WOFFT3, WOFFT4

COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA, & AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT, & AOC, AOCB, AOCR1, AOCR2, AOCR3, AOCR4, AOCR5, AOCR6, AOCR7, AOCR8, AOCR9,
CALL BETAGN (RAND, AIR, AIT, AIA, AIB, AI)

IF (KPROB .EQ. 1) THEN
  CALL BETAGN (RAND, WOFFR, WOFFT, WOFFLO, WOFFHI, WOFF)
  CALL BETAGN (RAND, THICR, THICT, THICA, THICB, THIC)
  CALL BETAGN (RAND, AOCR, AOCT, AOCA, AOCB, AOC)
  CI = AI/AOC
ELSEIF (KPROB .EQ. 2) THEN
  CALL BETAGN (RAND, WITHER, WITHT, WITHA, WITHB, WIDTH)
ENDIF

CALL NORMGN (RAND, LAMNMU, LAMNSG, LAMN)
CALL NORMGN (RAND, LAMSMU, LAMSSG, LAMS)

IF (KPROB .EQ. 1) THEN
  CALL NORMGN (RAND, TIMU, TISIG, TIN)
  CALL NORMGN (RAND, TOMU, TOSIG, TOUT)
  DLAT = TIN - TOUT
  CALL NORMGN (RAND, PCMU, PCSIG, PC)
ENDIF

IF (IOUT .EQ. 15) THEN
  WRITE(8,*) 'AI=', AI, 'AOC=', AOC, 'CI=', CI
  WRITE(8,*) 'LAMN=', LAMN, 'LAMS=', LAMS
  WRITE(8,*) 'THIC=', THIC
  WRITE(8,*) 'INDIA=', INDIA, 'PC=', PC
  WRITE(8,*) 'WOFF=', WOFF, 'WIDTH=', WIDTH
ENDIF

RETURN
END

C***************************************************************
C SUBROUTINE STRIFI CALCULATES THE STRESS INTENSITY FACTOR
C IN A FINITE WIDTH PLATE WITH AN ELLIPTIC FLAW
C
C PROGRAMMER : S. SUTHARSHANA
C
C SIF EXPRESSIONS ADAPTED FROM NAS8609A.FOR VERSION OF NASA/FLAGRO CODE
C EXPRESSIONS SURFACE CRACK IN A FINITE WIDTH PLATE IS EMPLOYED
C THE WIDTH IS SET TO THE CIRCUMFERENCE OF THE TUBE
C
C DATE : DECEMBER 1992
C VERSION 92.5
C***************************************************************
AOT = A/T
AOTSQ = (A/T)**2
C = CRACK LENGTH, C
COA = C/A
COA4 = (C/A)**4
CONST = SQRT(PI*A)
PI = 3.14159...
POWR = POWR TO BE RAISED
SQCOA = SQRT(C/A)
TRM = 1 - A/C

IMPLICIT NONE

INTEGER IOUT

REAL A, AOC, AOC75, AOT, AOTSQ, C, COA, COA4, EM0, & EM1, EM2, EM3, F0(2), F2(2), FPA, FPC, FW, G1, G2, & GA, GC, H1, H2, INDIA, PI, Q, & SQCOA, THIC, WIDTH

REAL*8 POWR, TRM

COMMON IOUT

DATA PI/3.14159265358979/

WIDTH = PI * (INDIA + THIC)
AOC = A/C
AOT = A/THIC
IF (IOUT .EQ. 22) THEN
WRITE(8,*) 'AOC, AOT', AOC, AOT
ENDIF

Compute SIFs

IF (AOC.LE.1.0) THEN
  G1 = -1.22-0.12*AOC
  AOC75 = AOC75 * SQRT(AOC75)
  G2 = 0.55 + AOC75 * ((-1.05) + AOC75*0.47)
  H1 = 1.0 - AOT * (0.34 + 0.11*AOC)
  H2 = 1.0 + AOT*(G1 + G2*AOT)
  EM1 = 1.13 - 0.09*AOC
  EM2 = -0.54 + 0.89 / (0.2 + AOC)
  TRM = 1.0-AOC
  IF (ABS(TRM).GT.0.001D0) THEN
    POWR = TRM**24
  ELSE
    POWR = 0D0
  ENDIF
  EM3 = 0.5 - 1.0/(0.65+AOC) + 14.0*POWR
  AOTSQ = AOT*AOT
  EM0 = EM1 + AOTSQ*(EM2 + AOTSQ*EM3)
  Q = 1.0 + 1.464*AOC**1.65
  FPA = 1.0
  FPC = SQRT(AOC)
  GA = 1.0
  GC = 1.0 + (0.1 + 0.35*AOT*AOT)
ELSE
  AOC > 1
  COA = 1.0/AOC
  SQCOA = SQRT(COA)
  EM1 = (1.0+0.04*COA)*SQCOA
  COA4 = COA*COA*COA*COA
  EM2 = 0.2*COA4
  EM3 = -0.11*COA4
  AOTSQ = AOT*AOT
  EM0 = EM1 + AOTSQ*(EM2 + AOTSQ*EM3)
  GA = 1.0
  GC = 1.0 + (0.1 + 0.35*AOTSQ*COA)
  AOC75 = SQCOA*SQR(AOC75)
  H1 = 1.0 + AOT*(-0.4 - 0.41*COA + AOT*(0.55 +

7 - 56
&

H2 = 1.0 + AOT\{0.55 + AOC75\{-0.72 + AOC75\*0.14\}\}\)
Q = 1.0 + 1.464\*COA**1.65
FPA = SQRT(COA)
FPC = 1.0
ENDIF

TREAT IT AS AN EDGE CRACK IF 2C > WIDTH

IF ( 2.0\*C .LT. WIDTH ) THEN
FW = SQRT( 1.0/COS( SQRT(AOT)*PI*C/WIDTH ) )
ELSE
FW = 1.0
ENDIF
IF(IOUT .EQ. 22) THEN
WRITE(8,*), 'WIDTH, C, A, FW'
ENDIF

for "a" direction
F0(1) = EM0*GA*FPA*FW/SQRT(Q)
F2(1) = F0(1)*H2
IF(IOUT .EQ. 20) THEN
WRITE(8,*), 'A DIR F0, F2', F0(1), F2(1)
ENDIF
RETURN

END

*************************************************************************
SUBROUTINE STRIF2(A, F0, F2, WIDTH)
C IMPLICIT NONE
REAL A, F0(2), F2(2), PI, PIA, WIDTH
DATA PI/3.14159265358979/
PIA = PI * A
F0(1) = SQRT( 1.0/COS(PIA/WIDTH) )
F2(1) = F0(1)
RETURN
END

***************************************************************************
SUBROUTINE GRODAT READS MATERIAL PROPERTIES AND PERFORMS REGRESSION ON CRACK GROWTH DATA
PROGRAMMER : S. SUTHARSHAN
DATE: DECEMBER 1992
VERSION: 92.5
C Copyright (C) 1991, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.
SUBROUTINE GRODAT (CEE, CO, DEE, DKTHO, EMM, ENN, FTY, 
&                    KC, PEE, QUE)
C IMPLICIT NONE
EXTERNAL DETER4
INTEGER MAXDAT, MAXDIV
PARAMETER (MAXDAT = 200, MAXDIV = 10)
INTEGER IOUT, IREGOP, I, J, K, NDIV, NP(MAXDIV), 
&                    NFR, NFRMN1
REAL AA, BB, CC, CEE, DADN(MAXDIV, MAXDAT), 
&                     CO, DEE, DELK(MAXDIV, MAXDAT), DENOM, DETER4, 
&                     DIFFXI, DIFFX2, DIFFX3, DIFFX4, DIFFY, DKTH, DKTHO, 
&                     EMM, ENN, FTY, KC, LNCEE, MEANXI, MEANX2, MEANX3, MEANX4, 
&                     MEANY, PEE, QUE, RDATA(MAXDIV), RKC, 
&                     SXIY, SX2Y, SX3Y, SXIX1, SXIX2, SXIX3, SX2X2, SX2X3, 
&                     SX3X3, SX4X4, SXIX4, SX2X4, SX3X4, SX4X4, Y
C== ======== DESCRIPTION OF LOCAL VARIABLES ==_-=_==----
C DADN() CRACK GROWTH RATE
C DELK() SIF RANGE
C DENOM DENOMINATOR
C RDATA() STRESS RATIOS
C RKC (1-R)KC
C CHARACTER*40 DESCRP
COMMON IOUT
C READ THE da/dN vs. DK DATA FOR THE DIFFERENT REGIONS
READ(1,*) DESCRP, FTY, KC, NDIV, IREGOP
WRITE(3,6000) DESCRP, FTY, KC, NDIV, IREGOP
IF(IREGOP.LT.0 .OR. IREGOP.GT.4) THEN
   WRITE ( 8, *) 'INVALID REGRESSION OPTION SPECIFICATION'
   CALL TRMNAT
ENDIF
C READ THRESHOLD DESCRIPTION INFORMATION
READ(1,*) DKTHO, CO, DEE
WRITE(3, 6050) DKTHO, CO, DEE
IF(IREGOP .EQ. 0) THEN
   READ(1,*) PEE
ELSEIF(IREGOP .EQ. 1) THEN
   READ(1,*) EMM, PEE
ELSEIF(IREGOP .EQ. 2) THEN
   READ(1,*) QUE, PEE
ELSEIF(IREGOP .EQ. 3) THEN
   READ(1,*) EMM, QUE, PEE
ENDIF
DO 190 I = 1, NDIV
   READ(1,*) NP(I), RDATA(I)
   WRITE(3,6150) RDATA(I)
   IF(NP(I) .GT. MAXDAT) THEN
      WRITE(8,*) 'NUMBER OF GROWTH RATE DATA POINTS PER DIVISION 
&                     EXCEEDED'
   CALL TRMNAT
ENDIF
DO 180 J = 1, NP(I)
   READ(1,*) DADN(I,J), DELK(I,J)
   WRITE(3,6200) DADN(I,J), DELK(I,J)
180 CONTINUE
190 CONTINUE
C============================================
C PERFORM REGRESSION ON THE DATA
C==============================================

C CALCULATE SX2, SY2, SXY

IF(IREGOP .EQ. 0) THEN

NPR = 0
MEANY = 0.0
MEANX1 = 0.0
MEANX2 = 0.0
MEANX3 = 0.0
SX1Y = 0.0
SX2Y = 0.0
SX3Y = 0.0
SX1X1 = 0.0
SX1X2 = 0.0
SX1X3 = 0.0
SX2X2 = 0.0
SX2X3 = 0.0
SX3X3 = 0.0

DO 275 J = 1, NDIV
RKC = (1.-RDATA(J))*KC
DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
DO 250 K = 1, NP(J)
Y = LOG10(DADN(J,K))
  + PEE * LOG10(DELK(J,K) - DKTH)
MEANY = MEANY + Y
MEANX1 = MEANX1 + LOG10(DELK(J,K))
MEANX2 = MEANX2 + LOG10(1.-RDATA(J))
MEANX3 = MEANX3 - LOG10(RKC - DELK(J,K))

CONTINUE
NPR = NPR + NP(J)
275 CONTINUE

MEANY = MEANY/FLOAT(NPR)
MEANX1 = MEANX1/FLOAT(NPR)
MEANX2 = MEANX2/FLOAT(NPR)
MEANX3 = MEANX3/FLOAT(NPR)

C NOW CALCULATE SY2, SX2, AND SXY

DO 350 J= 1, NDIV
RKC = (1.-RDATA(J))*KC
DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
DO 300 K = 1, NP(J)
Y = LOG10(DADN(J,K))
  - PEE * LOG10(DELK(J,K) - DKTH)
DIFFY = Y - MEANY
DIFFX1 = LOG10(DELK(J,K)) - MEANX1
DIFFX2 = LOG10(1.-RDATA(J)) - MEANX2
DIFFX3 = MEANX3 - LOG10(RKC - DELK(J,K))
SX1Y = SX1Y + DIFFX1 * DIFFY
SX2Y = SX2Y + DIFFX2 * DIFFY
SX3Y = SX3Y + DIFFX3 * DIFFY
SX1X1 = SX1X1 + DIFFX1 * DIFFY
SX1X2 = SX1X2 + DIFFX1 * DIFFX2
SX1X3 = SX1X3 + DIFFX1 * DIFFX3
SX2X2 = SX2X2 + DIFFX2 * DIFFY
SX2X3 = SX2X3 + DIFFX2 * DIFFX3
SX3X3 = SX3X3 + DIFFX3 * DIFFY

CONTINUE
350 CONTINUE

NPRN1 = NPR - 1
SX1Y = SX1Y/FLOAT(NPRN1)
SX2Y = SX2Y/FLOAT(NPRN1)
SX3Y = SX3Y/FLOAT(NPRN1)
SX1X1 = SX1X1/FLOAT(NPRN1)
SX1X2 = SX1X2/FLOAT(NPRN1)
SX1X3 = SX1X3/FLOAT(NPRN1)
SX2X2 = SX2X2/FLOAT(NPRN1)
SX2X3 = SX2X3/FLOAT(NPRN1)
SX3X3 = SX3X3/FLOAT(NPRN1)
C CALCULATE THE COEFFICIENTS

\[ AA = Sx2x2 \times Sx3x3 - Sx2x3 \times Sx1x3 \]
\[ BB = Sx1x2 \times Sx3x3 - Sx2x3 \times Sx1x3 \]
\[ CC = Sx1x2 \times Sx2x3 - Sx2x2 \times Sx1x3 \]
\[ DENOM = Sx1x1 \times AA - Sx1x2 \times BB + Sx1x3 \times CC \]
\[ ENN = \left( Sx1y \times AA - Sx1x2 \times (Sx2y \times Sx3x3 - Sx2x3 \times Sx3y) \right) / DENOM \]
\[ EMM = \left( Sx1x1 \times (Sx2y \times Sx3x3 - Sx2x3 \times Sx3y) - Sx1y \times BB \right) / DENOM \]
\[ QUE = \left( Sx1x1 \times (Sx2x2 \times Sx3y - Sx2y \times Sx2x3) + Sx1y \times CC \right) / DENOM \]
\[ LNCEE = MEANY - EMM \times MEANX1 - EMM \times MEANX2 - QUE \times MEANX3 \]
\[ CEE = 10.0^{(LNCEE)} \]

ELSEIF (IREG0P.EQ. 1) THEN

NPR = 0
MEANY = 0.0
MEANX1 = 0.0
MEANX3 = 0.0
Sx1y = 0.0
Sx3y = 0.0
Sx1x1 = 0.0
Sx1x3 = 0.0
Sx3x3 = 0.0

DO 1275 J = 1, NDIV
   RKC = (1. - RDATA(J)) \times KC
   DKTH = DKTH0 \times (1. - CO \times RDATA(J)) \times DEE
   DO 1250 K = 1, NP(J)
      Y = LOG10(DADN(J,K))
      MEANY = MEANY + Y
      MEANX1 = MEANX1 + LOG10(DELK(J,K) - DKTH)
      MEANX3 = MEANX3 - LOG10(RKC - DELK(J,K))
   CONTINUE
   NPR = NPR + NP(J)
   CONTINUE
1275

CONTINUE
MEANY = MEANY/FLOAT(NPR)
MEANX1 = MEANX1/FLOAT(NPR)
MEANX3 = MEANX3/FLOAT(NPR)

C NOW CALCULATE SY2, SX2, AND SXY

DO 1350 J = 1, NDIV
   RKC = (1. - RDATA(J)) \times KC
   DKTH = DKTH0 \times (1. - CO \times RDATA(J)) \times DEE
   DO 1300 K = 1, NP(J)
      Y = LOG10(DADN(J,K))
      DIFFY = Y - MEANY
      DIFFX1 = LOG10(DELK(J,K)) - MEANX1
      DIFFX3 = LOG10(RKC - DELK(J,K)) - MEANX3
      Sx1y = Sx1y + DIFFX1 \times DIFFY
      Sx3y = Sx3y + DIFFX3 \times DIFFY
      Sx1x1 = Sx1x1 + DIFFX1 \times DIFFX1
      Sx1x3 = Sx1x3 + DIFFX1 \times DIFFX3
      Sx3x3 = Sx3x3 + DIFFX3 \times DIFFX3
   CONTINUE
1300

CONTINUE
1350

NPRM1 = NPR - 1
Sx1y = Sx1y/FLOAT(NPRM1)
Sx3y = Sx3y/FLOAT(NPRM1)
Sx1x1 = Sx1x1/FLOAT(NPRM1)
SX1X3 = SX1X3/FLOAT(NPRMNI)
SX3X3 = SX3X3/FLOAT(NPRMNI)

C CALCULATE THE COEFFICIENTS

DENOM = SX1X1 * SX3X3 - SX1X3 ** 2
ENN = ( SX1Y * SX3X3 - SX1X3 * SX3Y ) / DENOM
QUE = ( SX1X1 * SX3Y - SX1Y * SX1X3 ) / DENOM
LNCEE = MEANY - ENN*MEANXI - QUE*MEANX3
CEE = 10.0**(LNCEE)

1600 CONTINUE

ELSEIF( IREGOP .EQ. 2 ) THEN

NPR = 0
MEANY = 0.0
MEANXI = 0.0
MEANX2 = 0.0
SX1Y = 0.0
SX2Y = 0.0
SX1X1 = 0.0
SX1X2 = 0.0
SX2X2 = 0.0

DO 2275 J = 1, NDIV
   RKC = (1.-RDATA(J))*KC
   DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
   DO 2250 K = 1, NP(J)
      Y = LOG10(DADN(J,K))
      & - PEE * LOG10(DELK(J,K) - DKTH)
      & + QUE * LOG10(RKC - DELK(J,K))
      MEANY = MEANY + Y
      MEANXI = MEANXI + LOG10(DELK(J,K))
      MEANX2 = MEANX2 + LOG10(1.-RDATA(J))
   2250 CONTINUE

NPR = NPR + NP(J)
2275 CONTINUE

CONTINUE

MEANY = MEANY/FLOAT(NPR)
MEANXI = MEANXI/FLOAT(NPR)
MEANX2 = MEANX2/FLOAT(NPR)

C NOW CALCULATE SY2, SX2, AND SXY

DO 2350 J = 1, NDIV
   RKC = (1.-RDATA(J))*KC
   DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
   DO 2300 K = 1, NP(J)
      Y = LOG10(DADN(J,K))
      & - PEE * LOG10(DELK(J,K) - DKTH)
      & + QUE * LOG10(RKC - DELK(J,K))
      DIFFY = Y - MEANY
      DIFFX1 = LOG10(DELK(J,K)) - MEANXI
      DIFFX2 = LOG10(1.-RDATA(J)) - MEANX2
      SX1Y = SX1Y + DIFFX1 * DIFFY
      SX2Y = SX2Y + DIFFX2 * DIFFY
      SX1X1 = SX1X1 + DIFFX1 * DIFFX1
      SX1X2 = SX1X2 + DIFFX1 * DIFFX2
      SX2X2 = SX2X2 + DIFFX2 * DIFFX2
   2300 CONTINUE

2350 CONTINUE

NPRMNI = NPR - 1

SX1Y = SX1Y/FLOAT(NPRMNI)
SX2Y = SX2Y/FLOAT(NPRMNI)
SX1X1 = SX1X1/FLOAT(NPRMNI)
SX1X2 = SX1X2/FLOAT(NPRMNI)
SX2X2 = SX2X2/FLOAT(NPRMNI)

C CALCULATE THE COEFFICIENTS
DENOM = SX1X1 * SX2X2 - SX1X2 ** 2
ENN = ( SX1Y * SX2X2 - SX1X2 * SX2Y) / DENOM
EMM = ( SX1X1 * SX2Y - SX1Y * SX1X2 ) / DENOM
LNCEE = MEANY - ENN*MEANXI - EMM*MEANX2
CEE = 10.0**(LNCEE)
ELSEIF (IREGOP .EQ. 3) THEN
    NPR = 0
    MEANY = 0.0
    MEANXI = 0.0
    SX1Y = 0.0
    SX1X1 = 0.0
    DO 3275 J = 1, NDIV
        RKC = (1. - RDATA(J)) * KC
        DKTH = DKTHO * (1. - CO * RDATA(J)) ** DEE
        DO 3250 K = 1, NP(J)
            Y = LOG10(DADN(J,K))
            & - EMM * LOG10(1. - RDATA(J))
            & - PEE * LOG10(DELK(J,K) - DKTH)
            & + QUE * LOG10(RKC - DELK(J,K))
            MEANY = MEANY + Y
            MEANXI = MEANXI + LOG10(DELK(J,K))
        3250 CONTINUE
        NPR = NPR + NP(J)
    3275 CONTINUE
    MEANY = MEANY/FLOAT(NPR)
    MEANXI = MEANXI/FLOAT(NPR)
ENDIF
C NOW CALCULATE SY2, SX2, AND SXY
DO 3350 J = 1, NDIV
    RKC = (1. - RDATA(J)) * KC
    DKTH = DKTHO * (1. - CO * RDATA(J)) ** DEE
    DO 3300 K = 1, NP(J)
        Y = LOG10(DADN(J,K))
        & - EMM * LOG10(1. - RDATA(J))
        & - PEE * LOG10(DELK(J,K) - DKTH)
        & + QUE * LOG10(RKC - DELK(J,K))
        DIFFY = Y - MEANY
        DIFFX1 = LOG10(DELK(J,K)) - MEANXI
        SX1Y = SX1Y + DIFFX1 * DIFFY
        SX1X1 = SX1X1 + DIFFX1 * DIFFX1
    3300 CONTINUE
    NPRMN1 = NPR - 1
    SX1Y = SX1Y/FLOAT(NPRMN1)
    SX1X1 = SX1X1/FLOAT(NPRMN1)
ENDIF
C CALCULATE THE COEFFICIENTS
ENN = SX1Y / SX1X1
LNCEE = MEANY - ENN * MEANXI
CEE = 10.0**(LNCEE)
ELSEIF (IREGOP .EQ. 4) THEN
    NPR = 0
    MEANY = 0.0
    MEANXI = 0.0
    MEANX2 = 0.0
    MEANX3 = 0.0
    MEANX4 = 0.0
    SX1Y = 0.0
    SX2Y = 0.0
    SX3Y = 0.0
    SX4Y = 0.0
    SX1X1 = 0.0
ENDIF
7 - 62
SX1X2 = 0.0
SX1X3 = 0.0
SX1X4 = 0.0
SX2X2 = 0.0
SX2X3 = 0.0
SX2X4 = 0.0
SX3X3 = 0.0
SX3X4 = 0.0
SX4X4 = 0.0

DO 4275 J = 1, NDIV
RKC = (1.-RDATA(J))*KC
DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
DO 4250 K = 1, NP(J)
   MEANY = MEANY + LOG10(DADN(J,K))
   MEAX1 = MEAX1 + LOG10(DELK(J,K))
   MEAX2 = MEAX2 + LOG10(1.-RDATA(J))
   MEAX3 = MEAX3 + LOG10(DELK(J,K) - DKTH)
   MEAX4 = MEAX4 - LOG10(RKC - DELK(J,K))
4250 CONTINUE
NPR = NPR + NP(J)
4275 CONTINUE
MEANY = MEANY/FLOAT(NPR)
MEAX1 = MEAX1/FLOAT(NPR)
MEAX2 = MEAX2/FLOAT(NPR)
MEAX3 = MEAX3/FLOAT(NPR)
MEAX4 = MEAX4/FLOAT(NPR)

C NOW CALCULATE SY2, SX2, AND SXY

DO 4350 J = 1, NDIV
RKC = (1.-RDATA(J))*KC
DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
DO 4300 K = 1, NP(J)
   DIFFY = LOG10(DADN(J,K)) - MEANY
   DIFFX1 = LOG10(DELK(J,K)) - MEAX1
   DIFFX2 = LOG10(1.-RDATA(J)) - MEAX2
   DIFFX3 = LOG10(DELK(J,K) - DKTH) - MEAX3
   DIFFX4 = - LOG10(RKC - DELK(J,K)) - MEAX4
   SX1Y = SX1Y + DIFFX1 * DIFFY
   SX2Y = SX2Y + DIFFX2 * DIFFY
   SX3Y = SX3Y + DIFFX3 * DIFFY
   SX4Y = SX4Y + DIFFX4 * DIFFY
   SX1X = SX1X + DIFFX1 * DIFFX2
   SX1X2 = SX1X2 + DIFFX1 * DIFFX2
   SX1X3 = SX1X3 + DIFFX1 * DIFFX3
   SX1X4 = SX1X4 + DIFFX1 * DIFFX4
   SX2X = SX2X + DIFFX2 * DIFFX3
   SX2X2 = SX2X2 + DIFFX2 * DIFFX3
   SX2X3 = SX2X3 + DIFFX2 * DIFFX4
   SX2X4 = SX2X4 + DIFFX2 * DIFFX4
   SX3X = SX3X + DIFFX3 * DIFFX3
   SX3X2 = SX3X2 + DIFFX3 * DIFFX4
   SX3X3 = SX3X3 + DIFFX3 * DIFFX4
   SX3X4 = SX3X4 + DIFFX3 * DIFFX4
   SX4X = SX4X + DIFFX4 * DIFFX4
   SX4X2 = SX4X2 + DIFFX4 * DIFFX4
   SX4X3 = SX4X3 + DIFFX4 * DIFFX4
   SX4X4 = SX4X4 + DIFFX4 * DIFFX4
4300 CONTINUE
4350 CONTINUE
NPRMN1 = NPR - 1
SX1Y = SX1Y/FLOAT(NPRMN1)
SX2Y = SX2Y/FLOAT(NPRMN1)
SX3Y = SX3Y/FLOAT(NPRMN1)
SX4Y = SX4Y/FLOAT(NPRMN1)
SX1X2 = SX1X2/FLOAT(NPRMN1)
SX1X3 = SX1X3/FLOAT(NPRMN1)
SX1X4 = SX1X4/FLOAT(NPRMN1)
SX2X2 = SX2X2/FLOAT(NPRMN1)
SX2X3 = SX2X3/FLOAT(NPRMN1)
SX2X4 = SX2X4/FLOAT(NPRMN1)
SX3X2 = SX3X2/FLOAT(NPRMN1)
SX3X3 = SX3X3/FLOAT(NPRMN1)
SX3X4 = SX3X4/FLOAT(NPRMN1)
SX4X2 = SX4X2/FLOAT(NPRMN1)
SX4X3 = SX4X3/FLOAT(NPRMN1)
SX4X4 = SX4X4/FLOAT(NPRMN1)

C CALCULATE THE COEFFICIENTS

7 - 63
DENOM = DETER4(SXIX1, SXIX2, SXIX3, SXIX4, SXIX2, SXIX3, SXIX4, 
& SXIX4, SX2X4, SX3X4, SX4X4)
&
ENN = DETER4(SXIX1, SX2Y, SX3Y, SX4Y, SXIX2, SX2X2, 
& SX2X3, SX2X4, SX3X3, SX3X4, SX3X4, SX4X4) / DENOM
&
EMM = DETER4(SXIX1, SXIX2, SXIX3, SXIX4, SXIX2, SXIX3, SXIX4, 
& SXIX4, SX2X2, SX2X3, SX2X4, SX3X3, SX3X4, SX4X4) / DENOM
&
Pee = DETER4(SXIX1, SXIX2, SXIX3, SXIX4, SXIX2, SXIX3, SXIX4, 
& SXIX4, SX2X2, SX2X3, SX2X4, SX3X3, SX3X4, SX4X4, SX4X4) / DENOM
&
QUE = DETER4(SXIX1, SXIX2, SXIX3, SXIX4, SXIX2, SXIX3, SXIX4, 
& SXIX4, SX2X2, SX2X3, SX2X4, SX3X3, SX3X4, SX4X4, SX4X4) / DENOM
&
LNCEE = MEANY - ENN*MEANXI - EMM*MEANX2 - PEE*MEANX3 - QUE*MEANX4
&
CEE = 10.0**(LNCEE)
&
ENDIF

C WRITE OUT THE REGRESSED VALUES
WRITE(3,6300) CEE, ENN, EMM, PEE, QUE
RETURN

C------------------------------ FORMATS -----------------------------
6000 FORMAT(///,13X,'MATERIAL INPUT',///,2X,'DESCRIPTION:',2X,A40,///, 
& 18X,'YIELD STRENGTH',18X,F7.0,///, 
& 2X,'CRITICAL S I F', 
& 18X,F7.0,///,2X,'NUMBER OF DIVISIONS',14X,I1, 
& //,2X,'REGRESSION OPTION',16X,I1,///)
6050 FORMAT(///,2X,'THRESHOLD MODEL DESCRIPTION', 
& ///,2X,'DKTHO = ',E12.5, 
& //,2X,'CO = ',E12.5, 
& //,2X,'d = ',E12.5)
6150 FORMAT(///,2X,'STRESS RATIO R = ',F7.2,///,6X,'da/dN',8X,'DELK')
6200 FORMAT(2X,E12.5,2X,E12.5)
6300 FORMAT(///,25X,'REGRESSION OUTCOME',///,3X, 
6400 FORMAT(10X,I1,'-',11,5X,E12.5,5X,E12.5)
6600 FORMAT(IOX,5X,'-',5X,E12.5)
END

C***************************************************************
C FUNCTION DETER4 CALCULATES DETERMINANT OF A 4x4 MATRIX
C***************************************************************
C PROGRAMMER: S. SUTHARSHANA
C DATE: 25 SEP 1989
C VERSION: 92.1, 92.2, 92.3, 92.4, 92.5
C***************************************************************

REAL FUNCTION DETER4 (A11, A21, A31, A41, A12, A22, A32, A42, 
& A13, A23, A33, A43, A14, A24, A34, A44)
C IMPLICIT NONE
DETER4 = A11*( A22 * (A33*A44 - A34*A43)
SUBROUTINE INPUT (ANGLE, BLFPER, COEXP, E, EM, LOCAT, NBLIFE, NHYPER, NLI"FE, NLI"FET, NRAN, NU, NUMSEG, PERIOD, RSO, SE, STRHIS, TRUNC)

IMPLICIT NONE

INTEGER MAXBLF, MAXLD, MAXLIF, MAXM, MAXSEG
REAL PI
PARAMETER (M_L "20006, MAXSEG = 10, PI = 3.141592654)
INTEGER INEUB, IRET, KGROW, KPROB
INTEGER FILNUM(MAXLD), I, IOUT, J, LOCAT, NBLIFE, NHYPER,
NLIFE, NLIFET, NLOAD, NRAN, NU, NUMSEG, TYPE(MAXLD)
DOUBLE PRECISION RAND
REAL AERDA, AERDB, AERSA, AERSB, AIA, AIB,
AIR1, AIR2, AIT1, AIT2, AIR,
AIT, ANGLE, BLFPER(MAXBLF),
COEXP, AOCA, AOCH, AOCHR, AOCH2, AOCHT, AOCHT2,
AOCH, AOCHT, ASTR, DPCMU, DPCSIG,
ASTR, DSTRB, TISIG, DTISIG,
DTOMU, DTISIG, E(MAXSEG), EM, FK(10),
INDIA, INDIB, INDIR, INDIRI,
INDIR2, INDIRT, INDIRT1, INDIRT2,
KLAMA, KLAMB, LAMKA, LAMKB, LAMKCA, LAMKCB,
LAMGR, LAMGRB,
LAMNA, LAMNB, LAMNC, LAMND, LAMNU, LAMNSG,
LAMSA, LAMSB, LAMSC, LAMSD, LAMSMU, LAMSSG,
LAMWA, LAMWB
REAL M(2, MAXLD), MSTAT(2), MVARA,
MVARB, NEUB, NEUUB, NU,
P(MAXLD), PCMU, PCO, PCSIG, PCSIGA,
PCSIGB, PERIOD,
FSTAT, RSO, RT(10), SE(MAXSEG), SSTRA,
SXSTB, STRHIS(MAXLIF, MAXM), SX(MAXLD), SXST,
SX(10), SXST, SXS(MAXLD), SXST, SY(SXST), SYST,
T(MAXLD) THICA, THICB, THICR, THICRI,
THICR2, THICT, THICT1, THICT2, TIMU, TIMUA,
TIMUB, TISIG, TISIGA, TISIGB, TOMU,
TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
TRUNC, TSTAT, V(2, MAXLD),
VSTAT(2), WITHA, WITHB,
WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
WOFFA, WOFFB, WOFFC, WOFFD,
WOFFE, WOFFHI, WOFFLO, WOFFR, WOFFR1, WOFFR2, WOFFR3,
WOFFR4, WOFFT, WOFFTI, WOFFT2, WOFFT3, WOFFT4

CHARACTER*6 LDNAME(MAXLD)

COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
   P, T, M, V, PCO, SXST, SYST, SZST, SXST,
   SXST, SYST, SX, SY, S2, SY, S2, SY,

COMMON/DRIVRS/ AENA, AERD, AERSA, AERSB, AIA,
   AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT,
   AOCR, AOCR1, AOCR2, AOCR1, AOCT, AOCT2,
   ASTR1, ASTR2,
   DPCMU, DPCSIG, DSTRA, DSTRB, DTMU, DTSIG, DTOMU, DTOSIG,
   INDAA, INDIAB, INDIR1, INDIR2, INDIR, INDIR1, INDIR2, INDIR,
   KLAMA, KLAMB, LAMRA, LAMRBA, LAMKB, LAMKBA, LAMKCB,
   LAMNA, LAMNB, LAMNC, LAMBND, LAMMU, LAMNSG,
   LAMSA, LAMSB, LAMSC, LAMSD, LAMSS, LAMSG,
   LAMWA, LAMWB, MVARA, MVARB, NEUBA, NEUBB,
   PCMU, PCMUA, PCMUB, PCSIG, PCSIQA, PCSIQB,
   RAND,
   SSTR1, SSTR2,
   THICA, THICB, THICR1, THICR2, THICT1, THICT2, THICT,
   TIMU, TIMUA, TIMUB, TISIG, TISIGA, TISIGB,
   TOME, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
   WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2, WITHR, WITHT,
   WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
   WOFFR, WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR, WOFFT1, WOFFT2,
   WOFFT3, WOFFT4, WOFFT

COMMON/FKVSRT/FK, RT
COMMON/NAMES/LDNAME

COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
COMMON IOUT

LOGICAL FTEST

DATA (FILNUM(I), I = 1, MAXLD) /
   11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
   24, 25, 26/

READ(1,*) KPROB
WRITE(8, *)'PROBLEM TYPE (HEX COIL = 1, EXHEX = 2) =', KPROB
IF(KPROB .LT. 1 .OR. KPROB .GT. 2) THEN
   WRITE(8, *) 'INVALID PROBLEM TYPE SPECIFICATION'
   CALL TRMNAT
ENDIF

READ(1,*) KGROW
WRITE(8, *)'FORMAN EQUATION WITH R (CONST = 1, VARY = 2) =’, KGROW
IF(KGROW .LT. 1 .OR. KGROW .GT. 2) THEN
   WRITE(8, *) 'INVALID FORMAN EQUATION SPECIFICATION'
   CALL TRMNAT
ENDIF

READ(1,*) RAND
WRITE(8, *)'RANDOM NUMBER SEED =’, RAND

READ(1,*) IOUT
WRITE(8, *)'OUTPUT CONTROL VARIABLE =', IOUT

READ(1,*) NLIfe
WRITE(8, *)'INNER LOOP SIZE =’, NLIfe

READ(1,*) NHYPER
WRITE(8, *)'OUTER LOOP SIZE =', NHYPER

READ(1,*) IRET
WRITE(8, *)'RETARDATION SWITCH (0 - NO, 1 - YES) =’, IRET
IF(IRET .LT. 0 .OR. IRET .GT. 1) THEN
   WRITE(8, *) 'INVALID RETARDATION SWITCH SPECIFICATION'
   CALL TRMNAT
ENDIF

READ(1,*) INEUB
WRITE(8, *)'NEUBER SWITCH (0 - NO, 1 - YES) =’, INEUB
IF(INEUB .LT. 0 .OR. INEUB .GT. 1) THEN
   WRITE(8, *) 'INVALID NEUBERS RULE SPECIFICATION'

7 - 66
CALL TRMNAT
ENDIF

C CALCULATE TOTAL NUMBER OF LIVES ... IF NLIFET = 1 DETERMINISTIC RUN

NLIFET = NLIFE * NHYPER

READ(1,*) NBLIFE
IF(NBLIFE .GT. 0) THEN
READ(1,*) (BLFPER(J), J = 1, NBLIFE)
ENDIF

C READ DRIVER INFORMATION

IF (KPROB .EQ. 1) THEN
READ(1,*) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2, $ WOFFC, WOFFD, WOFFR3, WOFFR4, WOFFT3, WOFFT4, $ WOFFE, $ INDIAA, INDIB, INDIR1, INDIR2, INDIT1, INDIT2, $ THICA, THICB, THICR1, THICR2, THICT1, THICT2, $ AOCA, AOCB, AOCR1, AOCR2, AOCR1, AOCR2, AOCR2
ELSEIF (KPROB .EQ. 2) THEN
READ(1,*) WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2
ENDIF

READ(1,*) AIA, AIB, AIR1, AIR2, AIT1, AIT2, $ LAMNA, LAMNB, LAMNC, LAMND, $ LAMSA, LAMSAB, LAMSC, LAMSD

IF (KPROB .EQ. 1) THEN
READ(1,*) TIMUA, TIMUB, TISIGA, TISIGB, $ TOMUA, TOMUB, TOSIGA, TOSIGB, $ PCMUA, PCMUB, PCSIGA, PCSIGB
ENDIF

C CALCULATE SOME DRIVER VARIABLES

DTIMU = TIMUB - TIMUA
DTISIG = TISIGB - TISIGA
DTOMU = TOMUB - TOMUA
DTOSIG = TOSIGB - TOSIGA
DPCMU = PCMUB - PCMUA
DPCSIG = PCSIGB - PCSIGA

IF (IOUT .EQ. 15) THEN
WRITE(8,*) 'DTIMU = ', DTIMU, ' DTISIG = ', DTISIG
WRITE(8,*) 'DTOMU = ', DTOMU, ' DTOSIG = ', DTOSIG
WRITE(8,*) 'DPCMU = ', DPCMU, ' DPCSIG = ', DPCSIG
ENDIF

C READ ACCURACY FACTORS

READ(1,*) LAMNA, LAMNB, AERDA, AERDB, AERSA, AERSB, $ ASTRA, ASTRB, DSTRA, DSTRB
IF(INEUB .EQ. 1) THEN
READ(1,*) NEUBA, NEUBB
ENDIF
ELSE
READ(1,*) SSTR, SSTRB, DSTRA, DSTRB
ENDIF

READ(1,*) LAMKH, LAMKHKB, LAMKHC, LAMKCB, $ KLAMA, KLAMB, LAMGRA, LAMGB

IF(KGROW .EQ. 2) THEN
READ(1,*) MVARA, MVARB
ENDIF

C READ THE LOADS OR STRESSES

IF(KPROB .EQ. 1) THEN
READ(1,*) NLOAD, PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1), $ VSTAT(2)
DO 15 I = 1, NLOAD
READ(1,*) LDNAME(I), TYPE(I), P(I), T(I), M(I,I), M(2,I),
15 CONTINUE
ELSE
READ(1,*) NLOAD, PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1), $ VSTAT(2)
DO 15 I = 1, NLOAD
READ(1,*) LDNAME(I), TYPE(I), P(I), T(I), M(I,I), M(2,I),
15 CONTINUE
ENDIF
& \text{V(I,I), V(2,I)}

\text{IF ((TYPE(I) .LT. 1) .OR. (TYPE(I) .GT. 3)) THEN}
\text{WRITE(*,*) 'ERROR: LOAD INCORRECTLY TYPED'}
\text{CALL TRMNAT}
\text{ENDIF}

15 \text{CONTINUE}

\text{ELSEIF(KPROB .EQ. 2) THEN}
\text{READ(1,*) NLOAD, SXST, SYST, SZST, SXYST, SYYST, SZYST}
\text{DO 16 I = 1, NLOAD}
\text{READ(1,*) LDNAME(I), TYPE(I), SX(I), SY(I), SZ(I), SXY(I),}
\text{& sxz(I), sYz(i)}
\text{IF ((TYPE(I) .LT. i) .OR. (TYPE(I) .GT. 2)) THEN}
\text{WRITE(8,*) 'ERROR: LOAD INCORRECTLY TYPED'}
\text{CALL TRMNAT}
16 \text{CONTINUE}
\text{ENDIF}
\text{ENDIF}

C READ MISCELLANEOUS INFO

\text{IF(KPROB .EQ. 1) THEN}
\text{READ(1,*) PCO, LOCAT, ANGLE}
\text{ENDIF}

C ECHO DATA TO CRKRES

\text{WRITE(3,900)}
\text{IF(KPROB .EQ. 1) THEN}
\text{WRITE(3,901) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,}
\text{& WOFFC, WOFFD, WOFFR3, WOFFR4, WOFFT3, WOFFT4,}
\text{WRITE(3,904) INDIAA, INDIIA, INDIII, INDIV, INDIT, INDIT2}
\text{WRITE(3,905) THICA, THICB, THICRI, THICR2, THICTI, THICT2}
\text{WRITE(3,911) AOCA, AOCB, AOCR1, AOCR2, AOCR3, AOCR4}
\text{ELSEIF(KPROB .EQ. 2) THEN}
\text{WRITE(3,902) WITHA, WITHB, WITHR1, WITHR2, WITHIT, WITHIT2}
\text{ENDIF}
\text{WRITE(3,910) AIA, AIB, AIRI, AIR2, AIT1, AIT2}
\text{WRITE(3,906) LAMNA, LAMNB, LAMNC, LAMND}
\text{WRITE(3,907) LAMSA, LAMSB, LAMSC, LAMSD}
\text{IF(KPROB .EQ. 1) THEN}
\text{WRITE(3,908) TIMUA, TIMUB, TISIGA, TISIGB,}
\text{& PCUMA, PCUMB, PCUSGA, PCUSGB}
\text{WRITE(3,9081) LAMWA, LAMWB, AERDA, AERDB, AERSA, AERSB,}
\text{ASTRA, ASTRB, DASTA, DASTB}
\text{IF(INEUB .EQ. 1) THEN}
\text{WRITE(3,9083) NEUBA, NEUBB}
\text{ENDIF}
\text{ELSE}
\text{WRITE(3,9082) SSTRA, SSTRB, DASTA, DASTB}
\text{ENDIF}

\text{WRITE(3,909) LAMKHA, LAMKHB, LAMKCA, LAMKCB,}
\text{& KLAMA, KLAMB, LAMGRA, LAMGRB}
\text{IF(KGROW .EQ. 2) THEN}
\text{WRITE(3,9091) MVARA, MVARB}
\text{ENDIF}
\text{IF(KPROB .EQ. 1) THEN}
\text{WRITE(3,920) PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1), VSTAT(2)}
\text{DO 20 I = 1, NLOAD}
\text{WRITE(3,921) LDNAME(I), P(I), T(I), M(I,1), M(2,I), V(I,1),}
\text{V(2,I)}
20 \text{CONTINUE}
\text{ELSEIF(KPROB .EQ. 2) THEN}
\text{WRITE(3,922) SXST, SYST, SZST, SXYST, SYYST, SZYST}
\text{DO 21 I = 1, NLOAD}
\text{WRITE(3,921) LDNAME(I), SX(I), SY(I), SZ(I), SXY(I),}
\text{& SXZ(I), SYZ(I)}
CONTINUE
ENDIF
WRITE(3,924)
IF(KPROB .EQ. 1) THEN
  WRITE(3,925) PCO, LOCAT, ANGLE
ENDIF
WRITE(3,926) RSO, PERIOD, TRUNC, NLOAD, NRAN

C CONVERT ANGLE TO RADIANS FOR CALCULATIONS
ANGLE = ANGLE/180.00000 * PI

C READ TIME HISTORIES FROM SPECIFIED FILES
IF (NRAN .GT. MAX/M) THEN
  WRITE(8,*) 'ERROR: STRESS-TIME HISTORY TOO LARGE'
  CALL TRMNAT
ENDIF
DO 25 I = 1, NLOAD
  INQUIRE (FILE = LDNAME(I), EXIST = FTEST)
  IF (FTEST .EQV. TRUE.) THEN
    OPEN (FILNUM(I), FILE = LDNAME(I), STATUS = 'OLD')
    DO 26 J = 1, NRAN
      READ (FILNUM(I), *) STRHIS(I,J)
    CONTINUE
    CLOSE (FILNUM(I))
  ELSE
    WRITE (8,*) 'ERROR: CANNOT OPEN FILE ', LDNAME(I),
    CALL TRMNAT
  ENDIF
25 CONTINUE
IF(KPROB .EQ. 1) THEN
  READ(1,*), EM, COEXP, NU
  WRITE(3,927) EM, COEXP, NU
ENDIF

C READ THE FK VS. RT CURVE FOR WELD STRESS CONCENTRATION FOR HEX COIL PROBLEM
WRITE(3,928)
DO 30 I = 1, 10
  READ(1,*), FK(I), RT(I)
WRITE(3,929) FK(I), RT(I)
30 CONTINUE

C READ IN THE STRESS-STRAIN VALUES IF NEUBER'S RULE IS TO BE USED IN HEX
IF (KPROB .EQ. 1 .AND. INEUB .EQ. 1) THEN
  READ(I.*) NUMSEG
  WRITE(3,930) NUMSEG
  DO 35 J = 1, NUMSEG
    READ(l,*), SE(J), E(J)
    WRITE(3,931) SE(J), E(J)
  CONTINUE
35 CONTINUE
ENDIF

C=
FORMAT STATEMENTS TO ECHO INPUT DATA TO CRKRES =========

900 FORMAT(2X,'Copyright (C) 1991, California Institute of ',
& 'Technology, U.S. Government',/,'Sponsorship under ',
& 'NASA Contract NAS7-918 is acknowledged.',/,'/
& 30X,'PARAMETER DISTRIBUTIONS',/,'4X,'/
& 16X,'THETA')
901 FORMAT(/,2X,'WELD OFFSET (%)',/,'3X,'Be(',F4.2,','F5.2,')',/,'6X,'/
& 4X,'U(',F4.1,','F5.1,')',/,'20X,'Be(',F4.2,','F5.2,')',/,'6X,'/
& 4X,'U(',F4.1,','F5.1,')',/,'26X,'TEST = ',F4.2)
902 FORMAT(/,2X,'CHANNEL WIDTH',4X,'Be('',F6.4,')',',F7.4,'')/,2X, & & 'U('',F7.5,')',',F8.5,'')/,4X,'U('',F4.1,')',',F5.1,'')}  
904 FORMAT(/,2X,'INNER DIAMETER',4X,'Be('',F6.4,')',',F7.4,'')/,2X, & & 'U('',F7.5,')',',F8.5,'')/,4X,'U('',F4.1,')',',F5.1,'')}  
905 FORMAT(/,2X,'WALL THICKNESS',4X,'Be('',F6.4,')',',F7.4,'')/,2X, & & 'U('',F7.5,')',',F8.5,'')/,4X,'U('',F4.1,')',',F5.1,'')}  
906 FORMAT(/,2X,'LAMBDA RANDOM',5X,'k: U('',F7.5,')',',F8.5,'')/, & & 20X,'COEFFICIENT OF VARIATION: ',F5.3, & & 20X,'STRAIN GAGE FACTOR: ',F9.7)  
907 FORMAT(/,2X,'LAMBDA SINE',7X,'k: U('',F7.5,')',',F8.5,'')/, & & 20X,'COEFFICIENT OF VARIATION: ',F5.3, & & 20X,'STRAIN GAGE FACTOR: ',F9.7)  
908 FORMAT(/,2X,'INNER TEMPERATURE',4X,'NORMAL: MU('',F7.,')',',F6.1,''), & & 20X,'OUTER TEMPERATURE',4X,'NORMAL: MU('',F7.,')',',F6.1,''), & & 20X,'INNER PRESSURE',7X,'NORMAL: MU('',F7.1,')',',F6.1,'')}  
909 FORMAT(/,2X,'WELD OFFSET K FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'DYN AERO LOAD FAC 3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'STAT AERO LOAD FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'AERO STR ANAL FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'DYSTR ANAL FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'DYN STR ANAL FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'K CALC FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'GROWTH CALC FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'GROWTH COEFF m',3X,'U('',F8.5,')',',F9.5,'')/)  
910 FORMAT(/,2X,'CRACK SIZE A',5X,'Be('',F6.4,')',',F7.4,'')/,2X, & & 'U('',F7.5,')',',F8.5,'')/,4X,'U('',F4.1,')',',F5.1,'')}  
911 FORMAT(/,2X,'CRACK SHAPE A/C',3X,'Be('',F6.4,')',',F7.4,'')/,2X, & & 'U('',F7.5,')',',F8.5,'')/,4X,'U('',F4.1,')',',F5.1,'')}  
920 FORMAT(/,28X,'LOADS INPUT', & & 5X,'P LOADS',5X,'T LOADS',5X,'M2 LOADS', & & 4X,'M3 LOADS',4X,'V2 LOADS',4X,'V3 LOADS', & & 2X,'STATIC AERO', & & 2X,'STAT STR ANAL FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'DYN STR ANAL FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'K CALC FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'GROWTH CALC FAC',3X,'U('',F8.5,')',',F9.5,'')/, & & 2X,'WILLENBORG OVERLOAD FACTOR',25X,E12.5, & & 2X,'STRESS-TIME HISTORY PERIOD',25X,F10.5, & & 2X,'STRESS-TIME HISTORY NOISE FILTER',16X,F7.1, & & 2X,'NUMBER OF TIME-VARYING LOADS',23X,I2, & & 2X,'NUMBER OF POINTS IN HISTORIES',19X,I4,/)  
927 FORMAT(/,2X,'Elastic modulus',32X,E9.3,)
SUBROUTINE SETDEF(LIFE, NCRL)

IMPLICIT NONE

INTEGER MAXLIF
PARAMETER (MAXLIF = 1000)
INTEGER K, NCRL
REAL LIFE(MAXLIF)

C INITIALIZE LIFE VARIABLE
DO 40 K = 1, MAXLIF
   LIFE(K) = 1.0E+36
40 CONTINUE

C SET THE NUMBER OF CRACK LENGTHS BETWEEN AI AND AF
NCRL = 25

RETURN
END

SUBROUTINE STRAN1(AERD, AERS, ASTR, ANGLE, COEXP, DLAT,
   DSTR, EM, INDIA, LAMN, LAMS, LAMW, LOCAT,
   NRAN, NU, FC, SPR, STRHIS, THIC, WOFF)

C SUBPROGRAMS: M4L1, M4L2

IMPLICIT NONE

INTEGER MAXLD, MAXM
REAL PI
PARAMETER (MAXLD = 16, MAXM = 20000, PI = 3.141592654)
COMMON IOUT
INTEGER I, II, IOUT, J, LOCAT, NLOAD, NRAN, TYPE(MAXLD)
REAL AERD, AERS, ASTR, ANGLE, COEXP, DLTAT, DSTR, EM, FK(10),
& INDIA, K(T,2), LAMN, LAMS, LAMM,
& M(2, MAXLD), MSLAM(2, MAXLD), MSTAT(2),
& NU, P(MAXLD), PC, PCO, PLAM(MAXLD), PSLAM, PSTAT,
& RT(I,10), SCLFAC, SFR(MAXM), STATIC(4),
& STRAMP(4, MAXLD), STRHIS(MAXLD, MAXM), SX(MAXLD), SXST,
& SXY(MAXLD), SXST, SXZ(MAXLD), SXZST, SY(MAXLD),
& SYST, SYZ(MAXLD), SYZST, SZ(MAXLD), SZST,
& T(MAXLD), THIC, TLAM(MAXLD), TSLAM, TSTAT,
& V(2, MAXLD), VLAM(2, MAXLD), VSLAM(2), VSTAT(2), WOFF

COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
&P, T, M, V, PCO,
& SXST, SYST, SZST, SXZST, SYZST,
& SX, SY, SZ, SXZ, SYZ
COMMON/FKVSRT/FK, RT

DATA KT/I.0, 1.0, 1.0, 1.0, 1.0/

C SCALE AERO STATIC LOADS

SCLFAC = AERS * ASTR

PSLAM = SCLFAC * PSTAT
TSLAM = 0.0
MSLAM(1) = SCLFAC * MSTAT(1)
MSLAM(2) = SCLFAC * MSTAT(2)
VSLAM(1) = 0.0
VSLAM(2) = 0.0

C SCALE TIME-VARYING LOADS

DO 230 II = 1, NLOAD
IF (TYPE(II) .EQ. 1) THEN
  SCLFAC = LAMN * DSTR
  PLAM(II) = SCLFAC * P(II)
  TLAM(II) = 0.0
  MLAM(1,II) = SCLFAC * M(1,II)
  MLAM(2,II) = SCLFAC * M(2,II)
  VLAM(1,II) = 0.0
  VLAM(2,II) = 0.0
ELSE IF (TYPE(II) .EQ. 2) THEN
  SCLFAC = LAMS * DSTR
  PLAM(II) = SCLFAC * P(II)
  TLAM(II) = 0.0
  MLAM(1,II) = SCLFAC * M(1,II)
  MLAM(2,II) = SCLFAC * M(2,II)
  VLAM(1,II) = 0.0
  VLAM(2,II) = 0.0
ELSE
  SCLFAC = AERD * ASTR
  PLAM(II) = SCLFAC * P(II)
  TLAM(II) = 0.0
  MLAM(1,II) = SCLFAC * M(1,II)
  MLAM(2,II) = SCLFAC * M(2,II)
  VLAM(1,II) = 0.0
  VLAM(2,II) = 0.0
ENDIF
230 CONTINUE

IF (IOUT .EQ. 15) THEN
  WRITE(8,*) 'AERO STATIC LOADS'
  WRITE(8,*) 'P = ', PSLAM, ' T = ', TSLAM,
  & ' M2 = ', MSLAM(1), ' M3 = ', MSLAM(2),
  & ' V2 = ', VSLAM(1), ' V3 = ', VSLAM(2)
  WRITE(8,*) 'TIME-VARYING LOADS'
  DO 240 II = 1, NLOAD
      WRITE(8,*) 'P = ', PLAM(II), ' T = ', TLAM(II),
      & ' M2 = ', MSLAM(1,II), ' M3 = ', MSLAM(2,II),
      & ' V2 = ', VSLAM(1,II), ' V3 = ', VSLAM(2,II)
 240 CONTINUE
ENDIF
IF (LOCAT .EQ. 1) THEN
   CALL M4L1 (COEXP, ANGLE, DLTAT, EM, INDIA, KT, LAMW, MLAM,
 & MSLAM, NLOAD, NU, PLAM, PC, PCO, PSLAM, STATIC,
 & STRAMP, TLAM, THIC, TSLAM, VLAM, VSLAM, WOFF, FK, RT)
ELSE IF (LOCAT .EQ. 2) THEN
   CALL \ M4L2 (COEXP, ANGLE, DLTAT, EM, INDIA, KT, LAMW, MLAM,
 & MSLAM, NLOAD, NU, PLAM, PC, PCO, PSLAM, STATIC,
 & STRAMP, TLAM, THIC, TSLAM, VLAM, VSLAM, WOFF, FK, RT)
ELSE
   WRITE(8,*), 'ERROR: INVALID LOCATION SPECIFICATION'
   CALL TRMNAT
ENDIF
C==========
.......

C =========== DERIVE THE EQUIVALENT STRESS HISTORY ===========

DO 50 J = 1, NRAN
   SPR(J) = STATIC(1)
50 CONTINUE

DO 100 I = 1, NLOAD
   DO 150 J = 1, NRAN
      SPR(J) = SPR(J) + STRHIS(I,J) * STRAMP(I,I)
150 CONTINUE
100 CONTINUE

IF (IOUT .EQ. 25) THEN
   DO 125 J = 1, NRAN
      WRITE(8,*) J, 'SPR = ', SPR(J)
125 CONTINUE
ENDIF
RETURN
END

***************************************************************************
C SUBROUTINE STRAN2 PERFORMS THE STRESS CALCULATION FOR THE EXHEX
C
C PROGRAMMER: S. SUTHARSHANA
C DATE: 19 NOV 1989
C VERSION: 92.1, 92.2, 92.3, 92.4, 92.5
C***************************************************************************

SUBROUTINE STRAN2 (DSTR, LAMN, LAMS, NRAN, SPR, SSTR, STRHIS)
C
IMPLICIT NONE
INTEGER MAXLD, MAXM
REAL PI
PARAMETER (MAXLD = 16, MAXM = 20000, PI = 3.141592654)
COMMON IOUT
INTEGER II, IOUT, J, NLOAD, NRAN, TYPE(MAXLD)
REAL DSTR, LAMN, LAMS, M(2, MAXLD),
 & MSTAT(2), P(MAXLD), PCO, PSTAT, SPR(MAXM), SSTR,
 & STRAMP(MAXLD), STRHIS(MAXLD, MAXM), SX(MAXLD), SXST,
 & SXY(MAXLD), SXYST, SXZ(MAXLD), SXZST, SY(MAXLD),
 & SYST, SYZ(MAXLD), SYZST, SZ(MAXLD), SZST,
 & T(MAXLD), TSTAT, V(2, MAXLD), VSTAT(2)
COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
 & P, T, M, V, PCO,
& SXST, SYST, SZST, SXYST, SXZST, SYZST, 
& SX, SY, SZ, SXY, SXZ, SYZ

C SET UP THE STRESS AMPLITUDES
DO 50 II = I, NLOAD
IF (TYPE(II).EQ.1) THEN
  STRAMP(II) = LAMN * DSTR * SZ(II)
ENDIF
IF (TYPE(II).EQ.2) THEN
  STRAMP(II) = LAMS * DSTR * SZ(II)
ENDIF
50 CONTINUE

C ASSIGN STATIC LOADS
DO 100 J = I, NRAN
  SPR(J) = SZST * SSTR
100 CONTINUE

C SCALE TIME-VARYING LOADS
DO 300 II = I, NLOAD
  DO 200 J = I, NRAN
    SPR(J) = SPR(J) + STRHIS(II,J) * STRAMP(II)
 200 CONTINUE
300 CONTINUE

IF (IOUT.EQ.25) THEN
  DO 425 J = I, NRAN
    WRITE(8,*) J, 'SPR = ', SPR(J)
 425 CONTINUE
ENDIF
RETURN
END

*******************************************************************************
C THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: GAM
C
The random variates are generated using the method described in:
C Univariate Distributions - 1, Houghton Mifflin Company, 1970,
C pp. 181-182.
*******************************************************************************

SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)

COMMON IOUT
DOUBLE PRECISION RAND
REAL A, B, GAM, RHO, THETA, W, X, Y1, Y2
INTEGER IOUT

IF (IOUT.EQ.15) WRITE(8,*) 'RAND =', RAND, ' RHO =', RHO,
Y1 = GAM((RHO * THETA + 1.), RAND)
Y2 = GAM(((1. - RHO) * THETA + 1.), RAND)
W = Y1 / (Y1 + Y2)
C IF (IOUT.EQ.15) WRITE(8,*) 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION
X = W * (B - A) + A
IF (IOUT.EQ.15) WRITE(8,*) 'W =', W, ' X =', X
RETURN
END
REAL FUNCTION GAM (ALPHA, RAND)

SUBPROGRAM: RANDOM

COMMON IOUT

INTEGER IOUT

REAL A, ALPHA, ARG, U1, U2, V1, V2

DOUBLE PRECISION RAND

A = ALPHA - 1.

IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA

CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALOG(U1)
V2 = - ALOG(U2)

IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =', V1, ' V2 =', V2

ARG = A * (V1 - ALOG(V1) - 1.)

IF (V2 .LT. ARG) GOTO 10

GAM = ALPHA * V1

GOTO 10

IF (IOUT .EQ. 15) WRITE(8,*) 'GAMMA =', GAM

RETURN

END
Section 7.2
Low Cycle Fatigue Failure Program BLDLCF

The program tree structures, list of subprograms, descriptions of the key variables, and the FORTRAN source listings for the low cycle fatigue analysis code BLDLCF are given here. The pertinent LCF methodology is given in Section 3. The overall description of the program and the flowcharts are given in Section 5.2. The user's guide for running BLDLCF is given in Section 6.2.

7.2.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for BLDLCF, using Uniform variation on the materials shape parameter $m$, is given in Figure 7.2-1, while the tree structure for the truncated Normal case is given in Figure 7.2-2. The tree structure for BLDLCF V3.4B1.3 is given in Figure 7.2-3. In all trees, those subprograms not "shadow-boxed" are part of the materials characterization model. The program, subprogram, and file names are indicated by UPPERCASE letters.

7.2.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7.2-1. The section numbers where the subprograms are described by means of flowcharts are given next to the names.
Figure 7.2-1  Tree Structure for Program BLDLCF for the Uniform Variation in Materials Shape Parameter $m$
Figure 7.2-2  Tree Structure for Program BLDLCF for the Truncated Normal Variation in Materials Shape Parameter $m$
Figure 7.2-3  Tree Structure for Program BLDLCF V3.4B1.3 for the Bootstrapping of the Materials Shape Parameter $m$
<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDREG¹</td>
<td>4.1.3.9*</td>
<td>Adds the m ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.</td>
</tr>
<tr>
<td>ADDRGN²</td>
<td>4.1.3.15*</td>
<td>Adds the m ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.</td>
</tr>
<tr>
<td>BETAGN³</td>
<td>4.4.5³</td>
<td>Generates Beta(a, b, p, θ) random variates.</td>
</tr>
<tr>
<td>BLDLCF</td>
<td>5.2.2.1</td>
<td>The main routine that controls the logical flow of the low cycle fatigue turbine blade program.</td>
</tr>
<tr>
<td></td>
<td>5.2.3.1</td>
<td>The main routine that controls the logical flow of the low cycle fatigue turbine blade program with the nonparametric materials characterization model.</td>
</tr>
<tr>
<td>BDLIF</td>
<td>5.2.2.2</td>
<td>Performs the calculations of the driver transformation and then calls RAINF3 to calculate the fatigue life.</td>
</tr>
<tr>
<td>CONCAV⁴</td>
<td>4.1.3.10*</td>
<td>Adjusts the upper bound of the posterior ranges on m to be consistent with concavity constraints.</td>
</tr>
<tr>
<td>CONVRT⁵</td>
<td>4.1.3.3*</td>
<td>Transforms strain data to equivalent zero-mean strains with strain ratio of -1.0.</td>
</tr>
<tr>
<td>EXPB</td>
<td>5.2.3.4</td>
<td>Calculates the median S/N curve parameters from the results of the linear regression and residual calculations of Section 3.2.7.</td>
</tr>
<tr>
<td>EXPCTD⁶</td>
<td>4.1.3.12*</td>
<td>Calculates the median S/N curve parameters from the results of the information aggregation calculations.</td>
</tr>
<tr>
<td>FINDK</td>
<td>4.1.5.6*</td>
<td>Calculates the value of the location parameter K (where A = K⁰) for each life region by using Equations 2-37 and 2-41 of [1].</td>
</tr>
<tr>
<td>FINDM⁶</td>
<td>4.1.5.1*</td>
<td>Obtains the value of m for each life region by adjusting the range (to ensure concavity) and then sampling from the Uniform distribution over the appropriate m range.</td>
</tr>
<tr>
<td>FINDMC</td>
<td>4.1.3.5*</td>
<td>Calculates the m range implied by the constraint on the coefficient of variation of fatigue strength, C, for each life region, by using Equations 2-28 through 2-32 of [1].</td>
</tr>
<tr>
<td>FINDMN⁷</td>
<td>4.1.5.2*</td>
<td>Obtains the value of m for each life region by sampling from the appropriate truncated Normal distribution on m.</td>
</tr>
<tr>
<td>FINDSB</td>
<td>4.1.5.7*</td>
<td>Calculates the life region “tie-points” or strain values which correspond to the “life boundaries,” conditional on the randomly selected m for each region. Also calculates K, characterizing the specific material S/N data set, which is a function of β₀, and k.</td>
</tr>
<tr>
<td>FNDRNG⁷</td>
<td>4.1.3.8*</td>
<td>Combines the 95% confidence interval, J₀, with the implicit and explicit constraints on m, to obtain posterior credibility ranges on m for each life region.</td>
</tr>
</tbody>
</table>

Table 7.2-1 List of Subprograms For Program BLDLCF
(Footnotes are at the end of the table)
Table 7.2-1  List of Subprograms For Program BDLCF (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAM</td>
<td>4.4.4*</td>
<td>Generates Gamma(α, 1) random variates.</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>4.1.8*</td>
<td>Calculates the cycles to failure for a particular strain, based upon the materials characterization model S/N curve of Equation 2-48 of [1].</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>4.1.3.7*</td>
<td>Calculates σ^2, the extent of departures from the multiple heat median S/N curve warranted by the information available, by using Equation 2-49 of [1].</td>
</tr>
<tr>
<td>INFAGG</td>
<td>5.2.3.2</td>
<td>Controls the logical flow for the information aggregation portion of the materials characterization model.</td>
</tr>
<tr>
<td>INIT</td>
<td>4.1.3.1*</td>
<td>Initializes the entries of the arrays used in the information aggregation subroutine, INFAGG, to zero.</td>
</tr>
<tr>
<td>INSORT</td>
<td>5.B*</td>
<td>Performs an insertion sort for the lowest fifty percent of the lives calculated.</td>
</tr>
<tr>
<td>INTRVL</td>
<td>4.1.3.6*</td>
<td>Calculates the 95% confidence intervals J_o for C, and J_o for m, for each region by using Equations 2-24 through 2-26 of [1].</td>
</tr>
<tr>
<td>KBETA</td>
<td>4.1.5.5*</td>
<td>Calculates k and β_o from the sample mean and variance of Z, where Z is a function of strain, life, the life region boundaries, and the m's, by using Equation 2-42 of [1].</td>
</tr>
<tr>
<td>KOMO</td>
<td>4.1.6*</td>
<td>Calculates K_o and m_o for the zero region, the no data region to the left of the first data region. Extends the S/N curve consistent with the tensile point at S_o.</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>4.1.3.11*</td>
<td>Calculates the median values of m, based on the posterior credibility ranges of m, by using Equation 2-34 of [1].</td>
</tr>
<tr>
<td>MREGR</td>
<td>5.2.3.7</td>
<td>Performs the regression to obtain the parameter m for the non-parametric materials characterization model.</td>
</tr>
<tr>
<td>MUSIG</td>
<td>4.1.3.13*</td>
<td>Calculates the posterior Normal distribution parameters, mean m_*, and standard deviation σ_m, for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NORMGN</td>
<td>4.4.3*</td>
<td>Generates Normal(μ, σ^2) random variates.</td>
</tr>
<tr>
<td>NORRNG</td>
<td>4.1.3.14*</td>
<td>Combines the implicit and explicit constraints on m to obtain the posterior credibility ranges of m for each life region.</td>
</tr>
<tr>
<td>PAREST</td>
<td>4.1.5*</td>
<td>Controls the logical flow for the parameter estimation model portion of the materials characterization model.</td>
</tr>
<tr>
<td>PEB</td>
<td>5.2.3.5</td>
<td>Controls the logical flow of the bootstrapping portion of the non-parametric materials characterization model described in Section 3.2.7.</td>
</tr>
<tr>
<td>PICRES</td>
<td>5.2.3.6</td>
<td>Bootstraps the residuals and performs the pseudo S/N data generation described in Section 3.2.7.</td>
</tr>
<tr>
<td>NAME</td>
<td>SECTION</td>
<td>PURPOSE</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PRYRV$^{13}$</td>
<td>7.6.6</td>
<td>Generates the Uniform($a$, $b$) and Uniform($c$, $d$) pair of independent random variates.</td>
</tr>
<tr>
<td>RAINF3</td>
<td>5.2.2.3</td>
<td>Performs rainflow cycle counting, Miner's rule damage accumulation, and calls GTLIFE to calculate the fatigue life.</td>
</tr>
<tr>
<td>RANDOM$^{13}$</td>
<td>4.4.2</td>
<td>Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.</td>
</tr>
<tr>
<td>RCE</td>
<td>4.1.3.2</td>
<td>Reads the data from BLDLCD and RELATD; calls CONVRT to transform the strain data to a strain ratio of -1.0; and echoes the data to BDLCO and RELATO. RCE also breaks S/N data sets into regions as specified by the user.</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>4.1.5.4</td>
<td>Calculates the sample mean and variance of $Z$, where $Z$ is a function of strain, life, the life region boundaries, and the $m$'s, by using Equation 2-42 of [1].</td>
</tr>
<tr>
<td>SORTM$^{14}$</td>
<td>4.1.10</td>
<td>Sorts the $m$ values in increasing order for each life region for the truncated Normal distribution case.</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>5.2.3.3</td>
<td>Calculates the residual variances from the $Y$ on $X$ and $X$ on $Y$ regressions for each life region where $Y = \ln(Endurance cycles)$ and $X = \ln(Strain)$ by using Equations 2-20 and 2-21 of [1]; to be used in the credibility range calculations.</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>4.1.11</td>
<td>Performs premature program termination when required.</td>
</tr>
<tr>
<td>TRNSFM$^{15}$</td>
<td>4.1.5.3</td>
<td>Performs the calculations necessary to transform the specific material S/N data into the variable $Z$, where $Z$ is a function of strain, life, the life region boundaries, and the $m$'s.</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>4.4.6</td>
<td>Generates Weibull($\beta$, $\eta(\beta)$) random variates.</td>
</tr>
<tr>
<td>WORSTN</td>
<td>5.2.3.8</td>
<td>Performs the &quot;worst of $N$&quot; selection described in Section 3.2.7.3 for both Weibull and Lognormal distributions.</td>
</tr>
</tbody>
</table>
See [1].

1 No data regions to the right are discussed in [1], Page 2-17.
2 The Beta distribution is discussed in [1], Page 2-25.
3 Concavity constraints are discussed in [1], Pages 2-13 through 2-14.
4 The strain transformation is discussed in [1], Page 2-7.
5 The median S/N curve parameter estimation calculations are described in [1], Pages 2-15 through 2-18.
6 Selection of the \{m_j\} parameters is discussed in [1], Page 2-15.
7 Combining information to obtain the posterior credibility ranges on \(m\) is discussed in [1], Page 2-13.
8 The information aggregation calculations are discussed in [1], Pages 2-6 through 2-14.
9 Extension of the S/N curve to the left is discussed in [1], Page 2-17.
10 Calculation of the truncated Normal distribution parameters is discussed in [1], Page 2-14.
11 The Normal distribution is discussed in [1], Page 2-23.
12 The parameter estimation calculations are discussed in [1], Pages 2-15 through 2-18.
13 The Uniform distribution is discussed in [1], Page 2-23.
14 The need for saving \(m\)'s is discussed in [1], Page 2-15.
15 The S/N data transformation is discussed in [1], Page 2-16.
7.2.3 Description of Variables

A list of variables used in the ATD-HPFTP first stage turbine blade LCF code, BLDLCF, is given in Table 7.2-2. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" can be interpreted as follows: INT is a standard integer variable; RE is a standard real variable; and DRE is a double precision variable. The various array dimensions are defined by using the following parameters: **MAXBLF, MAXDAT, MAXLIF, MAXM, MAXMM, and MAXREG.**

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLM(MAXMM, MAXREG)</td>
<td>RE</td>
<td>2-D array containing the materials model shape parameters ($m'$s) for each life region which are to be used in the truncated Normal median S/N curve calculation.(^1)</td>
</tr>
<tr>
<td>BIGK(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the materials model location parameter $K$, where $A = K^m$, given in Equation 2-12 of [1].</td>
</tr>
<tr>
<td>BIGK1</td>
<td>RE</td>
<td>Dummy variable used during calls to subroutine EXPCTD, equal to BIGK(1).</td>
</tr>
<tr>
<td>BLDLIF</td>
<td>RE</td>
<td>Real function that performs the calculations of the driver transformation, calls RAINF3 to calculate a fatigue life, and returns the fatigue life (missions).</td>
</tr>
<tr>
<td>BLFPER(MAXBLF)</td>
<td>RE</td>
<td>1-D array containing user specified B-lives which are obtained from the simulated failure distribution. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent: e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.</td>
</tr>
<tr>
<td>BLFPOS(MAXBLF)</td>
<td>INT</td>
<td>1-D array containing the indices for the array variable LIFE( ) corresponding to the user-requested simulated failure distribution B-lives contained in variable BLFPER( ).</td>
</tr>
<tr>
<td>BZERO</td>
<td>RE</td>
<td>Estimate of Weibull distribution shape parameter $\beta_o$, that characterizes the intrinsic variation of the S/N data set, by using Equation 2-11 of [1].</td>
</tr>
<tr>
<td>DUM</td>
<td>RE</td>
<td>Dummy variable.</td>
</tr>
<tr>
<td>EBEND</td>
<td>RE</td>
<td>The randomly selected value for $\varepsilon_B$, the bending strain due to gas bending and blade tilt, given in Equation 3-1.</td>
</tr>
</tbody>
</table>

Table 7.2-2 List of Variables For Program BLDLCF
(Footnotes are at the end of the table)
<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBENDA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\varepsilon_B$.</td>
</tr>
<tr>
<td>EBENDB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\varepsilon_B$.</td>
</tr>
<tr>
<td>EM(MAXM)</td>
<td>RE</td>
<td>1-D array containing the total mechanical strain-time history, $\varepsilon_M(t_i) (%)$, in Equation 3-1.</td>
</tr>
<tr>
<td>EMNOM</td>
<td>RE</td>
<td>$\varepsilon_{Mnom} (%)$ in Equation 3-5, the nominal mechanical strain.</td>
</tr>
<tr>
<td>EPSL</td>
<td>RE</td>
<td>$\varepsilon$ in Equation 3-9, the material's intrinsic variation or scatter, given by a Lognormal random variate.</td>
</tr>
<tr>
<td>EPSW</td>
<td>RE</td>
<td>$\varepsilon$ in Equation 3-9, the material's intrinsic variation or scatter, given by a Weibull random variate.</td>
</tr>
<tr>
<td>ETH(MAXM)</td>
<td>RE</td>
<td>1-D array containing the total thermal strain-time history $\varepsilon_{TH}(t_i) (%)$ in Equation 3-1.</td>
</tr>
<tr>
<td>ETHNOM(MAXM)</td>
<td>RE</td>
<td>$\varepsilon_{THnom}(t_i) (%)$ in Equation 3-4, the 1-D array containing the nominal thermal strain-time history.</td>
</tr>
<tr>
<td>ETOT(MAXM)</td>
<td>RE</td>
<td>1-D array containing the total strain-time history, $\varepsilon_T(t_i) (%)$, in Equation 3-1.</td>
</tr>
<tr>
<td>FA</td>
<td>RE</td>
<td>$f_A(T_{gas}, h_{gas}) + e_A$ in Equation 3-2, the acceleration response surface.</td>
</tr>
<tr>
<td>FAA, FAB, FAC, FAD, FAE, FAF</td>
<td>RE</td>
<td>The coefficients for the acceleration response surface $f_A(T_{gas}, h_{gas})$ in Equation 3-2.</td>
</tr>
<tr>
<td>FACTR</td>
<td>RE</td>
<td>Equal to FACTOR = PHI * KRATIO * Z. Used by the materials model.</td>
</tr>
<tr>
<td>FAERRM</td>
<td>RE</td>
<td>Mean, $\mu$, of Normally distributed $e_A$, the additive modeling uncertainty for the acceleration response surface, given in Equation 3-2.</td>
</tr>
<tr>
<td>FAERRS</td>
<td>RE</td>
<td>Standard deviation, $\sigma$, of Normally distributed $e_A$, the additive modeling uncertainty for the acceleration response surface, given in Equation 3-2.</td>
</tr>
<tr>
<td>FD1</td>
<td>RE</td>
<td>$f_{D1}(m, T_s) + e_D$ in Equation 3-3, the deceleration response surface for the thermal strain.</td>
</tr>
<tr>
<td>FD1A, FD1B, FD1C, FD1D, FD1E, FD1F</td>
<td>RE</td>
<td>The coefficients for the deceleration response surface $f_{D1}(m, T_s)$ in Equation 3-3.</td>
</tr>
<tr>
<td>FD2</td>
<td>RE</td>
<td>$f_{D2}(m, T_s)$ in Equation 3-6, the deceleration response surface for the time of deceleration $t_d$.</td>
</tr>
<tr>
<td>FD2A, FD2B</td>
<td>RE</td>
<td>The coefficients for the deceleration response surface $f_{D2}(m, T_s)$ in Equation 3-6.</td>
</tr>
</tbody>
</table>
Table 7.2-2 List of Variables For Program BLDLCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD3</td>
<td>RE</td>
<td>( f_{D3}(t_d) ) in Equation 3-7, the deceleration response surface for the rotor speed ( \omega(t_6) ).</td>
</tr>
<tr>
<td>FD3A, FD3B</td>
<td>RE</td>
<td>The coefficients for the deceleration response surface ( f_{D3}(t_d) ) in Equation 3-7.</td>
</tr>
<tr>
<td>FDERRM</td>
<td>RE</td>
<td>Mean, ( \mu ), of Normally distributed ( \theta_D ), the additive modeling uncertainty for the deceleration response surface, given in Equation 3-3.</td>
</tr>
<tr>
<td>FDERRS</td>
<td>RE</td>
<td>Standard deviation, ( \sigma ), of Normally distributed ( \theta_D ), the additive modeling uncertainty for the deceleration response surface, given in Equation 3-3.</td>
</tr>
<tr>
<td>FIFTY</td>
<td>RE</td>
<td>Variable used to access the fifty-percent point in the ( \text{LIFE()} ) array.</td>
</tr>
<tr>
<td>FTU</td>
<td>RE</td>
<td>Material ultimate strength (%).</td>
</tr>
<tr>
<td>FTY</td>
<td>RE</td>
<td>Material yield strength (%).</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>RE</td>
<td>Function given by Equation 2-48 of [1] that calculates the fatigue cycles to failure at a given strain.</td>
</tr>
<tr>
<td>HGAS</td>
<td>RE</td>
<td>( h_{gas} ) in Equation 3-2, the randomly selected gas film coefficient.</td>
</tr>
<tr>
<td>HGASA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on ( h_{gas} ).</td>
</tr>
<tr>
<td>HGASB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on ( h_{gas} ).</td>
</tr>
<tr>
<td>HGASR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for ( h_{gas} ).</td>
</tr>
<tr>
<td>HGASR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \rho ) in the Beta distribution of ( h_{gas} ).</td>
</tr>
<tr>
<td>HGASR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \rho ) in the Beta distribution of ( h_{gas} ).</td>
</tr>
<tr>
<td>HGAST</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter ( \theta ) for ( h_{gas} ).</td>
</tr>
<tr>
<td>HGAST1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the Beta distribution of ( h_{gas} ).</td>
</tr>
<tr>
<td>HGAST2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the Beta distribution of ( h_{gas} ).</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls inner DO loop.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls DO loop for each point in the time history.</td>
</tr>
</tbody>
</table>
Table 7.2-2  List of Variables For Program BLDLCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller. IOUT = 0, no intermediate calculation output; IOUT = 10, materials characterization model calculations; IOUT = 15, driver sampling and driver transformation calculations; and IOUT = 20, rainflow cycle counting and damage accumulation.</td>
</tr>
<tr>
<td>J</td>
<td>INT</td>
<td>Controls DO loop for each B-life.²</td>
</tr>
<tr>
<td>K</td>
<td>INT</td>
<td>Controls outer DO loop.</td>
</tr>
<tr>
<td>KRATIO</td>
<td>RE</td>
<td>Ratio of MED K*/MED K in Equation 2-48 of [1]. KRATIO is constant over life regions for the materials model.</td>
</tr>
<tr>
<td>L</td>
<td>INT</td>
<td>Controls DO loop for each life region of the S/N curve.</td>
</tr>
<tr>
<td>LAMA</td>
<td>RE</td>
<td>$\lambda_\alpha$ in Equation 3-4, the randomly selected uncertainty factor for the coefficient of thermal expansion.</td>
</tr>
<tr>
<td>LAMAA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda_\alpha$.</td>
</tr>
<tr>
<td>LAMAB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\lambda_\alpha$.</td>
</tr>
<tr>
<td>LAMDA</td>
<td>RE</td>
<td>$\lambda_{dam}$ in Equation 2-91 of [1], the randomly selected damage accumulation model accuracy factor. See [1], Section 2.2.1.4, for a discussion of the damage calculations.</td>
</tr>
<tr>
<td>LAMDAAB</td>
<td>RE</td>
<td>Uniform distribution lower bound of the damage accumulation model accuracy factor.</td>
</tr>
<tr>
<td>LAMDAB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the damage accumulation model accuracy factor.</td>
</tr>
<tr>
<td>LAMG</td>
<td>RE</td>
<td>$\lambda_G$ in Equation 3-4, the randomly selected thermal strain uncertainty factor due to gas temperature variation during start.</td>
</tr>
<tr>
<td>LAMGA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on $\lambda_G$.</td>
</tr>
<tr>
<td>LAMGB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on $\lambda_G$.</td>
</tr>
<tr>
<td>LAMGR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for $\lambda_G$.</td>
</tr>
<tr>
<td>LAMGR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $\lambda_G$.</td>
</tr>
<tr>
<td>LAMGR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $\lambda_G$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>LAMGT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ for $\lambda_G$.</td>
</tr>
<tr>
<td>LAMGT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $\lambda_G$.</td>
</tr>
<tr>
<td>LAMGT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $\lambda_G$.</td>
</tr>
<tr>
<td>LAMP</td>
<td>RE</td>
<td>$\lambda_p$ in Equation 3-5, the randomly selected deviation in blade pull load due to uncertainty in blade mass.</td>
</tr>
<tr>
<td>LAMPA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda_p$.</td>
</tr>
<tr>
<td>LAMPB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\lambda_p$.</td>
</tr>
<tr>
<td>LAMTM</td>
<td>RE</td>
<td>$\lambda_{TMF}$ in Section 3.2.6, the randomly selected thermal-mechanical fatigue (TMF) model accuracy factor.</td>
</tr>
<tr>
<td>LAMTMA</td>
<td>RE</td>
<td>Uniform distribution lower bound of the TMF model accuracy factor.</td>
</tr>
<tr>
<td>LAMTMB</td>
<td>RE</td>
<td>Uniform distribution upper bound of the TMF model accuracy factor.</td>
</tr>
<tr>
<td>LIFE(MAXLIF)</td>
<td>RE</td>
<td>1-D array containing values of the lives generated by program BDLDCF. The lives are sorted values for the left-hand tail simulated failure distribution.</td>
</tr>
<tr>
<td>LIFEL(MAXLIF)</td>
<td>RE</td>
<td>1-D array containing values of the lives generated by program BDLDCF V3.4B1.3 for Lognormal intrinsic materials variation. The lives are sorted values for the left-hand tail simulated failure distribution.</td>
</tr>
<tr>
<td>LIFEW(MAXLIF)</td>
<td>RE</td>
<td>1-D array containing values of the lives generated by program BDLDCF V3.4B1.3 for Weibull intrinsic materials variation. The lives are sorted values for the left-hand tail simulated failure distribution.</td>
</tr>
<tr>
<td>LIFL</td>
<td>RE</td>
<td>Fatigue life value (missions) equal to $\text{EPSL} \times \text{NEWLIF}$ to be inserted in LIFEL( ) for the non-parametric materials characterization model with Lognormal intrinsic materials variation.</td>
</tr>
<tr>
<td>LIFW</td>
<td>RE</td>
<td>Fatigue life value (missions) equal to $\text{EPSW} \times \text{NEWLIF}$ to be inserted in LIFEW( ) for the non-parametric materials characterization model with Weibull intrinsic materials variation.</td>
</tr>
<tr>
<td>LNA(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of $\ln(A) = \ln(\text{BIGK}) \times \text{MM}$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LNPHI</td>
<td>RE</td>
<td>The natural logarithm of $\varphi$ in Equation 2-11 of [1], the material's intrinsic variation, or scatter, given by a Lognormal($0$, PHISIG$^2$) random variate.</td>
</tr>
<tr>
<td>LNZ</td>
<td>RE</td>
<td>$\ln(Z)$ in Equation 2-48 of [1], the Normal($0$, PVAR) random variate for the materials process variation aspect of the materials model.</td>
</tr>
<tr>
<td>LPHIM(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of $\ln(\Phi) \times MM$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>M</td>
<td>INT</td>
<td>Controls symmetry DO loop.</td>
</tr>
<tr>
<td>MANAL</td>
<td>RE</td>
<td>The randomly selected mechanical strain analysis accuracy factor, $\lambda_{MA}$ in Equation 3-5.</td>
</tr>
<tr>
<td>MANALA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda_{MA}$.</td>
</tr>
<tr>
<td>MANALB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\lambda_{MA}$.</td>
</tr>
<tr>
<td>MAXBLF</td>
<td>INT</td>
<td>Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.</td>
</tr>
<tr>
<td>MAXDAT</td>
<td>INT</td>
<td>Maximum number of points per data set per region allowed for the S/N curve. The maximum number of data points per set allowed is 50.</td>
</tr>
<tr>
<td>MAXLIF</td>
<td>INT</td>
<td>Maximum number of fatigue lives allowed for the simulated failure distribution. The maximum number of fatigue lives to be saved is 10,000.</td>
</tr>
<tr>
<td>MAXM</td>
<td>INT</td>
<td>Maximum number of points allowed in the time history arrays. The maximum number of points is 50.</td>
</tr>
<tr>
<td>MAXMM</td>
<td>INT</td>
<td>Maximum number of $m$'s to be saved and sorted for the truncated Normal median S/N curve.$^1$ The maximum number of $m$'s is 20,000.</td>
</tr>
<tr>
<td>MAXREG</td>
<td>INT</td>
<td>Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.</td>
</tr>
<tr>
<td>MCOUNT</td>
<td>INT</td>
<td>Counts number of $m$'s to be used to calculate the median S/N curve for the truncated Normal distribution case.$^1$</td>
</tr>
<tr>
<td>MEDKB(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the median $K$ for each life region of the S/N curve for the bootstrapping option.</td>
</tr>
<tr>
<td>MEDM(MAXMM)</td>
<td>RE</td>
<td>1-D array containing the empirical median $m$ for each life region of the S/N curve.$^3$</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>MEDMB(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the median m for each life region of the S/N curve for the bootstrapping option.</td>
</tr>
<tr>
<td>MID</td>
<td>INT</td>
<td>Pointer to the median m values in array SORTM() for the truncated Normal median S/N curve. Value of half of MCOUNT.</td>
</tr>
<tr>
<td>MINPHI</td>
<td>RE</td>
<td>Value of min(PHI), the minimum of NSYM draws of the materials scatter parameter ( \varphi ).</td>
</tr>
<tr>
<td>MM(0:MAXREG)</td>
<td>RE</td>
<td>( m_j ) in Equation 2-12 of [1], the 1-D array containing randomly selected values of the materials model shape parameter m for each life region of the S/N curve.</td>
</tr>
<tr>
<td>MODER1</td>
<td>RE</td>
<td>( e_A ) in Equation 3-2, the randomly selected additive modeling uncertainty for the acceleration response surface.</td>
</tr>
<tr>
<td>MODER2</td>
<td>RE</td>
<td>( e_D ) in Equation 3-3, the randomly selected additive modeling uncertainty for the deceleration response surface.</td>
</tr>
<tr>
<td>MPROC</td>
<td>INT</td>
<td>Materials PROCess variation. Controls materials process variation. A value of 0 indicates no materials process variation, while a value of 1 indicates that materials process variation should be included.</td>
</tr>
<tr>
<td>MU(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution mean of the materials shape parameter m for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>NBLIFE</td>
<td>INT</td>
<td>Number of B-lives to be obtained from the simulated failure distribution.</td>
</tr>
<tr>
<td>NBND(0:MAXREG)</td>
<td>RE</td>
<td>( N_{*, i, i+1} ) in Equation 2-35 of [1], the 1-D array containing upper bounds for the NUMREG life regions of interest for the specific material S/N data set.</td>
</tr>
<tr>
<td>NEWLIF</td>
<td>RE</td>
<td>Fatigue life value (missions) returned from call to function BDLIF.</td>
</tr>
<tr>
<td>NF(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values from the array RAWNF() for the specific material S/N data set partitioned into life regions.</td>
</tr>
<tr>
<td>NHYPER</td>
<td>INT</td>
<td>The outer loop size.</td>
</tr>
<tr>
<td>NLIFE</td>
<td>INT</td>
<td>The inner loop size.</td>
</tr>
</tbody>
</table>
Table 7.2-2  List of Variables For Program BLDLCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLIFET</td>
<td>INT</td>
<td>Total number of lives calculated by program BDLDCF. Value of NHYPER * NLIFE.</td>
</tr>
<tr>
<td>NMED</td>
<td>INT</td>
<td>Controls S/N curve median calculation for the truncated Normal distribution case. A value of 0 indicates that the user does not desire a median calculation or that the Uniform distribution case is being used; while a value of 1 indicates that the user desires the median calculation to be performed.</td>
</tr>
<tr>
<td>NOMSPD</td>
<td>RE</td>
<td>( \omega_0 ) (rpm) in Equation 3-5, the nominal rotor speed.</td>
</tr>
<tr>
<td>NPTS(MAXREG)</td>
<td>INT</td>
<td>1-D array containing the number of points per life region for the specific material S/N data set.</td>
</tr>
<tr>
<td>NSYM</td>
<td>INT</td>
<td>Symmetry number, usually equal to the multiplicity of the modeling unit in the component.</td>
</tr>
<tr>
<td>NTIME</td>
<td>RE</td>
<td>Number of points in strain-time history.</td>
</tr>
<tr>
<td>NUMREG</td>
<td>INT</td>
<td>R in Equation 2-11 of [1], the number of life regions of interest in the S/N curve.</td>
</tr>
<tr>
<td>PERIOD</td>
<td>RE</td>
<td>T (missions) in Equation 2-91 of [1], the length of time in missions of the strain-time history.</td>
</tr>
<tr>
<td>PHI</td>
<td>RE</td>
<td>( \varphi ) in Equation 2-11 of [1], the material's Intrinsic variation, or scatter, given by a Weibull((\beta_0, \eta_0(\beta_0))) random variate.</td>
</tr>
<tr>
<td>PHISIG</td>
<td>RE</td>
<td>( \sigma ) in the distribution ( \Lambda(0, \sigma^2) ) of Section 3.2.7.2, a parameter of the Lognormal distribution of the Intrinsic materials variation.</td>
</tr>
<tr>
<td>PSIG</td>
<td>RE</td>
<td>( \sigma ) in Equation 2-48 of [1], the value of SQRT(PVAR).</td>
</tr>
<tr>
<td>PVAR</td>
<td>RE</td>
<td>( \sigma^2 ) in Equation 2-48 of [1], characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.</td>
</tr>
<tr>
<td>RAINF3</td>
<td>RE</td>
<td>Real function which performs rainflow cycle counting, Miner's Rule damage accumulation, and calls GTLIFE to calculate the fatigue life.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>RANGEM(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the posterior credibility ranges on the materials model shape parameter m for each life region in the S/N curve. RANGEM(1,L) is the lower bound and RANGEM(2,L) is the upper bound in region L.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RESID(MAXDAT)</td>
<td>RE</td>
<td>1-D array containing the values of the residuals of the regression for each point in the specific material S/N data for the bootstrapping option.</td>
</tr>
<tr>
<td>RESNF(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of N for the generated pseudo S/N data for the bootstrapping option.</td>
</tr>
<tr>
<td>RPM(MAXM)</td>
<td>RE</td>
<td>1-D array containing $\omega(t_j)$ (rpm) in Equation 3-5, the rotor speed time history.</td>
</tr>
<tr>
<td>SBND(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the strain values (%) with strain ratio = -1.0, corresponding to the “life boundary” values for each life region of the S/N curve contained in array NBND().</td>
</tr>
<tr>
<td>SIG(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution standard deviation of the materials model shape parameter $m$ for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>SLOPE</td>
<td>RE</td>
<td>The randomly selected deceleration slope at shutdown, $m$ (°R/sec) in Equation 3-3.</td>
</tr>
<tr>
<td>SLOPEA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on $m$.</td>
</tr>
<tr>
<td>SLOPEB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on $m$.</td>
</tr>
<tr>
<td>SLOPR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for $m$.</td>
</tr>
<tr>
<td>SLOPR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $m$.</td>
</tr>
<tr>
<td>SLOPR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $m$.</td>
</tr>
<tr>
<td>SLOPT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ for $m$.</td>
</tr>
<tr>
<td>SLOPT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $m$.</td>
</tr>
<tr>
<td>SLOPT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $m$.</td>
</tr>
<tr>
<td>SPEED</td>
<td>RE</td>
<td>$\omega(t_\mu)$ (rpm) in Equation 3-5, the randomly selected steady state rotor speed.</td>
</tr>
<tr>
<td>SPEEDM</td>
<td>RE</td>
<td>Mean, $\mu$, of Normally distributed steady state rotor speed (rpm).</td>
</tr>
<tr>
<td>SPEEDS</td>
<td>RE</td>
<td>Standard deviation, $\sigma$, of Normally distributed steady state rotor speed (rpm).</td>
</tr>
</tbody>
</table>
## Table 7.2-2  List of Variables For Program BLDLCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>STR(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing strain points with strain ratio = -1.0, for the specific material S/N data set partitioned into life regions.</td>
</tr>
<tr>
<td>SZERO</td>
<td>RE</td>
<td>Strain tensile test point, ( S_o ) (%) (^8).</td>
</tr>
<tr>
<td>TANAL</td>
<td>RE</td>
<td>The randomly selected thermal strain analysis accuracy factor, ( \lambda_{TA} ) in Equation 3-4.</td>
</tr>
<tr>
<td>TANALA</td>
<td>RE</td>
<td>Uniform distribution lower bound of ( \lambda_{TA} ).</td>
</tr>
<tr>
<td>TANALB</td>
<td>RE</td>
<td>Uniform distribution upper bound of ( \lambda_{TA} ).</td>
</tr>
<tr>
<td>TGAS</td>
<td>RE</td>
<td>( T_{gas} ) (°R) in Equation 3-2, the randomly selected gas temperature at ( t_1 ).</td>
</tr>
<tr>
<td>TGASA</td>
<td>RE</td>
<td>Lower bound of the Beta distribution on ( T_{gas} ).</td>
</tr>
<tr>
<td>TGASB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on ( T_{gas} ).</td>
</tr>
<tr>
<td>TGASR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for ( T_{gas} ).</td>
</tr>
<tr>
<td>TGASR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \rho ) in the Beta distribution of ( T_{gas} ).</td>
</tr>
<tr>
<td>TGASR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \rho ) in the Beta distribution of ( T_{gas} ).</td>
</tr>
<tr>
<td>TGAST</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter ( \theta ) for ( T_{gas} ).</td>
</tr>
<tr>
<td>TGAST1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter ( \theta ) in the Beta distribution of ( T_{gas} ).</td>
</tr>
<tr>
<td>TGAST2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter ( \theta ) in the Beta distribution of ( T_{gas} ).</td>
</tr>
<tr>
<td>TRBIGK(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the materials model location parameter ( K ) consistent with the tensile point ( S_o ) (^8).</td>
</tr>
<tr>
<td>TRSBND(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the strain values (%) with strain ratio = -1.0, corresponding to the &quot;life boundary&quot; values for each region of the S/N curve contained in array NBND( ) for each PHI draw consistent with the tensile point ( S_o ) (^8).</td>
</tr>
<tr>
<td>TRUNC</td>
<td>RE</td>
<td>Value used to filter out noise in the composite strain-time history during rainflow cycle counting. See [1], Section 2.2.1.4, for a discussion of rainflow cycle counting.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TSTART</td>
<td>RE</td>
<td>$T_s$ (°R) in Equation 3-3, the randomly selected gas temperature at the start of deceleration.</td>
</tr>
<tr>
<td>TSTMU</td>
<td>RE</td>
<td>Mean, $\mu$, of Normally distributed $T_s$, the gas temperature at the start of deceleration, given in Equation 3-3.</td>
</tr>
<tr>
<td>TSTSIG</td>
<td>RE</td>
<td>Standard deviation, $\sigma$, of Normally distributed $T_s$, gas temperature at the start of deceleration, given in Equation 3-3.</td>
</tr>
<tr>
<td>TSUBI</td>
<td>INT</td>
<td>The time index for the rotor time history for which the distribution on steady state rotor speed is valid.</td>
</tr>
<tr>
<td>VARPHI</td>
<td>INT</td>
<td>Controls type of material's intrinsic variation desired. A value of 1 indicates Weibull variation and a value of 2 indicates Lognormal variation.</td>
</tr>
<tr>
<td>VARY</td>
<td>INT</td>
<td>Controls type of S/N curve variation desired. A value of 0 indicates that no variation is required; a value of 1 means that intrinsic materials variation only is desired; a value of 2 indicates that the user desires a Uniform distribution on $m$; while a value of 3 indicates that a truncated Normal distribution is desired; a value of 4 indicates the user desires the bootstrapping option.</td>
</tr>
<tr>
<td>WEXP</td>
<td>RE</td>
<td>$w$ in Equation 3-8, the exponent for the Walker relation.</td>
</tr>
<tr>
<td>Z</td>
<td>RE</td>
<td>$Z$ in Equation 2-48 of [1], the randomly selected process variation shift factor given by a Lognormal(0, PVAR) random variate.</td>
</tr>
<tr>
<td>ZROREG</td>
<td>INT</td>
<td>Zero REGION, the variable permits the inclusion of the tensile point $S_o$. The value of 0 implies a DO loop from zero to NUMREG, while a value of 1 causes the DO loop to be executed from one to NUMREG.</td>
</tr>
</tbody>
</table>
1 The need for saving m’s is discussed in [1], Page 2-15.
2 See variable BLFPER( ) for a description of B-life.
3 The median S/N curve for the truncated Normal case is discussed in [1], Page 2-15.
4 See [1], Section 2.1.2.3, for a discussion on process variation in materials.
5 $m_*$ of the posterior density of $m$ is discussed in [1], Page 2-14.
6 The posterior credibility ranges $\pi(m)$ are discussed in [1], Page 2-13.
7 $\sigma_*$ of the posterior density of $m$ is discussed in [1], Page 2-14.
8 Extension of the S/N curve to the left using the tensile point is discussed in [1], Page 2-17.
### 7.2.4 Program BLDLCF Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program BLDLCF Listing Temporal Order, Uniform Distribution</td>
<td>7-98</td>
</tr>
<tr>
<td>Program BLDLCF Listing Temporal Order, Truncated Normal Distribution</td>
<td>7-100</td>
</tr>
<tr>
<td>BLDLCF</td>
<td>7-102</td>
</tr>
<tr>
<td>BLDLIF</td>
<td>7-111</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-114</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-115</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-115</td>
</tr>
<tr>
<td>GAM</td>
<td>7-116</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-116</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-122</td>
</tr>
<tr>
<td>INIT</td>
<td>7-122</td>
</tr>
<tr>
<td>RCE</td>
<td>7-123</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-130</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-131</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-134</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-137</td>
</tr>
<tr>
<td>GTVPAR</td>
<td>7-139</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-140</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-153</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-155</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-157</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-159</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-161</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-162</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-163</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-164</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-165</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-166</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-167</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-168</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-169</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-170</td>
</tr>
<tr>
<td>RAINF3</td>
<td>7-171</td>
</tr>
</tbody>
</table>

BLDLCF Version 3.4
Program BLDLCF Listing Temporal Order, Uniform Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDLCF</td>
<td>7-102</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-116</td>
</tr>
<tr>
<td>INIT</td>
<td>7-122</td>
</tr>
<tr>
<td>RCE</td>
<td>7-130</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-131</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-137</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-138</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-139</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-140</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-144</td>
</tr>
<tr>
<td>ADDRREG</td>
<td>7-145</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-146</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-147</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-158</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-158</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-158</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-158</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-158</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-158</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-158</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-162</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-162</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-163</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-163</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-163</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-163</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-163</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-163</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-163</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-163</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-166</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-166</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-166</td>
</tr>
<tr>
<td>GAM</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-166</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-166</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-166</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-166</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-166</td>
</tr>
<tr>
<td>BLDLIF</td>
<td>7-166</td>
</tr>
<tr>
<td>RAINF3</td>
<td>7-166</td>
</tr>
</tbody>
</table>

7 - 98
<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTLIFE</td>
<td>7-169</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-114</td>
</tr>
</tbody>
</table>
Program BLDLCF Listing Temporal Order, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDLCF</td>
<td>7-102</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-116</td>
</tr>
<tr>
<td>INIT</td>
<td>7-122</td>
</tr>
<tr>
<td>RCE</td>
<td>7-123</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-130</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-131</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-137</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-149</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-139</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-151</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-148</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-145</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-115</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-105</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-159</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-161</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-162</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-163</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-164</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-165</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-166</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-161</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-115</td>
</tr>
<tr>
<td>GAM</td>
<td>7-116</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-161</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-115</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-167</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-168</td>
</tr>
<tr>
<td>BLDLIF</td>
<td>7-111</td>
</tr>
<tr>
<td>RAINF3</td>
<td>7-171</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-169</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-114</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-170</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-147</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-162</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-163</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-164</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-165</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-166</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-168</td>
</tr>
</tbody>
</table>
**C**

C PROGRAM BLDLCF CONTROLS THE FLOW OF LOGIC OF THE LOW CYCLE
C FATIGUE ANALYSIS OF THE TURBINE BLADE FOIL PROBLEM
C
C PROGRAMMER : L. NEWLIN
C
C DATE: CODE: 7JAN92  COMMENTS: 3APR92
C
C VERSION: 3.4 (MATCHR V8.5, RAINF3 V1.1, INSORT V2.1)
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
C
C*****************************************************************************

C PROGRAM BLDLCF

C SUBPROGRAMS: INFAGG, PAREST, PRYRV, BETAGN, NORMGN, WEIBGN,
C TRMNAT, BLDLIF, INSORT, SORTM, EXPTCD
C
C 7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
C
NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

C IMPLICIT NONE

INTEGER MAXBLF, MAXDAT, MAXLIF, MAXM, MAXMM, MAXREG
C
PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLIF = 10000,
& MAXM = 50, MAXMM = 20001, MAXREG = 3)
C
COMMON IOUT
INTEGER BLFPOS(MAXBLF), I, IOUT, J, K, L, M, MCOUNT, MID,
& MPROC, NBLIFE, NHYPER, NLIFE, NLIFET, NMED,
& NPTS(MAXREG), NSYM, NTIME, NUMREG, TSUBI, VARPHI,
& VARY, ZROREG
C
DOUBLE PRECISION RAND
REAL ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1, BLDLIF,
& BLFPER(MAXBLF), BSERO, DUM, EBEND, EBENDA, EBENDB,
& ENNOM, ETHNOM(MAXM), FAP, FAB, FAC, FACTR, FAD, FAE,
& FAF, FAERR, FAERRS, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
& FD2A, FD2B, FD3A, FD3B, FDERR, FDERRS, FIFTY, FTU, FTY,
& HGAS, HGASA, HGASB, HGASR, HGASRI, HGASR2, HGAST,
& HGAST1, HGAST2, KRATIO, LAMA, LAMAA, LAMAB, LAMAD,
& LAMADA, LAMDB, LAM2, LAMGA, LAMGB, LAMKR, LAMGR,
& LAMGR1, LAMGT, LAMGT1, LAMGT2, LAMP, LAMPA, LAMPS,
& LAMTM, LAMTM2, LAMTM3, LIFE(MAXLIF), LNA(0:MAXREG),
& LNPHI, LNZ, LPHIM(0:MAXREG), MANAL, MANALA, MANALB,
& MEDM(MAXREG), MINPHI, MM(0:MAXREG), MODER1, MODER2,
& MU(MAXREG), NBND(0:MAXREG), NEWLIF, NF(MAXDAT, MAXREG),
& NMSG, PERIOD, PHI, PHISIG, PSIG, PVAR,
& RANGEM(2, MAXREG), RPM(MAXM), SBND(0:MAXREG),
& SIG(MAXREG), SLOPE, SLOPEA, SLOPEB, SLOPR, SLOPR1,
& SLOPR2, SLOPT, SLOPT1, SLOPT2, SPEED, SPEEDM, SPEEDS,
& STR(MAXDAT, MAXREG), ZERO, TANAL, TANALA, TANALB, TANGES,
& TGASA, TGASB, TGASR, TGASRI, TGASt, TGAST1,
& TGAST2, TRBIGK(0:MAXREG), TRSBND(0:MAXREG), TRUNC,
& & TSTART, TSIG, TSTSIG, WEKF, Z
C
C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

C

OPEN (1, FILE = 'BLDLCD', STATUS = 'OLD')
OPEN (3, FILE = 'BLDLCO', STATUS = 'NEW')
OPEN (7, FILE = 'DUMP', STATUS = 'NEW')
OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')

READ(1,*) RAND
WRITE(8,*) '
READ(1,*) IOUT
WRITE(8,*) 'IOUT (MATCHR = 10, BLDLCF = 15, RAINF3 = 20) = ', IOUT
READ(1,*) NLIPE
WRITE(8,*) '
READ(1,*) NHYPER
WRITE(8,*)'

C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

7 - 102
READ(1,*) NSYM
WRITE(8,*)' SYMMETRY NUMBER =',NSYM
READ(1,*) VARY
WRITE(8,*)' TYPE OF S/N VARIATION DESIRED =',VARY
READ(1,*) NMED
WRITE(8,*)' NORMAL MEDIAN CURVE (0 - NO, 1 - YES) =',NMED
READ(1,*) MPROC
WRITE(8,*)' MATERIALS PROCESS VARIATION DESIRED =',MPROC
READ(1,*) VARPHI
WRITE(8,*)' TYPE OF INTRINSIC VARIATION DESIRED =',VARPHI
IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
WRITE(8,*) 'ERROR: INVALID TYPE OF S/N VARIATION DESIRED'
CALL TRMNAT
ENDIF
IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
WRITE(8,*) 'ERROR: INVALID RESPONSE TO NORMAL MEDIAN CURVE QUESTION'
CALL TRMNAT
ENDIF
IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) THEN
WRITE(8,*) 'ERROR: INVALID TYPE OF MATERIALS PROCESS VARIATION DESIRED'
CALL TRMNAT
ENDIF
IF ((VARPHI .LT. 1) .OR. (VARPHI .GT. 2)) THEN
WRITE(8,*) 'ERROR: INVALID TYPE OF INTRINSIC VARIATION DESIRED'
CALL TRMNAT
ENDIF
READ(1,*) NBLIFE
IF (NBLIFE .GT. 0) READ(1,*) (BLFPER(J),J = 1,NBLIFE)
C ** READ DATA FROM BLDLC
READ(1,*) HGASA, HGASB, HGASR1, HGASR2, HGAST1, HGAST2,
& TGASA, TGASB, TGASR1, TGASR2, TGAST1, TGAST2,
& SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
& LAMGA, LAMGB, LAMGR1, LAMGR2, LAMGT1, LAMGT2,
& TSUBI, SPEEM, SPEEDS,
& FAERRM, FAERRS, TSTMU, TSTSIG,
& FDERRM, FDERRS,
& EBENDA, EBENDB, LAMPA, LAMPB,
& MANALA, MANALB, LAMAA, LAMAB,
& TANALA, TANALB, LAMAAA, LAMABB,
& LAMTMA, LAMTMB
READ(1,*) EMNM, NOSPAP, PERIOD, TRUNC, NTIME, WEXP
READ(1,*) FAA, FAB, FAC, FAD, FAE, FAF,
& FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
& FD2A, FD2B,
& FD3A, FD3B
IF (NTIME .GT. MAXM) THEN
WRITE(8,*) 'ERROR: STRAIN-TIME HISTORY TOO LARGE'
CALL TRMNAT
ENDIF
DO 20 I = 1, NTIME - 1
READ(1,*) RPM(I), ETHNOM(I)
20 CONTINUE
C ** ECHO DATA TO BLDLC
WRITE(3,900)
WRITE(3,901) HGASA, HGASB, HGASR1, HGASR2, HGAST1, HGAST2,
& TGASA, TGASB, TGASR1, TGASR2, TGAST1, TGAST2
WRITE(3,902) SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
& LAMGA, LAMGB, LAMGR1, LAMGR2, LAMGT1, LAMGT2
7 - 103
WRITE(3,903) TSUB1, SPEEDM, SPEEDS, FAERRM, FAERRS,
& TSTMU, TSTSIG, FDERRM, FDERRS,
WRITE(3,904) EBENDA, EBENDB, LAMPA, LAMPB, MANALA, MANALB,
& LAMAA, LAMAB, TANALA, TANALB
WRITE(3,905) EXP(LAMDA), EXP(LAMDB), EXP(LAMTA), EXP(LAMTB)
WRITE(3,906) EMOM, NORMPD, PERIOD, TRUNC, NTIME, WEXP
WRITE(3,907) FAA, FAB, FAC, FAD, FAE, FAF,
& FDIA, FDIB, FDIC, FDID, FDIE, FDIF,
& FD2A, FD2B,
& FD3A, FD3B
DO 25 I = 1, (NTIME - 1)
WRITE(3,908) RPM(I), ETHNOM(I)
25 CONTINUE
C ** CALL INFAGG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT
C OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, ZZERO, ZROREG, NUMREG,
& NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)
IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)
MCOUNT = 0
C ** INITIALIZE VARIABLES
DO 35 K = 1, MAXLIF
LIFE(K) = 1.0E+36
35 CONTINUE
NLIFET = NHYPER * NLIFE
C ** OUTER LOOP -- THIS LOOP SAMPLES HYPER-PARAMETER SETS
DO 150 K = 1, NHYPER
C ** CALL PRYRV TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS
CALL PRYRV (RAND, HGASRI, HGASR2, HGASTI, HGAST2, HGASR, HGAST)
CALL PRYRV (RAND, TGASRI, TGASR2, TGASTI, TGAST2, TGASR, TGAST)
CALL PRYRV (RAND, LAMGRI, LAMGR2, LAMGTI, LAMGT2, LAMGR, LAMGT)
CALL PRYRV (RAND, MANALA, MANALB, TANALA, TANALB, MANAL, TANAL)
C ** CALL PAREST TO PERFORM THE PARAMETER ESTIMATION ASPECT
C OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG,
& RAND, NBND, STR, BIGK, BZERO, MM, SBND)
PHISIG = 1.282550 / BZERO
C ** OBTAIN MATERIALS PROCESS VARIATION IF DESIRED
CALL NORMGN (RAND, 0.0, PSIG, LNZ)
IF (MPROC .EQ. 1) THEN
  Z = EXP (LNZ)
ELSE
  KRATIO = 1.0
  Z = 1.0
  LNZ = 0.0
ENDIF
MCOUNT = MCOUNT + 1
DO 175 L = 1, NUMREG
ALLM(MCOUNT, L) = MM(L)
175 CONTINUE
C ** INNER LOOP -- THIS LOOP GENERATES BLADE FAILURE TIMES
DO 200 I = 1, NLIFE
C ** INITILIZE S/N CURVE PARAMETERS

7 - 104
DO 225 L = 0, MAXREG
   LNA(L) = 0.0
   LPHIM(L) = 0.0
   TRSBND(L) = 0.0
225 CONTINUE

C ** SELECT DRIVERS FOR CALCULATING LIFE

  CALL BETAGN (RAND, HGASR, HGAST, HGASA, HGASB, HGAS)
  CALL BETAGN (RAND, TGASR, TGAST, TGASA, TGASB, TGAS)
  CALL BETAGN (RAND, SLOPR, SLOPT, SLOPEA, SLOPEB, SLOPE)
  CALL BETAGN (RAND, LAMGR, LAMGT, LAMGA, LAMGB, LAMG)

  CALL NORMGN (RAND, SPEEDM, SPEEDS, SPEED)
  CALL NORMGN (RAND, FAERRM, FAERRS, MODER1)
  CALL NORMGN (RAND, TSTMU, TSTSIG, TSTART)
  CALL NORMGN (RAND, FDERRM, FDERRS, MODER2)

  CALL PRYRV (RAND, EBENDA, EBENDB, LAMPA, LAMPB, EBEND, LAMP)
  CALL PRYRV (RAND, LAMAA, LAMAB, LAMAA, LAMAB, LAMA, DUM)
  CALL PRYRV (RAND, LAMDA, LAMDB, LAMTA, LAMTB, LAMDA, LAMTM)

  LAMDA = EXP (LAMDA)
  LAMTM = EXP (LAMTM)

  MINPHI = 1.0E+36

  IF (VARPHI .EQ. 1) THEN
    C WEIBULL INTRINSIC MATERIALS VARIATION
    DO 230 M = i, NSYM
      CALL WEIBGN (BZERO, RAND, PHI)
      MINPHI = MIN (PHI, MINPHI)
    230 CONTINUE
    PHI = MINPHI
  ELSE
    C LOGNORMAL INTRINSIC MATERIALS VARIATION
    DO 231 M = i, NSYM
      CALL NORMGN (RAND, 0.0, PHISIG, LNPHI)
      MINPHI = MIN (LNPHI, MINPHI)
    231 CONTINUE
    PHI = EXP (MINPHI)
  ENDIF

  IF (VARY .EQ. 0) PHI = 1.0

  IF (IOUT .EQ. 15) THEN
    WRITE(8,*) 'HGAS = ', HGAS,
                ' HGAS = ', HGAS
    WRITE(8,*) 'SLOPE = ', SLOPE,
                ' SLOPE = ', SLOPE
    WRITE(8,*) 'LAM = ', LAM,
                ' LAM = ', LAM
    WRITE(8,*) 'SPEED = ', SPEED,
                ' SPEED = ', SPEED
    WRITE(8,*) 'LAMDA = ', LAMDA
    WRITE(8,*) 'LAMTM = ', LAMTM,
                ' PHI = ', PHI
    WRITE(8,*) 'MANAL = ', MANAL,
                ' TANAL = ', TANAL
    WRITE(8,*) 'TSTART = ', TSTART,
                ' MODER1 = ', MODER1,
                ' MODER2 = ', MODER2
  ENDIF

C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS

  FACTR = PHI * KRATIO * 2

  DO 235 L = ZROREG, NUMREG
    TRSBND(L) = FACTR * SBND(L)
    TRBIGK(L) = BIGK(L)
  235 CONTINUE

  TRSBND(0) = SBND(0)

  IF (ZROREG .EQ. 0) CALL KOMO (SZERO, BIGK, MM, NBND,
                                TRSBND, TRBIGK, FACTR, NUMREG)

  DO 250 L = ZROREG, NUMREG
    LNA(L) = MM(L) * ALOG (TRBIGK(L))
    LPHIM(L) = MM(L) * ALOG (PHI)
  250 CONTINUE

  IF (IOUT .EQ. 15) THEN
    WRITE(8,*) 'L = ', L,
                ' MM = ', MM(L),
                ' BIGK = ', TRBIGK(L)
    WRITE(8,*) 'LNA = ', LNA(L),
                ' PHI = ', PHI
    WRITE(8,*) 'LPHIM = ', LPHIM(L),
                ' SBND = ', SBND(L)
  ENDIF
C ** CALL BDLIF TO OBTAIN BLADE LCF LIFE

NEWLIF = LAMDA * LAMTM * BDLIF (TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF, MODE1, RPM, TSB, SPEED, SLOPE, TSTART, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F, MODE2, FD2A, FD2B, FD2A, FD2B, ETHNOM, MANAL, LAMP, NOMSPD, EMNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, PERIOD, WEXP, MM, LMA, LPHIM, K, LNZ, TRSBND, ZROREG, NUMREG, SZERO)

IF (IOUT EQ. 15) WRITE(8,*) 'NEWLIF = ', NEWLIF

IF (NLIFET .GE. 100) CALL INSORT (NEWLIF, LIFE, NLIFET)

200 CONTINUE
150 CONTINUE

C ** PRINT SORTED LIVES TO FILE LOWLIF

DO 300 J = 1, (NLIFET / 100)
WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
300 CONTINUE

C ** INITIALIZE VARIABLE BLFPOS()

DO 325 J = 1, MAXBLF
BLFPOS(J) = 0
325 CONTINUE

FIFTY = 0.50E0

C ** PRINT EMPirical BLIVES

IF (VARPHI EQ. 1) THEN
WRITE(3,925)
ELSE
WRITE(3,927)
ENDIF

DO 350 J = 1, NBLIFE
BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
WRITE(3,926) BLFPER(J), LIFE(BLFPOS(J))
350 CONTINUE
WRITE(3,926) FIFTY, LIFE(NLIFET/2)

ENDIF

C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED

IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) THEN
CALL SORTM (ALLM, NUMREG, MCOUNT)
MID = MCOUNT / 2
DO 400 L = 1, NUMREG
MEDM(L) = ALLM(MID,L)
400 CONTINUE
CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND, BIGKI, BZERO)
ENDIF

C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO BDLCO
INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO, THETA,

Hgas, Be(', F5.0,, F6.0,), U(', F7.5,, F8.5,), Tgas (deg R),

be(', F5.0,, F6.0,), U(', F7.5,, F8.5,), TgasUncert.,

DECEL SLOPE, Be(', F5.0,, F6.0,), U(', F7.5,, F8.5,),

TgasUncert.,

DECEL MODELING ERROR, N(', F4.1,, F5.1,),

STAR, TING DEC, TEMPERATURE (deg R),

Fdecel MODELING ERROR, N(', F4.1,, F5.1,),

STRAIN DUE TO GAS BENDING (%),

LAMBDA BLADEULL, Mechanical Analysis Factor,

COEFFICIENT OF THERMAL EXPANSION FACTOR,

THERMAL ANALYSIS FACTOR,

DAMAGE MODEL ACCURACY,

TMFMODEL ACCURACY,

OTHER STRAIN HISTORY INPUT,

COEFFICIENTS,

WEIBULL VARIATION,

B LIVES: EMPIRICAL,

LOGNORMAL VARIATION,
STOP
END

C**************************************************************
C SAMPLE 'BLDLCD' INPUT FILE
C**************************************************************
C 675 .................................. RANDOM NUMBER SEED
C 0 .................................. OUTPUT DUMP CONTROLLER
C 100 ................................ INNER LOOP SIZE
C 200 ................................ OUTER LOOP SIZE
C 50 .................................. SYMMETRY NUMBER
C 0 .................................. UNIFORM S/N VARIATION
C 0 .................................. MAT. PROC. VAR. NOT REQUIRED
C 0 .................................. WEIBULL INTRINSIC VARIATION
C 0 .................................. NUMBER OF LIVES REQUESTED
C 0.0001 ................................ B.01 LIFE
C 0.001 ................................ B.1 LIFE
C 0.01 .................................. B1 LIFE
C 0.0004 ................................ B.01 LIFE
C 0.001 ................................ B.1 LIFE
C 0.0004 ................................ B1 LIFE
C 0.00727362 0.000067442 -0.000059109 -3.52929E-08 1.07611E-08 -2.74419E-08
C Fdecel Modeling Error Mean & std. dev.
C 1640.0 40.67 .................................. DECEL Tstart MEAN & STANDARD DEVIATION
C 975.3 28.6 ................................ STANDARD RESPONSE PROBE MEAN & STD DEV
C 0.0 0.03 .................................. Fdecel Modeling Error Mean & std. dev
C 0.0 0.0 .................................. STRAIN DUE TO GAS BENDING (%)
C 0.96 1.02 .................................. MECHANICAL ANALYSIS ACCURACY FACTOR
C 0.80 1.20 .................................. THERMAL ANALYSIS ACCURACY FACTOR
C -0.693147 0.563283 .................. DAMAGE ACCUMULATION MODEL ACCURACY
C 0.00 0.00 .................................. NMP MODEL ACCURACY
C 0.295 38482.0 .......................... NOMINAL MECH. STRAIN & ROTOR SPEED (% RPM)
C 1.0 .................................. STRAIN-TIME HISTORY PERIOD (MISSIONS)
C 0.000 .................................. STRAIN-TIME HISTORY NOISE FILTER (%)
C 6 .................................. NUMBER OF POINTS IN STRAIN-TIME HISTORY
C 0.5 .................................. WALKER EXPONENT

COEFFICIENTS FOR STARTUP RESPONSE SURFACE FOR THERMAL STRAIN:
Fdecel (Tgas,Hgas) = A + B * T + C * H + D * T**2 + E * H**2 + F * T * H
C A B C D E F
C 0.00727362 0.000067442 -0.000059109 -3.52929E-08 1.07611E-08 -2.74419E-08
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR THERMAL STRAIN:
Fdecel1 (m,Tstart) = A + B * Tstart + C * m + D * Tstart ** 2 + E * m ** 2 + F * Tstart * m
C A B C D E F
C -0.132623 0.000227427 -0.000059290 0.00 0.00 4.71714E-08
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR RFM:
Fdecel2 (m,Tstart) = A + (Tstart - B) / m
C A B
C 30523.07 -21846.15
C COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR RPM:
Fdecel3 (t) = A + B * t
C A B
C 30523.07 -21846.15
C RPM(TIME) THERMAL STRAIN (%) ........... STRAIN HISTORY INFORMATION
C 225.8 0.0
C 3025.1 -0.196921
C 6138.8 0.146025
C 8539.0 -0.200128
C 0.0 0.007393

7 - 108
C SPECIFIC MATERIAL DESCRIPTION
C YIELD & ULTIMATE STRENGTHS, NDIV, NPTS
C # PTS IN DIV, STRAIN RATIO, REGION
C 0.89 6800 S(1) N(1) RAW
C 0.89 15000 S(2) N(2) STRAIN-LIFE
C 0.87 27000 S(3) N(3)
C 0.67 43200 S(4) N(4) DATA
C 0.56 139300 S(5) N(5) POINTS
C 0.56 147000 S(6) N(6)
C 0.39 434480 S(7) N(7)
C 0.00 434480 S(8) N(8) MATERIAL
C 1.00 NO VALUE OF SC SUPPLIED (%)
C 1.00+36 NUMBER OF REGIONS/W/DATA W/O DATA
C 0.00 CONSTRAINT ON COEFF. OF VARIATION
C 0.0 0.0 0.0 0.0 0.0 0.0 LIFE BOUNDARIES: REGION 1
C 0.00 0.00 0.00 0.00 0.00 0.00 NORMAL DIST. PRIORS: DELTA, NO, SIGMA2
C*********************************************************************
C*********************************************************************
C LIST OF VARIABLES
C*********************************************************************
C 2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
C EQUAL TO BIGK(1) -- DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND LCF
C 1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C 1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPIRICAL BLIVES
C VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING S/N DATA SET
C DUMMY VARIABLE
C SELECTED VALUE FOR BENDING STRAIN (%)
C SELECTED LOWE BOUND
C SELECTED UPPER BOUND
C NOMINAL MECHANICAL STRAIN (%)
C I-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
C COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C SCALE FACTOR EQUAL TO PHI * KRATIO * Z
C STARTUP THERMAL STRAIN RESPONSE SURFACE MEAN
C STARTUP THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C COEFFICIENTS FOR BD1, ONE OF THE DECELERATION FUNCTIONS
C COEFFICIENTS FOR BD2, ONE OF THE DECELERATION FUNCTIONS
C COEFFICIENTS FOR BD3, ONE OF THE DECELERATION FUNCTIONS
C DECELERATION THERMAL STRAIN RESPONSE SURFACE MEAN
C DECELERATION THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C EQUAL TO .5 -- USED TO ACCESS 50% POINT IN LIFE()
C MATERIAL ULTIMATE STRENGTH (%)
C MATERIAL YIELD STRENGTH (%)
C SELECTED HOT GAS FILM COEFFICIENT, Hgas
C Hgas LOWER BOUND
C Hgas UPPER BOUND
C SELECTED RHO FOR Hgas
C - RHO LOWER BOUND
C - RHO UPPER BOUND
C SELECTED THETA FOR Hgas
C - THETA LOWER BOUND
C - THETA UPPER BOUND
C CONTROLS INNER DO LOOP
C CONTROLS DUMP TO FILE IOUTPR
C CONTROLS DO LOOP FOR EACH BLIFE
C CONTROLS OUTER DO LOOP
C RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C CONTROLS DO LOOP FOR EACH REGION
C SELECTED COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR,
C LAMbda Alpha
C LAMAA LAMAA LOWER BOUND
C LAMAB LAMAB UPPER BOUND
C SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, LAMbda

7 - 109
Damage Accumulation

- LAMDA = LAMDA LOWER BOUND
- LAMDB = LAMDA UPPER BOUND
- LAMG = SELECTED UNCERTAINTY IN Tgas
- LAMGA = LAMG LOWER BOUND
- LAMGB = LAMG UPPER BOUND
- LAMG1 = LAM - RHO LOWER BOUND
- LAMG2 = LAM - RHO UPPER BOUND
- LAMGT = SELECTED THETA FOR LAM
- LAMGT1 = LAM - THETA LOWER BOUND
- LAMGT2 = LAM - THETA UPPER BOUND
- LAMP = SELECTED DEVIATION IN BLADE PULL DUE TO BLADE MASS, LAMda Pull
- LAMPA = LAMP LOWER BOUND
- LAMPB = LAMP UPPER BOUND
- LAMTM = SELECTED TMF MODEL ACCURACY FACTOR, Lamda TMf
- LAMTM = LAMTM LOWER BOUND
- LAMTMB = LAMTM UPPER BOUND
- LIFE() = 1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
- LNA() = 1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
- LNPHI = LOGNORMAL(0,PHISIG**2) GENERATED RANDOM VARIABLE
- LN2 = NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
- LPHTIM() = 1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
- M = CONTROLS SYMMETRY DO LOOP
- MANAL = SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
- MANALA = MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND
- MANALB = MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR UPPER BOUND
- MAXBLF = MAXIMUM NUMBER OF BLIVES TO BE PROVIDED
- MAXDAT = MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED
- MAXLIF = MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA,
  ALPHA CALCULATION
- MAXM = MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY
- MAXMM = MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
- MCOUNT = NUMBER OF M's TO BE PROVIDED FOR THE TRUNCATED NORMAL
- MEDM() = 1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
- MID = POINTER TO THE MEDIAN M VALUES -- EQUAL TO HALF OF MCOUNT
- MINTPHI = EQUAL TO MIN(PHI) -- THE MINIMUM OF NSYM DRAWS OF PHI
- MM() = 1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
- MODE1 = MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
- MODE2 = MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
- MPROC = Materials PROCess variation -- CONTROLS MATERIALS PROCESS
- MED() = 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
- DBLIFE = NUMBER OF BLIVES TO BE PROVIDED
- NBND() = 2-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE
- REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE) MATERIAL
- S/N DATA SET
- NEWLIF = LIFE VALUE RETURNED FROM CALL TO BLDLIF
- NF() = 2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL
- S/N DATA SET BROKEN INTO LIFE REGIONS
- NHYPER = SIZE OF OUTER LOOP
- NLIFE = SIZE OF INNER LOOP
- NLIFE = TOTAL NUMBER OF LIVES CALCULATED BY PFM
- NMED = CONTROLS MEDIAN CALCULATION FOR THE TRUNCATED NORMAL
- NED() = DISTRIBUTION CASE -- 0 - NO MEDIAN CALCULATION;
  1 - MEDIAN CALCULATION DESIRED
- NOMSPD = NOMINAL ROTOR SPEED, RPM
- NPTS() = 1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION
- FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
- NSYM = SYMMETRY NUMBER
- NTIME = NUMBER OF POINTS IN STRAIN-TIME HISTORY
- NUMREG = NUMBER OF REGIONS OF INTEREST
- PERIOD = LENGTH OF TIME IN MISSIONS OF TIME HISTORY
- PHI = WEIBULL(BETA, ETAO) GENERATED RANDOM VARIABLE
- PHISIG = EQN TO PT * (S**2)*PVAR / BZERO -- VALUE OF LOGNORMAL
- PSIG = EQUAL TO SQRT(PVAR) -- MATERIALS PROCESS STANDARD DEVIATION
- PVAR = MATERIALS PROCESS VARIATION
- RANGEM() = 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON M FOR
EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

RPM() 1-D ARRAY CONTAINING ROTOR SPEED HISTORY (RPM)

SBND() 1-D ARRAY CONTAINING THE STRAIN VALUES (% R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()

SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

SLOPE SELECTED DECELERATION SLOPE, m (deg R / sec)

SLOPEA m LOWER BOUND

SLOPEB m UPPER BOUND

SLOPR SELECTED RHO FOR m

SLOPR1 m - RHO LOWER BOUND

SLOPR2 m - RHO UPPER BOUND

SLOPT SELECTED THETA FOR m

SLOPT1 m - THETA LOWER BOUND

SLOPT2 m - THETA UPPER BOUND

SPEED SELECTED STEADY STATE ROTOR SPEED, RPM

SPEEDM MEAN OF ROTOR SPEED (MU, NORMAL DISTRIBUTION)

SPEEDS STANDARD DEVIATION OF ROTOR SPEED (SIGMA, NORMAL DISTRIBUTION)

STR() 2-D ARRAY CONTAINING STRAIN POINTS (STRAIN RATIO = -1.0) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS

SZERO STRAIN TENSILE TEST POINT, 50

TANAL SELECTED THERMAL STRAIN ANALYSIS ACCURACY FACTOR

TANALB THERMAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND

TGAS SELECTED GAS TEMPERATURE Tgas

TGASA GAS TEMPERATURE LOWER BOUND

TGASB GAS TEMPERATURE UPPER BOUND

TGASR SELECTED RHO FOR GAS TEMPERATURE

TGASR1 GAS TEMPERATURE - RHO LOWER BOUND

TGASR2 GAS TEMPERATURE - RHO UPPER BOUND

TGAST SELECTED THETA FOR GAS TEMPERATURE

TGAST1 GAS TEMPERATURE - THETA LOWER BOUND

TGAST2 GAS TEMPERATURE - THETA UPPER BOUND

TRBIGK() 1-D ARRAY CONTAINING VALUES OF BIGK() CORRECTED FOR SZERO, PHI, KRATIO, AND Z

TRSSND() 1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR EACH REGION CALCULATED FOR EACH TRIAL

TRUNC VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)

TSTART STARTING DECELERATION TEMPERATURE (deg R)

TSTMU MEAN OF TSTART

TSTSIG STANDARD DEVIATION OF TSTART

VARPHI THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS

VARY CONTROLS TYPE OF INTRINSIC MATERIALS VARIATION DESIRED --

1 - WEIBULL VARIATION;

2 - LOGNORMAL VARIATION

C VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;

1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION

WEXP WALKER EXPONENT

Z LOGNORMAL(0, PVAR) GENERATED RANDOM VARIATE

ZROREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP

BEGINNING VALUE = 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

FUNCTION BLDLIF PERFORMS THE DRIVER TRANSFORMATION AND CALLS RAINF3 TO CALCULATE THE FATIGUE LIFE

PROGRAMMER: L. NEWLIN

DATE: CODE: 7JAN92 COMMENTS: 3APR92

VERSION: BLDLCF 3.4 (MATCHR V8.5, RAINF3 V1.1)

Copyright (C) 1990, California Institute of Technology.

U.S. Government Sponsorship under NASA Contract NAS7-918

is acknowledged.

FUNCTION BLDLIF (TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF, & MODER1, RPM, TSUBI, SPEED, SLOPE, TSTART, FD1A, & FD1B, FD1C, FD1D, FD1E, FD1F, MODER2, FD2A, & FD2B, FD3A, FD3B. ETHNOM, MANAL, LAMG, EBEND, NTIME, TRUNC, & ETHNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, & ETHNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, 7 - 111
LIST OF VARIABLES

EBEND SELEC TED VALUE FOR BENDING STRAIN (%) 
EM() 1-D ARRAY CONTAINING THE SIMULATED MECHANICAL STRAIN-TIME HISTORY (%) 
EMNOM NOMINAL MECHANICAL STRAIN (%) 
ETH() 1-D ARRAY CONTAINING THE SIMULATED THERMAL STRAIN-TIME HISTORY 
ETHNOM() 1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY 
FA VALUE OF ACCELERATION FUNCTION FOR THERMAL STRAIN — SECOND ORDER POLYNOMIAL AS A FUNCTION OF TGAS AND HGAS 
FAA, FAB, FAC, FAD, FAE, FAF COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION 
FD1 VALUE OF DECELERATION FUNCTION FOR THERMAL STRAIN — SECOND ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE 
FD1A, FD1B, FD1D, FD1E, FD1F COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS 
FD2 VALUE OF DECELERATION FUNCTION FOR TIME — SECOND ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE 
FD2A, FD2B COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS 
FD3 VALUE OF DECELERATION FUNCTION FOR ROTOR SPEED — FIRST ORDER POLYNOMIAL (LINEAR) FUNCTION OF TIME 
FD3A, FD3B COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS 
HGAS SELECTED HOT GAS FILM COEFFICIENT, Hgas 
I CONTROLS DO LOOP FOR EACH POINT IN TIME HISTORY 
IOUT CONTROLS DUMP TO FILE IOUTPR 
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS 
LAMA SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR, LAMda Alpha 
LAMP THE UNCERTAINTY IN Tgas 
LAMP SELECTED VALUE FOR DEVIATION IN BLADE PULL DUE TO BLADE MASS, LAMda Pull 
LNA() 1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION 
LNA(0:MAXREG) I-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION 
LPHIM() 1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION 
MANAL SELECTED VALUE FOR MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR 
MAXM MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY 
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED 
MM() 1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION 
M ODERR MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
**MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE**

**NOMSPD**  NOMINAL ROTOR SPEED, RPM

**NTIME**  NUMBER OF POINTS IN STRAIN-TIME HISTORY

**NUMREG**  NUMBER OF REGIONS OF INTEREST

**PERIOD**  LENGTH OF TIME IN MISSIONS OF TIME HISTORY

**RAINF3**  REAL FUNCTION PERFORMING RAINFLOW COUNTING, DAMAGE ACCUMULATION AND FATIGUE LIFE PREDICTION (USING THE MATERIALS CHARACTERIZATION MODEL)

**RPM()**  1-D ARRAY CONTAINING ROTOR SPEED HISTORY

**SLOPE**  SELECTED VALUE FOR DECELERATION SLOPE, deg R/sec

**SPEED**  SELECTED VALUE FOR STEADY STATE ROTOR SPEED, rpm

**SZERO**  STRAIN TENSILE TEST POINT, So

**TANAL**  SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR

**TGAS**  SELECTED VALUE FOR HOT GAS TEMPERATURE Tgas (deg R)

**TRSBND()**  1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR EACH REGION CALCULATED FOR EACH TRIAL

**TRUNC**  VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)

**TSUBI**  THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS

**WEXP**  WALKER EXPONENT

**ZROREG**  ZERO REGION -- VALUES CHosen TO FACILITATE REGION DO LOOP

**ZROREGION**  BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

**C**  **CALCULATE STRAIN HISTORY**

```
    FA = FAA + FAB * TGAS + FAC * HGAS + FAD * TGAS ** 2 &
    + FAE * HGAS ** 2 + FAF * TGAS + HGAS + MODER1
    ETHNOM(1) = FA
    RPM(TSUBI) = SPEED
    FD1 = FD1A + FD1B * TSTART + FD1C * SLOPE + FD1D * TSTART ** 2 &
    + FD1E * SLOPE ** 2 + FD1F * TSTART * SLOPE + MODER2
    FD2 = FD2A + (TSTART - FD2B) / SLOPE
    FD3 = FD3A + FD3B * FD2
    RPM(NTIME) = FD3
    ETHNOM(NTIME) = FD1
    DO 100 I = 1, NTIME
      EM(I) = MANAL * LAMP * (RPM(I) / NOMSPD) ** 2 * EMNOM
      ETH(I) = TANAL * LAMA * ETHNOM(I)
      IF ((I .GT. 1) .AND. (I .LT. T_SUBI)) &
      ETH(I) = LAMG * ETH(I)
      ETOT(I) = EBEND + EM(I) + ETH(I)
  100 CONTINUE
    IF (IOUT .EQ. 15) THEN
      WRITE(8,*), 'FA = ', FA, ' ETHNOM1 = ', ETHNOM(1)
      WRITE(8,*), 'RPMI = ', RPM(TSUBI), ' LAMG = ', LAMG
      WRITE(8,*), 'FD1 = ', FD1, ' FD2 = ', FD2
      WRITE(8,*), 'FD3 = ', FD3
      WRITE(8,*), 'ETH(I) = ', ETH(I), ' ETOT(I) = ', ETOT(I)
    ENDIF
```

**CALL RAINF3 TO CALCULATE DAMAGE AND RESULTING FATIGUE LIFE**

```
    BDLIF = RAINF3 (ETOT, NTIME, TRUNC, PERIOD, WEXP, MM, LNA, &
    LPHIM, KRATIO, LNZ, TRSBND, ZROREG, NUMREG, SZERO)
    RETURN
END
```

C**************************
SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED

PROGRAMMER: L. NEWLIN

DATE: 20JUL90

VERSION: 2.1

Copyright (C) 1990, California Institute of Technology.

U.S. Government Sponsorship under NASA Contract NAS7-918

is acknowledged.

SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C INPUTS: NEWLIF, LIFE, NLIFET

C OUTPUTS: LIFE

C IMPLICIT NONE

INTEGER MAXLIF

PARAMETER (MAXLIF = 10000)

COMMON IOUT

INTEGER I, IOUT, NLIFET, NUM, PLACE

REAL LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

LIST OF VARIABLES

I CONTROLS DO LOOP FOR INSERTION

IOUT OUTPUT DUMP CONTROLLER

LIFE() 1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE

PFM TO BE SORTED

MAXLIF MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,

CALCULATION

NEWLIF LIFE VALUE TO BE INSERTED INTO LIFE()

NLIFET TOTAL NUMBER OF LIVES CALCULATED BY PFM

NUM NUMBER OF LIFE VALUES IN LIFE()

PLACE POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()

TEMP() 1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON

INSERTION OF NEWLIF

NUM = NLIFET / 2

C FIND POSITION IN LIFE() FOR NEWLIF

IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

DO 100 I = 1, NUM

IF (NEWLIF .LT. LIFE(I)) THEN

PLACE = I

GOTO 110

ENDIF

100 CONTINUE

110 CONTINUE

C STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

DO 200 I = (PLACE + 1), NUM

TEMP(I) = LIFE(I - 1)

200 CONTINUE

C INSERT NEWLIF

LIFE(PLACE) = NEWLIF

C SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

DO 300 I = (PLACE + 1), NUM

LIFE(I) = TEMP(I)

300 CONTINUE
C IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN
400 CONTINUE

RETURN
END

C*******************************************************************************
C SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C INDEPENDENT RANDOM VARIATES
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: RANDOM
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
C*******************************************************************************

SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)
COMMON IOUT
DOUBLE PRECISION RAND
REAL FRAC, RHO1, RHO2, THE1, THE2, X, Y
INTEGER IOUT
CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE(8,*) FRAC =', FRAC
X = FRAC * (RHO2 - RHO1) + RHO1
CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE(8,*) FRAC =', FRAC
Y = FRAC * (THE2 - THE1) + THE1
RETURN
END

C*******************************************************************************
C THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: GAM
C The random variates are generated using the method described in:
C Univariate Distributions - 1, Houghton Mifflin Company, 1970,
C pp. 181-182.
C*******************************************************************************

SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)
COMMON IOUT
DOUBLE PRECISION RAND
REAL A, B, GAM, RHO, THETA, W, X, Y1, Y2
INTEGER IOUT
Y1 = GAM((RHO * THETA + 1.), RAND)
Y2 = GAM(((1. - RHO) * THETA + 1.), RAND)
W = Y1 / (Y1 + Y2)

IF (IOUT .EQ. 15) WRITE(8,*), 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

X = W * (B - A) + A

IF (IOUT .EQ. 15) WRITE(8,*), 'W =', W, ' X =', X

RETURN

END

C*******************************************************************************

C The random variates are generated using an "Acceptance/Rejection Method"
C Fishman, George S., "Sampling From the Gamma Distribution on a
C Computer." Communications of the ACM, Volume 19, Number 7, July 1976,

REAL FUNCTION GAM (ALPHA, RAND)

C SUBPROGRAM: RANDOM

COMMON IOUT
INTEGER IOUT
REAL A, ALPHA, ARG, U1, U2, V1, V2
DOUBLE PRECISION RAND

A = ALPHA - 1.

IF (IOUT .EQ. 15) WRITE(8,*), 'A =', A, ' ALPHA =', ALPHA

CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALOG(U1)
V2 = - ALOG(U2)

IF (IOUT .EQ. 15) WRITE(8,*), 'U1 =', U1, ' U2 =', U2, ' V1 =', V1, ' V2 =', V2

ARG = A * (V1 - ALOG(V1)) - 1.

IF (V2 .LT. ARG) GOTO 10

GAM = ALPHA * V1

IF (IOUT .EQ. 15) WRITE(8,*), 'GAMMA =', GAM

RETURN

END

C*******************************************************************************

C SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C FOR THE STRESS FORMULATION
C PROGRAMMER: L. NEWLIN
C DATE: 13JUL89
C FORMAT/COMMENTS: 12AUG91
C VERSION: MATCHR V8.4, V8.5 MATGRM V4.4, V4.5
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
& NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
& KRATIO, PVAR)

C INPUTS: READS DATA FROM SPECIFIED AND RELATED; VARY, MPROC
C OUTPUTS: RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG,
& NBND, STR, FTUZ, FTYZ, VARY, MPROC
C SUBPROGRAMS: INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDNFX, ADDREG,
& CONCAV, MEDIAN, EXPCTD, MUSIG, Norris, ADDRGN, GTPVAR

7 - 116
LIST OF VARIABLES

BIGKHT  EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
BZERO  VALUE OF WEIBULL PARAMETER, BETA0, CHARACTERIZING THE S/N DATA SET
CZERO  EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
       COEFFICIENT OF VARIATION, CO
DD()  1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
DELTAMU  1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
       AND SIG() CALCULATION
FTUZ  ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTYZ  YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
IOUT  OUTPUT DUMP CONTROLLER
IZERO()  2-D ARRAY CONTAINING IO, THE 95% CONFIDENCE INTERVALS ON C
         FOR EACH REGION
JZERO()  2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
         FOR EACH REGION
KRATIO  RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
LAMN  CONTROLS DO LOOP FOR EACH REGION
LNSTR()  3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
       AND MATERIALS DATA SET (PER REGION) ALLOWED
MAXDAT  MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION)
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET  MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MC()  2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
       BASED ON MATERIALS DATA ONLY — MC(1,L) IS THE LOWER BOUND AND Mc(2,L) IS THE UPPER
       BOUND
MCHAT()  2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
       FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
       MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
       MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT()  1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
         MC() FOR EACH REGION
MEDM()  1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
MO()  1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
       MEAN FOR EACH REGION
MPNT()  1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
         MZERO() FOR EACH REGION
MPROC  MATERIALS PROCESS VARIATION — CONTROLS MATERIALS PROCESS
       VARIATION — 0 — NO VARIATION; 1 — VARIATION
MU()  1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
       DISTRIBUTION MEAN FOR EACH REGION
MZERO()  2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
         EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
OPEN(6, FILE = 'RELATO', STATUS = 'NEW')

C RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET INFORMATION
C PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION
C INITIALIZE PRIMARY ARRAYS
   CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
              NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)
C READ, CONVERT, ECHO INFORMATION
   CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR,
             & LF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NODAT,
             & NSETS, NBND, CZERO, MPNT, MZERO, PFR, ZROREG, DELTA, MO,
             & SIGMA2, KFRAZIO, LAMN)
C CALCULATE RESIDUAL VARIANCES
   CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
                 & SWHAT2, SUHAT2, NPPR)
CALL FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN
C CALCULATIONS FOR ALL TYPES OF VARIATION SAVE NORMAL
C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS
& CALL INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO, MCHAT)
C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
IF (MPROC .EQ. 1) THEN
CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
ENDIF
C COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C OBTAIN POSTERIOR RANGES ON M
& CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT, RANGEM)
C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
CALL ADDRNG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
CALL CONCAV (NUMREG, RANGEM)
C WRITE RESULTS TO FILE DUMP
WRITE(7,900)
DO 25 L = 1, NUMREG
WRITE(7,905) L, IZERO(I, L), IZERO(2, L), JZERO(I, L), JZERO(2, L)
25 CONTINUE
WRITE(7,910)
DO 50 L = 1, NUMREG
WRITE(7,915) L, MCHAT(2,L), MCHAT(I,L)
50 CONTINUE
IF (CZERO .GT. 0.0) THEN
WRITE(7,960)
DO 150 L = 1, NUMREG
IF (MCPNT(L) .EQ. 1) THEN
WRITE(7,965) L, MC(I,L)
ELSEIF (MCPNT(L) .EQ. 2) THEN
WRITE(7,970) L, MC(I,L), MC(2,L)
ENDIF
150 CONTINUE
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 100 L = 1, NUMREG
WRITE(7,940) L, RANGEM(I,L), RANGEM(2,L)
100 CONTINUE
WRITE(7,950)
C CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO
CALL MEDIAN (NUMREG, RANGEM, MEDM)
C CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
CALL EXPCTD (1, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG, NBND, BIGN1, BZERO)

C CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED
IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
   DO 200 L = 1, NUMREG
      RANGEM(1,L) = MEDM(L)
      RANGEM(2,L) = MEDM(L)
   CONTINUE
ENDIF
ELSE
C NORMAL VARIATION IS DESIRED
C CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION
   CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO,
& SIGMA2, MCHAT, MU, SIG)
C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
   IF (MPROC .EQ. 1) THEN
      CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
   ENDIF
C COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M
   CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
   CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO,
& MPNT, MO, SIGMA2)
C ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
   CALL CONCAV (NUMREG, RANGEM)
C WRITE RESULTS TO FILE DUMP
   WRITE(7,975)
   DO 350 L = 1, NUMREG
      WRITE(7,980) L, MCHAT(I,L)
   CONTINUE
   IF (CZERO .GT. 0.0) THEN
      WRITE(7,960)
      DO 360 L = 1, NUMREG
         IF (MCPNT(L) .EQ. 1) THEN
            WRITE(7,965) L, MC(I,L)
         ELSEIF (MCPNT(L) .EQ. 2) THEN
            WRITE(7,970) L, MC(I,L), MC(2,L)
         ENDIF
      CONTINUE
   ENDIF
   WRITE(7,920)
   WRITE(7,930)
   DO 370 L = 1, NUMREG
      WRITE(7,940) L, RANGEM(I,L), RANGEM(2,L)
   CONTINUE
   WRITE(7,950)
   WRITE(7,985)
   DO 380 L = 1, NUMREG
      WRITE(7,990) L, MU(L), SIG(L)
   CONTINUE
ENDIF

7 - 120
C PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS

IF (MPROC .EQ. 1) THEN
WRITE(?,995) PVAR
ENDIF

C FORMAT STATEMENTS

900 FORMAT(2X,'Copyright (C) 1990, California Institute of ', &
' Technology. U.S. Government',/2X,'Sponsorship under ', &
' NASA Contract NAS7-918 is acknowledged.'/2X, &
' RESULTS OF INFORMATION AGGREGATION CALCULATIONS', &
' FOR EACH REGION'/)

905 FORMAT(7X,'REGION: ',11X,'To = ('',F12.9,')', &
',/24X,'Jo = ('',F12.9,')',/)

910 FORMAT(/,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION', &
' ESTIMATE OF m FOR EACH REGION',/)

915 FORMAT(//,2X,'REGION',8X,'E(C)',12X,'E(m)',/)

920 FORMAT(/,///),2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH REGION'/)

930 FORMAT(/,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

940 FORMAT(6X,11X,8X,F8.4,8X,F8.4)

950 FORMAT(/,///),2X,'RANGE ON m FOR EACH REGION IMPLIED BY C CONSTRAINT', &
' RANGE ON m FOR EACH REGION IMPLIED BY C CONSTRAINT', &
',/2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

960 FORMAT(//,///),2X,'REGION',12X,'E(m)',/)

965 FORMAT(6X,11X,8X,F8.4,8X,'INFINITY')

970 FORMAT(6X,11X,8X,F8.4,8X,F8.4)

975 FORMAT(2X,'Copyright (C) 1990, California Institute of ', &
' Technology. U.S. Government',/2X,'Sponsorship under ', &
' NASA Contract NAS7-918 is acknowledged.'/2X, &
' RESULTS OF INFORMATION AGGREGATION CALCULATIONS', &
' ESTIMATE OF m FOR EACH REGION', &
',/2X,'REGION',12X,'E(m)',/)

980 FORMAT(9X,11X,F10.6)

985 FORMAT(2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS', &
'/,2X,'REGION',5X,'MEAN',8X,'STD DEV',/)

990 FORMAT(5X,11X,F7.4,5X,E11.5)

995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ', &
' MEDIAN S/N CURVE',/2X,'WARRANTED BY THE AVAILABLE ', &
' INFORMATION',/7X,E11.5)

RETURN
END

C******************************************************************************************

C SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C PROGRAMMER: L. NEWLIN
C DATE: 5OCT87
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
SUBROUTINE TRMNAT
WRITE (8,*) 'PROGRAM EXECUTION TERMINATED'
STOP
END

C*******************************************************************************
C SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88   COMMENTS: 13JUL89
C SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR,
& RFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)
C INPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, RFNP,
C NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2
C IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
& NPTS(0:MAXSET), REFNP(MAXREG)
REAL
& DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
& LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
& MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
& RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
& RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
& STR(MAXDAT, MAXREG)
LIST OF VARIABLES
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
SIG() CALCULATION
I OUT CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
L CONTROLS DO LOOP FOR EACH REGION
LNNF() 3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C EACH REGION – MZERO(1, L) IS THE LOWER BOUND AND MZERO(2, L)
IS THE UPPER BOUND
NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
IN EACH REGION
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
RATSTR() 2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS

7 - 122
SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, &
  LINSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,

C************************************************************

C SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATD; "CONVERTS"
C THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C SPECFO AND RELATO. RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C SPECIFIED BY USER
C SPECIFIED BY L. NEWLIN
C DATE: 21JUN88 FORMAT/COMMENTS: 12AUG91

SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, &
  LINSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
LIST OF VARIABLES

INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO, CONVERSION, AND BREAK UP

EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, Co

1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION

1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET

ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET

ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL

YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL

YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL

CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET

CONTROLS DO LOOP FOR EACH DATA DIVISION

MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED

MAXIMUM NUMBER OF REGIONS ALLOWED

MAXIMUM NUMBER OF S/N DATA SETS ALLOWED

1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION

1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION

MATERIALS PROCESS VARIATION — CONTROLS MATERIALS PROCESS VARIATION — 0 - NO VARIATION; 1 - VARIATION

2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION — MZERO(1,L) IS THE LOWER Bound AND MZERO(2,L) IS THE UPPER Bound

1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST

NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO, REGION PAIRS DURING INPUT

2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C NNODAT  Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
C NP()  2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C NPTS()  1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C NSETS  NUMBER OF RELATED MATERIAL S/N DATA SETS
C NUM  NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C NUMREG  NUMBER OF REGIONS OF INTEREST
C RATIO  STRESS RATIO (R = -1.0 IS DESIRED)
C RAISTR()  2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C RAWNP()  2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C RAWSTR()  2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
C REFP()  1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C SIGMA2()  1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C STR()  2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C SZERO  STRESS TENSILE TEST POINT, SO
C VARY  CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C ZRREG  ZERO REGION -- VALUES CHOSEN TO FACILITATE
C
C INITIALIZE COUNT AND NBND()
C COUNT = 0
C DO 10 L = 0, MAXREG
C  NBND(L) = 0.0
C 10 CONTINUE
C INPUT DATA ON SPECIFIC MATERIAL FROM SPECFD AND ECHO TO SPECFO
C READ(1,*) DESCRP(0), FTU, NPTS(0)
C IF (NPTS(0) .GT. MAXDAT) THEN
C  WRITE(8,*) 'ERROR: OVER NUMBER OF POINTS LIMIT IN ', &
C  'SPECIFIC MATERIAL'
C  CALL TRMNAT
C ENDIF
C WRITE(3,900) DESCRP(0), FTU, NPTS(0)
C IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTU, NPTS(0)
C WRITE(3,905)
C IF (IOUT .EQ. 10) WRITE(8,905)
C STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ
C FTUZ = FTU
C FTYZ = FTU
C INPUT STRESS/LIFE INFORMATION -- INCLUDING STRESS RATIO AND REGION
C INFORMATION FROM SPECFD AND ECHO TO SPECFO
C DO 100 M = 1, NDIV
C READ (1,*) NUM, RATIO, REG
C IF (ABS(RATIO) .GT. 1.0) THEN
C  WRITE(8,*) 'ERROR: INVALID VALUE FOR RATIO: ', RATIO
C  CALL TRMNAT
C ENDIF
C IF (REG .GT. MAXREG) THEN
C  WRITE(8,*) 'ERROR: OVER REGION LIMIT IN SPECIFIC DATA SET'
C  CALL TRMNAT
C ENDIF

7 - 125
DO 110 I = (COUNT + 1), (COUNT + NUM)
READ(1,*) RAWSTR(I,0), RAWNF(I,0)
CONTINUE

C CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
C IF (RATIO .EQ. -1.0) THEN
C STRESS RATIO IS CORRECT
DO 120 I = (COUNT + 1), (COUNT + NUM)
  RATSTR(I,0) = RAWSTR(I,0)
120 CONTINUE
ELSE
C STRESS RATIO TRANSFORMATION MUST BE DONE
  CALL CONVRT (0, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR, RATIO, FTU, FTY)
ENDIF
C ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL
DO 130 I = (COUNT + 1), (COUNT + NUM)
  WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
  RATSTR(I,0), RAWNF(I,0)
  IF (IOUT .EQ. I0) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
    RATIO, REG, RATSTR(I,0), RAWNF(I,0)
130 CONTINUE
C BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C EXPCTD, AND PAREST
K = NP(0,REG)
DO 140 I = (COUNT + 1), (COUNT + NUM)
  K = K + 1
  LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
  LNNF(K,0,REG) = ALOG(RAWNF(I,0))
  STR(K,REG) = RATSTR(I,0)
  NF(K,REG) = RAWNF(I,0)
140 CONTINUE
IF (K .GT. MAXDAT) THEN
  WRITE(8,*) 'ERROR: OVER NUMBER OF POINTS LIMIT IN ',
    'SPECIFIC MATERIAL'
  CALL TRMNAT
ENDIF
NP(0,REG) = K
REFNP(REG) = K
COUNT = COUNT + NUM
100 CONTINUE
IF (NPTS(0) .NE. COUNT) THEN
  WRITE(8,*), 'ERROR: NUMBER OF POINTS PER DIVISION ',
    'INCORRECTLY SPECIFIED'
  WRITE(8,*), 'IN SPECIFIC DATA SET'
  CALL TRMNAT
ENDIF
READ(1,*) SZERO
IF (NINT (SZERO) .GT. 0) THEN
  ZROREG = 0
ELSE
  ZROREG = 1

7 - 126
ENDIF
IF (IOUT .EQ. 10)
& WRITE(8,*), 'ZZERO = ', ZZERO, ' ZROREG = ', ZROREG
C
INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION
READ(1,*), NUMREG, NNODAT
IF ((NUMREG + NNODAT) .GT. MAXREG) THEN
WRITE(8,*), 'ERROR: EXCEEDED LIMIT ON NUMBER OF REGIONS'
CALL TRMNAT
ENDIF
DO 150 L = ZROREG, (NUMREG + NNODAT)
READ(1,*), NBND(L)
150 CONTINUE
READ(1,*), CZERO
DO 160 L = 1, (NUMREG + NNODAT)
READ(1,*), MPNT(L), MZERO(1,L), MZERO(2,L)
160 CONTINUE
WRITE(3,913)
IF (ZROREG .EQ. 0) WRITE(3,914) CZERO
IF (IOUT .EQ. 10) THEN
WRITE(8,913)
IF (ZROREG .EQ. 0) WRITE(8,914) CZERO
ENDIF
WRITE(3,915) NUMREG, NNODAT
IF (IOUT .EQ. 10) WRITE(8,915) NUMREG, NNODAT
DO 170 L = ZROREG, (NUMREG + NNODAT)
WRITE(3,920) NBND(L)
IF (IOUT .EQ. 10) WRITE(8,920) NBND(L)
170 CONTINUE
WRITE(3,925) CZERO
IF (IOUT .EQ. 10) WRITE(8,925) CZERO
DO 180 L = 1, (NUMREG + NNODAT)
WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
& WRITE(8,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
& IF ((VARY .EQ. 3) .AND. (MPNT(L) .EQ. 0)) THEN
& WRITE(8,*), 'ERROR: NORMAL VARIATION REQUIRES A PRIOR ',
& 'RANGE ON M'
CALL TRMNAT
180 CONTINUE
ENDIF
IF (VARY .EQ. 3) THEN
READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
WRITE(3,945)
IF (IOUT .EQ. 10) WRITE(8,945)
DO 190 L = 1, (NUMREG + NNODAT)
READ(1,*), DELTA(L), MO(L), SIGMA2(L)
WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
IF (IOUT .EQ. 10)
& WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
& IF ((DELTA(L) .LT. 0.0) .OR. [(DELTA(L) .LT. 0.0) .AND. (MO(L) .LE. 0.0)]) THEN
& WRITE(8,*), 'ERROR: BAD VALUE FOR DELTA OR VALUE OF MO ',
& 'INCONSISTENT WITH DELTA IN REGION ', L
CALL TRMNAT
190 CONTINUE
ENDIF
C
IF (MPROC .EQ. 1) THEN
READ(1,*), KRATIO, LAMN
WRITE(3,955) KRATIO, LAMN
IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
ENDIF
C BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATED 
C AND THEN ECHO TO RELATED

READ(5,*) NSETS

IF (NSETS .GT. MAXSET) THEN
   WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF RELATED DATA SETS'
   CALL TRMNAT
ENDIF

WRITE(6,935) NSETS

DO 200 J = 1, NSETS
   COUNT = 0
   IF (IOUT .EQ. 10) WRITE(8,*)'J = ', J, ' NSETS = ', NSETS
   READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)
   IF (NPTS(J) .GT. MAXDAT) THEN
      WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF POINTS IN ', 
      'SET ', J
      CALL TRMNAT
   ENDIF
   WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
   IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)
   WRITE(6,905)
   IF (IOUT .EQ. 10) WRITE(8,905)
   DO 300 M = 1, NDIV
      READ(5,*) NUM, RATIO, REG
      IF (ABS(RATIO) .GT. 1.0) THEN
         WRITE(8,*) 'ERROR: INVALID VALUE OF RATIO: ', RATIO
         CALL TRMNAT
      ENDIF
      IF (REG .GT. MAXREG) THEN
         WRITE(8,*) 'ERROR: OVER REGION LIMIT IN RELATED 
      MATERIAL ', J
         CALL TRMNAT
      ENDIF
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
         WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
      ENDIF
      DO 310 I = (COUNT + 1), (COUNT + NUM)
         READ(5,*) RAWSTR(I,J), RAWNF(I,J)
      310 CONTINUE
   ENDIF
C CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
C
IF (RATIO .EQ. -1.0) THEN
   C STRESS RATIO IS CORRECT
   DO 320 I = (COUNT + 1), (COUNT + NUM)
      RATSTR(I,J) = RAWSTR(I,J)
   320 CONTINUE
ELSE
   C STRESS RATIO TRANSFORMATION MUST BE DONE
   CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR, 
   RATSTR, RATIO, FTU, FTY)
ENDIF
C
RECORD BOTH S/N DATA SETS TO RELATO
DO 330 I = (COUNT + 1), (COUNT + NUM)
  WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG, RATSTR(I,J), RAWNF(I,J)
IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG, RATSTR(I,J), RAWNF(I,J)
330 CONTINUE
K = NP(J,REG)
DO 340 I = (COUNT + 1), (COUNT + NUM)
  K = K + 1
  LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
  LNNF(K,J,REG) = ALOG(RAWNF(I,J))
340 CONTINUE
IF (K .GT. MAXDAT) THEN
  WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF POINTS ',
  CALL TRMNAT
ENDIF
NP(J,REG) = K
COUNT = COUNT + NUM
300 CONTINUE
IF (NPTS(J) .NE. COUNT) THEN
  WRITE(8,*) 'ERROR: NUMBER OF POINTS PER DIVISION ',
  CALL TRMNAT
ENDIF
200 CONTINUE

C FORMAT STATEMENTS USED TO WRITE TO SPECFO AND RELATO

900 FORMAT(///,13X,'MATERIAL INPUT',///,2X,'DESCRIPTION:',2X,A40,///,
  & 2X,'YIELD STRENGTH',16X,EL11.5,///,2X,'ULTIMATE STRENGTH',
  & 15X,EL11.5,///,2X,'NUMBER OF POINTS',16X,I2)
905 FORMAT(///,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
  & /5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
  & 'STRESS','LIFE'/)
910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)
913 FORMAT(/)
914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN SO OF',
  & 5X,E11.5)
915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA ',
  & /2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA ',
  & '/2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
  & 'CYCLES': '/)
920 FORMAT(10X,E9.3)
925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
  & 'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
  & 2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
  & 5X,'UPPER BOUND:',/)
930 FORMAT(6X,I1,11X,I1,12X,F7.4,9X,F7.4)
C THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C STRESS RATIO, R, IS NOT -1.0
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87 COMMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C V8.3, V8.4, V8.5

SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)

C INPUTS: J, NUM1, NUM2, STR, RSTR, R, FTU, FTY
C OUTPUTS: RSTR
C IMPLICIT NONE
INTEGER MAXDAT, MAXSET
PARAMETER (MAXDAT = 50, MAXSET = 5)
COMMON IOUT
INTEGER I, IOUT, J, NUM1, NUM2
REAL FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
& STR(MAXDAT, 0:MAXSET), TEST

LIST OF VARIABLES

FTU ULTIMATE STRENGTH OF MATERIAL (PSI)
FTY YIELD STRENGTH OF MATERIAL (PSI)
I CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
IOUT OUTPUT DUMP CONTROLLER
J DATA SET OF INTEREST
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
NUM1 FIRST INDEX TO BE TRANSFORMED
NUM2 LAST INDEX TO BE TRANSFORMED
R STRESS RATIO (R = -1.0 IS DESIRED)
RSTR() STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
STR() ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
TEST Kt * Smax * (1 - R)/2, TO BE COMPARED WITH FTY

C Kt IS ASSUMED TO BE ONE
DO 100 I = NUM1, NUM2
TEST = STR(I,J) * (1.0 - R)/2.0
IF (IOUT.EQ.10) WRITE(8,*) 'I = ', I, ', J = ', J, ', TEST = ', TEST

IF (TEST .GE. FTY) THEN
   RSTR(I,J) = TEST
   IF (IOUT.EQ.10) WRITE(8,*)'1: RSTR() = ', RSTR(I,J)
ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
   RSTR(I,J) = TEST/(1.0 - ((FTY - TEST)/FTU))
   IF (IOUT.EQ.10) WRITE(8,*)'2: RSTR() = ', RSTR(I,J)
ELSE
   RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
                   /(2.0 * PTU)))
   IF (IOUT.EQ.10) WRITE(8,*)'3: RSTR() = ', RSTR(I,J)
END IF
100 CONTINUE
RETURN
END

C**************************************************************

C SUBROUTINE SW2SU2 CALCULATES, WHAT2, THE RESIDUAL VARIANCES OF Y ON X
C AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C & V8.4, V8.5

SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR)

C INPUTS: NUMREG, NSETS, NP, LNSTR, LNNF
C OUTPUTS: SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR

C IMPLICIT NONE

INTEGER AXDAT, AXRG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, J, K, L, NP(0:MAXSET, MAXREG), NPPR(MAXREG),
& NSETS, NUMREG
REAL BB(MAXREG), DD(MAXREG), DIFFX(MAXDAT, 0:MAXSET),
& DIFFY(MAXDAT, 0:MAXSET), LNNF(MAXDAT, 0:MAXSET, MAXREG),
& LNSTR(MAXDAT, 0:MAXSET, MAXREG), MEANX(0:MAXSET),
& MEANY(0:MAXSET), SUHAT2(MAXREG), WHAT2(MAXREG),
& SX2(MAXREG), SY2(MAXREG)

C LIST OF VARIABLES
C BB() 1-D ARRAY CONTAINING SXY(L)/SY2(L) FOR EACH REGION
C DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C DIFFX() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L)

7 - 131
DIFFY() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(X,J,L) AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
L CONTROLS DO LOOP FOR EACH REGION
LNNF() 3-D ARRAY CONTAINING LN(RAWNF(0)), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR(0)), ALSO INDEXED FOR REGION
MAXDAT MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MEANX() 1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION L AND DATA SET J (X = Ln S)
MEANY() 1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION L AND DATA SET J (Y = Ln N)
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP() - 1)) - 1) OVER ALL DATA SETS IN A REGION (NUMBER OF POINTS PER REGION)
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMREG NUMBER OF REGIONS OF INTEREST
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR EACH REGION (X = Ln S, Y = Ln N)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION (Y = Ln N)

C INITIALIZE ARRAYS

DO 50 L = 1, NUMREG
  SY2(L) = 0.0
  SX2(L) = 0.0
  SXY(L) = 0.0
  SWHAT2(L) = 0.0
  SUHAT2(L) = 0.0
  BB(L) = 0.0
  DD(L) = 0.0
  NPPR(L) = 0
50 CONTINUE

DO 60 J = 0, MAXSET
  DO 70 K = 1, MAXDAT
    DIFFY(K,J) = 0.0
    DIFFX(K,J) = 0.0
  70 CONTINUE
MEANY(J) = 0.0
MEANX(J) = 0.0
60 CONTINUE

C NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION

DO 100 L = 1, NUMREG
  DO 200 J = 0, NSETS
    FIRST CALCULATE SAMPLE X AND Y MEANS
    FOR DATA SET J IN REGION L
    MEANY(J) = 0.0
    MEANX(J) = 0.0
    IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
      ' NP =', NP(J,L)
    DO 250 K = 1, NP(J,L)
      MEANY(J) = MEANY(J) + LNNF(K,J,L)
      MEANX(J) = MEANX(J) + LNSTR(K,J,L)
      IF (IOUT .EQ. 10) WRITE(8,*)'LNNF =', LNNF(K,J,L),
        ' LNSTR =', LNSTR(K,J,L)
    250 CONTINUE
MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
MEANX(J) = MEANX(J)/FLOAT(NP(J,L))

IF (IOUT .EQ. 10) WRITE(8,*), 'MEANY(J) =', MEANY(J),
& 'MEANX(J) =', MEANX(J)

NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY,
OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
DATA SET IN REGION L

DO 300 K = i, NP(J,L)
  DIFFY(K,J) = LNNF(K,J,L) - MEANY(J)
  DIFFX(K,J) = LNSTR(K,J,L) - MEANX(J)
  SY2(L) = SY2(L) + DIFFY(K,J) ** 2
  SX2(L) = SX2(L) + DIFFX(K,J) ** 2
  SXY(L) = SXY(L) + DIFFX(K,J) * DIFFY(K,J)
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*), 'K =', K, 'DIFFY(K,J) =', DIFFY(K,J),
    'DIFFX(K,J) =', DIFFX(K,J),
    'SY2(L) =', SY2(L), 'SX2(L) =', SX2(L),
    'SXY(L) =', SXY(L)
  ENDIF
CONTINUE

NPPR(L) = NPPR(L) + NP(J,L) - 1
IF (IOUT .EQ. 10) WRITE(8,*), 'NPPR(L) =', NPPR(L)

300 CONTINUE

IF (SXY(L) .GE. 0.0) THEN
  LIFE WILL INCREASE WITH INCREASING STRESS -- INVALID FOR
  OUR MODEL
  WRITE(8,*), 'ERROR: SXY >= 0 IN REGION', L
  CALL TRMNAT
ENDIF

NPPR(L) = NPPR(L) - 1
IF (NPPR(L) .LE. 0) THEN
  WRITE(8,*), 'ERROR: TOO FEW POINTS FOR REGRESSION IN ',
  'REGION ', L
  CALL TRMNAT
ENDIF

SY2(L) = SY2(L) / FLOAT(NPPR(L))
SX2(L) = SX2(L) / FLOAT(NPPR(L))
SXY(L) = SXY(L) / FLOAT(NPPR(L))

NOW CALCULATE THE RESIDUAL VARIANCES, SWHAT2, SUHAT2, FOR EACH
REGION FROM THE Y ON X AND X ON Y REGRESSIONS

DD(L) = SXY(L) / SX2(L)
BB(L) = SXY(L) / SY2(L)

IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'NPPR(L) =', NPPR(L), 'SY2(L) =', SY2(L),
  'SX2(L) =', SX2(L), 'SXY(L) =', SXY(L), 'DD(L) =', DD(L),
  'BB(L) =', BB(L)
ENDIF

DO 400 J = 0, NSETS
  IF (IOUT .EQ. 10) WRITE(8,*), 'J =', J, 'NP(J,L) =', NP(J,L)

  DO 500 K = 1, NP(J,L)
    SWHAT2(L) = SWHAT2(L) + (DIFFX(K,J) - DD(L) * DIFFX(K,J)) ** 2
    SUHAT2(L) = SUHAT2(L) + (DIFFX(K,J) - BB(L) * DIFFX(K,J)) ** 2
    IF (IOUT .EQ. 10) WRITE(8,*), 'K =', K, 'SWHAT2(L) =', SWHAT2(L),
    'SUHAT2(L) =', SUHAT2(L)
  ENDIF
CONTINUE

400 CONTINUE

SWHAT2(L) = SWHAT2(L) / FLOAT(NPPR(L))
SUHAT2(L) = SUHAT2(L) / FLOAT(NPPR(L))

IF (IOUT .EQ. 10) WRITE(8,*), 'NPPR(L) =', NPPR(L),

7 - 133
& ' WHAT2(L) =', WHAT2(L), ' SUHAT2(L) =', SUHAT2(L)
100 CONTINUE

RETURN
END

C*****************************************************************************
C SUBROUTINE INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, IO, ON
C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 5OCT87 COMMENTS: 15SEP89
C VERSION: MATCHR V6, V6.1, V6.2, V8, V8.1, V8.2, V8.3,
C V8.4, V8.5 ' '
C*****************************************************************************

SUBROUTINE INTRVL

C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C OUTPUTS: IZERO, JZERO, MCHAT
C SUBPROGRAMS: TRMNAT

C IMPLICIT NONE
INTEGER CHITAB, MAXREG, TTAB
PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)
COMMON IOUT
INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG
REAL
& DATA (CHI025(I), I = 1, 75) /
& 0.000992669, 0.506356, 0.215795, 0.484419, 0.831211,
& 1.237347, 1.68975, 2.17973, 2.70039, 3.24697,
& 3.81575, 4.40379, 5.00874, 5.62872, 6.26214,
& 6.90766, 7.56418, 8.23075, 8.90655, 9.59083,
& 10.28293, 10.9823, 12.70039, 13.1197, 13.8439,
& 14.5733, 15.3079, 16.0471, 16.7908, 17.53,
& 18.28, 19.04, 19.80, 20.56, 21.33, 22.10, 22.87,
& 23.64, 24.41, 25.19, 25.96, 26.73, 27.51, 28.28,
& 29.05, 29.82, 30.59, 31.36, 32.13, 32.89, 33.66,
& 34.43, 35.20, 35.97, 36.74, 37.51, 38.28, 39.05,
& 39.82, 40.59, 41.36, 42.13, 42.90, 43.67, 44.44,
& 45.21, 45.98, 46.75, 47.52, 48.29, 49.06, 49.83,
& 50.60, 51.37, 52.14, 52.91, 53.68, 54.45, 55.22,
& 55.99, 56.76, 57.53, 58.30, 59.07, 59.84, 60.61,
& 61.38, 62.15, 62.92, 63.69, 64.46, 65.23, 66.00,
& 66.77, 67.54, 68.31, 69.08, 69.85, 70.62, 71.39,
& 72.16, 72.93, 73.70, 74.47, 75.24, 76.01, 76.78,
& 77.55, 78.32, 79.09, 79.86, 80.63, 81.40, 82.17,
& 82.94, 83.71, 84.48, 85.25, 86.02, 86.79, 87.56,
& 88.33, 89.10, 89.87, 90.64, 91.41, 92.18, 92.95,
& 93.72, 94.49, 95.26, 96.03, 96.80, 97.57, 98.34,
& 99.11, 99.88, 100.65, 101.42, 102.19, 102.96, 103.74,
& 104.51, 105.28, 106.05, 106.82, 107.59, 108.36, 109.14,
& 109.91, 110.68, 111.45, 112.22, 112.99, 113.76, 114.53,
& 115.30, 116.07, 116.84, 117.61, 118.38, 119.15, 119.92,
& 120.69, 121.46, 122.23, 122.99, 123.77, 124.54, 125.31,
& 126.07, 126.85, 127.61, 128.38, 129.14, 129.91, 130.68,
& 131.44, 132.21, 132.97, 133.74, 134.51, 135.27, 136.04,
& 136.80, 137.57, 138.33, 139.09, 139.86, 140.62, 141.38,
& 142.15, 142.91, 143.67, 144.43, 145.19, 145.95, 146.71,
```
C VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:

1 - 30, 40, 50, 60, 70, 80, 90, 100 – Theil, pp. 718-719
- CALCULATED USING CUBE RULE APPROXIMATION

DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447, 2.321, 2.195, 2.145, 2.101, 2.086, 2.056, 2.048, 2.045, 2.042, 1.960 /

LIST OF VARIABLES
ARG INTERMEDIATE CALCULATION VARIABLE
CHI025() TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHI975() TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHITAB MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
I0 CONTROLS LOOP FOR CHI025() AND CHI975()
JZERO() 2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C FOR EACH REGION
JZERO() 2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
L MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION BASED ON MATERIALS DATA ONLY –
MCHAT(I,L) = -DD, THE ESTIMATE FOR M AND
MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-I))-I) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
NUM EQUAL TO 0.025*NPPR(L) FOR A SET OF CALCULATIONS
NUMREG NUMBER OF REGIONS OF INTEREST
SUHAT EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
```

```
C
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SWHAT 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(X = Ln S)
TTAB TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
MAXIMUM NUMBER OF DEGREES OF FREEDOM IN TTAB

C
INITIALIZE IZERO, JZERO AND MCHAT

DO 50 L = 1, MAXREG
IZERO(1,L) = 0.0
IZERO(2,L) = 0.0
JZERO(1,L) = 0.0
JZERO(2,L) = 0.0
MCHAT(1,L) = 0.0
MCHAT(2,L) = 0.0
50 CONTINUE

C
CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

DO 75 L = 1, NUMREG
IF (NPPR(L) .GT. CHITAB) THEN
WRITE (8,*)'ERROR: EXCEEDED LIMIT ON DEGREES OF FREEDOM ',
          &' IN CHI-SQUARE TABLE, IN REGION ', L
CALL TRMNAT
ENDIF
75 CONTINUE

C
ASSIGN VALUES TO NUM, T, SWHAT, SUHAT AND THEN CALCULATE
CONFIDENCE INTERVALS FOR EACH REGION

DO 100 L = 1, NUMREG
NUM = NPPR(L)
IF (NUM .LT. 31) THEN
T = T025(NUM)
ELSE
T = T025(NUM)
ENDIF
SWHAT = SWHAT2(L) ** 0.5
SUHAT = SUHAT2(L) ** 0.5
SX = (NUM * SX2(L)) ** 0.5

C
CALCULATE ESTIMATED VALUES OF M AND C

ARG = T * SWHAT / SX
MCHAT(1,L) = - DD(L)
MCHAT(2,L) = SUHAT

C
CALCULATE CONFIDENCE INTERVALS

IZERO(1,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI975(NUM)) ** 0.5
IZERO(2,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI025(NUM)) ** 0.5
JZERO(1,L) = MCHAT(1,L) - ARG
JZERO(2,L) = MCHAT(1,L) + ARG

IF (IOUT .EQ. 10) THEN
WRITE (8,*)'L = ', L, ' NPPR = ', NPPR(L), ' NUM = ', NUM
WRITE (8,*)'SWHAT = ', SWHAT2(L), ' SUHAT2 = ', SUHAT2(L)
WRITE (8,*)'SWHAT = ', SWHAT
WRITE (8,*)'SUHAT2 = ', SUHAT2(L), ' SUHAT = ', SUHAT
WRITE (8,*)'SX2 = ', SX2(L), ' SX = ', SX
WRITE (8,*)'CHI025 = ', CHI025(NUM), ' CHI975 = ', CHI975(NUM)
WRITE (8,*)'T = ', T, ' DD = ', DD(L), ' ARG = ', ARG
WRITE (8,*)'IZERO(1,L) = ', IZERO(1,L), ' IZERO(2,L) = ', IZERO(2,L)
WRITE (8,*)'JZERO(1,L) = ', JZERO(1,L), ' JZERO(2,L) = '
&
7 - 136
& WRITE(8,*) 'MCHAIN(I,L) = ', MCHAIN(1,L), ' MCHAIN(2,L) = ',
& ENDIF
100 CONTINUE
RETURN
END

C***************************************************************

C SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C THE CO GIVEN BY THE USER
C PROGRAMMER: L. NEWLIN
C DATE: 8OCT87 COMMENTS: 13JUL89
C SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
C INPUTS: NUMREG, CZERO, SX2, SXY, SY2
C OUTPUTS: MCPNT, MC
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), NUMREG
REAL ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
& SXY(MAXREG), SY2(MAXREG)

LIST OF VARIABLES
ARG1 INTERMEDIATE CALCULATION VARIABLE
ARG2 INTERMEDIATE CALCULATION VARIABLE
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
COEFFICIENT OF VARIATION, CO
CZERO2 EQUAL TO CZERO ** 2
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA — MC(1,L) IS
THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MC() FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR
EACH REGION (X = Ln S, Y = Ln N)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
(Y = Ln N)

C INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
  MCPNT(L) = 0
  MC(1,L) = 0.0
  MC(2,L) = 0.0
50 CONTINUE
C BEGIN CALCULATIONS
CZERO2 = CZERO ** 2
IF (IOUT .EQ. 10)
  WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2
DO 100 L = 1, NUMREG
  ARG1 = SX2(L) - CZERO2
  ARG2 = 0.0
  IF (CZERO .EQ. 0.0) THEN
C THEN NO M CONSTRAINT IS REQUIRED
    MCPNT(L) = 0
  ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN
C THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M
    MCPNT(L) = 1
    MC(1,L) = - SY2(L) / (2.0 * SXY(L))
  ELSE
C THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C COMMON CALCULATIONS
    ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)
  IF (ARG2 .LT. 0.0) THEN
C ARG2 IS NEGATIVE -- IMPLIES M IS COMPLEX
    WRITE ( 8, * )
C 'ERROR: CO TOO LOW'
    CALL TRMNAT
  ELSE
    ARG2 = ARG2 ** 0.5
  ENDIF
  IF (SX2(L) .LT. CZERO2) THEN
C AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M
    MCPNT(L) = 1
    MC(1,L) = (- SXY(L) - ARG2) / ARG1
  ELSE
C SX2(L) .GT. CZERO2 -- THIS TIME THE M CONSTRAINT IS A RANGE
    MCPNT(L) = 2
    MC(1,L) = (- SXY(L) - ARG2) / ARG1
    MC(2,L) = (- SXY(L) + ARG2) / ARG1
  ENDIF
ENDIF

100 CONTINUE
IF (IOUT .EQ. 10) THEN
  DO 200 L = 1, NUMREG
    WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
    WRITE(8,*) 'ARG1 = ', ARG1, ' ARG1', ' ARG2 = ', ARG2
    WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
  200 CONTINUE
ENDIF
SUBROUTINE GTPVAR CALCULATES THE EXTENT OF DEPARTURE FROM THE MULTIPLE 
HEAT MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION

PROGRAMMER: L. NEWLIN
DATE: CODE: 21JUN88 COMMENTS: 13JUL89
VERSION: MATCHR V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)

INPUTS: NSETS, NP, NUMREG, LAMN, MCHAT
OUTPUTS: PVAR

IMPLICIT NONE

INTEGER MAXREG, MAXSET
PARAMETER (MAXREG = 3, MAXSET = 5)
COMMON IOUT
   INTEGER IOUT, J, L, LAMN, MAXREG
   REAL MAXSET, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
L CONTROLS DO LOOP FOR EACH REGION
LAMN LAMBDAN - RATIO OF Var (ln N given S) / (m**2 C**2),
   CONSTANT OVER REGIONS AND COMPONENTS
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
   FOR EACH REGION, BASED ON MATERIALS DATA ONLY -
   MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
   MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
   SET IN EACH REGION
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUM() EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
   NUMBER OF POINTS IN EACH DATA SET
NUMREG NUMBER OF REGIONS OF INTEREST
PSIG2() 1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
   VARIATION IN EACH REGION
PVAR THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
   CURVE WARRANTED BY THE AVAILABLE INFORMATION
SUM WEIGHTED SUM OF THE PSIG2s - USED TO CALCULATE A WEIGHTED
   AVERAGE
TOTAL SUM OF NUM() OVER ALL REGIONS

INITIALIZE VARIABLES

SUM = 0.0
TOTAL = 0.0

DO 50 L = 1, MAXREG
   PSIG2(L) = 0.0
   NUM(L) = 0
50 CONTINUE

DO 100 L = 1, NUMREG
DO 150 J = 0, NSETS
NUM(L) = NUM(L) + NP(J,L)
150 CONTINUE
NUM(L) = NUM(L) - 1
TOTAL = TOTAL + NUM(L)
100 CONTINUE
DO 200 L = 1, NUMREG
PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
SUM = SUM + PSIG2(L) * NUM(L)
200 CONTINUE
IF (IOUT .EQ. 10) THEN
WRITE(8,*) 'LAMN = ', LAMN
WRITE(8,*) 'NUM = ', NUM(L)
WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
300 CONTINUE
ENDIF
PVAR = SUM / FLOAT (TOTAL)
RETURN
END

C**********************************************************************

SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
M AND CO WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
PROGRAMMER: L. NEWLIN
DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91
SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT, RANGEM)

C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
& MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
JZERO() 2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
FOR EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
LOWER LOWER BOUND OF INTERSECTION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
BOUND
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MC() FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
IS THE UPPER BOUND
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND
UPPER UPPER BOUND OF INTERSECTION

C INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
RANGEM(1,L) = 0.0
RANGEM(2,L) = 0.0
50 CONTINUE

C PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
DO 100 L = 1, NUMREG
IF (IOUT .EQ. 10) THEN
WRITE(*,*) 'L = ', L, ' NUMREG = ', NUMREG
WRITE(*,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
ENDIF
IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN
THERE IS NO EXOGENOUS INFORMATION
ASSUME RANGE TO BE JO
RANGEM(1,L) = JZERO(1,L)
RANGEM(2,L) = JZERO(2,L)
IF (IOUT .EQ. 10) THEN
WRITE(*,*) 'RANGEM_1,L) =,,, RANGEM_1,L) = JZERO(I,L)
WRITE(*,*) 'RANGEM_2,L) = ,_, RANGEM_2,L) = JZERO(2,L)
ENDIF
ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN
NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
TO CO, ADJUST THE LOWER BOUND OF J0 ACCORDINGLY
LOWER = AMAX1(JZERO(1,L), MC(1,L))
UPPER = JZERO(2,L)
IF (UPPER .LT. LOWER) THEN
WRITE(*,*) 'ERROR: NO INTERSECTION BETWEEN JO AND MC'
CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF
ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN
THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
CORRESPONDING TO THE CO CONSTRAINT, ADJUST J0 ACCORDINGLY

C
LOWER = AMAX1(JZERO(1,L), MC(1,L))
UPPER = AMIN1(JZERO(2,L), MC(2,L))
IF (UPPER .LT. LOWER) THEN
   WRITE(8,*,'ERROR: NO INTERSECTION BETWEEN JO AND MC'
   CALL TRMNAT
ELSE
   RANGEM(1,L) = LOWER
   RANGEM(2,L) = UPPER
ENDIF
IF (IOUT .EQ. 10) THEN
   WRITE(8,*,'JZERO(1,L) = ', JZERO(I,L),
      MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
   WRITE(8,*,'LOWER = ', LOWER, ' UPPER = ', UPPER
   WRITE(8,*,'RANGEM(1,L) = ', RANGEM(1,L),
      RANGEM(2,L) = ', RANGEM(2,L)
ENDIF
ELSEIF (MPNT(L) .EQ. 1) THEN
   THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER
   INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
   RANGEM(1,L) = MZERO(1,L)
   RANGEM(2,L) = 0.0
IF (IOUT .EQ. 10) THEN
   WRITE(8,*,'MZERO(1,L) = ', MZERO(1,L),
      RANGEM(1,L) = ', RANGEM(1,L),
      RANGEM(2,L) = ', RANGEM(2,L)
ENDIF
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN
   THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT
   USE INTERSECTION BETWEEN JO AND MO
   LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
   UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
   IF (UPPER .LT. LOWER) THEN
      WRITE(8,*,'ERROR: NO INTERSECTION BETWEEN JO AND MO'
      CALL TRMNAT
   ELSE
      RANGEM(1,L) = LOWER
      RANGEM(2,L) = UPPER
   ENDIF
IF (IOUT .EQ. 10) THEN
   WRITE(8,*,'JZERO(1,L) = ', JZERO(I,L),
      MZERO(1,L) = ', MZERO(1,L),
      RANGEM(1,L) = ', RANGEM(1,L),
      RANGEM(2,L) = ', RANGEM(2,L)
ENDIF
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN
   THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO
   CONSTRAINT, INTERSECT JO AND MO, ADJUSTING THE LOWER BOUND
   BY MC ACCORDINGLY
   LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
   UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
   IF (UPPER .LT. LOWER) THEN
      WRITE(8,*,'ERROR: NO INTERSECTION BETWEEN JO, MO, ',
   CALL TRMNAT
   ELSE
      RANGEM(1,L) = LOWER
      RANGEM(2,L) = UPPER
   ENDIF
ENDIF
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  'JZERO(2,L) = ', JZERO(2,L),
  WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
  'MZERO(2,L) = ', MZERO(2,L),
  WRITE(8,*), 'MC(1,L) = ', MC(1,L),
  WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
  WRITE(8,*), 'RANGEM(1,L) = ', 'RANGEM(2,L)
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN
  THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO Co CONSTRAINT
  INTERSECT THESE TWO RANGES WITH Jo
  LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
  UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
  IF (UPPER .LT. LOWER) THEN
    WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN Jo, Mo, ',
    'AND MC'
    CALL TRMNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF
ELSEIF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  'JZERO(2,L) = ', JZERO(2,L),
  WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
  'MZERO(2,L) = ', MZERO(2,L),
  WRITE(8,*), 'MC(1,L) = ', MC(1,L),
  WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
  WRITE(8,*), 'RANGEM(1,L) = ', 'RANGEM(2,L)
ENDIF

ELSE
  WRITE(8,*), 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
  CALL TRMNAT
ENDIF

C

RESTRICT RANGE TO BE NON-NEGATIVE
RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)
IF (IOUT .EQ. 10) WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L)

100 CONTINUE

C

CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE
DO 300 L = 1, NUMREG
  IF ((MCHAT(1,L) .LT. RANGEM(1,L)) .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
    WRITE(8,*), 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
    'ON m IN REGION ', L
  CONTINUE
RETURN
END
SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS WITHOUT DATA

PROGRAMMER: L. NEWLIN

DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91


SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C INPUTS: RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT

C OUTPUTS: RANGEM, MCHAT, NUMREG

C IMPLICIT NONE

INTEGER MAXREG

PARAMETER (
MAXREG = 3
)

COMMON IOUT

INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

REAL MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LL, EQUAL TO NUMREG FOR A SET OF CALCULATIONS
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY — MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION — MZERO(I,LL) IS THE LOWER BOUND AND MZERO(2,LL) IS UPPER BOUND
NNODAT Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION — RANGEM(I,LL) IS THE LOWER BOUND AND RANGEM(2,LL) IS THE UPPER BOUND

IF (IOUT .EQ. 10) WRITE(8,*)'NUMREG =', NUMREG

DO 100 L = 1, NNODAT
NUMREG = NUMREG + 1
LL = NUMREG
IF (IOUT .EQ. 10) WRITE(8,*),'L =', L, 'NUMREG =', NUMREG,
& 'LL =', LL, 'MPNT(LL) =', MPNT(LL)

IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
POSTERIOR ON M IS SAME AS PRIOR ON M
RANGEM(1,LL) = MZERO(1,LL)
RANGEM(2,LL) = MZERO(2,LL)
IF (IOUT .EQ. 10) THEN
WRITE(8,*),'MZERO(1,LL) =', MZERO(1,LL),
& 'MZERO(2,LL) =', MZERO(2,LL),
ENDIF

SPECIFY E(M) OF POSTERIOR FOR SAKE OF CALCULATIONS IN SUBROUTINE EXPCTD

IF (RANGEM(2,LL) .EQ. 0.0) THEN
MCHAT(1,LL) = RANGEM(1,LL)
ELSE
MCHAT(I,LL) = (RANGEM(I,LL) + RANGEM(2,LL)) / 2.0
ENDIF
IF (IOUT .EQ. i0) WRITE(8,*) 'MCHAT =', MCHAT(I,LL)
ELSE
WRITE(8,*) 'ERROR: OVERALL PRIOR RANGE INCORRECTLY SPECIFIED IN REGION WITHOUT DATA'
CALL TRMNAT
ENDIF
100 CONTINUE
RETURN
END

C SUBROUTINE CONCAV ADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY RANGES ON M TO BE CONSISTENT WITH CONCAVITY CONSTRAINTS
C PROGRAMMER: L. NEWLIN
C DATE: 2FEB88  FORMAT/COMMENTS: 15SEP89
C
SUBROUTINE CONCAV (NUMREG, RANGEM)
C
INPUTS:  NUMREG, RANGEM
C OUTPUTS:  RANGEM
C SUBPROG_:  TRMNAT
C
IMPLICIT NONE
INTEGER MAXREG
PARAMETER (AXR G = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL RANGEM(2, MAXREG), TESTM

LIST OF VARIABLES

IOUT  OUTPUT DUMP CONTROLLER
L  CONTROLS DO LOOP FOR EACH REGION
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
NUMREG  NUMBER OF REGIONS OF INTEREST
RANGEM()  2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
          FOR EACH REGION - RANGEM(I,L) IS THE LOWER BOUND AND
          RANGEM(2,L) IS THE UPPER BOUND
TESTM  UPPER BOUND OF RANGE ON M IN REGION L-1 - USED DURING CONCAVITY ADJUSTMENT

ADJUST RANGE TO INSURE CONCAVITY

DO 100 L = NUMREG, 2, -1
  IF (RANGEM(2,L-1) .EQ. 0.0) THEN
    RANGE IS A POINT IN REGION L-1
    IF (RANGEM(1,L-1) .GT. AMAX1(RANGEM(1,L), RANGEM(2,L))) THEN
      WRITE(8,*) 'ERROR: POSTERIOR INTERVAL IN REGION ', L,
      & ' IS INCONSISTENT WITH POINT POSTERIOR IN REGION ', L-1
      CALL TRMNAT
    ENDIF
  ELSE
    RANGE IS AN INTERVAL IN REGION L-1
    TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
    IF (TESTM .LT. RANGEM(1,L-1)) THEN
      WRITE(8,*) 'ERROR: POSTERIOR INTERVAL IN REGION ', L,
IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN 
'REGION ', L-1 

ELSE 
ENDIF 

ENDIF 

IF (IOUT .EQ. I0) THEN 
WRITE(8,*)'RANGEM(1,L-I) = ', RANGEM(1,L-I), 
5 WRITE(8,*)'RANGEM(2,L-I) = ', RANGEM(2,L-I) 
6 WRITE(8,*)'TESTM = ', TESTM, ' L = ', L 
ENDIF 

100 CONTINUE 

RETURN 
END 

C SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER J0 HAS 
BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR CO 
PROGRAMMER: L. NEWLIN 
DATE: CODE: 5OCT87 COMMENTS: IDEC87 
VERSION: MATCHR V8, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3, 
V8.4, V8.5 
SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM) 
C INPUTS: NUMREG, RANGEM 
C IOUTPUT: MEDM 
C IMPLICIT NONE 
INTEGER MAXREG 
PARAMETER (MAXREG = 3) 
COMMON IOUT 
INTEGER IOUT, L, NUMREG 
REAL LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG) 

LIST OF VARIABLES 

IOUT OUTPUT DUMP CONTROLLER 
L CONTROLS DO LOOP FOR EACH REGION 
LOWER M LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION) 
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED 
MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION 
NUMREG NUMBER OF REGIONS OF INTEREST 
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M 
FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND 
RANGEM(2,L) IS THE UPPER BOUND 

C INITIALIZE ARRAY MEDM 
DO 50 L = 1, MAXREG 
MEDM(L) = 0.0 
50 CONTINUE
BEGIN CALCULATIONS FOR EACH REGION
DO 100 L = 1, NUMREG
   IF (RANGEM(2,L) .EQ. 0.0) THEN
      MEDM(L) = RANGEM(1,L)
   ELSEIF (L .EQ. 1) THEN
      MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
   ELSE
      MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
   ENDIF
IF (IOUT .EQ. I0) THEN
   WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
   WRITE(8,*), 'RANGEM(I,L) = ', RANGEM(I,L)
   WRITE(8,*), 'LOWERM = LOWERM, , MEDM(L) = MEDM(L)
ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE EXPCTD calculates the expected or median values of the S/N curve parameters
PROGRAMMER: L. NEWLIN
DATE: CODE: 13FEB89   format/comments: 15SEP89

SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, & ZROREG, NBND, BIGKI, BZHAT)

INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
OUTPUTS: BIGKI, BZHAT
SUBPROGRAMS: TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

IMPLICIT NONE

INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG
REAL BIGK(0:MAXREG), BIGKI, BZHAT, FACTR, KHAT, MEANZ, & MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG), & NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG),
LIST OF VARIABLES

BIGK()  1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
BIGKI = BIGK(1)
BZHAT = PHI * KRATIO * Z
FACTR = SCALE FACTOR = PHI * KRATIO * Z
IOUT = OUTPUT DUMP CONTROLLER
KHAT = E(K)
L = CONTROLS DO LOOP FOR EACH REGION
MAXDAT = MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG = MAXIMUM NUMBER OF REGIONS ALLOWED
MEANZ = SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
MEDM() = 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
MM() = 1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
NBND() = 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NCOMPS = Number of components – 1 FOR STRESS AND STRAIN WHEN DECOMPOSED DATA UNAVAILABLE = 2 FOR DECOMPOSED STRAIN DATA
NF() = 2-D ARRAY CONTAINING RANNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP = TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
NPTS() = 1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR THE SPECIFIC MATERIAL S/N DATA SET
NUMREG = NUMBER OF REGIONS OF INTEREST
SBND() = 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()
STR() = 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
SZ2 = SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
SZERO = STRESS TENSILE TEST POINT, So
TRBIGK() = 1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
TRBIGK(1) = BIGK(1)
ZROREG = ZER0 REGION – VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE = 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
ZZ() = 1-D ARRAY CONTAINING TRANSFORMED S-N DATA, Z = F(STR,NF,NBND,MM)

C INITIALIZE VARIABLES

DO 50 L = 0, MAXREG
  MM(L) = 0.0
50 CONTINUE

C CREATE MM() ARRAY FROM MEDM() ARRAY

DO 100 L = 1, NUMREG
  MM(L) = MEDM(L)
100 CONTINUE

C TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C CALCULATE BETAO AND k

CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

BIGKI = BIGK(1)

C CALCULATE BOUNDARIES OF STRESS REGIONS
CALL FINDSDB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C CALLS KOMO TO CALCULATE KO AND MO FOR THE NO DATA REGION TO THE LEFT IF REQUIRED

DO 150 L = ZROREG, NUMREG
  TRBIGK(L) = BIGK(L)
150 CONTINUE

IF (ZROREG .EQ. 0) THEN
  FACTR = 1.0
  CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK, FACTR, NUMREG)
ENDIF

C WRITE RESULTS TO FILE

IF (NCOMPS .EQ. 1) THEN
  WRITE(7,900) NUMREG, BZHAT, KHAT
  IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZHAT, KHAT

  DO 200 L = ZROREG, NUMREG
    WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
    IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
 200 CONTINUE

  WRITE(7,920)
ELSE
  WRITE(7,930) MM(1), BIGK(1), KHAT
ENDIF

FORMAT STATEMENTS

900 FORMAT(///, 'PARAMETER VALUES FOR MEDIAN S/N CURVE', ', ', 'NUMBER OF REGIONS: ', 'MEAN, MOV AND STANDARD DEVIATION, SIG:', //, I4, 5X, 'E(BETA) = ', 'E(K) = ', 'STRESS BOUND', // )
910 FORMAT(5X, I1, 5X, F9.5, 5X, E12.5, 5X, E9.3, 9X, E11.5)
920 FORMAT(///)

RETURN
END

C******************************************************************************

C SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88 COMMENTS: 13JUL89

SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2, MCHAT, MU, SIG)

C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2
C OUTPUTS: MCHAT, MU, SIG
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG, NPPR(MAXREG)
REAL ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
& MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
& SUHAT2(MAXREG), SUMX2, SWHAT2(MAXREG), SX2(MAXREG)

LIST OF VARIABLES

ARG INTERMEDIATE CALCULATION VARIABLE
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND 
SIG() CALCULATION
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGION ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR 
EACH REGION, BASED ON MATERIALS DATA ONLY – MCHAT(1,L) = 
– DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, 
THE ESTIMATE FOR C
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION 
MEAN FOR EACH REGION
MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL 
DISTRIBUTION MEAN FOR EACH REGION
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL 
DATA SETS IN A REGION (Number of Points Per Region)
NUMREG NUMBER OF REGIONS OF INTEREST
SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL 
DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION 
VARIANCE FOR EACH REGION
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X 
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SUMX2 EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y 
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION 
(X = Ln S)

C INITIALIZE ARRAYS

DO 50 L = 1, MAXREG
MCHAT(1,L) = 0.0
MCHAT(2,L) = 0.0
MU(L) = 0.0
SIG(L) = 0.0
50 CONTINUE

C BEGIN CALCULATION FOR EACH REGION

DO 100 L = 1, NUMREG

MCHAT(1,L) = - DD(L)
MCHAT(2,L) = SQRT(SUHAT2(L))
SUMX2 = NPPR(L) * SX2(L)
ARG = SUMX2 + DELTA(L)

IF (DELTA(L) .EQ. 0.0) THEN
  THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED 
  USE THE ESTIMATE OF M 
  MU(L) = MCHAT(1,L)
ELSE
  UPDATE THE ESTIMATE OF M WITH MO USING DELTA 
  MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
ENDIF

IF (SIGMA2(L) .EQ. 0.0) THEN
  THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
USE WHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
SIG(L) = SQRT (WHAT2(L) / ARG)
ELSE
SIG(L) = SQRT (SIGMA2(L) / ARG)
ENDIF
IF (IOUT .EQ. i_)
T,HEN
WRITE(8,*), L, 'DD = ', DD(L), ' MCHAT1 = ',
& MCHAT1(L), SUHAT2(L), ' MCHAT2 = ' ,
& MCHAT2(L), NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
& SUMX2 = ', SUMX2(L), ' MO = ', MO(L), ' MU = ', MU(L),
& WHAT2 = ', WHAT2(L), ' SIGMA2 = ', SIGMA2(L),
& SIG = ', SIG(L)
ENDIF
100 CONTINUE
RETURN
END

C*************************************************
C SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND CO TO
C OBTAIN POSTERIOR RANGES ON M FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 10FEB88 FORMAT/COMMENTS: 12AUG91
C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
& RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LOWER LOWER BOUND OF INTERSECTION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
BOUND
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
FOR EACH REGION - MCHAT(1,L) = -DD(L), THE ESTIMATE
FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C  EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
C  NUMREG NUMBER OF REGIONS OF INTEREST
C  RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
NOT FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND I
RANGEM(2,L) IS THE UPPER BOUND
C  UPPER UPPER BOUND OF INTERSECTION
C
C  INITIALIZE VARIABLES

DO 50 L = 1, MAXREG
  RANGEM(1,L) = 0.0
  RANGEM(2,L) = 0.0
50 CONTINUE

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

DO 100 L = 1, NUMREG

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
  WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
ENDIF

IF (MPNT(L) .EQ. 1) THEN
  THERE IS A POINT PRIOR ON M — THIS OVERRIDE ALL OTHER
  INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
  RANGEM(1,L) = MZERO(1,L)
  RANGEM(2,L) = 0.0

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
  WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
  &
  WRITE(8,*) 'MZERO(2,L) = ', MZERO(2,L),
  &
  WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN
  THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT USE Mo
  RANGEM(1,L) = MZERO(1,L)
  RANGEM(2,L) = MZERO(2,L)

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
  &
  WRITE(8,*) 'MZERO(2,L) = ', MZERO(2,L),
  &
  WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
  &
  WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN
  THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO
  CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc
  LOWER = AMAX1(MZERO(1,L), MC(1,L))
  UPPER = MZERO(2,L)
  IF (UPPER .LT. LOWER) THEN
    WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN Mo AND Mc'
    CALL TRMNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
  &
  WRITE(8,*) 'MZERO(2,L) = ', MZERO(2,L),
  &
  WRITE(8,*) 'MC(1,L) = ', MC(1,L),
  &
  WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
  &
  WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
  &
  WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

7 - 152
C IF (MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2) THEN
C THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO C0 CONSTRAINT
C INTERSECT THESE TWO RANGES
C
LOWER = AMAX1(MZERO(1,L), MC(1,L))
UPPER = AMIN1(MZERO(2,L), MC(2,L))
IF (UPPER .LT. LOWER) THEN
   WRITE(8,*) 'ERROR: NO INTERSECTION BETWEEN MC AND MC'
   CALL TRMNAT
ELSE
   RANGEM(1,L) = LOWER
   RANGEM(2,L) = UPPER
ENDIF
ENDIF
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN
C THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO C0 CONSTRAINT
C INTERSECT THESE TWO RANGES
C
LOWER = AMAX1(MZERO(1,L), MC(1,L))
UPPER = AMIN1(MZERO(2,L), MC(2,L))
IF (UPPER .LT. LOWER) THEN
   WRITE(8,*) 'ERROR: NO INTERSECTION BETWEEN MC AND MC'
   CALL TRMNAT
ELSE
   RANGEM(1,L) = LOWER
   RANGEM(2,L) = UPPER
ENDIF
ENDIF

C IF (IOUT .EQ. 10) THEN
WRITE(8,*) 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
CALL TRMNAT
ENDIF

C RESTRICT RANGE TO BE NON-NEGATIVE
RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)
IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)
100 CONTINUE

C CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE
DO 300 L = 1, NUMREG
   IF ((MCHAT(1,L) .LT. RANGEM(1,L))
   & .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
   & WRITE(8,*) 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
   & 'ON m IN REGION ', L
300 CONTINUE

RETURN
END
LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LL EQUAL TO NUMREG FOR A SET OF CALCULATIONS
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT( ) 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MO( ) 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
MEAN FOR EACH REGION
MPNT( ) 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MZERO() FOR EACH REGION
MU( ) 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION MEAN FOR EACH REGION
MZERO( ) 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
IS UPPER BOUND
NNODAT Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM( ) 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND
SIG( ) 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
SIGMA2( ) 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
VARIANCE FOR EACH REGION

IF (IOUT .EQ. 10) WRITE(8,*)'NUMREG =', NUMREG

DO 100 L = 1, NNODAT
NUMREG = NUMREG + 1
LL = NUMREG
IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, 'NUMREG =', NUMREG,
& 'LL =', LL, 'MPNT( LL ) =', MPNT( LL )
IF ((MPNT( LL ) .EQ. 1) .OR. (MPNT( LL ) .EQ. 2)) THEN
POSTERIOR ON M IS SAME AS PRIOR ON M
RANGEM(1, LL ) = MZERO(1, LL )
RANGEM(2, LL ) = MZERO(2, LL )
MU( LL ) = MO( LL )
SIG( LL ) = SQRT( SIGMA2( LL ) )
IF (IOUT .EQ. 10) THEN
WRITE(8,*)'RANGEM(1, LL ) =', RANGEM(1, LL ),
& 'MZERO(1, LL ) =', MZERO(1, LL ),
& WRITE(8,*)'RANGEM(2, LL ) =', RANGEM(2, LL ),
& 'MZERO(2, LL ) =', MZERO(2, LL ),
& WRITE(8,*)'MU( LL ) =', MU( LL ), 'MO( LL ) =', MO( LL )
WRITE(8,*)'SIG( LL ) =', SIG( LL ), 'SIGMA2( LL ) =', SIGMA2( LL )
ENDIF

SPECIFY E(M) OF POSTERIOR FOR SAKE OF
CALCULATIONS IN SUBROUTINE EXPCTD

7 - 154
IF (RANGEM(2,LL) .EQ. 0.0) THEN
  MCHAT(I,LL) = RANGEM(I,LL)
  MU(LL) = RANGEM(I,LL)
  SIG(LL) = 0.0
ELSE
  MCHAT(I,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
ENDIF

IF (IOUT .EQ. I0) WRITE(8,*) 'MCHAT =', MCHAT(I,LL),
& ' MU = ', MU(LL), ' SIG = ', SIG(LL)
& CALL TRMNAT

ENDIF 100 CONTINUE

RETURN
END

C******************************************************************************

C SUBROUTINE PAREST CONTROLS THE CALCULATIONS FOR THE PARAMETER
C ESTIMATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89
C VERSION: MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
C MATRM V4.3, V4.4, V4.5

C Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG,
& ZROREG, RAND, NBND, STR, BIGK, BZERO, MM, 
& SBND)

C INPUTS: VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,C NBND, STR
C OUTPUTS: BIGK, BZERO, MM, SBND
C SUBPROGRAMS: FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB
C IMPLICIT NONE
C IMPLICIT NONE
C INTEGER MAXDAT, MAXREG
C PARAMETER (MAXDAT = 50, MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG
C REAL BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG), 
& MU(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
& RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG), 
& STR(MAXDAT, MAXREG), S22, Z2(MAXDAT)
C DOUBLE PRECISION RAND

C LIST OF VARIABLES
C BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C EACH REGION
C BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING S/N DATA SET
C IOUT OUTPUT DUMP CONTROLLER
C K VALUE OF K -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C L CONTROLS DO LOOP FOR EACH REGION
C MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED

7 - 155
SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION

1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST

2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS

TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET

1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET

NUMBER OF REGIONS OF INTEREST

2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER Bound AND RANGEM(2,L) IS THE UPPER Bound

RANDOM NUMBER SEED

1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()

1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION

ZEROREG Zero Region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP

BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA, Z = F(STR, NF, NBND, MM)

OBTAIN THE VALUES OF M FOR EACH REGION

IF (VARY .LE. 2) THEN

UNIFORM OR NO VARIATION IN M IS DESIRED

CALL FINDM (RAND, NUMREG, RANGEM, MM)

ELSE

NORMAL VARIATION IN M IS DESIRED

CALL FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)

ENDIF

TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

CALCULATE THE VALUES FOR K AND BETAo FROM THE SAMPLE MEAN AND VARIANCE

CALL KBETA (MEANZ, SZ2, K, BZERO)

CALCULATE THE VALUE OF K FOR EACH REGION WHERE A = K ** M

CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

CALCULATE STRESS TIE-POINTS

CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

WRITE RESULTS TO FILE

WRITE(7,900} NUMREG, BZERO

DO 200 L = ZROREG, NUMREG
C SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION BY SAMPLING OFF THE APPROPRIATE M RANGE

PROGRAMMER: L. NEWLIN

DATE: CODE: 7JUN88 COMMENTS: 13JUL89

VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

INPUTS: RAND, NUMREG, RANGEM
OUTPUTS: MM
SUBPROGRAMS: RANDOM, TRMNAT

IMPLICIT NONE

INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
PICK() 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM RAND RANDOM NUMBER SEED
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
X UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED OFF THE RANGE ON M

C INITIALIZE MM()

DO 50 L = 0, MAXREG
     MM(MAXREG) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS
DO 100 L = 1, NUMREG
    PICK(1) = 0.0
    PICK(2) = 0.0
    IF (RANGEM(2,L).EQ. 0.0) THEN
        WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L), 'MM(L) = ', MM(L)
        MM(L) = RANGEM(1,L)
        IF (IOUT.EQ.10) WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L), 'MM(L) = ', MM(L)
    ELSEIF (L.EQ.1) THEN
        CALL RANDOM(X, RAND)
        IF (IOUT.EQ.10) THEN
            WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L), 'MM(L) = ', MM(L)
        ENDIF
    ELSE
        IF (L.EQ.I) THEN
            CALL RANDOM(X, RAND)
            IF (IOUT.EQ.10) THEN
                WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L), 'MM(L) = ', MM(L)
            ENDIF
        ELSE
            IF (L.EQ.1) THEN
                CALL RANDOM(X, RAND)
                IF (IOUT.EQ.10) THEN
                    WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L), 'MM(L) = ', MM(L)
                ENDIF
            ENDIF
        ENDIF
    ENDIF
100 CONTINUE
    RETURN
END

C***********************************************************************
C
C SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C Random Number Generator, JPL Publication 85-98, JPL Document
C
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: IDEC87
C           V4.3, V4.4, V4.5
C***********************************************************************
C
C SUBROUTINE RANDOM (FRAC, RAND)
C IMPLICIT NONE
C COMMON IOUT
C INTEGER IOUT

7 - 158
REAL FRAC
DOUBLE PRECISION RANA, RANC, RAND, RANDIV, RANM, RANSUB, &
RANT, RANX

LIST OF VARIABLES
C UNIFORM (0, 1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
RANA CONSTANT FOR LG
RANC CONSTANT FOR LCG
RAND RANDOM NUMBER SEED
RANDIV INTERNAL CALCULATION
RANM CONSTANT FOR LG
RANSUB INTERNAL CALCULATION
RANT INTERNAL CALCULATION
RANX INTERNAL CALCULATION

C USING LCG RANDOM # GENERATOR
RANA = 671093.0
RANC = 7090885.0
RANM = 33554432.0
10 RANX = RANA * RAND + RANC
RANDIV = RANX / RANM
RANT = DINT(RANDIV)
RANSUB = RANT * RANM
RAND = RANX - RANSUB
FRAC = SNGL(RAND / RANM)

IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
IF (IOUT .EQ. 2) WRITE(8,*),'RANX =', RANX,
& ' RANDIV =', RANDIV,
& ' RANT =', RANT,
& ' RANSUB =', RANSUB,
& ' RAND =', RAND,
& ' FRAC =', FRAC
RETURN
END

C NOTES: IOUT=2 DUMPS TO SCREEN

C*********************************************************
C SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88 COMMENTS: 13FEB89
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
C INPUTS: RAND, NUMREG, MU, SIG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: NORMGN, TRMNAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG), &
SIG(MAXREG), X

7 - 159
DOUBLE PRECISION RAND

LIST OF VARIABLES

IOUT | OUTPUT DUMP CONTROLLER
L    | CONTROLS DO LOOP FOR EACH REGION
MAXREG | MAXIMUM NUMBER OF REGIONS ALLOWED
MM()  | 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
MU()  | 1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
NUMREG | NUMBER OF REGIONS OF INTEREST
PICK() | 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RAND | RANDOM NUMBER SEED
RANGEM()  | 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
         | FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
SIG()  | RANGEM(2,L) IS THE UPPER BOUND
X     | NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED

C INITIALIZE MM()
DO 50 L = 0, MAXREG
  MM(MAXREG) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS
DO 100 L = 1, NUMREG
  PICK(1) = 0.0
  PICK(2) = 0.0
  IF (RANGEM(2,L) .EQ. 0.0) THEN
    M IS SPECIFIED AS A POINT VALUE
    MM(L) = RANGEM(1,L)
    IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L), ' MM(L) = ', MM(L)
    ELSEIF (L .EQ. I) THEN
      SAMPLE ON EXISTING RANGE
      CALL NORMGN (RAND, MU(L), SIG(L), X)
      IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
      MM(L) = X
      IF (IOUT .EQ. 10) THEN
        WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L), ' RANGEM(2,L) = ', RANGEM(2,L)
        WRITE(8,*) 'L = ', L, ' X = ', X, ' MM(L) = ', MM(L)
        ENDIF
    ELSE
      SAMPLE ON ADJUSTED RANGE
      CALL NORMGN (RAND, MU(L), SIG(L), X)
      IF ((L .LT. PICK(1)) .OR. (L .GT. PICK(2))) GOTO 20
      MM(L) = X
      IF (IOUT .EQ. 10) THEN
        WRITE(8,*) 'L = ', L, ' MM(L-1) = ', MM(L-1), ' RANGEM(1,L) = ', RANGEM(1,L)
        WRITE(8,*) 'PICK(1) = ', PICK(1), ' PICK(2) = ', PICK(2)
        WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L), ' X = ', X, ' MM(L) = ', MM(L)
        ENDIF
      ELSE
        CALL TRMNAT
        ENDIF
    ELSE
      STOP PROGRAM
      WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION ', L
      CALL TRMNAT
  ENDIF
100 CONTINUE
CONTINUE
RETURN
END

SUBROUTINE NORMGN

SUBPROGRAM: NORMGN

IMPLICIT NONE
COMMON IOUT
DOUBLE PRECISION RAND, FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2
PARAMETER (PI = 3.1415926536)
INTEGER IOUT

LIST OF VARIABLES
FRAC UNIFORM(0,1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
MU MEAN OF NORMAL DISTRIBUTION
RAND RANDOM NUMBER SEED
SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION
X NORMAL RANDOM VARIATE
U1 UNIFORM RANDOM NUMBER U(0,1)
U2 UNIFORM RANDOM NUMBER U(0,1)
Z1 NORMAL RANDOM NUMBER ON N(0,1)
Z2 NORMAL RANDOM NUMBER ON N(0,1)

IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'FRAC = ', FRAC, ' IOUT = ', IOUT, ' RAND = ', RAND, ' MU = ', MU,'
SIGMA = ', SIGMA
CALL RANDOM (FRAC, RAND)
U1 = FRAC
CALL RANDOM (FRAC, RAND)
U2 = FRAC
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'U1 = ', U1,' U2 = ', U2
Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)
X = SIGMA * Z1 + MU
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'Z1 = ', Z1,' Z2 = ', Z2,' X = ', X
SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

PROGRAMMER: L. NEWLIN
DATE: CODE: 7JUN88 COMMENTS: 13JUL89


SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

INPUTS: NPTS, STR, NF, NUMREG, MM, NBND
OUTPUTS: NP, ZZ

IMPLICIT NONE

INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG
REAL MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
& STR(MAXDAT, MAXREG), ZZ(MAXDAT)

LIST OF VARIABLES

I CONTROLS DO LOOP FOR EACH DATA POINT
IOUT OUTPUT DUMP CONTROLLER
K CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
LL CONTROLS INNER DO LOOP FOR EACH REGION
MAXDAT MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM(*) 1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
MML EQUAL TO MM(L) FOR A SET OF CALCULATIONS
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
NPTS() 1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET
NUMREG NUMBER OF REGIONS OF INTEREST
STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
ZZ() 1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
Z = F(STR, NF, NBND, MM)

INITIALIZE VARIABLES
NP = 0
DO 50 I = 1, MAXDAT
ZZ(I) = 0.0
50 CONTINUE

BEGIN CALCULATIONS
DO 100 L = 1, NUMREG
MML = MM(L)
100 CONTINUE
IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =', MML, ' NPTS =', NPTS(L)
& DO 200 K = 1, NPTS(L)  
  NW = NP + 1  
  ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
& IF (IOUT .EQ. 10) WRITE(8,*)'K =', K, ' NP =', NP, ' NF =', NF(K,L), ' STR =', STR(K,L), ' ZZ =', ZZ(NP)
& DO 300 LL = 2, L  
  ZZ(NP) = ZZ(NP) + ALOG(NBND(LL-I)) * ((1.0 / MM(LL-I)) - (1.0 / MM(LL)))
& IF (IOUT .EQ. 10) WRITE(8,*)'LL =', LL, ' NBND(LL-1) =', NBND(LL-1), ' MM(LL-1) =', MM(LL-1), ' MM(LL) =', MM(LL), ' ZZ =', ZZ(NP)
 300 CONTINUE
200 CONTINUE
100 CONTINUE
RETURN
END

C************************************
C SUBROUTINE SMNVAR CALCULATES THE Sample Mean and Variance OF
C Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 24AUG87 COMMENTS: 13JUL89
C VERSION: MATCHR V5.3, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C V8.3, V8.4, V8.5
C SUBROUTINE SMNVAR (NP, ZZ, MEANZ, S2Z)
C INPUTS: NP, ZZ
C OUTPUTS: MEANZ, S2Z
C IMPLICIT NONE
C INTEGER MAXDAT
C PARAMETER (MAXDAT = 50)
C COMMON IOUT
C INTEGER I, IOUT, NP
C REAL MEANZ, S2Z, ZZ(MAXDAT)
C
C LIST OF VARIABLES
C I  CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C IOUT OUTPUT DUMP CONTROLLER
C MAXDAT MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C DATA SET
C S2Z SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C ZZ() 1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C Z = F(STR,NF,NBND,MM)
C
C INITIALIZE VARIABLES
MEANZ = 0.0
S2Z = 0.0
C CALCULATE THE MEAN OF ZZ(), MEANZ

DO 100 I = 1, NP
    MEANZ = MEANZ + ZZ(I)
    IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I, ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
100 CONTINUE
    MEANZ = MEANZ / FLOAT(NP)
    IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ

C CALCULATE THE VARIANCE OF ZZ(), SZ2

DO 200 I = 1, NP
    SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
    IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
200 CONTINUE
    SZ2 = SZ2 / FLOAT(NP - 1)
    IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

RETURN
END

C******************************************************************************

C SUBROUTINE KBETA CALCULATES k AND BETAO FROM THE SAMPLE MEAN AND VARIANCE
C OF Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87 COMMENTS: 13JUL89

C SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)
C INPUTS: MEANZ, SZ2
C OUTPUTS: K, BZERO
C IMPLICIT NONE
REAL PI
PARAMETER (PI = 3.1415926536)
COMMON IOUT
INTEGER IOUT
REAL BZERO, K, MEANZ, SZ, SZ2

LIST OF VARIABLES

BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE SPECIFIC MATERIAL S/N DATA SET
IOUT OUTPUT DUMP CONTROLLER
K VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
PI SELF EXPLANATORY CONSTANT
SZ SZ2 ** 0.5
SZ2 SAMPLE VARIANCE OF THE TRANSFORMED DATA,
Z = F(STR, NF, NBND, MM)

C PERFORM CALCULATIONS

SZ = SZ2 ** 0.5
\[ BZERO = \frac{\pi}{\left( \text{SZ} \ast \left( 6.0 \ast \text{0.5} \right) \right)} \]

\[ K = \text{MEANZ} \]

C DATA DUMP STATEMENTS

IF (IOUT .EQ. 10) THEN
    WRITE(8,*), 'SZ2 =', SZ2, ' Sz =', Sz
    WRITE(8,*), 'MEANZ =', MEANZ, ' K = ', K, ' BZERO = ', BZERO
ENDIF

RETURN
END

C******************************************************************************

SUBROUTINE FINDK CALCULATES THE VALUE OF K, WHERE A = K ** M FOR EACH REGION

PROGRAMMER: L. NEWLIN

DATE: 7JUN88

VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

INPUTS: BZERO, K, MM, NBND, NUMREG

OUTPUTS: BIGK

IMPLICIT NONE

INTEGER MAXREG
REAL GAMMA
PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL BIGK(0:MAXREG), BZERO, K, MM(0:MAXREG), NBND(0:MAXREG)

LIST OF VARIABLES

BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING SPECIFIC MATERIAL DATA BASE
GAMMA EULER'S CONSTANT
IOUT OUTPUT DUMP CONTROLLER
K VALUE OF K -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL DATA BASE
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS OF INTEREST

C INITIALIZE VARIABLES

   DO 50 L = 0, MAXREG
         BIGK(L) = 0.0
   50 CONTINUE

C CALCULATE K FOR REGION ONE

C******************************************************************************
BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)

WRITE(7,*) 'REGION: ', K =', BIGK(1)
IF (IOUT .EQ. 10) WRITE(8,*) 'BZERO =', BZERO, ' K =', K,
& ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)

C CALCULATE K FOR REMAINING REGIONS

DO 100 L = 2, NUMREG
  BIGK(L) = BIGK(L-1) * NBND(L-1)
  & WRITE(7,*) 'L =', L, ' NBND(L-1) =', NBND(L-1),
  & ' MM(L) =', MM(L), ' MM(L-1) =', MM(L-1),
  & ' BIGK(L) =', BIGK(L)
100 CONTINUE

RETURN
END

C******************************************************************************

C SUBROUTINE FINDSB CALCULATES THE REGION 'TIE-POINTS' -- THE STRESS VALUES
C WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE RANDOMLY
C SELECTED Ms, AND THE Ks CALCULATED FROM THE BETA AND K CHARACTERIZING
C SPECIFIC MATERIAL
C PROGRAMMER: L. NEWLIN
C DATE: 22DEC88
C VERSION: MATCHR V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C INPUTS: NUMREG, ZROREG, NBND, BIGK, MM
C OUTPUTS: SBND

C IMPLICIT NONE

INTEGER MAXREG
PARAMETER ( MAXREG = 3 )
COMMON IOUT
INTEGER IOUT, L, NUMREG, ZROREG
REAL
BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
& SBND(0:MAXREG)

LIST OF VARIABLES

BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS OF INTEREST
SBND() 1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()
ZROREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE = 0 -- ZERO REGION EXISTS, 1 - NO REGION

C INITIALIZE SBND()

DO 50 L = 0, MAXREG
  SBND(L) = 0.0
50 CONTINUE
50 CONTINUE
C CALCULATE SBND(0) IF ZROREG = 0
    IF (ZROREG .EQ. 0) THEN
        SBND(0) = BIGK(1) * NBND(0) ** (-1.0 / MM(1))
    ENDIF
C CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES
    DO 100 L = 1, NUMREG
        IF (NBND(L) .GE. 1.0E+36) THEN
            SBND(L) = 0.0
        ELSE
            SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
        ENDIF
    100 CONTINUE
RETURN
END

C******************************************************************************

C THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
C MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C TRANSFORM METHOD"
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 16MAR87   COMMENTS: 15SEP89
C VERSION: MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C          V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C          MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C          V4.3, V4.4, V4.5
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

SUBROUTINE WEIBGN (BETA, RAND, WEIB)
C INPUTS:  BETA, RAND
C OUTPUTS: WEIB
C SUBPROGRAMS: RANDOM
C
C IMPLICIT NONE
C COMMON IOUT
C INTEGER IOUT
C REAL  ARG, BETA, ETA, FRAC, WEIB
C DOUBLE PRECISION RAND

LIST OF VARIABLES
C ARG       INTERMEDIATE CALCULATION VARIABLE
C BETA      WEIBULL DISTRIBUTION SHAPE PARAMETER
C ETA       WEIBULL DISTRIBUTION LOCATION PARAMETER
C FRAC      UNIFORM (0,1) RANDOM VARIATE
C IOUT      OUTPUT DUMP CONTROLLER
C RAND      RANDOM NUMBER SEED
C WEIB      WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE

C CALCULATE CONSTRAINED ETA
ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))
C GENERATE WEIBULL RANDOM VARIATE
CALL RANDOM(FRAC, RAND)
ARG = -ALOG(1.0 - FRAC)
WEIB = ETA * ARG**(1.0/BETA)
IF (IOUT .EQ. 10) WRITE(8,*('BETA = ', BETA, ' ETA = ', ETA, & ' FRAC = ', FRAC, ' ARG = ', ARG, ' WEIB = ', WEIB))
RETURN
END

C*********************************************************

C SUBROUTINE KOMO CALCULATES K0 AND MO FOR THE ZERO REGION (NO DATA REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT AT ZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE ZERO.
C PROGRAMMER: L. NEWLIN
C DATE: 1AUG91
C VERSION: MATCHR V8.5 MATGRM V4.5
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK, & FACTR, NUMREG)

C INPUTS: SZERO, BIGK, MM, NBND, TRSBND, FACTR
C OUTPUTS: TRBIGK, MM, TRSBND

C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT, L, NUMREG
REAL BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG), SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)

LIST OF VARIABLES

BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
FACTR SCALE FACTOR = PHI * KRATIO * Z
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS
SCLK ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
SZERO STRESS TENSILE TEST POINT, SO
TRBIGK() 1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP SBOUND(0) < SO FOR EACH TRIAL
TRSBND() 1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()

BIGK(0) = SZERO
IF (TRSBND(0) .GT. SZERO) THEN
  SCLK = SZERO/TRSBND(0)
DO 100 L = 0, NUMREG

7 - 168
TRBIGK(L) = BIGK(L) * SCLK
TRSBND(L) = TRSBND(L) * SCLK

100 CONTINUE
ELSE
TRBIGK(0) = SZERO/FACTR
MM(0) = MM(1) * ((ALOG (BIGK(1)) - ALOG (TRSBND(0)))
+ ALOG (FACTR)) / (ALOG (SZERO) - ALOG (TRSBND(0)))
ENDIF

C IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'SZERO = ', SZERO,
 WRITE(8,*), 'FACTOR = ', FACTR,
 WRITE(8,*), 'MM1 = ', MM(1), ' MM0 = ', MM(0)
ENDIF

RETURN
END

C******************************************************************************

FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
PROGRAMMER: L. NEWLIN
DATE: 10FEB89
VERSION: MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO)

C INPUTS: S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C OUTPUTS: GTLIFE
C
IMPLICIT NONE

INTEGER IOUT, L, MAXREG, NUMREG, ZROREG
PARAMETER (MAXREG = 3)
COMMON IOUT
REAL  GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
&        MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

LIST OF VARIABLES

GETLIF VALUE TO BE ASSIGNED TO GTLIFE -- CYCLES TO FAILURE FOR
THE REQUIRED STRESS LEVEL
IOUT OUTPUT DUMP CONTROLLER
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LNA() 1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
LNZ NORMAL (0, PVAR) GENERATED RANDOM VARIATE
LPHIM() 1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
PHI IS A WEIBULL(BETA0, ETA0) GENERATED RANDOM VARIATE
MAXREG Maximum number of regions allowed
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
S VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO
FAILRE) IS REQUIRED
SBND() 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
CONTAINED IN NBND()
SZERO STRESS TENSILE POINT, SO
TEMP TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
FLows
C ZOREG Zero Region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO REGION

GETLIF = 0.0
C CALCULATE CYCLES TO FAILURE
IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
GETLIF = 1.0
ELSE
DO 100 L = ZROREG, NUMREG
IF (S(L) .GT. SBND(L)) THEN
TEMP = LNA(L) + LPHIM(L) + MM(L) * ( - ALOG(S)
& + ALOG (KRATIO) + LN2)
IF (TEMP .GT. 86.0) THEN
TEMP = 86.0
ENDIF
GETLIF = EXP (TEMP)
GOTO 150
ENDIF
100 CONTINUE
ENDIF
150 CONTINUE
GTLIFE = GETLIF
RETURN
END

C***-----------------------------------------------------------------

C SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C M FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: 10FEB88
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

C SUBROUTINE SORTM (ALLM, NUMREG, NUM)
C
C INPUTS:  ALLM, NUMREG, NUM
C OUTPUTS:  ALLM
C
C IMPLICIT NONE
COMMON IOUT
INTEGER I, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG
PARAMETER (MAXMM = 20001, MAXREG = 3)
LOGICAL INORDR
REAL ALLM(MAXMM, MAXREG), TEMP

LIST OF VARIABLES
ALLM() 2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
INC CONTROLS INSERTION POINTER
INORDR FLAG TO INDICATE WHETHER SORT IS FINISHED
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXMM MAXIMUM NUMBER OF M'S TO BE SORTED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED

7 - 170
C NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C NUMBER OF REGIONS OF INTEREST
C TEMPORARY SORTING VARIABLE

DO 400 L = 1, NUMREG
  INC = NUM
  IF (INC .GT. 1) THEN
    INC = INC / 2
  ELSE
    INORDR = .TRUE.
    DO 300 I = 1, (NUM - INC)
       IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN
          TEMP = ALLM(I,L)
          ALLM(I,L) = ALLM(I + INC, L)
          ALLM(I + INC, L) = TEMP
          INORDR = .FALSE.
       ENDIF
  300 CONTINUE
  IF (.NOT. INORDR) GOTO 20
  GOTO 10
  ENDIF
400 CONTINUE
RETURN
END

C***FUNCTION RAINF3***
C FUNCTION RAINF3 CALCULATES THE TIME (in missions) TO FAILURE FOR
C THE GIVEN STRAIN-TIME HISTORY
C
C PROGRAMMER: L. SHIRAISHI, L. NEWLIN
C DATE: 27MAR90
C VERSION: 1.1 (BLDLCF V3.1, V3.2, V3.3, V3.4 MATHR V8.4, V8.5)
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

FUNCTION RAINF3(SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, &
KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO)
C INPUTS: SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO,
C LNZ, SBND, ZROREG, NUMREG, SZERO
C OUTPUTS: RAINF3
C
C IMPLICIT NONE
COMMON IOUT
INTEGER MAXR G, MAXM
PARAMETER (MAXREG = 3, MAXM = 50)
INTEGER I, INDEX(MAXM), IOUT, J, JMAX, K, M, N, NEWTOT, NUMREG, &
ZROREG
REAL CHKFT, E(MAXM), GTRLIFE, INVLIF(MAXM), KRATIO, &
LIFE(MAXM), LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), &
MM(0:MAXREG), PERIOD, RAINF3, S(MAXM), SBND(0:MAXREG), &
SEFF(MAXM), SEFM(2, MAXM), SEFMX, SP(MAXM), &
SRANGE(MAXM), SUMDAM, SZERO, TESTI(MAXM), TEST2(MAXM), &
TRUNC, WEXP

7 - 171
LIST OF VARIABLES

RAINF3  TIME TO FAILURE FOR THE GIVEN TIME HISTORY

input variables:

SEFF(M)  EFFECTIVE STRAINS BEFORE FILTERING/RAINFLOW
M  TOTAL NUMBER OF STRAIN DATA POINTS PER PERIOD
TRUNC  VALUE USED TO FILTER OUT NOISE
PERIOD  TIME IN SECONDS FOR ONE PERIOD
WEXP  WALKER EXPONENT

intermediate variables:

MAXM  MAXIMUM NUMBER OF POINTS ALLOWED IN STRAIN-TIME HISTORY ARRAYS
SEFMAX  LARGEST EFFECTIVE STRAIN
INDEX(MAXM), TEST1(MAXM), TEST2(MAXM)  COUNTERS FOR VARIOUS DO LOOPS
I,J,K  INTERMEDIATE CALCULATION ARRAYS USED DURING FILTERING
S(NEWTOT)  FILTERED EFFECTIVE STRAINS
B()  HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
N  NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
SEFFM(2,N)  EFFECTIVE STRAINS AFTER RESEQUENCING/FILTERING/RAINFLOW
SEFFM(1,I) = sigma max,eff,i
SEFFM(2,I) = sigma min,eff,i
SRANGE(I) = EQUIVALENT STRAIN RANGE FOR CYCLE I
LIFE(N)  REAL FUNCTION THAT CALCULATES FATIGUE LIFE FOR A GIVEN STRAIN
LIFE(I) = CALCULATED LIFE FOR STRAIN LEVEL SRANGE(I)
INVLIF(I) = I/LIFE(I); DAMAGE FRACTION
SUMDAM  SUM OF ALL THE DAMAGE FRACTIONS
CHKFT  DUMMY VARIABLE USED TO PRINT OUT RAINF3 RESULT

output data

if (lout.eq.20) then
write(8,*)' rainf3 inputs'
write(8,*)'m :',m,' period: ',period
write(8,*)'wexp : ', wexp
write(8,*)'numreg :',numreg,'zroreg :',zroreg
write(8,*)'szero :',szero,'kratio :',kratio,'lnz :',lnz
write(8,*)'lna(i), mm(i), iphim(1), sbnd(i)
write(8,*)'lna(i), mm(i), iphim(i), sbnd(i), i=zroreg,numreg)
endif

C INITIALIZE ARRAYS

DO 50 I = 1, MAXM
SP(I) = 0.0
S(I) = 0.0

7 - 172
E(I) = 0.0
SEFFM(1,1) = 0.0
SEFFM(2,1) = 0.0
SRANGE(I) = 0.0
LIFE(I) = 0.0
INVLIF(I) = 0.0
INDEX(I) = 0
TEST1(I) = 0.0
TEST2(I) = 0.0
50 CONTINUE

C******************************************************** BEGIN RESEQUENCE *********************************************************
C RESEQUENCE effective strains (needed for rainflow analysis);
C largest effective strain is placed at beginning and end of SP(M+1)
C
C find SEFMAX, the largest sigmэфф, and JMAX, its location within SEFF(M)
SEFMAX = -1.0E+20
DO 200 I=1,M
  IF (SEFF(I) .GT. SEFMAX ) THEN
    SEFMAX = SEFF(I)
    JMAX = I
  ENDIF
200 CONTINUE
C assign all points from JMAX out, to the beginning of SP()
DO 210 I = i, M-JMAX+1
  J = JMAX-I + I
  SP(I) = SEFF(J)
210 CONTINUE
C assign points before JMAX to the end of SP()
J = 0
DO 220 I = M-JMAX+2, M
  J = J + 1
  SP(I) = SEFF(J)
220 CONTINUE
C SP(M+1) = SEFF(JMAX)
if (iout.eq.20) then
  write(8,*) 'sefmax, ', sefmax, ', jmax, ', jmax
  write(8,*) 'sp(m+1): ', (sp(i),i=1,m+1)
endif

C******************************************************** END RESEQUENCE *********************************************************
C******************************************************** BEGIN FILTER *********************************************************
C FILTER the resequenced effective strains, leaving only peaks and valleys
C (excursions larger than TRUNC are deleted during rainflow counting) in
C SP(NEWTOT), where NEWTOT is the new number of points
C
DO 300 I = 2, M
  TEST1(I) = SP(I-1) - SP(I)
  TEST2(I) = TEST1(I) * (SP(I) - SP(I+1))
300 CONTINUE
C if (iout .eq. 20) then
  do 305 i = 2, m
    write(8,*) 'test1 = ', test1(i), ' test2 = ', test2(i)
  305 continue
C endif
K = 1
INDEX(K) = 1
DO 310 I = 2, M
  IF ((TEST1(I) .NE. 0) .AND. (TEST2(I) .LT. 0)) THEN
    K = K + 1
    INDEX(K) = I
  ENDIF
310 CONTINUE
NEWTOT = K + 1
INDEX(NEWTOT) = M + 1
DO 320 I = I, NEWTOT
  K = INDEX(I)
  S(I) = SP(K)
320 CONTINUE

if (iout.eq.20) then
  write(8,*),'newtot:',newtot
  write(8,*),'s(newtot):',(s(i),i=l,newtot)
endif

C******************************************************** END FILTER *********************************************************
C******************************************************** BEGIN RAINFLOW *************************************************************
C RAINFLOW ANALYSIS to identify cycles within effective strain data, S(NEWTOT);
C places each cycle's max and min values into SEFFM(2,N)
C
C counters: I counts # of cycles found, J counts how many S()'s counted,
C K accumulates unmatched points

I = 0
J = 0
K = 0
400 CONTINUE
  J = J+1
  K = K+I
  if (J .GT. NEWTOT ) GOTO 499
  E(K) = S(J)
  IF ( K .LT. 3 ) GOTO 400
  IF ( ABS(E(K) - E(K-I)) .LT. ABS(E(K-I) - E(K-2)) ) GOTO 400
  C cycle is large enough to save
  I = I+1
  SEFFM(I,I) = AMAX1( E(K-1), E(K-2) )
  SEFFM(2,I) = AMINI( E(K-1), E(K-2) )
  ENDIF
  C discard points K-1 and K-2, and decrement the counter of unmatched points
  E(K-2) = E(K)
  K = K-2
  GOTO 410
499 CONTINUE
C N equals the final number of cycles found
N = I
if (iout.eq.20) then
  write(8,*),'N :',n
  write(8,*) 'seffm(2,n) : '.
  do 12 i=1,n
    write(8,*) seffm(1,i), seffm(2,i)
 12 continue
endif

IF (N .EQ. 0) THEN
  C truncation filter value too large -- no cycles left
  SUMDAM = 1.0E-36
  GOTO 710
ENDIF

C******************************************************** END RAINFLOW *************************************************************
C calculate equivalent strain range
C
DO 500 I=1,N
  SRANGE(I) = (SEFFM(I,I) - SEFFM(2,I)) &
    / ((SEFFM(I,I) - SEFFM(2,I)) / (2.0 * SEFFM(I,I))) ** (WEXP - 1.0)
500 CONTINUE

if (iout.eq.20) write(8,*),'srange(n) :',(srange(i),i=1,n)
if (iout.eq.25) write(8,*),'srange(i),i=1,n,','
& \exp(lphim(1)/mm(1))

C calculate lives and damage fractions: LIFE(N) and INVLIF(N)
C
DO 600 I=1,N
  LIFE(I) = GTLIFE (SRANGE(I), MM, LNA, LPHIM, KRATIO, LNZ,
  & SBND, ZROREG, NUMREG, SZERO)
600 CONTINUE
DO 650 I=1,N
  INVLIF(I) = 1.0 / LIFE(I)
650 CONTINUE
if (iout.eq.20) then
  do 14 i=1,N
    write(8,*),'life(n):',life(i),',
  invlif(n):',invlif(i)
14 continue
endif
C Miner's Rule -- sum the damage fractions
C
SUMDAM = 0.0
DO 700 I=1,N
  SUMDAM = SUMDAM + INVLIF(I)
700 CONTINUE
710 CONTINUE
if (iout.eq.20) write(8,*),'sumdam:',sumdam
C calculate fatigue life (time to failure)
C
RAINF3 = PERIOD / SUMDAM
if (iout.eq.15) then
  chkft=period/sumdam
  write(8,*),'rainf3 life',chkft
  write(8,*)
endif

RETURN
END
### 7.2.5 Program BLDLCF V3.4B1.3 Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDLCF</td>
<td>7-178</td>
</tr>
<tr>
<td>BLDLIF</td>
<td>7-187</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-189</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-191</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-191</td>
</tr>
<tr>
<td>GAM</td>
<td>7-192</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-192</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-198</td>
</tr>
<tr>
<td>INIT</td>
<td>7-198</td>
</tr>
<tr>
<td>RCE</td>
<td>7-200</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-206</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-207</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-210</td>
</tr>
<tr>
<td>INTRVRL</td>
<td>7-212</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-215</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-216</td>
</tr>
<tr>
<td>ADDRREG</td>
<td>7-220</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-222</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-223</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-224</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-226</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-228</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-230</td>
</tr>
<tr>
<td>EXPB</td>
<td>7-232</td>
</tr>
<tr>
<td>PEB</td>
<td>7-233</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-235</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-236</td>
</tr>
<tr>
<td>PICRES</td>
<td>7-237</td>
</tr>
<tr>
<td>MREGR</td>
<td>7-238</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-240</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-241</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-242</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-243</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-244</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-245</td>
</tr>
<tr>
<td>WORSTN</td>
<td>7-246</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-248</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-249</td>
</tr>
<tr>
<td>RAINF3</td>
<td>7-250</td>
</tr>
</tbody>
</table>

BLDLCF Version 3.4B1.3
Program BLDLCF V3.4B1.3 Listing Temporal Order

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDLCF</td>
<td>7-178</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-192</td>
</tr>
<tr>
<td>INIT</td>
<td>7-198</td>
</tr>
<tr>
<td>RCE</td>
<td>7-200</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-206</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-207</td>
</tr>
<tr>
<td>EXPB</td>
<td>7-232</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-244</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-245</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-191</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-235</td>
</tr>
<tr>
<td>PEB</td>
<td>7-233</td>
</tr>
<tr>
<td>PICRES</td>
<td>7-237</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-235</td>
</tr>
<tr>
<td>MREGR</td>
<td>7-238</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-240</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-241</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-242</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-243</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-244</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-191</td>
</tr>
<tr>
<td>GAM</td>
<td>7-192</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-235</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-236</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-235</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-191</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-235</td>
</tr>
<tr>
<td>WORSTN</td>
<td>7-235</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-235</td>
</tr>
<tr>
<td>BLDLIF</td>
<td>7-187</td>
</tr>
<tr>
<td>RAINF3</td>
<td>7-250</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-248</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-189</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-249</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-224</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-240</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-241</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-242</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-243</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-244</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-245</td>
</tr>
</tbody>
</table>
C PROGRAM BLDLCF CONTROLS THE FLOW OF LOGIC OF THE LOW CYCLE
C FATIGUE ANALYSIS OF THE TURBINE BLADE FOIL PROBLEM
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 5FEB92 COMMENTS: 17APR92
C
VERSION: 3.4BI.3 (MATCHR VBI.3, RAINF3 V1.1, INSORT V2.1)
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
C
C*************************************************************************
C*************************************************************************
C*************************************************************************
C*************************************************************************
C*************************************************************************
C*************************************************************************

PROGRAM BLDLCF
C SUBPROGRAMS: INFAGG, PEB, PRYRV, BETAGN, NORMGN, WORSTN,
C TRMNAT, BLDLIF, INSORT, SORTM, EXPTCD
C 7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
C NOTE: 5 & 6 ARE OPENED IN 'INFAGG'
C IMPLICIT NONE

INTEGER MAXBLF, MAXDAT, MAXLIF, MAXM, MAXMM, MAXREG
PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLIF = 10000,
& MAXM = 50, MAXMM = 20001, MAXREG = 3)
COMMON I OUT
INTEGER BLFPOS(MAXBLF), I, IOUT, J, K, L, MCOUNT, MID,
& MPROC, NLIFE, NHYPER, NLIFE, NLIFET, NMED,
& NPTS(MAXREG), NSYM, NTIME, NUMREG, TSEBI, VARY,
& ZROREG
DOUBLE PRECISION RAND
REAL ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1, BLDLIF,
& BLFPER(MAXBFL), BZERO, DUM, EBEND, EBENDA, EBENDB,
& ENNOM, EPSL, EPSW, ETHNOM(MAXM), FAA, FAB, FAC, FACTR,
& FAD, FAE, FAF, FAERRM, FAERRS, FDIA, FDIB, FDIC, FDID,
& FTU,
& FTY,
& HGAS, HGASA, HGASR, HGASRI, HGASR2, HGAST, HGAST1, HGAST2,
& HGASTR, HGASTR2, HGASTR3, HGASTR4, HGASTR5, HTCLE,
& ID1, ID2, ID3, ID4, ID5, ID6, ID7, ID8, ID9, ID10,
& ID11, ID12, ID13, ID14, ID15, ID16, ID17, ID18,
& ID19, ID20, ID21, ID22, ID23, ID24, ID25, ID26,
& ID27, ID28, ID29, ID30, ID31, ID32, ID33, ID34,
& ID35, ID36, ID37, ID38, ID39, ID40, ID41, ID42,
& ID43, ID44, ID45, ID46, ID47, ID48, ID49, ID50,
& ID51, ID52, ID53, ID54, ID55, ID56, ID57, ID58,
& ID59, ID60, ID61, ID62, ID63, ID64, ID65, ID66,
& ID67, ID68, ID69, ID70, ID71, ID72, ID73, ID74,
& ID75, ID76, ID77, ID78, ID79, ID80, ID81, ID82,
& ID83, ID84, ID85, ID86, ID87, ID88, ID89, ID90,
& ID91, ID92, ID93, ID94, ID95, ID96, ID97, ID98,
& ID99, ID100, ID101, ID102, ID103, ID104, ID105,
& ID106, ID107, ID108, ID109, ID110, ID111, ID112,
& ID113, ID114, ID115, ID116, ID117, ID118, ID119,
& ID120, ID121, ID122, ID123, ID124, ID125, ID126,
& ID127, ID128, ID129, ID130, ID131, ID132, ID133,
& ID134, ID135, ID136, ID137, ID138, ID139, ID140,
& ID141, ID142, ID143, ID144, ID145, ID146, ID147,
& ID148, ID149, ID150, ID151, ID152, ID153, ID154,
& ID155, ID156, ID157, ID158, ID159, ID160, ID161,
& ID162, ID163, ID164, ID165, ID166, ID167, ID168,
& ID169, ID170, ID171, ID172, ID173, ID174, ID175,
& ID176, ID177, ID178, ID179, ID180, ID181, ID182,
& ID183, ID184, ID185, ID186, ID187, ID188, ID189,
& ID190, ID191, ID192, ID193, ID194, ID195, ID196,
& ID197, ID198, ID199, ID200, ID201, ID202, ID203,
& ID204, ID205, ID206, ID207, ID208, ID209, ID210,
& ID211, ID212, ID213, ID214, ID215, ID216, ID217,
& ID218, ID219, ID220, ID221, ID222, ID223, ID224,
& ID225, ID226, ID227, ID228, ID229, ID230, ID231,
& ID232, ID233, ID234, ID235, ID236, ID237, ID238,
& ID239, ID240, ID241, ID242, ID243, ID244, ID245,
& ID246, ID247, ID248, ID249, ID250, ID251, ID252,
& ID253, ID254, ID255, ID256, ID257, ID258, ID259,
& ID260, ID261, ID262, ID263, ID264, ID265, ID266,
& ID267, ID268, ID269, ID270, ID271, ID272, ID273,
& ID274, ID275, ID276, ID277, ID278, ID279, ID280,
& ID281, ID282, ID283, ID284, ID285, ID286, ID287,
& ID288, ID289, ID290, ID291, ID292, ID293, ID294,
& ID295, ID296, ID297, ID298, ID299, ID300, ID301,
& ID302, ID303, ID304, ID305, ID306, ID307, ID308,
& ID309, ID310, ID311, ID312, ID313, ID314, ID315,
& ID316, ID317, ID318, ID319, ID320, ID321, ID322,
& ID323, ID324, ID325, ID326, ID327, ID328, ID329,
& ID330, ID331, ID332, ID333, ID334, ID335, ID336,
& ID337, ID338, ID339, ID340, ID341, ID342, ID343,
& ID344, ID345, ID346, ID347, ID348, ID349, ID350,
& ID351, ID352, ID353, ID354, ID355, ID356, ID357,
& ID358, ID359, ID360, ID361, ID362, ID363, ID364,
& ID365, ID366, ID367, ID368, ID369, ID370, ID371,
& ID372, ID373, ID374, ID375, ID376, ID377, ID378,
& ID379, ID380, ID381, ID382, ID383, ID384, ID385,
& ID386, ID387, ID388, ID389, ID390, ID391, ID392,
& ID393, ID394, ID395, ID396, ID397, ID398, ID399,
& ID400, ID401, ID402, ID403, ID404, ID405, ID406,
& ID407, ID408, ID409, ID410, ID411, ID412, ID413,
& ID414, ID415, ID416, ID417, ID418, ID419, ID420,
WRITE(8,*) 'OUTER LOOP SIZE =', NHYPER
READ(1,*) NSYM
WRITE(8,*) 'SYMMETRY NUMBER =', NSYM
READ(1,*) VARY
WRITE(8,*) 'TYPE OF S/N VARIATION DESIRED', (4 - BOOTSTRAP) = 'VARY
READ(1,*) NMED
WRITE(8,*) 'NORMAL MEDIAN CURVE (0 - NO, 1 - YES) =', NMED
READ(1,*) MPROC
WRITE(8,*) 'MATERIALS PROCESS VARIATION DESIRED', (0 - NO, 1 - YES) = ', MPROC

IF (VARY .NE. 4) THEN
  WRITE(8,*) 'ERROR: INVALID TYPE OF S/N VARIATION DESIRED'
  CALL TRMNAT
ENDIF
IF ((NMED .NE. 0) .AND. (NMED .NE. i)) THEN
  WRITE(8,*) 'ERROR: INVALID RESPONSE TO NORMAL MEDIAN', 'CURVE QUESTION'
  CALL TRMNAT
ENDIF
IF ((MPROC .LT. 0) .OR. (MPROC .GT. i)) THEN
  WRITE(8,*) 'ERROR: INVALID TYPE OF MATERIALS PROCESS VARIATION DESIRED'
  CALL TRMNAT
ENDIF
READ(I,*) NBLIFE
IF (NBLIFE .GT. 0) READ(I,*) (BLFPER(J), J = i, NBLIFE)

C ** READ DATA FROM BLDLCO
READ(1,*) HGASA, HGASB, HGASR1, HGASR2, HGAST1, HGAST2,
& TGASA, TGASB, TGASR1, TGASR2, TGAST1, TGAST2,
& SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
& LAMGA, LAMGB, LAMGR1, LAMGR2, LAMGT1, LAMGT2,
& TSUBI, SPEEDM, SPEEDS,
& FAERRM, FAERRS, TSTMU, TSTSIG,
& FDERRM, FDERRS,
& EBENDA, EBENDB, LAMPA, LAMPB,
& MANALA, MANALB, LAMAA, LAMAB,
& TANALA, TANALB, LAMDA, LAMDB,
& LAMTA, LAMTMB
READ(1,*) EMNOM, NOMSPD, PERIOD, TRUNC, NTIME, WEXP
READ(1,*) FAA, FAB, FAC, FAD, FAE, FAF,
& FDIA, FDIB, FD1C, FD1D, FD1E, FD1F,
& FD2A, FD2B,
& IF (NTIME .GT. MAXM) THEN
  WRITE(8,*) 'ERROR: STRAIN-TIME HISTORY TOO LARGE'
  CALL TRMNAT
ENDIF
DO 20 I = i, (NTIME - I)
  READ(I,*) RPM(I), ETHNOM(I)
20 CONTINUE

C ** ECHO DATA TO BLDLCO
WRITE(3,900) HGASA, HGASB, HGASR1, HGASR2, HGAST1, HGAST2,
& TGASA, TGASB, TGASR1, TGASR2, TGAST1, TGAST2,
& SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
& LAMGA, LAMGB, LAMGR1, LAMGR2, LAMGT1, LAMGT2,
& TSUBI, SPEEDM, SPEEDS,
& FAERRM, FAERRS, TSTMU, TSTSIG,
& FDERRM, FDERRS,
& EBENDA, EBENDB, LAMPA, LAMPB,
& MANALA, MANALB, LAMAA, LAMAB,
& TANALA, TANALB, LAMDA, LAMDB,
& LAMTA, LAMTMB, EXP(LAMDA), EXP(LAMDB), EXP(LAMTA), EXP(LAMTMB)
WRITE(3,905) FAA, FAB, FAC, FAD, FAE, FAF,
& FDIA, FD1B, FD1C, FD1D, FD1E, FD1F,
& FD2A, FD2B,
& FD3A, FD3B
DO 25 I = 1, (NTIME - 1)
   WRITE(3,908) RPM(I), ETHNOM(I)
25 CONTINUE

C ** CALL INFAGG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT
C OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
C
CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZROREG, NUMREG,
& NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR,
& MEDMB, MEDKB, RESID)
IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)

C ** INITIALIZE VARIABLES
DO 35 K = 1, MAXLIF
   LIFEW(K) = 1.0E+36
   LIFE(L) = 1.0E+36
35 CONTINUE
NLIFET = NHYPER * NLIFE

C ** OUTER LOOP -- THIS LOOP SAMPLES HYPER-PARAMETER SETS
DO 150 K = 1, NHYPER

C ** CALL PRYRV TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS
CALL PRYRV (RAN, HGASR1, HGASR2, HGAST1, HGAST2, HGASR, HGAST)
CALL PRYRV (RAN, SLOPR1, SLOPR2, SLOPT1, SLOPT2, SLOPR, SLOPT)
CALL PRYRV (RAN, LAMGR1, LAMGR2, LAMGT1, LAMGT2, LAMGR, LAMGT)
CALL PRYRV (RAN, MANAL, MANALB, TANAL, TANALB)

C ** CALL PEB TO PERFORM THE BOOTSTRAPPING ASPECT OF THE
C MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL PEB (NPTS, NUMREG, ZROREG, RAND, NBND, STR, MEDMB,
& MEDKB, RESID, BIGK, BZERO, MM, SBND)

C ** OBTAIN MATERIALS PROCESS VARIATION PARAMETERS IF DESIRED
CALL NORMGN (RAND, 0.0, PSIG, LNZ)
IF (MPROC .EQ. 1) THEN
   Z = EXP (LNZ)
ELSE
   KRATIO = 1.0
   Z = 1.0
   LNZ = 0.0
ENDIF
MCOUNT = MCOUNT + 1
DO 175 L = 1, NUMREG
   ALLM(MCOUNT, L) = MM(L)
175 CONTINUE

C ** INNER LOOP -- THIS LOOP GENERATES BLADE FAILURE TIMES
DO 200 I = 1, NLIFE

C ** INITILIZE S/N CURVE PARAMETERS
DO 225 L = 0, MAXREG
   LNA(L) = 0.0
   LPHIM(L) = 0.0
   TRSBND(L) = 0.0
225 CONTINUE

C ** SELECT DRIVERS FOR CALCULATING LIFE
CALL BETAGN (RAND, HGASR, HGAST, HGASA, HGASB, HGAS)
CALL BETAGN (RAND, TGASR, TGAST, TGASA, TGASB, TGAS)
CALL BETAGN (RAND, SLOPR, SLOPE, SLOPEA, SLOPEB, SLOPE)
CALL BETAGN (RAND, LAMGR, LAMGT, LAMGA, LAMGB, LAMG)
CALL NORMGN (RAND, SPEEDM, SPEEDS, SPEED)
CALL NORMGN (RAND, FAERRM, FAERRS, MODER1)
CALL NORMGN (RAND, TSTMU, TSTSIG, TSTART)
CALL NORMGN (RAND, FDERRM, FDERRS, MODER2)
CALL PVRV (RAND, EBENDA, EBENDB, LAMPA, LAMPB, EBEND, LAMP)
CALL PVRV (RAND, LAMDA, LAMAB, LAMA, LAMAB, LAMA, DUM)
CALL PVRV (RAND, LAMDA, LAMAB, LAMA, LAMAB, LAMA, LAMMT)
LAMDA = EXP (LAMDA)
LAMTM = EXP (LAMTM)
CALL WORSTN (RAND, NSYM, BZERO, MM, EPSW, EPSL)
IF ((VARY .EQ. 0) .OR. (VARY .EQ. 4)) THEN
   PHI = 1.0
   IF (IOUT .EQ. 15) THEN
      WRITE(8,*) 'HGAS = ', HGAS, ' TGAS = ', TGAS
      WRITE(8,*) 'SLOPE = ', SLOPE, ' LAMG = ', LAMGA
      WRITE(8,*) 'LAMP = ', LAMP, ' EBEND = ', EBEND
      WRITE(8,*) 'SPEED = ', SPEED, ' LAMDA = ', LAMDA
      WRITE(8,*) 'LAMTM = ', LAMTM, ' PHI = ', PHI
      WRITE(8,*) 'MANAL = ', MANAL, ' TANAL = ', TANAL
      WRITE(8,*) 'TSTART = ', TSTART, ' MODER1 = ', MODER1,
      ' MODER2 = ', MODER2
      WRITE(8,*) 'EPSW = ', EPSW, ' EPSL = ', EPSL
   ENDIF
C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS
FACTR = PHI * KRATIO * Z
DO 235 L = ZROREG, NUMREG
   TRSBND(L) = FACTR * SBND(L)
   TRBIGK(L) = BIGK(L)
CONTINUE
TRSBND(0) = SBND(0)
IF (ZROREG .EQ. 0) CALL KOMO (SZERO, BIGK, MM, NBND,
   TRSBND, TRBIGK, FACTR, NUMREG)
DO 250 L = ZROREG, NUMREG
   LNA(L) = MM(L) * ALOG(TRBIGK(L))
   LPHIM(L) = MM(L) * ALOG(PHI)
   TRSBND(L) = SBND(L) * PHI * KRATIO * Z
   IF (IOUT .EQ. 15) THEN
      WRITE(8,*) 'L = ', L, ' MM = ', MM(L), ' BIGK = ', TRBIGK(L)
      WRITE(8,*) 'LNA = ', LNA(L), ' PHI = ', PHI
      WRITE(8,*) 'LPHIM = ', LPHIM(L), ' SBND = ', SBND(L)
      WRITE(8,*) 'KRATIO = ', KRATIO, ' Z = ', Z
      WRITE(8,*) 'TRSBND = ', TRSBND(L), ' FACTR = ', FACTR
   ENDIF
CONTINUE
C ** CALL BLIDLIF TO OBTAIN BLADE LCF LIFE
NEWLIF = LAMDA * LAMTM * BLIDLIF (TGAS, HGAS, FAA, FAB, FAC,
   FAD, FAE, FAF, MODER1, RPM, TSUBI, SPEED, SLOPE,
   TSTART, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
   MODER2, FD2A, FD2B, FD3A, FD3B, NMODT, EMMOB, LAMP,
   LAMMT, TRUNC, PERIOD, WEXP, MM, LMA, LPHIM,
   ZROREG, NUMREG, NZERO, TRSBND, LAMDA, LAMTM
LIFW = EPSW * NEWLIF
LIFL = EPSL * NEWLIF
IF (IOUT .EQ. 15) THEN
   WRITE(8,*) 'NEWLIF = ', NEWLIF
   WRITE(8,*) 'LIFW = ', LIFW, ' LIFL = ', LIFL
ENDIF
IF (NLIFET .GE. 100) THEN
    CALL INSORT (LIFW, LIFEW, NLIFET)
    CALL INSORT (LIFL, LIFEL, NLIFET)
ENDIF

200 CONTINUE
150 CONTINUE

IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES TO FILE LOWLIF
    DO 300 J = 1, (NLIFET / 100)
    WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFEW(J), LIFEL(J)
300 CONTINUE

C ** INITIALIZE VARIABLE BLFPOS()
    DO 325 J = 1, MAXBLF
    BLFPOS(J) = 0
325 CONTINUE

    FIFTY = 0.50E0

C ** PRINT EMPIRICAL BLIVES
    WRITE(3,925)
    DO 350 J = 1, NBLIFE
    BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
    WRITE(3,926) BLFPER(J), LIFEW(BLFPOS(J)), LIFEL(BLFPOS(J))
350 CONTINUE
    WRITE(3,926) FIFTY, LIFEW(NLIFET/2), LIFEL(NLIFET/2)
ENDIF

C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED
    IF (((VARY .EQ. 3) .AND. (NMED .EQ. 1)) .OR. (VARY .EQ. 4)) THEN
        CALL SORTM (ALLM, NUMREG, MCOUNT)
        MID = MCOUNT / 2
        DO 400 L = 1, NUMREG
            MEDM(L) = ALLM(MID,L)
        400 CONTINUE
        CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, &
            NBND, BIGKI, BZERO)
    ENDIF

C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO BDLCO

900 FORMAT(2X,'Copyright (C) 1990, California Institute of ', &
    'Technology. U.S. Government',/2X,'Sponsorship under ', &
    'NASA Contract NAS7-918 is acknowledged',/,, &
    '33X,'INPUT DATA', &
    '/14X,'DRIVERS',25X,'PARAMETER DISTRIBUTIONS', &
    '/48X,'RHO',I6X,'THETA')

901 FORMAT(/2X,'Hgas',13X,'Be(',F5.0,'*',F6.0,')',5X, &
    'U(',F7.5,'*',F8.5,')',4X,'U(',F4.1,'*',F5.1,')', &
    '/2X,'Tgas (deg R)',5X,'Be(',F5.0,'*',F6.0,')',5X, &
    'U(',F7.5,'*',F8.5,')',4X,'U(',F4.1,'*',F5.1,')')

902 FORMAT(/2X,'DECEL SLOPE',6X,'Be(',F5.0,'*',F6.0,')',5X, &
    'U(',F7.5,'*',F8.5,')',4X,'U(',F4.1,'*',F5.1,')', &
    '/2X,'Tgas UNCERT.',5X,'Be(',F5.2,'*',F6.2,')',5X, &
    'U(',F7.5,'*',F8.5,')',4X,'U(',F4.1,'*',F5.1,')')

7 - 182
903 FORMAT(//,50X,60X,'N( MEAN, STD. DEV.)', \ & 10X,'N(',F8.1,'',F7.1,')',//, \ & 2X,'ROTOR SPEED VARIATION (rpm) AT TIME T',II, \ & 'N(',F8.2,'',F7.2,')',//, \ & 2X,'Faccel MODELING ERROR',27X,'N(',F4.1,'',E11.4,')',\ & 2X,'STARTING DECEL TEMPERATURE (deg R)',14X, \ & 'N(',F8.2,'',F7.2,')',//, \ & 2X,'Fdecel MODELING ERROR',27X,'N(',F4.1,'',E11.4,')',\ & 2X,'STRAIN DUE TO GAS BENDING (%), 17X, \ & 'U(',F6.5,'',F9.5,')',\ & 2X,'LAMBDA BLADE PULL',29X, \ & 'U(',F6.5,'',F9.5,')',\ & 2X,'MECHANICAL ANALYSIS FACTOR',20X, \ & 'U(',F8.5,'',F9.5,')',\ & 2X,'COEFFICIENT OF THERMAL EXPANSION FACTOR',7X, \ & 'U(',F8.5,'',F9.5,')',\ & 2X,'THERMAL ANALYSIS FACTOR',23X, \ & 'U(',F8.5,'',F9.5,')',\ & 2X,'DAMAGE MODEL ACCURACY',23X, \ & 'U(in',F8.5,'',F9.5,')',\ & 2X,'B LIVES: WEIBULL LOGNORMAL',/) \ & 2X,'STRAIN-TIME HISTORY _E_'OD (mlsslons),14X,F5.2, \ & 2X,STRAIN-TIME HISTORY NOISE FILTER (%),I6X,F7.5, \ & 2X,INNER LOOP SIZE,15X,F5.2) \ & 2X,'COEFFICIENTS OF ACCELERATION AND DECELERATION ', \ & 'FUNCTIONS',//,2X,'TNE,RM_ STRA, IN AT STARTUP (%) :',//,5X, \ & 'accel(accel' 'gas)= ,E13.6, + ,E13.6,' * Tgas + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tgas ** 2 + ,E13.6,' * Tg
ROTOR SPEED VARIATION PARAMETERS:

- Mean, Std. Dev. (Normal Dist.)
- Decel Model Error Mean & Std Dev
- Decel Tstart Mean & Standard Deviation
- Standard Response Probe Mean & Std Dev
- Strain Due to Gas Bending (%)
- Lambda Blade Pull
- Mechanical Analysis Accuracy Factor
- Coefficient of Thermal Expansion
- Thermal Analysis Accuracy Factor
- Delta Model Accuracy
- Nominal Mech. Strain & Rotor Speed (% RPM)
- Strain-Time History Period (Missions)
- Number of Points in Strain-Time History
- Walker Exponent

COEFFICIENTS FOR STARTUP RESPONSE SURFACE FOR THERMAL STRAIN:

Faccel(Tgas,Hgas) = A + B * T + C * H + D * T**2 + E * H**2 + F * T * H

COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR THERMAL STRAIN:

Fdecel1(m,Tstart) = A + B * Tstart + C * m + D * Tstart**2 + E * m**2 + F * Tstart * m

COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR RPM:

Fdecel13(t) = A + B * t

RPM (TIME) THERMAL STRAIN (%) STRAIN HISTORY INFORMATION

RT, PWA 1480, 001 DIRECTION MATERIAL DESCRIPTION
1.54 1.57 1.8 YIELD & ULTIMATE STRENGTHS, NDIV, NPTS
0.89 -1.0 1.0 #PTS IN DIV, STRAIN RATIO, REGION
0.89 6800 S(1) N(1) RAW
0.89 15000 S(2) N(2) STRAIN-LIFE
0.67 27000 S(3) N(3) (S/N)
0.74 4320 S(4) N(4) DATA
0.56 139300 S(5) N(5) POINTS
0.56 545200 S(6) N(6) FOR THE
0.56 147000 S(7) N(7) SPECIFIC
0.56 434400 S(8) N(8) MATERIAL
0.00 0.56 143400 NO VALUE OF SO SUPPLIED (%)
0.00 0.00 0.00 NUMBER OF REGIONS: W/DATA W/O DATA
0.00 0.00 0.00 LIFE BOUNDARIES: REGION 1
0.00 0.00 0.00 CONSTRAINT ON COEFF. OF VARIATION
0.00 0.00 0.00 #PTS IN RANGE, LOWER BOUND, UPPER BOUND
0.00 0.00 0.00 NORMAL DIST. PRIORS: DELTA, MO, SIGMA2

**LIST OF VARIABLES**
C ALLM() 2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
C BIGK1 EQUAL TO BIGK(1) -- DUMMY PARAMETER FOR CALLS TO SUBROUTINE EXPCTD
C BLDDLIF REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND LCF LIFE CALCULATION
C BLFFER() 1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS() 1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPERICAL BLIVES
C BZERO WEIBULL SHAPE PARAMETER, BETAC, CHARACTERIZING S/N DATA SET
C DUM DUMMY VARIABLE
C EBEND SELECTED VALUE FOR BENDING STRAIN (%)
C EBENDA EBEND LOWER BOUND
C EBENDB EBEND UPPER BOUND
C EMNOM NOMINAL MECHANICAL STRAIN (%)
C EPSL LOGNORMAL WORST OF NSYM RANDOM VARIATE
C EPSW WEIBULL WORST OF NSYM RANDOM VARIATE
C ETHNOM() 1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
C FAA, FAB, FAC, FAD, FAE, FAF COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C FACTR SCALE FACTOR EQUAL TO PHI * KRATIO * 2
C FAERRM STARTUP THERMAL STRAIN RESPONSE SURFACE MEAN
C FAERRS STARTUP THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C FD1A, FD1B COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS
C FD2A, FD2B COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
C FD3A, FD3B COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
C FDERRM DECEL THERMAL STRAIN RESPONSE SURFACE MEAN
C FDERRS DECEL THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
C FIFTY EQUAL TO .5 -- USED TO ACCESS 50% POINT IN LIFEL() AND LIFEW()
C HGAS SELECTED HOT GAS FILM COEFFICIENT, Hgas
C HGASB HGAS LOWER BOUND
C HGASH HGAS UPPER BOUND
C HGASR SELECTED RHO FOR HGAS
C HGASRR HGAS - RHO LOWER BOUND
C HGASRS HGAS - RHO UPPER BOUND
C HGAST SELECTED THETA FOR HGAS
C HGAST1 HGAS - THETA LOWER BOUND
C HGAST2 HGAS - THETA UPPER BOUND
C HGASL SELECTED RHO FOR HGAS
C HGASLR HGAS - RHO LOWER BOUND
C HGASLS HGAS - RHO UPPER BOUND
C HGASLT SELECTED THETA FOR HGAS
C HGAST21 HGAS - THETA LOWER BOUND
C HGAST22 HGAS - THETA UPPER BOUND
C I CONTROL INNER DO LOOP
C IOUT CONTROLS DUMP TO FILE IOUTPR
C J CONTROL DO LOOP FOR EACH BLIFE
C K CONTROL OUTER DO LOOP
C KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C L CONTROL DO LOOP FOR EACH REGION
C LAMA SELECTED COEFFICIENT OF THERMAL ACCURACY FACTOR, Lambda Alpha
C LAMAA LAMA LOWER BOUND
C LAMAB LAMA UPPER BOUND
C LAMDA SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, Lambda
C LAMDA SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, Lambda
C LAMDAAA LAMDA LOWER BOUND
C LAMDAB LAMDA UPPER BOUND
C LAMB SELECTED UNCERTAINTY IN Tgas
C LAMBAA LAMB LOWER BOUND
C LAMBA LAMB UPPER BOUND
C LAMGR SELECTED RHO FOR LAMG
C LAMG SELECTED DEVIATION IN BLADE PULL DUE TO BLADE MASS, Lambda Pull
C LAMGAA LAMG LOWER BOUND
C LAMGB LAMG UPPER BOUND
C LAMGT SELECTED THETA FOR LAMG
C LAMGRT1 LAMG - THETA LOWER BOUND
C LAMGRT2 LAMG - THETA UPPER BOUND
C LAMMP SELECTED TMF MODEL ACCURACY FACTOR, Lambda Tmf
C LAMMPA LAMMP LOWER BOUND
C LAMMPB LAMMP UPPER BOUND
C LIFEL() 1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM USING THE LOGNORMAL DISTRIBUTION -- SORTED VALUES OF THE
LEFT-HAND TAIL

LEFT-HAND TAIL

MISSIONS TO FAILURE BASED ON EPLS

MISSIONS TO FAILURE BASED ON EPSW

1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION

SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR

SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND

SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR UPPER BOUND

MAXIMUM NUMBER OF LIVES TO BE PROVIDED

MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED

MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA CALCULATION

MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY

MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION

MAXIMUM NUMBER OF REGIONS ALLOWED

NUMBER OF M's TO BE USED TO CALCULATE MEDIAN S/N CURVE

I-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION

I-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION

I-D ARRAY CONTAINING THE MEDIAN M VALUES FOR EACH REGION

POINTER TO THE MEDIAN M VALUES — EQUAL TO HALF OF MCOUNT

1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE

MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE

Materials Process variation — CONTROLS MATERIALS PROCESS VARIATION — 0 - NO VARIATION; 1 - VARIATION

I-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION

I-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

I-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH POINT IN THE SPECIFIC MATERIAL S/N DATA SET

I-D ARRAY CONTAINING ROTOR SPEED HISTORY (RPM)

I-D ARRAY CONTAINING THE STRAIN VALUES (%, R = -1.0)

CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()

1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

SELECTED DECELERATION SLOPE, m (deg R / sec)

m LOWER BOUND

m UPPER BOUND

SELECTED RHO FOR m

m - RHO LOWER BOUND

m - RHO UPPER BOUND

SELECTED THETA FOR m

m - THETA LOWER BOUND

m - THETA UPPER BOUND

7 - 186
FUNCTION BLDLIF PERFORMS THE DRIVER TRANSFORMATION AND CALLS RAINF3 TO CALCULATE THE FATIGUE LIFE

PROGRAMMER: L. NEWLIN

VERSION: BLDLIF 3.4 (MATCHR V8.5, RAINF3 V1.1)

Copyright (C) 1990, California Institute of Technology. U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

FUNCTION BLDLIF (TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF,
& MODERI, RPM, TSUBI, SPEED, SLOPE, TSTART, FDIA,
& FDIB, FDIC, FDID, FDL, MODER2, FD2A,
& FD2B, FD3A, FD3B, ETHNOM, MANAL, LAMP, NOSMPD,
& EMMOM, TANA, LAMA, LAMG, EBEND, NTIME, TRUNC,
& PERIOD, WEXP, MM, LNA, LPHIM, KRATIO, LNZ,
& TRSBND, ZROREG, NUMREG, SZERO)

SUBPROGRAMS: RAINF3

INPUTS: TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF, MODERI, RPM,
& TSUBI, SPEED, SLOPE, TSTART, FDIA, FDIB, FDIC, FDID,
& FD1E, FD1F, MODER2, FD2A, FD2B, FD3A, FD3B, ETHNOM, MANAL,
& LAMP, NOSMPD, EMMOM, TANA, LAMA, LAMG, EBEND, NTIME,
& TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO, LNZ, TRSBND,
& ZROREG, NUMREG, SZERO

OUTPUTS: BLDLIF

INTEGER MAXM, MAXREG

PARAMETER (MAXM = 50, MAXREG = 3)
COMMON IOUT

INTEGER I, IOUT, NTIME, NUMREG, TSUBI, ZROREG

REAL BDLIF, EBEND, EM(MAXM), EMNOM, ETH(MAXM), ETHNOM(MAXM),
& ETOF(MAXM), FA, FAA, FAB, FAC, FAD, FAE, FAP, FD1,
& FD1A, FD1B, FD1C, FD1D, FD1E, FD1F, FD2, FD2A, FD2B,
& FD2A, FD2B, FD3A, FD3B, HGAS, KRATIO, LAMA, LAMP, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MANAL,
& MM(0:MAXREG), MODER1, MODER2, NOMSPD, PERIOD, RAINF3,
& RPM(MAXM), SLOPE, SPEED, SZERO, TANAL, TGAS,
& TRSBND(0:MAXREG), TRUNC, TSTART, WEXP

LIST OF VARIABLES

EBEND SELECTED VALUE FOR BENDING STRAIN (%)
EM() 1-D ARRAY CONTAINING THE SIMULATED MECHANICAL STRAIN-TIME HISTORY (%)
EMNOM NOMINAL MECHANICAL STRAIN (%)
ETH() 1-D ARRAY CONTAINING THE SIMULATED THERMAL STRAIN-TIME HISTORY
ETHNOM() 1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
ETOT() 1-D ARRAY CONTAINING THE TOTAL STRAIN-TIME HISTORY
FA VALUE OF ACCELERATION FUNCTION FOR THERMAL STRAIN - SECOND ORDER POLYNOMIAL AS A FUNCTION OF TGAS AND HGAS
FAA, FAB, FAC, FAD, FAE, FAP COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
FD1 VALUE OF DECELERATION FUNCTION FOR THERMAL STRAIN - SECOND ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
FD1A, FD1B, FD1C, FD1D, FD1E, FD1F COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS
FD2 VALUE OF DECELERATION FUNCTION FOR TIME - SECOND ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
FD2A, FD2B COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
FD3 VALUE OF DECELERATION FUNCTION FOR ROTOR SPEED - FIRST ORDER POLYNOMIAL (LINEAR) FUNCTION OF TIME
FD3A, FD3B COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
HGAS SELECTED HOT GAS FILM COEFFICIENT, Hgas
I CONTROLS DO LOOP FOR EACH POINT IN TIME HISTORY
IOUT CONTROLS DUMP TO FILE IOUTPR
K RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
LAMA SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR, Lambda Alpha
LAMP THE UNCERTAINTY IN Tgas
LNA() 1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
LNZ NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
LPHIM() 1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
MANAL SELECTED VALUE FOR MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
MAXM MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
MODER1 MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
MODER2 MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
NOMSPD NOMINAL ROTOR SPEED, RPM
NTIME NUMBER OF POINTS IN STRAIN-TIME HISTORY
NUMREG NUMBER OF REGIONS OF INTEREST
PERIOD LENGTH OF TIME IN MISSIONS OF TIME HISTORY
RAINF3 REAL FUNCTION PERFORMING RAINFLOw COUNTING, DAMAGE ACCUMULATION AND FATIGUE LIFE PREDICTION (USING THE MATERIALS CHARACTERIZATION MODEL)
RPM() 1-D ARRAY CONTAINING ROTOR SPEED HISTORY
SLOPE SELECTED VALUE FOR DECELERATION SLOPE, deg R / sec
SPEED SELECTED VALUE FOR STEADY STATE ROTOR SPEED, rpm
SZERO STRAIN TENSILE TEST POINT, So
TANAL SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR
TGAS SELECTED VALUE FOR HOT GAS TEMPERATURE Tgas (deg R)
TRSBND() 1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR EACH REGION CALCULATED FOR EACH TRIAL
TRUNC VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)
TSTART STARTING DECELERATION TEMPERATURE (deg R)
TSUBI THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS
C WEXP WALKER EXPONENT
C ZROREG Zero REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

C ** CALCULATE STRAIN HISTORY

\[
FA = FAA + FAB \times TGAS + FAC \times HGAS + FAD \times TGAS \times 2
\]
& \[ + FAE \times HGAS \times 2 + FAF \times TGAS \times HGAS + MODER1 \]

ETHNOM(1) = FA

RPM(TSUBLI) = SPEED

\[
FD1 = FD1A + FD1B \times TSTART + FD1C \times SLOPE + FD1D \times TSTART \times 2
\]
& \[ + FD1E \times SLOPE \times 2 + FD1F \times TSTART \times SLOPE + MODER2 \]

FD2 = FD2A + (TSTART - FD2B) / SLOPE

FD3 = FD3A + FD3B \times FD2

RPM(NTIME) = FD3

ETHNOM(NTIME) = FD1

DO 100 I = 1, NTIME

EM(I) = MANAL \times LAMP \times (RPM(I) / NOMSPD) \times 2 \times EMNOM

ETH(I) = TANAL \times LAMA \times ETHNOM(I)

IF ((I .GT. I) .AND. (I .LT. TSUBLI))

ETH(I) = LAMG \times ETH(I)

ETOT(I) = EBEND + EM(I) + ETH(I)

100 CONTINUE

IF (IOUT .EQ. 15) THEN

WRITE(8,*) 'FA = ', FA, ' ETHNOM1 = ', ETHNOM(1)

WRITE(8,*) 'RPMI = ', RPM(TSUBLI), ' LAMG = ', LAMG

WRITE(8,*) 'FD1 = ', FD1, ' FD2 = ', FD2

WRITE(8,*) 'FD3 = ', FD3, ' RPM = ', RPM(NTIME)

WRITE(8,*) 'ETH = ', ETH(I), ' ETOT = ', ETOT(I)

125 CONTINUE

ENDIF

C ** CALL RAINF3 TO CALCULATE DAMAGE AND RESULTING FATIGUE LIFE

BLDLIF = RAINF3 (ETOT, NTIME, TRUNC, PERIOD, WEXP, MM, LNA,
& LPHIM, FRAI0, LN2, TRSBND, ZROREG, NUMREG,
& SZERO)

RETURN
END

*****************************************************************
PARAMETER (MAXLIF = 10000)
COMMON IOUT
INTEGER I, IOUT, NLFET, NUM, PLACE
REAL LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

LIST OF VARIABLES

I CONTROLS DO LOOP FOR INSERTION
IOUT OUTPUT DUMP CONTROLLER
LIFE() 1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
PPM TO BE SORTED
MAXLIF MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
CALCULATION
NEWLIF LIFE VALUE TO BE INSERTED INTO LIFE()
NLFET TOTAL NUMBER OF LIVES CALCULATED BY PFM
NUM NUMBER OF LIFE VALUES IN LIFE()
PLACE POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
TEMP() 1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
INSERTION OF NEWLIF

NUM = NLFET / 2

C FIND POSITION IN LIFE() FOR NEWLIF
IF (NEWLIF .GT. LIFE(NUM)) GOTO 400
DO 100 I = 1, NUM
   IF (NEWLIF .LT. LIFE(I)) THEN
      PLACE = I
      GOTO 110
   ENDIF
100 CONTINUE
110 CONTINUE

C STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()
DO 200 I = (PLACE + 1), NUM
   TEMP(I) = LIFE(I-1)
200 CONTINUE

C INSERT NEWLIF
LIFE(PLACE) = NEWLIF

C SHIFT VALUES OF LIFE() FOLLOWING NEWLIF
DO 300 I = (PLACE + 1), NUM
   LIFE(I) = TEMP(I)
300 CONTINUE

C IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN
400 CONTINUE

RETURN
END

**************************************************************************************************
SUBROUTINE PFRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THEL1,THEL2)
INDEPENDENT RANDOM VARIATES
PROGRAMMER: L. GRONDALSKI, L. NEWLIN
DATE: 9MAR87
SUBPROGRAM: RANDOM

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-1918
C is acknowledged.

**SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)**

**COMMON IOUT**

**DOUBLE PRECISION RAND**

**REAL FRAC, RHO1, RHO2, THE1, THE2, X, Y**

**INTEGER IOUT**

**CALL RANDOM (FRAC, RAND)**

C

**IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC=', FRAC**

**X = FRAC * (RHO2 - RHO1) + RHO1**

C

**CALL RANDOM (FRAC, RAND)**

C

**IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC=', FRAC**

**Y = FRAC * (THE2 - THE1) + THE1**

**& THE1=', THE1, ' THE2=', THE2, ' x=', X, ' y=', Y**

RETURN

END

**C**************************************************************

**THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE**

**PROGRAMMER: L. GRONDALESKI, L. NEWLIN**

**DATE: 9MAR87**

**SUBPROGRAM: GAM**

The random variates are generated using the method described in:

**SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)**

**COMMON IOUT**

**DOUBLE PRECISION RAND**

**REAL A, B, GAM, RHO, THETA, W, X, Y1, Y2**

**INTEGER IOUT**


**Y1 = GAM((RHO * THETA + 1.), RAND)**

**Y2 = GAM((1. - RHO) * THETA + 1.), RAND)**

**W = Y1 / (Y1 + Y2)**

C

**IF (IOUT .EQ. 15) WRITE(8,*) 'Y1=', Y1, ' Y2=', Y2, ' W=', W**

C

**TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION**

**X = W * (B - A) + A**

**IF (IOUT .EQ. 15) WRITE(8,*) 'W=', W, ' X=', X**

RETURN

END

**C**************************************************************

The random variates are generated using an “Acceptance/Rejection Method”
REAL FUNCTION GAM (ALPHA, RAND)

C SUBPROGRAM: RANDOM
COMMON IOUT
INTEGER IOUT
REAL A, ALPHA, ARG, U1, U2, V1, V2
DOUBLE PRECISION RAND

A = ALPHA - 1.
C IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA
10 CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALOG(U1)
V2 = - ALOG(U2)
C IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =', V1, ' V2 =', V2
ARG = A * (V1 - ALOG(V1) - 1.)
C IF (V2 .LT. ARG) GOTO 10
GAM = ALPHA * V1
C IF (IOUT .EQ. 15) WRITE(8,*) 'GAMMA =', GAM
RETURN
END

C*********************************************************************/

SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
FOR THE STRESS FORMULATION
PROGRAMMER: L. NEWLIN
DATE: 30NOV90 FORMAT/COMMENTS: 15JAN92
VERSION: MATCHR VB1.2, VB1.3
MATGRM VB1, VB1.1
C
C Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
& NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
& KRATIO, PVAR, MEDMB, MEDKB, RESID)

INPUTS: READS DATA FROM SPECFD AND RELATD; VARY, MPROC
OUTPUTS: RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG,
& NBND, STR, FTUZ, FTYZ, VARY, MPROC, KRATIO, PVAR, MEDMB, MEDKB, RESID
SUBPROGRAMS: INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDRGG, CONCAV
& MEDIAN, EXPCTD, MUSIG, NORMNG, ADDRGN, GTPVAR, EXPB
FILES: 5:RELATD-OLD; 6:RELATO-NEW
C
IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
& NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
& NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG
REAL BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
& FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
LIST OF VARIABLES

BIGKHT: EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
BZERO: VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N DATA SET
CZERO: EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, CO
DD(): 1-D ARRAY CONTAINING SXY(L)/SX2(L), THE SLOPE OF THE REGRESSION, FOR EACH REGION
DELTA(): 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
FTUZ: ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTYZ: YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
IOUT: OUTPUT DUMP CONTROLLER
IZERO() - D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C FOR EACH REGION
JZERO() - D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
KRATIO: RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
LAMN: LAMBDA-N -- RATIO OF Var(Ln N given S) / (m**2 C**2), CONSTANT OVER REGIONS AND COMPONENTS
LNNF(): 3-D ARRAY CONTAINING LN(RAWNFn()), ALSO INDEXED FOR REGION
LNSTR(): 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT: MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG: MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET: MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MC(): 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCHAT(): 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT(): I-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION
MEDKB(): 1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION (BOOTSTRAP OPTION)
MEDM(): 1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION (BOOTSTRAP OPTION)
MEDMB(): 1-D ARRAY CONTAINING THE MEAN M VALUES FOR EACH REGION (BOOTSTRAP OPTION)
MO(): I-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT(): I-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MPROC: MATERIALS PROCESS VARIATION -- CONTROLS MATERIALS PROCESS VARIATION -- 0 - NO VARIATION; 1 - VARIATION
MU(): I-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MZERO(): 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NNODAT: NUMBER OF NO DATA REGIONS (REGIONS WITHOUT ANY S/N DATA)
NF(): 2-D ARRAY CONTAINING RAWNFn() (CYCLES TO FAILURE) FOR EACH SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NNP(): 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NP(): 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP())-1))-1 OVER ALL DATA SETS IN A REGION (Number of Points Per Region)

7 - 193
**NPTS()**: 1-D array containing number of points in S/N data sets

**NSETS**: Number of related material S/N data sets

**NUMREG**: Number of regions of interest

**PVAR**: Materials process variation

**RANGEM()**: 2-D array containing values of the posterior ranges on M for each region — RANGEM(1,L) is the lower bound and RANGEM(2,L) is the upper bound

**RATSTR()**: 2-D array containing stress data (psi) corrected for stress ratio or total strain data (%) for all S/N data sets

**RAWNF()**: 2-D array containing raw cycles to failure data for all S/N data sets

**RAWSTR()**: 2-D array containing raw stress data (psi) or total strain data (%) for all S/N data sets

**REFNP()**: 1-D array containing the number of points for the specific (reference) material S/N data set in each region

**RESID()**: 1-D array containing the residuals of the regression for each point in the specific material S/N data set

**SIG()**: 1-D array containing values of the posterior normal distribution standard deviation for each region

**SIGMA2()**: 1-D array containing values of the prior normal distribution variance for each region

**STR()**: 2-D array containing RATSTR() for the specific material S/N data set broken into regions (psi or %)

**SUHAT2()**: 1-D array containing residual variances from X on Y regression for each region with (x = Ln S, y = Ln N)

**SWHAT2()**: 1-D array containing residual variances from Y on X regression for each region with (x = Ln S, y = Ln N)

**SX2()**: 1-D array containing sample x variance for each region (x = Ln S)

**SXY()**: 1-D array containing sample x, sample y covariance for each region (x = Ln S, y = Ln N)

**SY2()**: 1-D array containing sample y variance for each region (y = Ln N)

**SZERO**: Stress tensile test point, S, controls type of curve variation desired — 0 - no variation; 1 - S/N randomness only; 2 - uniform variation; 3 - truncated normal variation; 4 - bootstrap

**ZROREG**: Zero region — values chosen to facilitate region do loop beginning value — 0 - zero region exists, 1 - no zero region

OPEN(5, FILE = ‘RELATD’, STATUS = ’OLD’)
OPEN(6, FILE = ’RELATO’, STATUS = ’NEW’)

C RELATED CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET INFORMATION
C PERFORM CALCULATIONS COMMON TO UNIFORM, NORMAL, AND BOOTSTRAP TYPE OF VARIATION
C INITIALIZE PRIMARY ARRAYS
CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NF, LNNF, LNSTR, REFN, &
NFR, STR, MPNT, MZERO, DELTA, MO, SIGMA2)
C READ, CONVERT, ECHO INFORMATION
CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, &
LNNF, REFN, STR, NF, SZERO, ZROREG, NUMREG, NNODAT, &
NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, &
SIGMA2, KRATIO, LAMN)
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD, &
SWHAT2, SUHAT2, SFPR, MEDMB, MEDKB, RESID)
C CALCULATE M CONSTRAINT BASED ON CC
CALL FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN
C CALCULATIONS FOR ALL TYPES OF VARIATION SAVE NORMAL
C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS
    CALL INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO, MCHAT)
&
C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
    IF (MPROC .EQ. i) THEN
        CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
    ENDIF
C COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO OBTAIN POSTERIOR RANGES ON M
    CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT, RANGEM)
&
C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
    CALL ADDRNG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
    CALL CONCAV (NUMREG, RANGEM)
C WRITE RESULTS TO FILE DUMP
    WRITE(7,900)
    DO 25 L = 1, NUMREG
        WRITE(7,905) L, IZERO(I, L), IZERO(2, L), JZERO(1, L), JZERO(2, L)
    CONTINUE
25 WRITE(7,910)
    DO 50 L = 1, NUMREG
        WRITE(7,915) L, MCHAT(2,L), MCHAT(I,L)
    CONTINUE
50 WRITE(7,920)
    IF (CZERO .GT. 0.0) THEN
        DO 150 L = 1, NUMREG
            IF (MCPNT(L) .EQ. 1) THEN
                WRITE(7,960) L, MC(I,L)
            ELSEIF (MCPNT(L) .EQ. 2) THEN
                WRITE(7,965) L, MC(I,L), MC(2,L)
            ENDIF
        ENDIF
150 WRITE(7,970)
    ENDIF
    WRITE(7,930)
    DO 100 L = 1, NUMREG
        WRITE(7,940) L, RANGEM(I,L), RANGEM(2,L)
    CONTINUE
100 WRITE(7,950)
C CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO
    CALL MEDIAN (NUMREG, RANGEM, MEDM)
C CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
    CALL EXPCTD (1, MEDM, REFPN, STR, NF, ZZERO, NUMREG, ZOREG, NBND, BIGKHT, BZERO)
&
C CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED
IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
   DO 200 L = 1, NUMREG
      RANGEM(1,L) = MEDM(L)
      RANGEM(2,L) = MEDM(L)
   CONTINUE
ENDIF
ELSEIF (VARY .EQ. 3) THEN
C NORMAL VARIATION IS DESIRED
C CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION
   CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO,
                 SIGMA2, MCHAT, MU, SIG)
C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
   IF (MPROC .EQ. 1) THEN
      CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
   ENDIF
C COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M
   CALL NORMNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
   CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO,
                 MPNT, MO, SIGMA2)
C ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
   CALL CONCAV (NUMREG, RANGEM)
C WRITE RESULTS TO FILE DUMP
   WRITE(7,975)
   DO 350 L = 1, NUMREG
      WRITE(7,980) L, MCHAT(I,L)
   CONTINUE
   IF (CZERO .GT. 0.0) THEN
      DO 360 L = 1, NUMREG
         IF (MCPNT(L) .EQ. 1) THEN
            WRITE(7,965) L, MC(I,L)
         ELSEIF (MCPNT(L) .EQ. 2) THEN
            WRITE(7,970) L, MC(I,L), MC(2,L)
         ENDIF
      CONTINUE
   ENDIF
   WRITE(7,920)
   WRITE(7,930)
   DO 370 L = 1, NUMREG
      WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
   CONTINUE
   WRITE(7,950)
   WRITE(7,960)
   DO 380 L = 1, NUMREG
      WRITE(7,990) L, MU(L), SIG(L)
   CONTINUE
ELSE
C BOOTSTRAPPING IS REQUIRED
   WRITE(7,999)
ENDIF
FIRST CALCULATE OTHER REGION PARAMETERS BASED ON THE EXPECTED M AND K VALUES

CALL EXPB (MEDMB, MEDKB, SZERO, NUMREG, ZOREG, NBND)

ENDIF

PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS

IF (MPROC .EQ. 1) THEN
    WRITE (7,995) PVAR
ENDIF

FORMAT STATEMENTS

900 FORMAT(2X,'Copyright (C) 1990, California Institute of Technology. U.S. Government',//,2X,'NASA Contract NAS7-918 is acknowledged.',///,2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',& && ///,2X,'95% CONFIDENCE INTERVALS ON C AND m',& && ///,FOR EACH REGION',/
905 FORMAT(7X,'REGION: ',II,7X,'Io = (''F12.9','','F12.9',''),',& && ///,24X,'Io = (''F12.9','','F12.9','')')&
910 FORMAT(///,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',& && ///,7X,'REGION',8X,'E(C)',12X,'E(m)',//)&
915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)&
920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGES ON m FOR EACH REGION')&
930 FORMAT(///,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',//)&
940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)&
950 FORMAT(///)&
960 FORMAT(///,2X,'RANGE ON m FOR EACH REGION IMPLIED BY C CONSTRAINT',& && ///,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',//)&
965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')&
970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)&
975 FORMAT(2X,'Copyright (C) 1990, California Institute of Technology. U.S. Government',//,2X,'NASA Contract NAS7-918 is acknowledged.',///,2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',& && ///,2X,'ESTIMATES OF m FOR EACH REGION',& && ///,7X,'REGION',12X,'E(m)',//)&
980 FORMAT(9X,I1,11X,F10.6)&
985 FORMAT(2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS',& &&///,2X,'REGION',5X,'MEAN',8X,'STD DEV',//)&
990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)&
995 FORMAT(///,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',& && ///,'MEDIAN S/N CURVE',/2X,'WARRANTED BY THE AVAILABLE ',& && ///,'INFORMATION',//,7X,E11.5)

RETURN
END
SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
PROGRAMMER: L. NEWLIN
DATE: 5OCT87

SUBROUTINE TRMNAT
WRITE (8, *) 'PROGRAM EXECUTION TERMINATED'
STOP
END

SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
PROGRAMMER: L. NEWLIN
DATE: CODE: 21JUN88 COMMENTS: 13JUL89
VERSION: MATCHR V8.3, V8.4, V8.5
MATGRM V4, V4.3, V4.5

SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, & REFNF, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

INPUTS: 
OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNF, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER ( AXDAT = 50, MAXREG = 3, MAXSET = 5)

COMMON IOUT
INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG), & NPTS(0:MAXSET), REFNF(MAXREG)
REAL DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG), & LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG), & NF(MAXDAT, MAXREG) & RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET), & RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG), & STR(MAXDAT, MAXREG)

LIST OF VARIABLES

DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
I Controls DO loop for each data point in a data set
IOUT OUTPUT DUMP CONTROLLER
J Controls DO loop for each data set
K Controls DO loop for each point in a region
L Controls DO loop for each region
LNNF() 3-D ARRAY CONTAINING LN(RANNF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
C SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATD; "CONVERTS" THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO SPECFO AND RELATO. RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS SPECIFIED BY USER.
PROGRAMMER: L. NEWLIN
DATE: 21JUN88 FORMAT/COMMENTS: 12AUG91
VERSION: MATCHR V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE RCE

INPUTS: VARY, MPROC
OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN

SUBPROGRAMS: TRMNAT, CONVRT

IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT, COUNT, I, IOUT, J, K, L, M, MPNT(MAXREG), MPROC, NDIV, NNODAT, NP(0:MAXSET, MAXREG), NPTS(0:MAXSET), NSETS, NUM, NUMREG, REFNP(MAXREG), PEG, VARY, ZROREG
REAL CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ, K, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG), LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG), MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG), RAT, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG), STR(MAXDAT, MAXREG), SZERO

CHARACTER*40 DESCRP(0:MAXSET)

LIST OF VARIABLES

COUNT INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO, CONVERSION, AND BREAK UP
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, CO
DELT A() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
DESCRP() 1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
FTU ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
FTUZ ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTY YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTYZ YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LAMN LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C*-2), CONSTANT OVERALL REGIONS AND COMPONENTS
LNHF() 3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
M CONTROLS DO LOOP FOR EACH DATA DIVISION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MPROC MATERIALS PROCESS variation -- CONTROLS MATERIALS PROCESS VARIATION -- 0 - NO VARIATION; 1 - VARIATION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
IS THE UPPER BOUND
1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
REGIONS OF INTEREST

NDIV
NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
REGION PAIRS DURING INPUT

NF()
2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS

NNODAT
Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)

NP()
2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
IN EACH REGION

NPTS()
1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS

NSETS
NUMBER OF RELATED MATERIAL S/N DATA SETS

NUM
NUMBER OF DATA POINTS IN A PARTICULAR DIVISION

NUMREG
NUMBER OF REGIONS OF INTEREST

RATIO
STRESS RATIO (R = -1.0 IS DESIRED)

RATSTR()
2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS

RAWNF()
2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
DATA SETS

RAWSTR()
2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
DATA (%) FOR ALL S/N DATA SETS

REFNP()
1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
(REFERENCE) MATERIAL S/N DATA SET IN EACH REGION

REG
REGION OF INTEREST IN A PARTICULAR DIVISION

SIGMA2()
1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
VARIANCE FOR EACH REGION

STR()
2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

SZERO
STRESS TENSILE TEST POINT, So

VARY
CONTROLS TYPE OF CURVE VARIATION DESIRED - 0 - NO
VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM
VARIATION; 3 - TRUNCATED NORMAL VARIATION

ZREG
ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
REGION

C INITIALIZE COUNT AND NBND()
C
COUNT = 0
DO 10 L = 0, MAXREG
NBND(L) = 0.0
10 CONTINUE
C
C INPUT DATA ON SPECIFIC MATERIAL FROM SPECFD AND ECHO TO SPECFO
C
READ(I,*) DESCRP(0), FTY, FTU, NDIV, NPTS(0)
IF (NPTS(0) .GT. MAXDAT) THEN
WRITE(8,*) 'ERROR: OVER NUMBER OF POINTS LIMIT IN',
& 'SPECIFIC MATERIAL'
CALL TRMNAT
ENDIF
WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)
WRITE(3,905)
IF (IOUT .EQ. i0) WRITE(8,905)
C
C STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ
FTUZ = FTU
FTYZ = FTY
C
C INPUT STRESS/LIFE INFORMATION - INCLUDING STRESS RATIO AND REGION
C INFORMATION FROM SPECFD AND ECHO TO SPECFO
C
DO 100 M = 1, NDIV
READ (1,*) NUM, RATIO, REG
IF (ABS(RATIO) .GT. 1.0) THEN
WRITE(8,* 'ERROR: INVALID VALUE FOR RATIO: ', RATIO
ENDIF
CALL TRMNAT
ENDIF

IF (REG .GT. MAXREG) THEN
WRITE(8,*) 'ERROR: OVER REGION LIMIT IN SPECIFIC DATA SET'
CALL TRMNAT
ENDIF

DO 110 I = (COUNT + 1), (COUNT + NUM)
110 CONTINUE

C CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
C
IF (RATIO .EQ. -1.0) THEN
C STRESS RATIO IS CORRECT
DO 120 I = (COUNT + 1), (COUNT + NUM)
120 CONTINUE
ELSE
C STRESS RATIO TRANSFORMATION MUST BE DONE
C ENDIF

CALL CONVRT (0, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR,
& RATIO, FTU, PFTY)
ENDIF

C ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL
DO 130 I = (COUNT + 1), (COUNT + NUM)
130 CONTINUE

C BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C EXPECTED, AND PAREST
C
K = NP(0,REG)
DO 140 I = (COUNT + 1), (COUNT + NUM)
K = K + 1
LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
LNNF(K,0,REG) = ALOG(RAWNF(I,0))
STR(K,REG) = RATSTR(I,0)
NF(K,REG) = RAWNF(I,0)
140 CONTINUE

IF (K .GT. MAXDAT) THEN
WRITE(8,*) 'ERROR: OVER NUMBER OF POINTS LIMIT IN ',
& 'SPECIFIC MATERIAL'
CALL TRMNAT
ENDIF

NP(0,REG) = K
REFPN(REG) = K
COUNT = COUNT + NUM
100 CONTINUE

IF (NPTS(0) .NE. COUNT) THEN
WRITE(8,*) 'ERROR: NUMBER OF POINTS PER DIVISION ',
& 'INCORRECTLY SPECIFIED'
WRITE(8,*) 'IN SPECIFIC DATA SET'
CALL TRMNAT
ENDIF
READ(1,*) SZERO
IF (NINT(SZERO) .GT. 0) THEN
  ZROREG = 0
ELSE
  ZROREG = 1
ENDIF
IF (IOUT .EQ. 10) WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG
C INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION
READ(1,*) NUMREG, NNODAT
IF ((NUMREG + NNODAT) .GT. MAXREG) THEN
  WRITE(8,*) 'ERROR: EXCEEDED LIMIT ON NUMBER OF REGIONS'
  CALL TRMNAT
ENDIF
DO 150 L = ZROREG, (NUMREG + NNODAT)
  READ(L,*) NBND(L)
150 CONTINUE
READ(1,*)
CZERO
DO 160 L = NUMREG, (NUMREG + NNODAT)
  READ(L,*) MPNT(L), MZERO(I,L), MZERO(2,L)
160 CONTINUE
WRITE(3,913)
IF (ZROREG .EQ. 0) WRITE(3,914) SZERO
IF (IOUT .EQ. 10) WRITE(8,913)
IF (ZROREG .EQ. 0) WRITE(8,914) SZERO
ENDIF
WRITE(3,915) NUMREG, NNODAT
IF (IOUT .EQ. 10) WRITE(8,915) NUMREG, NNODAT
DO 170 L = ZROREG, (NUMREG + NNODAT)
  WRITE(3,920) NBND(L)
  IF (IOUT .EQ. 10) WRITE(8,920) NBND(L)
170 CONTINUE
WRITE(3,925) CZERO
IF (IOUT .EQ. 10) WRITE(8,925) CZERO
DO 180 L = NUMREG, (NUMREG + NNODAT)
  WRITE(3,930) L, MPNT(L), MZERO(I,L), MZERO(2,L)
  IF (IOUT .EQ. 10) WRITE(8,930) L, MPNT(L), MZERO(I,L), MZERO(2,L)
  IF ((VARY .EQ. 3) .AND. (MPNT(L) .EQ. 0)) THEN
    WRITE(8,*) 'ERROR: NORMAL VARIATION REQUIRES A PRIOR ',
    'RANGE ON M'
    CALL TRMNAT
  ENDIF
180 CONTINUE
IF (VARY .EQ. 3) THEN
C READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
WRITE(3,945)
IF (IOUT .EQ. 10) WRITE(8,945)
DO 190 L = 1, (NUMREG + NNODAT)
  READ(L,*) DELTA(L), MO(L), SIGMA2(L)
  WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
  IF (IOUT .EQ. 10) WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
  IF ((DELTA(L) .LT. 0.0) .OR. ((DELTA(L) .GT. 0.0) .AND. (MO(L) .LE. 0.0))) THEN
    WRITE(8,*) 'ERROR: BAD VALUE FOR DELTA OR VALUE OF MO ',
    'INCONSISTENT WITH DELTA IN REGION ', L
    CALL TRMNAT
  ENDIF
190 CONTINUE
ENDIF

IF (MPROC .EQ. 1) THEN
  READ(1,*) KRATIO, LAMN
  WRITE(3,955) KRATIO, LAMN
ENDIF

IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN

C BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATED
C AND THEN ECHO TO RELATED

READ(5,*) NSETS

IF (NSETS .GT. MAXSET) THEN
  WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF RELATED DATA SETS'
  CALL TRMNAT
ENDIF

WRITE(6,935) NSETS

DO 200 J = 1, NSETS
  COUNT = 0
  IF (IOUT .EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS
  READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)
  IF (NPTS(J) .GT. MAXDAT) THEN
    WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF POINTS IN ', &
    'SET ', J
    CALL TRMNAT
  ENDIF
  WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
  IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)
  IF (IOUT .EQ. 10) WRITE(8,905)

DO 300 M = 1, NDIV
  READ(5,*) NUM, RATIO, REG
  IF (ABS(RATIO) .GT. 1.0) THEN
    WRITE(8,*) 'ERROR: INVALID VALUE OF RATIO: ', RATIO
    CALL TRMNAT
  ENDIF
  IF (REG .GT. MAXREG) THEN
    WRITE(8,*) 'ERROR: OVER REGION LIMIT IN RELATED MATERIAL ', J
    CALL TRMNAT
  ENDIF
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
    WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
  ENDIF
  DO 310 I = (COUNT + 1), (COUNT + NUM)
    READ(5,*) RAWSTR(I,J), RAWNF(I,J)
  310 CONTINUE

C CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT

IF (RATIO .EQ. -1.0) THEN
  C STRESS RATIO IS CORRECT
  DO 320 I = (COUNT + 1), (COUNT + NUM)
    RATSTR(I,J) = RAWSTR(I,J)
  320 CONTINUE
ELSE
C STRESS RATIO TRANSFORMATION MUST BE DONE
CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR,
& RATSTR, RATIO, FTU, FTY)
ENDIF
C RECORD BOTH S/N DATA SETS TO RELATO
DO 330 I = (COUNT + 1), (COUNT + NUM)
& WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG,
& RATSTR(I,J), RAWNF(I,J)
& IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,J), RAWNF(I,J),
& RATIO, REG, RATSTR(I,J), RAWNF(I,J)
330 CONTINUE
K = NP(J,REG)
DO 340 I = (COUNT + 1), (COUNT + NUM)
& K = K + 1
& LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
& LNNF(K,J,REG) = ALOG(RAWNF(I,J))
340 CONTINUE
IF (K .GT. MAXDAT) THEN
& WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF POINTS ',
& 'IN SET ', J
& CALL TRMNAT
ENDIF
NP(J,REG) = K
COUNT = COUNT + NUM
300 CONTINUE
IF (NPTS(J) .NE. COUNT) THEN
& WRITE(*,*) 'ERROR: NUMBER OF POINTS PER
& DIVISION INCORRECTLY SPECIFIED IN SET ', J
& CALL TRMNAT
ENDIF
200 CONTINUE
C FORMAT STATEMENTS USED TO WRITE TO SPECFO AND RELATO
900 FORMAT(///, 13X, 'MATERIAL INPUT', ///, 2X, 'DESCRIPTION:', 2X, A40, ///,
& 2X, 'YIELD STRENGTH', 18X, E11.5, ///, 2X, 'ULTIMATE STRENGTH',
& 15X, E11.5, ///, 2X, 'NUMBER OF POINTS', 16X, I2)
& 5X, 'STRESS', 7X, 'LIFE', 7X, 'RATIO', 3X, 'REGION', 5X,
& 'STRESS', 7X, 'LIFE')
910 FORMAT(2X, E11.5, 2X, F9.0, 5X, F5.2, 5X, I1, 5X, E11.5, 2X, F9.0)
913 FORMAT(///)
914 FORMAT(2X, ' THERE IS A NO DATA REGION TO THE  LEFT WITH AN SO OF',
& 5X, E11.5)
915 FORMAT(2X, ' THERE IS', 'I2', ' REGION(S) WITH DATA ',
& '/', 2X, ' AND', 'I2', ' REGION(S) TO THE RIGHT WITHOUT DATA',
& 'I2', ' THE UPPER BOUND(S) OF THE REGION(S) ARE ',
& '(CYCLES): ', '/')
920 FORMAT(10X, E9.3)
C** THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C STRESS RATIO, R, IS NOT -1.0
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87  COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C           V8.3, V8.4, V8.5
C
C SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)
C
C INPUTS:  J, NUM1, NUM2, STR, R, FTU, FTY
C OUTPUTS: RSTR
C
C IMPLICIT NONE
INTEGER MAXDAT, MAXSET
PARAMETER ( AXDAT = 50, MAXSET = 5)
COMMON IOUT
INTEGER I, IOUT, J, NUM1, NUM2
REAL FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
     & STR(MAXDAT, 0:MAXSET), TEST

LIST OF VARIABLES
FTU   ULTIMATE STRENGTH OF MATERIAL (PSI)
FTY   YIELD STRENGTH OF MATERIAL (PSI)
I     CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
IOUT  OUTPUT DUMP CONTROLLER
J     DATA SET OF INTEREST
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
NUM1  FIRST INDEX TO BE TRANSFORMED
NUM2  LAST INDEX TO BE TRANSFORMED
R     STRESS RATIO (R = -1.0 IS DESIRED)
RSTR() STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
STR() ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
C TEST  
Kt * Smax * (1 - R)/2, TO BE COMPARED WITH FTY

C Kt IS ASSUMED TO BE ONE

DO 100 I = NUM1, NUM2
  TEST = STR(I,J) * (1.0 - R)/2.0
  IF (IOUT.EQ.10) WRITE(8,*),' I =',I,' J =',J,' TEST =',TEST
  IF (TEST .GE. FTY) THEN
    RSTR(I,J) = TEST
    IF (IOUT.EQ.10) WRITE(8,*),' 1: RSTR() =',RSTR(I,J)
  ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
    RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))
    IF (IOUT.EQ.10) WRITE(8,*),' 2: RSTR() =',RSTR(I,J)
  ELSE
    RSTR(I,J) = TEST/(1.0 -((1.0 + R) * STR(I,J)
      / (2.0 * FTU)))
    IF (IOUT.EQ.10) WRITE(8,*),' 3: RSTR() =',RSTR(I,J)
  END IF
100 CONTINUE
RETURN
END

C******************************************************************************

C SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C PROGRAMMER:  L. NEWLIN
C DATE: 15JAN92
C VERSION: MATCHR VBI.3 MATGRM VB1.1

SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY,
  & SY2, DD, SWHAT2, SUHAT2, NPPR, MEDMB, MEDKB,
  & RESID)
C INPUTS: NUMREG, NSETS, NP, LNSTR, LNNF
C OUTPUTS: SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR, MEDMB, MEDKB, RESID
C IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NPPR(MAXREG),
  & NSETS, NUMREG
REAL BB(MAXREG), CC(MAXREG), DD(MAXREG),
  & DIFFX(MAXDAT, 0:MAXSET), DIFFY(MAXDAT, 0:MAXSET),
  & LNNF(MAXDAT, 0:MAXSET, MAXREG),
  & LNSTR(MAXDAT, 0:MAXSET, MAXREG),
  & MEANX(0:MAXSET), MEANY(0:MAXSET), MEDKB(0:MAXREG),
  & MEDMB(0:MAXREG), RESID(MAXDAT), SUHAT2(MAXREG),

7 - 207
& WHAT2(MAXREG), SX2(MAXREG), SXY(MAXREG), SY2(MAXREG), WHAT(MAXDAT, 0:MAXSET)

LIST OF VARIABLES

BB()  1-D ARRAY CONTAINING SXY(L)/SY2(L), THE SLOPE OF THE X ON Y
      REGRESSION, FOR EACH REGION
CC()  1-D ARRAY CONTAINING MEANX-DD(L)*MEANX, THE Y-INTERCEPT OF
      THE Y ON X REGRESSION, FOR EACH REGION
DD()  1-D ARRAY CONTAINING SXY(L)/SX2(L), THE SLOPE OF THE Y ON X
      REGRESSION, FOR EACH REGION
DIFFX()  2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K, J, L)
         AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
DIFFY()  2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K, J, L)
         AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
IOUT  OUTPUT DUMP CONTROLLER
J    CONTROLS DO LOOP FOR EACH DATA SET
K    CONTROLS DO LOOP FOR EACH POINT IN A REGION
L    CONTROLS DO LOOP FOR EACH REGION
LNFR()  3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
LNFM()  3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT  MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET  MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MEANX()  1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION
         L AND DATA SET J (X = Ln S)
MEANY()  1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
         L AND DATA SET J (Y = Ln N)
MEDKB()  1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION
         (BOOTSTRAP OPTION)
MEDMB()  1-D ARRAY CONTAINING THE MEAN M VALUES FOR EACH REGION
         (BOOTSTRAP OPTION)
NP()  2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
      SET IN EACH REGION
NPFR()  1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
      ALL DATA SETS IN A REGION (NUMBER OF POINTS PER REGION)
NSETS  NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMREG  NUMBER OF REGIONS OF INTEREST
RESID()  1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
         POINT IN THE SPECIFIC MATERIAL S/N DATA SET
SUHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
          REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SUHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
          REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SX2()  1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
       (X = Ln S)
SXY()  1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
       EACH REGION (X = Ln S, Y = Ln N)
SY2()  1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
       (Y = Ln N)
WHAT()  2-D ARRAY CONTAINING THE RESIDUALS OF THE Y ON X REGRESSION
        (X = Ln S, Y = Ln N)

C INITIALIZE ARRAYS

DO 50 L = 1, MAXREG
   SY2(L) = 0.0
   SX2(L) = 0.0
   SXY(L) = 0.0
   SUHAT2(L) = 0.0
   BB(L) = 0.0
   CC(L) = 0.0
   DD(L) = 0.0
   NPPR(L) = 0
50 CONTINUE

DO 55 L = 0, MAXREG
   MEDMB(L) = 0.0
   MEDKB(L) = 0.0
55 CONTINUE

DO 60 J = 0, MAXSET

7 - 208
DO 70 K = 1, MAXDAT
DIFFY(K,J) = 0.0
DIFFX(K,J) = 0.0
WHAT(K,J) = 0.0
70 CONTINUE
MEANY(J) = 0.0
MEANX(J) = 0.0
60 CONTINUE
DO 75 K = 1, MAXDAT
RESID(K) = 0.0
75 CONTINUE
NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION
DO 100 L = 1, NUMREG
DO 200 J = 0, NSETS
FIRST CALCULATE SAMPLE X AND Y MEANS
FOR DATA SET J IN REGION L
MEANY(J) = 0.0
MEANX(J) = 0.0
IF (IOUT .EQ. 10) WRITE(8,*),'L =', L, ' J =', J,
& ' NP =', NP(J,L)
DO 250 K = 1, NP(J,L)
MEANY(J) = MEANY(J) + LNNF(K,J,L)
MEANX(J) = MEANX(J) + LNSTR(K,J,L)
IF (IOUT .EQ. 10) WRITE(8,*), 'LNNF =', LNNF(K,J,L),
& ' LNSTR =', LNSTR(K,J,L)
250 CONTINUE
MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
IF (IOUT .EQ. 10) WRITE(8,*),'MEANY(J) =', MEANY(J),
& ' MEANX(J) =', MEANX(J)
NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY,
OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
DATA SET IN REGION L
DO 300 K = 1, NP(J,L)
DIFFY(K,J) = LNNF(K,J,L) - MEANY(J)
DIFFX(K,J) = LNSTR(K,J,L) - MEANX(J)
SX2(L) = SX2(L) + DIFFX(K,J) ** 2
SY2(L) = SY2(L) + DIFFY(K,J) ** 2
SX2(L) = SX2(L) + DIFFX(K,J) * DIFFY(K,J)
300 CONTINUE
NPPR(L) = NPPR(L) + NP(J,L) - 1
IF (IOUT .EQ. 10) WRITE(8,*), 'NPPR(L) =', NPPR(L)
200 CONTINUE
IF (SY2(L) .GE. 0.0) THEN
WRITE(8,*), 'ERROR: SY2 >= 0 IN REGION', L
CALL TRMNAT
ENDIF
NPPR(L) = NPPR(L) - 1
IF (NPPR(L) .LE. 0) THEN
WRITE(8,*), 'ERROR: TOO FEW POINTS FOR REGRESSION IN',
& ' REGION ', L
CALL TRMNAT
ENDIF
CALCULATE THE REGRESSION PARAMETERS
C

SY2(L) = SY2(L) / FLOAT(NPPR(L))
SXY(L) = SXY(L) / FLOAT(NPPR(L))

DD(L) = SXY(L) / SX2(L)
BB(L) = SXY(L) / SY2(L)
CC(L) = MEANX(0) - DD(L) * MEANX(0)
MEDMB(L) = - DD(L)
MEDKB(L) = EXP (- CC(L) / DD(L))

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
  & ' SX2(L) =', SX2(L)
  WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
  & ' BB(L) =', BB(L)
  WRITE(8,*) 'CC(L) =', CC(L), ' MEDMB(L) = ', MEDMB(L),
  & ' MEDKB(L) = ', MEDKB(L)
ENDIF

NOW CALCULATE THE RESIDUAL VARIANCES, WHAT2, SUHAT2, FOR EACH
REGION FROM THE Y ON X AND X ON Y REGRESSIONS:

DO 400 J = 0, NSETS
  IF (IOUT .EQ. 10) WRITE(8,*)
    'J =', J, ' NP(J,L) =', NP(J,L)
  DO 500 K = 1, NP(J,L)
    WHAT(K,J) = DIFFY(K,J) - DD(L) * DIFFX(K,J)
    SUHAT2(L) = SUHAT2(L) + WHAT(K,J) ** 2
  ENDIF
  WRITE(8,*)
  'WHAT2(L) = ', WHAT2(L), ' SUHAT2(L) = ', SUHAT2(L)
ENDIF

CONTINUE

DO 600 K = 1, NP(0,L)
  RESID(K) = WHAT(K,0) * SQRT (FLOAT(NP(0,L)) / FLOAT(NP(0,L)-2))
  WRITE(4,*)
  'K = ', K, ' WHAT = ', WHAT(K,0), ' RESID = ', RESID(K)
ENDIF

CONTINUE

RETURN
END

C******************************************************************************
C SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C THE CODE GIVEN BY THE USER
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 8OCT87 COMMENTS: 13JUL89
C             V8.4, V8.5
C SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
C INPUTS:  NUMREG, CZERO, SX2, SXY, SY2
C OUTPUTS: MCPNT, MC
C
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), NUMREG
REAL ARGI, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
     & SXY(MAXREG), SY2(MAXREG)

C LIST OF VARIABLES
ARG1 INTERMEDIATE CALCULATION VARIABLE
ARG2 INTERMEDIATE CALCULATION VARIABLE
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
    COEFFICIENT OF VARIATION, Co
CZERO2 EQUAL TO CZERO ** 2
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
      CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA -- MC(1,L) IS
      THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
      MC() FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
       (X = Ln S)
SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR
       EACH REGION (X = Ln S, Y = Ln N)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
       (Y = Ln N)
C
C INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
   MCPNT(L) = 0
   MC(1,L) = 0.0
   MC(2,L) = 0.0
50 CONTINUE
C BEGIN CALCULATIONS
CZERO2 = CZERO ** 2

IF (IOUT .EQ. 10) &
   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2
DO 100 L = 1, NUMREG
   ARG1 = SX2(L) - CZERO2
   ARG2 = 0.0
   IF (CZERO .EQ. 0.0) THEN
      MCPNT(L) = 0
   ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN
      MCPNT(L) = 1
      MC(1,L) = - SX2(L) / (2.0 * SXY(L))
   ELSE
      MCPNT(L) = 2
      MC(1,L) = - SX2(L) / (2.0 * SXY(L))
ELSE

THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
COMMON CALCULATIONS

ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)

IF (ARG2 .LT. 0.0) THEN
ARG2 IS NEGATIVE -- IMPLIES M IS COMPLEX
WRITE(8,*), 'ERROR: CO TOO LOW'
CALL TRMNAT
ELSE
ARG2 = ARG2 ** 0.5
ENDIF

IF (SX2(L) .LT. CZERO2) THEN
AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M
MCPNT(L) = 1
MC(1,L) = (- SXY(L) - ARG2) / ARG1
ELSE
SX2(L) .GT. CZERO2 -- THIS TIME THE M CONSTRAINT IS A RANGE
MCPNT(L) = 2
MC(1,L) = (- SXY(L) - ARG2) / ARG1
MC(2,L) = (- SXY(L) + ARG2) / ARG1
ENDIF

100 CONTINUE

ENDIF

100 CONTINUE

IF (IOUT .EQ. 10) THEN
DO 200 L = 1, NUMREG
WRITE(8,*), 'L = ', L, ' MCPNT = ', MCPNT(L)
WRITE(8,*), 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
WRITE(8,*), 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
200 CONTINUE
ENDIF

RETURN
END
INTEGER CHITAB, MAXREG, TTAB

PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)

COMMON IOUT

INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG

REAL ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
  IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
  SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
  SX2(MAXREG), T, T025(TTAB)

DATA (CHI025(I), I = 1, 75) /
  0.000982669, 0.0503656, 0.215795, 0.484419, 0.831211,
  1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
  3.81575, 4.40379, 5.00974, 5.62872, 6.26214,
  6.90766, 7.56418, 8.23075, 8.90655, 9.59083,
  10.28293, 10.9823, 11.6885, 12.4011, 13.1197,
  13.8439, 14.5733, 15.3079, 16.0471, 16.7908,
  17.53, 18.28, 19.04, 19.80, 20.56,
  21.33, 22.10, 22.87, 23.63, 24.39,
  25.21, 25.99, 26.78, 27.57, 28.36,
  29.13, 29.92, 30.72, 31.51, 32.3574,
  33.15, 33.96, 34.77, 35.58, 36.39,
  37.21, 38.02, 38.84, 39.66, 40.4817,
  41.30, 42.12, 42.95, 43.77, 44.60,
  45.43, 46.26, 47.09, 47.92, 48.7576,
  49.59, 50.42, 51.26, 52.10, 52.94 /

DATA (CHI975(I), I = 1, 75) /
  53.78, 54.62, 55.46, 56.30, 57.1532,
  57.90, 58.84, 59.69, 60.54, 61.39,
  62.24, 63.09, 63.94, 64.79, 65.6466,
  66.50, 67.35, 68.21, 69.07, 69.92,
  70.78, 71.64, 72.50, 73.36, 74.2219,
  75.08, 75.94, 76.80, 77.67, 78.53,
  79.40, 80.27, 81.13, 82.00, 82.87,
  83.73, 84.60, 85.47, 86.34, 87.21,
  88.07, 88.95, 89.83, 90.70, 91.57,
  92.45, 93.32, 94.19, 95.07, 95.94,
  96.82, 97.70, 98.57, 99.45, 100.33,
  101.21, 102.09, 102.97, 103.86, 104.73,
  105.61, 106.49, 107.37, 108.25, 109.14,
  110.02, 110.90, 111.79, 112.67, 113.56,
  114.44, 115.33, 116.21, 117.10, 117.98 /

DATA (CHI975(I), I = 1, 75) /
  5.02369, 7.37776, 9.34840, 11.1433, 12.8325,
  14.4494, 16.0128, 17.5346, 19.0228, 20.4831,
  21.9200, 23.3367, 24.7356, 26.1190, 27.4884,
  28.8454, 30.1910, 31.5264, 32.8523, 34.1696,
  35.5789, 36.8207, 38.0751, 39.3364, 40.6465,
  41.9232, 43.1944, 44.4607, 45.7222, 46.9992,
  48.23, 49.48, 50.72, 51.96, 53.20,
  54.44, 55.67, 56.89, 58.12, 59.3417,
  60.56, 61.77, 62.99, 64.20, 65.42,
  66.62, 67.82, 69.02, 70.22, 71.4202,
  72.61, 73.81, 75.00, 76.19, 77.38,
  78.50, 79.75, 80.93, 82.12, 83.32976,
  84.48, 85.65, 86.83, 88.00, 89.18,
  90.35, 91.52, 92.69, 93.86, 95.0231,
  96.19, 97.35, 98.52, 99.68, 100.84 /

DATA (CHI975(I), I = 1, 75) /
  102.00, 103.16, 104.31, 105.47, 106.629,
  107.78, 108.94, 110.09, 111.24, 112.40,
  113.54, 114.69, 115.84, 116.99, 118.136,
  119.28, 120.43, 121.57, 122.72, 123.86,
  125.00, 126.14, 127.28, 128.42, 129.561,
  130.70, 131.84, 132.98, 134.11, 135.25,
  136.38, 137.52, 138.65, 139.79, 140.92,
  142.05, 143.18, 144.31, 145.44, 146.57,
  147.70, 148.83, 149.96, 151.09, 152.21,
  153.34, 154.47, 155.59, 156.72, 157.84,
  158.97, 160.09, 161.21, 162.33, 163.46,
VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:

1 - 30, 40, 50, 60, 70, 80, 90, 100 — Theil, pp. 718-719


CALCULATED USING CUBE RULE APPROXIMATION

DATA T025 / 12.706, 1.303, 3.182, .776, 2.571, .447, .179, .228, 2.201, 2.365, 3.06, 2.262, 2.228, 2.201, 2.179, 2.160, 2.145, 2.131, 2.120, 2.110, 2.101, 2.093, 2.086, 2.080, 2.074, 2.069, 2.064, 2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /

LIST OF VARIABLES

ARG INTERMEDIATE CALCULATION VARIABLE
CHI025() TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHI975() TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHITAB MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
I CONTROLS LOOP FOR CHI025() AND CHI975()
IOUT OUTPUT DUMP CONTROLLER
IZER() 2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C FOR EACH REGION
JZERO() 2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY — MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP(I)-1))-1) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
NUM EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
NUMREG NUMBER OF REGIONS OF INTEREST
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SUMSQUARE2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SX EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
T VALUE OF T025() USED IN CALCULATIONS
T025() TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
TTAB MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C INITIALIZE IZERO, JZERO AND MCHAT

DO 50 L = 1, MAXREG
IZER(1,L) = 0.0
IZER(2,L) = 0.0
JZERO(1,L) = 0.0
JZERO(2,L) = 0.0
MCHAT(1,L) = 0.0
MCHAT(2,L) = 0.0
50 CONTINUE

C CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

DO 75 L = 1, NUMREG
IF (NPPR(L) .GT. CHITAB) THEN
WRITE(6,*) 'ERROR: EXCEEDED LIMIT ON DEGREES OF FREEDOM '
7 - 214
C ASSIGNED VALUES TO NUM, T, SWHAT, SUHAT AND THEN CALCULATE
C CONFIDENCE INTERVALS FOR EACH REGION

DO 100 L = 1, NUMREG
    NUM = NPPR(L)
    IF (NUM .LT. 31) THEN
        T = T025(NUM)
    ELSE
        T = T025(NUM)
    ENDIF
    SWHAT = SWHAT2(L) ** 0.5
    SUHAT = SUHAT2(L) ** 0.5
    SX = (NUM * SX2(L)) ** 0.5
    CALL TRMNAT
    75 CONTINUE
    C CALCULATE ESTIMATED VALUES OF M AND C
    ARG = T * SWHAT / SX
    MCHAT(1,L) = - DD(L)
    MCHAT(2,L) = SUHAT
    C CALCULATE CONFIDENCE INTERVALS
    IZERO(1,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI975(NUM)) ** 0.5
    IZERO(2,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI025(NUM)) ** 0.5
    JZERO(1,L) = MCHAT(1,L) - ARG
    JZERO(2,L) = MCHAT(1,L) + ARG
    IF (IOUT .EQ. i0) THEN
        WRITE(8,*) 'L =', L, ' NPPR =', NPPR(L), ' NUM =', NUM
        WRITE(8,*) 'SWHAT2 =', SWHAT2(L), ' SWHAT =', SWHAT
        WRITE(8,*) 'SUHAT2 =', SUHAT2(L), ' SUHAT =', SUHAT
        WRITE(8,*) 'SX2 =', SX2(L), ' SX =', SX
        WRITE(8,*) 'CHI025 =', CHI025(NUM), ' CHI975 =', CHI975(NUM)
        WRITE(8,*) 'T =', T, ' DD =', DD(L), ' ARG =', ARG
        WRITE(8,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
        & 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
        & 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
    ENDIF
100 CONTINUE
RETURN
END

C******************************************************************************
C SUBROUTINE GTPVAR CALCULATES THE EXTENT OF DEPARTURE FROM THE MULTIPLE
C HEAT MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 2JUN88 COMMENTS: 13JUL89

SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
C INPUTS: NSETS, NP, NUMREG, LAMN, MCHAT
C OUTPUTS: PVAR
C IMPLICIT NONE
INTEGER MAXREG, MAXSET
PARAMETER (MAXREG = 3, MAXSET = 5)

COMMON IOUT
INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG), &
   NUMREG, TOTAL
REAL LAMN, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM

LIST OF VARIABLES
C
C IOUT OUTPUT DUMP CONTROLLER
C J CONTROLS DO LOOP FOR EACH DATA SET
C L CONTROLS DO LOOP FOR EACH REGION
C LAMN LAMBDA-N = RATIO OF Var {ln N given S} / (m**2 c**2),
C CONSTANT OVER REGIONS AND COMPONENTS
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C SET IN EACH REGION
C NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
C NUM() EQUAL TO N_{j-1} FOR EACH REGION WHERE N_{j} IS THE SUM OF THE
C NUMBER OF POINTS IN EACH DATA SET
C NUMREG NUMBER OF REGIONS OF INTEREST
C PSIG2() 1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C VARIATION IN EACH REGION
C PVAR THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C CURVE WARRANTED BY THE AVAILABLE INFORMATION
C SUM WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
C AVERAGE
C TOTAL SUM OF NUM() OVER ALL REGIONS
C
C INITIALIZE VARIABLES
C
SUM = 0.0
TOTAL = 0.0
DO 50 L = 1, MAXREG
   PSIG2(L) = 0.0
   NUM(L) = 0
50 CONTINUE

DO 100 L = 1, NUMREG
   DO 150 J = 0, NSETS
      NUM(L) = NUM(L) + NP(J,L)
150 CONTINUE
   NUM(L) = NUM(L) - 1
   TOTAL = TOTAL + NUM(L)
100 CONTINUE

DO 200 L = 1, NUMREG
   PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
   SUM = SUM + PSIG2(L) * NUM(L)
200 CONTINUE

IF (IOUT .EQ. 10) THEN
   WRITE (8,*) 'LAMN = ', LAMN
   DO 300 L = 1, NUMREG
      WRITE (8,*) 'L = ', L, ' NUM = ', NUM(L)
   300 CONTINUE
   WRITE (8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
ENDIF

PVAR = SUM / FLOAT (TOTAL)
SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH M AND CO WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL) TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION

PROGRAMMER: L. NEWLIN
DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91

SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT, RANGEM)

INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
OUTPUTS: RANGEM
SUBPROGRAMS: TRM.NAT

IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
& MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
JZERO() 2-D ARRAY CONTAINING J0, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
LOWER LOWER BOUND OF INTERSECTION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION -- MCHAT(1,L) = DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C FOR M AND MCHAT(2,L) = SUIAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
UPPER UPPER BOUND OF INTERSECTION

INITIALIZE VARIABLES

DO 50 L = 1, MAXREG
  RANGEM(1,L) = 0.0
  RANGEM(2,L) = 0.0
50 CONTINUE

PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
DO 100 L = 1, NUMREG

IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'L = ', L, ' NUMREG = ', NUMREG,
  WRITE(8,*), ' MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
ENDIF

IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN
  THERE IS NO EXOGENOUS INFORMATION
  ASSUME RANGE TO BE 0
  RANGEM(1,L) = JZERO(1,L)
  RANGEM(2,L) = JZERO(2,L)

  IF (IOUT .EQ. 10) THEN
    WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
    & ' JZERO(1,L) = ', JZERO(1,L)
    WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,L),
    & ' JZERO(2,L) = ', JZERO(2,L)
  ENDIF
ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN
  THERE IS NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE TO Co, ADJUST THE LOWER BOUND OF 0 ACCORDINGLY
  LOWER = AMAX1(JZERO(1,L), MC(1,L))
  UPPER = JZERO(2,L)
  IF (UPPER .LT. LOWER) THEN
    WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN 0 AND MC'
    CALL TRMNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF
ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN
  THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE CORRESPONDING TO THE Co CONSTRAINT, ADJUST 0 ACCORDINGLY
  LOWER = AMAX1(JZERO(1,L), MC(1,L))
  UPPER = AMIN1(JZERO(2,L), MC(2,L))
  IF (UPPER .LT. LOWER) THEN
    WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN 0 AND MC'
    CALL TRMNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF
ELSEIF (MPNT(L) .EQ. 1) THEN
  THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
RANGEM(1,L) = MZERO(1,L)
RANGEM(2,L) = 0.0

IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
& RANGEM(1,L) = ', RANGEM(1,L),
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCNP(L) .EQ. 0)) THEN
THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT
USE INTERSECTION BETWEEN JO AND MO
LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
IF (UPPER .LT. LOWER) THEN
WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO AND MO'
CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCNP(L) .EQ. 1)) THEN
THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO CONSTRAINT
INTERSECT JO AND MO, ADJUSTING THE LOWER BOUND
BY MC ACCORDINGLY
LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
IF (UPPER .LT. LOWER) THEN
WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO, MO, ',
& ' AND MC'
CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCNP(L) .EQ. 2)) THEN
THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO CO CONSTRAINT
INTERSECT THESE TWO RANGES WITH JO
LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
IF (UPPER .LT. LOWER) THEN
WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO, MO, ',
& ' AND MC'
CALL TRMNAT
ELSE
ENDIF
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  & WRITE(8,*), 'JZERO(2,L) = ', JZERO(2,L),
  & WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
  & WRITE(8,*), 'MZERO(2,L) = ', MZERO(2,L),
  & WRITE(8,*), 'MC(I,L) = ', MC(I,L),
  & WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
  & WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
  & WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

ELSE
  WRITE(8,*), 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', I.
  CALL TRMNAT
ENDIF

C RESTRICT RANGE TO BE NON-NEGATIVE
RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)
IF (IOUT .EQ. 10) WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L)
100 CONTINUE

C CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE
DO 300 L = 1, NUMREG
  IF ((MCHAT(I,L) .LT. RANGEM(I,L))
  & .OR. (MCHAT(I,L) .GT. RANGEM(2,L)))
  & WRITE(8,*), 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
  & 'ON m IN REGION ', L
300 CONTINUE

RETURN
END

C***************************************************************

C SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C WITHOUT DATA
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91
C VERSION: MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.3,
V8.4, V8.5
C SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C INPUTS:  RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C OUTPUTS: RANGEM, MCHAT, NUMREG
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

7 - 220
REAL MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
LL CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS UPPER BOUND
NNODAT Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

IF (IOUT .EQ. 10) WRITE(8,*),'NUMREG = ', NUMREG

DO 100 L = 1, NNODAT
   NUMREG = NUMREG + 1
   LL = NUMREG
   IF (IOUT .EQ. 10) WRITE(8,*),'L = ', L, ' NUMREG = ', NUMREG,
   & ' LL = ', LL, ' MPNT(LL) = ', MPNT(LL)
   IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
      RANGEM(1,LL) = MZERO(1,LL)
      RANGEM(2,LL) = MZERO(2,LL)
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*),'RANGEM(1,LL) = ', RANGEM(1,LL),
         & ' MZERO(1,LL) = ', MZERO(1,LL),
         & 'RANGEM(2,LL) = ', RANGEM(2,LL),
         & ' MZERO(2,LL) = ', MZERO(2,LL)
      ENDIF
   ELSE
      SPECIFY E(M) OF POSTERIOR FOR SAKE OF
      CALCULATIONS IN SUBROUTINE EXPCTD
      IF (RANGEM(2,LL) .EQ. 0.0) THEN
         MCHAT(1,LL) = RANGEM(1,LL)
      ELSE
         MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
      ENDIF
      IF (IOUT .EQ. 10) WRITE(8,*), 'MCHAT = ', MCHAT(1,LL)
      ELSE
         WRITE(8,*), 'ERROR: OVERALL PRIOR RANGE INCORRECTLY ',
         & 'SPECIFIED IN REGION WITHOUT DATA'
      ENDIF
   ENDIF

100 CONTINUE
RETURN
END
SUBROUTINE CONCAV (NUMREG, RANGEM)

C INPUTS : NUMREG, RANGEM
C OUTPUTS : RANGEM
C SUBPROGRAMS : TRMNAT

C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL RANGEM(2, MAXREG), TESTM

C LIST OF VARIABLES
OUTPUT DUMP CONTROLLER
DO LOOP FOR EACH REGION
MAXIMUM NUMBER OF REGIONS ALLOWED
NUMBER OF REGIONS OF INTEREST
2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND
UPPER BOUND OF RANGE ON M IN REGION L-1 -- USED DURING
CONCAVITY ADJUSTMENT

C ADJUST RANGE TO INSURE CONCAVITY
DO 100 L = NUMREG, 2, -1
  IF (RANGEM(2,L-1) .EQ. 0.0) THEN
    RANGE IS A POINT IN REGION L-1
    IF (RANGEM(1,L-1) .GT. AMAX1(RANGEM(1,L), RANGEM(2,L))) THEN
      WRITE(8,*) 'ERROR: POSTERIOR INTERVAL IN REGION ', L,
      & ' IS INCONSISTENT WITH POINT POSTERIOR IN REGION ', L-1
      CALL TRMNAT
    ENDIF
  ELSE
    RANGE IS AN INTERVAL IN REGION L-1
    TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
    IF (TESTM .LT. RANGEM(1,L-1)) THEN
      WRITE(8,*) 'ERROR: POSTERIOR INTERVAL IN REGION ', L,
      & ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
      & 'REGION ', L-1
      CALL TRMNAT
    ELSE
      RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
    ENDIF
  ENDIF
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'RANGEM(1,L-1) = ', RANGEM(1,L-1),
    & 'RANGEM(2,L-1) = ', RANGEM(2,L-1)
    WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
    & 'RANGEM(2,L) = ', RANGEM(2,L)
    WRITE(8,*) 'TESTM = ', TESTM, ' L = ', L
  ENDIF
100 CONTINUE

RETURN
END
SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co

PROGRAMMER: L. NEWLIN
DATE: CODE: 5OCT87
COMMENTS: IDEC87


SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C INPUTS: NUMREG, RANGEM
C IOUTPUT: MEDM
C
IMPLICIT NONE
INTEGER MAXREG
PARAMETER ( MAXREG = 3 )
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)

LIST OF VARIABLES
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LOWERM LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

C INITIALIZE ARRAY MEDM
DO 50 L = 1, MAXREG
  MEDM(L) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS FOR EACH REGION
DO 100 L = 1, NUMREG
  IF (RANGEM(2,L) .EQ. 0.0) THEN
      MEDM(L) = RANGEM(1,L)
  ELSEIF (L .EQ. 1) THEN
      WE ARE IN REGION ONE -- NOT AFFECTED BY OTHER REGIONS -- MEDIAN WILL JUST BE AVERAGE OF RANGEM VALUES
      MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
  ELSE
      MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT
      LOWERM = AMAX1(RANGEM(1,L), MEDM(L-1))
      MEDM(L) = (LOWERM + RANGEM(2,L)) / 2.0
  END IF
100 CONTINUE
C SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N CURVE PARAMETERS
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C
C SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, &...)
C INPUTS: NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C OUTPUTS: BIGKI, BZHAT
C SUBPROGRAMS: TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO
C IMPLICIT NONE
C
C INTEGER MAXDAT, MAXREG
C PARAMETER (MAXDAT = 50, MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG
C REAL BIGK(0:MAXREG), BIGKI, BZHAT, FACTR, KHAT, MEANZ, &...;
C MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG), &...;
C NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG), &...;
C SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)
C
C LIST OF VARIABLES
C BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
C BIGK1 EQUAL TO BIGK(1)
C BZHAT E(BETAo)
C FACTR A SCALE FACTOR = PHI * KRATIO * Z
C IOUT OUTPUT DUMP CONTROLLER
C KHAT E(x)
C L CONTROLS DO LOOP FOR EACH REGION
C MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C MM() 1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
C NCOMPS Number of Components -- 1 FOR STRESS AND STRAIN WHEN DECOMPOSED DATA UNAVAILABLE -- 2 FOR DECOMPOSED STRAIN DATA
C NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
INITIALIZE VARIABLES

DO 50 L = 0, MAXREG
   MM(L) = 0.0
50 CONTINUE

CREATE MM() ARRAY FROM MEDM() ARRAY

DO 100 L = 1, NUMREG
   MM(L) = MEDM(L)
100 CONTINUE

TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

CALCULATE BETAo AND k

CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

CALCULATE BOUNDARIES OF STRESS REGIONS

CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

CALCULATE Ko AND Mo FOR THE NO DATA REGION TO THE LEFT IF REQUIRED

DO 150 L = ZROREG, NUMREG
   TRBIGK(L) = BIGK(L)
150 CONTINUE

IF (ZROREG .EQ. 0) THEN
   FACTR = 1.0
   CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK, FACTR, NUMREG)
ENDIF

WRITE RESULTS TO FILE

IF (NCOMPS .EQ. i) THEN
   WRITE(7,900) NUMREG, BZHAT, KHAT
   IF (IOUT .EQ. i0) WRITE(8,900) NUMREG, BZHAT, KHAT
   DO 200 L = ZROREG, NUMREG
      WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
200 CONTINUE
ENDIF
IF (IOUT .EQ. i0) WRITE(8,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
200 CONTINUE
WRITE(7,920)
ELSE
WRITE(7,930) MM(1), BIGK(1), KHAT
ENDIF
C FORMAT STATEMENTS
900 FORMAT(///,2X, 'PARAMETER VALUES FOR MEDIAN S/N CURVE',//,2X,
 & 'NUMBER OF REGIONS: ', I4,5X, 'E(BETA0) = ', F8.4, 5X, 'E(k) = ',
 & 'F8.4, ///, 2X, 'REGION', ', 'X, m', 15X, 'K', 9X, 'LIFE BOUND', 7X,
 & 'STRESS BOUND', //)
910 FORMAT(5X, I1, 5X, F9.5, 5X, EI2.5, 5X, E9.3, 9X, EL1.5)
920 FORMAT(///)
930 FORMAT(///, 2X, 'PARAMETER VALUES FOR MEDIAN S/N CURVE',,
 & ///, 11X, 'm', 14X, 'K', 13X, 'E(k)',
 & ///, 7X, F8.5, 5X, E12.5, 6X, F7.4, //)
RETURN
END
C SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION
C PARAMETERS: MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88 COMMENTS: 13JUL89
SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA,
 & MO, SIGMA2, MCHAT, MU, SIG)
C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2
C OUTPUTS: MCHAT, MU, SIG
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG, NPPR(MAXREG)
REAL ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
 & MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
 & SUHAT2(MAXREG), SUMX2, SWHAT2(MAXREG), SX2(MAXREG)
C LIST OF VARIABLES
ARG INTERMEDIATE CALCULATION VARIABLE
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
 & SIG() CALCULATION
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGION ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR

7 - 226
EACH REGION, BASED ON MATERIALS DATA ONLY - MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1)/-1) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SUMX2 EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
SUMX2 = NPPR(L) * SX2(L)
SIG() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)

C INITIALIZE ARRAYS
DO 50 L = I, MAXREG
  MCHAT(1,L) = 0.0
  MCHAT(2,L) = 0.0
  MU(L) = 0.0
  SIG(L) = 0.0
50 CONTINUE

C BEGIN CALCULATION FOR EACH REGION
DO 100 L = I, NUMREG
  MCHAT(1,L) = - DD(L)
  MCHAT(2,L) = SQRT (SUHAT2(L))
  SUMX2 = NPPR(L) * SX2(L)
  ARG = SUMX2 + DELTA(L)
  IF (DELTA(L).EQ.0.0) THEN
    THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
    USE THE ESTIMATE OF M
    MU(L) = MCHAT(1,L)
  ELSE
    UPDATE THE ESTIMATE OF M WITH MO USING DELTA
    MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
    ENDIF
  IF (SIGMA2(L).EQ.0.0) THEN
    THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
    USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
    SIG(L) = SQRT (SWHAT2(L) / ARG)
  ELSE
    SIG(L) = SQRT (SIGMA2(L) / ARG)
    ENDIF
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ', MCHAT(1,L), ' MCHAT2 = ', MCHAT(2,L), ' SUMX2 = ', SUMX2, ' Delta = ', DELTA(L), ' Arg = ', ARG,
    & WRITE(8,*) 'SUHAT2 = ', SUHAT2(L), ' MCHAT1 = ', MCHAT1(L), ' MCHAT2 = ', MCHAT2(L), ' SUMX2 = ', SUMX2, ' Delta = ', DELTA(L), ' Arg = ', ARG,
    & WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L), ' SUHAT2 = ', SUHAT2(L), ' MCHAT1 = ', MCHAT1(L), ' MCHAT2 = ', MCHAT2(L), ' SUMX2 = ', SUMX2,
    & WRITE(8,*) ' Delta = ', DELTA(L), ' Arg = ', ARG
    ENDIF
100 CONTINUE

RETURN
SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND CO TO OBTAIN POSTERIOR RANGES ON M FOR EACH REGION

PROGRAMMER: L. NEWLIN
DATE: CODE: 10FEB88 FORMAT/COMMENTS: 12AUG91

SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
OUTPUTS: RANGEM
SUBPROGRAMS: TRMNAT

IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LOWER LOWER BOUND OF INTERSECTION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
UPPER UPPER BOUND OF INTERSECTION

C INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
RANGEM(1,L) = 0.0
RANGEM(2,L) = 0.0
50 CONTINUE
C PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
DO 100 L = 1, NUMREG
IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'L' = , L, ' NUMREG = ', NUMREG
WRITE(8,*), 'MPNT ' = , MPNT(L), ' MCPNT ' = , MCPNT(L)
ENDIF
ENDIF

IF (MPNT(L) .EQ. 1) THEN

THERE IS A POINT PRIOR ON M — THIS OVERRIDES ALL OTHER
INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

RANGEM(1,L) = MZERO(1,L)
RANGEM(2,L) = 0.0

IF (IOUT .EQ. 10) THEN
WRITE (*,*) 'MZERO(1,L) = ', MZERO(1,L),
&
WRITE (*,*) 'RANGEM(1,L) = ', RANGEM(1,L),
&
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT USE MO

RANGEM(1,L) = MZERO(1,L)
RANGEM(2,L) = MZERO(2,L)

IF (IOUT .EQ. 10) THEN
WRITE (*,*) 'MZERO(1,L) = ', MZERO(1,L),
&
WRITE (*,*) 'RANGEM(1,L) = ', RANGEM(1,L),
&
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN

THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO
CONSTRAINT ADJUST THE LOWER BOUND OF MO BY MC

LOWER = AMAX1(MZERO(1,L), MC(1,L))
UPPER = MZERO(2,L)

IF (UPPER .LT. LOWER) THEN
WRITE (*,*) 'ERROR: NO INTERSECTION BETWEEN MO AND MC'
CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

IF (IOUT .EQ. 10) THEN
WRITE (*,*) 'MZERO(1,L) = ', MZERO(1,L),
&
WRITE (*,*) 'MC(1,L) = ', MC(1,L),
&
WRITE (*,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER,
&
Write (*,*) 'RANGEM(1,L) = ', RANGEM(1,L),
&
RANGEM(2,L) = ', RANGEM(2,L),
&
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN

THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO CO CONSTRAINT
INTERSECT THESE TWO RANGES

LOWER = AMAX1(MZERO(1,L), MC(1,L))
UPPER = AMIN1(MZERO(2,L), MC(2,L))

IF (UPPER .LT. LOWER) THEN
WRITE (*,*) 'ERROR: NO INTERSECTION BETWEEN MO AND MC'
CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

IF (IOUT .EQ. 10) THEN
WRITE (*,*) 'MZERO(1,L) = ', MZERO(1,L),
&
WRITE (*,*) 'MC(1,L) = ', MC(1,L),
&
WRITE (*,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER,
&
WRITE (*,*) 'RANGEM(1,L) = ', RANGEM(1,L),
&
RANGEM(2,L) = ', RANGEM(2,L),
&
ENDIF
ENDIF

ELSE

WRITE(8,*) 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
CALL TRMNAT

ENDIF

C

RESTRICT RANGE TO BE NON-NEGATIVE

RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

IF (IOUT .EQ. I0) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

100 CONTINUE

C

CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

DO 300 L = i, NUMREG

IF ((MCHAT(I,L) .LT. RANGEM(I,L))
& .OR. (MCHAT(I,L) .GT. RANGEM(2,L)))
& WRITE(8,*) 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
& 'ON m IN REGION ', L

300 CONTINUE

RETURN

END

C*********************************************************

SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 10FEB88 FORMAT/COMMENTS: 12AUG91
C SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
& MZERO, MPNT, MO, SIGMA2)
C INPUTS: RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C OUTPUTS: RANGEM, MCHAT, MU, SIG, NUMREG
C IMPLICIT NONE
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG
C REAL MCHAT(2, MAXREG), MO(MAXREG), MU(MAXREG),
& MZERO(2, MAXREG), RANGEM(2, MAXREG), SIG(MAXREG),
& SIGMA2(MAXREG)
C
C LIST OF VARIABLES
C IOUT OUTPUT DUMP CONTROLLER
C L CONTROLS DO LOOP FOR EACH REGION
EQUAL TO NUMREG FOR A SET OF CALCULATIONS

MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED

MCHAT()  2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
          C FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
          MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
          MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C

MO()  1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
      MEAN FOR EACH REGION

MPNT()  1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
        MZERO() FOR EACH REGION

MU()  1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
      DISTRIBUTION MEAN FOR EACH REGION

MZERO()  2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
         EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
         IS UPPER BOUND

NNODAT  NUMBER OF NO DATa regions (REGIONS WITHOUT ANY S/N DATA)

NUMREG  NUMBER OF REGIONS OF INTEREST

RANGEM()  2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
          FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
          RANGEM(2,L) IS THE UPPER BOUND

SIG()  1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
       DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

SIGMA2()  1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
          VARIANCE FOR EACH REGION

IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

DO 100 L = 1, NNODAT
    NUMREG = NUMREG + 1
    LL = NUMREG
    IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
    & ' LL =', LL, ' MPNT(LL) =', MPNT(LL)
    IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
        POSTERIOR ON M IS SAME AS PRIOR ON M
        RANGEM(1,LL) = MZERO(1,LL)
        RANGEM(2,LL) = MZERO(2,LL)
        MU(LL) = MO(LL)
        SIG(LL) = SQRT(SIGMA2(LL))
        IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
            & ' MZERO(1,LL) =', MZERO(1,LL),
            & ' RANGEM(2,LL) =', RANGEM(2,LL),
            & ' MZERO(2,LL) =', MZERO(2,LL),
            & ' MU(LL) =', MU(LL), ' MO(LL) =', MO(LL),
            & ' SIG(LL) =', SIG(LL), ' SIGMA2(LL) =',
            & SIGMA2(LL)
        ENDIF
    ELSE
        SPECIFY E(M) OF POSTERIOR FOR SAKE OF
        CALCULATIONS IN SUBROUTINE EXPCTD
        IF (RANGEM(2,LL) .EQ. 0.0) THEN
            MCHAT(1,LL) = RANGEM(1,LL)
            SIG(LL) = RANGEM(1,LL)
        ELSE
            MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
        ENDIF
        IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
        & ' MU = ', MU(LL), ' SIG = ', SIG(LL)
        ELSE
            WRITE(8,*) 'ERROR: OVERALL PRIOR RANGE INCORRECTLY '
            & 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
    ENDIF
100 CONTINUE
RETURN
SUBROUTINE EXPB CALCULATES THE EXPECTED VALUES OF THE S/N CURVE PARAMETERS FOR THE BOOTSTRAP IMPLEMENTATION

PROGRAMMER: L. NEWLIN
DATE: 11OCT90  COMMENTS: 15JAN92
VERSION: MATCHR VB1.1, VB1.2, VB1.3
MATRM VB1, VB1.1

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

SUBROUTINE EXPB (MM, BIGK, SZERO, NUMREG, ZROREG, NBND)

INPUTS: MM, BIGK, SZERO, NUMREG, ZROREG, NBND
SUBPROGRAMS: FINDSB, KOMO

IMPLICIT NONE

INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)

COMMON I OUT
INTEGER IOUT, L, NUMREG, ZROREG

REAL BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
& SBND(0:MAXREG), SZERO, TRBIGK(0:MAXREG)

LIST OF VARIABLES
I-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
FACTR A SCALE FACTOR = PHI * KRATIO * Z
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM(0:MAXREG) 1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
NBND(0:MAXREG) 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS OF INTEREST
SBND(0:MAXREG) 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
SZERO STRESS TENSILE TEST POINT, SO CONTAINED IN NBND()
TRBIGK(0:MAXREG) 1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
ZROREG ZERO REGION VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE = 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

CALCULATE BOUNDARIES OF STRESS REGIONS
CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

CALCULATE Ko AND Mo FOR THE NO DATA REGION TO THE LEFT IF REQUIRED
DO 150 L = ZROREG, NUMREG
TRBIGK(L) = BIGK(L)
150 CONTINUE
IF (ZROREG .EQ. 0) THEN
FACTR = 1.0
CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK, FACTR, NUMREG)
ENDIF

WRITE RESULTS TO FILE

7 - 232
WRITE(7,900) NUMREG
IF (IOUT .EQ. 10) WRITE(8,900) NUMREG
DO 200 L = ZROREG, NUMREG
    WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
    IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
                   & NBND(L), SBND(L)
200 CONTINUE
WRITE(7,920)
C FORMAT STATEMENTS
900 FORMAT(///.2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
       & ///,2X,'NUMBER OF REGIONS:',I4,
       & ///,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,
       & 'STRESS BOUND',/)
910 FORMAT(5X, Ii, 5X, F9.5,5X, El2.5, 5X,E9.3,9X, E11.5)
920 FORMAT(///)
RETURN
END

C***SUBROUTINE PEB CONTROLS THE CALCULATIONS FOR THE Parameter Estimation
C MODEL PORTION OF THE MATERIALS CHARACTERIZATION Bootstrap MODEL
C PROGRAMMER: L. NEWLIN
C DATE: 13NOV90 FORMAT/COMMENTS: 15JAN92
C VERSION: MATCHR VBI.2, VBI.3
C MATGRM VBI,
C VERSION:
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

SUBROUTINE PEB (NPTS, NUMREG, ZROREG, RAND, NBND, STR, MEDM,
 & MEDK, RESID, BIGK, BZERO, MM, SBND)
 C INPUTS: NPTS, NUMREG, ZROREG, RAND, NBND, STR, MEDM, MEDK, RESID
 C OUTPUTS: BIGK, BZERO, MM, SBND
 C SUBPROGRAMS: PIRCRS, MREGR, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB
 C IMPLICIT NONE
 INTEGER MAXDAT, MAXREG
 PARAMETER (MAXDAT = 50, MAXREG = 3)
 COMMON IOUT
 INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, ZROREG
 REAL BIGK(0:MAXREG), BZERO, K, MEANZ, MEDM(0:MAXREG),
 & MEDK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
 & RESID(MAXDAT), RESNF(MAXDAT, MAXREG), SBND(0:MAXREG),
 & STR(MAXDAT, MAXREG), ZZ2, ZZ(MAXDAT)
 DOUBLE PRECISION RAND

LIST OF VARIABLES

BIGK()  1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
        EACH REGION
BZERO  VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING S/N DATA SET
IOUT  OUTPUT DUMP CONTROLLER
VALUE OF k -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL DATABASE

L -- CONTROLS DO LOOP FOR EACH REGION
MAXDAT -- MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG -- MAXIMUM NUMBER OF REGIONS ALLOWED
MEANZ -- SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
MEDK() -- 1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION
(BOOTSTRAP OPTION)
MEDM() -- 1-D ARRAY CONTAINING THE MEAN VALUES M FOR EACH REGION
(BOOTSTRAP OPTION)
MEDK() -- 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() -- 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NP -- TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
NPTS() -- 1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET
NUMREG -- NUMBER OF REGIONS OF INTEREST
RAND -- RANDOM NUMBER SEED
RESID() -- 1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH POINT IN THE SPECIFIC MATERIAL S/N DATA SET
RESNF() -- 2-D ARRAY CONTAINING NF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS BASED ON THE RANDOMLY SELECTED RESIDUALS
SBND() -- 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()
STR() -- 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
S22 -- SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
ZROREG -- ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
ZZ() -- 1-D ARRAY CONTAINING TRANSFORMED S/N DATA, Z = F(STR,NF,NBND,MM)

OBTAIN THE VALUES OF THE RANDOMLY SELECTED RESIDUALS FOR EACH DATA POINT
CALL PICRES (RAND, MEDM, MEDK, RESID, NPTS, STR, RESNF)

BOOTSTRAPPING M IS DESIRED
CALL MREGR (NUMREG, NPTS, STR, RESNF, MM)

TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)
CALL TRNSFM (NPTS, STR, RESNF, NUMREG, MM, NBND, NP, ZZ)

CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)
CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

CALCULATE THE VALUES FOR k AND BETAo FROM THE SAMPLE MEAN AND VARIANCE
CALL KBETA (MEANZ, SZ2, K, BZERO)

CALCULATE THE VALUE OF k FOR EACH REGION WHERE A = K ** M
CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

CALCULATE STRESS TIE-POINTS
CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

WRITE RESULTS TO FILE
CALL WRITE(7,900) NUMREG, BZERO
DO 200 L = ZROREG, NUMREG
CALL WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
200 CONTINUE
CALL WRITE(7,920)
FORMAT STATEMENTS
C SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE UNIFORMLY DISTRIBUTED RANDOM NUMBERS


PROGRAMMER: L. GRONDALSKI, L. NEWLIN
DATE: IDEC87

C******************************************************************************

SUBROUTINE RANDOM (FRAC, RAND)
IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL FRAC
DOUBLE PRECISION RANA, RANC, RAND, RANDIV, RANM, RANSUB,
& RANT, RANX

C LIST OF VARIABLES
FRAC UNIFORM (0, 1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
RANA CONSTANT FOR LCG
RANC CONSTANT FOR LCG
RAND RANDOM NUMBER SEED
RANDIV INTERNAL CALCULATION
RANM CONSTANT FOR LCG
RANSUB INTERNAL CALCULATION
RANT INTERNAL CALCULATION
RANX INTERNAL CALCULATION

C USING LCG RANDOM # GENERATOR
RANA = 671093.0
RANC = 7090885.0
RANM = 33554432.0

10 RANX = RANA * RAND + RANC
RANDIV = RANX / RANM
RANT = DINT(RANDIV)
RANSUB = RANT * RANM
RAND = RANX - RANSUB
FRAC = SNGL(RAND / RANM)

IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
IF (IOUT .EQ. 2) WRITE(8,*),'RANX =', RANX, ' RANDIV =', RANDIV,
SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C SUBPROGRAM: RANDOM
C IMPLICIT NONE
C COMMON IOUT
C DOUBLE PRECISION RAND
C REAL FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2
C PARAMETER (PI = 3.1415926536)
C INTEGER IOUT

LIST OF VARIABLES

FRAC UNIFORM(0,1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
MU MEAN OF NORMAL DISTRIBUTION
RAND RANDOM NUMBER SEED
SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION
X NORMAL RANDOM VARIATE
U1 UNIFORM RANDOM NUMBER U(0,1)
U2 UNIFORM RANDOM NUMBER U(0,1)
Z1 NORMAL RANDOM NUMBER ON N(0,1)
Z2 NORMAL RANDOM NUMBER ON N(0,1)

IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*)'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA
CALL RANDOM (FRAC, RAND)
U1 = FRAC
CALL RANDOM (FRAC, RAND)
U2 = FRAC
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*)'U1 =', U1, ' U2 =', U2
Z1 = SQRT (-(2. * ALOG(U1)) * COS(2. * PI * U2))
Z2 = SQRT (-(2. * ALOG(U1)) * SIN(2. * PI * U2))
X = SIGMA * Z1 + MU
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15)),
& WRITE(8,*), '21 = ', 21, ' 22 = ', 22, ' X = ', X

RETURN
END

C***************************************************************

C SUBROUTINE PICRES PERFORMS THE RESIDUAL SELECTION AND THEN CALCULATES
C THE NEW S/N PAIRS
C PROGRAMMER: L. NEWLIN
C DATE: 10OCT90
C VERSION: MATCHR VBI.1, VBI.2, VBI.3
C MAXGRM VBI, VBI.1
C
C SUBROUTINE PICRES (RAND, MEDM, MEDK, RESID, NPTS, STR, RESNF)
C INPUTS: RAND, MEDM, MEDK, RESID, NPTS, STR
C OUTPUTS: RESNF
C
C IMPLICIT NONE
INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
INTEGER I, INDEX, IOUT, K, L, NPTS(MAXREG)
REAL LNK, MEDK(0:MAXREG), MEDM(0:MAXREG), RESID(MAXDAT),
& RESNF(MAXDAT, MAXREG), STR(MAXDAT, MAXREG), X

DOUBLE PRECISION RAND

LIST OF VARIABLES

I CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
INDEX THE RANDOMLY SELECTED INDEX FOR THE RESIDUAL SELECTION
K CONTROLS DO LOOP FOR EACH POINT IN EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
LNK EQUAL TO ln(K)
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MEDK() 1-D ARRAY CONTAINING THE MEDIAN K FOR EACH REGION
MEDM() 1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
NPTS() 1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
REFERENCE) MATERIAL S/N DATA SET
RAND RANDOM NUMBER SEED
RESID() 1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
POINT IN THE SPECIFIC MATERIAL S/N DATA SET
RESNF() 2-D ARRAY CONTAINING THE CALCULATED CYCLES TO FAILURE FOR THE
SPECIFIC MATERIAL S/N DATA SET AND SELECTED RESIDUALS
STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
S/N DATA SET BROKEN INTO REGIONS (PSI)
X UNIFORM(0,1) RANDOM VARIATE

DO 50 L = 1, MAXREG
  DO 75 K = 1, MAXDAT
    RESNF(K,L) = 0.0
    75 CONTINUE
  CONTINUE

LNK = ALOG (MEDK(1))

IF ((IOUT .EQ. 10) THEN
  WRITE(8,*), 'MEDK = ', MEDK(1), ' LNK = ', LNK
  WRITE(8,*), 'NPTS = ', NPTS(1), ' MEDM = ', MEDM(1)

7 - 237
C SUBROUTINE MREGR CALCULATES, M, THE MATERIALS SHAPE PARAMETER
FOR EACH REGION WHERE Y = LN(NF) AND X = LN(STR)

PROGRAMMER: L. NEWMAN

DATE: 13NOV90

VERSION: MATCHRB VB1.2, VB1.3

SUBROUTINE MREGR (NUMREG, NPTS, STR, NF, MM)

C INPUTS: NUMREG, NPTS, STR, NF
C OUTPUTS: MM

C IMPLICIT NONE

INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)

COMMON IOUT
INTEGER IOUT, K, L, NPTS(MAXREG), NUMREG

REAL DIFFX(MAXDAT), DIFFY(MAXDAT), MM(0:MAXREG),
& NF(MAXDAT, MAXREG), STR(MAXDAT, MAXREG), MEANX, MEANY,
& SX2(MAXREG), SXY(MAXREG)

LIST OF VARIABLES

DIFFX() 1-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LN(STR(K,L))
AND MEANX FOR EACH POINT IN REGION L
DIFFY() 1-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LN(NF(K,L))
AND MEANX FOR EACH POINT IN REGION L
IOUT OUTPUT DUMP CONTROLLER
K CONTROLS DO LOOP FOR EACH POINT IN REGION L
L CONTROLS DO LOOP FOR EACH REGION
MAXDAT MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
MEANX SAMPLE X MEAN FOR POINTS FROM REGION L (X = Ln S)
MEANY SAMPLE Y MEAN FOR POINTS FROM REGION L (Y = Ln N)
MM() 1-D ARRAY CONTAINING VALUES OF THE MEAN M FOR EACH REGION
(N = BOOTSTRAP OPTION)
NF() 2-D ARRAY CONTAINING LN(RESNF()), ALSO INDEXED FOR REGION
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
STR() 2-D ARRAY CONTAINING LN(STR()), ALSO INDEXED FOR REGION
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(X = Ln S
SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
EACH REGION (X = Ln S, Y = Ln N)
C INITIALIZE ARRAYS
   DO 50 L = 1, MAXREG
      SX2(L) = 0.0
      SXY(L) = 0.0
      MM(L) = 0.0
   50 CONTINUE
   DO 70 K = 1, MAXDAT
      DIFFY(K) = 0.0
      DIFFX(K) = 0.0
   70 CONTINUE
C NOW PERFORM CALCULATION OF SX2 AND SXY, FOR EACH REGION
   DO 100 L = 1, NUMREG
      C FIRST CALCULATE SAMPLE X AND Y MEANS IN REGION L
      MEANY = 0.0
      MEANX = 0.0
      IF (IOUT .EQ. 10) WRITE(8,*) 'L = ', L, ' NPTS = ', NPTS(L)
      DO 250 K = 1, NPTS(L)
         MEANY = MEANY + ALOG(NF(K,L))
         MEANX = MEANX + ALOG(STR(K,L))
      IF (IOUT .EQ. 10) WRITE(8,*) 'NF = ', NF(K,L), ' STR = ', STR(K,L)
      250 CONTINUE
      MEANY = MEANY/FLOAT(NPTS(L))
      MEANX = MEANX/FLOAT(NPTS(L))
      IF (IOUT .EQ. 10) WRITE(_,*) 'MEANY = ', MEANY,
      ' MEANX = ', MEANX
      C NOW CALCULATE SAMPLE VARIANCES, SX2 AND SXY,
      OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
      DATA POINT IN REGION L
      DO 300 K = 1, NPTS(L)
         DIFFY(K) = ALOG(NF(K,L)) - MEANY
         DIFFX(K) = ALOG(STR(K,L)) - MEANX
         SX2(L) = SX2(L) + DIFFX(K) ** 2
         SXY(L) = SXY(L) + DIFFX(K) * DIFFY(K)
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'K = ', K, ' DIFFY(K) = ', DIFFY(K),
         ' DIFFX(K) = ', DIFFX(K)
         WRITE(8,*) 'SX2(L) = ', SX2(L), ' SXY(L) = ', SXY(L)
      ENDIF
      300 CONTINUE
      IF (SXY(L) .GE. 0.0) THEN
         C LIFE WILL INCREASE WITH INCREASING STRESS -- INVALID FOR
         C OUR MODEL
         CALL TRMNAT
      ENDIF
      C NOW CALCULATE THE M FOR REGION L
      MM(L) = - SXY(L) / SX2(L)
      IF (IOUT .EQ. 10) WRITE(8,*) 'SX2(L) = ', SX2(L), ' SXY(L) = ',
      ' MM(L) = ', MM(L)
   100 CONTINUE
   RETURN
END


SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM THE S/N DATA INTO THE VARIABLE Z = ln(X)

C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88  COMMENTS: 13JUL89
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

C INPUTS: NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ
C OUTPUTS: NP, ZZ
C IMPLICIT NONE

INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG
REAL MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
& STR(MAXDAT, MAXREG), ZZ(MAXDAT)

C LIST OF VARIABLES
C I OUT CONTROLS DO LOOP FOR EACH DATA POINT
C IOUT OUTPUT DUMP CONTROLLER
C K CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C L CONTROLS DO LOOP FOR EACH REGION
C LL CONTROLS INNER DO LOOP FOR EACH REGION
C MAXDAT MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MM() 1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C MML EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
C NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C NPTS() 1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET
C NUMREG NUMBER OF REGIONS OF INTEREST
C STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C ZZ() 1-D ARRAY CONTAINING TRANSFORMED S/N DATA, Z = F(STR, NF, NBND, MM)

C INITIALIZE VARIABLES
NP = 0
DO 50 I = 1, MAXDAT
  ZZ(I) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS
DO 100 L = 1, NUMREG
  MML = MM(L)
  IF (IOUT .EQ. 10) WRITE(8, *) 'L = ', L, ' MM = ', MM(L), ' MML = ', MML, ' NPTS = ', NPTS(L)
  NP = NP + 1
  ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
  IF (IOUT .EQ. 10) WRITE(8, *) 'K = ', K, NP = ', NP = ', NP, ' NF = ', NF(K,L), ' STR = ', STR(K,L), ' ZZ = ', ZZ(NP)
100 CONTINUE
DO 300 LL = 2, L

7 - 240
SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)

C INPUTS: NP, ZZ
C OUTPUTS: MEANZ, SZ2
C IMPLICIT NONE
INTEGER MAXDAT
PARAMETER (MAXDAT = 50)
COMMON IOUT
INTEGER I, IOUT, NP
REAL MEANZ, SZ2, ZZ(MAXDAT)

LIST OF VARIABLES

I CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT OUTPUT DUMP CONTROLLER
MAXDAT MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
SZ2 SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
ZZ() 1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
 Z = F(STR, NF, NBND, MM)

C INITIALIZE VARIABLES
MEANZ = 0.0
SZ2 = 0.0

C CALCULATE THE MEAN OF ZZ(), MEANZ
DO 100 I = 1, NP
  MEANZ = MEANZ + ZZ(I)
  IF (IOUT .EQ. I0) WRITE(8,*)' I = ', I,
   & ' ZZ = ', ZZ(I), ' MEANZ = ', MEANZ
100 CONTINUE
MEANZ = MEANZ / FLOAT(NP)
IF (IOUT .EQ. I0) WRITE(8,*)' MEANZ = ', MEANZ

C******************************************************************************

C SUBROUTINE SMNVAR CALCULATES THE Sample Mean and Variance OF
C Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 24AUG87 COMMENTS: 13JUL89
C VERSION: MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C V8.3, V8.4, V8.5

C******************************************************************************

C SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)
C CALCULATE THE VARIANCE OF ZZ(), SZ2

DO 200 I = 1, NP
   SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
IF (IOUT .EQ. 10) WRITE(8,*)' I = ', I, ', SZ2 = ', SZ2
200 CONTINUE
   SZ2 = SZ2 / FLOAT(NP - 1)
IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 = ', SZ2
RETURN
END

C***************************************************************

C SUBROUTINE KBETA CALCULATES k AND BETAO FROM THE SAMPLE MEAN AND
C VARIANCE OF Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87   COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C          V8.4, V8.5
C SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)
C INPUTS: MEANZ, SZ2
C OUTPUTS: K, BZERO
C IMPLICIT NONE
REAL PI
PARAMETER (PI = 3.1415926536)
COMMON IOUT
INTEGER IOUT
REAL MEANZ, SZ, SZ2
C
C LIST OF VARIABLES
C
C BZERO   VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE
C          SPECIFIC MATERIAL S/N DATA SET
C IOUT    OUTPUT DUMP CONTROLLER
C K       VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL
C          DATA BASE
C MEANZ   SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C PI      SELF EXPLANATORY CONSTANT
C SZ      SZ2 ** 0.5
C SZ2     SAMPLE VARIANCE OF THE TRANSFORMED DATA,
C          Z = F(STR, NF, NBND, MM)
C
C PERFORM CALCULATIONS

   SZ = SZ2 ** 0.5
   BZERO = PI / (SZ * (6.0 ** 0.5))
   K = MEANZ

C DATA DUMP STATEMENTS

   IF (IOUT .EQ. 10) THEN
      WRITE(8,*)' SZ2 = ', SZ2, ', SZ = ', SZ
SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C INPUTS: BZERO, K, MM, NBND, NUMREG
C OUTPUTS: BIGK
C IMPLICIT NONE

INTEGER MAXREG
REAL GAMMA
PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL BIGK(0:MAXREG), BZERO, K, MM(0:AXREG), NBND(0:MAXREG)

C LIST OF VARIABLES

I-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
FOR EACH REGION
VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
MATERIAL DATA BASE
Euler’s CONSTANT
OUTPUT DUMP CONTROLLER
VALUE OF k -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
DATA BASE
CONTROLS DO LOOP FOR EACH REGION
MAXIMUM NUMBER OF REGIONS ALLOWED
1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
REGIONS OF INTEREST
NUMBER OF REGIONS OF INTEREST

C INITIALIZE VARIABLES

DO 50 L = 0, MAXREG
BIGK(L) = 0.0
50 CONTINUE

C CALCULATE K FOR REGION ONE

BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C WRITE(7,*),'REGION: i, K =', BIGK(1)
C IF (IOUT .EQ. i0) WRITE(8,*),'BZERO =', BZERO, ' k =', K,
    & ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)

C CALCULATE K FOR REMAINING REGIONS

DO 100 L = 2, NUMREG
    BIGK(L) = BIGK(L-1) * NBND(L-1) ** ((1.0 / MM(L)) - (1.0 / MM(L-1)))
100 CONTINUE
C
   WRITE(7,*) 'REGION ', L, ' K = ', BIGK(L)
   IF (IOUT .EQ. 10) WRITE(8,*) 'L = ', L, ' NBND(L-1) = ', NBND(L-1),
     & ' MM(L-1) = ', MM(L-1), ' MM(L-1) = ', MM(L-1),
     & ' BIGK(L) = ', BIGK(L)
100 CONTINUE
RETURN
END

C*************************************************************************

C SUBROUTINE FINDSB CALCULATES THE REGION 'TIE-POINTS' — THE STRESS
C VALUES WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE
C RANDOMLY SELECTED MS, AND THE KS CALCULATED FROM THE BETA AND K
C CHARACTERIZING SPECIFIC MATERIAL
C PROGRAMMER: L. NEWLIN
C DATE: 22DEC88
C VERSION: MATCHR V8.2, V8.4, V8.5

SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C INPUTS: NUMREG, ZROREG, NBND, BIGK, MM
C OUTPUTS: SBND
C
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG, ZROREG
REAL BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
& SBND(0:MAXREG)

LIST OF VARIABLES
BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
FOR EACH REGION
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS OF INTEREST
SBND() 1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
REGION CONTAINED IN NBND()
ZROREG ZERO REGION — VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE — 0 — ZERO REGION EXISTS, 1 — NO REGION

C INITIALIZE SBND()
   DO 50 L = 0, MAXREG
      SBND(L) = 0.0
   50 CONTINUE
C CALCULATE SBND(0) IF ZROREG = 0
   IF (ZROREG .EQ. 0) THEN
      SBND(0) = BIGK(1) * NBND(0) ** (-1.0 / MM(1))
   ENDIF
C CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES

7 - 244
DO 100 L = 1, NUMREG
   IF (NBND(L) .GE. 1.0E+36) THEN
      SBND(L) = 0.0
   ELSE
      SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
   ENDIF
100 CONTINUE
RETURN
END

C*********************************************************
SUBROUTINE KOMO
CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
PROGRAMMER : L. NEWLIN
DATE: IJAN91
VERSION: MATCHR V8.5 MATGRM V4.5

C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK, FACTR, NUMREG)
C INPUTS: SZERO, BIGK, MM, NBND, TRSBND, FACTR
C OUTPUTS: TRBIGK, MM, TRSBND
C IMPLICIT NONE

INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1 SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)

LIST OF VARIABLES
BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
FACTR SCALE FACTOR = PHI * KRATIO * Z
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS
SCLK ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
SZERO STRESS TENSILE TEST POINT, S0
TRBIGK() 1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP SBND(0) < S0 FOR EACH TRIAL
TRSBND() 1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
ADJUSTED BY VARIATION PARAMETERS FOR EACH TRIAL

BIGK(0) = SZERO
IF (TRSBND(0) .GT. SZERO) THEN
   SCLK = SZERO/TRSBND(0)
   DO 100 L = 0, NUMREG
   ENDIF
RETURN
END
SUBROUTINE WORSTN FINDS THE WORST OF N FOR BOTH THE WEIBULL AND LOGNORMAL DISTRIBUTIONS

PROGRAMMER: L. NEWLIN
DATE: 14NOV90
VERSION: MATCHR VB1.2, VB1.3
MATGRM VB1, VB1.1

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

SUBROUTINE WORSTN (RAND, NSYM, BZERO, MM, EPSW, EPSL)

C INPUT: RAND, NSYM, BZERO, MM
C OUTPUT: EPSW, EPSL
C ROUTINE: RANDOM

C IMPLICIT NONE

COMMON IOUT
INTEGER IOUT, MAXREG, NSYM
PARAMETER (MAXREG = 3)
REAL BZERO, C0, C1, C2, D1, D2, D3, EPSW, EPSL, F,
& MM(0:MAXREG), P, P0, SIGMA, T, T2, T3, X
DOUBLE PRECISION RAND

LIST OF VARIABLES

BZERO  WEIBULL SHAPE PARAMETER, BETA
C0, C1, C2, D1, D2, D3  COEFFICIENTS OF FUNCTION FOR LOGNORMAL DISTRIBUTION CALCULATIONS
EPSL  LOGNORMAL(0, SIGMA**2) WORST OF NSYM RANDOM VARIATE
EPSW  WEIBULL(BZERO) WORST OF NSYM RANDOM VARIATE
F  UNIFORM(0,1) RANDOM VARIATE, VALUE OF CDF
IOUT  OUTPUT DUMP CONTROLLER
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MM()  1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
NSYM  SYMMETRY NUMBER
P0  VALUE OF P USED TO CHECK/INSURE P>.5
P  INTERMEDIATE CALCULATION VARIABLE FOR LOGNORMAL DISTRIBUTION CALCULATIONS
SIGMA  STANDARD DEVIATION OF THE LOGNORMAL DISTRIBUTION
T  INTERMEDIATE CALCULATION VARIABLE FOR LOGNORMAL DISTRIBUTION CALCULATIONS
T2  EQUAL TO T**2
C T3 EQUAL TO T**3
C X NORMAL(0,SIGMA**2) WORST OF NSYM RANDOM VARIATE

C0 = 2.515517
C1 = 0.802853
C2 = 0.010328
D1 = 1.432788
D2 = 0.189269
D3 = 0.001308
SIGMA = 1.282550 * MM(1) / BZERO

CALL RANDOM (F, RAND)
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'BZERO = ', BZERO, ' SIGMA = ', SIGMA
  WRITE(8,*), 'F = ', F, ' MM = ', MM(1)
ENDIF

EPSW = EXP ((LOG (- LOG(1.0 - F) / NSYM))
& (- LOG (LOG (2.0))) * MM(1) / BZERO)
IF (IOUT .EQ. 10) WRITE(8,*), 'EPSW = ', EPSW

P0 = (1.0 - F)**(1.0/FLOAT(NSYM))
IF (P0 .LE. 0.5) THEN
  P = P0
  T = SQRT (LOG (1.0 / P ** 2))
  IF (IOUT .EQ. 10) WRITE(8,*), 'P = ', P, ' T = ', T
  T2 = T**2
  T3 = T**3
  X = T - ((C0 + C1 * T + C2 * T2)&
  & / (1.0 + D1 * T + D2 * T2 + D3 * T3))
ENDIF
ELSE
  P = 1.0 - P0
  T = SQRT (LOG (1.0 / P ** 2))
  IF (IOUT .EQ. 10) WRITE(8,*), 'P = ', P, ' T = ', T
  T2 = T**2
  T3 = T**3
  X = - (T - ((C0 + C1 * T + C2 * T2)&
  & / (1.0 + D1 * T + D2 * T2 + D3 * T3)))
ENDIF

EPSL = EXP (SIGMA * X)
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'BZERO = ', BZERO, ' SIGMA = ', SIGMA
  WRITE(8,*), 'F = ', F, ' EPSW = ', EPSW
  WRITE(8,*), 'P = ', P, ' T = ', T
  WRITE(8,*), 'T2 = ', T2, ' T3 = ', T3
  WRITE(8,*), 'X = ', X, ' EPSL = ', EPSL
ENDIF

RETURN
END
REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO)

C INPUTS: S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C OUTPUTS: GTLIFE
C
C IMPLICIT NONE
C
INTEGER IOUT, L, MAXREG, NUMREG, ZROREG
PARAMETER (MAXREG = 3)
COMMON IOUT
REAL GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

C
C LIST OF VARIABLES
C
GETLIF VALUE TO BE ASSIGNED TO GTLIFE - CYCLES TO FAILURE FOR THE REQUIRED STRESS LEVEL
IOUT OUTPUT DUMP CONTROLLER
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LNA() 1-D ARRAY CONTAINING VALUES OF ln(A) = M ln K FOR EACH REGION
LNZ NORMAL(0,PVAR) GENERATED RANDOM VARIATE
LPHIM() 1-D ARRAY CONTAINING VALUES OF M ln PHI FOR EACH REGION WHERE PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NREG NUMBER OF REGIONS OF INTEREST
S VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO FAILURE) IS REQUIRED
SBND() 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE “LIFE BOUNDARY” VALUES FOR EACH REGION CONTAINED IN NSND()
SZERO STRESS TENSILE POINT, So
TEMP TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER FLOWS
ZREG Z0REG Region - VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO REGION

GETLIF = 0.0

C CALCULATE CYCLES TO FAILURE

IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
GETLIF = 1.0
ELSE
DO 100 L = ZROREG, NUMREG
IF (S .GT. SBND(L)) THEN
TEMP = LNA(L) + LPHIM(L) + MM(L) * (- ALOG(S) & + ALOG(KRATIO) + LNZ)
IF (TEMP .GT. 86.0) THEN
TEMP = 86.0
ENDIF
GETLIF = EXP (TEMP)
GOTO 150
ENDIF
ENDIF
END!
END!

END

C**************************************************************************
SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
M FOR EACH REGION
PROGRAMMER: L. NEWLIN
DATE: 10FEB88

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE SORTM (ALLM, NUMREG, NUM)

C INPUTS: ALLM, NUMREG, NUM
C OUTPUTS: ALLM

C IMPLICIT NONE
C COMMON IOUT
C INTEGER I, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG
C PARAMETER (MAXMM = 20001, MAXREG = 3)
C LOGICAL INORDR
C REAL ALLM(MAXMM, MAXREG), TEMP

LIST OF VARIABLES

ALLM() 2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
I CONTROLS INSERTION POINTER
INC SORT INCREMENT VARIABLE
INORDR FLAG TO INDICATE WHETHER SORT IS FINISHED
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXMM MAXIMUM NUMBER OF M'S TO BE SORTED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
NUM NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
NUMREG NUMBER OF REGIONS OF INTEREST
TEMP TEMPORARY SORTING VARIABLE

DO 400 L = 1, NUMREG
5 INC = NUM
10 IF (INC .GT. 1) THEN
20 INC = INC / 2
30 INORDR = .TRUE.
DO 300 I = 1, (NUM - INC)
   IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN
      TEMP = ALLM(I,L)
      ALLM(I,L) = ALLM(I + INC, L)
      ALLM(I + INC, L) = TEMP
      INORDR = .FALSE.
   ENDIF
300 CONTINUE
IF (.NOT. INORDR) GOTO 20
GOTO 10
ENDIF
400 CONTINUE

RETURN
END
FUNCTION RAINF3 CALCULATES THE TIME (in missions) TO FAILURE FOR
THE GIVEN STRAIN-TIME HISTORY

PROGRAMMER: L. SHIRAISHI, L. NEWLIN
DATE: 27MAR90
VERSION: 1.1 (BLDLCF V3.1, V3.2, V3.3, V3.4 MATCHR V8.4, V8.5)

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

FUNCTION RAINF3 (SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM,
& KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO)

INPUTS: SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO,
& LNZ, SBND, ZROREG, NUMREG, SZERO

OUTPUTS: RAINF3

IMPLICIT NONE

COMMON IOUT
INTEGER MAXREG, MAXM
PARAMETER (MAXREG = 3, MAXM = 50)
INTEGER I, INDEX(MAXM), IOUT, J, JMAX, K, M, N, NEWTOT, NUMREG,
& ZROREG
REAL CHKFT, E(MAXM), GTLIFE, INVLIF(MAXM), KRATIO,
& LIFE(MAXM), LN(0:MAXREG), LPHIM(0:MAXREG),
& MM(0:MAXREG), PERIOD, RAINF3, S(MAXM), SBND(0:MAXREG),
& SEFF(MAXM), SEFFM(2, MAXM), SEFMAX, SP(MAXM),
& SRANGE(MAXM), SUMDAM, SZERO, TEST1(MAXM), TEST2(MAXM),
& TRUNC, WEXP

LIST OF VARIABLES

RAINF3 TIME TO FAILURE FOR THE GIVEN TIME HISTORY

input variables:
SEFF(M) EFFECTIVE STRAINS BEFORE FILTERING/RAINFLOW
M TOTAL NUMBER OF STRAIN DATA POINTS PER PERIOD
TRUNC VALUE USED TO FILTER OUT NOISE
PERIOD TIME IN SECONDS FOR ONE PERIOD
WEXP WALKER EXPONENT

intermediate variables:
MAXM MAXIMUM NUMBER OF POINTS ALLOWED IN STRAIN-TIME HISTORY
ARRAYS
SEFMAX LARGEST EFFECTIVE STRAIN
JMAX INDEX (LOCATION) OF SEFMAX IN SEFF()
I, J, K COUNTERS FOR VARIOUS DO LOOPS
SP(M+1) RESEQUENCED EFFECTIVE STRAINS; # OF PTS = M+1
INDEX(MAXM), TEST1(MAXM), TEST2(MAXM)
INTERMEDIATE CALCULATION ARRAYS USED DURING FILTERING
S(NEWTOT) FILTERED EFFECTIVE STRAINS
NEWTOT TOTAL NUMBER OF EFFECTIVE STRAIN VALUES AFTER FILTERING
E() HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
N NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
SEFFM(2,N) EFFECTIVE STRAINS AFTER RESEQUENCING/FILTERING/RAINFLOW
& SEFFM(1,I) = sigma max,eff,i
& SEFFM(2,I) = sigma min,eff,i
& SRANGE(N) SRANGE(I) = EQUIVALENT STRAIN RANGE FOR CYCLE I

7 - 250
REAL FUNCTION THAT CALCULATES FATIGUE LIFE FOR A GIVEN STRAIN LEVEL
LIFE(I) = CALCULATED LIFE FOR STRAIN LEVEL SRANGE(I)

SUM OF ALL THE DAMAGE FRACTIONS
INVLIF(I) = 1/LIFE(I)

DUMMY VARIABLE USED TO PRINT OUT RAINF3 RESULT

OUTPUT DUMP CONTROLLER

1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION

NORMAL(0, PVAR) GENERATED RANDOM VARIATE

1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE

MAXIMUM NUMBER OF REGIONS ALLOWED

1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

NAME OF REGIONS OF INTEREST

I-D ARRAY CONTAINING THE STRAIN VALUES (%) FOR EACH REGION

STRAIN TENSILE POINT, So (%)

ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP

BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

dump input data

if (lout.eq.20) then
  write(*,*)'
  write(*,*)'
  write(*,*)'
  write(*,*)'
  write(*,*)'
  write(*,*)'
  endif

DO 50 I = 1, MAXM
  SP(I) = 0.0
  S(I) = 0.0
  E(I) = 0.0
  SEFFM(I,1) = 0.0
  SEFFM(I,2) = 0.0
  SRANGE(I) = 0.0
  LIFE(I) = 0.0
  INDEX(I) = 0
  TEST1(I) = 0.0
  TEST2(I) = 0.0
50 CONTINUE

RESEQUENCE effective strains (needed for rainflow analysis);
largest effective strain is placed at beginning and end of SP()

DO 200 I=1,M
  IF ( SEFF(I) .GT. SEFMAX ) THEN
    SEFMAX = SEFF(I)
    JMAX = I
  ENDIF
200 CONTINUE

assign all points from JMAX out, to the beginning of SP()

DO 210 I = M-JMAX+1, M
  J = JMAX-I + I
  SP(I) = SEFF(J)
210 CONTINUE

assign points before JMAX to the end of SP()

J = 0

7 - 251
DO 220 I = M-JMAX+2, M
  J = J + 1
  SP(I) = SEFF(J)
220 CONTINUE

SP(M+1) = SEFF(JMAX)
if (iout.eq.20) then
  write(*,*) 'seffmax, jmax:' , jmaxwrite(8,*),'seffmax:
  write(8,*)'sp(m+1): ,(sp(i),i=1,m+1)
endif

C******************************************************************************
END RESIDENCE******************************************************************************

C FILTER the resequenced effective strains, leaving only peaks and valleys
C (excursions larger than TRUNC are deleted during rainfall counting) in
C S(NEWTOT), where NEWTOT is the new number of points

DO 300 I = 2, M
  TESTI(T) = SP(I-1) - SP(I)
  TEST2(I) = TESTI(I) * (SP(I) - SP(I+1))
300 CONTINUE
if (iout.eq.20) then
  DO 305 I = 2, m
    write(8,*),'test1 = ', test1(i), 'test2 = ', test2(i)
  305 CONTINUE
endif

K = 1
INDEX(1) = 1
DO 310 I = 2, M
  IF K((TEST(I) .NE. 0) .AND. (TEST2(I) .LT. 0)) THEN
    INDEX(K) = I
  ENDIF
310 CONTINUE

NEWTOT = K + 1
INDEX(NEWTOT) = M + 1
DO 320 I = 1, NEWTOT
  K = INDEX(I)
  S(I) = SP(K)
320 CONTINUE
if (iout .eq. 20) then
  write ( 8, * ) ' newtot : ', newtot
  write ( 8, * ) ' s (newtot) : ', ( s ( i ), i=1, newtot )
endif

C******************************************************************************
END FILTER******************************************************************************

C RAINFLOW ANALYSIS to identify cycles within effective strain data, S(NEWTOT);
C places each cycle's max and min values into SEFFM(2,N)
C counters: I counts # of cycles found, J counts how many S()'s counted,
C K accumulates unmatched points
I = 0
J = 0
K = 0
400 CONTINUE

J = J+1
K = K+1
check J to avoid reading beyond end of filtered strain data
IF ( J .GT. NEWTOT ) GOTO 499
read strain point into a holding array to be checked for cycles
  E(K) = S(J)
410 CONTINUE
IF ( K .LT. 3 ) GOTO 400
IF ( ABS( E(K) - E(K-1) ) .LT. ABS( E(K-1) - E(K-2) ) ) GOTO 400
C if not, then a cycle has been found, but we need to check for truncation
IF (ABS (E(K-1) - E(K-2)) .GT. TRUNC) THEN

7 - 252
C cycle is large enough to save
I = I+1
SEFFM(1,I) = AMAX1(E(K-1), E(K-2))
SEFFM(2,I) = AMIN1(E(K-1), E(K-2))
ENDIF
C discard points K-1 and K-2, and decrement the counter of unmatched points
E(K-2) = E(K)
K = K-2
C return for more counting
GOTO 410

499 CONTINUE
C N equals the final number of cycles found
N = I
if (iout.eq.20) then
    write(8,*')'N :',n
    write(8,*')'seffm(2,n) :
    do 12 i=1,n
        write(8,*') seffm(1,i), seffm(2,i)
    12 continue
endif
IF (N .EQ. 0) THEN
C truncation filter value too large -- no cycles left
SUMDAM = 1.0E-36
GOTO 710
ENDIF
C
*************** END RAINFLOW **************
C calculate equivalent strain range
C
DO 500 I=I,N
SRANGE(I) = (SEFFM(I,I) - SEFFM(2,I))
& * ((SEFFM(I,I) - SEFFM(2,I)) / (2.0 * SEFFM(I,I)))
& ** (WEXP - 1.0)
500 CONTINUE
if (iout.eq.20) write(8,*')'srang(n) :',(srange(i),i=1,n)
if (iout.eq.25) write(8,*') (srang(i),i=1,n),
& exp(lphim(1)/mm(1))
C calculate lives and damage fractions: LIFE(N) and INVLIF(N)
C
DO 600 I=1,N
LIFE(I) = GTLIFE (SRANGE(I), MM, LNA, LPHIM, KRATIO, LNZ,
& SBND, ZROREG, NUMREG, SZERO)
600 CONTINUE
DO 650 I=I,N
INVLIF(I) = 1.0 / LIFE(I)
650 CONTINUE
if (iout.eq.20) then
    do 14 i=1,n
        write(6,*')'life(n):',life(i),
& invlif(n):',invlif(i)
    14 continue
endif
C Miner's Rule -- sum the damage fractions
C
SUMDAM = 0.0
DO 700 I=1,N
SUMDAM = SUMDAM + INVLIF(I)
700 CONTINUE
710 CONTINUE
if (iout.eq.20) write(8,*')'sumdam:',sumdam
C calculate fatigue life (time to failure)
C
RAINF3 = PERIOD / SUMDAM
if (iout.eq.15) then
    chkft=period/sumdam

7 - 253
write(8,*)' rainf3 life',chkft
write(8,*)
endif

RETURN
END
Section 7.3
High Cycle Fatigue Failure Program BLDHCF

The program tree structures, list of subprograms, descriptions of the key variables, and the FORTRAN source listing for the high cycle fatigue analysis code BLDHCF are given here. The pertinent HCF methodology is given in Section 4. The overall description of the program and the flowcharts are given in Section 5.3.

7.3.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for BLDHCF, using Uniform variation on the materials shape parameter $m$, is given in Figure 7.3-1, while the tree structure for the truncated Normal case is given in Figure 7.3-2. In both trees, those subprograms not “shadow-boxed” are part of the materials characterization model. The program, subprogram, and file names are indicated by UPPERCASE letters.

7.3.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7.3-1. The section numbers where the subprograms are described by means of flowcharts are given next to the names.
Figure 7.3-1  Tree Structure for Program BLDHCF for the Uniform Variation in Materials Shape Parameter $m$
Figure 7.3-2  Tree Structure for Program BLDHCF for the Truncated Normal Variation in Materials Shape Parameter $m$
Table 7.3-1  List of Subprograms For Program BLDHCF
(Footnotes are at the end of the table)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDREG</td>
<td>4.1.3.9</td>
<td>Adds the $m$ ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>4.1.3.15</td>
<td>Adds the $m$ ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.</td>
</tr>
<tr>
<td>BLDHCF</td>
<td>5.3.2.1</td>
<td>The main routine that controls the logical flow of the high cycle fatigue turbine blade program.</td>
</tr>
<tr>
<td>BLDHLF</td>
<td>5.3.2.4</td>
<td>Performs the calculations of the driver transformation and the fatigue life.</td>
</tr>
<tr>
<td>CONCAV</td>
<td>4.1.3.10</td>
<td>Adjusts the upper bound of the posterior ranges on $m$ to be consistent with concavity constraints.</td>
</tr>
<tr>
<td>CONVRT</td>
<td>4.1.3.3</td>
<td>Transforms stress data to equivalent zero-mean stresses with stress ratio of $-1.0$.</td>
</tr>
<tr>
<td>DRVRIN</td>
<td>5.3.2.2</td>
<td>Reads the driver distributions and other structural and geometric parameters from BLDHCD and echoes the data to BLDHCO.</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>4.1.3.12</td>
<td>Calculates the median S/N curve parameters from the results of the information aggregation calculations.</td>
</tr>
<tr>
<td>FINDK</td>
<td>4.1.5.6</td>
<td>Calculates the value of the location parameter $K$ (where $A = K^m$) for each life region by using Equations 2-37 and 2-41 of [1].</td>
</tr>
<tr>
<td>FINDM</td>
<td>4.1.5.1</td>
<td>Obtains the value of $m$ for each life region by adjusting the range (to ensure concavity) and then sampling from the Uniform distribution over the appropriate $m$ range.</td>
</tr>
<tr>
<td>FINDMC</td>
<td>4.1.3.5</td>
<td>Calculates the $m$ range implied by the constraint on the coefficient of variation of fatigue strength, $C$, for each life region, by using Equations 2-28 through 2-32 of [1].</td>
</tr>
<tr>
<td>FINDMN</td>
<td>4.1.5.2</td>
<td>Obtains the value of $m$ for each life region by sampling from the appropriate truncated Normal distribution on $m$.</td>
</tr>
<tr>
<td>FINDSB</td>
<td>4.1.5.7</td>
<td>Calculates the life region &quot;tie-points&quot; or stress values which correspond to the &quot;life boundaries,&quot; conditional on the randomly selected $m$ for each region. Also calculates $K$, characterizing the specific material S/N data set, which is a function of $\beta_o$ and $k$.</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>4.1.3.8</td>
<td>Combines the 95% confidence interval, $J_o$, with the implicit and explicit constraints on $m$, to obtain posterior credibility ranges on $m$ for each life region.</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>4.1.8</td>
<td>Calculates the cycles to failure for a particular stress, based upon the materials characterization model S/N curve of Equation 2-48 of [1].</td>
</tr>
<tr>
<td>NAME</td>
<td>SECTION</td>
<td>PURPOSE</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>4.1.3.7*</td>
<td>Calculates $\sigma^2$, the extent of departures from the multiple heat median S/N curve warranted by the information available, by using Equation 2-49 of [1].</td>
</tr>
<tr>
<td>INFAGG</td>
<td>4.1.3*</td>
<td>Controls the logical flow for the information aggregation portion of the materials characterization model.</td>
</tr>
<tr>
<td>INIT</td>
<td>4.1.3.1*</td>
<td>Initializes the entries of the arrays used in the information aggregation subroutine, INFAGG, to zero.</td>
</tr>
<tr>
<td>INSORT</td>
<td>5.B*</td>
<td>Performs an insertion sort for the lowest fifty percent of the lives calculated.</td>
</tr>
<tr>
<td>INTRVL</td>
<td>4.1.3.6*</td>
<td>Calculates the 95% confidence intervals $J_o$ for $C$, and $I_o$ for $m$, for each region by using Equations 2-24 through 2-26 of [1].</td>
</tr>
<tr>
<td>KBETA</td>
<td>4.1.5.5*</td>
<td>Calculates $k$ and $\beta_o$ from the sample mean and variance of $Z$, where $Z$ is a function of stress, life, the life region boundaries, and the $m$'s by using Equation 2-42 of [1].</td>
</tr>
<tr>
<td>KOMO</td>
<td>4.1.6*</td>
<td>Calculates $K_o$ and $m_o$ for the zero region, the no data region to the left of the first data region. Extends the S/N curve consistent with the tensile point at $S_o$.</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>4.1.3.11*</td>
<td>Calculates the median values of $m$, based on the posterior credibility ranges of $m$, by using Equation 2-34 of [1].</td>
</tr>
<tr>
<td>MUSIG</td>
<td>4.1.3.13*</td>
<td>Calculates the posterior Normal distribution parameters, mean $m_o$, and standard deviation $\sigma_o$, for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NORMGN</td>
<td>4.4.3*</td>
<td>Generates $\text{Normal}(\mu, \sigma^2)$ random variates.</td>
</tr>
<tr>
<td>NORRNG</td>
<td>4.1.3.14*</td>
<td>Combines the implicit and explicit constraints on $m$ to obtain the posterior credibility ranges of $m$ for each life region.</td>
</tr>
<tr>
<td>PAREST</td>
<td>4.1.5*</td>
<td>Controls the logical flow for the parameter estimation model portion of the materials characterization model.</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7.6.6*</td>
<td>Generates the $\text{Uniform}(a, b)$ and $\text{Uniform}(c, d)$ pair of independent random variates.</td>
</tr>
<tr>
<td>RANDOM</td>
<td>4.4.2*</td>
<td>Uses a Linear Congruential random number Generator (LCG) to generate $\text{Uniform}(0, 1)$ random variates.</td>
</tr>
<tr>
<td>RCE</td>
<td>4.1.3.2*</td>
<td>Reads the data from BLDHCD and RELATD; calls CONVRT to transform the stress data to a stress ratio of $-1.0$; and echoes the data to BLDHCO and RELATO. RCE also breaks S/N data sets into regions as specified by the user.</td>
</tr>
<tr>
<td>SELECT</td>
<td>5.3.2.3</td>
<td>Performs the driver selection.</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>4.1.5.4*</td>
<td>Calculates the sample mean and variance of $Z$, where $Z$ is a function of stress, life, the life region boundaries, and the $m$'s, by using Equation 2-42 of [1].</td>
</tr>
</tbody>
</table>
Table 7.3-1 List of Subprograms For Program BLDHCF (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTM¹³</td>
<td>4.1.10*</td>
<td>Sorts the $m$ values in increasing order for each life region for the truncated Normal distribution case.</td>
</tr>
<tr>
<td>SW2SU²</td>
<td>4.1.3*</td>
<td>Calculates the residual variances from the $Y$ on $X$ and $X$ on $Y$ regressions for each life region where $Y = \ln(\text{Endurance cycles})$ and $X = \ln(\text{Stress})$ by using Equations 2-20 and 2-21 of [1]; to be used in the credibility range calculations.</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>4.1.11*</td>
<td>Performs premature program termination when required.</td>
</tr>
<tr>
<td>TRNSFM¹⁴</td>
<td>4.1.5.3*</td>
<td>Performs the calculations necessary to transform the specific material S/N data into the variable $Z$, where $Z$ is a function of stress, life, the life region boundaries, and the $m$'s.</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>4.4.6*</td>
<td>Generates Weibull($\beta, \eta(\beta)$) random variates.</td>
</tr>
</tbody>
</table>

* See [1].

1. No data regions to the right are discussed in [1], Page 2-17.
2. Concavity constraints are discussed in [1], Pages 2-13 through 2-14.
3. The stress transformation is discussed in [1], Page 2-7.
4. The median S/N curve parameter estimation calculations are described in [1], Pages 2-15 through 2-18.
5. Selection of the $\{m_j\}$ parameters is discussed in [1], Page 2-15.
6. Combining information to obtain the posterior credibility ranges on $m$ is discussed in [1], Page 2-13.
7. The information aggregation calculations are discussed in [1], Pages 2-6 through 2-14.
8. Extension of the S/N curve to the left is discussed in [1], Page 2-17.
9. Calculation of the truncated Normal distribution parameters is discussed in [1], Page 2-14.
10. The Normal distribution is discussed in [1], Page 2-23.
11. The parameter estimation calculations are discussed in [1], Pages 2-15 through 2-18.
12. The Uniform distribution is discussed in [1], Page 2-23.
13. The need for saving $m$'s is discussed in [1], Page 2-15.
14. The S/N data transformation is discussed in [1], Page 2-16.
7.3.3 Description of Variables

A list of variables used in the ATD-HPOTP first and third stage turbine blade HCF code, BLDHCF, is given in Table 7.3-2. The variable names are indicated by **BOLD** **UPPERCASE** letters; the variable “type” can be interpreted as follows: INT is a standard integer variable; RE is a standard real variable; and DRE is a double precision variable. The various array dimensions are defined by using the following parameters: **MAXBLF**, **MAXDAT**, **MAXLIF**, **MAXMM**, and **MAXREG**.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0, A1</td>
<td>RE</td>
<td>The coefficients for the flow rate $\dot{m}$ response surface function (performance balance characterization).</td>
</tr>
<tr>
<td>ALLM(MAXMM, MAXREG)</td>
<td>RE</td>
<td>2-D array containing the materials model shape parameters ($m$'s) for each life region which are to be used in the truncated Normal median S/N curve calculation.</td>
</tr>
<tr>
<td>B0, B1</td>
<td>RE</td>
<td>The coefficients for the enthalpy change $\Delta h$ response surface function (performance balance characterization).</td>
</tr>
<tr>
<td>BIGK(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the materials model location parameter $K$, where $A = K^m$, given in Equation 2-12 of [1].</td>
</tr>
<tr>
<td>BIGK1</td>
<td>RE</td>
<td>Dummy variable used during calls to subroutine EXPCTD, equal to BIGK(1).</td>
</tr>
<tr>
<td>BLDHLF</td>
<td>RE</td>
<td>Real function that performs the calculations of the driver transformation and fatigue life, and returns the fatigue life (sec).</td>
</tr>
<tr>
<td>BLFPER(MAXBLF)</td>
<td>RE</td>
<td>1-D array containing user specified B-lives which are obtained from the simulated failure distribution. A B-life is the value of accumulated operating time to failure at a failure probability specified as a percent: e.g., B.1 is the failure time at a probability of 0.001 or 0.1%.</td>
</tr>
<tr>
<td>BLFPOS(MAXBLF)</td>
<td>INT</td>
<td>1-D array containing the indices for the array variable LIFE( ) corresponding to the user-requested simulated failure distribution B-lives contained in variable BLFPER( ).</td>
</tr>
</tbody>
</table>
### Table 7.3-2 List of Variables For Program BLDHCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BZERO</td>
<td>RE</td>
<td>Estimate of Weibull distribution shape parameter, $\beta_\nu$, that characterizes the intrinsic variation of the S/N data set, by using Equation 2-11 of [1].</td>
</tr>
<tr>
<td>C</td>
<td>RE</td>
<td>$C$ (in.) in Equation 4-1, the randomly selected distance from the turbine blade neutral axis.</td>
</tr>
<tr>
<td>C0, C10, C11, C20, C21</td>
<td>RE</td>
<td>The coefficients for the damper effectiveness response surface.</td>
</tr>
<tr>
<td>CM</td>
<td>RE</td>
<td>Mean, $\mu$, of Normally distributed $C$, the distance from the turbine blade neutral axis (in.), given in Equation 4-1.</td>
</tr>
<tr>
<td>CS</td>
<td>RE</td>
<td>Standard deviation, $\sigma$, of Normally distributed $C$, the distance from the turbine blade neutral axis (in.), given in Equation 4-1.</td>
</tr>
<tr>
<td>DELTAH</td>
<td>RE</td>
<td>$\Delta h$ (Btu/lbm) in Equation 4-1, the enthalpy change across the turbine stage.</td>
</tr>
<tr>
<td>DUM</td>
<td>RE</td>
<td>Dummy variable.</td>
</tr>
<tr>
<td>FACTR</td>
<td>RE</td>
<td>Equal to FACTOR = PHI * KRATIO * Z. Used by the materials model.</td>
</tr>
<tr>
<td>IFTY</td>
<td>RE</td>
<td>Variable used to access the fifty-percent point in the LIFE() array.</td>
</tr>
<tr>
<td>FTU</td>
<td>RE</td>
<td>Material ultimate strength (psi).</td>
</tr>
<tr>
<td>FTY</td>
<td>RE</td>
<td>Material yield strength (psi).</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>RE</td>
<td>Function given by Equation 2-48 of [1] that calculates the fatigue cycles to failure at a given stress.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls inner DO loop.</td>
</tr>
<tr>
<td>IMIN</td>
<td>RE</td>
<td>$I_{min}$ (in.$^4$) in Equation 4-1, the minimum moment of inertia of the turbine blade cross section.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>J</td>
<td>INT</td>
<td>Controls DO loop for each B-life.</td>
</tr>
<tr>
<td>K</td>
<td>INT</td>
<td>Controls outer DO loop.</td>
</tr>
<tr>
<td>KRATIO</td>
<td>RE</td>
<td>Ratio of $MED K^*/MED K$ in Equation 2-48 of [1]. KRATIO is constant over life regions for the materials model.</td>
</tr>
<tr>
<td>L</td>
<td>INT</td>
<td>Controls DO loop for each life region of the S/N curve.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>LAMB</td>
<td>RE</td>
<td>$\lambda_B$, the randomly selected turbopump performance balance model accuracy factor.</td>
</tr>
<tr>
<td>LAMBA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda_B$.</td>
</tr>
<tr>
<td>LAMBB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\lambda_B$.</td>
</tr>
<tr>
<td>LAMD</td>
<td>RE</td>
<td>$\lambda_D$, the randomly selected damper coefficient of friction model accuracy factor.</td>
</tr>
<tr>
<td>LAMDA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda_D$.</td>
</tr>
<tr>
<td>LAMDB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\lambda_D$.</td>
</tr>
<tr>
<td>LIFE</td>
<td>RE</td>
<td>$L$, the fatigue life in seconds.</td>
</tr>
<tr>
<td>LIFE(MAXLIF)</td>
<td>RE</td>
<td>1-D array containing values of the lives generated by program BLDHCF. The lives are sorted values for the left-hand tail simulated failure distribution.</td>
</tr>
<tr>
<td>LNA(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of $\ln(A) = \ln(BIGK) \times MM$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>LNZ</td>
<td>RE</td>
<td>$\ln(Z)$ in Equation 2-48 of [1], the Normal($0, \text{PVAR}$) random variate for the materials process variation aspect of the materials model.</td>
</tr>
<tr>
<td>LPHIM(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of $\ln(\text{PHI}) \times MM$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>M</td>
<td>INT</td>
<td>Controls symmetry DO loop.</td>
</tr>
<tr>
<td>MAXBLF</td>
<td>INT</td>
<td>Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is $10^2$.</td>
</tr>
<tr>
<td>MAXDAT</td>
<td>INT</td>
<td>Maximum number of points per data set per region allowed for the S/N curve. The maximum number of data points per set allowed is 50.</td>
</tr>
<tr>
<td>MAXLIF</td>
<td>INT</td>
<td>Maximum number of fatigue lives allowed for the simulated failure distribution. The maximum number of fatigue lives to be saved is 10,000.</td>
</tr>
<tr>
<td>MAXMM</td>
<td>INT</td>
<td>Maximum number of $m'$s to be saved and sorted for the truncated Normal median S/N curve. The maximum number of $m'$s is 20,000.</td>
</tr>
<tr>
<td>MAXREG</td>
<td>INT</td>
<td>Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.</td>
</tr>
<tr>
<td>MCOUNT</td>
<td>INT</td>
<td>Counts number of $m'$s to be used to calculate the median S/N curve for the truncated Normal distribution case.</td>
</tr>
</tbody>
</table>
## Table 7.3-2 List of Variables For Program BLDHCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD</td>
<td>RE</td>
<td>$m_d$, the damper mass (lbm).</td>
</tr>
<tr>
<td>MDOT</td>
<td>RE</td>
<td>$\dot{m}$ (lbm/sec) in Equation 4-1, the fluid mass flow rate.</td>
</tr>
<tr>
<td>MEDM(MAXMM)</td>
<td>RE</td>
<td>1-D array containing the empirical median $m$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>MID</td>
<td>INT</td>
<td>Pointer to the median $m$ values in array SORTM( ) for the truncated Normal median S/N curve. Value of half of MCOUNT.</td>
</tr>
<tr>
<td>MINPHI</td>
<td>RE</td>
<td>Value of $\min(\phi)$, the minimum of $\text{NSYM}$ draws of the materials scatter parameter $\phi$.</td>
</tr>
<tr>
<td>MM(0:MAXREG)</td>
<td>RE</td>
<td>$m_j$ in Equation 2-12 of [1], the 1-D array containing randomly selected values of the materials model shape parameter $m$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>MPROC</td>
<td>INT</td>
<td>Materials PROCess variation. Controls materials process variation. A value of 0 indicates no materials process variation, while a value of 1 indicates that materials process variation should be included.</td>
</tr>
<tr>
<td>MRW2</td>
<td>RE</td>
<td>$m_d \cdot r_d \cdot \omega^2$, the damper normal load (lbf).</td>
</tr>
<tr>
<td>MU(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution mean of the materials shape parameter $m$ for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>MW</td>
<td>RE</td>
<td>$m_w$ in Equation 4-7, the randomly selected characteristic exponent for the Walker relation.</td>
</tr>
<tr>
<td>MWA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $m_w$.</td>
</tr>
<tr>
<td>MWB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $m_w$.</td>
</tr>
<tr>
<td>NB</td>
<td>INT</td>
<td>$N_b$ in Equation 4-1, the number of rotor blades.</td>
</tr>
<tr>
<td>NBLIFE</td>
<td>INT</td>
<td>Number of B-lives to be obtained from the simulated failure distribution.</td>
</tr>
<tr>
<td>NBND(0:MAXREG)</td>
<td>RE</td>
<td>$N_{f_i+1}$ in Equation 2-35 of [1], the 1-D array containing upper bounds for the NUMREG life regions of interest for the specific material S/N data set.</td>
</tr>
<tr>
<td>NEWLIF</td>
<td>RE</td>
<td>Fatigue life value (missions) returned from call to function BLDHLF.</td>
</tr>
<tr>
<td>NF</td>
<td>RE</td>
<td>$N_f$, the fatigue life in cycles.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NF(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values from the array RAWNF( ) for the specific material S/N data set partitioned into life regions.</td>
</tr>
<tr>
<td>NHYPER</td>
<td>INT</td>
<td>The outer loop size.</td>
</tr>
<tr>
<td>NLIFE</td>
<td>INT</td>
<td>The inner loop size.</td>
</tr>
<tr>
<td>NLIFET</td>
<td>INT</td>
<td>Total number of lives calculated by program BLDHCF. Value of NHYPER * NLIFE.</td>
</tr>
<tr>
<td>NMED</td>
<td>INT</td>
<td>Controls S/N curve median calculation for the truncated Normal distribution case. A value of 0 indicates that the user does not desire a median calculation or that the Uniform distribution case is being used; while a value of 1 indicates that the user desires the median calculation to be performed.</td>
</tr>
<tr>
<td>NPTS(MAXREG)</td>
<td>INT</td>
<td>1-D array containing the number of points per life region for the specific material S/N data set.</td>
</tr>
<tr>
<td>NS</td>
<td>INT</td>
<td>$N_s$, the number of stator blades.</td>
</tr>
<tr>
<td>NSYM</td>
<td>INT</td>
<td>Symmetry number, usually equal to the multiplicity of the modeling unit in the component.</td>
</tr>
<tr>
<td>NUMREG</td>
<td>INT</td>
<td>$R$ in Equation 2-11 of [1], the number of life regions of interest in the S/N curve.</td>
</tr>
<tr>
<td>PHI</td>
<td>RE</td>
<td>$\phi$ in Equation 2-11 of [1], the material's intrinsic variation, or scatter, given by a Weibull($\beta_0, \eta_0(\beta_0)$) random variate.</td>
</tr>
<tr>
<td>PSIG</td>
<td>RE</td>
<td>$\sigma$ in Equation 2-48 of [1], the value of SQRT(PVAR).</td>
</tr>
<tr>
<td>PVAR</td>
<td>RE</td>
<td>$\sigma^2$ in Equation 2-48 of [1], characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.</td>
</tr>
<tr>
<td>R</td>
<td>RE</td>
<td>$R$ in Equation 4-6, the stress ratio.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>RANGEM(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the posterior credibility ranges on the materials model shape parameter $m$ for each life region in the S/N curve. RANGEM(1,L) is the lower bound and RANGEM(2,L) is the upper bound in region L.</td>
</tr>
<tr>
<td>RAVG</td>
<td>RE</td>
<td>$r_{avg}$ (in.) in Equation 4-1, the randomly selected average turbine blade radius relative to the shaft center.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>RAVGM</td>
<td>RE</td>
<td>Mean, ( \mu ), of Normally distributed ( r_{avg} ), the average turbine blade radius relative to the shaft center (in.).</td>
</tr>
<tr>
<td>RAVGS</td>
<td>RE</td>
<td>Standard deviation, ( \sigma ), of Normally distributed ( r_{avg} ), the average turbine blade radius relative to the shaft center (in.).</td>
</tr>
<tr>
<td>RD</td>
<td>RE</td>
<td>( r_d ), the randomly selected damper radius (in.).</td>
</tr>
<tr>
<td>RDM</td>
<td>RE</td>
<td>Mean, ( \mu ), of Normally distributed ( r_d ), the damper radius (in.).</td>
</tr>
<tr>
<td>RDS</td>
<td>RE</td>
<td>Standard deviation, ( \sigma ), of Normally distributed ( r_d ), the damper radius (in.).</td>
</tr>
<tr>
<td>RPM</td>
<td>RE</td>
<td>( \omega ) (rpm) in Equation 4-1, the randomly selected steady state rotor speed.</td>
</tr>
<tr>
<td>RPMM</td>
<td>RE</td>
<td>Mean, ( \mu ), of Normally distributed ( \omega ), the steady state rotor speed (rpm).</td>
</tr>
<tr>
<td>RPMS</td>
<td>RE</td>
<td>Standard deviation, ( \sigma ), of Normally distributed ( \omega ), the steady state rotor speed (rpm).</td>
</tr>
<tr>
<td>RROOT</td>
<td>RE</td>
<td>( r_{root} ) (in.) in Equation 4-1, the randomly selected turbine blade root radius relative to the shaft center.</td>
</tr>
<tr>
<td>RROOTM</td>
<td>RE</td>
<td>Mean, ( \mu ), of Normally distributed ( r_{root} ), the turbine blade root radius relative to the shaft center (in.).</td>
</tr>
<tr>
<td>RROOTS</td>
<td>RE</td>
<td>Standard deviation, ( \sigma ), of Normally distributed ( r_{root} ), the turbine blade root radius relative to the shaft center (in.).</td>
</tr>
<tr>
<td>SALT</td>
<td>RE</td>
<td>( \sigma_{ALT} ) (psi) in Equation 4-3, the alternating stress.</td>
</tr>
<tr>
<td>SBND(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the stress values (psi) with stress ratio = -1.0, corresponding to the &quot;life boundary&quot; values for each life region of the S/N curve contained in array NBND().</td>
</tr>
<tr>
<td>SBRM</td>
<td>RE</td>
<td>( \sigma_{BR} ) (psi) in Equation 4-1, the blade root mean stress.</td>
</tr>
<tr>
<td>SDSUD</td>
<td>RE</td>
<td>( \sigma_D / \sigma_{UD} ) (psi) in Equation 4-3, the ratio of the damped blade vibratory stress to the undamped blade vibratory stress.</td>
</tr>
<tr>
<td>SEQ</td>
<td>RE</td>
<td>( \sigma_{EQ} ) (psi) in Equation 4-7, the equivalent zero-mean stress amplitude.</td>
</tr>
</tbody>
</table>
Table 7.3-2  List of Variables For Program BLDHCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIG(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution standard deviation of the materials model shape parameter ( m ) for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>SMAX</td>
<td>RE</td>
<td>( \sigma_{\text{MAX}} ) (psi) in Equation 4-4, the maximum or peak stress.</td>
</tr>
<tr>
<td>SMEAN</td>
<td>RE</td>
<td>( \sigma_{\text{MEAN}} ) (psi) in Equation 4-2, the mean stress.</td>
</tr>
<tr>
<td>SMIN</td>
<td>RE</td>
<td>( \sigma_{\text{MIN}} ) (psi) in Equation 4-5, the minimum or trough stress.</td>
</tr>
<tr>
<td>STR(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing stress points with stress ratio ( = -1.0 ), for the specific material S/N data set partitioned into life regions.</td>
</tr>
<tr>
<td>SUD</td>
<td>RE</td>
<td>( \sigma_{\text{UD}} ) (psi) in Equation 4-3, the undamped blade vibratory stress.</td>
</tr>
<tr>
<td>SZERO</td>
<td>RE</td>
<td>Stress tensile test point, ( S_o ) (psi).</td>
</tr>
<tr>
<td>TRBIGK(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the materials model location parameter ( K ) consistent with the tensile point ( S_o ).</td>
</tr>
<tr>
<td>TRSBND(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the stress values (psi) with stress ratio ( = -1.0 ), corresponding to the “life boundary” values for each region of the S/N curve contained in array NBND( ) for each PHI draw consistent with the tensile point ( S_o ).</td>
</tr>
<tr>
<td>VARY</td>
<td>INT</td>
<td>Controls type of S/N curve variation desired. A value of 0 indicates that no variation is required; a value of 1 means that intrinsic materials variation only is desired; a value of 2 indicates that the user desires a Uniform distribution on ( m ); while a value of 3 indicates that a truncated Normal distribution is desired.</td>
</tr>
<tr>
<td>Z</td>
<td>RE</td>
<td>( Z ) in Equation 2-48 of [1], the randomly selected process variation shift factor given by a Lognormal(0, PVAR) random variate.</td>
</tr>
<tr>
<td>ZROREG</td>
<td>INT</td>
<td>Zero Region, the variable permits the inclusion of the tensile point ( S_o ). The value of 0 implies a DO loop from zero to NUMREG, while a value of 1 causes the DO loop to be executed from one to NUMREG.</td>
</tr>
</tbody>
</table>
The need for saving $m$'s is discussed in [1], Page 2-15.

See variable BLFPER() for a description of B-life.

The median S/N curve for the truncated Normal case is discussed in [1], Page 2-15.

See [1], Section 2.1.2.3, for a discussion on process variation in materials.

$m_0$ of the posterior density of $m$ is discussed in [1], Page 2-14.

The posterior credibility ranges $\pi(m)$ are discussed in [1], Page 2-13.

$\sigma_0$ of the posterior density of $m$ is discussed in [1], Page 2-14.

Extension of the S/N curve to the left using the tensile point is discussed in [1], Page 2-17.
7.3.4 Program BLDHCF Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program BLDHCF Listing Temporal Order, Uniform Distribution</td>
<td>7-270</td>
</tr>
<tr>
<td>Program BLDHCF Listing Temporal Order, Truncated Normal Distribution</td>
<td>7-271</td>
</tr>
<tr>
<td>BLDHCF</td>
<td>7-272</td>
</tr>
<tr>
<td>DRVRAIN</td>
<td>7-276</td>
</tr>
<tr>
<td>SELECT</td>
<td>7-280</td>
</tr>
<tr>
<td>BLDHLF</td>
<td>7-281</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-283</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-284</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-285</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-290</td>
</tr>
<tr>
<td>INIT</td>
<td>7-290</td>
</tr>
<tr>
<td>RCE</td>
<td>7-292</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-298</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-300</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-302</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-305</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-307</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-308</td>
</tr>
<tr>
<td>ADDREG</td>
<td>7-312</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-313</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-314</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-316</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-318</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-319</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-322</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-323</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-325</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-328</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-329</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-330</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-331</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-332</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-333</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-334</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-336</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-336</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-337</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-339</td>
</tr>
</tbody>
</table>

BLDHCF Version 1.1
# Program BLDHCF Listing Temporal Order, Uniform Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDHCF</td>
<td>7-272</td>
</tr>
<tr>
<td>DRVFIN</td>
<td>7-278</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-285</td>
</tr>
<tr>
<td>INIT</td>
<td>7-290</td>
</tr>
<tr>
<td>RCE</td>
<td>7-292</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-298</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-300</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-305</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-301</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-307</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-308</td>
</tr>
<tr>
<td>ADDRGEN</td>
<td>7-312</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-313</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-314</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-316</td>
</tr>
<tr>
<td>TRNSFHM</td>
<td>7-330</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-331</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-332</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-333</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-334</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-336</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-323</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-325</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>TRNSFHM</td>
<td>7-330</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-331</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-332</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-333</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-334</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-329</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>SELECT</td>
<td>7-280</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-329</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-284</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-336</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-336</td>
</tr>
<tr>
<td>BLDHALF</td>
<td>7-281</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-337</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-283</td>
</tr>
</tbody>
</table>
## Program BLDHCF Listing Temporal Order, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDHCF</td>
<td>7-272</td>
</tr>
<tr>
<td>DRVRRIN</td>
<td>7-278</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-285</td>
</tr>
<tr>
<td>INIT</td>
<td>7-290</td>
</tr>
<tr>
<td>RCE</td>
<td>7-291</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-298</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-300</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-305</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-318</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-307</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-319</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-322</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-313</td>
</tr>
<tr>
<td>PAAREST</td>
<td>7-323</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-328</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-329</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-330</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-331</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-332</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-333</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-334</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-329</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>SELECT</td>
<td>7-260</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-329</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-284</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-336</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-336</td>
</tr>
<tr>
<td>BLDHLF</td>
<td>7-281</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-337</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-283</td>
</tr>
<tr>
<td>SORTTM</td>
<td>7-339</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-316</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-330</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-331</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-332</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-333</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-334</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-336</td>
</tr>
</tbody>
</table>
C PROGRAM BLDHCF CONTROLS THE FLOW OF LOGIC OF THE HIGH CYCLE
C FATIGUE ANALYSIS OF THE TURBINE BLADE FOIL PROBLEM
C PROGRAMMER: L. NEWLIN
C DATE: 20APR92
C VERSION: 1.1 (MATCHR V8.5, INSORT V2.1)
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

C*****************************************************
C PROGRAM BLDHCF
C
C SUBPROGRAMS: DRVRIN, INFAGG, PAREST, NORMGN, SELECT, WEIBGN,
C TRMNAT, BLDHLF, INSORT, SORTM, EXPTCD
C 7:DUMP-NEW; 8: IOUTPR-NEW; 9:LOWLIF-NEW;
C NOTE: 5 & 6 ARE OPENED IN 'INFAGG'
C
C IMPLICIT NONE
C
INTEGER MAXBLF, MAXDAT, MAHIF, MAXMM, MAXREG
PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLIF = 10000,
& MAXMM = 20001, MAXREG = 3)
COMMON IOUT
INTEGER BLFPOS(MAXBLF), I, IOUT, J, K, L, M, MCOUNT, MID,
& MPROC, NS, NALIFE, NHYP, NLI, NLFET, NMED,
& NPTS(MAXREG), NS, NSYM, NUMREG, VARY, ZROREG

DOUBLE PRECISION RAND
REAL A0, AI, ALLM(MAXMM, MAXREG), B0, B1, BIGK(0:MAXREG),
& BIGK1, BLDHLF, BLFPER(MAXBLF), BZERO, C, C0, C10, C11,
& C20, C21, CH, CS, FACTR, FIFTY, FTU, FTU, IHIM, KRATIO,
& LAMB, LAMBA, LAMBDA, LAMBDA, LAMDA, LAMDB, LIFE(MAXLIF),
& LNA(0:MAXREG), LR, LPHIM(0:MAXREG), MD, MEDM(MAXREG),
& MINPHI, MM(0:MAXREG), MU(MAXREG), MWA, MNB,
& NBND(0:MAXREG), NEWLIF, NF(MAXDAT, MAXREG), PHI, PSIG,
& PVAR, RANGEM(2, MAXREG), RAVG, RAVG, RAVG, RD, RD,
& RDS, RPM, RPHM, RROOT, RROOT, RROOTS,
& SBND(0:MAXREG), SIG(MAXREG), STR(MAXDAT, MAXREG), SZERO,
& TRBIGK(0:MAXREG), TRSBND(0:MAXREG), Z
C ** SEE BOTTOM OF PROGRAM FOR LIST
C OF VARIABLES
C
OPEN (1, FILE = 'BLDHCD', STATUS = 'OLD')
OPEN (3, FILE = 'BLDHCO', STATUS = 'NEW')
OPEN (7, FILE = 'DUMP', STATUS = 'NEW')
OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')

READ(1, *) RAND
WRITE(8, *) ' RANDOM NUMBER SEED = ', RAND
READ(1, *) IOUT
WRITE(8, *) ' IOUT (MATCHR = 10, BLDHCF = 15) = ', IOUT
READ(1, *) NLI
WRITE(8, *) ' INNER LOOP SIZE = ', NLI
READ(1, *) NHYP
WRITE(8, *) ' OUTER LOOP SIZE = ', NHYP
READ(1, *) NSYM
WRITE(8, *) ' SYMMETRY NUMBER = ', NSYM
READ(1, *) VARY
WRITE(8, *) ' TYPE OF S/N VARIATION DESIRED
WRITE(8, *) ' (0 - NONE; 1 - INTRINSIC; 2 - UNIFORM; 3 - NORMAL) = ', VARY
READ(1, *) NMED
WRITE(8, *) ' NORMAL MEDIAN CURVE (0 - NO, 1 - YES) = ', NMED
READ(1, *) MPROC
WRITE(8, *) ' MATERIALS PROCESS VARIATION DESIRED
WRITE(8, *) ' (0 - NO, 1 - YES) = ', MPROC
IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
  WRITE (8, *) 'ERROR: INVALID TYPE OF S/N VARIATION DESIRED'
  CALL TRMNAT
ENDIF

IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
  WRITE (8, *) 'ERROR: INVALID RESPONSE TO NORMAL MEDIAN CURVE QUESTION'
  CALL TRMNAT
ENDIF

IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) THEN
  WRITE(8,*) 'ERROR: INVALID TYPE OF MATERIALS PROCESS VARIATION DESIRED'
  CALL TRMNAT
ENDIF

READ(I,*) NBLIFE
IF (NBLIFE .GT. 0) READ(1,*) (BLFPER(J), J = 1, NBLIFE)

C ** CALL DRVHRIN TO READ DATA FROM BLDHCD AND ECHO DATA TO BLDHCO
CALL DRVHRIN (RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS, CM, CS, &
  RDM, RDS, LAMBA, LAMBB, LAMDA, LAMDB, MWA, MWB, &
  IMIN, MD, NB, NS, A0, A1, B0, B1, C0, C10, C11, &
  C20, C21)

C ** CALL INFAGG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, ZZERO, ZROREG, NUMREG, &
  NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)

IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)

MCOUNT = 0

C ** INITIALIZE VARIABLES
DO 35 K = 1, MAXLIF
  LIFE(K) = 1.0E+36
35 CONTINUE

NLIFET = NHYPER * NBLIFE

C ** OUTER LOOP – THIS LOOP SAMPLES HYPER-PARAMETER SETS
DO 150 K = 1, NHYPER

C ** CALL PAREST TO PERFORM THE PARAMETER ESTIMATION ASPECT OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, &
  RAND, NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)

C ** OBTAIN MATERIALS PROCESS VARIATION IF DESIRED
CALL NORMGN (RAND, 0.0, PSIG, LNZ)

  IF (MPROC .EQ. 1) THEN
    Z = EXP (LNZ)
  ELSE
    KRATIO = 1.0
    Z = 1.0
    LNZ = 0.0
  ENDIF

M_COUNT = M_COUNT + 1
DO 175 L = 1, NUMREG
  ALLM(M_COUNT, L) = MM(L)
175 CONTINUE

C ** INNER LOOP – THIS LOOP GENERATES BLADE FAILURE TIMES
DO 200 I = 1, NBLIFE

7 - 273
C ** INITIALIZE S/N CURVE PARAMETERS

DO 225 L = 0, MAXREG
  LNA(L) = 0.0
  LPHIM(L) = 0.0
  TRSBND(L) = 0.0
225 CONTINUE

C ** CALL SELECT TO "SELECT" DRIVERS FOR CALCULATING LIFE

CALL SELECT (RAND, RPM, RPMs, RROOT, RROOTM, &
  RROOTS, RAVG, RAVGm, RAVGn, C, CM, CS, RD, &
  RDM, RDS, LAMB, LAMBn, LAMBb, LAMD, LAMDA, &
  LAMDB, MW, MWA, MWB)

MINPHI = 1.0E+36
DO 230 M = i, NSYM
  CALL WEIBGN (BZERO, RAND, PHI)
  MINPHI = MIN (PHI, MINPHI)
230 CONTINUE

PHI = MINPHI

IF (VARY .EQ. 0) PHI = 1.0

IF (IOUT .EQ. 15) WRITE(8,*) 'PHI = ', PHI

C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS

FACTR = PHI * KRATIO * Z
DO 235 L = ZROREG, NUMREG
  TRSBND(L) = FACTR * SBND(L)
  TRBIGK(L) = BIGK(L)
235 CONTINUE

TRSBND(0) = SBND(0)

IF (ZROREG .EQ. 0) CALL KOMO (SZERO, BIGK, MM, NBND, &
  TRSBND, TRBIGK, FACTR, NUMREG)

DO 250 L = ZROREG, NUMREG
  LNA(L) = MM(L) * ALOG(TRBIGK(L))
  LPHIM(L) = MM(L) * ALOG(PHI)
  IF (IOUT .EQ. 15) THEN
    WRITE(8,*) 'L =', L, ' MM =', MM(L), ' BIGK =', BIGK(L), ' PHI =', PHI
    WRITE(8,*) 'LNA =', LNA(L), ' LPHIM =', LPHIM(L), ' SBND =', SBND(L)
    WRITE(8,*) 'KRATIO = ', KRATIO, ' Z = ', Z
  ENDIF
250 CONTINUE

C ** CALL BLDHLF TO OBTAIN BLADE HCF LIFE

NEWLIF = BLDHLF (RPM, RROOT, RAVG, C, RD, LAMB, LAMD, MW, &
  IMIN, MD, NB, NS, A0, A1, B0, B1, C0, C10, &
  C11, C20, C21, MM, LNA, LPHIM, KRATIO, &
  LN2, SBND, ZROREG, NUMREG, SZERO)

IF (IOUT .EQ. 15) WRITE(8,*), 'NEWLIF = ', NEWLIF

IF (NLIFET .GE. 100) CALL INSORT (NEWLIF, LIFE, NLIFET)

200 CONTINUE

C ** PRINT SORTED LIVES TO FILE LOWLIF

DO 300 J = 1, (NLIFET / 100)
  WRITE(9,*), J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
300 CONTINUE

C ** INITIALIZE VARIABLE BLFPOS()
DO 325 J = 1, MAXBLF
   BLFPOS(J) = 0
325  CONTINUE
FIFTY = 0.50E0

C ** PRINT EMPIRICAL BLIVES
WRITE(3,925)
DO 350 J = 1, NBLIFE
   BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
   WRITE(3,926) BLFPER(J), LIFE(NLIFET/2)
350  CONTINUE
WRITE(3,926) FIFTY, LIFE(NLIFET/2)

ENDIF
C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED
IF ((VARY .EQ. 3) .AND. (NMED .EQ. i)) THEN
   CALL SORTM (ALLM, NUMREG, MCOUNT)
   MID = MCOUNT / 2
   DO 400 L = 1, NUMREG
      MEDM(L) = ALLM(MID,L)
400  CONTINUE
   CALL EXPCRD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, 
                  NBND, BIGKI, BZERO)
ENDIF

925 FORMAT(///,2X,'B LIVES: ', EMPIRICAL',/)
926 FORMAT(2X,F7.5,5X,E13.6)

STOP
END
C 78 ................................ NUMBER OF STATOR VANES
C
COEFFICIENTS OF RESPONSE SURFACE FUNCTIONS

FLOW RATE:
\[ F_{\text{mdot}}(w) = A + B \times w \]
\[ A = 24.41242623 \]
\[ B = 0.3307822 \times 10^{-2} \]

ENTHALPY CHANGE:
\[ F_{\text{delta h}}(w) = A + B \times w \]
\[ A = 29.65037673 \]
\[ B = 0.6433368 \times 10^{-2} \]

BLADE DAMPER EFFECTIVENESS:
\[ \text{IF } mrw^2 < A \]
\[ \text{Feff}(m, r, w) = B + C \times mrw^2 \]
\[ \text{IF } mrw^2 > A \]
\[ \text{Feff}(m, r, w) = D + E \times mrw^2 \]
\[ A = 26 \]
\[ B = 1.0 \]
\[ C = -0.03750 \]
\[ D = 5.683003 \times 10^{-3} \]
\[ E = 7.429614 \times 10^{-4} \]

'RT, FWA 1480, 001 DIRECTION'...
MATERIAL DESCRIPTION

YIELD & ULTIMATE STRENGTHS, NDIV, NPTS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

# PTS IN DIV, STRESS RATIO, REGION

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>-1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>80000.0</td>
<td>6800.0</td>
</tr>
<tr>
<td>80000.0</td>
<td>15000.0</td>
</tr>
<tr>
<td>60000.0</td>
<td>27600.0</td>
</tr>
<tr>
<td>60000.0</td>
<td>43200.0</td>
</tr>
<tr>
<td>50000.0</td>
<td>139300.0</td>
</tr>
<tr>
<td>50000.0</td>
<td>54500.0</td>
</tr>
<tr>
<td>50000.0</td>
<td>137000.0</td>
</tr>
<tr>
<td>35000.0</td>
<td>4344800.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>137000.0</td>
<td>142600.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>137000.0</td>
<td>150000.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.683003E-3</td>
<td>7.429614E-4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1E+36</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

**LIST OF VARIABLES**

- **A0, A1** COEFFICIENTS OF THE FLOW RATE, m-dot, RESPONSE SURFACE
- **ALLM()** (PERFORMANCE BALANCE MODEL)
- **B0, B1** COEFFICIENTS OF THE ENTHALPY CHANGE, delta-h, RESPONSE SURFACE
- **BIGK()** (PERFORMANCE BALANCE MODEL)
- **BIGK1** 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
- **HIG** EQUAL TO BIGK(1) – DUMMY PARAMETER FOR CALLS TO SUBROUTINE
- **BDF7** REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND HCF LIFE CALCULATION
- **BPF** 1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
- **BLFPOS()** 1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPirical BLIVES
- **BZERO** VALUE OF WEIBULL PARAMETER, BETAC, CHARACTERIZING S/N DATA SET
- **C** SELECTED DISTANCE FROM NEUTRAL AXIS (in)
- **CM** COEFFICIENTS OF THE BLADE DAMPER EFFECTIVENESS RESPONSE SURFACE
- **C0, C10, C11, C20, C21** COEFFICIENTS OF THE BLADE DAMPER EFFECTIVENESS RESPONSE SURFACE
- **CS** STANDARD DEVIATION OF DISTANCE FROM NEUTRAL AXIS (in)
- **FACTR** SCALE FACTOR EQUAL TO PHI * KRATIO * Z
- **FITY** EQUAL TO .5 – USED TO ACCESS 50% POINT IN LIFE()
- **FTU** MATERIAL ULTIMATE STRENGTH (psi)
- **FY** MATERIAL YIELD STRENGTH (psi)
- **I** CONTROLS INNER DO LOOP
- **IMMN** MINIMUM MOMENT OF INERTIA (in**4)
- **IOUT** CONTROLS DUMP TO FILE IOUTPR
- **J** CONTROLS DO LOOP FOR EACH BLIFE
- **K** CONTROLS OUTER DO LOOP
- **KRATIO** RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
- **LAMB** SELECTED UNCERTAINTY IN PERFORMANCE BALANCE MODEL, LAMBDA
- **LAMBA** UNCERTAINTY IN PERFORMANCE BALANCE MODEL, LAMBDA, UNIFORM
**DISTRIBUTION LOWER BOUND**: LAMBbB, UNIFORM

**DISTRIBUTION UPPER BOUND**: LAMDB

**SELECTED UNCERTAINTY IN DAMPER COEFFICIENT OF FRICITION**: LAMDdD

**UNCERTAINTY IN DAMPER COEFFICIENT OF FRICITION**: LAMDB, UNIFORM

**1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM**

**1-D ARRAY CONTAINING VALUES OF THE LEFT-HAND TAIL**: LIFE()

**NORMAL(0, PVAR) GENERATED RANDOM VARIABLE**: LNA()

**1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION**: LPHIM()

**MAXIMUM NUMBER OF BLIVES TO BE PROVIDED**: MAXBLF

**MAXIMUM NUMBER OF POINTS PER DATA SET PER REGION ALLOWED**: MAXDAT

**MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA CALCULATION**: MAXIF

**MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION**: MAXMM

**MAXIMUM NUMBER OF REGIONS ALLOWED**: MAXREG

**NUMBER OF M's TO BE USED TO CALCULATE THE TRUNCATED NORMAL MEDIAN S/N CURVE**: MD

**DAMPER MASS (lbm)**

**1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION**: MEDM()

**POINTER TO THE MEDIAN M VALUES -- EQUAL TO HALF OF MCOUNT**: MINPHI

**EQUAL TO MIN(PHI) -- THE MINIMUM OF NSYM DRAWS OF PHI**: MM()

**1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION**: MPROC

**2-D ARRAY CONTAINING RANGEM() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS**

**SIZE OF OUTER LOOP**: NHYPER

**SIZE OF INNER LOOP**: NLIFE

**TOTAL NUMBER OF LIVES CALCULATED BY PFM**: LILNF

**CONTROLS MEDIAN CALCULATION FOR THE TRUNCATED NORMAL DISTRIBUTION CASE -- 0 -- NO MEDIAN CALCULATION; 1 -- MEDIAN CALCULATION DESIRED**: NPTS()

**1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET**: NS

**NUMBER OF STATOR BLADES**: NSYM

**NUMBER OF REGIONS OF INTEREST**: NNUMREG

**WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE**: PSI

**EQUAL TO SQRT(PVAR) -- MATERIALS PROCESS STANDARD DEVIATION**: PVAR

**RANDOM NUMBER SEED**: RND

**2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1, L) IS THE LOWER BOUND AND RANGEM(2, L) IS THE UPPER BOUND**

**SELECTED BLADE AVERAGE RADIUS (in)**

**MEAN OF AVERAGE BLADE RADIUS**: RAVG

**MEAN OF AVERAGE BLADE RADIUS (in)**

**STANDARD DEVIATION OF AVERAGE BLADE RADIUS (in)**

**SELECTED DAMPER RADIUS (in)**

**MEAN OF DAMPER RADIO (in)**

**STANDARD DEVIATION OF DAMPER RADIUS (in)**

**SELECTED ROTOR SPEED (rpm)**

**MEAN OF ROTOR SPEED (rpm)**

**STANDARD DEVIATION OF ROTOR SPEED (rpm)**

**SELECTED BLADE ROOT RADIUS (in)**

**MEAN OF BLADE ROOT RADIUS (in)**

**STANDARD DEVIATION OF BLADE ROOT RADIUS (in)**

**2-D ARRAY CONTAINING THE STRESS VALUES (psi, R = -1.0)**

**CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()**: SIG()
DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

STR() 2-D ARRAY CONTAINING STRESS POINTS (STRESS RATIO = -1.0) FOR
SPECIAL MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS
SZERO STRESS TENSILE TEST POINT, S
TRBIGN() 1-D ARRAY CONTAINING VALUES OF BIGN() CORRECTED FOR SZERO,
PHI, KRATIO, AND Z
TRSBND() 1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR EACH
REGION CALCULATED FOR EACH TRIAL
VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 -
TRUNCATED NORMAL VARIATION
Z LOGNORMAL(0, PVAR) GENERATED RANDOM VARIATE
ZROREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 ZERO REGION EXISTS, 1 - NO ZERO REGION EXISTS

C******************************************************************************
Csubroutine drvrin reads and echoes the input data
CPROGRAMMER: L. NEWLIN
CDATE: 31OCT90 COMMENTS: 20APR92
CVERSION: BLHCF V1, V1.1

CCopyright (C) 1990, California Institute of Technology.
CU.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE DRVIRIN (RMPP, RPKS, RROOTM, RROOTS, RAVGM, RAVGS,
& CM, CS, RDM, RDS, LAMBA, LAMB, LAMDA,
& LAMDB, MW, MB, IMIN, MD, NB, NS, A0, A1,
& B0, B1, C0, C10, C11, C20, C21)

COUTPUT: RMPP, RPKS, RROOTM, RROOTS, RAVGM, RAVGS, CM, CS, RDM, RDS,
& LAMBA, LAMB, LAMDA, LAMDB, MW, MB, IMIN, MD, NB, NS,
& A0, A1, B0, B1, C0, C10, C11, C20, C21

CIMPLICIT NONE

COMMNO IOUT

INTEGER IOUT, NB, NS

REAL A0, A1, B0, B1, C0, C10, C11, C20, C21, CM, CS, IMIN,
& LAMBA, LAMB, LAMDA, LAMDB, MD, MW, MB, RAVGM, RAVGS,
& RDM, RDS, RMPP, RPKS, RROOTM, RROOTS

LIST OF VARIABLES

A0, A1 Coefficients of the flow rate, m-dot, response surface
B0, B1 Coefficients of the enthalpy change, delta-h, response surface
C0, C10, C11, C20, C21 Coefficients of the blade damper effectiveness response surface
CM Mean of distance from neutral axis (in)
CS Standard deviation of distance from neutral axis (in)
IMIN Minimum moment of inertia (in**4)

LAMBA Uncertainty in performance balance model, LAMbdAB, Uniform
distribution lower bound
LAMBB Uncertainty in performance balance model, LAMbdAB, Uniform
distribution upper bound
LAMDA Uncertainty in damper coefficient of friction, LAMbdAD, Uniform
distribution lower bound
LAMDB Uncertainty in damper coefficient of friction, LAMbdAD, Uniform
distribution upper bound
MD Damper mass (lbm)
MW A Walker m Uniform distribution lower bound
MBB Walker m Uniform distribution upper bound
NB Number of rotor blades
NS Number of stator vanes

7 - 278
C SUBROUTINE SELECT PERFORMS THE DRIVER SELECTION
C PROGRAMMER: L. NEWLIN
C DATE: 31OCT90    COMMENTS: 20APR92

READ(I,*)(RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS,& CM, CS, RDM, RDS,& LAMBA, LAMBDA, LAMDB, MWA, MB,& IMIN, MD, NB, NS,& A0, A1, B0, B1, C0, C10, C11, C20, C21)

WRITE(3,900) RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS,& CM, CS, RDM, RDS
WRITE(3,901) LAMBA, LAMBDA, LAMDB, MWA, MB
WRITE(3,902) IMIN, MD, NB, NS
WRITE(3,903) A0, A1, B0, B1, C0, C10, C11, C0, C20, C21

900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',& 'Technology. U.S. Government',/2X,'Sponsorship under ',& 'NASA Contract NAS7-918 is acknowledged.',///,33X,'INPUT DATA',///,14X,'DRIVERS',31X,'DISTRIBUTIONS',& //,49X,'N( ',MEAN, 'STD. DEV. )')

901 FORMAT(/,2X,'ROTOR SPEED VARIATION (rpm)',20X,& 'N(',F8.1,',',FS.I,,),& //,2X,'BLADE ROOT RADIUS (in)',25X,& 'N(',F6.3,',',EI0.3,'),& //,2X,'BLADE AVERAGE RADIUS (in)',22X,& 'N(',F6.3,',',EI0.3,'),& //,2X,'DISTANCE FROM NEUTRAL AXIS (in)',16X,& 'N(',F7.4,',',E9.2,'),& //,2X,'DAMPER RADIUS (in)',29X,'N(',F6.3,',',',E10.3,')')

902 FORMAT(/,2X,'UNCERTAINTY IN PERFORMANCE BALANCE',14X,& 'U(',F7.4,',',F8.4,')',& //,2X,'DAMPER COEFFICIENT OF FRICTION',18X,& 'U(',F7.4,',',F8.4,')',& //,2X,'WALKER m',40X,'U(',F7.4,',',F8.4,')'

903 FORMAT(/,2X,'OTHER GEOMETRIC INPUT',& //,2X,'MINIMUM MOMENT OF INERTIA (in**4)',19X,E10.4,& //,2X,'DAMPER MASS (lb)',36X,E11.5,& //,2X,'NUMBER OF ROTOR BLADES',29X,I2,& //,2X,'NUMBER OF STATOR VANES',29X,I2)

904 FORMAT(/,13X,'COEFFICIENTS OF RESPONSE SURFACE FUNCTIONS',& //,2X,'FLOW RATE:',& //,5X,'P',12.8,' + ',E14.7,' * w',& //,2X,'ENTHALPY CHANGE:',& //,5X,'F',12.8,' + ',E14.7,' * w',& //,2X,'BLADE DAMPER EFFECTIVENES:',& //,5X,'IF m',12.8,' < ',F4.1,& //,10X,'F',12.8,' + ',E14.7,' * m',12.8,'**2',& //,5X,'IF m',12.8,' > ',F4.1,& //,10X,'F',12.8,' + ',E14.7,' * m',12.8,'**2'

RETURN
END
VERSION: BLDHC V1, V1.1

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE SELECT (RAND, RPM, RPMM, RPM, RROOT, RROOTM, & RROOTS, RAVG, RAVG, RAVGS, C, CM, CS, RD, & RDM, RDS, LAMB, LAMBA, LAMBB, LAMDA, LAMDB, MW, MWA, MWB)

C INPUT: RAND, RPMM, RPM, RROOT, RROOTM, RROOTS, RAVG, RAVG, RAVGS, CM, CS, RD, RDM, RDS, LAMB, LAMBA, LAMBB, LAMDA, LAMDB, MW, MWA, MWB
C OUTPUT: RPM, RROOT, RAVG, C, RD, LAMB, LAMDA, MW
C SUBPROGRAMS: NORMGN, PRYRV

IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL C, CM, CS, DUM, LAMB, LAMBA, LAMBB, LAMDA, LAMDB, & MW, MWA, MWB, RAVG, RAVG, RAVGS, RD, RDM, RDS, RPM, & RPM, RPMM, RPM, RROOT, RROOTM, RROOTS

DOUBLE PRECISION RAND

LIST OF VARIABLES

C Selected distance from neutral axis (in)
CM Mean of distance from neutral axis (in)
CS Standard deviation of distance from neutral axis (in)
DUM Dummy variable
IOUT Output dump controller
LAMB Selected uncertainty in performance balance model, LAMbdada
LAMBA Uncertainty in performance balance model, LAMbdada, Uniform distribution lower bound
LAMBB Uncertainty in performance balance model, LAMbdada, Uniform distribution upper bound
LAMDA Uncertainty in damper coefficient of friction, LAMbdad
LAMDB Uncertainty in damper coefficient of friction, LAMbdad, Uniform distribution lower bound
LAMDAB Uncertainty in damper coefficient of friction, LAMbdad, Uniform distribution upper bound
MW Selected Walker m
MWA Walker m Uniform distribution lower bound
MWB Walker m Uniform distribution upper bound
RAND Random number seed
RAVG Selected blade average radius (in)
RAVG Mean of average blade radius (in)
RAVGS Standard deviation of average blade radius (in)
RD Selected damper radius (in)
RDM Mean of damper radius (in)
RDS Standard deviation of damper radius (in)
RPM Mean of rotor speed (rpm)
RPMM Mean of rotor speed (rpm)
RPMS Standard deviation of rotor speed (rpm)
RROOT Selected blade root radius (in)
RROOTM Mean of blade root radius (in)
RROOTS Standard deviation of blade root radius (in)

CALL NORMGN (RAND, RPM, RPMM, RPM, RPM)
CALL NORMGN (RAND, RROOTM, RROOTS, RROOT)
CALL NORMGN (RAND, RAVG, RAVG, RAVG)
CALL NORMGN (RAND, CM, CS, CM)
CALL NORMGN (RAND, RDM, RDS, RD)
CALL PRYRV (RAND, MWA, MWB, MWB, MWA, MWB, DUM)
CALL PRYRV (RAND, MWA, MWB, MWA, MWB, MWB, DUM)

7 - 280
IF (IOUT .EQ. 15) THEN
  WRITE(8,*) 'RPM = ', RPM, ' RROOT = ', RROOT
  WRITE(8,*) 'RAVG = ', RAVG, ' C = ', C
  WRITE(8,*) 'RD = ', RD, ' MW = ', MW
  WRITE(8,*)
  'RD = ', RD, ' MW = ', MW
ENDIF
RETURN
END

C**********************************************************************
C FUNCTION BLDHLF PERFORMS/CONTROLS THE DRIVER TRANSFORMATION AND LIFE
C CALCULATION FOR THE BLADE HCF MODEL
C PROGRAMMER: L. NEWLIN
C DATE : 20APR92
C VERSION: BLDHCF V1.1 (MATCHV V8.5)
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

FUNCTION BLDHLF (RPM, RROOT, RAVG, C, RD, LAMB, LAMD, MW, IMIN,
& MD, NB, NS, A0, A1, B0, B1, C0, C10, C11, C20, C21, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG,
& NUMREG, SZERO)

  IMPLICIT NONE
  COMMON IOUT
  INTEGER IOUT, MAXREG, NB, NS, NUMREG, ZROREG
  PARAMETER (MAXREG = 3)
  REAL
  A0, A1, B0, B1, BLDHLF, C, C0, C10, C11, C20, C21,
  & DELTAH, GTLIFE, IMIN, KRATIO, LAMB, LAMD, LIFE,
  & LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MD, MDOT,
  & MM(0:MAXREG), MRW2, MW, NF, R, RAVG, RD, RPM, RROOT,
  & SALT, SBND(0:MAXREG), SBRM, SDSUD, SEQ, SMAX, SMean,
  & SMIN, SUD, SZERO

  LIST OF VARIABLES
  A0, A1 Coefficients of the flow rate, m-dot, response surface
  (performance balance model)
  B0, B1 Coefficients of the enthalpy change, delta-h, response surface
  (performance balance model)
  BLDHLF Real function performing the driver transformation and HCF life
  calculation
  C Selected distance from neutral axis (in)
  C0, C10, C11, C20, C21
  Coeficients of the blade damper effectiveness response surface
  DELTAH Enthalpy change, delta-h
  GTLIFE Function which calculates the cycles to failure at a given stress
  IMIN Minimum moment of inertia (in**4)
  IOUT Output dump controller
  KRATIO Ratio of K*/K, constant over regions and components
  LAMB Selected uncertainty in performance balance model, LAMbdab
  LAMD Selected uncertainty in damper coefficient of friction, LAMbdad
  LIFE Fatigue life in seconds

  7 - 281
C LNA()  1-D array containing ln(A) = ln(BIGK) * MM for each region
C LN2  Normal(0,PVAR) generated random variate
C LHIM()  1-D array containing ln(PHI) * MM for each region
C MAXREG  Maximum number of regions allowed
C MD  Damper mass (lbm)
C MDOT  Flow rate, m-dot
C MM()  1-D array containing selected values of m for each region
C MRW2  Damper normal load (lbf)
C MW  Selected Walker m
C NB  Number of rotor blades
C NF  Fatigue life in cycles
C NUMREG  Number of regions of interest
C NS  Number of stator vanes
C R  Stress ratio
C RAVG  Selected blade average radius (in)
C RD  Selected damper radius (in)
C RPM  Selected rotor speed (rpm)
C RROOT  Selected blade root radius (in)
C SALT  Alternating stress (psi)
C SBND()  1-D array containing the stress values (psi, R = -1.0) corresponding to the “life boundary” values for each region contained in NBND()
C SBRM  Blade root mean stress (psi)
C SDSUD  Ratio of damped to undamped vibratory stress
C SEQ  Equivalent zero mean stress (psi)
C SMAX  Maximum or peak stress (psi)
C SMEAN  Mean stress (psi)
C SMIN  Minimum stress (psi)
C SUD  Blade undamped vibratory stress (psi)
C SZERO  Stress tensile test point, So
C ZROREG  Zero Region — values chosen to facilitate region DO loop beginning value — 0 — zero region exists, 1 — no zero region

IF (IOUT .EQ. 15) THEN
  WRITE(8,*) 'RPM = ', RPM, ' RROOT = ', RROOT
  WRITE(8,*) 'RAVG = ', RAVG, ' C = ', C
  WRITE(8,*) 'RD = ', RD, ' MW = ', MW
  WRITE(8,*) 'LAMB = ', LAMB, ' LAMD = ', LAMD
  WRITE(8,*) 'IMIN = ', IMIN, ' MD = ', MD
  WRITE(8,*) 'NB = ', NB, ' NS = ', NS
ENDIF

CALCULATE FLOW CONDITIONS

MDOT = LAMB * (A0 + A1 * RPM)
DELTAH = LAMB * (B0 + B1 * RPM)

CALCULATE BLADE ROOT MEAN STRESS

SBRM = (MDOT * DELTAH / RPM) * (C / (IMIN * FLOAT (NB)))
  &     * (1.0 - (RROOT / RAVG)) * 9336

OBTAIN BLADE UNDAMPED VIBRATORY STRESS

SUD = (8.55300181 + 34.06551173 * (SBRM / 9336)) * 1000.0

CALCULATE DAMPER NORMAL LOAD

MRW2 = (MD * RD * (RPM ** 2)) * 2.83805E-5

OBTAIN BLADE DAMPER EFFECTIVENESS — THE RATIO OF THE DAMPED TO UNDAMPED VIBRATORY STRESS

IF (MRW2 .LT. CO) THEN
  SDSUD = LAMD * (C10 + C11 * MRW2)
ELSE
  SDSUD = LAMD * (C20 + C21 * MRW2)
ENDIF

CALCULATE ALTERNATING & MEAN STRESSES, MAX & MIN STRESSES, AND THE STRESS RATIO

SALT = SUD * (SDSUD)

7 - 282
C
SMEAN = SBRM
SMAX = SMEAN + SALT
SMIN = SMEAN - SALT
R = SMIN / SMAX
C
CALCULATE EQUIVALENT ZERO MEAN STRESS USING WALKER RELATION
SEQ = SMAX * ((1.0 - R) / 2.0) ** MW
C
OBTAIN FATIGUE LIFE (IN CYCLES) FROM MATERIALS MODEL
NF = GTLIFE (SEQ, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, &
NUMREG, SZERO)
C
TRANSFORM LIFE FROM CYCLES TO SECONDS
LIFE = (60.0 / RPM) * NF / FLOAT (NS)
BLDHLF = LIFE
IF (IOUT .EQ. 15) THEN
WRITE(8,*) 'MDOT = ', MDOT, ' DELTAH = ', DELTAH
WRITE(8,*) 'SBRM = ', SBRM, ' SUD = ', SUD
WRITE(8,*) 'MRW2 = ', MRW2, ' SDSUD = ', SDSUD
WRITE(8,*) 'SALT = ', SALT, ' SMEAN = ', SMEAN
WRITE(8,*) 'SMAX = ', SMAX, ' SMIN = ', SMIN
WRITE(8,*) 'R = ', R, ' SEQ = ', SEQ
WRITE(8,*) 'NF = ', NF, ' LIFE = ', LIFE
ENDIF
RETURN
END

C******************************************************************************
C
SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)
C
C IMPLICIT NONE
INTEGER MAXLIF
PARAMETER (MAXLIF = I0000)
COMMON IOUT
INTEGER I, IOUT, NLIFET, NUM, PLACE
REAL LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

LIST OF VARIABLES
I
IOUT
LIFE() 1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
PFM TO BE SORTED
MAXLIF MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
CALCULATION
NEWLIF LIFE VALUE TO BE INSERTED INTO LIFE()
C NLIFET TOTAL NUMBER OF LIVES CALCULATED BY PFM
C NUM NUMBER OF LIFE VALUES IN LIFE()
C PLACE POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C TEMP() 1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
INsertion OF NEWLIF

NUM = NLIFET / 2

C FIND POSITION IN LIFE() FOR NEWLIF
IF (NEWLIF .GT. LIFE(NUM)) GOTO 400
DO 100 I = 1, NUM
   IF (NEWLIF .LT. LIFE(I)) THEN
      PLACE = I
   GOTO 110
100 CONTINUE
110 CONTINUE

C STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()
DO 200 I = (PLACE + 1), NUM
   TEMP(I) = LIFE(I-1)
200 CONTINUE

C INSERT NEWLIF
LIFE(PLACE) = NEWLIF

C SHIFT VALUES OF LIFE() FOLLOWING NEWLIF
DO 300 I = (PLACE + 1), NUM
   LIFE(I) = TEMP(I)
300 CONTINUE

C IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN
400 CONTINUE
RETURN
END

**************************************************************************************
C SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C INDEPENDENT RANDOM VARIATES
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: RANDOM
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
**************************************************************************************
SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

COMMON IOUT
DOUBLE PRECISION RAND
REAL FRAC, RHO1, RHO2, THE1, THE2, X, Y
INTEGER IOUT

CALL RANDOM (FRAC, RAND)
C IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
X = FRAC * (RHO2 - RHO1) + RHO1
CALL RANDOM (FRAC, RAND)
C IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC

7 - 284
Y = FRAC * (THE2 - THE1) + THE1

IF (IOUT .EQ. 15) WRITE(8,*), 'RHOI =', RHOI, ' RHO2 =', RHO2,

RETURN
END

C SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C FOR THE STRESS FORMULATION
C PROGRAMMER: L. NEWLIN
C DATE : 13JUL89 FORMAT/COMMENTS: 12AUG91
C VERSION: MATCHR V8.4, V8.5 MATGRM V4.4, V4.5
Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
& NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
& KRATIO, PVAR)

C INPUTS: READS DATA FROM SPECFD AND RELATD; VARY, MPROC
C OUTPUTS: RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG,
& NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C SUBPROGRAMS: INIT, RCE, SW2SU2, FINDMC, INTRLV, FNDNG, ADDRNG,
C CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTVPAR
C FILES: 5:RELATD-OLD; 6:RELATO-NEW
C IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
& NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
& NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG
REAL
& BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
& FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
& KRATIO, LAMN, LNPV(MAXDAT, 0:MAXSET, MAXREG),
& LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
& MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG),
& MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
& PVAR, RANGEM(2, MAXREG), RAWSTR(MAXDAT, 0:MAXSET),
& RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
& SIG(MAXREG), SIGMA2(MAXREG), STR[MAXDAT, MAXREG],
& SUHAT2[MAXREG], SWHAT2[MAXREG], SX2[MAXREG],
& SY2[MAXREG], SZERO

LIST OF VARIABLES

BIGKHT EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE S/N
DATA SET
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
COEFFICIENT OF VARIATION, CO
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
AND SIG() CALCULATION
FTUZ ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTYZ YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
IOUT OUTPUT DUMP CONTROLLER
IZERO() 2-D ARRAY CONTAINING IO, THE 95% CONFIDENCE INTERVALS ON C
FOR EACH REGION
JZERO() 2-D ARRAY CONTAINING JO, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LAMN LAMBDA-N = RATIO OF VAR(Ln N given S) / (m**2 C**2), CONSTANT OVER REGIONS AND COMPONENTS
LNRF() 3-D ARRAY CONTAINING LN(RAWRF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) = -DO, THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION
MEDM() 1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MNPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MPROC Materials Process variation -- CONTROLS MATERIALS PROCESS VARIATION -- 0 - NO VARIATION; 1 - VARIATION
MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(I,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NNODAT Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))/-1) OVER ALL DATA SETS IN A REGION (NUMBER OF POINTS PER REGION)
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMREG NUMBER OF REGIONS OF INTEREST
PVAR MATERIALS PROCESS VARIATION
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
RATSTR() 2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
RAWRF() 2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS
RAWSTR() 2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
REFNP() 1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
SUHAT() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION (Y = Ln N)
SZERO STRESS TENSILE TEST POINT, SO
C VARY CONTROLS TYPE OF CURVE VARIATION DESIRED — 0 - NO
VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
C ZROREG Zero Region — VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION EXISTS

OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
OPEN(6, FILE = 'RELATO', STATUS = 'NEW')

C RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET INFORMATION
C PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION
C INITIALIZE PRIMARY ARRAYS
CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)
C READ, CONVERT, ECHO INFORMATION
CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR,
& LNNF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NNODAT,
& NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO,
& SIGMA2, KRAITO, LAMN)
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
& SWAT2, SUHAT2, NPPR)
C CALCULATE M CONTRAINT BASED ON CO
CALL FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN
C CALCULATIONS FOR ALL TYPES OF VARIATION SAVE NORMAL
C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS
CALL INTRVL (NUMREG, SX2, DD, SWAT2, SUHAT2, NPPR, IZERO,
& JZERO, MCHAT)
C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
IF (MPROC .EQ. 1) THEN
CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
ENDIF
C COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C OBTAIN POSTERIOR RANGES ON M
CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
& RANGEM)
C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONVEXITY CONSTRAINTS
CALL CONCAV (NUMREG, RANGEM)
C WRITE RESULTS TO FILE DUMP
WRITE(7,900)
DO 25 L = 1, NUMREG
WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
& JZERO(1, L), JZERO(2, L)
25 CONTINUE
CONTINUE
WRITE(7,910)
DO 50 L = 1, NUMREG
WRITE(7,915) L, MCHAT(2,L), MCHAT(1,L)
50 CONTINUE
IF (CZERO .GT. 0.0) THEN
WRITE(7,960)
DO 150 L = 1, NUMREG
IF (MCPNT(L) .EQ. 1) THEN
WRITE(7,965) L, MC(1,L)
ELSEIF (MCPNT(L) .EQ. 2) THEN
WRITE(7,970) L, MC(1,L), MC(2,L)
ENDIF
150 CONTINUE
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 100 L = 1, NUMREG
WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
100 CONTINUE
WRITE(7,950)
C CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO
CALL MEDIAN (NUMREG, RANGEM, MEDM)
C CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
CALL EXPCTD (I, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG, &
NEND, BIGKHT, BZERO)
C CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED
IF (((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
DO 200 L = I, NUMREG
RANGEM(I,L) = MEDM(L)
RANGEM(2,L) = MEDM(L)
200 CONTINUE
ENDIF
ELSE
C NORMAL VARIATION IS DESIRED
C CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION
CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, &
SIGMA2, MCHAT, MU, SIG)
C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
IF (MPROC .EQ. 1) THEN
CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
ENDIF
C COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M
CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, &
MPNT, MO, SIGMA2)
C ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
CALL CONCAV (NUMREG, RANGEM)
C WRITE RESULTS TO FILE DUMP
WRITE(7,975)
DO 350 L = 1, NUMREG
WRITE(7,980) L, MCHAT(I,L)
350 CONTINUE
IF (CZERO .GT. 0.0) THEN
WRITE(7,960)
DO 360 L = 1, NUMREG
IF (MCPNT(L) .EQ. 1) THEN
WRITE(7,965) L, MC(I,L)
ELSEIF (MCPNT(L) .EQ. 2) THEN
WRITE(7,970) L, MC(I,L), MC(2,L)
ENDIF
360 CONTINUE
ENDIF
WRITE (7,920)
WRITE(7,930)
DO 370 L = 1, NUMREG
WRITE(7,940) L, RANGEM(I,L), RANGEM(2,L)
370 CONTINUE
WRITE(7,950)
WRITE(7,985)
DO 380 L = 1, NUMREG
WRITE(7,990) L, MU(L), SIG(L)
380 CONTINUE
ENDIF
C PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS
IF (MPROC .EQ. 1) THEN
WRITE(7,995) PVAR
ENDIF
C FORMAT STATEMENTS
900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
& 'Technology. U.S. Government',/,,2X,'Sponsorship under ',
& 'NASA Contract NAS7-918 is acknowledged.',//,,
& 'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
& ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
& 'FOR EACH REGION',//)
905 FORMAT(7X,'REGION: ',I1,7X,'Io = (',F12.9,',' ',F12.9,')',
& ',2X,'Jo = (',F12.9,',' ',F12.9,')')
910 FORMAT(///,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
& ///,7X,'REGION',8X,'E(C)',I2X,'E(m)',//)
915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)
920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH ',
& 'REGION')
930 FORMAT(///,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',//)
940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)
950 FORMAT///,
960 FORMAT(///,2X,'RANGE ON m FOR EACH REGION IMPLIED BY C ',
& 'CONSTRAINT',
& ///,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',//)
965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')
970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)
SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED

PROGRAMMER: L. NEWLIN

DATE: 5OCT87


SUBROUTINE TRMNAT

WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'

STOP

END

SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG

PROGRAMMER: L. NEWLIN

DATE: CODE: 21JUN88 COMMENTS: 13JUL89

VERSION: MATCHR V8.1, V8.2, V8.3, V8.4, V8.5


SUBROUTINE INIT (NPTS, RRAWF, RAWSTR, RATSTR, NP, LNNF, LNSTR,

& REFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

INPUTS: --

OUTPUTS: NPTS, RRAWF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,

& NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET

PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

COMMON IOUT

INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),

& NPTS(0:MAXSET), REFNP(MAXREG)

REAL DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),

& LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
LIST OF VARIABLES

DELTA()  1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
IOUT  CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
JOUT  OUTPUT DUMP CONTROLLER
J  CONTROLS DO LOOP FOR EACH DATA SET
K  CONTROLS DO LOOP FOR EACH POINT IN A REGION
L  CONTROLS DO LOOP FOR EACH REGION
LNRF()  3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
LNSTR()  3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT  MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET  MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
RATSTR() 2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
RAWNF() 2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS
RAWSTR() 2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OF TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
REFNP() 1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

DO 100 J = 0, MAXSET
    NPTS(J) = 0.0
100 CONTINUE

DO 200 L = 1, MAXREG
    DO 250 J = 0, MAXSET
        NP(J, L) = 0.0
    250 CONTINUE
200 CONTINUE

DO 300 J = 0, MAXSET
    DO 350 I = 1, MAXDAT
        RAWNF(I, J) = 0.0
        RAWSTR(I, J) = 0.0
    350 CONTINUE
300 CONTINUE

DO 400 L = 1, MAXREG
    DO 450 K = 1, MAXDAT
        NF(K, L) = 0.0
        STR(K, L) = 0.0
    450 CONTINUE
400 CONTINUE

DO 500 K = 1, MAXDAT
    DO 550 L = 1, MAXREG
        NF(K, L) = 0.0
        STR(K, L) = 0.0
    550 CONTINUE
500 CONTINUE
SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, & LNSTR, LNNF, REFNP, STR, NF, ZRORF, ZROREG, & NUMREG, NODAT, NSETS, NUM, ZRORF, MPNT, MZERO, & FTU2, FTUZ, DELTA, MO, SIGMA2, KRATIO, LAMN)
C INPUTS: VARY, MPROC
C OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP, & STR, NF, ZRORF, ZROREG, NUMREG, NODAT, NSETS, NUM, ZRORF, MPNT, MZERO, & FTU2, FTUZ, DELTA, MO, SIGMA2, KRATIO, LAMN
C IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER COUNT, I, IOUTF, J, K, L, M, MPNT(MAXREG), MPROC, NDIV, & NODAT, NF(0:MAXSET, MAXREG), NPTS(0:MAXSET), NSETS, & NUM, NUMREG, REFNP(MAXREG), REG, VARY, ZROREG
REAL CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTY2, & KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG), & LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG), & MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG), & RATIO, RATSTR(MAXDAT, 0:MAXSET, RAWNF(MAXDAT, 0:MAXSET), & RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG), & SIG(1), STR(MAXDAT, MAXREG), ZRORF
CHARACTER*40 DECP(0:MAXSET)

LIST OF VARIABLES
COUNT INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO, CONVERSION, AND BREAK UP
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, CO
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
DESCRP() 1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
FTU
ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
FTUZ
ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTY
YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTYZ
YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
I
CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT
OUTPUT DUMP CONTROLLER
J
CONTROLS DO LOOP FOR EACH DATA SET
K
CONTROLS DO LOOP FOR EACH POINT IN A REGION
KRATIO
RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L
CONTROLS DO LOOP FOR EACH REGION
LAMN
LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2), CONSTANT OVER ALL REGIONS AND COMPONENTS
LNlf( )
3-D ARRAY CONTAINING LN(RAWNF( )), ALSO INDEXED FOR REGION
LNSTr( )
3-D ARRAY CONTAINING LN(RATSTR( )), ALSO INDEXED FOR REGION
M
CONTROLS DO LOOP FOR EACH DATA DIVISION
MAXDAT
MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG
MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET
MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO( )
1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT( )
1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO( ) FOR EACH REGION
MPROC
Materials Process variation -- CONTROLS MATERIALS PROCESS VARIATION - 0 - NO VARIATION; 1 - VARIATION
MZERO( )
2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON N FOR EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NBND( )
1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NDIV
NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO, REGION PAIRS DURING INPUT
NF( )
2-D ARRAY CONTAINING RAWNF( ) (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NNODAT
Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
Np( )
2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NPTS( )
1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
NSETS
NUMBER OF RELATED MATERIAL S/N DATA SETS
NUM
NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
NUMREG
NUMBER OF REGIONS OF INTEREST
RATIO
STRESS RATIO (R = -1.0 IS DESIRED)
RATSTR( )
2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
RAWNF( )
2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS
RAWSTR( )
2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
REFNP( )
1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
REG
REGION OF INTEREST IN A PARTICULAR DIVISION
SIGMA2( )
1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
STR( )
2-D ARRAY CONTAINING RATSTR( ) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
SZERO
STRESS TENSILE TEST POINT, SO VARY
CURRENT TYPE OF CURVE VARIATION DESIRED - 0 - NO VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
Zzero
Zero region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C
initialize count and NBND( )
COUNT = 0
DO 10 L = 0, MAXREG
   NBND(L) = 0.0
10 CONTINUE
C
input data on specific material from specfd and echo to specfo
READ(1,*), DESCRP(0), FTU, FTUZ, NDIV, NPTS(0)
IF (NPTS(0) .GT. MAXDAT) THEN
  WRITE(6,*): 'ERROR: OVER NUMBER OF POINTS LIMIT IN ',
  & 'SPECIFIC MATERIAL'
  CALL TRMNAT
ENDIF

WRITE(3,900) DESCP(0), FTU, FTU, NPTS(0)
IF (IOUT .EQ. 10) WRITE(8,900) DESCP(0), FTU, FTU, NPTS(0)
WRITE(3,905)
IF (IOUT .EQ. 10) WRITE(8,905)

C STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ
FTUZ = FTU
FTYZ = FTY

C INPUT STRESS/LIFE INFORMATION — INCLUDING STRESS RATIO AND REGION
C INFORMATION FROM SPECFD AND ECHO TO SPECFO
DO 100 M = 1, NDIV
  READ (1,*), NUM, RATIO, REG
  IF (ABS(RATIO) .GT. 1.0) THEN
    WRITE(6,*), 'ERROR: INVALID VALUE FOR RATIO: ', RATIO
    CALL TRMNAT
  ENDIF
  IF (REG .GT. MAXREG) THEN
    WRITE(6,*), 'ERROR: OVER REGION LIMIT IN SPECIFIC DATA SET'
    CALL TRMNAT
  ENDIF
  DO 110 I = (COUNT + 1), (COUNT + NUM)
    RATSTR(I,0) = RAWSTR(I,0)
  CONTINUE
  IF (RATIO .EQ. -1.0) THEN
    STRESS RATIO IS CORRECT
    DO 120 I = (COUNT + 1), (COUNT + NUM)
      RATSTR(I,0) = RAWSTR(I,0)
    CONTINUE
  ELSE
    STRESS RATIO TRANSFORMATION MUST BE DONE
    CALL CONVRT (0, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR,
    & RATIO, FTU, FTY)
  ENDIF
C ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL
DO 130 I = (COUNT + 1), (COUNT + NUM)
  WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
  & RATSTR(I,0), RAWNF(I,0)
  IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
  & RATIO, REG, RATSTR(I,0), RAWNF(I,0)
  CONTINUE
C BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C EXPCTD, AND PAREST
K = NP(0,REG)
DO 140 I = (COUNT + 1), (COUNT + NUM)
\[ K = K + 1 \]
\[ \text{LNSTR}(K, 0, \text{REG}) = \text{ALOG} (\text{RATSTR}(I, 0)) \]
\[ \text{LNNF}(K, 0, \text{REG}) = \text{ALOG} (\text{RAWNF}(I, 0)) \]
\[ \text{STR}(K, \text{REG}) = \text{RATSTR}(I, 0) \]
\[ \text{NF}(K, \text{REG}) = \text{RAWNF}(I, 0) \]

140 CONTINUE

IF \( (K \gt \text{MAXDAT}) \) THEN
\> \text{WRITE}(8, \*) 'ERROR: OVER NUMBER OF POINTS LIMIT IN '
\> \& \text{SPECIFIC MATERIAL'}
\> \text{CALL TRMNAT}
\> \text{ENDIF}
NP(0,\text{REG}) = K
\> \text{REFNP}(\text{REG}) = K
\> \text{COUNT} = \text{COUNT} + \text{NUM}

100 CONTINUE

IF \( (\text{NPTS}(0) .\ne. \text{COUNT}) \) THEN
\> \text{WRITE}(8, \*) 'ERROR: NUMBER OF POINTS PER DIVISION ',
\> \& 'INCORRECTLY SPECIFIED'
\> \text{CALL TRMNAT}
\> \text{ENDIF}
\> \text{READ}(1, \*) \text{SZERO}
\> \text{IF} (\text{NINT} (\text{SZERO}) .\gt. 0) \text{THEN}
\> \> \text{ZROREG} = 0
\> \> \text{ELSE}
\> \> \text{ZROREG} = 1
\> \> \text{ENDIF}
\> \text{IF} (\text{IOUT} .\eq. 10)
\> \& \text{WRITE}(8, \*) 'SZERO = ', \text{SZERO}, ', \text{ZROREG} = ', \text{ZROREG}

C INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION
\> \text{READ}(1, \*) \text{NUMREG, NNODEAT}
\> \text{IF} ((\text{NUMREG} + \text{NNODEAT}) .\gt. \text{MAXREG}) \text{THEN}
\> \> \text{WRITE}(8, \*) 'ERROR: EXCEEDED LIMIT ON NUMBER OF REGIONS'
\> \> \text{CALL TRMNAT}
\> \text{ENDIF}
\> \text{DO 150 L = ZROREG, (\text{NUMREG} + \text{NNODEAT})}
\> \text{READ}(1, \*) \text{NBND}(L)
150 CONTINUE
\> \text{READ}(1, \*) \text{CZERO}
\> \text{DO 160 L = 1, (\text{NUMREG} + \text{NNODEAT})}
\> \text{READ}(1, \*) \text{MPNT}(L), \text{MZERO}(1,L), \text{MZERO}(2,L)
160 CONTINUE
\> \text{WRITE}(3,913)
\> \text{IF} (\text{ZROREG} .\eq. 0) \text{WRITE}(3,914) \text{SZERO}
\> \text{IF} (\text{IOUT} .\eq. 10) \text{THEN}
\> \> \text{WRITE}(8,913)
\> \> \text{IF} (\text{ZROREG} .\eq. 0) \text{WRITE}(8,914) \text{SZERO}
\> \text{ENDIF}
\> \text{WRITE}(3,915) \text{NUMREG, NNODEAT}
\> \text{IF} (\text{IOUT} .\eq. 10) \text{WRITE}(8,915) \text{NUMREG, NNODEAT}
\> \text{DO 170 L = ZROREG, (\text{NUMREG} + \text{NNODEAT})}
\> \text{WRITE}(3,920) \text{NBND}(L)
\> \text{IF} (\text{IOUT} .\eq. 10) \text{WRITE}(8,920) \text{NBND}(L)
170 CONTINUE
\> \text{WRITE}(3,925) \text{CZERO}
\> \text{IF} (\text{IOUT} .\eq. 10) \text{WRITE}(8,925) \text{CZERO}
\> \text{DO 180 L = 1, (\text{NUMREG} + \text{NNODEAT})}
WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
IF (IOUT .EQ. 10)
  WRITE(8,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
&
  IF ((VARY .EQ. 3) .AND. (MPNT(L) .EQ. 0)) THEN
    WRITE(8,*)'ERROR: NORMAL VARIATION REQUIRES A PRIOR '
&
    CALL TRMNAT
ENDIF
180 CONTINUE

IF (VARY .EQ. 3) THEN
  READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
  WRITE(3,945)
  IF (IOUT .EQ. 10) WRITE(8,945)
  DO 190 L = 1, (NUMREG + NMODAT)
    READ(I,*)(DELTA(L), MO(L), SIGMA2(L))
    WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
    IF (IOUT .EQ. 10) WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
    IF ((DELTA(L) .LT. 0.0) .OR. ((DELTA(L) .GT. 0.0) .AND. (MO(L) .LE. 0.0))) THEN
      WRITE(_,*) 'ERROR: BAD VALUE FOR DELTA OR VALUE OF MO '
      'INCONSISTENT WITH DELTA IN REGION ', L
      CALL TRMNAT
    ENDIF
  190 CONTINUE
ENDIF

IF (MPROC .EQ. 1) THEN
  READ(1,*) KRATIO, LAMN
  WRITE(3,955) KRATIO, LAMN
  IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
ENDIF

C BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATED AND THEN ECHO TO RELATO
READ(5,*) NSETS
IF (NSETS .GT. MAXSET) THEN
  WRITE(8,*)'ERROR: OVER LIMIT ON NUMBER OF RELATED DATA SETS'
  CALL TRMNAT
ENDIF
WRITE(6,935) NSETS
DO 200 J = 1, NSETS
  COUNT = 0
  IF (IOUT. EQ. 10) WRITE(8,*)'J = ', J, ' NSETS = ', NSETS
  READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)
  IF (NPTS(J) .GT. MAXDAT) THEN
    WRITE(8,*)'ERROR: OVER LIMIT ON NUMBER OF POINTS IN ', 
    'SET ', J
    CALL TRMNAT
  ENDIF
  WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
  IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)
WRITE(6,905)
DO 300 M = 1, NDIV
  READ(5,*) NUM, RATIO, REG
  IF (ABS(RATIO) .GT. 1.0) THEN
    WRITE(8,*)'ERROR: INVALID VALUE OF RATIO: ', RATIO
    CALL TRMNAT
  ENDIF
300 CONTINUE

7 - 296
IF (REG .GT. MAXREG) THEN
   WRITE(8,*), 'ERROR: OVER REGION LIMIT IN RELATED MATERIAL ', J
   CALL TRMNAT
ENDIF

IF (IOUT .EQ. 10) THEN
   WRITE(8,*), 'NUM = ', NUM, ' COUNT = ', COUNT
   WRITE(8,*), 'RATIO = ', RATIO, ' REG = ', REG
ENDIF

DO 310 I = (COUNT + 1), (COUNT + NUM)
   READ(5,*), RAWSTR(I,J), RAWNF(I,J)
310 CONTINUE

C CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
C
IF (RATIO .EQ. -1.0) THEN
   STRESS RATIO IS CORRECT
   DO 320 I = (COUNT + 1), (COUNT + NUM)
      RATSTR(I,J) = RAWSTR(I,J)
320 CONTINUE
ELSE
   STRESS RATIO TRANSFORMATION MUST BE DONE
   CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR, RATIO, FTU, FTY)
ENDIF

C RECORD BOTH S/N DATA SETS TO RELATO
DO 330 I = (COUNT + 1), (COUNT + NUM)
   WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG, RATSTR(I,J), RAWNF(I,J)
330 CONTINUE

K = NP(J,REG)
DO 340 I = (COUNT + 1), (COUNT + NUM)
   K = K + 1
   LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
   LNNF(K,J,REG) = ALOG(RAWNF(I,J))
340 CONTINUE

IF (K .GT. MAXDAT) THEN
   WRITE(8,*), 'ERROR: OVER LIMIT ON NUMBER OF POINTS ', 'IN SET ', J
   CALL TRMNAT
ENDIF

NP(J,REG) = K
COUNT = COUNT + NUM

300 CONTINUE

IF (NPTS(J) .NE. COUNT) THEN
   WRITE(8,*), 'ERROR: NUMBER OF POINTS PER DIVISION ', 'INCORRECTLY SPECIFIED IN SET ', J
   CALL TRMNAT
ENDIF

200 CONTINUE
SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)
C INPUTS: J, NUM1, NUM2, STR, RSTR, R, FTU, FTY
C OUTPUTS: RSTR
C IMPLICIT NONE

C THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C STRESS RATIO, R, IS NOT -1.0
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87  COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C          V8.3, V8.4, V8.5
C          SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)
C          INPUTS: J, NUM1, NUM2, STR, RSTR, R, FTU, FTY
C          OUTPUTS: RSTR
C          IMPLICIT NONE

C FORMAT STATEMENTS USED TO WRITE TO SPECFO AND RELATO

900 FORMAT(//,13X,'MATERIAL INPUT',//,2X,'DESCRIPTION:',2X,A40,//,
      2X,'YIELD STRENGTH',18X,E11.5,//,2X,'ULTIMATE STRENGTH',
      15X,E11.5,//,2X,'NUMBER OF POINTS',16X,I2)

905 FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
      //,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
      'STRESS',7X,'LIFE')

910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

913 FORMAT(//)

914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN SO OF',
      & 5X,E11.5)

915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA ',
      & //,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
      & //,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
      & '(Cycles): ')'//)

920 FORMAT(10X,E9.3)

925 FORMAT(//,2X,'EXOGENOUS INFORMATION',//,2X,
      & 'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
      & //,2X,'EXPLICIT CONSTRAINT ON $m$ FOR EACH REGION:',
      & //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
      & 5X,'UPPER BOUND')//

930 FORMAT(6X,II,IIX,
      & 'NUMBER OF DATA SETS:',2X,I2,//,
      & 'NOTE: ALL $K_1$ ASSUMED TO BE 1.0',///,23X,
      & 'TRANSFORMED DATA')//

935 FORMAT(//,2X,'DESCRIPTION:',2X,A40,
      & //,2X,'YIELD STRENGTH',18X,F7.0,
      & //,2X,'ULTIMATE STRENGTH',15X,F7.0,
      & //,2X,'NUMBER OF POINTS',16X,I2)

940 FORMAT(//,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS:',
      & //,2X,'REGION',5X,'DELTA',8X,'SIGMA^2',/)

945 FORMAT(5X,II,5X,F7.2,5X,F7.4,5X,EII.5)

950 FORMAT(5X,I1,5X,F7.2,5X,F7.4,5X,EII.5)

955 FORMAT(//,2X,'MATERIALS PROCESS VARIATION INFORMATION',
      & //,2X,'MEDK*/MEDK: ',5X,EII.5,/,5X,'LAMBDA: ',5X,EII.5)

RETURN
END

C**********************************************************************

C THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C STRESS RATIO, R, IS NOT -1.0
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87  COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C          V8.3, V8.4, V8.5
C          SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)
C          INPUTS: J, NUM1, NUM2, STR, RSTR, R, FTU, FTY
C          OUTPUTS: RSTR
C          IMPLICIT NONE

C**********************************************************************
INTEGER MAXDAT, MAXSET
PARAMETER (MAXDAT = 50, MAXSET = 5)
COMMON IOUT
INTEGER I, IOUT, J, NUM1, NUM2
REAL FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
& STR(MAXDAT, 0:MAXSET), TEST

LIST OF VARIABLES

FTU ULTIMATE STRENGTH OF MATERIAL (PSI)
FTY YIELD STRENGTH OF MATERIAL (PSI)
I CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
IOUT OUTPUT DUMP CONTROLLER
J DATA SET OF INTEREST
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
NUM1 FIRST INDEX TO BE TRANSFORMED
NUM2 LAST INDEX TO BE TRANSFORMED
R STRESS RATIO (R = -1.0 IS DESIRED)
RSTR() STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
STR() ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
TEST Kt * Smax * (I - R)/2 , TO BE COMPARED WITH FTY

C Kt IS ASSUMED TO BE ONE

DO 100 I = NUM1, NUM2
  TEST = STR(I,J) * (1.0 - R)/2.0
  IF (IOUT.EQ.10) WRITE(8,*),'I =',I,' J = ',J,' TEST = ',TEST

  IF (TEST .GE. FTY) THEN
    RSTR(I,J) = TEST
    IF (IOUT.EQ.10) WRITE(8,*),'1:RSTR() = ',RSTR(I,J)
  ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
    RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))
    IF (IOUT.EQ.10) WRITE(8,*),'2:RSTR() = ',RSTR(I,J)
  ELSE
    RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
      /(2.0 * FTU)))
  END IF

100 CONTINUE
RETURN
END

********************************************************************

C SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C AND, SWHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87 COMMENTS: 13JUL89

7 - 299
SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, 
& SY2, DD, SWHAT2, SUHAT2, NPPR)
C INPUTS: NUMREG, NSETS, NP, LNSTR, LNNF
C OUTPUTS: SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR
C
IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON I OUT
INTEGER IOUT, J, K, L, NP(0:MAXSET, MAXREG), NPPR(MAXREG), 
& NSETS, NUMREG
REAL
BB(MAXREG), DD(MAXREG), DIFFX(MAXDAT, 0:MAXSET), 
& DIFFY(MAXDAT, 0:MAXSET), LNNF(MAXDAT, 0:MAXSET, MAXREG), 
& LNSTR(MAXDAT, 0:MAXSET, MAXREG), MEANY(0:MAXSET), 
& MEANY(0:MAXSET), SUHAT2(MAXREG), SWHAT2(MAXREG), 
& SX2(MAXREG), SXY(MAXREG), SY2(MAXREG)
LIST OF VARIABLES
BB() 1-D ARRAY CONTAINING SX(L)/SY2(L) FOR EACH REGION
DD() 1-D ARRAY CONTAINING SX(L)/SX2(L) FOR EACH REGION
DIFFX() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L) AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
DIFFY() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L) AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
L CONTROLS DO LOOP FOR EACH REGION
LNNF() 3-D ARRAY CONTAINING LN(RAFNF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MEANX() 1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION L AND DATA SET J (X = Ln S)
MEANY() 1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION L AND DATA SET J (Y = Ln N)
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMREG NUMBER OF REGIONS OF INTEREST
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR EACH REGION (X = Ln S, Y = Ln N)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION (Y = Ln N)
C
INITIALIZE ARRAYS
DO 50 L = 1, MAXREG
SY2(L) = 0.0
SX2(L) = 0.0
SXY(L) = 0.0
SWHAT2(L) = 0.0
SUHAT2(L) = 0.0
C
50 CONTINUE
**BB(L)** = 0.0
**DD(L)** = 0.0
**NPPR(L)** = 0

50 CONTINUE

DO 60 J = 0, MAXSET
   DO 70 K = I, MAXDAT
      **DIFFY(K,J)** = 0.0
      **DIFFX(K,J)** = 0.0
   70 CONTINUE
   **MEANY(J)** = 0.0
   **MEANX(J)** = 0.0
60 CONTINUE

C NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION

DO 100 L = 1, NUMREG

   C NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION
   C
   C FIRST CALCULATE SAMPLE X AND Y MEANS
   C FOR DATA SET J IN REGION L
   MEANY(J) = 0.0
   MEANX(J) = 0.0
   IF (IOUT .EQ. IO) WRITE(8,*) 'L =', L, ' J =', J,
   & ' NP =', NP(J,L)

   DO 250 K = I, NP(J,L)
      MEANY(J) = MEANY(J) + LNNF(K,J,L)
      MEANX(J) = MEANX(J) + LNSTR(K,J,L)
   250 CONTINUE
   MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
   MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
   IF (IOUT .EQ. IO) WRITE(8,*) 'MEANY(J) =', MEANY(J),
   & ' MEANX(J) =', MEANX(J)

   C NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION
   C
   C OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
   C DATA SET IN REGION L
   DO 300 K = I, NP(J,L)
      **DIFFY(K,J)** = LNNF(K,J,L) - MEANY(J)
      **DIFFX(K,J)** = LNSTR(K,J,L) - MEANX(J)
      **SY2(L)** = **SY2(L)** + **DIFFY(K,J)** ** 2
      **SX2(L)** = **SX2(L)** + **DIFFX(K,J)** ** 2
      **SXY(L)** = **SXY(L)** + **DIFFX(K,J)** * **DIFFY(K,J)
      IF (IOUT .EQ. IO) THEN
         WRITE(8,*) 'K =', K, ' DIFFY(K,J) =', **DIFFY(K,J),
         & ' DIFFX(K,J) =', **DIFFX(K,J)
         WRITE(8,*) 'SY2(L) =', **SY2(L), ' SX2(L) =', **SX2(L),
         & ' SXY(L) =', **SXY(L)
      ENDIF
300 CONTINUE

   **NPPR(L)** = **NPPR(L)** + NP(J,L) - 1
   IF (IOUT .EQ. IO) WRITE(8,*) 'NPPR(L) =', **NPPR(L)
200 CONTINUE

   IF (**SXY(L)** .GE. 0.0) THEN
      C LIFE WILL INCREASE WITH INCREASING STRESS -- INVALID FOR
      C OUR MODEL
      WRITE(8,*) 'ERROR: **SXY** >= 0 IN REGION', L
      CALL TRMNAT
   ENDIF

   **NPPR(L)** = **NPPR(L)** - 1
   IF (**NPPR(L)** .LE. 0) THEN
      C TOO FEW POINTS FOR REGRESSION IN ', **REGION** ', L
      CALL TRMNAT
   ENDIF

7 - 301
SY2(L) = SX2(L) / FLOAT(NPPR(L))
SX2(L) = SX2(L) / FLOAT(NPPR(L))
SX2(L) = SYX(L) / FLOAT(NPPR(L))

NOW CALCULATE THE RESIDUAL VARIANCES, SWHAT2, SUHAT2, FOR EACH REGION FROM THE Y ON X AND X ON Y REGRESSIONS

DD(L) = SXY(L) / SX2(L)
BB(L) = SXY(L) / SY2(L)

IF (IOUT .EQ. I0) THEN
  WRITE(8,*), 'NPPR(L) =', NPPR(L), 'SY2(L) =', SY2(L), 'SX2(L) =', SX2(L), 'SXY(L) =', SXY(L), 'DD(L) =', DD(L), 'BB(L) =', BB(L)
ENDIF

DO 400 J = 0, NSETS
  IF (IOUT .EQ. I0) WRITE(8,*), 'J =', J, 'NP(J,L) =', NP(J,L)
  DO 500 K = i, NP(J,L)
    SWHAT2 (L) = SWHAT2 (L) + (DIFFY(K,J) - DD(L) * DIFFX(K,J)) ** 2
    SUHAT2 (L) = SUHAT2 (L) + (DIFFX(K,J) - BB(L) * DIFFY(K,J)) ** 2
  IF (IOUT .EQ. I0) WRITE(8,*), 'K =', K, 'SWHAT2(L) =', SWHAT2(L), 'SUHAT2(L) =', SUHAT2(L)
  500 CONTINUE
  400 CONTINUE

SWHAT2(L) = SWHAT2(L) / FLOAT(NPPR(L))
SUHAT2(L) = SUHAT2(L) / FLOAT(NPPR(L))

IF (IOUT .EQ. I0) WRITE(8,*), 'NPPR(L) =', NPPR(L), 'SWHAT2(L) =', SWHAT2(L), 'SUHAT2(L) =', SUHAT2(L)

100 CONTINUE

RETURN
END

C******************************************************************************

SUBROUTINE INTRVL CALCULATES THE 95\% CONFIDENCE INTERVAL, Io, ON C AND THE 95\% CONFIDENCE INTERVAL, Jo, ON M
PROGRAMMER: L. NEWLIN
DATE: CODE: 5OCT87 COMMENTS: 15SEP89


SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO, MCHAT)

INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
OUTPUTS: IZERO, JZERO, MCHAT
SUBPROGRAMS: TMNAT

IMPLICIT NONE
INTEGER CHITAB, MAXREG, TTAB
PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)
COMMON IOUT
INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG
REAL ARG, CH1025(CHITAB), CH1975(CHITAB), DD(MAXREG), IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG), SWHAT, SUHAT2(MAXREG), SWHAT2(MAXREG), SX, SX2(MAXREG), T, T025(TTAB)
VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:

1 - 30, 40, 50, 60, 70, 80, 90, 100 – Theil, pp. 718-719
   – Calculated using cube rule approximation

<table>
<thead>
<tr>
<th>DATA (CHI025(I, I = 1, 75) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00098266, 0.506356, 0.215795, 0.484419, 0.831211,</td>
</tr>
<tr>
<td>1.237347, 1.68987, 2.17973, 2.70039, 3.24697,</td>
</tr>
<tr>
<td>3.81575, 4.40379, 5.00874, 5.62872, 6.26214,</td>
</tr>
<tr>
<td>6.90766, 7.56418, 8.23075, 8.90655, 9.59083,</td>
</tr>
<tr>
<td>10.28293, 10.9823, 11.6885, 12.4011, 13.1197,</td>
</tr>
<tr>
<td>13.8439, 14.5733, 15.3079, 16.0471, 16.7908,</td>
</tr>
<tr>
<td>17.53, 18.28, 19.04, 19.80, 20.56,</td>
</tr>
<tr>
<td>21.33, 22.10, 22.87, 23.65, 24.431,</td>
</tr>
<tr>
<td>25.22, 25.99, 26.78, 27.57, 28.36,</td>
</tr>
<tr>
<td>29.15, 29.95, 30.75, 31.55, 32.354,</td>
</tr>
<tr>
<td>33.16, 33.96, 34.77, 35.58, 36.39,</td>
</tr>
<tr>
<td>37.21, 38.02, 38.84, 39.66, 40.4817,</td>
</tr>
<tr>
<td>38.30, 42.12, 42.95, 43.77, 44.60,</td>
</tr>
<tr>
<td>45.43, 46.26, 47.09, 47.92, 48.7576,</td>
</tr>
<tr>
<td>49.59, 50.42, 51.26, 52.10, 52.94 /</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA (CHI105(I, I = 1, 150) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>53.78, 54.62, 55.46, 56.30, 57.1532,</td>
</tr>
<tr>
<td>57.80, 58.84, 59.69, 60.54, 61.39,</td>
</tr>
<tr>
<td>62.24, 63.09, 63.94, 64.79, 65.6466,</td>
</tr>
<tr>
<td>66.50, 67.35, 68.21, 69.07, 69.92,</td>
</tr>
<tr>
<td>70.78, 71.64, 72.50, 73.36, 74.2219,</td>
</tr>
<tr>
<td>75.08, 75.94, 76.80, 77.67, 78.53,</td>
</tr>
<tr>
<td>79.40, 80.27, 81.13, 82.00, 82.87,</td>
</tr>
<tr>
<td>83.73, 84.60, 85.47, 86.34, 87.21,</td>
</tr>
<tr>
<td>88.08, 88.95, 89.83, 90.70, 91.57,</td>
</tr>
<tr>
<td>92.45, 93.32, 94.19, 95.07, 95.94,</td>
</tr>
<tr>
<td>96.82, 97.70, 98.57, 99.45, 100.33,</td>
</tr>
<tr>
<td>100.81, 101.06, 101.97, 103.85, 104.73,</td>
</tr>
<tr>
<td>105.61, 106.49, 107.37, 108.25, 109.14,</td>
</tr>
<tr>
<td>110.02, 110.90, 111.79, 112.67, 113.56,</td>
</tr>
<tr>
<td>114.44, 115.33, 116.21, 117.10, 117.98 /</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA (CHI975(I, I = 1, 75) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.02889, 7.37776, 9.34840, 11.1433, 12.8325,</td>
</tr>
<tr>
<td>14.4994, 16.0128, 17.5346, 19.0228, 20.4831,</td>
</tr>
<tr>
<td>21.9200, 23.3367, 24.7536, 26.1190, 27.4884,</td>
</tr>
<tr>
<td>28.8454, 30.1910, 31.5264, 32.8523, 34.1696,</td>
</tr>
<tr>
<td>35.4789, 36.7807, 38.0757, 39.3641, 40.6465,</td>
</tr>
<tr>
<td>41.9232, 43.1944, 44.4607, 45.7222, 46.9792,</td>
</tr>
<tr>
<td>48.23, 49.48, 50.72, 51.96, 53.20,</td>
</tr>
<tr>
<td>54.44, 55.67, 56.89, 58.12, 59.3417,</td>
</tr>
<tr>
<td>60.56, 61.77, 62.99, 64.20, 65.41,</td>
</tr>
<tr>
<td>66.62, 67.82, 69.02, 70.22, 71.4202,</td>
</tr>
<tr>
<td>72.61, 73.81, 75.00, 76.19, 77.38,</td>
</tr>
<tr>
<td>78.57, 79.75, 80.93, 82.12, 83.2976,</td>
</tr>
<tr>
<td>84.48, 85.65, 86.83, 88.00, 89.18,</td>
</tr>
<tr>
<td>90.65, 91.82, 92.98, 94.16, 95.25,</td>
</tr>
<tr>
<td>96.19, 97.35, 98.52, 99.68, 100.84 /</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA (CHI975(I, I = 76, 150) /</th>
</tr>
</thead>
<tbody>
<tr>
<td>102.00, 103.16, 104.31, 105.47, 106.62,</td>
</tr>
<tr>
<td>107.78, 108.94, 110.09, 111.24, 112.39,</td>
</tr>
<tr>
<td>113.56, 114.69, 115.84, 116.99, 118.136,</td>
</tr>
<tr>
<td>119.28, 120.43, 121.57, 122.72, 123.86,</td>
</tr>
<tr>
<td>125.00, 126.14, 127.28, 128.42, 129.56,</td>
</tr>
<tr>
<td>130.73, 131.88, 132.98, 134.01, 135.22,</td>
</tr>
<tr>
<td>136.38, 137.92, 138.69, 139.79, 140.92,</td>
</tr>
<tr>
<td>142.05, 143.18, 144.31, 145.44, 146.57,</td>
</tr>
<tr>
<td>147.70, 148.83, 149.96, 151.09, 152.21,</td>
</tr>
<tr>
<td>153.34, 154.47, 155.59, 156.72, 157.84,</td>
</tr>
<tr>
<td>158.97, 160.09, 161.21, 162.33, 163.46,</td>
</tr>
<tr>
<td>164.58, 165.70, 166.82, 167.94, 169.06,</td>
</tr>
<tr>
<td>170.18, 171.30, 172.41, 173.53, 174.65,</td>
</tr>
<tr>
<td>175.77, 176.88, 178.00, 179.12, 180.23,</td>
</tr>
<tr>
<td>181.35, 182.46, 183.58, 184.69, 185.80 /</td>
</tr>
</tbody>
</table>

7 - 303
DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447, 2.365, 2.306, 2.262, 2.225, 2.201, 2.179, 2.160, 2.145, 2.131, 2.120, 2.110, 2.101, 2.101, 2.093, 2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /

LIST OF VARIABLES

ARG
INTERMEDIATE CALCULATION VARIABLE

CHI025() TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHI975() TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHITAB MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
I CONTROLS LOOP FOR CHI025() AND CHI975()
IOUT OUTPUT DUMP CONTROLLER
IZERO() 2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C FOR EACH REGION
JZERO() 2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY – MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
NPFR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP(I)-1))-1) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
NUM EQUAL TO NPFR(L) FOR A SET OF CALCULATIONS
NUMREG NUMBER OF REGIONS OF INTEREST
SUHAT EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
SX REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SX2() I-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
SWHAT EQUAL TO SWAT2(L)**0.5 FOR A SET OF CALCULATIONS
SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
T VALUE OF T025() USED IN CALCULATIONS
T025() TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
TTAB MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C INITIALIZE IZERO, JZERO AND MCHAT

DO 50 L = 1, MAXREG
  IZERO(1,L) = 0.0
  IZERO(2,L) = 0.0
  JZERO(1,L) = 0.0
  JZERO(2,L) = 0.0
  MCHAT(1,L) = 0.0
  MCHAT(2,L) = 0.0
50 CONTINUE

C CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

DO 75 L = 1, NUMREG
  IF (NPFR(L) .LT. 31) THEN
    7 - 304
  ENDIF
75 CONTINUE

C ASSIGN VALUES TO NUM, T, SWAT, SUHAT AND THEN CALCULATE
C CONFIDENCE INTERVALS FOR EACH REGION

DO 100 L = 1, NUMREG
  NUM = NPFR(L)
  IF (NUM .LT. 31) THEN

7 - 304
T = T025(NUM)
ELSE
T = T025(NUM)
ENDIF

SWHAT = SWHAT2(L) ** 0.5
SUHAT = SUHAT2(L) ** 0.5
SX = (NUM * SX2(L)) ** 0.5

C CALCULATE ESTIMATED VALUES OF M AND C
ARG = T * SWHAT / SX
MCHAT(1,L) = - DD(L)
MCHAT(2,L) = SUHAT

C CALCULATE CONFIDENCE INTERVALS
IZERO(1,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI975(NUM)) ** 0.5
IZERO(2,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI025(NUM)) ** 0.5
JZERO(I,L) = MCHAT(I,L) - ARG
JZERO(2,L) = MCHAT(I,L) + ARG

IF (IOUT .EQ. i0) THEN
WRITE(8,* 'L = ', L, ' NPPR =', NPPR(L), ' NUM =', NUM
WRITE(8,* 'SWHAT2 =', SWHAT2(L), ' SWHAT =', SWHAT
WRITE(8,* 'SUHAT2 =', SUHAT2(L), ' SUHAT =', SUHAT
WRITE(8,* 'SX2 =', SX2(L), ' SX =', SX
WRITE(8,* 'CHI025 =', CHI025(NUM), ' CHI975 =', CHI975(NUM)
WRITE(8,* 'T =', T, ' DD =', DD(L), ' ARG =', ARG
WRITE(8,* 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =', IZERO(2,L)
& WRITE(8,* 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =', JZERO(2,L)
& WRITE(8,* 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =', MCHAT(2,L)
ENDIF
100 CONTINUE
RETURN
END

******************************************************************************

C SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C THE CO GIVEN BY THE USER
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 80CT87 COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
C INPUTS: NUMREG, CZERO, SX2, SXY, SY2
C OUTPUTS: MCPNT, MC
C
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, MCPNT(MAXREG), NUMREG
REAL ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
& SXY(MAXREG), SY2(MAXREG)
LIST OF VARIABLES

INTERMEDIATE CALCULATION VARIABLE
INTERMEDIATE CALCULATION VARIABLE
EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
COEFFICIENT OF VARIATION, CO
EQUAL TO CZERO ** 2
OUTPUT DUMP CONTROLLER
CONTROLS DO LOOP FOR EACH REGION
MAXIMUM NUMBER OF REGIONS ALLOWED
2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA -- MC(I,L) IS
THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
THE NUMBER OF POINTS, 0, 1, OR 2, IN
MC() FOR EACH REGION
1-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA -- MC(I,L) IS
THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
1-D ARRAY CONTAINING THE NUMBER
1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
X = Ln S
1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR
EACH REGION
Y = Ln N
1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR
EACH REGION
X = Ln S, Y = Ln N
1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR
EACH REGION
Y = Ln N

INITIALIZE VARIABLES

DO 50 L = 1, MAXREG
MCPNT(L) = 0
MC(1,L) = 0.0
MC(2,L) = 0.0
50 CONTINUE

BEGIN CALCULATIONS

CZERO2 = CZERO ** 2

IF (IOUT .EQ. 10)
& WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

DO 100 L = 1, NUMREG
ARG1 = SX2(L) - CZERO2
ARG2 = 0.0

IF (CZERO .EQ. 0.0) THEN

THEN NO M CONSTRAINT IS REQUIRED
MCPNT(L) = 0
ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M
MCPNT(L) = 1
MC(1,L) = - SY2(L) / (2.0 * SXY(L))
ELSE

THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
COMMON CALCULATIONS
ARG2 = (SXY(L)** 2 - SY2(L) * ARG1)

IF (ARG2 .LT. 0.0) THEN
ARG2 IS NEGATIVE -- IMPLIES M IS COMPLEX
WRITE(8,*) 'ERROR: CO TOO LOW'
CALL TRNAT
ELSE
ARG2 = ARG2 ** 0.5
ENDIF

IF (SX2(L) .LT. CZERO2) THEN
AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

\[
\text{MCPNT}(L) = 1 \\
\text{MC}(1,L) = (- \text{SXY}(L) - \text{ARG2}) / \text{ARG1}
\]

ELSE

\[
\text{SXZ}(L) .GT. \text{CZERO2} - THIS TIME THE M CONSTRAINT IS A RANGE
\]

\[
\text{MCPNT}(L) = 2 \\
\text{MC}(1,L) = (- \text{SXY}(L) - \text{ARG2}) / \text{ARG1} \\
\text{MC}(2,L) = (- \text{SXY}(L) + \text{ARG2}) / \text{ARG1}
\]

ENDIF

ENDIF

100 CONTINUE

IF (IOUT .EQ. 10) THEN

DO 200 L = 1, NUMREG

WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

200 CONTINUE

ENDIF

RETURN

END

C SUBROUTINE GTPVAR CALCULATES THE EXTENT OF DEPARTURE FROM THE MULTIPLE
C HEAT MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88 COMMENTS: 13JUL89
C SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
C INPUTS: NSETS, NP, NUMREG, LAMN, MCHAT
C OUTPUTS: PVAR
C IMPLICIT NONE
C INTEGER MAXREG, MAXSET
C PARAMETER (MAXREG = 3, MAXSET = 5)
C COMMON IOUT
C INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG),
C & NUMREG, TOTAL
C REAL LAMN, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
L CONTROLS DO LOOP FOR EACH REGION
LAMN LAMBDA-N - RATIO OF Var (ln N given S) / (m**2 C**2),
**SUBROUTINE** FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH M AND C WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL) TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION.

**PROGRAMMER:** L. NEWLIN

**DATE:** CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91


**SUBROUTINE** FNDRNG (NUMREG, NPNT, MZERO, MCPNT, MC, JZERO, & MCHAT, RANGEM)

---

### CONSTANT OVER REGIONS AND COMPONENTS

- MAXREG: Maximum number of regions allowed
- MAXSET: Maximum number of S/N data sets allowed
- MCHAT(): 2-D array containing values of the estimates of M and C for each region, based on materials data only — MCHAT(1,L) = -DD(L), the estimate for M and MCHAT(2,L) = SUHAT, the estimate for C
- NP(): 2-D array containing number of points of each S/N data set in each region
- NSETS: Number of related material S/N data sets
- NUM(): Equal to Nj-1 for each region where Nj is the sum of the number of points in each data set
- NUMREG: Number of regions of interest
- PSIG2(): 1-D array containing estimates of the materials process variation in each region
- PVAR: The extent of departure from the multiple heat median S/N curve warranted by the available information
- SUM: Weighted sum of the PSIG2s — used to calculate a weighted average
- TOTAL: Sum of NUM() over all regions

### C INITIALIZE VARIABLES

```c
SUM = 0.0
TOTAL = 0.0
DO 50 L = 1, MAXREG
    PSIG2(L) = 0.0
    NUM(L) = 0
50 CONTINUE
DO 100 L = 1, NUMREG
    DO 150 J = 0, NSETS
        NUM(L) = NUM(L) + NP(J,L)
150 CONTINUE
    NUM(L) = NUM(L) - 1
    TOTAL = TOTAL + NUM(L)
100 CONTINUE
DO 200 L = 1, NUMREG
    PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L)**2
    SUM = SUM + PSIG2(L) * NUM(L)
200 CONTINUE
IF (IOUT.EQ.10) THEN
    WRITE(8,*) 'LAMN = ', LAMN
    DO 300 L = 1, NUMREG
        WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
300 CONTINUE
    WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
END IF
PVAR = SUM / FLOAT(TOTAL)
RETURN
END
```
C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT

C IMPLICIT NONE

INTEGER MAXREG
PARAMETER (MAXREG = 3)

COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), & MZERO(2, MAXREG); RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES

OUTPUT DUMP CONTROLLER
JZERO() 2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
MAXREG LOWER BOUND OF INTERSECTION
MC() MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION — MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MCPNT() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
C UPPER BOUND OF INTERSECTION

C INITIALIZE VARIABLES

DO 50 L = 1, MAXREG
  RANGEM(1,L) = 0.0
  RANGEM(2,L) = 0.0
50 CONTINUE

C PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

DO 100 L = 1, NUMREG
  IF (IOUT .EQ. 10) THEN
    WRITE(*,*) 'L = ', L, ' NUMREG = ', NUMREG
    WRITE(*,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
  ENDIF

  IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN
    THERE IS NO EXOGENOUS INFORMATION
    ASSUME RANGE TO BE Jo
    RANGEM(1,L) = JZERO(1,L)
    RANGEM(2,L) = JZERO(2,L)
  ELSE
    IF (IOUT .EQ. 10) THEN
      WRITE(*,*) 'RANGEM(1,L) = ', RANGEM(1,L), &
      ' JZERO(1,L) = ', JZERO(1,L)
WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,L), &
   'JZERO(2,L) = ', JZERO(2,L)
ENDIF

ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

   NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE TO CO, ADJUST THE LOWER BOUND OF JO ACCORDINGLY

   LOWER = AMAX1(JZERO(1,L), MC(1,L))
   UPPER = JZERO(2,L)
   IF (UPPER .LT. LOWER) THEN
      WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO AND MC'
      CALL TRMNAT
   ELSE
      RANGEM(1,L) = LOWER
      RANGEM(2,L) = UPPER
   ENDIF

ENDIF

ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

   THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE CORRESPONDING TO THE CO CONSTRAINT, ADJUST JO ACCORDINGLY

   LOWER = AMAX1(JZERO(1,L), MC(1,L))
   UPPER = AMIN1(JZERO(2,L), MC(2,L))
   IF (UPPER .LT. LOWER) THEN
      WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO AND MC'
      CALL TRMNAT
   ELSE
      RANGEM(1,L) = LOWER
      RANGEM(2,L) = UPPER
   ENDIF

ENDIF

ELSEIF (MPNT(L) .EQ. 1) THEN

   THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

   RANGEM(1,L) = MZERO(1,L)
   RANGEM(2,L) = 0.0

   IF (IOUT .EQ. 10) THEN
      WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L)
      WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L), ' RANGEM(2,L) = ', RANGEM(2,L)
   ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN

   THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT
   USE INTERSECTION BETWEEN JO AND MC

   LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
   UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
   IF (UPPER .LT. LOWER) THEN
      WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO AND MC'
   ENDIF

ENDIF

7-310
CALL TRMNAT
ELSE
  RANGEM(1,L) = LOWER
  RANGEM(2,L) = UPPER
ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  WRITE(8,*), 'JZERO(2,L) = ', JZERO(2,L),
  WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
  WRITE(8,*), 'MZERO(2,L) = ', MZERO(2,L),
  WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN
  THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO
  CONSTRAINT, INTERSECT JO AND MO, ADJUSTING THE LOWER BOUND
  BY MC ACCORDINGLY
  LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
  UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
  IF (UPPER .LT. LOWER) THEN
    WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO, MO, ',
    'AND MC'
    CALL TRMNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 2)) THEN
  THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO CO CONSTRAINT
  INTERSECT THESE TWO RANGES WITH JO
  LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
  UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
  IF (UPPER .LT. LOWER) THEN
    WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN JO, MO, ',
    'AND MC'
    CALL TRMNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF

ELSE
  WRITE(8,*), 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
  CALL TRMNAT
C ENDIF

C RESTRICT RANGE TO BE NON-NEGATIVE
RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)
IF (IOUT .EQ. 10) WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L)
100 CONTINUE

C CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE
DO 300 L = 1, NUMREG
  IF ((MCHAT(1,L) .LT. RANGEM(1,L))
    & .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
    & WRITE(8,*), 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
    & 'ON m IN REGION ', L
300 CONTINUE

RETURN
END

C************************************************************************************************

C SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C PROGRAMMER: L. NEWLIN
C DATE : CODE : 2FEB88 FORMAT/COMMENTS : 12AUG91
C VERSION: MATCHR
                      V6.!, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
                      V8.4, V8.5
SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C INPUTS: RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C OUTPUTS: RANGEM, MCHAT, NUMREG
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG
REAL MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)

LIST OF VARIABLES
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
IS UPPER BOUND

NUMBER OF NO DATA regions (REGIONS WITHOUT ANY S/N DATA)

NUMBER OF REGIONS OF INTEREST

2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M

FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND

RANGEM(2,L) IS THE UPPER BOUND

IF (IOUT .EQ. 10) WRITE(8,*), 'NUMREG =', NUMREG

DO 100 L = 1, NNODAT
    NUMREG = NUMREG + 1
    LL = NUMREG
    IF (IOUT .EQ. 10) WRITE(8,*), 'L = ', L, ' NUMREG = ', NUMREG,

    2, ' LL = ', LL, ' MPNT(LL) = ', MPNT(LL)
    IF ((MPNT(LL) .EQ. 1) OR (MPNT(LL) .EQ. 2)) THEN
        POSTERIOR ON M IS SAME AS PRIOR ON M
        RANGEM(1,LL) = MZERO(1,LL)
        RANGEM(2,LL) = MZERO(2,LL)
        IF (IOUT .EQ. 10) THEN
            WRITE(8,*), 'RANGEM(1,LL) = ', RANGEM(1,LL),

            2, ' MZERO(1,LL) = ', MZERO(1,LL)
            WRITE(8,*), 'RANGEM(2,LL) = ', RANGEM(2,LL),

            2, ' MZERO(2,LL) = ', MZERO(2,LL)
        ENDIF
    C SPECIFY E(M) OF POSTERIOR FOR SAKE OF
    C CALCULATIONS IN SUBROUTINE EXPCTD
    IF (RANGEM(2,LL) .EQ. 0.0) THEN
        MCHAT(1,LL) = RANGEM(1,LL)
    ELSE
        MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
    ENDIF
    ELSE IF (IOUT .EQ. 10) WRITE(8,*), 'MCHAT = ', MCHAT(1,LL)
    C CALL TRMNAT
    ENDIF
100 CONTINUE

RETURN
END

C SUBROUTINE CONCAV ADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY
C RANGES ON M TO BE CONSISTENT WITH CONCAVITY CONSTRAINTS
C PROGRAMMER: L. NEWLIN
C DATE: 2FEB88 FORMAT/COMMENTS: 1SSEP89
C VERSION: MATCHR V5.1, V5.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C V8.4, V8.5
C
SUBROUTINE CONCAV (NUMREG, RANGEM)
C INPUTS: NUMREG, RANGEM
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL RANGEM(2, MAXREG), TESTM

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
MAXREG CONTROLS DO LOOP FOR EACH REGION
NUMREG MAXIMUM NUMBER OF REGIONS ALLOWED
RANGEM(2) NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
TESTM UPPER BOUND OF RANGE ON M IN REGION L-1 – USED DURING

C ADJUST RANGE TO INSURE CONCAVITY

DO 100 L = NUMREG, 2, -1
    IF (RANGEM(2,L-1) .EQ. 0.0) THEN
        RANGE IS A POINT IN REGION L-1
        IF (RANGEM(1,L-1) .GT. AMAXI(RANGEM(1,L), RANGEM(2,L))) THEN
            WRITE(8,*)'ERROR: POSTERIOR INTERVAL IN REGION ',L
            CALL TRMNAT
        ENDIF
    ELSE
        RANGE IS AN INTERVAL IN REGION L-1
        TESTM = AMAXI(RANGEM(1,L), RANGEM(2,L))
        IF (TESTM .LT. RANGEM(1,L-1)) THEN
            WRITE(8,*)'ERROR: POSTERIOR INTERVAL IN REGION ',L
            CALL TRMNAT
        ELSE
            RANGEM(2,L-1) = AMINI(RANGEM(2,L-1), TESTM)
        ENDIF
    ENDIF
    IF (IOUT .EQ. I0) THEN
        WRITE(8,*),RANGEM(I,L-1), RANGEM(2,L-1), TESTM, L
    ENDIF
100 CONTINUE

RETURN
END

C*****************************************************************************

C SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER IO HAS
C BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR CO
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 5OCT87 COMMENTS: 1DEC87
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C V8.4, V8.5

SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)
C INPUTS: NUMREG, RANGEM
C OUTPUT: MEDM
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)

LIST OF VARIABLES
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LOWER M LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION) TO BE USED IN MEDIAN CALCULATION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

C INITIALIZE ARRAY MEDM
DO 50 L = 1, MAXREG
MEDM(L) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS FOR EACH REGION
DO 100 L = 1, NUMREG
IF (RANGEM(2,L) .EQ. 0.0) THEN
C RANGE IS A POINT
MEDM(L) = RANGEM(1,L)
ELSEIF (L .EQ. 1) THEN
C WE ARE IN REGION ONE -- NOT AFFECTED BY OTHER REGIONS -- MEDIAN WILL JUST BE AVERAGE OF RANGEM VALUES
MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
ELSE
C MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT
LOWERM = AMAX1(RANGEM(1,L), MEDM(L-1))
MEDM(L) = (LOWERM + RANGEM(2,L)) / 2.0
ENDIF
IF (IOUT .EQ. 10) THEN
WRITE(*,*),'L = ', L, ' NUMREG = ', NUMREG
WRITE(*,*),'RANGEM(1,L) = ', RANGEM(1,L),
& 'RANGEM(2,L) = ', RANGEM(2,L),
& 'LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
ENDIF
100 CONTINUE
RETURN
END
SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N CURVE PARAMETERS

PROGRAMMER: L. NEWLIN

DATE: CODE: 13FEB89

FORMAT/COMMENTS: 15SEP89


Copyright (C) 1990, California Institute of Technology. U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, & ZROREG, NBND, BIGK, BZHAT)

INPUTS: NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND

OUTPUTS: BIGK, BZHAT

SUBPROGRAMS: TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

IMPLICIT NONE

INTEGER MAXDAT, MAXREG

PARAMETER (MAXDAT = 50, MAXREG = 3)

COMMON IOUT

INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG), STR(MAXDAT, MAXREG), SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)

LIST OF VARIABLES

BIGK(0:MAXREG), BIGK, BZHAT, FACTR, KHAT, MEANZ, & MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG), & NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG), & SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)

1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION

BIGK1 EQUAL TO BIGK(1)

BZHAT E(BETA)

FACTR A SCALE FACTOR = PHI * KRATIO * Z

IOUT OUTPUT DUMP CONTROLLER

KHAT E(k)

L CONTROLS DO LOOP FOR EACH REGION

MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED

MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED

MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION

MM() 1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST

NCOMPS NUMBER OF COMPONENTS -- 1 FOR STRESS AND STRAIN WHEN DECOMPOSED DATA UNAVAILABLE -- 2 FOR DECOMPOSED STRAIN DATA

NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS

NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET

NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR THE SPECIFIC MATERIAL S/N DATA SET

NUMREG NUMBER OF REGIONS OF INTEREST

SBND() 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()

STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

SZ2 SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

SZERO STRESS TENSILE TEST POINT, SO

TRBIGK() 1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE TRBIGK(i) = BIGK(i)

ZROREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE = 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

ZZ() 1-D ARRAY CONTAINING TRANSFORMED S-N DATA, Z = F(STR, NF, NBND, MM)
C INITIALIZE VARIABLES
   DO 50 L = 0, MAXREG
       MM(L) = 0.0
   50 CONTINUE
C CREATE MM() ARRAY FROM MEDM() ARRAY
   DO 100 L = 1, NUMREG
       MM(L) = MEDM(L)
   100 CONTINUE
C TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)
   CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)
C CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)
   CALL SMNVAR (NP, ZZ, MEANZ, S22)
C CALCULATE BETA0 AND k
   CALL KBETA (MEANZ, S22, KHAT, BZHAT)
C CALCULATE THE VALUES OF K, WHERE A = K**M FOR EACH REGION
   CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)
   BIGKI = BIGK(1)
C CALCULATE BOUNDARIES OF STRESS REGIONS
   CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C CALCULATE Ko AND Mo FOR THE NO DATA REGION TO THE LEFT IF REQUIRED
   DO 150 L = ZROREG, NUMREG
       TRBIGK(L) = BIGK(L)
   150 CONTINUE
   IF (ZROREG .EQ. 0) THEN
       FACTR = 1.0
       CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK, FACTR, NUMREG)
   ENDIF
C WRITE RESULTS TO FILE
   IF (NCOMPS .EQ. 0) THEN
       WRITE(7,900) NUMREG, BZHAT, KHAT
       WRITE(7,900) NUMREG, BZHAT, KHAT
   ELSE
       WRITE(7,920)
   ENDIF
   WRITE(7,930) MM(1), BIGK(1), KHAT
C FORMAT STATEMENTS
900 FORMAT (//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',//,2X,
   & 'NUMBER OF REGIONS:',I4,5X,'E(BETA0) =',F8.4,5X,'E(k) =',
   & 'F8.4',//,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,
   & 'STRESS BOUND',/)

C SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS: MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88 COMMENTS: 13JUL89

SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, &
& MO, SIGMA2, MCHAT, MU, SIG)

INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2

OUTPUTS: MCHAT, MU, SIG

IMPLICIT NONE

INTEGER MAXREG

PARAMETER ( MAXREG = 3 )

COMMON IOUT

INTEGER IOUT, L, NUMREG, NPPR(MAXREG)

REAL ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG), &
& MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG), &
& SUHAT2(MAXREG), SUMX2, SWHAT2(MAXREG), SX2(MAXREG)

C

C LIST OF VARIABLES
C
C ARG
C INTERMEDIATE CALCULATION VARIABLE
C DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
C IOUT
C OUTPUT DUMP CONTROLLER
C L
C CONTROLS DO LOOP FOR EACH REGION
C MAXREG
C MAXIMUM NUMBER OF REGION ALLOWED
C MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) = DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
C MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
C NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
C NUMREG
C NUMBER OF REGIONS OF INTEREST
C SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
C SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C SUMX2 = EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
C SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND CO TO
OBTAIN POSTERIOR RANGES ON M FOR EACH REGION

SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
OUTPUTS: RANGEM
SUBPROGRAMS: TRMNAT

C SX2()  1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C (X = Ln S)
C
C INITIALIZE ARRAYS
DO 50 L = 1, MAXREG
MCHAT(1,L) = 0.0
MCHAT(2,L) = 0.0
MU(L) = 0.0
SIG(L) = 0.0
50 CONTINUE

C BEGIN CALCULATION FOR EACH REGION
DO 100 L = 1, NUMREG

MCHAT(1,L) = DD(L)
MCHAT(2,L) = SQRT (SUHAT2(L))
SUMX2 = NPPR(L) * SX2(L)
ARG = SUMX2 + DELTA(L)

IF (DELTA(L) .EQ. 0.0) THEN
  THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
  USE THE ESTIMATE OF M
  MU(L) = MCHAT(1,L)
ELSE
  UPDATE THE ESTIMATE OF M WITH MO USING DELTA
  MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
ENDIF

IF (SIGMA2(L) .EQ. 0.0) THEN
  THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
  USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
  SIG(L) = SQRT (SWHAT2(L) / ARG)
ELSE
  SIG(L) = SQRT (SIGMA2(L) / ARG)
ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ', MCHAT(1,L)
  & WRITE(8,*) 'SUHAT2 = ', SUHAT2(L), ' MCHAT2 = ', MCHAT(2,L)
  & WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
  & SUMX2 = ', SUMX2
  & WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
  & WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
  & WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
  & SIG = ', SIG(L)
ENDIF
100 CONTINUE

RETURN
END
IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
 & RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LOWER LOWER BOUND OF INTERSECTION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
 REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
 - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
 BOUND
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
 FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
 FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
 MC() FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
 MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
 EACH REGION - MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
 IS THE UPPER BOUND
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
 FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
 RANGEM(2,L) IS THE UPPER BOUND
UPPER UPPER BOUND OF INTERSECTION

INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
 RANGEM(1,L) = 0.0
 RANGEM(2,L) = 0.0
50 CONTINUE

PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
DO 100 L = 1, NUMREG
 IF (IOUT .EQ. 10) THEN
 WRITE(8,*),'L = ', L, ' NUMREG = ', NUMREG
 WRITE(8,*),'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
 ENDIF
 IF (MPNT(L) .EQ. 1) THEN
 THERE IS A POINT PRIOR ON M - THIS OVERRIDES ALL OTHER
 INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
 RANGEM(1,L) = MZERO(1,L)
 RANGEM(2,L) = 0.0
 ELSEIF (MFNL(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0) THEN
 THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT USE M0
 IF (IOUT .EQ. 10) THEN
 WRITE(8,*),'MZERO(1,L) = ', MZERO(1,L)
 WRITE(8,*),'RANGEM(1,L) = ', RANGEM(1,L),
 & ' RANGEM(2,L) = ', RANGEM(2,L)
 ENDIF
 ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN
 THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT USE M0

7 - 320
RANGEM(1,L) = MZERO(1,L)
RANGEM(2,L) = MZERO(2,L)

IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
&
WRITE(8,*), 'MZERO(2,L) = ', MZERO(2,L),
&
WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
&
RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPT(L) .EQ. 2)) THEN
C
THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO CO
CONSTRAINT INTERSECT THESE TWO RANGES
LOWER = AMAX1(MZERO(1,L), MC(1,L))
UPPER = AMIN1(MZERO(2,L), MC(2,L))
IF (UPPER .LT. LOWER) THEN
WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN MO AND MC'
CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPT(L) .EQ. 1)) THEN
C
THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO
CONSTRAINT ADJUST THE LOWER BOUND OF MO BY MC
LOWER = AMAX1(MZERO(1,L), MC(1,L))
UPPER = MZERO(2,L)
IF (UPPER .LT. LOWER) THEN
WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN MO AND MC'
CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

ELSEIF ((MPNT(L) .EQ. 10) THEN
WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
&
WRITE(8,*), 'MZERO(2,L) = ', MZERO(2,L),
&
WRITE(8,*), 'MC(1,L) = ', MC(1,L),
&
WRITE(8,*), 'MC(2,L) = ', MC(2,L),
&
WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
&
RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

ELSE
WRITE(8,*), 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
CALL TRMNAT
ENDIF

C
RESTRICT RANGE TO BE NON-NEGATIVE
RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)
IF (IOUT .EQ. 10) WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L)
100 CONTINUE
CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

DO 300 L = 1, NUMREG

   IF ( (MCHAT(1,L) .LT. RANGEM(1,L))
     & .OR. (MCHAT(1,L) .GT. RANGEM(2,L)) )
     & WRITE(*,*) 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
     & 'ON m IN REGION ', L

300 CONTINUE

RETURN

END

C SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 10FEB88

C SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
    & MZERO, MPNT, MO, SIGMA2)
C INPUTS: RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C OUTPUTS: RANGEM, MCHAT, MU, SIG, NUMREG
C IMPLICIT NONE

INTEGER MAXREG
PARAMETER (MAXREG=3)

COMMON IOUT
INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

REAL MCHAT(2, MAXREG), MO(MAXREG), MU(MAXREG),
    & MZERO(2, MAXREG), RANGEM(2, MAXREG), SIG(MAXREG),
    & SIGMA2(MAXREG)

LIST OF VARIABLES

IOUT  OUTPUT DUMP CONTROLLER
L     CONTROLS DO LOOP FOR EACH REGION
LL    EQUAL TO NUMREG FOR A SET OF CALCULATIONS
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
         C FOR EACH REGION, BASED ON MATERIALS DATA ONLY —
         MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND
         MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MO()   1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
         MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
        EACH REGION
MU()   1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
        DISTRIBUTION MEAN FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
        EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
        IS UPPER BOUND
NNODAT Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
         FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND
1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

SIGMA2( ) 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
VARIANCE FOR EACH REGION

IF (IOUT .EQ. 10) WRITE(8,*),'NUMREG =', NUMREG
DO 100 L = 1, NNODAT
NUMREG = NUMREG + 1
LL = NUMREG
IF (IOUT .EQ. 10) WRITE(8,*),'L =', L, ' NUMREG =', NUMREG,
& ' LL =', LL, ' MPNT(LL) = ', MPNT(LL)
IF ((MPNT(LL) .EQ. 1) OR. (MPNT(LL) .EQ. 2)) THEN
POSTERIOR ON M IS SAME AS PRIOR ON M
RANGEM(1,LL) = MZERO(1,LL)
RANGEM(2,LL) = MZERO(2,LL)
MU(LL) = MO(LL)
SIG(LL) = SQRT(SIGMA2(LL))
IF (IOUT .EQ. 10) THEN
WRITE(8,*),'RANGEM(1,LL) = ', RANGEM(1,LL),
& ' MZERO(1,LL) = ', MZERO(1,LL)
WRITE(8,*),'RANGEM(2,LL) = ', RANGEM(2,LL),
& ' MZERO(2,LL) = ', MZERO(2,LL)
WRITE(8,*),'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
WRITE(8,*),'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
& SIGMA2(LL)
ENDIF

SPECIFY E(M) OF POSTERIOR FOR SAKE OF
CALCULATIONS IN SUBROUTINE EXPTMD

IF (RANGEM(2,LL) .EQ. 0.0) THEN
MCHAT(1,LL) = RANGEM(1,LL)
MU(LL) = RANGEM(1,LL)
SIG(LL) = 0.0
ELSE
MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
ENDIF
IF (IOUT .EQ. 10) WRITE(8,*),'MCHAT =', MCHAT(1,LL),
& MU = ', MU(LL), ' SIG = ', SIG(LL)
ELSE
WRITE(8,*)'ERROR: OVERALL PRIOR RANGE INCORRECTLY ',
& 'SPECIFIED IN REGION WITHOUT DATA'
CALL TRMNAT
ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE PAREST CONTROLS THE CALCULATIONS FOR THE PARAMETER
ESTIMATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
PROGRAMMER: L. NEWLIN
DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89
VERSION: MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
MATGRM V4.3, V4.4, V4.5

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.
INPUTS: VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
        NBDN, STR
OUTPUTS: BIGK, BZERO, MM, SBND
SUBPROGRAMS: FINDM, FINDMN, TRANSFM, SMNVAR, KBETA, FINDK, FINDSB

IMPLICIT NONE

INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
    INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG
    REAL
        & BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG),
            & MU(MAXREG), NBDN(0:MAXREG), NF(MAXDAT, MAXREG),
            & RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
            & STR(MAXDAT, MAXREG), ZZ(MAXDAT)

DOUBLE PRECISION RAND

LIST OF VARIABLES

BIGK()  1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
        EACH REGION
BZERO   VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING S/N DATA SET
IOUT    OUTPUT DUMP CONTROLLER
K       VALUE OF K -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
L       CONTROLS DO LOOP FOR EACH REGION
MAXDAT  MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MEANZ   SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBDN, MM)
MM()    1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
MU()    1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
        DISTRIBUTION MEAN FOR EACH REGION
NBND()  1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
        REGIONS OF INTEREST
NF()    2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC
        MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP()    TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
NPTS()  1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
        SPECIFIC MATERIAL S/N DATA SET
NUMREG  NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
        FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
        RANGEM(2,L) IS THE UPPER BOUND
RAND    RANDOM NUMBER SEED
SBND()  1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
        CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
        REGION CONTAINED IN NBND()
SIG()   1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
        STANDARD DEVIATION FOR EACH REGION
STR()   2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
        DATA SET BROKEN INTO REGIONS (PSI OR %)
SZ2    SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBDN, MM)
VARY   CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
        1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION;
        3 - TRUNCATED NORMAL VARIATION
ZROREG ZeRO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
ZZ()    1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
        Z = F(STR,NF,NBDN,MM)

C OBTAIN THE VALUES OF M FOR EACH REGION

IF (VARY .LE. 2) THEN
    UNIFORM OR NO VARIATION IN M IS DESIRED
    CALL FINDM (RAND, NUMREG, RANGEM, MM)
ELSE

7 - 324
C NORMAL VARIATION IN M IS DESIRED
CALL FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
ENDIF
C TRANSFORM THE S/N DATA INTO THE VARIABLE \( Z = \ln(X) \)
CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)
C CALCULATE THE SAMPLE MEAN AND VARIANCE OF \( Z = \ln(X) \)
CALL SMNVAR (NP, ZZ, MEANZ, SZ2)
C CALCULATE THE VALUES FOR \( k \) AND BETAO FROM THE SAMPLE MEAN
C AND VARIANCE
CALL KBETA (MEANZ, SZ2, K, BZERO)
C CALCULATE THE VALUE OF \( k \) FOR EACH REGION WHERE \( A = k \times k \times M \)
CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)
C CALCULATE STRESS TIE-POINTS
CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C WRITE RESULTS TO FILE
C WRITE(7,900) NUMREG, BZERO
C DO 200 L = ZROREG, NUMREG
C WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
C 200 CONTINUE
C WRITE (7,920)
C FORMAT STATEMENTS
900 FORMAT(///,2X,'SELECTED VALUES OF S/N CURVE PARAMETERS', &
      //,2X,'NUMBER OF REGIONS:',I4,5X,'BETAo = ',F8.4,
      &
      //,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',5X,
      &
      'STRESS BOUND'/)
910 FORMAT(5X,I5,5X,F9.5,5X,E12.5,5X,E9.3,6X,E11.5)
920 FORMAT(///)
RETURN
END

C**********************************************************************

C SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION BY
C SAMPLING OFF THE APPROPRIATE M RANGE
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88 COMMENTS: 13JUL89
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)
C INPUTS: RAND, NUMREG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: RANDOM, TRMNAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
PICK() 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RANGEM(2) RANDOM NUMBER SEED
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
X UNIFORM (0, 1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED OFF THE RANGE ON M

C INITIALIZE MM()

DO 50 L = 0, MAXREG
  MM(MAXREG) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS

DO 100 L = 1, NUMREG
  PICK(1) = 0.0
  PICK(2) = 0.0
  IF (RANGEM(2,L) .EQ. 0.0) THEN
    M IS SPECIFIED AS A POINT VALUE
    MM(L) = RANGEM(1,L)
    IF (IOUT .EQ. 10) WRITE(8,*), 'RANGEM(1,L) =', RANGEM(1,L), 'MM(L) =', MM(L)
  ELSEIF (L .EQ. 1) THEN
    SAMPLE ON EXISTING RANGE
    CALL RANDOM(X, RAND)
    MM(L) = (RANGEM(2,L) - RANGEM(1,L)) * X + RANGEM(1,L)
    IF (IOUT .EQ. 10) THEN
      WRITE(8,*), 'RANGEM(1,L) =', RANGEM(1,L), 'RANGEM(2,L) =', RANGEM(2,L), 'L =', L, 'X =', X, 'MM(L) =', MM(L)
    ENDIF
  ELSE
    ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
    AND THEN SAMPLE
    PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
    PICK(2) = RANGEM(2,L)
    IF (PICK(1) .GT. PICK(2)) THEN
      NO RANGE EXISTS - THIS SHOULD NOT BE POSSIBLE
      STOP PROGRAM
      WRITE(8,*), 'IMPOSSIBLE M RANGE IN REGION', L
      CALL TRMNAT
    ELSE
      SAMPLE ON ADJUSTED RANGE
      CALL RANDOM(X, RAND)
      MM(L) = (PICK(2) - PICK(1)) * X + PICK(1)
      IF (IOUT .EQ. 10) THEN
        WRITE(8,*), 'L =', L, 'MM(L-1) =', MM(L-1), 'RANGEM(1,L) =', RANGEM(1,L), 'RANGEM(2,L) =', RANGEM(2,L), 'PICK(1) =', PICK(1), 'PICK(2) =', PICK(2), 'X =', X, 'MM(L) =', MM(L)
      ENDIF
    ENDIF
  ENDIF
100 CONTINUE
SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE UNIFORMLY DISTRIBUTED RANDOM NUMBERS


PROGRAMMER: L. GRONDALSKI, L. NEWLIN

DATE: IDEC87


LIST OF VARIABLES

FRAC UNIFORM (0,1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
RANA CONSTANT FOR LCG
RANC CONSTANT FOR LCG
RAND RANDOM NUMBER SEED
RANDIV INTERNAL CALCULATION
RANM CONSTANT FOR LCG
RANSUB INTERNAL CALCULATION
RANT INTERNAL CALCULATION
RANX INTERNAL CALCULATION

USING LCG RANDOM # GENERATOR

RANA = 671093.0
RANC = 7090885.0
RANM = 33554432.0

10 RANX = RANA * RAND + RANC
RANDIV = RANX / RANM
RANT = DINT(RANDIV)
RANSUB = RANT * RANM
RAND = RANX - RANSUB
FRAC = SNGL(RAND / RANM)

IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
IF (IOUT .EQ. 2) WRITE(8,*','RANX = ', RANX, ', RANDIV = ', RANDIV, ' RANT = ', RANT, ' RANSUB = ', RANSUB, ' RAND = ', RAND, ' FRAC = ', FRAC

RETURN
END
C NOTES: IOUT=2 DUMPS TO SCREEN

******************************************************************************

C SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88 COMMENTS: 13FEB89
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)

C INPUTS: RAND, NUMREG, MU, SIG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: NORMGN, TRMNAT

C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG),
& SIG(MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
MU() 1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
PICK() 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RAND RANDOM NUMBER SEED
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
& FOR EACH REGION -- RANGEM(I,L) IS THE LOWER BOUND AND
& RANGEM(2,L) IS THE UPPER BOUND
SIG() 1-D ARRAY CONTAINING THE STANDARD DEVIATION OF M FOR EACH REGION
X NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
OFF THE RANGE ON M

C INITIALIZE MM()

DO 50 L = 0, MAXREG
  MM(MAXREG) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS

DO 100 L = 1, NUMREG

  PICK(1) = 0.0
  PICK(2) = 0.0

  IF (RANGEM(2,L) .EQ. 0.0) THEN
    WRITE (8,*) 'RANGEM(I,L) =,, RANGEM(I,L),'
    ELSEIF (L .EQ. I) THEN ' MM(L) =', MM(L)
  END IF

  C M IS SPECIFIED AS A POINT VALUE
  MM(L) = RANGEM(1,L)
  IF (IOUT .EQ. 10) WRITE(8,*), 'RANGEM(1,L) =', RANGEM(1,L),
    ' MM(L) =', MM(L)
  ELSEIF (L .EQ. 1) THEN
C SAMPLE ON EXISTING RANGE
CALL NORMGN (RAND, MU(L), SIG(L), X)
IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
MM(L) = X
IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
  & ' RANGEM(2,L) =', RANGEM(2,L)
ENDIF
ELSE
  C ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
  AND THEN SAMPLE
  PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
  PICK(2) = RANGEM(2,L)
  IF (PICK(1) .GT. PICK(2)) THEN
    C NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
    STOP PROGRAM
    CALL TRMNAT
  ELSE
    C SAMPLE ON ADJUSTED RANGE
20 CALL NORMGN (RAND, MU(L), SIG(L), X)
    IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
    MM(L) = X
  ENDIF
ENDIF
IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'L =', L,
  & ' MM(L-1)=', MM(L-1),
  & ' RANGEM(1,L) =', RANGEM(1,L)
  WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
  WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
  & ' MM(L) =', MM(L)
ENDIF
ENDIF
100 CONTINUE
RETURN
END

C***************************************************************

C SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 3FEB88
C VERSION: MATCHR V1.0, V1.1, V1.2, V1.3, V1.4, V1.5
C MATGRM V1.0, V1.1, V1.2, V1.3, V1.4, V1.5
C The random variates are generated using the "Direct Method"
C Abramowitz, M., and Stegun, I. A., editors, Handbook of
C Mathematical Functions, National Bureau of Standards, Applied
C Mathematics Series 55, issued June 1964, Ninth Printing, November
C 1970 with corrections, pg. 953.
C***************************************************************

SUBROUTINE NORMGN (RAND, MU, SIGMA, X)
C SUBPROGRAM: RANDOM
C IMPLICIT NONE
COMMON IOUT
DOUBLE PRECISION RAND
REAL FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2
PARAMETER (PI = 3.1415926536)
INTEGER IOUT
LIST OF VARIABLES

FRAC  UNIFORM(0,1) RANDOM VARIATE
IOUT  OUTPUT DUMP CONTROLLER
MU    MEAN OF NORMAL DISTRIBUTION
RAND  RANDOM NUMBER SEED
SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION
X     NORMAL RANDOM VARIATE
U1    UNIFORM RANDOM NUMBER U(0,1)
U2    UNIFORM RANDOM NUMBER U(0,1)
Z1    NORMAL RANDOM NUMBER ON N(0,1)
Z2    NORMAL RANDOM NUMBER ON N(0,1)

IF (((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*), 'RAND =', RAND, ', MU =', MU, ', SIGMA =', SIGMA
CALL RANDOM (FRAC, RAND)
U1 = FRAC
CALL RANDOM (FRAC, RAND)
U2 = FRAC
IF (((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*), 'U1 =', U1, ', U2 =', U2
Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)
X = SIGMA * Z1 + MU
IF (((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*), 'Z1 =', Z1, ', Z2 =', Z2, ', X =', X
RETURN
END

C******************************************************************************
C
C SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM
C THE S/N DATA INTO THE VARIABLE Z = Ln(X)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88 COMMENTS: 13JUL89
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C INPUTS:  NPTS, STR, NF, NUMREG, MM, NBND
C OUTPUTS: NF, ZZ

C IMPLICIT NONE

INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
INTEGER I, IOUT, K, LL, NP, NPTS(MAXREG), NUMREG
REAL MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
&       STR(MAXDAT, MAXREG), ZZ(MAXDAT)

LIST OF VARIABLES

I  CONTROLS DO LOOP FOR EACH DATA POINT
IOUT OUTPUT DUMP CONTROLLER
K  CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
L  CONTROLS DO LOOP FOR EACH REGION
SUBROUTINE SMNVAR CALCULATES THE Sample Mean and Variance OF Z = F(STR, NF, NBND, MM)

SUBROUTINE SMNVAR (NP, ZZ, MEANZ, S2Z)

INPUTS: NP, ZZ

C*SUBROUTINE SMNVAR CALCULATES THE Sample Mean and Variance OF Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 24AUG87 COMMENTS: 13JUL89

C**initialize variables

NP = 0
DO 50 I = 1, MAXDAT
  ZZ(I) = 6.0
50 CONTINUE

C BEGIN CALCULATIONS

DO 100 L = 1, NUMREG
  MML = MM(L)
  IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ', MM =', MM(L), ', MML =', MML, ', NPTS =', NPTS(L)
  DO 200 K = 1, INPTS(L)
    NP = NP + 1
    ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
    IF (IOUT .EQ. 10) WRITE(8,*)'K =', K, ', NP =', NP, ', NF =', NF(K,L), ', STR =', STR(K,L), ', ZZ =', ZZ(NP)
 200 CONTINUE
100 CONTINUE

RETURN
END
OUT PUTS: MEANZ, SZ2

IMPLICIT NONE

INTEGER MAXDAT
PARAMETER (MAXDAT = 50)

COMMON IOUT
INTEGER I, IOUT, NP

REAL MEANZ, SZ2, ZZ(MAXDAT)

LIST OF VARIABLES

I CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT OUTPUT DUMP CONTROLLER
MAXDAT MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
SZ2 SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

C INITIALIZE VARIABLES

MEANZ = 0.0
SZ2 = 0.0

C CALCULATE THE MEAN OF ZZ(), MEANZ

DO 100 I = 1, NP
      MEANZ = MEANZ + ZZ(I)
      IF (IOUT .EQ. I0) WRITE(8,*)'NP =', NP, ', I = ', I,
       \ MEANZ = ', MEANZ
100 CONTINUE

C CALCULATE THE VARIANCE OF ZZ(), SZ2

DO 200 I = 1, NP
      SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
      IF (IOUT .EQ. I0) WRITE(8,*)'I = ', I, ', SZ2 = ', SZ2
200 CONTINUE

RETURN
END

SUBROUTINE KBETA CALCULATES K AND BETAO FROM THE SAMPLE MEAN AND
VARIANCE OF Z = F(STR, NF, NBND, MH)

SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)

INPUTS: MEANZ, SZ2

OUTPUTS: K, BZERO

7 - 332
C IMPLICIT NONE
REAL PI
PARAMETER (PI = 3.1415926536)
COMMON IOUT
INTEGER IOUT
REAL BZERO, K, MEANZ, SZ, SZ2

C LIST OF VARIABLES
C BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE
C SPECIFIC MATERIAL S/N DATA SET
C IOUT OUTPUT DUMP CONTROLLER
C K VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL
C DATA BASE
C MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C PI SELF EXPLANATORY CONSTANT
C SZ SZ2 ** 0.5
C SZ2 SAMPLE VARIANCE OF THE TRANSFORMED DATA,
C Z = F(STR, NF, NBND, MM)

C PERFORM CALCULATIONS

SZ = SZ2 ** 0.5
BZERO = PI / (SZ * (6.0 ** 0.5))
K = MEANZ

C DATA DUMP STATEMENTS

IF (IOUT .EQ. 10) THEN
WRITE(8,*) 'SZ2 =', SZ2, ' SZ =', SZ
WRITE(8,*) 'MEANZ =', MEANZ, ' K = ', K, ' BZERO =', BZERO
ENDIF
RETURN
END

C SUBROUTINE FINDK CALCULATES THE VALUE OF K, WHERE A = K ** M FOR
C EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: 7JUN88
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C INPUTS: BZERO, K, MM, NBND, NUMREG
C OUTPUTS: BIGK

C IMPLICIT NONE
INTEGER MAXREG
REAL GAMMA
PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL BIGK(0:MAXREG), BZERO, K, MM(0:MAXREG), NBND(0:MAXREG)

LIST OF VARIABLES

BIGK( )  1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
FOR EACH REGION
BZERO  VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING SPECIFIC
MATERIAL DATABASE
GAMMA  EULER'S CONSTANT
IOUT  OUTPUT DUMP CONTROLLER
K  VALUE OF K -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
    DATABASE
L  CONTROLS DO LOOP FOR EACH REGION
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MM( )  1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND( )  1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
    REGIONS OF INTEREST
NUMREG  NUMBER OF REGIONS OF INTEREST

C INITIALIZE VARIABLES
    DO 50 L = 0, MAXREG
        BIGK(L) = 0.0
      50 CONTINUE
C CALCULATE K FOR REGION ONE
    BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
    WRITE(7,*) 'REGION: K =', BIGK(1)
    IF (IOUT .EQ. 10) WRITE(8,*)'BZERO =', BZERO, ' k =', K,
        & ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)
C CALCULATE K FOR REMAINING REGIONS
    DO 100 L = 2, NUMREG
        BIGK(L) = BIGK(L-1) * NBND(L-1)
        & ** ((1.0 / MM(L)) - (1.0 / MM(L-1)))
        WRITE(7,*) 'REGION: K =', BIGK(L)
        IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, NBND(L-1) =',
            & ' MM(L) =', MM(L), ' MM(L-1) =', MM(L-1),
            & ' BIGK(L) =', BIGK(L)
      100 CONTINUE
    RETURN
END

C******************************************************************************

C SUBROUTINE FINDSB CALCULATES THE REGION 'TIE-POINTS' -- THE STRESS
VALUES WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE
RANDOMLY SELECTED MS, AND THE KS CALCULATED FROM THE BETA AND K
CHARACTERIZING SPECIFIC MATERIAL
PROGRAMMER:  L. NEWLIN
DATE:  22DEC88
VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C INPUTS:  NUMREG, ZROREG, NBND, BIGK, MM
C OUTPUTS:  SBND
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)

COMMON IOUT

INTEGER IOUT, L, NUMREG, ZROREG

REAL BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG), &
     SBND(0:MAXREG)

LIST OF VARIABLES

BIGK()  1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
        FOR EACH REGION
IOUT   OUTPUT DUMP CONTROLLER
L   CONTROLS DO LOOP FOR EACH REGION
MM()  1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG MAXIMUM NUMBER OF REGIONS ALLOWED
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
        REGIONS OF INTEREST
SBND() 1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
        CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
        REGION CONTAINED IN NBND()
ZROREG ZERO REGION — VALUES CHOSEN TO FACILITATE REGION DO LOOP
        BEGINNING VALUE — 0 — ZERO REGION EXISTS, 1 — NO REGION

C INITIALIZE SBND()

   DO 50 L = 0, MAXREG
      SBND(L) = 0.0
   50 CONTINUE

C CALCULATE SBND(0) IF ZROREG = 0

   IF (ZROREG .EQ. 0) THEN
      SBND(0) = BIGK(1) * NBND(0) ** (-1.0 / MM(1))
   ENDIF

C CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES

   DO 100 L = 1, NUMREG
      IF (NBND(L) .GE. 1.0E+36) THEN
         SBND(L) = 0.0
      ELSE
         SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
      ENDIF
   100 CONTINUE

RETURN

END

C*****************************************************************************

C THIS SUBROUTINE GENERATES WEIBULL(ETA,ETA) RANDOM VARIATES WITH
C MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C TRANSFORM METHOD"
C PROGRAMMER:  L. NEWLIN
C DATE:  CODE:  18MAR87  COMMENTS:  15SEP89
C VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C           V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C           MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C           V4.3, V4.4, V4.5
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
SUBROUTINE WEIBGN (BETA, RAND, WEIB)

C INPUTS: BETA, RAND
C OUTPUTS: WEIB
C SUBPROGRAMS: RANDOM

C IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL ARG, BETA, ETA, FRAC, WEIB
DOUBLE PRECISION RAND

C LIST OF VARIABLES
INTERMEDIATE CALCULATION VARIABLE
WEIBULL DISTRIBUTION SHAPE PARAMETER
WEIBULL DISTRIBUTION LOCATION PARAMETER
UNIFORM (0,1) RANDOM VARIATE
OUTPUT DUMP CONTROLLER
RANDOM NUMBER SEED
WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE

C CALCULATE CONSTRAINED ETA
ETA = 1.0 / (ALOG(2.0)**(1.0 / BETA))

C GENERATE WEIBULL RANDOM VARIATE
CALL RANDOM(FRAC, RAND)
ARG = -ALOG(1.0 - FRAC)
WEIB = ETA * ARG**(1.0 / BETA)
IF (IOUT .EQ. 10) WRITE(8,*)'BETA = ', BETA, ' FRAC =', FRAC, ' ARG =', ARG, ' ETA =', ETA, ' WEIB =', WEIB
RETURN
END

C SUBROUTINE KOMO CALCULATES KO AND MO FOR THE ZERO REGION (NO DATA
REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
AT ZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
PROGRAMMER: L. NEWLIN
DATE: IAUG91
VERSION: MATCHR V8.5 MATGRM V4.5

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK, &
FACTR, NUMREG)

C INPUTS: SZERO, BIGK, MM, NBND, TRSBND, TRBIGK, &
FACTR, NUMREG
C OUTPUTS: TRBIGK, MM, TRSBND
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT

7 - 336
INTEGER IOUT, L, NUMREG

REAL BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1 SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)

LIST OF VARIABLES

BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
       EACH REGION
 FACTR SCALE FACTOR = PHI * KRATIO * Z
 IOUT OUTPUT DUMP CONTROLLER
 L CONTROLS DO LOOP FOR EACH REGION
 MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
 MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
 NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
        REGIONS OF INTEREST
 NUMREG NUMBER OF REGIONS
 SCLK ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
 SZERO STRESS TENSILE TEST POINT, SO
 TRBIGK() 1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
 TRSBND() 1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
        LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()

BIGK(0) = SZERO

IF (TRSBND(0) .GT. SZERO) THEN
   SCLK = SZERO/TRSBND(0)
   DO 100 L = 0, NUMREG
       TRBIGK(L) = BIGK(L) * SCLK
       TRSBND(L) = TRSBND(L) * SCLK
   CONTINUE
ELSE
   TRBIGK(0) = SZERO/FACTR
   MM(0) = MM(1) * ((ALOG (BIGK(1)) - ALOG (TRSBND(0))
   & + ALOG (FACTR)) / (ALOG (SZERO) - ALOG (TRSBND(0))))
ENDIF

IF (IOUT .EQ. 10) THEN
   WRITE(8,*) 'SZERO = ', SZERO, ', BIGK0 = ', TRBIGK(0)
   WRITE(8,*) 'FACTOR = ', FACTR, ', BIGK1 = ', TRBIGK(1)
   WRITE(8,*) 'MM0 = ', MM(0), ', MMO = ', MM(6)
ENDIF

RETURN
END

C******************************************************************************

FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
PROGRAMMER: L. NEWLIN
DATE: 10FEB89
VERSION: MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND,
   & Z0REG, NUMREG, SZERO)

C INPUTS: S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, Z0REG, NUMREG, SZERO
C OUTPUTS: GTLIFE
C IMPLICIT NONE

7 - 337
INTEGER IOUT, L, MAXREG, NUMREG, ZROREG
PARAMETER (MAXREG = 3)
COMMON IOUT
REAL GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), &
MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

LIST OF VARIABLES
GETLIF VALUE TO BE ASSIGNED TO GTLIFE - CYCLES TO FAILURE FOR
IOUT OUTPUT DUMP CONTROLLER
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LNA() 1-D ARRAY CONTAINING VALUES OF LN(A) = M LN K FOR EACH REGION
LNZ NORMAL(0, PVAR) GENERATED RANDOM VARIATE
LPHIM() 1-D ARRAY CONTAINING VALUES OF M LN PHI FOR EACH REGION WHERE
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
S VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO
FAILURE) IS REQUIRED
SBND() 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
CONTAINED IN NBND()
SZERO STRESS TENSILE POINT, SO
TEMP TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
FLOWS
ZROREG ZERO REGION - VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE = 0 - ZERO REGION EXISTS, 1 - NO REGION

GETLIF = 0.0
C CALCULATE CYCLES TO FAILURE
IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
GETLIF = 1.0
ELSE
DO 100 L = ZROREG, NUMREG
IF (S .GT. SBND(L)) THEN
TEM = LNA(L) + LPHIM(L) + MM(L) * ( - ALOG(S) &
+ ALOG (KRATIO) + LNZ)
IF (TEMP .GT. 86.0) THEN
TEMP = 86.0
ENDIF
GETLIF = EXP (TEMP)
GOTO 150
ENDIF
100 CONTINUE
ENDIF
150 CONTINUE
GTLIFE = GETLIF
RETURN
END

C*******************************************************************************

C SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C M FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: 10FEB88

7 - 338
SUBROUTINE SORTM (ALLM, NUMREG, NUM)

INPUTS: ALLM, NUMREG, NUM

OUTPUTS: ALLM

IMPLICIT NONE

COMMON IOUT

INTEGER I, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

PARAMETER (MAXMM = 20001, MAXREG = 3)

LOGICAL INORDR

REAL ALLM(MAXMM, MAXREG), TEMP

LIST OF VARIABLES

ALLM() 2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
I CONTROLS INSERTION POINTER
INC SORT INCREMENT VARIABLE
INORDR FLAG TO INDICATE WHETHER SORT IS FINISHED
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXMM MAXIMUM NUMBER OF M'S TO BE SORTED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
NUM NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
NUMREG NUMBER OF REGIONS OF INTEREST
TEMP TEMPORARY SORTING VARIABLE

DO 400 L = 1, NUMREG

      INC = NUM

  5     IF (INC .GT. 1) THEN

  10    INC = INC / 2

  20   INORDR = .TRUE.

      DO 300 I = 1, (NUM - INC)

            IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN

            TEMP = ALLM(I,L)
            ALLM(I,L) = ALLM(I + INC, L)
            ALLM(I + INC, L) = TEMP
            INORDR = .FALSE.

            ENDIF

  300   CONTINUE

      IF (.NOT. INORDR) GOTO 20
      GOTO 10

   ENDIF

400 CONTINUE

RETURN

END
Reference
